# Table of Contents

About this book

Arduino Uno (85 questions)

Programming (37 questions)

Serial (36 questions)

C++ (21 questions)

Sensors (19 questions)

Pins (15 questions)

Uploading (14 questions)

Library (14 questions)

Timers (13 questions)

Power (12 questions)

ATMega328 (12 questions)

Compile (11 questions)

Shields (11 questions)

Arduino Mega (11 questions)

Arduino Ide (10 questions)

Interrupt (10 questions)

LED (9 questions)

PWM (9 questions)

C (9 questions)

Avrdude (9 questions)

Battery (8 questions)

IDE (8 questions)

Motor (8 questions)

Voltage Level (8 questions)

Attiny (8 questions)

Arduino Yun (7 questions)

Safety (7 questions)

Arduino Nano (7 questions)

AnalogRead (7 questions)

LCD (6 questions)

Arduino DUE (6 questions)

Sketch (6 questions)

Hardware (6 questions)

Bootloader (5 questions)

Time (5 questions)

Software (5 questions)

I2c (5 questions)

Button (5 questions)

Memory Usage (5 questions)

Linux (4 questions)

Clones (4 questions)
Variables (4 questions)
Flash (4 questions)
EEPROM (4 questions)
Sram (3 questions)
Arduino Uno SMD (3 questions)
Temperature Sensor (3 questions)
PROGMEM (3 questions)
Electricity (3 questions)
GSM (3 questions)
Emulation (2 questions)
Enclosure (2 questions)
Tcpip (2 questions)
Oscillator Clock (2 questions)
IR (2 questions)
Web Service (1 question)
Project (1 question)
RFID (1 question)
Osx (1 question)
Copyright
About this book

This book has been divided into categories where each question belongs to one or more categories. The categories are listed based on how many questions they have; the question appears in the most popular category. Everything is linked internally, so when browsing a category you can easily flip through the questions contained within it. Where possible links within questions and answers link to appropriate places within in the book. If a link doesn’t link to within the book, then it gets a special icon, like this.[link]

**Arduino Uno**

**Questions**

**Q: Is an Arduino capable of running 24/7?**

Tags: arduino-uno (Next Q)

I’m making a simple Arduino web server and I want to keep it turned on all the time. So it must endure to stay working continuously.

I’m using an Arduino Uno with a Ethernet Shield. It’s powered with a simple outlet power supply 5V @ 1A.

My Questions:

- Will I have any problems leaving the Arduino turned on all the time?
- Is there some other Arduino board better recommended for this?
- Are there any precautions that I need to heed regarding this?

Tags: arduino-uno (Next Q)

User: butzke

---

**Answer** by sachleen

You shouldn’t have any issues keeping it on all the time, however, some thing to consider is any counters you may have, like the use of `millis()`.

From the [Arduino docs on millis](https://www.arduino.cc/en/Reference/millis):

> This number will overflow (go back to zero), after approximately 50 days.

So for projects that are on for long periods of time, you may not see an issue immediately but something like this could pop up and cause errors down the road.

---

**Answer** by matthew-g.

A couple of things to keep in mind (outside of @Sachleen’s mention of Milli’s):

- Like any electronics, heat can be disruptive. The micro-controller itself isn’t likely going to be a huge issue from the perspective of heat, but other components like the power supply might cause issues.
- If your code uses `EEPROMWrite`, be aware that the EEPROM is only rated for something in the neighbourhood of 100,000 writes.

---

**Answer** by thedoctor
Keep in mind that the flash and EEPROM have limited lifetimes (about 10,000 read/write cycles) so if you’re doing a lot of reading and writing to those, they may become corrupted. In a test I did, an external EEPROM took about 3 days to start becoming corrupted.

**Tags: arduino-uno (Next Q)**

---

**Q: How to pause Arduino for 1 millisecond through MATLAB?**

**Tags: arduino-uno (Prev Q) (Next Q)**

I have been able to connect MATLAB to my Uno with this line of code: a = arduino('COM4'); through this package: [Matlab support package for Arduino](https://www.mathworks.com/products/matlab-support-for-arduino.html).

Right now, MATLAB is my main script that will synchronize all the components [like the Arduino].

I am able to send pulses to my stepper-motor just by using:

```c
void loop()
{
  digitalWrite(2, HIGH);
  delay(1);
  digitalWrite(2, LOW);
  delay(1);
}
```

This works fine, and will make the motor move about once every 2 milliseconds. My problem is that I cannot find a way to produce this same delay through MATLAB’s interface. I do know that MATLAB has a `pause()` function, but when I set up a loop in MATLAB like this:

```matlab
a = arduino('COM4');
for m = 1:400
  a.digitalWrite(2, 1);
  pause(0.001);
  a.digitalWrite(2, 0);
  pause(0.001);
end
```

Each step takes WAY longer - about 200 milliseconds each.

What are other options for creating the pause between digital High / Low being sent to the Arduino? It would be nice if I could control the outcome from Matlab, through Arduino.

**Tags: arduino-uno (Prev Q) (Next Q)**

**User: radical-fanatic**

**Answer by mpflaga**

The increased in observed delay is due to the time it takes to send the command from MATLAB to the Arduino have its code execute the requested command. In your case the a.digitalWrite. So actual time between pulses using MATLAB’s delay is 0.001 pause plus the time of baud rate times the packet size and decoding then execution.

I would recommend not pursuing this solution of the MATLAB controlling the timing of
Step Motors, remotely. As this latent coupling will be a frustrating limitation. Noting that for step motors the timing can not afford such latent execution. Hence all the 3D printers and many CNC with Step Motors use Arduino’s which get G-Code (motor move commands with destinations) and the Real-time firmware handles the stepping, within tolerances.

I would suggest using something like grbl firmware with step motor shield and then have the MATLAB send the needed G-code. Where there are others with much more features but tailored to 3D Printing, grbl is simpler and straight to the point of moving motors.

Tags: arduino-uno (Prev Q) (Next Q)

Q: How do I figure out if my board is approaching end of life?

I have a uno that I have been using for 3 years now. I will be using it again in a rather critical project in which failure on the part of the board could be rather expensive and dangerous. So, I would like to be sure that the board is not approaching end of life or going to fail anytime soon. Is there any reliable way to figure out how long the board will function without failing or reduction in performance?

Tags: arduino-uno (Prev Q) (Next Q), safety (Next Q)

User: asheesh

Answer by connor-wolf

Unfortunately, there isn’t much way to really determine “wear” in the context of solid-state electronics.

Probably the things that are most likely to fail are the electrolytic capacitors and the connectors.

First, if you’re using an ATmega CPU for something that could possibly injure someone CONTACT ATMEL AND TALK ABOUT SAFETY PRECAUTIONS. The ATmega CPU used in most of the Arduino models is not rated for use in such situations.

In EVERY datasheet:

Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

Now, realistically, this is probably mostly lawyer repellent, but you should still take appropriate precautions.

Really, while there isn’t anything on a common Arduino board that really wears out except the connectors, why are you trying to save $30 at a potential huge cost? Just buy a new board.

I’d also strongly recommend you chose a board with a SMT ATmega328P, since that
removes the IC socket contacts from the list of concerns. If possible, also remove the pin-
headers, and solder wires to the board directly. Try to minimize connectors, since they are
frequent points of failure.

One of the sections of the Arduino that is likely to become unreliable over time is its
memory. There are three pools of memory in the microcontroller used on avr-based
Arduino boards:

- Flash memory (program space), is where the Arduino sketch is stored.
- SRAM (static random access memory) is where the sketch creates and manipulates
  variables when it runs.
- EEPROM is memory space that programmers can use to store long-term information.

The memory is one part of the board that can be checked and verified, and thus evaluated
for reliability/health. A very basic way to check memory would be to write a certain 8-bit
pattern (byte character) over every address in the memory and then read the value present
from every address. If the value that was written matches the value that is read, then that
specific 8 bit block in memory is functioning correctly at the present moment.

Wear in ROM memory usually occurs in a blockwise pattern i.e. n*8-bit blocks become
degraded over time. So, for a 2K byte ROM chip, the health of the chip can be estimated
by writing and reading from every byte on the chip, and calculating the percentage of
correctly functioning blocks. If the percentage of failed blocks is significant (15%-20%),
that means that the memory is likely to fail soon.

The test code can be written using separate methods for each of the memory sections.

**SRAM**

Any variables declared statically or dynamically are allocated on the SRAM. So, we could
declare a large character array (~2000) and fill every element with 255 (all bits 1). Then,
we could attempt to read each of those elements and see if the value being read is indeed
255.

**EEPROM**

The EEPROM can be manipulated using the **EEPROM library**. The library provides
functions to read and write from specific locations in the EEPROM. So, all memory
addresses can be tested by simply looping over the entire memory space. *This operation
will require 500 writes and reads.*

Depending on the board usage, EEPROM is most likely to fail first but is not critical to
board operation.

**Flash**

Data can be stored on the flash memory using the **PROGMEM** directive. Similar to SRAM,
a large array can be declared and initialized here. Then, values can be read and checked.

**Q: Does Arduino Uno R3 require cooling in a closure?**

**Tags:** arduino-uno (Prev Q) (Next Q), enclosure (Next Q)

I was looking at getting an enclosure to house my Arduino Uno R3, but I’m not sure if I should also attach a fan. Does anyone have experience with this? It doesn’t look like much heat is being generated, but with no ventilation except some slits in the enclosure, there won’t be much air circulation without a fan. My concern with adding a fan was the fact that I’d need to power it.

**Tags:** arduino-uno (Prev Q) (Next Q), enclosure (Next Q)

**User:** b.k.

**Answer** by jippie

The only times I had an AVR run hot was when I applied reverse power to it. You’ll be fine without a fan for the Arduino board itself, as long as you don’t run power hungry peripherals from it.

Probably the only thing you really need to check (apart from absolute maximum ratings: 40mA max per IO pin and 200mA max for the whole microcontroller) is the on board voltage regulator (usually three pin device with a heat sink soldered down to the PCB) if you power it from a voltage > 7V or so.

**Answer** by jvarhol

No, You shouldn’t need any type of cooling device such as a fan as long as the case has a few holes in order to allow the heat to rise out of the case. I do not recommend using cases that are completely sealed, unless you are using it in an environment that has things that could hard the Arduino such as ROVs or Nuclear waste facility (although I don’t know why you would be at one in the first place).

**Tags:** arduino-uno (Prev Q) (Next Q), enclosure (Next Q)

**Q: Why can’t I upload a sketch while other components/devices are connected to my Uno?**

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Next Q), pins (Next Q), uploading (Next Q)

I wanted to make a fairly simple circuit which would flash a series of LEDs in sequence, using my Arduino Uno (more specifically, a SainSmart clone). I wrote my sketch and it compiled fine. After that, I connected 8 LEDs+resistors to pins 0 through 7, and then connected the Uno to my computer via USB.
I’ve uploaded sketches successfully in the past, so I’m sure my settings and drivers etc. are correct. However, when I tried to upload my sketch this time, it didn’t work.

I tried removing everything I’d connected to the Arduino’s pins, and suddenly the upload worked again.

Why does this happen? Does it mean I have to disconnect everything from the board every time I upload a sketch?

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Next Q), pins (Next Q), uploading (Next Q)

**User:** peter-r.-bloomfield

---

**Answer** by peter-r.-bloomfield

The problem is specifically pins 0 and 1. Although they can be used as regular digital IO pins, they also serve as the RX and TX pins for the Uno’s serial port. The USB connection (for uploading sketches etc.) is routed to the same pins internally. Unfortunately that means anything connected on pins 0 and 1 can interfere with the serial connection, preventing communication via USB.

In short, it’s not necessary to disconnect *everything* when uploading a sketch. It should only be necessary to disconnect anything from pins 0 and 1. Rather than going through that hassle every time a sketch is uploaded though, it may be best just to avoid using those pins unless necessary (e.g. you run out of other pins, or your project needs a serial connection to another device).

---

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Next Q), pins (Next Q), uploading (Next Q)

---

**Q: Is there a limit on how much current a pin can sink?**

**Tags:** arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q)

I’ve been experimenting with an 8x8 LED matrix controlled by an Arduino Uno. As I think is typical, the matrix uses a common anode for each row, and a common cathode for each column.

At the moment, I’ve got all the matrix pins connected directly to IO pins on the Uno, and I haven’t had any problems lighting the LEDs one-by-one. Going through the whole matrix like this makes updates a little slow though, meaning the LEDs aren’t as bright as I’d like.

As far as I know, I can’t safely light up an entire row at once, because an individual pin on the Uno isn’t capable of sourcing enough current to drive 8 separate LEDs at the same time (requiring at least 10mA each to be bright enough).

It occurred to me that the reverse might be safer. If I light up an entire column at a time, then each pin only has to source enough current for one LED, which should be no problem. However, it relies on one pin potentially sinking the current from all 8, totalling at least 80mA.

Is this possible, or am I going to fry my board?
am I going to fry my board?

Pretty likely outcome. Here’s why:

The Arduino Uno uses the atMEGA328 microcontroller, which has an absolute maximum rating of 40 mA source or sink per GPIO. Also, the total current through the supply or ground rails (i.e. the total of all current OP wants the GPIO pins to sink, or source) is rated to a maximum of ±50 200 mA.

Realistically, the current graphs in the atMEGA328 datasheet do not even go all the way to 40 mA, hence it is safest to restrict the current to perhaps 20 mA total per GPIO.

In other words, whether a GPIO is used as source or sink, the limits remain pretty low, and in practice, should be kept much lower than the absolute maximum ratings. Even below absolute maximum, running a microcontroller to too high a temperature reduces its life expectancy.

Solution:

Use either BJTs, MOSFETs, or some driver IC such as the ULN2003 to actually drive the current through the LEDs.

Q: Is there a way to have more than 14 Output pins on arduino?

Is it possible to have more than 14 output pins on the Arduino, I am working on a project in which I need to light up several LEDs individually. I only have an Arduino Uno, and I don’t want to get a Mega.

A common way to expand the set of available output pins on the Arduino is to use shift registers like the 74HC595 IC (link to datasheet).

You need 3 pins to control these chips:

1. Clock
2. Latch
3. Data
In a program, you pass on the data one bit at a time to the shift register using the `shiftOut()` command, like so:

```c
shiftOut(dataPin, clockPin, data);
```

With that command, you set each of the 8 outputs on the 595 IC with the 8 bits in the data variable.

With one 595, you gain 5 pins (8 on the IC, but you spend 3 to talk to it). To get more outputs, you can daisy-chain a series of 595 together, by connecting its serial-out pin, to the data pin of the next one. You also must connect together the clock and latch pins of all of the 595 ICs.

The resulting circuit (using one 595) would look like this:

![Circuit Diagram](http://www.codeproject.com/Articles/144606/Arduino-Platform-Working-with-Shift- Registers)

The latch pin is used to keep the 595 outputs steady while you are shifting out data into it, like so:

```c
digitalWrite(latchPin, LOW);
shiftOut(dataPin, clockPin, data);
digitalWrite(latchPin, HIGH);
```

Answer by jvarhol

There are two ways you can get more pins out of an Arduino.
The first way is by using the Analog pins as digital output pins, which is really easy to do. All you need to do is refer to A0-A5 as pins 14,15,16,17,18,19. For example to write high to pin A0 just use digitalWrite(14, HIGH).

The other way to get more pins out of the Arduino is by using a Shift Register. To do this I recommend using the EZ-Expander Shield, which allows you to use digitalWrite([20-35], HIGH) when you import the EZ-Expander Library. This shield however only allows the pins to be used as outputs only and uses pins 8,12 and 13 to control the shift registers.

The great thing is, is that you can use both of the two methods above together, without any problems.

Answer by jfpoilpret

If you want to drive LEDs, then you can also use a MAX7219 that can drive 64 LEDs, without extra circuitry (no need for transistor to amplify signal).

Driving a MAX7219 requires only 3 output pins on Arduino. Also, you can find a few Arduino libraries for it.

You can also chain several of them if you need to power more than 64 LEDs.

I have used it successfully for multiple 7-segment LED displays.

Downside: it is expensive (about $10).

Tags: arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q)

Q: Is my ATmega broken, and would replacing it with another break that one, too?

Tags: arduino-uno (Prev Q) (Next Q), atmega328 (Next Q)

I was communicating between my computer and my Arduino Uno R3 through Bluetooth using this Bluetooth module. The connector I use has a red wire where it connects to GND, and a black wire where it connects to +5V.
I forgot this one time when I connected the Bluetooth module to the Uno, so the polarity was reversed. This fried the Bluetooth module.

After getting a new module, I discovered that the Uno was apparently fried too. Plugging a
USB cable into the Uno made the little green integrated LED turn on, but that was all that happened; none of the pins put out any electricity. When I tried to upload a program, it gave the infamous `avrdude: stk500_getsync(): not in sync: resp=0x00` error. I could upload programs to my other Uno fine, and I could not upload to the broken Uno on my other computer.

Two of the Arduino experts I know told me that the ATmega328 chip might be the only broken part. Is this likely to be the case? I have an Uno R2 with a working ATmega328. Would temporarily replacing the R3’s ATmega with that R2’s ATmega have a change of damaging anything?

**Tags:** [arduino-uno](https://examples.com/arduino-uno) ([Prev Q](https://examples.com/arduino-uno)) ([Next Q](https://examples.com/arduino-uno)), [atmega328](https://examples.com/atmega328) ([Next Q](https://examples.com/atmega328))

**User:** the-guy-with-the-hat

---

**Answer** by jfpoilpret

Besides breaking a pin while removing the IC from its socket, or static electricity during manipulation, I don’t see what could fry the ATmega.

However, it is possible that the R2 ATmega has a different bootloader and that may be an issue.

If I was you I would just remove the ATmega from the R3 board and test it on a breadboard (don’t need a lot of components to do that); at least that could tell you if it is fried or if it is the R3 board that is fried…

**Tags:** [arduino-uno](https://examples.com/arduino-uno) ([Prev Q](https://examples.com/arduino-uno)) ([Next Q](https://examples.com/arduino-uno)), [atmega328](https://examples.com/atmega328) ([Next Q](https://examples.com/atmega328))

---

**Q:** What are the AREF, IOREF, and the unlabeled pin next to IOREF on the Uno R3?

**Tags:** [arduino-uno](https://examples.com/arduino-uno) ([Prev Q](https://examples.com/arduino-uno)) ([Next Q](https://examples.com/arduino-uno)), [pins](https://examples.com/pins) ([Prev Q](https://examples.com/pins)) ([Next Q](https://examples.com/pins))

**User:** tbodt

---

**Answer** by thedoctor

**AREF:**

This is the voltage reference analog-to-digital converter (ADC). It can be used instead of the standard 5V reference for the top end of the analog spectrum – for example, if you…
wanted to use the ADC to monitor a signal that had a 0-1.5 volt range you could get the full scale of the ADC by connect AREF to a 1.5 V signal.

**IOREF:**
This is a voltage corresponding to the i/o of that board, for example an Uno would supply 5v to this pin, but a Due would supply 3.3v

**Unlabeled pin:**
This pin is unused, but is there to ensure compatibility with future products. It is not connected to anything on any R3 boards I have seen.

---

**Answer** by klaus-warzecha

Concerning the AREF pin, you have to remember that the Arduino comes with a 10bit ADC (Analog-Digital-Converter), which converts incoming voltages between 0V and 5V to integer values between 0 and 1023. This results in a resolution of roughly 4.8 mV.

If a sensor only delivers a lower maximum voltage, it is resonable to apply this voltage to the AREF pin, just in order to obtain a higher resolution.

---

**Q: What is the difference between the revisions of the Arduino Uno board?**

**Tags:** arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q)

There are three different revisions of the Arduino Uno. What changed in each revision?

**Tags:** arduino-uno (Prev Q) (Next Q)

**User:** tstew

---

**Answer** by asheeshr

**User Side Changes**

- Both revision 2 and 3 boards add four solder pads (JP2) connecting to pins PB4 to PB7 of the USB ATMEGA.
- Revision 2 and 3 boards are both supplied with header pins in the USB ATMEGA ICSP header rather than just solder pads in the Arduino Uno.
- The revision 3 board changes the 8 pin connector that contains Arduino pins 8 to 13, GND and AREF for a 10 pin connector. The extra 2 pins are connected to AD4/SDA and AD5/SCL. These are the two analog input pins that can be used for I2C.
- Revision 3 boards change the 6 pin connector that has the reset pin connected to it to an 8 pin connector. One of the new pins on this header is the IOREF pin that allows shields connected to the board to adapt to the voltage of the board. The second pin is reserved for future use.
Electronics Based Changes

- The Arduino Uno and Arduino Uno revision 2 both have an ATMEGA8U2 USB microcontroller on board – this is upgraded to an ATMEGA16U2 on the revision 3 board.
- Revision 3 adds a diode across the USB ATMEGA reset pin pull-up resistor.
- Both revision 2 and 3 boards add a 1k pull-down resistor to the DTR (HWB) line coming from the USB ATMEGA microcontroller – from the PD7 pin.
- The Arduino Uno and Arduino Uno revision 2 both have a LED and resistor connected in series on Arduino pin 13. The revision 3 board buffers this LED/resistor through a unity gain op-amp. This is the spare op-amp that was unused on previous boards.

Sources:

1. [http://startingelectronics.com/articles/arduino/uno-r3-r2-differences/](http://startingelectronics.com/articles/arduino/uno-r3-r2-differences/)

Tags: [arduino-uno](http://www.example.com) (Prev Q) ([Next Q](http://www.example.com))
Q: What happens when I call exit() from my program?

Tags: arduino-uno (Prev Q) (Next Q), programming (Next Q)

In either of setup or loop, if I were to add an exit(0) call, where would control be passed to? What would the next state of the microcontroller be? Would it stop execution and power down?

I am using a revision 2 Arduino Uno.

Tags: arduino-uno (Prev Q) (Next Q), programming (Next Q)

User: asheeshr

Answer by mpflaga

My initial guess is wrong. I would have thought it would simply return from loop and the core library would just call loop() again. However, I see the following code was created. Noticing that __stop_program is a hard loop...

An extract of Blink.ino’s listing, with exit(0) added:

```c
void loop()
{
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
  exit(0);
}
```

The disassembly of the above:

```text
Skip code block
```

```c
void loop()
{
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage level)
  leds[0] = 0x91 00 01     // turn the LED on (HIGH is the voltage level)
    lds r24, 0x0100
  r0:  61 e0
    ldi r22, 0x01 ; 1
  r0:  0e 94 ca 01  call  0x394 ; 0x394 <digitalWrite>
    delay(1000);          // wait for a second
  r0:  68 ee
    ldi r22, 0xE8 ; 232
  r0:  73 e0
    ldi r23, 0x03 ; 3
  r0:  80 e0
    ldi r24, 0x00 ; 0
  r0:  90 e0
    ldi r25, 0x00 ; 0
  r0:  0e 94 f7 00  call  0x1ee ; 0x1ee <delay>
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  r0:  80 91 00 01     // turn the LED on (HIGH is the voltage level)
    lds r24, 0x0100
  r0:  60 e0
    ldi r22, 0x00 ; 0
  r0:  0e 94 ca 01  call  0x394 ; 0x394 <digitalWrite>
    delay(1000);          // wait for a second
  r0:  68 ee
    ldi r22, 0xE8 ; 232
  r0:  73 e0
    ldi r23, 0x03 ; 3
  r0:  80 e0
    ldi r24, 0x00 ; 0
  r0:  90 e0
    ldi r25, 0x00 ; 0
  r0:  0e 94 f7 00  call  0x1ee ; 0x1ee <delay>
    exit(0);
  r0:  80 e0
    ldi r24, 0x00 ; 0
  r0:  90 e0
    ldi r25, 0x00 ; 0
  r0:  0e 94 1e 02  call  0x43c ; 0x43c <_exit>
...
Note that if _exit had not called cli, interrupts would be able to do stuff. But that is not the case.

**Answer** by jamolnng

Well I just tested it with my Arduino Uno and it just completely stopped the code and left all the outputs as they were when the code stopped running (so it left an LED I had on on). There seems to not be a IO cleanup when you call exit. This was what I expected because the Arduino IDE provides the setup and loop functions, if you program the ATMEGA*28 with with any other AVR IDE you start with the main function like all C/C++ programs. The setup and loop functions are not standard on AVR MCU’s.

**Note:** The press of the reset button restarts the code, if you were wondering.

**Tags:** arduino-uno (Prev Q) (Next Q), programming (Next Q)

---

**Q: Why do some pins have a different PWM frequency?**

**Tags:** arduino-uno (Prev Q) (Next Q), pwm (Next Q)

According to the [Arduino reference for analogWrite()](https://www.arduino.cc/en/reference/analogWrite), the PWM frequency on most pins is ~490 Hz. However, it’s ~980 Hz for pins 5 and 6 on the Uno, and for pins 3 and 11 on the Leonardo.

Why are these different? Is it a deliberate design feature, or is it somehow dictated by the hardware?

**Tags:** arduino-uno (Prev Q) (Next Q), pwm (Next Q)

**User:** peter-r.-bloomfield

**Answer** by ricardo

Those aren’t the only frequencies available for the PWM signals. However, they are the frequencies as determined by the applied prescaler (which you can readily change as detailed below).

Each of the 3 pairs of PWM pins is tied to one timer, each of which has its own base frequency, as follows:

- Pins 5 and 6 are paired on timer0, with base frequency of 62500Hz
- Pins 9 and 10 are paired on timer1, with base frequency of 31250Hz
- Pins 3 and 11 are paired on timer2, with base frequency of 31250Hz

Then each set of pins have a number of prescaler values that can be chosen, that will divide the base frequency of that pair of pins. The prescaler values available are:

- Pins 5 and 6 have prescaler values of 1, 8, 64, 256, and 1024
- Pins 9 and 10 have prescaler values of 1, 8, 64, 256, and 1024
- Pins 3 and 11 have prescaler values of 1, 8, 32, 64, 128, 256, and 1024

The different combinations yield different frequencies in a given PWM pin. Notice that timer 2 (tied to pins 3 and 11) have more prescaler values available, resulting in more frequencies available.

Now, why timer 2 is different, that's a separate question.

Edit: Here's a list of possible PWM frequencies per pin (from this article):

For pins 6 and 5 (OC0A and OC0B):

- If TCCR0B = xxxxx001, frequency is 64kHz
- If TCCR0B = xxxxx010, frequency is 8 kHz
- If TCCR0B = xxxxx011, frequency is 1kHz (this is the default from the Diecimila bootloader)
- If TCCR0B = xxxxx100, frequency is 250Hz
- If TCCR0B = xxxxx101, frequency is 62.5 Hz

For pins 9, 10, 11 and 3 (OC1A, OC1B, OC2A, OC2B):

- If TCCRnB = xxxxx001, frequency is 32kHz
- If TCCRnB = xxxxx010, frequency is 4 kHz
- If TCCRnB = xxxxx011, frequency is 500Hz (this is the default from the Diecimila bootloader)
- If TCCRnB = xxxxx100, frequency is 125Hz
- If TCCRnB = xxxxx101, frequency is 31.25 Hz

TCCRnB is where you set the prescaler bits for timer n, replacing n by 0, 1 or 2, depending on the timer you want to set. If you are still unsure about bitwise operations, read this bit math tutorial.

My sources:

- [http://playground.arduino.cc/Code/PwmFrequency](http://playground.arduino.cc/Code/PwmFrequency)

Note that there seems to be divergence in those sources about whether pins 9 and 10 have the same behavior as 5 and 6 or 3 and 11, but you get the idea anyway. I’m reading the datasheet to try and figure out which is correct, or whether this is a difference between boards.

**Answer by jippie**

I am not aware of the design considerations, but if you check the datasheet for the microcontroller on your Arduino, you will notice that PWM pins are grouped together and
per group connected to a timer. The speed at which this timer is increased varies by the configured prescaler. If you change the prescaler for a certain timer, you change the PWM frequency for the related PWM pins. I believe some timers double for other purposes like the `millis()` function. If you change the prescaler for that timer, values returned by `millis()` will be off by the same factor.

You can calculate the setting for the prescaler as follows:

$$\text{prescaler} = \dfrac{f_{CPU}}{\text{PWMresolution} \times f_{PWM}} = \dfrac{16\text{MHz}}{256×490} \approx 128$$

Check the datasheet and you’ll find that 128 is indeed one of the prescaler values you can select.

---

**Q: Is it possible to program an Arduino Pro Mini using an Arduino Uno R3?**

_Tags:_ [arduino-uno](#) ([Prev Q](#)) ([Next Q](#)), [pwm](#) ([Next Q](#))

I’ve recently received an Arduino Pro Mini as a gift, and I don’t have an external USB programmer to program the Pro Mini. However, I do own an Arduino Uno R3, and I think it would be possible because they both have the same clock speed and micro-controller (16Mhz with an atmega328).

Is it possible to program an Arduino Pro Mini using an Arduino Uno? If yes, how does one do so?

_Tags:_ [arduino-uno](#) ([Prev Q](#)) ([Next Q](#)), [uploading](#) ([Prev Q](#)) ([Next Q](#))

**User:** zeldarulez

**Answer** by sachleen

Just hook up the pins (RX, TX, reset, Power, and Ground) to the corresponding pins on your Uno.

You may have to take the ATmega328 out of your Uno for this to work.

Keep in mind that you need to hook it up to the correct power pin! If your pro mini is running at 3.3v then you must hook it up to the 3.3 pin on your Uno.

Check out the [ArduinoToBreadboard](#) page for a bit more info. Scroll down to “Uploading Using an Arduino Board”. Although it shows you how to program an Arduino on a breadboard, the wiring is the same.

_Tags:_ [arduino-uno](#) ([Prev Q](#)) ([Next Q](#)), [uploading](#) ([Prev Q](#)) ([Next Q](#))
Q: How can I create multiple running threads?

Tags: arduino-uno (Prev Q) (Next Q)

Is there a way I can have multiple parts of the program running together without doing multiple things in the same code block?

One thread waiting for an external device while also blinking a LED in another thread.

Tags: arduino-uno (Prev Q) (Next Q)

User: bja

Answer by sachleen

There is no multi-process, nor multi-threading, support on the Arduino. You can do something close to multiple threads with some software though.

You want to look at Protothreads:

Protothreads are extremely lightweight stackless threads designed for severely memory constrained systems, such as small embedded systems or wireless sensor network nodes. Protothreads provide linear code execution for event-driven systems implemented in C. Protothreads can be used with or without an underlying operating system to provide blocking event-handlers. Protothreads provide sequential flow of control without complex state machines or full multi-threading.

Of course, there is an Arduino example here with example code. This SO question might be useful, too.

ArduinoThread is a good one too.

Answer by jippie

AVR based Arduino’s do not support (hardware) threading, I am unfamiliar with the ARM based Arduino’s. One way around this limitation is the use of interrupts, especially timed interrupts. You can program a timer to interrupt the main routine every so many microseconds, to run a specific other routine.


Answer by asheesh

It is possible to do software side multi-threading on the Uno. Hardware level threading is not supported.

To achieve multithreading, it will require the implementation of a basic scheduler and maintaining a process or task list to track the different tasks that need to be run.

The structure of a very simple non-preemptive scheduler would be like:

```c
//Pseudocode
void loop()
{
```
for(i=0; i<n; i++)
run(tasklist[i] for timelimit):
}

Here, tasklist can be an array of function pointers.

tasklist [] = {function1, function2, function3, ...}

With each function of the form:

int function1(long time_available)
{
  top:
   //Do short task
   if (run_time<time_available)
     goto top;
}

Each function can perform a separate task such as function1 performing LED manipulations, and function2 doing float calculations. It will be the responsibility of each task(function) to adhere to the time allocated to it.

Hopefully, this should be enough to get you started.

Tags: arduino-uno (Prev Q) (Next Q)

Q: How can an Arduino output a specific (i.e. 56khz) carrier frequency?

Tags: arduino-uno (Prev Q) (Next Q), timers (Next Q), ir (Next Q)

I’m working on a free space optics project to send data wirelessly between two points. To accomplish this I’m using an IR LED connected to an Arduino Uno that pulses with a 56kGhz carrier frequency for the transmitter and a second Arduino with a 56kGhz IR detector module for the receiver.

I tried using delayMicroseconds() between pin high and pin low commands in order to create the carrier frequency. This kind of works, but the frequency isn’t always the same and any additional delays for pulsing the signal (i.e. the time required to call the function and decrementing) can change it.

Reading the datasheet for the Atmega328 it appears that there is a way to set a more accurate pulse using the chips timers. Is that possible, and if so, how do you create a 56kGhz pulse using the timers?

Tags: arduino-uno (Prev Q) (Next Q), timers (Next Q), ir (Next Q)

User: jlbnjmn

Answer by jfpoilpret

It is indeed possible to generate a 56KHz signal with an Arduino timer.

A timer actually can be seen as a special register, in the MCU, that holds a value (starting at 0) that gets incremented at a frequency that is the MCU clock frequency (16MHz on
UNO), possibility divided by a factor called *prescaler*. When that value reaches a limit, called **Compare Match**, that you specify, then 2 things happen:

- the timer register value is reset to 0
- one **ISR** (Interrupt Service Routine) callback function gets called (you can define it to point to your own code).

The idea is to use that **ISR** to change the output of a logical pin every time it is called (HIGH, then LOW, then HIGH...)

Now, in order to generate a 56KHz square wave, you’ll need your **ISR** to be called 56000 * 2 times per second (* 2 because you need to change the output value twice per period).

You can choose the prescaler value you want for a timer among the following list:

- 1 (clock frequency is not divided, hence 16MHz)
- 8 (clock frequency is divided by 8, hence 2MHz)
- 64
- 256
- 1024

There are 2 sizes of timers/counters on UNO (they are called *timer/counter* actually): 8 bits and 16 bits.

On UNO (Atmega328P), you have 3 timers overall, but some may be used by the Arduino core library or other libraries used in your sketches (you’ll have to check that by youself):

- timer0 (8-bit)
- timer1 (16-bit)
- timer2 (8-bit): this one has more prescaling options (1, 8, 32, 64, 128, 256, 1024)

Now you need to generate a 56KHz wave from 16MHz, hence, without prescaling, you would need to count to:

\[
16000000 / (56000 \times 2) - 1 = 141.857 \quad (\text{- 1 because a timer counts from 0 to this value and resets only after it has been reached})
\]

From this calculation, we can draw 2 observations:

1. 141.857 is not an integer and thus you won’t be able to generate a wave of exactly 56KHz
2. Without prescaling, you need a 16-bit timer as 285 is not representable as an 8-bit unsigned integer

From now you have 2 options:

1. Use a 16-bit timer **(timer1)**, use prescaler = 1, and select 142 as the Compare Match; that will give you the following frequency: \(16000000 / (2 \times (142 + 1)) = 55944Hz\)
2. Use an 8-bit timer **(timer0)**, use prescaler = 8, and select 17 as the Compare Match;
that will give less precision with the following frequency: \( \frac{16000000}{(8 \times 2 \times (17 + 1))} = 55555\text{Hz} \) which is still within the required range.

Now, regarding how to write your sketch for that, I advise you to check out this [instructable](https://www.instructables.com) which is very complete and very interesting to read.

Of course, [ATmega328P complete datasheet](https://www.siliconix.com) is also important if you want to understand, in the slightest details, what you are doing.

Some important notes:

- an ISR is executed with disabled interrupts and must thus be as short as possible. In particular, there are several functions from the Arduino library that shall not be called from an ISR.
- Arduino UNO clock is not very accurate (it uses a ceramic resonator instead of a quartz, which would have been much more accurate), so this means the output frequency will shift further.

---

**Answer** by [peter-r.-bloomfield](https://www.instructables.com)

I’ve found [tone()](https://www.instructables.com) useful for generating high frequency pulses on any pin. It should be able to handle 56 KHz. *(Edit: As noted by jfpoilpret, the closest you could actually get on a 16 MHz Arduino is about 55.944 KHz)*

The difficulty will obviously be combining it with your data signal. I don’t think you could do that in software without resorting to low level code. It should be pretty easy in hardware though, since it’s digital.

All you’d need to do is output your data signal on a different pin, and then combine it with the carrier using an AND gate. The combined signal can go straight to your IR transmitter.

If you don’t have an AND gate handy then it’s quite simple to make your own using a pair of transistors. Just search online for “transistor and gate”.

---

**Answer** by [zzz](https://www.instructables.com)

The accepted answer of jfpoilpret is very well written, perfectly valid and in 99% of the cases I will do exactly what he explains. His solutions are well within your defined parameters, so they should work very well. But what is better than “very well“?

**Perfection!** After all, question is about generating a exact value. As said close enough is good in most cases (arguably all), and even when dealing with something as a clocks when 1 second need to be 1 second, you still have to suffer inherited parts imperfections.

What I will suggest is not always possible. In some cases, it’s possible but with much more hassle and effort than this case. Is it worthy depend on case by case basis. My goal is mostly to show a alternative for future references, that is better in somewhat fringe cases. This is written with novice Arduino users in mind, that don’t have extensive experience in electronics. For more advanced people this probably will look too verbose and dumbed down. But I believe that, those same people probably already know it and don’t need this answer. This is also applicable to every MCU and every manufacturer and architecture. But
for other MCU’s you will need to consult the correct datasheet to find out proper registers and prescale names and values.

In your case, you need a specific frequency and the nice thing about it, is that exactly 56 Khz actually can be achieved very easy (not counting practical imperfections of the parts). So this is also a perfect example case.

Generating a signal depends on the timers and clock source of MCU, as explained well by jfpoilpret. His answer deal with the problem of one point of view only and that is fiddling with timers. But you can fiddle with clock source too, or even better with both for synergy and awesome results. By changing the parameters of the environment, in this case hacking the system and replacing the clock source, we can deal with specific problem with much, much more ease and simplicity.

First to remind, because of toggling the pin state, you need to execute ISR two times more than signal frequency. This is 112 000 times per second. 56 000 and 16 000 000 don’t add up very nicely as pointed already. We need to change either the signal frequency or tact frequency. Let’s deal for now with immutable signal frequency and find a better clock speed. It would be most strait-forward to choose a clock with some order of magnitude bigger than 56 (or 112 but is practically the same), as you only add zeros and this kind of math is simplest for most people. Unfortunately everything in this world is some kind of compromiss with something. Not every value will work.

First example is with too low tact generator speed. If you choose a 56 000 hz clock you won’t be able to do anything as you will need to call the ISR every cycle and can’t do anything else. It is utterly useless. If you choose 10 times faster speed (560 Khz), you will have 9 (10 cycles for timer to reach it’s max value - 1 cycle to call ISR function) mcu cycles to do your work and this quite possible can be not enough. You simply often need more computational power.

If you choose value far too great at other hand, as 56 Mhz the MCU simply can’t work with it. It is way too fast. So, simply choosing biggest value in the shop won’t cut it either.

Original Arduino UNO R3 have a stock clock at 16 Mhz, so anything slower that that is guaranteed to work. Next value that is order of magnitude bigger than 56 and lower than 16 Mhz is 5,6 Mhz (5.6 Mhz for USA citizens). This will lead to be able to call the ISR every 50 cycles and will create perfect 112 000 hz timer frequency. And your signal will be exactly 56 Khz. You will have 49 MCU cycles to execute your program between ISR calls, but it is still around 1/3 of the speed of the original clock. One can use 112 as base and use 11,2 Mhz clock and this will give a about 2/3 of the stock 16 Mhz resonator. ISR function will be called every 100 cycles and still generating perfect 56 Khz signal.

However two major problems exist with this values.

- First problem severity depend on your needs: You sacrifice about 1/3 (with 11,2 Mhz) of your maximum computational power to get exact signal frequency that use easy to find register value ( OCR iirc). You may be fine with it or you may not.

- Second problem is a hard showstopper: This very easy to find values, very often simply do not exist as manufactured clock source. This is Farnell resonator web page, that simply lack both 5,6 Mhz and 11,2 Mhz.
To circumvent this we can look at available resonator values and find out something else that can be used to generate exactly desired values. If we divide 56 by 4 we get 14 and luckily there is a 14 Mhz resonator. This provide us with much higher speed and more power and with equally easy to find register value. To call the ISR 112 000 times per second we need to put value of decimal 124 or heximal 0x7C in OCR register so with counting 124 cycles + 1 for calling ISR, we get our desired perfect value.

N.B.

1. ISR - Interrupt Service Routine (this is the code that is executed only on generated interrupts)
2. How big your program can be, depend ot the memory size! Nothing to do with clock speed. Nothing to do with how often you call ISR.
3. When MCU start with program command a counter is incremented, if interrupt is generated ,the ISR is called and this value is stored in special register. When ISR code complete, value of the program counter is restered from this special register and the program continue from where it was interrupted as if it was never happend. I will give a extremly dumbed down example, if you a purist, I warn you: Nose and eye bleeding can occur.
   Imagine you have to walk from somewhere to somewhere. The step by step route instructions are your main program and it’s commands. How fast you walk or run, depend on your “clock speed”, but not on the route instructions (30 steps forward, 1 turn 90 grad. left, 10 steps forward, 45 grad. right, etc.) They are always the same. Now imagine a little kid or greedy corrupt local politician untie your shoes now and then. This is the event that generate a interrupt. Then you halt after your last step, kneel and tie your shoe again. This is your ISR program. Then you continue from the place you have stopped, you do not start from beginning. When you walk without a care in the world and with all the time, you don’t care even if you have to tie your shoe every other step. If you however do it with time constrains, like running in 100 metters on olimpics (or running from hungry flesh eating predator), stopping and tieing your shoes can have dire consequences. The same is with MCU’s. Even if you execute only 1 line of code your program will continue, albeit slow. If you don’t care about speed at all, it won’t be a problem. If you have to do some time related, like using other timer dependant actions, interferance can be very unwanted and problematic.
4. Less is more! Not always faster clock is better. Slower clocked devices use considerably less power. This can be crucial point in battery operated device.
5. Needed cycles are derived from this formulae:
   \[
   \text{(clock speed} / (\text{prescaler value} \times \text{needed ISR calling frequency}) \text{)} - 1
   \]

---

Tags: arduino-uno (Prev Q) (Next Q), timers (Next Q), ir (Next Q)

**Q: How can a 5v Uno talk to a 3.3v Due?**

Tags: arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), pins (Prev Q) (Next Q),
I currently have an Uno (which operates at 5v), and I’m looking at buying a Due (which only operates at 3.3v). I’d like to make them communicate via Serial or SPI or similar, but presumably connecting them directly isn’t an option (I guess the Due’s pins would get damaged).

Is there any way to make the connection safe? Or is there an alternative form of communication I could safely use between them instead?

Answer by jippie

- For a 5V logic output to a 3V3 logic input, you can use a resistive divider to lower the voltage.
- When unloaded, a 3V3 logic output is just enough to drive a 5V logic input. Check the AVR datasheet for the exact voltages (0.6 × Vcc = 3V, found under DC Characteristics in the datasheet).

In other words, with a little bit of special care it may just work.

To construct a more reliable and bidirectional solution, use a level shifter. There are several listed on this page but other manufacturers make similar devices.

This document describes how a MOSFET can be used to bidirectionally convert logic levels between different supply voltages.
accomplished?
I currently have a bare Arduino Uno.

Tags: arduino-uno (Prev Q) (Next Q)

User: jlbnjmn

---

Answer by sachleen

You have a few options for connecting your Arduino to the network/Internet.

**Ethernet**

Something like the Arduino Ethernet Shield allows you to plug in an Ethernet cable from the wall or router into your Arduino. Obviously, the main limitation is that your device is now tethered by the cable. For outdoor use, I wouldn’t do this.

**WiFi**

The Arduino WiFi Shield allows you to connect to your home WiFi network. This is just like the Ethernet except its now wireless.

**RF**

If you have a lot of sensors or other devices that need to communicate with each other, the best option is usually an RF module. You have many options here, XBee being one of them. Check out the Sparkfun XBee Buying Guide to look at all the options available. And that’s just XBee. There are many other wireless options available, at all sorts of prices.

The thing with RF is that none of these will connect to the Internet. You will have all your devices communicate with each other or a base station, which will then be connected to the network by either a WiFi or Ethernet module.

**Wireless Router Serial**

Depending on what kind of wireless router you use, you can have the Arduino communicate directly with it and use that as your connection to a network.

- Arduino - Cheap wifi connectivity
- Converting your Ethernet Shield to a wireless shield

---

Answer by madivad

I’ve been working with xbee on the arduino and raspberry pi for a few months now. Whilst it does have it’s issues and quirks, it is a great tool in the communication chain. It’s not perfect but given the mesh side of it, for me it reaches further than my wifi and can
even be more reliable (my raspberry pi with wifi occasionally drops out, however the connected xbee continues to function in the background.

Specifically in my case it allows for a serial connection between my devices (being several arduinos, a raspberry pi (coordinator) and my mac (used primarily for monitoring but also occasionally injecting serial data)). In this case it’s perfect. It doesn’t allow internet access or access beyond the xbee framework, but that suits me perfectly since implementation is as simple as Serial.print and Serial.read with no Ethernet overhead.

---

**Answer** by **suneesh**

ESP8266 is a cheap 5$ wifi module with UART, SPI connectivity. This can be integrated with Arduino seamlessly either through the stock firmware using AT commands, or by writing custom firmware using the SDK provided and GIPO available. For more reference http://www.esp8266.com/

---

**Tags:** arduino-uno (Prev Q) (Next Q)

---

**Q: Arduino as USB HID**

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q)

Is it possible to build a HID device (like a keyboard) using an Arduino uno?

At the time being, I have button inputs on the Arduino giving outputs on the serial line. So, how can I transform my current firmware into something that can behave like a HID keyboard?

- Shall I write a new window device driver?
- Shall I create a software layer that could take my serial data and have windows read it as HID input?
- Is there any better way to achieve this, if it is possible at all?

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q)

**User:** anomaly

---

**Answer** by **zmo**

Is it possible to build a HID device (like a keyboard) using an Arduino uno?

yes it is! And there’s a great tutorial on how to do that over here

At the time being, I have button inputs on the Arduino giving outputs on the serial line. So, how can I transform my current firmware into something that can behave like a HID keyboard?

As the tutorial shows you, you need to transform your firmware code so instead of printing the events, you actually trigger events. What you need to do, is add to your code:
What you’re doing here, is send from your main Arduino microcontroller events over the serial line to the smaller microcontroller that handles USB communication. That microcontroller will take those key “buffers” and transfer them as USB HID Keyboard key presses over USB.

Beware that the key character (chr in the code here), is actually a keycode sent over USB that gets interpreted by your USB HID driver, which then translates it into a real character using the current layout you’ve setup for your keyboard.

The mod character is one of the modifier keys you have on the keyboard (shift, control, alt…), 0x00 being no modifier.

- Shall I write a new window device driver?

No, you don’t need! With the DFU upgrade, your arduino will be seen as a USB keyboard by windows.

- Shall I create a software layer that could take my serial data and have windows read it as HID input?

You neither need to do that, and it’s actually a very bad idea for many reasons:

- your solution would not be portable, and thus you’d need to hack something for every platform you may use,
- your solution would imply a lot of new code ; the more code one write, the more bugs is likely to be in your system,
- your solution would be really hackish, as it would behave as a device driver (i.e. kernel land) taking input from a user program (i.e. user land) and is likely to not work, be a hell to debug, and prone for security issues… If that’s even possible to do!

I’m using an Uno, but I’ll look at what you posted because last time we considered the DFU update but we’re kinda scared of bricking stuff

Finally you’re having concerns, in a comment, about bricking your arduino. Please believe me and the author of the article that you cannot brick your arduino doing a firmware flash:

**NOTE: There is no chance of ‘brickling’ the device using this method. The Arduino bootloader firmware can always be updated using the DFU protocol!**

In case something gets really badly wrong, and you get a bootloader corruption in the little AVR (atmega8u2), you still can reflash it using another arduino or an atmel
programmer using the 6 pins next to the USB plug.

In the end, know that the **only** way to brick an AVR, thus an Arduino as well, is to play with the fuses and reset the programming fuses. And even then, you may use HVSP to save your device, though it’s often less trouble to just take a new chip.

---

**Answer** by user3486

I created a project that fully replaces the Arduino Bootloader. You are able to upload sketches and use it as HID. See Project HID on Github: [https://github.com/NicoHood/HID](https://github.com/NicoHood/HID)

---

**Q: Should all unused analog input ports be tied to ground for accurate A2D measurements?**

**Tags:** arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q)

I am using the A0 port on my Arduino Uno to read the voltage output from an LM35 temperature sensor. I observed that the reading fluctuates a lot if I leave the remaining analog input ports A1 to A5 floating. Would floating analog inputs affect the accuracy of readings on input ports connected to some signals.

**Tags:** arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q)

**User:** kavka

**Answer** by matt-clark

Any floating pin will act as an antenna, and may interfere with values around it, this is one of the major disadvantages of using analog.

To get the most accurate readings with minimal amount of interference, you can pull all unused inputs to ground.

The accuracy will not change, as the accuracy is determined by the sensor itself. However it will reduce the amount of noise / interference, which will give you more accurate readings.

These pull to grounds can be accomplished by configuring the unused analog pins as digital outputs, and setting them low.

```c
pinMode(A1, OUTPUT);
digitalWrite(A1, LOW);
```

**Tags:** arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q)

---

**Q: Webserver on Arduino**
How can I run a webserver in an Arduino Uno? I want to use my Arduino to monitor things in my home such as temperature. I want to be able to access this data through the internet.

You’ll need something like the Ethernet Shield to connect your Arduino Uno to a LAN. As for webserver functionality, the Webserver Example sketch does the bulk of what you need.

```c
/*
   Web Server

   A simple web server that shows the value of the analog input pins.
   using an Arduino Wiznet Ethernet shield.

   Circuit:
   * Ethernet shield attached to pins 10, 11, 12, 13
   * Analog inputs attached to pins A0 through A5 (optional)

   created 18 Dec 2009
   by David A. Mellis
   modified 9 Apr 2012
   by Tom Igoe
*/

#include <SPI.h>
#include <Ethernet.h>

#include <ESP8266WiFi.h>

const char* ssid = "yourSSID";
const char* password = "yourPassword";

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192,168,1,177);

//
//
```

For the “access this data through the internet” part, assuming you want access beyond your LAN; you’ll need to perform Port Forwarding on your local router (i.e. allow the “Internet” to see your Arduino).

If you need wifi, then personally I prefer the CC3300 to the Arduino Wifi shield.

The Arduino Yún has been made with this use case in mind. The Bridge library
allows your sketch to publish data such as sensor readings: this data is available through a REST api.

Say you want to publish the temperature of your living room: you’ll write something like
Bridge.put("living_room", String(19)); //celsius

You can then access the data via web with a URL like
http://arduino.local/data/get/living_room

You can get all sensor readings without specifying the key name, with
http://arduino.local/data/get

Q: Arduino serial port reset in Serial monitor & Python

So am I using an Arduino with a shield to gain some voltage values for an ECG. I am then plotting that data in Python. However whenever I “observe” the serial port, be that through the Arduino serial monitor or in attempting to plot the values of time (using millis()) against ECG (voltage) the values reset, so millis() resets the zero, which is fine, except that some of the older values still show up. Now this is especially a problem in Python because it means the plot is malfunctioning at the start of the script, because it is plotting some of the older values and then after a couple of seconds it resets and the plot looks normal again.

To demonstrate this I have recorded it on video and put it on YouTube, here it is:
https://www.youtube.com/watch?v=dNpUakcRPec

Now initially I thought the problem was with Python or the Pyserial module so I labelled the video as such, but since it occurs in the serial monitor I realised the solution would be in the Arduino sketch. I have tried flushing, didn’t work, I have tried even to not start the data till I send a start byte, but this didn’t work on Python.

Ideally I could start the Python plotting script and it would skip, or ignore those initial values from the serial port before beginning again.

Here is the Arduino sketch I am using:

```c
#include <eHealth.h>

unsigned long time;
// The setup routine runs once when you press reset:
void setup() {
  Serial.begin(9600);
}

// The loop routine runs over and over again forever:
void loop() {
  float ECG = eHealth.getECG();
}
```
time = millis();
Serial.print(time);
Serial.print(" ");
Serial.print(ECG, 3);
Serial.println(" ");

delay(50);    // wait for 50 millisecond
}

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q)

**User:** user3284376

**Answer** by **thedoctor**

The Arduino is designed to reset when a Serial monitor or script connects to it. The best way to mitigate this would be to send the Arduino some kind of “start sending” signal when you connect to the serial port. It would then send the data when your script was ready.

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q)
Q: Can I program a 3.3v chip on a 5v board?

Tags: arduino-uno (Prev Q) (Next Q), uploading (Prev Q) (Next Q), atmega328 (Prev Q) (Next Q), safety (Prev Q) (Next Q), oscillator-clock (Next Q)

I bought a Gertboard a while ago, and one of the features it has on-board is an ATMega328 (DIP). However, since the Gertboard is designed for use with the Raspberry Pi, it only operates at 3.3v, so the clock speed is limited to ~12MHz.

Programming it via the Raspberry Pi is quite cumbersome though. Instead, I’d like to transfer the chip to an Arduino Uno board so I can upload sketches from my desktop PC. I would then transfer it back to the Gertboard for normal operation.

The problem is that my Uno operates at the more typical 5v / 16MHz.

Is it safe to attempt this? Do I need to make any configuration changes in the IDE to tell it to account for a different clock speed?

Tags: arduino-uno (Prev Q) (Next Q), uploading (Prev Q) (Next Q), atmega328 (Prev Q) (Next Q), safety (Prev Q) (Next Q), oscillator-clock (Next Q)

User: peter-r.-bloomfield

Answer by ignacio-vazquez-abrams

The clock speed selected will affect both delay(*) and millis()/micros() as well as the functions in <util/delay.h>, therefore you must use the target system’s correct clock speed if you are using any of these functions. This can be easily done by editing boards.txt and copying an existing entry for the Uno and changing the value of the f_cpu parameter to match the target system (in this case, 12000000).

The difference in voltage does not matter, since the chip can run at either voltage and no connection is being made between the 5V system and the 3.3V system.

Tags: arduino-uno (Prev Q) (Next Q), uploading (Prev Q) (Next Q), atmega328 (Prev Q) (Next Q), safety (Prev Q) (Next Q), oscillator-clock (Next Q)

Q: What are some steps to attach an Arduino to an electric heater?

Tags: arduino-uno (Prev Q) (Next Q)

I have the below portable electric heater, but I would love to control it’s start AND stop time using an Arduino. As-is, it does not come with a stop feature. SO, I have a burning question. How can I get an Arduino to start and stop this heater at specific times of the day?

As you can see, it does have a start time, but it does not have a stop time. The power button is a simple push button. I suspect I would need to ‘interject’ into that and the board by using a switch? Apologies if the terminology is off, but I’m very green. Where would a noob begin?
-EDIT- The unit is always off when it’s plugged back in. That likely means I cannot put a timer on the power cord.
Would or should begin? Don’t fiddle with the internals of the heater.

Set its switch to the always on position and control switching on and off with an external relay. Make a box that has plenty of room (you can reduce size at a later moment), a power cord, a mains adapter for your Arduino and a mains outlet in which you can plug your heater.

Use:

- an **RTC (real time clock) breakout** to keep accurate time
- optionally a **7 segment display** to display date/time. Just make sure it has a driver chip on it so you don’t need your Arduino to multiplex the individual LEDs
- couple push buttons to configure the device
- a **relay** rated for the maximum power of the heater to turn it on/off, that can be driven from 5V. Possibly needs a transistor and flyback diode to be controlled from an Arduino GPIO line.

And be very careful when experimenting with mains power, it can be lethal.

---

**Q: Most compact method of powering Arduino from wall socket**

There are a lot of methods to power an Arduino from a large range of voltages:

- USB cable from PC or from a phone charger or an USB hub
- step down converters
- step up converters
- switching power supply
- batteries (connected to the power jack or USB or to Vin)

What I can’t seem to find is a small 220V-5V converter. I’d like to put an Arduino in the wall, under a light switch, so size matters.

The last resort option is to open a 5V phone charger and replace the metal prongs that go in a socket with two wires, and replace the USB socket with another pair of wires that go the Arduino board, but still, an Arduino Pro Mini or an Arduino Nano is smaller than the phone charger.
The last resort option is to open a 5v phone charger and replace the metal prongs that go in a socket with two wires, and replace the usb socket with another pair of wires that go the arduino board, but still, an Arduino Pro Mini or an Arduino Nano is smaller than the phone charger.

You will probably have to do something like that.

Because of the nature of conversion between AC and DC, there has to be some large parts.* I’m not going to go into the specifics here. There is usually a diode bridge (because AC waveforms reverse the current ~60 times a second) and a transformer (to lower the voltage for the diode bridge). They almost always include a regulator and a capacitor to make sure voltage is exact and it outputs a “clean” power supply.

*Switching power supplies[1], although considerably more expensive, are a lot smaller and more power efficient.

This adapter[2] seems to be the most compact adapter I can find for 220V. You’ll never get something near the size you are desiring, even with a switching power supply. I don’t know the size of the Amazon adapter I gave a link for, but in the US (120V), iPhone chargers are very small and they should fit in a standard depth outlet box (with a little
room left over for relays and such). If you really need to get everything smaller, I recommend using a ATtiny chip.

**How I recommend connecting this:**

I would avoid ripping this apart at all costs from a safety perspective. It would be really easy to send 220V to your Arduino if you’re not careful. If you absolutely have to, you might want to coat it with some non-conductive epoxy or other coding. Note: this may act as a thermal insulator, therefore reducing the lifespan of the part. You should just coil some wire around the prongs, and then melt a little solder onto each of the terminals. Then, cover that with electrical tape and pigtail (with a wire nut) into a source power line. Note: you probably are doing wiring that could be illegal in your area. Check your local building codes before attempting this.

After that, attach a USB cord to the socket and to the Arduino or the ATtiny/ATmega328 chip 5V/GND input (of coarse, after cutting and stripping one end of the USB cord). I would also, after finalizing all the code and circuitry, seal that with some epoxy or somehow add it to an enclosure to make sure you don’t fry your Arduino if a loose wire touches the Arduino circuitry.

**If you don’t feel safe doing any of the above, don’t do it. You can live without a Arduino light switch.**

That being said, happy hacking! :P

---

**Answer** by kom

I can suggest you to try this: [AC DC Step Down Converter](#)

It is as small as the Apple’s tiny cube, and it can be placed inside of a wall easily. (I’m personally using this with an arduino micro)

---

**Answer** by brainwash

An obvious answer has been omitted which is: transformerless power supply. If your average power requirements are low (<10mA) then this is the most compact way to achieve the desired voltage.


Another solution: jeelabs.org /2011/11/27/ultra-low-power-supply/

Some limitations:

- **safety - NONE** - your circuit might now be floating at line voltage
- heat dissipation - calculate here: www. daycounter.com/Circuits/Transformerless-Power-Supplies/Transformerless-Power-Supplies.phtml
- low current - no servos or relays
- voltage fluctuations on the line might affect the circuit
Q: Timer2 does not work as it should

Tags: arduino-uno (Prev Q) (Next Q), c++ (Next Q), timers (Prev Q) (Next Q)

I am currently playing with Arduino timers (on UNO currently) and I am building a library that has a function that must be called every millisecond. There is plenty of code samples on Internet so I went with something that seemed perfect for my needs. I have then updated it to make it a reusable C++ class.

Here is my code (simplified to just isolate the issue):

```cpp
#define BIT(x) (0x01 << (x))

static volatile uint32_t milliseconds = 0;

class Timer
{
    public:
        Timer()
        {
            // Disable interrupts
            uint8_t savedStatus = SREG;
            cli();
            // Use Timer2 with 1 ms interrupts
            // OC2A & OC2B disconnected, mode 2 (CTC, Clear Timer on Compare match)
            TCCR2A = BIT(WGM21);
            // Don't force output compare (FOCA & FOCB), mode 2, Clock Select clk/128 (CS = 5)
            TCCR2B = BIT(CS22) | BIT(CS20);
            // Set timer counter compare match (when value reached, 1ms has elapsed)
            OCR2A = 125 - 1;
            // Reset Timer2 counter
            TCNT2 = 0;
            // Set Timer2 interrupt mode (Set interrupt on OCR2A compare match)
            TIMSK2 = BIT(OCIE2A);

            // Restore interrupt enabling state
            SREG = savedStatus;
        }

    static void callEveryMillisecond(uint32_t ms)
    {
        // Do something here
    }

    // Attach interrupt routine to the Timer Compare Interrupt
    ISR(TIMER2_COMPA_vect)
    {
        milliseconds++;
        callEveryMillisecond(milliseconds);
    }
};

static void callEveryMillisecond(uint32_t ms)
{
    // Do something here
}
```

After checking ATmega328 datasheet (section 17) several times, I came to the conclusion that what I do in Timer class constructor above is correct, and for Arduino UNO, the values used must lead to one call of the ISR every ms:

- CPU frequency = 16MHz
- Prescaler = 128
- -> timer frequency = 16MHz/128 = 125KHz
- max counter value = 125

I decided to use the **CTC** (Clear Timer on Compare match) mode, which means that when the timer TCNT2 (which is incremented at 125KHz frequency), reaches OCR2A value, then it will trigger the ISR, and then will be reset on next tick; the additional tick to reset the counter explains why we use 125 - 1 for OCR2A and not just 125.

Now comes the weird stuff. In order to check that my code worked, I decided to blink the LED on pin 13 every 10 seconds, hence I have added the following code to my program:

```cpp
static void callEveryMillisecond(uint32_t ms) {
    if (ms % 10000 == 0)
        digitalWrite(13, digitalRead(13) ^ 1);
}

static Timer timer;
void setup() {
    pinMode(13, OUTPUT);
}
void loop() {
}
```

Then I measured the time during which pin 13 LED is lit on (or off, both measures are the same). Measures are not very accurate (I just used my wristwatch) but good enough for my check.

I was surprised to see the measured time to be about **20 seconds instead of 10s**.

Initially I thought about a mismatch in CPU frequency, but after performing a lot of small changes, here is what I found which worked perfectly:

```cpp
//static Timer timer;
void setup()
{
    static Timer timer;
    pinMode(13, OUTPUT);
}
```

Just moving the `Timer` instance into `setup()` did the trick: my measures rightly indicated **10 seconds** between 2 states of the pin 13 LED.

**Now my questions are:**

1. Why the second code sample works perfectly, whereas the first does not?
2. Is there, in standard Arduino included code, some code that would overwrite some of the settings I perform for Timer2 in `Timer::Timer()`?
3. Is there a way to make the first code sample work? Note that I don’t like the second sample because it does not seem natural coding, also, I will need access to `Timer` instance in several locations of my future program, hence it can’t be possible if it is declared in the scope of `setup()`.

**Tags:** [arduino-uno](Prev Q) (Next Q), [c++](Next Q), [timers](Prev Q) (Next Q)
I have located the source of the problem.

In file wiring.c (arduino-1.5.6-r2\hardware\arduino\avr\cores\arduino) which gets called before setup(), there is the following part that tries to changes the timer 2 prescaler to 64 by setting the CS22 bit.

```c
// set timer 2 prescale factor to 64
#if defined(TCCR2) && defined(CS22)
  sbi(TCCR2, CS22);
#elif defined(TCCR2B) && defined(CS22)
  sbi(TCCR2B, CS22);
#else
  #warning Timer 2 not finished (may not be present on this CPU)
#endif
```

You already set CS20 and CS22 so this doesn’t affect you. What does affect you is the following part which sets bit WGM20

```c
// configure timer 2 for phase correct pwm (8-bit)
#if defined(TCCR2) && defined(WGM20)
  sbi(TCCR2, WGM20);
#elif defined(TCCR2A) && defined(WGM20)
  sbi(TCCR2A, WGM20);
#else
  #warning Timer 2 not finished (may not be present on this CPU)
#endif
```

and you have already set WGM21 so WGM21 and WGM20 set changes the timer mode to fast PWM, top=0xFF which gives almost double delay that the mode you were trying to use (CTC top=124).

**How to workaround this issue?**

Timer2 is used when its associated PWM pins are used with an analogWrite() call, so the Arduino setup code sets all prescalers to well-defined values just in case the timers might get used later.

I would not try to fight that code; instead, I would just make sure that my setup code runs later, e.g. by moving it to a separate setup() method in your Timer class and calling that from the global setup() routine.

---

**Tags:** arduino-uno (Prev Q) (Next Q), c++ (Next Q), timers (Prev Q) (Next Q)

---

**Q: How to use RS-232 sheild as software serial port?**

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), software (Next Q)

I would like to be able to use a USB port at the same time as an RS-232 port with my Arduino Uno board. I unwittingly bought a shield that (I think?) overrides the USB port when it is mounted to the Arduino board:


Would anyone be able to give me any advice on how to connect the hardware so that I can
have two serial ports accessible from my Arduino Uno? The format I was thinking was (Arduino => RS-232 Shield):

GND => GND 5V => 5V Digital Pin 2 => 232RXD Digital Pin 3 => 232TXD

The accompanying code for initializing the serial port which I’ve written:

```cpp
const int rxpin = 2; // pin used to receive
const int txpin = 3; // pin used to send
SoftwareSerial Serial_rs232(rxpin, txpin); // new serial port
```

Any ideas as to why this may or may not be a good method of using the hardware I have available for the purpose I had in mind?

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), software (Next Q)

**User:** user_name

---

**Answer** by anonymous-penguin

GND => GND 5V => 5V Digital Pin 2 => 232RXD Digital Pin 3 => 232TXD

A better choice of board would be the Leonardo, since you could connect such a shield and use the USB port right out of the box. However, since you have the Uno, there is a solution.

*From Arduino:*  
The SoftwareSerial library has been developed to allow serial communication on other digital pins of the Arduino, using software to replicate the functionality (hence the name “SoftwareSerial”). It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

It then goes on to recocmmend the library AltSoftSerial:

Here is some example code from the link above:

```cpp
#include <AltSoftSerial.h>

AltSoftSerial altSerial;

void setup() {
  Serial.begin(9600);
  Serial.println("AltSoftSerial Test Begin");
  altSerial.begin(9600);
  altSerial.println("Hello World");
}

void loop() {
  char c;

  if (Serial.available()) {
    c = Serial.read();
    altSerial.print(c);
  }
  if (altSerial.available()) {
    c = altSerial.read();
    Serial.print(c);
  }
}
```
It acts like a standard serial port, only you declare the connection as a variable first. This
code is written for a board with an additional serial port, like you have.

**All is good, except for one problem:**

The problem is the shield is wired to connect to pins 0 and 1: the ones used by the USB
adapter. The AltSoftSerial library uses pins 8 for TX and 9 for RX. (Note: You cannot use
PWM on pin 10 on the Uno… it should still work as a digital pin.) In that case you would
**not attach the shield by plugging it into the Arduino**, and you would wire it like this:

```
GND => GND | 5V => 5V | Digital Pin 8 => 232RXD pin 0 on the shield headers |
Digital Pin 9 => 232TXD 1 on the shield headers
```

So then, you would be manually wiring the shield to the Arduino with jumpers. As long as
it doesn’t connect to any other Arduino pins I’m not aware of (it shouldn’t besides maybe
13 for a LED or something like that), it should function like it would stacked and using
pins 0 and 1.

Note: (Obviously) You cannot connect anything to the headers on the shield (You cannot
connect to pin 5 and get a signal).

---

**Q: How to build (concat) a string with big numbers**

Tags: arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), software (Next Q)

new to arduino I’m struggling with what sounds like fairly n00b problem… I’ve wired up
a adafruit GPS-board to my Arduino and it is working as it spits out GPS data to the serial
port with `Serial.print(GPS.latitude, DEC)`

I Now want to concat a string which I can process (read: I want to sent it via an ethernet
client.) This is what I got already:

```
Skip code block
```

```
String vntnt = "$VNNT,";
if (GPS.fix) {
    vntnt += "GPS,;"
    //this works:
    vntnt.concat(GPS.fix);
    //but this not:
    vntnt.concat(GPS.latitude);
} else{
    vntnt += "INFO,Acquiring Sats";
}
Serial.println(vntnt);
```
The error message is: Call of overloaded 'concat(float&)' is ambiguous. When I Serial.print(GPS.latitude, DEC) it results in: 4418.5937996050

So it is probably too big or something...

**How can I concat the vars and create the long string?**

**Tags:** arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q), software (Prev Q) (Next Q)

**User:** sturb

**Answer** by drodi

The concat function does not implement a float version, but some of char, int, unsigneds...

```cpp
unsigned char String::concat(long unsigned int)
unsigned char String::concat(unsigned int)
unsigned char String::concat(int)
...```

so the compiler does not know how to cast (truncating probably the float) to integer, as there are several options.

You have to convert first your float to string, use dtostrf() or sprintf() then concat to your string.

```cpp
char outstr[25];
sprintf(outstr, "%f", GPS.latitude);
```

or

```cpp
dtostrf(GPS.latitude, 6, 2, outstr); //check docs for 6, 2 values, and set them for your needs
```

And then:

```cpp
vnnt.concat(outstr)
```

Also note that sprintf is very handy for compact creation of a (char) string:

```cpp
char msg[80];
sprintf(msg, "$VNNT,GPS, %s %f", GPS.fix, GPS.latitude);
```

**Q: Ultrasonic Sensor scanning too often resulting in a whirring noise**

**Tags:** arduino-uno (Prev Q) (Next Q), sensors (Next Q)

I am using an HC-SRO4 ultrasonic sensor for a wall-avoidance, maze-solving robot. However, I have recently run into an issue where I am getting a whirring noise from the sensor because it is taking measurements way too often.

This is the code I have right now:

```cpp
const int serialPeriod = 250; // only print to the serial console every 1/4 second
```
unsigned long timeSerialDelay = 0;
const int loopPeriod = 20; // a period of 20ms = a frequency of 50Hz
unsigned long timeLoopDelay = 0;

// specify the trig & echo pins used for the ultrasonic sensors
const int ultrasonic2TrigPin = 8;
const int ultrasonic2EchoPin = 9;

int ultrasonic2Distance;
int ultrasonic2Duration;

void setup()
{
  Serial.begin(9600);
  // ultrasonic sensor pin configurations
  pinMode(ultrasonic2TrigPin, OUTPUT);
  pinMode(ultrasonic2EchoPin, INPUT);
}

void loop()
{
  debugOutput(); // prints debugging messages to the serial console
  if(millis() - timeLoopDelay >= loopPeriod)
  {
    readUltrasonicSensors(); // read and store the measured distances
    timeLoopDelay = millis();
  }
}

void readUltrasonicSensors()
{
  // ultrasonic 2
  digitalWrite(ultrasonic2TrigPin, HIGH);
  delayMicroseconds(10); // must keep the trig pin high for at least 10us
  digitalWrite(ultrasonic2TrigPin, LOW);
  ultrasonic2Duration = pulseIn(ultrasonic2EchoPin, HIGH);
  ultrasonic2Distance = (ultrasonic2Duration/2)/29;
}

void debugOutput()
{
  Serial.print("timeSerialDelay: ");
  Serial.print(timeSerialDelay);
  Serial.println();
  if(((millis() - timeSerialDelay) > serialPeriod)
  {
    Serial.print("ultrasonic2Distance: ");
    Serial.print(ultrasonic2Distance);
    Serial.print("cm: ");
    Serial.println();
    timeSerialDelay = millis();
  }
}

and my wiring:
Does anyone know why the scanning is happening constantly? It should only be happening every 1/4 of a second, not constantly. The weird thing is that this wasn’t happening earlier before I started using the cable.

I believe it has something to do with the 8.5” extension cable I have (needed because I can’t mount three sensors, one facing to each side and one in front, on a single breadboard). This is because when I mounted the sensor directly onto the breadboard, the delay was correct and the whirring was not present. However, as I just mentioned, I need the extension cable to be able to have the three sensors.

Does anyone have a solution for this?

Thanks!

Tags: arduino-uno (Prev Q) (Next Q), sensors (Next Q)

User: rpi-awesomeness

Answer by thedoctor

If you want to sample at 4Hz (4 times per second), you need to change your headers accordingly:

```c
const int serialPeriod = 250;    // only print to the serial console every 1/4 second
unsigned long timeSerialDelay = 0;

const int loopPeriod = 250;      // a period of 250ms = a frequency of 4Hz
unsigned long timeLoopDelay = 0;
```

Tags: arduino-uno (Prev Q) (Next Q), sensors (Next Q)

Q: Arduino : What happens if I power the arduino with both the USB and
I am a novice user of Arduino. I am not sure whether I can connect both USB and external supply through power adaptor to Arduino simultaneously. Would this burn the controller?

Basically, I want to automatically shutdown the computer after saving all open documents before my UPS battery goes out whenever I am not there to manually shut it down. I am trying to notify the computer about the power failures of main supply - AC 220V 50 Hz (notification/signal sent when power goes out) via USB using Arduino. I am thinking that this can be done by using the external voltage power supply pin. Whenever power goes out in the mains, the external voltage power supply pin will drop to 0 V from 7 V. The value of the voltage can be read using software (I am unclear about this: Can it be done? If yes, how?). The arduino will still be powered up as it is connected to USB of computer and thereby communicate to the computer about the power failure.

---

**Power**

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

Put the external power through a voltage divider or transistor-based inverter and into an analog (divider) or digital (inverter) pin and check the appropriate value from the pin. Note that this does not require powering the Uno from external power, only connecting the grounds.

---

**Q: How to call C functions from Arduino sketch?**

I would like to know if there is a way to call functions that are contained within C files using an Arduino sketch?

My C file declares and defines a function. To save putting the messy function definition into my Arduino sketch, I’d like to call the function straight from the sketch.

Is there a standard way to do this using Arduino and C? Here is the sketch:

```c
#include "crc16.h"
```
void setup(){
}
void loop(){
    CalculateCRC16("<09M", 4);
}

and this is the trimmed down C file:

```c
#include <stdio.h>
#include <stdint.h>

uint16_t crctable[256] =
{ 0x0000, 0x1189, ......

uint16_t crctable[2] = { 0x0000, 0x1189,

void CalculateCRC16(const void *c_ptr, size_t len) {
    uint16_t crc = 0xFFFF; // Seed for CRC calculation
    const uint8_t *c = c_ptr;

    while (len--)
        crc = (crc << 8) ^ crctable[((crc >> 8) ^ *c++)];

    return crc;
}
```

Tags: arduino-uno (Prev Q) (Next Q), c (Next Q), software (Prev Q) (Next Q)

User: user_name

**Answer** by drodri

You can extern “C” #include like the following:

```c
extern "C"{
    #include "crc16.h"
};
```

void setup(){
}
void loop(){
    CalculateCRC16("<09M", 4);
}

And the crc16.h file could be (some minor fixes, the #pragma once, a cast):

```c
#pragma once

#include <stdio.h>
#include <stdint.h>

uint16_t crctable[2] = { 0x0000, 0x1189};

uint16_t CalculateCRC16(const void *c_ptr, size_t len) {
    const uint8_t *c = c_ptr;

    while (len--)
        crc = (crc << 8) ^ crctable[((crc >> 8) ^ *c++)];

    return crc;
}
```
Your CRC function can easily be converted to C++ so that it can go into a *.cpp file. All you need to do is use an explicit cast when you initialise your c pointer. Here’s the ‘proper’ C++ way to do it:

```cpp
const uint8_t *c = static_cast<const uint8_t*>(c_ptr);
```

However, an old C-style cast would also work:

```cpp
const uint8_t *c = (const uint8_t*)c_ptr;
```

The problem is basically that C can be a little more permissive about letting you convert pointers implicitly between types. To do it in C++, you need to tell the compiler explicitly that the conversion is intentional.

**Tags:** arduino-uno (Prev Q) (Next Q), c (Next Q), software (Prev Q) (Next Q)

---

**Q: “Broken Pipe” when uploading to Arduino UNO**

**Tags:** arduino-uno (Prev Q) (Next Q), uploading (Prev Q) (Next Q)

I just got my Arduino UNO and I’m trying to upload the blink example but the upload fails with

```none
ioctl("TIOCMSET"): Broken pipe
ioctl("TIOCMSET"): Broken pipe
avrdude: stk500_recv(): programmer is not responding
ioctl("TIOCMSET"): Broken pipe
```

I have tried both Arduino IDE 1.0.1 (which I installed via my package manager) as well as version 1.0.5 which I downloaded from the arduino.cc website. I’m running Ubuntu Linux 12.10 if that makes a difference.

I’d appreciate any and all help in getting my arduino up and running!

**Tags:** arduino-uno (Prev Q) (Next Q), uploading (Prev Q) (Next Q)

**User:** jonas

---

**Answer** by the-guy-with-the-hat

As with any communication error, try these:

- Disconnect and reconnect the USB cable.
- Use a different USB cable.
- Press the reset button on the board.
• Restart the Arduino IDE.
• Make sure you select the right board in Tools ► Board ►, e.g. If you are using the Duemilanove 328, select that instead of Duemilanove 128. The board should say what version it is on the microchip.
• Make sure you selected the right port in Tools ► Serial Port ►. One way to figure out which port it is on is by following these steps:
  1. Disconnect the USB cable.
  2. Go to Tools ► Serial Port ► and see which ports are listed (e.g. COM4 COM5 COM14).
  3. Reconnect the USB cable.
  4. Go back to Tools ► Serial Port ►, and see which port appeared that wasn’t there before.
• In extreme cases, you may need to burn the bootloader.

Answer adapted from here.

---

**Q: What’s the 2nd ICSP header for in Arduino Uno R3?**

Tags: arduino-uno (Prev Q) (Next Q), uploading (Prev Q) (Next Q)

Looking at my Arduino Uno R3 board and its reference design schematic I noticed that there’s a second ICSP 6-pin header. In the schematic, the connector I’m referring to is named ICSP1.

**What’s that for?** What useful things can an average Arduino user like me do with it?

The relevant part of the schematic is below (marked in red).
It’s for the other MCU on the board.

The main MCU on the Uno is labelled “ZIC1”. This is the one that is programmed when you press “Upload” in the IDE. But unlike other Arduinos that use a FTDI chip to connect to the serial port on the main MCU, the Uno uses another MCU instead, labelled “U3” on the left. This MCU runs a small bit of code that acts as a serial port via USB and passes through the bytes to its USART1. This then connects to the USART on the main MCU, where the bootloader on that chip accepts the bytes coming through and writes to flash and EEPROM on the main MCU.

You can use this additional MCU as you would any other, but the only I/O it has available are the USB connection, the SPI connection via its ICSP header, the UART1 connection to both the main MCU and pins 0 and 1, and two LEDs on PD4 and PD5.
Q: Can the 2nd MCU on the UNO R3 be used for keyboard emulation?

Tags: arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q)

Following on from What's the 2nd ICSP header for in Arduino Uno R3? I was wondering how this could be hacked to our advantage.

For example, could that firmware be rewritten to have the UNO recognised as a mouse/keyboard input? Could this be done after the boot loader process and leave the uploading process in tact?

Given the limited IO, it seems it’s functions might be limited. Plus I have no idea how much code is space is available.

Tags: arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q)

User: madivad

Answer by thedoctor

Yes, it is possible. using LUFA, you can burn new firmware to the Serial converter. The only problem is that you need to re-burn the USB-Serial converter to upload another program.

Info: http://www.fourwalledcubicle.com/LUFA.php

Specific page: http://mitchtech.net/arduino-usb-hid-keyboard/

Answer by ignacio-vazquez-abrams

The second MCU is similar to that used in the Leonardo, but is smaller and less capable (ATmega16U2 vs. ATmega32U4). It is certainly possible to reprogram it to do what you like, but its limits means that you can have fewer USB “devices” at the same time.

I have not looked at the existing firmware, but it should certainly be possible to add your own custom code into it such that both the normal serial connection to the main MCU as well as your additional code can coexist.

Tags: arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q)

Q: Can the Arduino Uno become an AVR programmer and use the ICSP header to program the target board?

Tags: arduino-uno (Prev Q) (Next Q)

The Arduino Uno has an ICSP header that I imagine is for programming the board using a separate ICSP programmer. That is, its ICSP header turn the board into a target.

Can one use that same ICSP to program other boards, turning the Arduino Uno effectively into a ICSP AVR programmer? How?
I’m assuming that the target boards would also be Arduinos and wanted the Uno to program them using the Arduino IDE. Is that possible?

Reading the [AVR In-System Programming Application Note](https://www.microchip.com/wwwsection/techpubs/avr/avr-in-system-programming-application-note) it looks to me that the only issue would be to make the programmer (the Arduino Uno) control the target board reset line. I’ve seen something like that in ArduinoISP sketch, but in that sketch, the target reset is controlled by port D10. But one could program the reset line to be an output, right? So could we change the ArduinoISP sketch to use its own reset line as output to control the target board reset line? If not, **why not?**

**Tags**: arduino-uno (Prev Q) (Next Q)

**User**: ricardo

---

**Answer**, by **microtherion**

First of all, as configured, the RESET pin can **not** be used as an output pin. In order to use it as an I/O pin, you would have to program the RSTDISBL fuse. But once you do that, as jippie said, it becomes near-impossible to reprogram your Arduino (if you have the socketed through hole variety of the MCU, you could pull it out and put it into a high voltage programmer; in all other cases, it’s effectively impossible).

So in theory, you can use the ICSP header the way you suggest. In practice, it’s under almost any conceivable circumstance a terrible idea.

---

**Answer**, by **kiwiron**

I recommend going out and buying an AVR Dragon - costs around $US50. That gives you ISCP programming plus debugwire debugging (using the Atmel Studio IDE).

---

**Answer**, by **jippie**

Yes you can program the Arduino’s reset input pin as an output and use that to program another AVR, but you can no longer program your Arduino as that relies on the controller being reset through its own reset pin.

Coming to think about it, I am not sure how the other devices (like the USB controller) responds to being reset by Arduino’s controller itself. It resets the USB controller which may or may not reset the connection to the computer, possibly loosing data. With the USB controller being reset, GPIO-pins D0 and D1 (the serial interface between PC and Arduino’s AVR) may corrupt the datastream.

---

**Tags**: arduino-uno (Prev Q) (Next Q)

---

**Q**: Parse JSON with arduino to turn on LED

**Tags**: arduino-uno (Prev Q) (Next Q), shields (Next Q)

Hi i’m trying to parse JSON resposne from webserver in my arduino in order to turn on and off a LED light. I’m using the wifi client repeating example to make a get request to
my server:

Here is what I get printed back from the serial port

```c
// Code block
Connecting...
HTTP/1.1 200 OK
Date: Sun, 06 Apr 2014 01:14:37 GMT
Server: Apache
X-Powered-By: PHP/5.5.10
Cache-Control: no-cache
X-Frame-Options: SAMEORIGIN
Set-Cookie: expires=Sun, 06-Apr-2014 03:14:37 GMT; Max-Age=7200; path=/; httponly
Connection: close
Transfer-Encoding: chunked
Content-Type: application/json

19
{"lightstatus":"on"}
0
```

How do I parse the JSON portion of this response only so that I can use it to control my LED?

Thanks

**Tags:** arduino-uno (Prev Q) (Next Q), shields (Next Q)

**User:** user1424508

---

**Answer** by sachleen

I’m aware of a few JSON parsing libraries for Arduino.

- aJSON
- Arduino JSON Parser
- JSON Arduino

I’ve never used any of these but I did do some simple JSON parsing in a project I’m working on so I’ll show you how I did that.

**Note:** I’m reading serial data using the software serial library. You’ll need to change this code to work for you. This will only work on very simple JSON strings. It’s very limited but if that’s all you’re parsing then it’ll work.

**Example of JSON response from server:**

```json
{"id":"TEST1","lat":"38.56050207","lng":"-121.42158374","total":"3","available":"2"}
```

First, only read data between curly braces.

```c
// Code block
String response = "";
bool begin = false;
while (SIM900.available() || !begin) {
  char in = SIM900.read();
  if (in == '{') {
    begin = true;
  }
```
if (begin) response += (in);
if (in == '}') {
  break;
}
delay(1);

This code reads data one byte at a time and once it gets an open brace, it starts saving it into response. When it gets a closing brace, it ends it. So here’s a clear limitation, you can only have one set of opening/closing braces in your string.

Once I have the string, I use `indexOf` and `substring` to extract relevant information:

```cpp
start = response.indexOf("id":"") + 5;
end = start + 5;
odeId = response.substring(start, end);
```

This code sets `start` to the beginning of `id":"` + 5 characters in the string. It’s +5 because that’s how long `id":"` is. So `start` points to TEST1 in the JSON string. In my system, the ID is always going to be 5 characters long so `end` is `start + 5`. I then use `substring` to extract that.

Again, before anyone starts down voting me for this horrible solution: if you know exactly what you’re working with, and understand the limitations of this code, then this is not a bad solution. It’s a solution that gets the job done.

---

**Answer** by lasote □

I have programmed a class that will receive char by char the JSON document. It only will store in memory a few bytes for known JSON structure using a state machine and the results you need. So you can query the class for the results you want and will process the JSON.

Its ideal for your purpose. I have used for connecting to a weather service that returns a json:

```cpp
static const char* queries[] = { "list.0.deg", "list.0.weather.0.main"};
StreamJsonReader jsonreader(queries, 2); // 2 queries
while(char c = read()){  
  jsonreader.process_char(c);
}
cout << jsonreader.results[0] << endl;
cout << jsonreader.results[1] << endl;
```

Check this blog post I just wrote: [http://web.biicode.com/blog/big-json-on-arduino](http://web.biicode.com/blog/big-json-on-arduino).

Source code is [here](http://web.biicode.com/blog/big-json-on-arduino) in Biicode.

---

**Answer** by eternityforest □

I wrote a parser for the PIC that has very low ram use because it works directly with char pointers and does not build a tree structure. If you ask to get item N of a list, it gives you an actual pointer into the direct JSON file right at the start of item N of the list. Then you can ask what type of object the pointer represents, etc. It’s buggy and doesn’t support floats and I think has a few other limitations(twas a long time ago) but it’s all on
You’ll need to look in libfortune which is part of the C version.

**Q: Controlling LED from web server**

I have a LED attached to my Arduino wifi shield that I want to control from my PHP web server. In terms of communication, should I

1. Make my Arduino a web server and then communicate with PHP web server via HTTP
2. Make my Arduino a web client and receive commands from my PHP web service. I would parse the JSON response

Also my other question is should I communicate with JSON between my PHP web server or is there a better way to send commands to my LED light?

**Answer** by ignacio-vazquez-abrams

There’s no need for either PHP or JSON. Simply detect when one of two URLs (say, /led/on and /led/off) has been accessed on the Arduino’s web server, change the LED state, and return a 204 response.

**Answer** by akellyirl

The REST protocol is ideal for this and there’s a nice Sketch already written, called RESTduino on Github: [https://github.com/jjg/RESTduino](https://github.com/jjg/RESTduino)

Basically (and loosely) REST treats a URL as a command. Quoting from the RESTduino doc’s:

“To turn on the LED attached to pin #9 (currently case sensitive!):

```
http://192.168.1.177/9/HIGH
```

This will set the pin to the HIGH state and the LED should light. Next try this:

```
http://192.168.1.177/100
```

This will use PWM to illuminate the LED at around 50% brightness (valid PWM values are 0-255).”
Q: Simultaneous read and read/write on two serial connections

Tags: arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), pins (Prev Q) (Next Q), interrupt (Next Q)

At the moment I’m using two softserials to connect to a GPS and an GSM module. It seems not possible to have two open software-serials. So I was looking for an solution

After the initial-setup, the GPS module is only needed to be read from; while the GSM module needs to be bidirectional. So:

<table>
<thead>
<tr>
<th>GPS &gt; listen only</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM &gt; listen and write</td>
</tr>
</tbody>
</table>

Now I came across the [GPS library from Adafruit](https://www.adafruit.com), which uses an interrupt to receive data. Is this instead of a software serial. Or do I again run into the limitation?

An other solution might be to use the D0 and D1 to attach eg the GPS module. But than I won’t be able to see the debug messages in my serial monitor. Is that correct?

Sorry in advance for these n00b questions. But I’m frustrated the things dont work as I want :)

edit

- [This is the adafruit GPS board](https://www.adafruit.com/product/1403).
- And the board with whom it is connected: [Gboard](https://www.adafruit.com/product/1403).
- The Sim900 is connected to D2/D3 and the GPS is connected to A2/A3.
- D0/D1 are connected to a FTDI breakout board, which is plugged into my USB.

Tags: arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), pins (Prev Q) (Next Q), interrupt (Next Q)

User: sturb

---

**Answer** by jippie

[Adafruit_GPS.h](https://www.adafruit.com/product/1403) (line 132-), which is included in [Adafruit_GPS.cpp](https://www.adafruit.com/product/1403) (line 13) refers in turn to the SoftwareSerial library on AVR platform and Arduino IDE version > 100.

```c
#ifndef __AVR__
#if ARDUINO >= 100
  SoftwareSerial *gpsSwSerial;
#else
  NewSoftSerial *gpsSwSerial;
#endif
#endif
```

This means it uses software serial on an Uno. It also means if your problem is indeed with Software Serial, that is the library to check for bugs / updates or commit a patch to.

Tags: arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q), pins (Prev Q) (Next Q), interrupt (Next Q)
Q: Understanding the relationship of Pulse and Stepper RPM

This may be a better question for the Physics StackExchange so tell me if I should move the question.

I have a pretty simple script that I’m using to control the RPM of my stepper motor:

```cpp
void loop(){
    digitalWrite(PIN, HIGH);
    delayMicroseconds(wait);
    digitalWrite(PIN, LOW);
}
```

as you can see it’s just producing a pretty standard pulse, I have my Arduino connected to a driver that manages the motor. The wait variable comes from a second order equation I derived from measuring the RPM with a Tachometer and tweaking the value.

```cpp
// From data RPM = 31729/x + 17.327 thus x = 31729/(RPM-17.527)
wait = M/(RPM-C);
```

It works pretty well, I get readings consistently within only 1 - 2 rotations off. But the slope and intercept seem completely arbitrary to me. Is there a chance it has to do with the clock speed of the Arduino? I’m using an Uno and from playing around with the numbers I can’t seem to find a relationship. From what I can tell, the driver looks hardwired so I don’t think it has much to do with the equation.

Any idea what these values, the slope/intercept mean?

Equation: RPM=31729/wait + 17.327 M:31729 and C:17.327

User: dylan-madisetti

Answer by brettam

The real answer to this question requires understanding exactly what hardware you have. The formula, assuming one step per pulse, is that the delay should be:

\[
\text{wait} = \left( \frac{\text{MicroSecondsPerMinute}}{1} \right) \left( \frac{1}{\text{RPM}} \right) \left( \frac{1}{\text{PulsesPerRevolution}} \right) - \text{overhead}
\]

where overhead is the extra time it takes to run digitalWrite twice and loop. It is small, but if it were zero your code would not work because the amount of time your signal is LOW would be zero as well.

Your numbers suggest Pulses Per Revolution is around 1891. This will be the product of how many steps per revolution the motor is wired for and how many bits of microstepping the driver is doing. A 225 step motor with 3 bit microstepping (8 fractional steps per “true” step) would have PPR of 1800 for example. Your number does not precisely match any common setups, but is exactly in the range expected, with 1600 and 2000 being normal for small steppers.

Because of the “overhead” term, RPM as a function of wait should not be linear. Your constant value should be a function of RPM; in fact, when RPM gets much larger than the
number of Microseconds per minute it should converge to being equal to RPM. In this case your RPM is so low that approximating it to 17 is probably valid and all error too small to notice. It could be causing some of the oddness of your PPR value though.

**Q: Is the Sparkfun RedBoard pre-loaded with a blink sketch?**

As soon as I plugged my brand new RedBoard (Sparkfun’s Uno) into my computer, I saw the onboard LED (for pin 13) blink on and off every 1 second. This was before I even sent the first demo sketch to blink the LED. After I uploaded a sketch to change the blink time, the LED’s blink cycle changed as I expected.

But why was the LED blinking *before* I sent my very first sketch? Is the RedBoard pre-loaded with a sketch?

---

**Q: What’s the deal with Uno’s pin 13 LED?**

(I have a Sparkfun RedBoard, but this question seems to apply to R3 Unos and Uno-compatible boards.) As I was building the first circuit in my SIK guidebook (add a resistor, LED, hook it up and make it blink from code), I noticed a blue LED on the board itself did everything that the circuit’s LED did - blinking according to the programming.

Why is this LED here? What use cases is it for?

What kind of circuits will I have to adjust to account for this LED?

Is pin 13 traditionally a debug or a troubleshooting pin? Where did this convention come from?
The LED on pin 13 is used by the **optiboot** loader (the one used on **UNO**):

- at Arduino **boot time** (the LED blinks a few times)
- when **uploading a sketch** to Arduino

I haven’t checked other bootloaders, they may provide the same behavior as the optiboot.

For optiboot, there are optional defines (at compile time) to modify this behavior:

- **LED_START_FLASHES** defines the number of flashes of pin 13 LED at boot time (can be set to 0)
- **LED_DATA_FLASH** will use pin 13 LED during sketch upload if **defined** at compile-time

These defines are explained in `hardware/arduino/bootloaders/optiboot/optiboot.c` from within your Arduino IDE install directory.

If you want to change these, you will have to recompile the optiboot loader first and then burn it to your Arduino through an **ISP programmer**.

I guess one other reason for the LED on pin 13 was to simplify the demonstration of the “**Hello World**” sketch for Arduino, namely the **Blink** sketch, without the need for any extra component.

---

**Answer** by **sachleen**

What kind of circuits will I have to adjust to account for this LED?

You’ll need to take it into account if you’re using that pin as a digital input.

**NOTE:** Digital pin 13 is harder to use as a digital input than the other digital pins because it has an LED and resistor attached to it that’s soldered to the board on most boards. If you enable its internal 20k pull-up resistor, it will hang at around 1.7V instead of the expected 5V because the onboard LED and series resistor pull the voltage level down, meaning it always returns LOW. If you must use pin 13 as a digital input, set its `pinMode()` to INPUT and use an external pull down resistor.

**Arduino - DigitalPins**

---

**Answer** by **gwideman**

What everyone else said, plus: if you find the digital-13 LED gives you a problem, you can always remove it from the circuit, either by removing the LED, or the series resistor.

This might be useful if you are running out of digital I/Os, or you want several contiguous I/O bits and don’t want one of them to behave differently from the others.

---

**Tags:** **arduino-uno** (Prev Q) (Next Q)

---

**Q: Reset an Arduino UNO by an command (software)**
Is it possible to reset an Arduino (i.e., to reboot it) from code (i.e from the sketch itself)? I know that is possible with a special circuit but is there a chance to make it just with code?

Below is my code and the comment //reset is where I want to force a reset.

```cpp
#include <TrueRandom.h>

int i;
int randSeed;
long randNumber;

void setup(){
    Serial.begin(9600);
    Serial.println(20 " pseudo Zufallszahlen:");
    for (i=1;i<=20;i++) Serial.print(random(10));
    Serial.println();
    Serial.println();
    //randomSeed(TrueRandom.random());
    randSeed = analogRead(A0);
    randomSeed(randSeed);
    Serial.print("Der 'seed' Wert: ");
    Serial.println(randSeed);
    Serial.println();
    Serial.println("20 Zufallszahlen mit analogem 'seed' Wert:");
    for (i=1;i<=20;i++) Serial.print(random(10));
    Serial.println();
    Serial.println("-----------------------------");
    Serial.println();
    delay(500);
    //reset
}

void loop() {
}
```

I want to reset the microcontroller at the end of the setup-function to show the effect of random numbers with and without a seed.

There are three ways to accomplish this. (last is my favorite)

1) Jumper an unused IO to the RESET pin. Leave it as INPUT for normal run, As it is externally pulled high. And when desired to reset set it as LOW and Output. (bang its rebooting).

```cpp
setup() {
    pinMode(PINtoRESET, INPUT); // Just to be clear, as default is INPUT. Not really needed.
    digitalWrite(PINtoRESET, LOW); // Prime it, but does not actually set output.
    ... // Does disable 10K pull Up, but who cares.
    ...
}
```

then when desired...
2) Jump to beginning of the code.

```c
void(* resetFunc)(void) = 0; // declare reset fuction at address 0…
resetFunc(); //call reset
```

**But be careful**, this does not perform a true reset, in that all the registers ARE NOT DEFAULTED. Rather they and the IO are left as is. Where somethings from the bootloader and then the heap will be initialized. And reset are not!

3) Use the watchdog. The [SoftReset](https://example.com) library makes it easy. Although it is not difficult to implement directly. Shown below..

```
#include <avr/wdt.h>
...
setup() {
  ...
  MCUSR = 0; // clear out any flags of prior resets.
  ...
then when desired…
...
  wdt_enable(WDTO_15MS); // turn on the WatchDog and don't stroke it.
  for(;;) {
    // do nothing and wait for the eventual…
  }
...
```

**Tags:** arduino-uno (Prev Q) (Next Q), software (Prev Q) (Next Q)

---

**Q: Convert a Bitstring into an integer value**

**Tags:** arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q)

I have got a String filled up with 0 and 1 and would like to get an Integer out of it:

```c
String bitString = "";
int Number;
int tmp;

bitString = "";
for (i=1;i<10;i++)
{
  tmp= analogRead (A0);
  bitString += tmp % 2;
  delay(50);
} // now bitString contains for example "10100110"
// Number = bitstring to int <--------------
// In the end I want that the variable Number contains the integer 166
```

**Tags:** arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q)

**User:** kmliv

---

**Answer** by craig

If you only need the string for printing you can store value in an integer and then use the
Serial.print(number,BIN) function to format the output as a binary value. Appending integers to strings is a potentially costly operation both in performance and memory usage.

```cpp
int Number = 0;
int tmp;
for (int i=9;i>=0;i--)
{
    tmp = analogRead (A0);
    Number += (tmp % 2) << i;
    delay(50);
}
Serial.print(Number, BIN);
```

**Answer** by doowybbob

Check out strtoul()

It should work something like this:

```cpp```
unsigned long result = strtoul(bitstring.c_str(), NULL, 2);
```

Now you have a long variable which can be converted into an int if needed.

**Tags:** arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q)

---

**Q: Use Higher Resistance Than Instructed in a Circuit**

**Tags:** arduino-uno (Prev Q) (Next Q), led (Next Q)

I’m following the “Get Started with Arduino” Guidebook, and I’m at the part where pulse width modulation (PWM) is discussed.

The instructions say to create a simple circuit with a breadboard, red LED, arduino (connected at pin 9 and ground) and a 270 ohm resistor. My power source will be the usb connection.

I have a 510 ohm resistor.

1. May I use it instead?
2. In general, will using a resistor with a higher resistance than intended in a circuit fry the arduino?

Thanks very much for your help. I’m new to arduino and electronic circuits. I have a Uno R3.

**Tags:** arduino-uno (Prev Q) (Next Q), led (Next Q)

**User:** neuron

**Answer** by chris-stratton

The cases where using a higher value resistor will damage a circuit exist, but are a bit less usual than the cases where it may simply produce a weaker result than desired, or a different frequency response than desired.

The primary situations where you do have to watch out come about where “feedback” is
used to adjust one thing in relation to something else, and so putting a larger resistor on the “sense” side of the circuit can cause the “output” to be too high. Examples you might run into would be adjustable voltage regulators (obvious potential for damage with wrong output) and op-amps (higher than desired output, though more often distortion than damage).

It seems quite likely the the role of the resistor in your circuit is to limit the current through the LED - protecting that (and to a more vaguely best-practices degree, the ATmega pin) from damage.

For your needs it could be worth noting that a parallel combination of two 510 ohm resistors will give an effective resistance of half that, or 255 ohms. This is very close to the design intent, so might be a decent substitution (cheap resistors have a larger tolerance for piece-to-piece variation of value than that anyway).

Normally I would say the desired 270 ohms feels a little small, and that I’d prefer to start with 330 ohms. But if you do parallel two 510’s, the 255 ohms would give just under 20ma of current if connected directly across a 5v supply. Given your LED will have some forward voltage drop (varies widely with color), and your Arduino pin a little of its own, the actual voltage and thus current will be less - as a guess the end result is likely to be in the 15ma range, and not unreasonable for many common LEDs.

Resistors would also be a fairly easy component to salvage out of old electronics - the leads will end up short, but other than that you basically just need to read the color code, and perhaps confirm the value after removal with a meter.

Ultimately though it is worth buying an assortment of resistors - the 1/4 watt size are generally preferred for breadboarding as the leads are the best fit.

---

**Answer** by wayfaring-stranger

If the resistor is in series w the LED, it’ll just be about half as bright when you use 510 vs 270 ohms. I=E/R and all -> half the current. Your PWM will still work, but starting from a lesser brightness on down.

---

**Tags:** arduino-uno (Prev Q) (Next Q), led (Next Q)
**Q: What is Serial.begin(9600)?**

I know that this is to print something:

```
Serial.begin(9600);
```

But I want to know what it really means?

**User: shajib0o**

**Answer by heypete**

Serial.begin(9600) doesn’t actually print anything. For that you’d want to use Serial.print("Hello world!") to print the text “Hello world!” to the serial console. Rather it initializes the serial connection at 9600 bits per second.

Both sides of the serial connection (i.e. the Arduino and your computer) need to be set to use the same speed serial connection in order to get any sort of intelligible data. If there’s a mismatch between what the two systems think the speed is then the data will be garbled.

9600 bits per second is the default for the Arduino, and is perfectly adequate for the majority of users, but you could change it to other speeds: Serial.begin(57600) would set the Arduino to transmit at 57600 bits per second. You’d need to set whatever software you’re using on your computer (like the Arduino IDE’s serial monitor) to the same speed in order to see the data being sent.

**Q: Why does increasing resistance make a capacitive sensor more sensitive?**

From the Capacitive Sensor Page on the Arduino website, it states the following:

The capacitiveSensor method toggles a microcontroller send pin to a new state and then waits for the receive pin to change to the same state as the send pin. A variable is incremented inside a while loop to time the receive pin’s state change. The method then reports the variable’s value, which is in arbitrary units.

**AND**

Use a 1 megohm resistor (or less maybe) for absolute touch to activate. With a 10 megohm resistor the sensor will start to respond 4-6 inches away. With a 40 megohm resistor the sensor will start to respond 12-24 inches away (dependent on the foil...
size)…

However, as mentioned above, what I’m measuring is the time taken to change state, and I understand that more resistance = more time taken = higher value, how exactly does increasing the resistance make the sensor more sensitive, allowing it to respond without even touching it (i.e., inches away)?

Am I not simply increasing the time taken to change state by increasing resistance? How would that make it more sensitive?!

Tags: arduino-uno (Prev Q) (Next Q), sensors (Prev Q) (Next Q), library (Next Q)

User: kenneth-.j  

---

**Answer** by ignacio-vazquez-abrams

A capacitor is created from two conductors with a dielectric between them. The capacitance of the capacitor is determined by the surface area of the conductors and the distance between them. The larger the conductive surface, the higher the capacitance. **The further apart the conductors, the smaller the capacitance. Using a higher resistance means that you can use a smaller capacitance, i.e. be further away from the sensor,** to get the same results.

Tags: arduino-uno (Prev Q) (Next Q), sensors (Prev Q) (Next Q), library (Next Q)

---

**Q: Arduino ADC Reference Voltage if it is Battery Powered**

Tags: arduino-uno (Prev Q) (Next Q), battery (Next Q), analogread (Next Q)

I am looking to possibly monitor battery power to the Arduino using its ADC. This is fairly straight forward and simple (especially if using the Arduino API); however, if the battery is powering the Arduino and is unregulated externally, won’t the ADC reference voltage constantly be dropping with the battery? In other words, wouldn’t the ADC value would constantly read the same value (the max value) even though the battery would actually be decreasing in voltage?

If this is the case, it would be both inefficient and pointless to measure the battery voltage.

Tags: arduino-uno (Prev Q) (Next Q), battery (Next Q), analogread (Next Q)

User: ryeager  

---

**Answer** by ignacio-vazquez-abrams

… won’t the ADC reference voltage constantly be dropping with the battery?

Yes, which is why you either use or measure an internal bandgap reference instead.

Use the `analogReference()` function to select a reference appropriate for the board in use. Note that you will need to use a voltage divider to reduce the battery voltage to a value below that of the selected reference if you wish to measure it.
To measure the bandgap voltage instead (using $AV_{CC}$ as a reference and working “backwards”) you will need to set $MUX[3:0]$ in ADMUX to 0b1110 and then perform an ADC reading directly (set $ADSC$ in ADCSRA and wait until it resets, then read from $ADC[H:L]$).

As always, see the MCU datasheet for details.

Tags: arduino-uno (Prev Q) (Next Q), battery (Next Q), analogread (Next Q)

Q: Reducing lag between the arduino and a processing sketch on my computer

Tags: arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q), serial (Prev Q) (Next Q)

I’m currently on project #14 of the Arduino project book.

I’m trying to control a processing sketch on my laptop using my Arduino. This is accomplished by using a potentiometer to control the background of an image.

Arduino code:

```c
void setup(){
    Serial.begin(9600);
}
void loop(){
    Serial.write(analogRead(A0)/4);
}
```

Processing:

```c
//imports serial library
import processing.serial.*;
//sets up the serial object
Serial myPort;
//creates an object for the image
PImage logo;
//variable to store background color
int bgcolor = 0;
void setup(){
    colorMode(HSB,255);
    logo = loadImage("http://arduino.cc/logo.png");
    size(logo.width,logo.height);
    println("Available serial ports");
    println(Serial.list());
    myPort = new Serial(this,Serial.list()[0],9600);
}
//equivalent of arduino's loop function
void draw(){
    if(myPort.available() > 0)
    {
        bgcolor = myPort.read();
        println(bgcolor);
    }
    background(bgcolor,255,255);
    image(logo,0,0);
}
```

Now, while the code works, and the background color changes as I turn the potentiometer, there is a huge lag between turning the potentiometer and seeing the background change.
color, and the values from the Arduino/potentiometer change on the processing’s serial monitor.

What I’ve tried:

- Changing the speed of Serial communication

I’ve noticed that when I decrease the speed of Serial communication, e.g. around 100, the delay between turning the potentiometer and seeing it change on my laptop decreases to about 1 second. However, when I decrease the speed of Serial communication even further, e.g. a value of 1, the delay increases again.

On the flip side, at the standard speed of 9600, the delay is huge, roughly 5sec ++ before the changes in the potentiometer show up on the laptop/processing.

Why does decreasing the communication speed (up to a certain point) decrease the time lag, and increasing it increase the time lag? Also, is there anyway I can make it near instantaneous?

**Tags:** arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q), serial (Prev Q) (Next Q)

**User:** kenneth-.j

---

**Answer** by peter-r.-bloomfield

You’re outputting a reading every time round the Arduino `loop()`, so it seems likely that your Processing program isn’t running fast enough to keep up with it. Try putting a delay into the `loop()` in your Arduino code to slow it down, e.g.:

```cpp
void loop(){
    Serial.write(analogRead(A0)/4);
    delay(50);
}
```

As far as I know, Processing aims to run at a consistent framerate, which you can modify using the `frameRate()` function. By default, it’s 60 frames per second, although it may run slower on older systems (or where you’re running an intensive program). You can check how fast it’s running by reading the `frameRate` variable.

Introducing a 50 millisecond delay into the Arduino loop means it will be updating a little under 20 times per second. That means it should be fast enough for user interface purposes, but should also be well within the capabilities of your Processing program.

As far as the baud rate (communication speed) is concerned, adjusting it by arbitrary amounts is likely to have unpredictable results. That’s because the hardware will only support specific speeds, and trying to use anything else can result in the data appearing garbled on the other end. The `Serial.begin()` documentation has some more information about supported baud rates.

---

**Answer** by tom

As already pointed out, your Arduino is saying too much too fast. Adding `delay()` will slow it down, but still it keeps yelling at Processing. Ideally, you want Processing to ask
for the value when it’s convenient, and then receive one answer from your Arduino.

Enter SerialEvent().

As opposed to loop() on your Arduino and draw() in Processing, everything inside serialEvent() only executes when there is something new in the serial buffer. So instead of Processing asking questions as fast as possible and your Arduino yelling back even faster, they can have a nice, polite (asynchronous) conversation.

Both Processing and Arduino have a serialEvent. This is SerialEvent() on the Arduino and this is SerialEvent() in Processing. Using SerialEvent on both sides, this is what would happen:

1. Processing sends a character to the serial connection. This could be any character, but if we predetermine one we can filter out any unwanted requests caused by e.g. a noisy signal. For this example, let’s send a V every time we want a new reading of your potmeter. After the character is sent, we continue our business as usual. Not waiting around for an answer here!

2. On the Arduino side nothing is happening, until it receives data in the serial buffer. It checks whether the incoming character is a V, and lucky us, it is. The Arduino reads the potmeter’s value once, outputs that value to serial once, and goes back to chilling out, maxing relaxing all cool. Protip: terminate the value with a character (* in our case). This will help you in the next step.

3. Processing is doing its regular interfacey pixel business when all of a sudden there’s a disturbance in the force new data in the serial buffer. It switches to serialEvent(), and starts reading the serial data, until our terminating * is encountered. Knowing for sure this was the last character worth reading, we can now store the incoming value in a variable that stores the Arduino’s reading.

4. That’s it. Processing now knows the new sensor value and carries on with whatever we tell it to do. Meanwhile, your Arduino is enjoying the weather or contemplating its existence until there is incoming serial data.

Answer by volker-siegel

Your polling loop runs at full speed of your processor, and writes to the serial port in each round.

This way, you’re writing way more often to the serial port than it can handle.

The port writes out data as fast as you configured it, and buffer data that is comming in from your program too fast, to write it out as soon as possible. It the buffer is full, it just drops new data.

What’s important here is that it will keep the order of the values: It is a FIFO buffer, working in First In/First Out order.

What happens is:
The loop fills the port buffer, and keeps it 100% full.
If you turn the potentiometer, the changed value get’s written to the end of the buffer,
the port works as fast as it can to write out all elements in the buffer, that have still the old value.

And finally the value you are interested in. The most current value we wanted to see immediately was at the end of the FIFO, and first in/first out also means last in/last out. The opposite of what we want.

The maximum frequency it makes sense to read your data, is the frequency you can write it out, so you should use at least a delay that is long enough to write out the bytes at the current port speed.

As another, independent measure to prevent this kind of delay in general, you could additionally set the write buffer of the port to a minimum.

That would cause data to be dropped much earlier, instead of buffering a lot first.

Of course, in many applications that is not what you need; With bad luck, it could work anyway in the beginning, and get unstable in some situations when the timing changes based on things like processor load, and there are only some random data samples that get dropped. A large buffer generally behaves much more deterministic, so do use a large buffer by default.

Q: How to measure voltage with Arduino analog input

I’m totally new to Arduino and to electrodynamics in general.

I’m wondering what happens if I connect my digital pin to an analog input on Arduino, and make digitalWrite and analogRead on the according pins. Here are the options I can think about:

- Just connect them directly - it is not a short circuit, is it?
- Connect a resistor in between - the voltage should still be 5v, right? Whatever resistor I use?
- Make a circuit: digital output -> resistor -> ground, and connect the analog input in parallel (in between the resistor and ground) - Does it make sense? What is the difference with the previous option?

As you see, I’m really confused now. I would highly appreciate if someone could explain in detail what is happening in all the cases, with corresponding values for the current, voltage and resistance.
You can connect digital pin to analog input directly. That’s not very interesting though, since you’ll only see two values from the analog input (at least theoretically).

The digital pin gives out 0 or 5 Volts. The analog pin expects 0 to 5 Volts input. The only thing that matters to the analog input is the *voltage* with respect to ground. That can be between 0 to 5 volts (returned as 0…1023 by `analogRead` function).

Just connect them directly - it is not a short circuit, is it?

No, it’s not a short circuit. In general, connecting *anything* to an input is not a short circuit. Connecting output to another output is potentially a short circuit.

Connect a resistor in between - the voltage should still be 5v, right? Whatever resistor I use?

This makes no difference compared to connecting them directly, unless the resistor value is very high (should be hundreds of kOhms or even MOhms to see a difference, I suppose), because almost no current flows to the input.

Make a circuit: digital output -> resistor -> ground, and connect the analog input in parallel (in between the resistor and ground) - Does it make sense? What is the difference with the previous option?

This makes no other difference than that the resistor loads, i.e. draws current from, the digital output. If the resistor value is too small, then the digital output is unable to supply enough current so the 5 V voltage could become less than 5 V, or you could even burn the chip.

---

**Tags:** [arduino-uno](https://www.en.wikipedia.org/wiki/Arduino) (Prev Q) (Next Q), [electricity](https://www.en.wikipedia.org/wiki/Electricity) (Prev Q) (Next Q)

---

**Q: What is the work of `pulseIn`?**


I have code for an ultrasonic sensor which I found from a site. Here is the code:

```cpp
#define trigPin 12
#define echoPin 13

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  int duration, distance;
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
}
```
if (distance >= 200 || distance <= 0){
    Serial.println("Out of range");
}
else {
    Serial.print(distance);
    Serial.println(" cm");
}
delay(500);

However, I don’t understand the work of pulseIn() function. I mean, I want to know when the time count starts and when it ends. For example, in this code, does the time start at digitalWrite(trigPin, HIGH); or does the time start at the pulseIn() function?

If it’s the second one, when it stops, then how does the time give us the distance of an obstacle when I am already delaying 1000 microseconds after I send a ping in air?

Tags: arduino-uno (Prev Q) (Next Q), sensors (Prev Q) (Next Q)

User: shajib00

Answer by sachleen

From the docs:

Reads a pulse (either HIGH or LOW) on a pin. For example, if value is HIGH, pulseIn() waits for the pin to go HIGH, starts timing, then waits for the pin to go LOW and stops timing. Returns the length of the pulse in microseconds.

So, in this case, pulseIn(echoPin, HIGH) starts counting the number of microseconds until echoPin goes HIGH and stores that in duration.

It starts and ends on that line, it’s what is known as a blocking function. It will really sit there until echoPin goes high and tells you how long it took (unless you specify a timeout).

That also means that any delays you have before or after the pulseIn call don’t affect it in any way.

The way you get distance from this time is by the following equation:

distance = (duration/2) / 29.1;

You divide by two because it goes out and back so the time would be double that of a one-way travel. The 29.1 is some sort of calibration factor to relate duration to actual distance. You could probably figure it out by looking at the data sheet of the sensor or just get a lot of samples relating duration to distance (you would measure the distance manually) and get an equation that is very similar.

Tags: arduino-uno (Prev Q) (Next Q), sensors (Prev Q) (Next Q)

Q: Is it possible to press a button from Arduino and my computer will get a key event?
My idea seems easy:

I connect a button with arduino, then write some code and upload it to arduino. Then I open the notepad(or any other editor) on my computer. When I press the button, there will be a character “A” displayed in the notepad.

I have tried with my Arduino uno, and use Serial.write/print function to send an A to serial port 9600, but it seems the character A only shows in the serial monitor GUI.

My question is:

1. Is “Arduino uno” is OK to implement this?
2. If not, what other board should I use?
3. Is Serial.write/print the correct function?
4. Or shall I use the Keyboard??? function?
5. Do I have to write an application which runs on the computer and listens the serial port, and emit a real key event? (that’s sad)

---

**User:** freewind

**Answer** by sachleen

The Uno is not the best board for this. You may be able to get somewhere with the Virtual USB project but there are a couple of boards that are better suited for this.

Those are the Arduino Leonardo, Micro, and Due. In those, you can use the Keyboard Library just like you are using the Serial library now. This page has an example very similar to your project:

```cpp
void setup()
{
  // make pin 2 an input and turn on the
  // pullup resistor so it goes high unless
  // connected to ground:
  pinMode(2, INPUT_PULLUP);
  Keyboard.begin();
}

void loop()
{
  //if the button is pressed
  if(digitalRead(2)==LOW){
    //Send the message
    Keyboard.print("Hello!" motive);
  }
}
```

---

**Answer** by madivad

There is a micro designed more specifically for keyboard emulation, I haven’t done any experimenting with it (yet), but it’s very popular in the flight simulator world where these micro’s are rigged up with buttons and switches and rotary encoders to simulate a cockpit. These inputs are converted to keyboard inputs (as far as I can see).

There are some general projects here: [https://www.pjrc.com/teensy/projects.html](https://www.pjrc.com/teensy/projects.html)
Some more keyboard specific stuff: https://www.pjrc.com/teensy/usb_keyboard.html

More info on the teensy in the arduino environment: https://www.pjrc.com/teensy/teensyduino.html

More specifically about what you’re looking for, this could be quite useful: https://www.pjrc.com/teensy/td_keyboard.html

I’m not steering you away from Arduino, it’s just something else to look into

Q: What is causing my Uno to wake up?

I have a simple piece of code that puts the Arduino to sleep. When it is finished and integrated into my project, it will only wake the processor up when an alarm from an external RTC triggers one of the hardware interrupts.

```c
#include <avr/sleep.h>
#include <avr/power.h>

void setup(){
  Serial.begin(9600);
}

void loop(){
  set_sleep_mode(SLEEP_MODE_PWR_DOWN); //deep sleep mode
  Serial.println("Device is going to sleep...");
  sleep_enable(); //put device to sleep
  sleep_disable(); //Device restarts here on wake
  Serial.println("Device is awake...");
  delay(1000); //to slow down serial output
}
```

In theory, the program should shut down the processor indefinitely because there is nothing available to wake it up.

However, the device continues waking itself up immediately after shutting down.

I have tried attaching the interrupts hoping it was some kind of anomalous watch-dog function. I have tried tying the interrupt pins to +5V and also (in a separate attempt) to GND. Nothing has helped.

What is causing the processor (and/or program) to constantly wake itself up?
sleep_enable() allows the MCU to sleep, but it does not cause it to sleep. Call sleep_cpu() after enabling.

Tags: arduino-uno (Prev Q) (Next Q), programming (Prev Q) (Next Q), interrupt (Prev Q) (Next Q), hardware (Next Q)

**Q: Arduino Servo won’t move when using classes**

Tags: arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

I’m trying to make a class work with Arduino, however it doesn’t seem to trigger properly. The code is compiling perfectly, and it’s supposed to control the leg of an hexapod. Also, the example sweep works on this servo, so no problem here. However, the servo is not actually moving. I’m thinking it has something to do with the order of declaring the variables:

Skip code block

```cpp
#include <Servo.h>

// Abstract each leg, so only those functions can be accessed
class Leg {
    // Actions that can be triggered from outside
    public:
        Leg(int hipPin, int kneePin);
        void up();

    // These should be only available for the leg
    private:
        Servo hip;
        Servo knee;
    }

Leg::Leg(int hipPin, int kneePin) {
    hip.attach(hipPin);
    knee.attach(kneePin);
}

// Testing the leg: move it up
void Leg::up() {
    for(int pos = 0; pos < 180; pos += 1) {
        hip.write(pos);
        delay(15);
    }
}

// Initialize the leg
Leg topLeft(9, 10);

void setup() {
}

// Test the function up
void loop() {
    topLeft.up();
}
```

I’m testing only the pin 9 (leaving the 10 empty). It seems in pain, meaning, you can see the servo trembling but not moving really at 0 degrees.

Tags: arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

User: francisco-presencia
It looks this problem may be due to bad order of initializers calls for global variables.

In C++, the order of global variable initialization is unpredictable across different compilation units (i.e. C++ source files, not header files). Initialization order is only respected inside one compilation unit.

In your program, there are several global variables that, if initialized in the wrong order, will fail your program:

- `Leg topLeft;` this is your class instance
- `servo_t servos[MAX_SERVOS];`, `int8_t Channel[_Nbr_16timers ];` and `uint8_t ServoCount = 0;` all defined in Servo.cpp

Calling `Servo.attach()` requires that the 3 variables above have already been initialized, which you cannot be sure of.

How can you fix that?

By not performing the `attach()` in `Leg` constructor but delegating it to some kind of `init()` method that you can call from `setup()`:

```c
Skip code block
```

```c
class Leg {
    // Actions that can be triggered from outside
    public:
        void init(int hipPin, int kneePin);
        void up();

        // These should be only available for the leg
    private:
        Servo hip;
        Servo knee;
};

Leg::init(int hipPin, int kneePin) {
    hip.attach(hipPin);
    knee.attach(kneePin);
}

// Testing the leg: move it up
void Leg::up() {
    for(int pos = 0; pos < 180; pos += 1) {
        hip.write(pos);
        delay(15);
    }
}

// Declare the leg
Leg topLeft;

void setup() {
    // Initialize the leg
    topLeft.init(9, 10);
}

// Test the function up()
void loop() {
    topLeft.up();
}
```

**Tags:** arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q)
Q: How to send numbers to Arduino UNO via Python 3 and the module serial

Tags: arduino-uno (Prev Q) (Next Q)

I am new to Arduino (and computer programming in general), so I apologize if this question looks silly.

Once I set up a basic arduino-LED connection, I have problems sending INTEGERS to arduino through the serial port. I can easily send characters such as ‘m’, ‘o’ and so on.. However if I send a number it looks like it simply does not get it.

Here is the Arduino code, ideally it should get a value from the usb port through python or the serial monitor and then adjust the brightness of the LED according to the value. (value must be in range [0,255]).

NOTE: I am using ARDUINO UNO and PYTHON 3

```c
//––––––––—
Skip code block
int LED = 10;
int number;
void setup(){
pinMode(LED,OUTPUT);
Serial.begin(9600);
}
void loop(){
number = Serial.read();
Serial.print(number);
Serial.print('
');
if(number == -1){
number = 0;
}
else if(number > 255){
number = 255;
}
else if(number < 0){
number = 0;
}
analogWrite(LED,number);
delay(1000);
}
```

However, when I input a value into the Serial port or through Python, for instance 0, it gives me 48 as answer (which, interestingly, is the ASCII code for 0!) and lights up the LED which is not what should happen since at 0 the LED should be off!! I am missing something but I cannot find what… Could you please tell me what is wrong?

Here is the code I use in Python:

```python
import serial
import time
```
try:
    arduino = serial.Serial(port, speed)
    time.sleep(2)
    print("Connection to " + port + " established successfully!
")
except Exception as e:
    print(e)
#Note: for characters such as 'a' I set data = b'a' to convert the data in bytes
#However the same thing does not work with numbers...
data = 0
data = arduino.write(valueToWrite)
time.sleep(2)
arduino.close()

what am I doing wrong or misunderstanding? Thank you.

Tags: arduino-uno (Prev Q) (Next Q)

User: mickkk

Answer by ignacio-vazquez-abrams

Parsing on the Arduino can be slow and time-consuming (which is bad if you use clock prescaling or have time-critical tasks), so let’s do it in Python.

The problem is that you’re sending the numbers as ASCII whereas you need to be sending them as raw binary. This is where struct comes in.

```python
3>> import struct
3>> print(struct.pack('>B', 0))
b'\x00'
3>> print(struct.pack('>B', 255))
b'\xff'
3>> print(struct.pack('>2B', 255, 0))
b'\xff\x00'
3>> print(struct.pack('>H', 9000))
b'#('```

So what you really want is:

```python
data = arduino.write(struct.pack('>B', valueToWrite))
```

or something to that effect.

Tags: arduino-uno (Prev Q) (Next Q)

Q: Can tx and rx pins on the uno be used like regular digital pins?

Tags: arduino-uno (Prev Q) (Next Q)

The uno has digital pins marked 0-13.

0 is marked as rx and 1 is marked as tx. Can these two pins be used as regular digital pins if i am short of digital pins?

Tags: arduino-uno (Prev Q) (Next Q)

User: c_breeez

Answer by jfpoilpret
Yes you can perfectly use these 2 pins as long as your program does not use Serial.

EDIT: Serial data over USB goes through copper traces connected to the rx and tx pins, connecting them to the USB to serial converter chip.

Answer by sdcharle

Also you will probably want to ensure these wires are not connected when programming the Arduino. Otherwise you may have problems.

Tags: arduino-uno (Prev Q) (Next Q)
Q: Why bootloader sizes differ from 0.5 to 8 kilo bytes for different boards?

Tags: arduino-uno (Prev Q) (Next Q), bootloader (Next Q)

According to official web info bootloader sizes are 0.5, 2 and 8 kilo bytes for UNO, Pro mini and ATMega2560. I believe all three boot loaders do similar job of receiving serial link data and write to Flash memory.

UNO and Pro mini use same/similar ATMega168/328 chips but boot loader size are 0.5 and 2 KB.

ATMega2560 apparent has a more advanced MCU which presumably should use similar or even less memory to do the same task, but the size is significantly large at 8 KB.

Why sizes differ? Small difference may be explained by different development teams, but, should be that much different, from 0.5 to 8KB.

Tags: arduino-uno (Prev Q) (Next Q), bootloader (Next Q)

User: john-williams

Answer by gerben

Bootloader of the Mini and Uno are pretty much the same. (I can’t find the source-code but the hex files are only slightly different).

The 2k is because they haven’t updated mini328.upload.maximum_size in boards.txt. Probably because that would break the board for users that haven’t updated the bootloader yet.

The 2560 is using a stk500v2 compatible boot loader instead of Optiboot (uno and mini), because optiboot only supports upto 64kb addresses. Not sure why it is that much bigger though.

Tags: arduino-uno (Prev Q) (Next Q), bootloader (Next Q)

Q: Is it possible to run a binary from EEPROM?

Tags: arduino-uno (Prev Q) (Next Q), compile (Next Q), sketch (Next Q), flash (Next Q), eeprom (Next Q)

Say I wrote a compiled sketch to EEPROM then read it. Could I run the program from EEPROM? I guess the question is: Can an Arduino run software not in flash memory in the middle of executing the software in flash?

Tags: arduino-uno (Prev Q) (Next Q), compile (Next Q), sketch (Next Q), flash (Next Q), eeprom (Next Q)

User: zeb-mccorkle
Short answer: No; From the atmega328’s data sheet (though it applies to all AVR’s):

AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory

…

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section

It’s architecture prevents using external program memory, but you can load anything into program memory at boot. At that point I would venture to say you are just programming AVR and not arduino, since you would need to replace the arduino bootloader and break the arduino IDE’s ability to upload programs.

Alternately you could also use an emulator or interpreter of some intermediate language; basically code that steps through and runs other code. As a general rule an emulator runs around 8x slower.

There are micro-controllers from other brands that do support this functionality, I know a couple different PIC’s do.

No. The foundation of a Harvard architecture such as AVR is to only allow code that exists within program space to be executed, and EEPROM is not within program space. It is possible, however, to write a virtual machine that will run from flash. This VM can then read program-become-data from anywhere and take action based on it.

Tags: arduino-uno (Prev Q) (Next Q), compile (Next Q), sketch (Next Q), flash (Next Q), eeprom (Next Q)
The real time clock method is the most accurate way but otherwise use millis:

```c
unsigned long startMillis = millis();
while (millis() - startMillis < LONG_DELAY_MS);
```

This will delay up to approx. 4294967295ms (2^32-1) or 49 days, after which the timer will catch up to the value of startMillis.

You could use the watchdog interrupt and have your MCU sleep while waiting and save power.

But notice that you’ll only save power if your board also saves it. That means you have to have a low quiescent voltage regulator instead of the usual regulators that equip the most common Arduino boards, such as the Uno. Otherwise, it doesn’t matter whether your MCU saves energy if your board doesn’t.

Here’s the code (untested):

```c
#include <avr/sleep.h>

volatile int sleep_count = 0; // Keep track of how many sleep cycles have been completed.
const int interval = 720; // Interval in minutes between waking and doing tasks.
const int sleep_total = (interval*60)/8; // Approximate number of sleep cycles
// needed before the interval defined above elapses. Not that this does integer math.

void setup(void) {
    watchdogOn(); // Turn on the watch dog timer.
    // Disable the ADC by setting the ADEN bit (bit 7) to zero.
    ADCSRA = ADCSRA & B01111111;
    // Disable the analog comparator by setting the ACD bit (bit 7) to one.
    ACSR = B10000000;
    // Disable digital input buffers on all analog input pins by setting bits 0-5 to one.
    DIDR0 = DIDR0 | B00111111;
}

void loop(void) {
    goToSleep(); // ATmega328 goes to sleep for about 8 seconds
    // and continues to execute code when it wakes up
    if (sleep_count == sleep_total) {
        // CODE TO BE EXECUTED PERIODICALLY
    }
}

void goToSleep() {
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); // Set sleep mode.
    sleep_enable(); // Enable sleep mode.
    sleep_mode(); // Enter sleep mode.
    // After waking from watchdog interrupt the code continues
    // to execute from this point.
    sleep_disable(); // Disable sleep mode after waking.
}

void watchdogOn() {
    // Clear the reset flag, the WDRF bit (bit 3) of MCUSR.
    MCUSR = MCUSR & B11110111;
    // Set the WDCE bit (bit 4) and the WDE bit (bit 3) of WDTCR.
    WDTCR = WDTCR | B00011000;
    // Set the watchdog timeout prescaler value to 1024 K
    // which will yield a time-out interval of about 8.0 s.
    WDTCR = B00000001;
    // Enable the watchdog timer interrupt.
    WDTCR = WDTCR | B00100000;
    MCUSR = MCUSR & B11110111;
```
ISR(WDT_vect)
{
    sleep_count ++; // keep track of how many sleep cycles have been completed.
}

The code I copied is from this page: Low-Power Arduino Using the Watchdog Timer.

Answer by tom

delay() has its uses, but for long delays it’s no good. It simply tells the microcontroller to do nothing for x clock cycles. During that time, your Arduino can’t do anything else.

Your best bet would be to use a thing called a Real Time Clock (RTC). These chips are specifically made to keep track of time, and you can connect them to your Arduino with ease. Here’s an example of how you could do that.

Tags: arduino-uno (Prev Q) (Next Q), timers (Prev Q) (Next Q)

Q: In Linux, how do I get this Arduino library to work?

Tags: arduino-uno (Prev Q) (Next Q), library (Prev Q) (Next Q), timers (Prev Q) (Next Q), linux (Next Q)

In Linux Fedora 20 Arduino I want to use the Timer library from here, from which I have downloaded the folder Timer-master.zip, but, when I try to import it, I get the Arduino message that it cannot be imported because its name includes characters other than plain ACSII letters and numbers. I have tried copying the file to Timer-master.zip and importing that instead, but still got exactly the same failure message.

Please, how do I get out of this impasse?

Tags: arduino-uno (Prev Q) (Next Q), library (Prev Q) (Next Q), timers (Prev Q) (Next Q), linux (Next Q)

User: harry-weston

Answer by itchyzombie

What’s causing the problem is that the folder name inside the zip has a hyphen in it, just extract the folder from this zip and rename it to timer instead of timer-master. I just imported it on Fedora 20.

Tags: arduino-uno (Prev Q) (Next Q), library (Prev Q) (Next Q), timers (Prev Q) (Next Q), linux (Next Q)

Q: Why does an Arduino UNO need two GND ports?

Tags: arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q), power (Prev Q) (Next Q)

I have really bad eyesight, so whenever I plug in the power wire and the GND wire I
always double check if I’m actually plugging in to the power port not one of the two GND ports.

Why does an Arduino UNO have two GND ports, and for what application would one use both of them?

**Tags:** arduino-uno (Prev Q) (Next Q), pins (Prev Q) (Next Q), power (Prev Q) (Next Q)

**User:** look-behind-you

---

*Answer* by jfpoilpret

In electronics circuits, a major point in connecting two circuits and make them “talk” together is to ensure, at the minimum, that they have a common reference of voltage (typically called GND and defined as 0V voltage).

On Arduino, GND pins can be used in 2 situations:

1. to directly power the Arduino with an external supply which cannot be plugged to USB or power jack. In this case, the positive voltage of that external power source shall also be supplied to Vin pin.
2. to connect Arduino GND to your circuits.

Having 2 distinct GND pins is, I think, just a matter of convenience when you want to build prototypes but don’t necessarily have a breadboard (if you use a breadboard, you will just need to connect its power rail to Arduino GND pin and use this power rail directly).

---

**Q: Salvaged Stepper Motor Blues**

**Tags:** arduino-uno (Prev Q) (Next Q), motor (Prev Q) (Next Q)

So recently my company threw away some rather large and reasonably old laser printers. One was a LaserJet 9050dn the other was a Epson printer/scanner combination. Being a little industrial I thought let me salvage them and brought them home.

I managed to salvage a couple of stepper motors. However I am having huge trouble is getting them to step properly. So let’s get into the details.

Components used:

- 1x Freetronics eleven (Arduino Uno compatible)
- 1X H-Bridge driver shield(Freetronics rated for 40V/2A max).
- 1X Old scanner power supply which has a output of 24V/2A.

Currently there are two stepper motors I am interested in using. They have the following part numbers:
• 1X MITSUMI M42SP-4N
• 1X Minebea-Matshushita 17PM-J212-G2ST

Just to keep this short I am going to focus on the Minebea-Matshushita 17PM-J212-G2ST. I have been unable to find exact matches on these serial numbers when looking for datasheets but I have managed to find the following sheets on the net.

1. [Minebea-Matshushita 17PM-K](#) datasheets found from Minebea. However I cant match the model exactly.
2. [Minebea Stepper Motor Information & Specifications](#) also carries some useful information.

So it appears that these are 24V steppers(in both cases). The Minebea-Matshushita 17PM-J212-G2ST appears to have a step angle of 1.8 degrees which should give me 200 steps per revolution. I have not been able to find the current needed to drive this motor but judging by the looks of things it will be between 0.8A and 1.2A.

I then used my multimeter to find the pairs of wires. It has 4 wires coming out (Bipolar motor I guess). The colours of the wires are red, yellow, orange and blue. First pair was red and yellow with red being positive/anode. Second pair orange blue with orange positive/anode side.

I wired my power supply into my h-bridge and started using the [recipe and quickstart](#) found at the freetonics site. This is where the frustration begins. When asking the motor to do one complete revolution(360 degrees) it does more than 360 degrees at times and other times it does less. I tied a cable tie to the motor so I can confirm it.

So I tried the one-step-at a time sketch to see if I can troubleshoot it. Here is what I have noticed. The steps are mostly correct one step at a time clockwise however every now and again it will jump counter clock wise and then clock wise again. I suppose you can call it a jitter.

Interestingly both these motor show the same behaviour pretty good step action just every 25 degrees or so a jitter. So I suspect that this might not be bi-polar motors but uni-polar? [This blog](#) entry points me to believe that this might be the case. However the datasheet in this case points to a M42SP-7.

So my question would be simple how do I figure out if they are bi-polar or uni-polar?

Can someone help me with a proper datasheet?

Any tips would be really appreciated.

Update:

Note that I can get both motors to step clock-wise and counter clock-wise I just cant get them to do a whole 360 degrees without a jitter. Based on the [Wiring Mitsumi stepper (M42sp-4np) with arduino blog](#) it appears that the Mitsumi is a uni-polar motor with a 2-2 phase excitation. What stumps me is that the wires are paired thus it must be bi-polar?

So last night I tried 2-2 phase excitation with the MITSUMI M42SP-4N and it appears to be solve the problem with this motor in particular. From early observations the motor now
steps without jitter, however it was late so I will need to confirm this.

I was previously using 1-1 phase excitation i.e.

<table>
<thead>
<tr>
<th>Step number</th>
<th>1a</th>
<th>1b</th>
<th>2a</th>
<th>2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

When using 2-2 phase excitation I get a more steady step however this needs to be confirmed a bit more:

<table>
<thead>
<tr>
<th>Step number</th>
<th>1a</th>
<th>1b</th>
<th>2a</th>
<th>2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Just some added observations. The headache continues but I am learning a LOT about steppers motors.

Update:

Not a 2-2 phase excitation these are definitely bi-polar. I am currently watching the Minebea-Matshushita 17PM-J212-G2ST do 360 degrees both clock and counter-clockwise. I hooked them up to a 12V/1A power supply both motors are running smoother and cooler and stepping relatively jitter free. I suspect the whole issue is a lack of a proper data-sheet.

Will update as things progress and once solved add pictures and sketches so the next lucky sod wont loose as much hair as me.

Update:

Thanks to Chris who made me look at the sketch in detail again. These motors seem to have a minimum rpm of sorts. I was trying to go for a really slow rpm under 5rpm at this speed these motors show the jitter effect. However when moving the speed to 10rpm and up no jitter.

Tags: arduino-uno (Prev Q) (Next Q), motor (Prev Q) (Next Q)

User: namphibian

Answer by namphibian

Keeping this here for future reference. It appears these motors have a minimum RPM of sorts and running them on a lower RPM causes the jitter effect described above.

Like I mentioned in the updates when running at 10RPM or above these motors seem to perform very nicely.

Update: I recently bought a ADAFruit motorshield and I am getting much more stable results with this shield and the stepper motors. I am even running them off 6V and they are working perfectly even at lower rpm.
Q: Cannot compile C++11 code via Makefile

Tags: arduino-uno (Prev Q) (Next Q), motor (Prev Q) (Next Q)

I am building my Arduino project with a makefile:

```
BOARD_TAG = uno
ARDUINO_PORT = /dev/ttyACM0
USER_LIB_PATH = /home/prakhar/dev/alfred/arduino/libraries
CPPFLAGS=-x c++ -std=c++11 -Wall -DUNIX_ENVIRONMENT -DHAVE_NAMESPACES -DHAVE_STD
include /usr/share/arduino/Arduino.mk
```

I keep getting this error:

```
cc1plus: error: unrecognized command line option "-std=c++11"
```

I googled and this says that I need to update g++/gcc. I already have the newest versions:

```
prakhar@sim74stic ~ $ avr-g++ --version
avr-g++ (GCC) 4.9.1
Copyright (C) 2014 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
prakhar@sim74stic ~ $ avr-gcc --version
avr-gcc (GCC) 4.9.1
Copyright (C) 2014 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
prakhar@sim74stic ~ $ g++ --version
g++ (GCC) 4.9.1 20140903 (prerelease)
Copyright (C) 2014 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

I installed arduino from the AUR (packages `aur/arduino 1:1.0.6-1` and `aur/arduino-mk 1.3.4-2`). Can anybody point out what I’m doing wrong?

Tags: arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q), compile (Prev Q) (Next Q)

User: prakharsingh95

Answer by prakharsingh95

As I turn out, the avr-gcc (GCC) 4.9.1 goodies weren’t being used at all! The arduino package was using a decrepit version of gcc,

```
prakhar@sim74stic ~ $ /usr/share/arduino/hardware/tools/avr/bin/avr-g++ --version
avr-g++ (GCC) 4.3.2
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
prakhar@sim74stic ~ $ /usr/share/arduino/hardware/tools/avr/bin/avr-gcc --version
avr-gcc (GCC) 4.3.2
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```
All I had to do was specify:

```bash
AVR_TOOLS_DIR = /usr
```

to execute with `-std=c++11` flag (I also had to add an `-fpermissive` flag to compile C with g++).

---

**Q: How do functions outside of void loop work?**

I am used to arduino sketches with a `void setup()` part that runs once, and a `void loop()` part that keeps looping. What happens when you have void functions outside of the main `void loop()`? Will these all keep looping in parallel or do they run one after the other? Or do certain void functions only run once certain criteria has been met (like a while loop)?

For example in the code below, when will the `void receiveData(int byteCount)` and the `void sendData()` functions run?

```cpp
#include <Wire.h>
int number = 0; //Declare variables
int val = 0;

void setup() {
  pinMode(0, INPUT); //Set pin 0 as input and 3 as output
  pinMode(3, OUTPUT);
  Serial.begin(9600); //Set the data rate for serial transmission at 9600bps
  Wire.begin(0x04); //Initiate the Wire library, join the arduino as a slave
  Wire.onReceive(receiveData); //Define callbacks for i2c communication
  Wire.onRequest(sendData);
}

void loop() { //The code between the curly brackets keeps repeating
  delay(100);
}

void receiveData(int byteCount){
  while(Wire.available()) {
    number = Wire.read(); //Set the variable "number" to the data sent by the master
    analogWrite(3, number); //Write this number to pin 3 (PWM). This controls the motor speed
  }
  val = analogRead(0); //Read the voltage on pin 0 (connected to the sensor)
}

void sendData(){
  Wire.write(val); //Send the data read from the sensor to the master.
}```
Answer by peter-r.-bloomfield

The `setup()` and `loop()` functions are unusual because they are called automatically for you by the Arduino code. No other functions behave this way.

Generally speaking, a function will never run unless you explicitly call it yourself (e.g. from within `setup()` or `loop()`), or instruct another part of the program to call it. (There are other ways to execute functions, but that usually involves some very advanced tinkering which is best avoided.)

For example, `pinMode()` is a function just like any other. It only runs when you actually put something like `pinMode(3, INPUT)` in your code. At that point, it runs once, finishes, and then the calling function carries on from where it left off (they never run in parallel).

The example code you’ve posted is quite interesting. Look at these lines in `setup()`:

```cpp
Wire.onReceive(receiveData);
Wire.onRequest(sendData);
```

These lines are telling the `Wire` object to call `receiveData()` and `sendData()` in response to I2C events. It does this by passing function pointers which are stored and used by `Wire`.

I’d recommend searching for information about C/C++ function pointers online if you want to learn more about this. You may also be interested to explore Arduino’s `attachInterrupt()` function.

---

Q: Performance of internal vs external resistors

Is there any significant performance trade-off between buttons wired with the internal vs external pull-up/down resistors? I see a lot of tutorials demonstrating the use of external resistors when wiring a simple button, like [this](#) and [this](#). But since the Arduino/AVR includes internal pull-up resistors, you can wire a button without any external resistor as demonstrated [here](#).

Is there any good reason to complicate your design with an extra resistor?

---

User: cerin

Answer by milliways

If you are using the pin as an input in close proximity to the Arduino there is little reason
to use an external pullup.

There are a number of factors to consider.

If the switch is some distance from the chip and/or in a noisy environment it will be more susceptible to interference as the internal pullups are high impedance (20kΩ to 150kΩ depending on model).

If the pin is used as both input & output the pullup can reduce the voltage swing.

In other applications e.g. Raspberry Pi the state of the pin between boot and software initialisation can be significant, but this should not be an issue with Arduino.

---

**Q: Extremely cheap “Arduino” for single tasks?**

**Tags:** arduino-uno (Prev Q) (Next Q), button (Next Q)

I have been thinking of a million things to do with my Arduino, but those things are extremely simple and it would leave my Arduino as a $25 waste.

For example, programming Arduino to control my living room lights with my smartphone … I was thinking of connecting the Arduino in the wall, onto the light switch wires. I am looking for a “permanent” solution, but I figure this would leave me with a $25 waste of circuitry inside my wall.

Are there any “Arduino” types of electronics that perhaps have many less ports, JUST enough to get a single task done?

**Tags:** arduino-uno (Prev Q) (Next Q)

**User:** excaliber

---

**Answer** by patrick-tudor

Digispark comes to mind, or any of the ATTiny breakout boards at Tindie. Or, make your own PCBs, I just now finished soldering a few ATMega328p circuits to boards I designed. Microcontroller is $1.50, PCBs range from $2 to $8 each, time is “free.”

Also consider “compatible with” designs; while you’re not directly supporting the ecosystem, if you’ve already paid full price once and your conscience is clear, eBay has plenty.

---

**Answer** by goleztrol

You can embed a bootloaded ATMega328 (or any of the other compatible ATMega processors) [http://arduino.cc/en/pmwiki.php?n=Main/Standalone](http://arduino.cc/en/pmwiki.php?n=Main/Standalone) in your project. That way, you will only need the processor and a couple of components, setting you back only a couple of dollars and saving quite a lot of space:

If that is one bridge to far, you can have a look at the Arduino Pro or Arduino Pro Mini. Those boards are complete boards, only smaller and cheaper, and specifically designed for a more definitive use in projects.

If you need even more space and cost reduction, you can switch to a stand-alone ATTiny processor. The better models of those can also contain an Arduino program. Instructables has this great tutorial that shows you how to use an ATTiny85 or ATTiny45 as a lightweight ‘Arduino’.

Recently, I have tried to use an ATTiny13A for a project, but that one only has 1kB of flash memory, which leaves you with hardly any space for the Arduino bootloader, let alone your program, so therefor I chose to program it directly using AVR Studio, at which point I think you cannot in any way call it an Arduino project anymore. :) Still, it’s not that different. You only miss out on some of the library functions, and the implied application loop.

---

**Answer** by federico-fissore

This is a typical scenario for an Arduino Uno board: you buy one and use it for prototyping your project. Once done, you design your own PCB, as small as possible, connecting only the required wires to the ATMega328p.

For the next project, you buy another ATMega328p (just the MCU), plug it on the Uno and start again.

---

**Q: Is there a pullup on I2C?**

**Tags:** arduino-uno (Prev Q) (Next Q), i2c (Prev Q) (Next Q)

I am using an Arduino Uno R3.

It is not clear on the schematic whether there are pullups on the A4 and A5 I2C pins.

Are there hardwired pullups or do I have to add them myself?

**Tags:** arduino-uno (Prev Q) (Next Q), i2c (Prev Q) (Next Q)

**User:** harry_p_6

**Answer** by ignacio-vazquez-abrams

There are very weak internal pullups, but they are not strong enough to be considered compliant with I²C specs. Supply your own external 2.2k-10kohm pullup resistors.

**Tags:** arduino-uno (Prev Q) (Next Q), i2c (Prev Q) (Next Q)
**Q: Powering 5V Accessories**

**Tags:** [arduino-uno](#) (Prev Q) (Next Q), [power](#) (Prev Q) (Next Q)

I’m using an Uno to control NeoPixels LED strips (60 per metre). These are powered with a 5V DC supply. Now, when using USB to power the Arduino it works lovely. However I want to move the Arduino into a standalone “installation” where I’ll only want one power supply.

I’ve already got a 5V DC supply which outputs about 5.2V. What is the best way to power both the Arduino and the LEDs? I understand the need for at least 7V at the Arduino’s voltage regulator input. Would my best bet be to get a 12V supply (or similar) and use a buck converter to power the LEDs? The total current draw *could* reach 3.6A but will be 400mA for the majority of the time.

**Tags:** [arduino-uno](#) (Prev Q) (Next Q), [power](#) (Prev Q) (Next Q)

**User:** cheechm

---

**Answer** by [ignacio-vazquez-abrams](#)

If your supply is regulated (put a 47ohm resistor across it and measure the AC voltage with a DMM; it should be almost nothing if properly regulated) then you can use it for both the Uno and the NeoPixels. The ATmega328P has a maximum input voltage of 5.5V, and the WS2812B has a maximum voltage of 5.3V, putting the supply within the proper range to run them.

**Tags:** [arduino-uno](#) (Prev Q) (Next Q), [power](#) (Prev Q) (Next Q)

---

**Q: How to move from arduino to physical prototype**

**Tags:** [arduino-uno](#) (Prev Q) (Next Q), [sensors](#) (Prev Q) (Next Q)

I’ve been reading a bit about moving from Arduino to PCB and it seems a bit daunting. For now, all I really want to do is take something I’ve built, and make a few changes so that I can actually create a casing for it and make sure everything stays connected.

I’ve attached a photo of my project. What I am trying to do is create a box, where 4 pushbuttons sit in a box, with LEDs next to them. I will probably create foamboard cutouts for the spaces where the pushbuttons and leds sit. The wiring for the push buttons and the LEDs need to be hidden below the foam board, along with the arduino itself and the servo motor - which is where I get stuck.

What do I need to do to essentially “lengthen” the wires for the pushbuttons and LEDs, and make sure all the components stay secure in the breadboard (the push buttons in particular like to pop out)? Is there a way to do this without making a PCB? I understand the concept of soldering components into protoboard, but as far as cutting / printing the circuits or whatever, it is really confusing to me.
I’d appreciate any guidance. If you see anything wrong with my wiring setup too, I’d love a learning experience on that as well. Thanks!
The best thing to do is, of course, to use a PCB. But it doesn’t need to be a custom PCB; there are plenty of pre-made PCBs that are usable for approximately 98% of designs. Here are just a few:

- **Adafruit Perma-Proto boards** are connected just like breadboards, so direct transfer is possible.
- **BusBoard proto boards** come in a *ridiculous* variety of configurations, including enclosure kits, and they even have SMD boards if you’re up to it.

For switches just use a panel mount form. They come in everything from simple to robust to ridiculous.

LEDs can also be panel mount, or you can use a light pipe to make a LED on the PCB visible from the outside.

For power you can use a USB connection or a separate power jack with a voltage regulator. If you also need a serial connection then you should consider a pre-made serial
module, and either use its built-in USB jack or a separate panel mount USB jack. The servo can be mounted inside the case if appropriate, or can be connected to a PCB mount or panel mount connector instead.

Naturally the MCU itself should also be placed in a socket directly on the PCB. Don’t forget the decoupling, the oscillator, and the ISP connector.

As for the circuit itself, you can get rid of the pullup resistors on the switches if you use internal pullups instead. And 100ohm is way too low a value for switch pullups regardless.

Answer by patrick-tudor

Learning to design a PCB can be a challenge. First, there’s software choices, there’s file formats, and then actually laying out a good board. That’s before production and assembly, each with its own struggles along the way. It’s no different than the first time you open Xcode and learn about Storyboards and Frameworks and Simulators and all that, or for that matter, learning any new skill, which at first is a daunting experience but through both setbacks and success eventually becomes easy.

So, I agree with everyone that suggests perfboards and variants. But I also want to show, as perhaps motivation, where you could be in a few months. I tend to see Arduinos as the development tool, not the final product, but I understand others are content to just use a standalone Nano or Pro Mini and forever dedicate it to their project.

These are three variants of the same device, it combines an Arduino/ATMega, a GPS, a serial port, and an RGB LCD. I first created it in on a breadboard, then moved to a perfboard, thought “that is ugly” and made a simple PCB. Pleased with that experience, I kept refining the schematic and layout until I finally called the project “done.”
You also might consider making tiny PCBs just for specific components, like these two boards I make for buttons and lights.
The very first PCB I ordered from OSHPark was an amazing experience, opening the envelope and holding it like some religious relic. But since then it’s become routine. If you do later go the PCB route, you might appreciate my summary of low-volume shops I’ve ordered from.

Answer by jfpoilpret

Using a **stripboard** is perfectly convenient to create a circuit that:

- will last longer than a breadboard prototype
- will cost less than wasting a breadboard for a long time

Stripboards can be found in any good electronics retailer (I buy mines at Conrad). You will need a soldering iron (and the skills that are needed for it, but that is not so difficult as some people think).

You could even go one step further and create an Arduino on the same stripboard (many components from the UNO are not required for a circuit you want to build and keep “forever”).

Another option, just a little bit more expensive, would consist in affording a **prototype shield** for Arduino, solder you circuit onto it and directly plug it onto your Arduino. The principles are the same as for a stripboard, except the prototype are has typically no strip, only holes on which you can solder components and wires.

**Tags:** arduino-uno (Prev Q) (Next Q), sensors (Prev Q) (Next Q)
I have this Arduino motor shield which is hooked up with two DC motors. The board is the same as this one from Sparkfun.

A PWM output controls the speed of the motor, and a digital out controls the direction. The problem is that it only works in one direction, namely if the direction pin is high.

With the direction pin HIGH and 255 is written to the PWM pin, Vout on the terminals is similar to Vin. If I reverse the direction by setting the direction pin LOW I would expect Vout to be -Vin, ie. the polarity reversed. But not so. I get a little more than zero volt across the terminals.

It’s the same for both channels.

Edit: The setup is powered from a 9V power adaptor through the barrel jack on Arduino.

Is the board defective, or am I misunderstanding something?

SOLUTION: Thanks to the edited post (especially #3) in the accepted answer, I not only looked at the schematic (again), I actually understood it. I had already noticed the jumper called V-LOGIC which on the picture from Sparkfun was set to 5V where it on my shield was not set. I really didn’t know what it was used for, but I set it to 5V so it was the same. But by looking at the schematic and the inverters, I realized that the inverter was fed from V-LOGIC which was connected to 5V. And bingo: I had not connected the 5V pin on the shield, only vIN which I assumed would be sufficient. The thing is, I am awaiting a set of stackable headers instead of the ones supplied, so I only connected the minimum numbers of pins because I will need to desolder them when I get the new headers.
I think you may be having an issue with how you are powering the shield. Are you trying to do it with just your USB connection?

In your photo, I do not see a connection to your Arduino’s barrel jack or to the Vin connector of the shield itself. According to the SparkFun site:

The Ardumoto Shield should be powered through one of two power supply inputs. Pick one or the other:

1. The **barrel jack** input on the Arduino.
2. The **Vin input** on the shield

If you don’t want to use the Arduino’s barrel jack input, you can use the **Vin input on the shield** instead. This voltage input will supply **both the shield and the Arduino**. Like the motor outputs, this connection is broken out to both a 3.5mm screw terminal and a 0.1”-pitch header.

Do not supply power to both the Arduino barrel jack input and Vin on the shield!

I hope this helps!

**EDIT BASED ON COMMENTS**

Since you are properly powering the shield, I would recommend the following (see the [shield’s schematic](#) for specifics):

1. Maintaining the connection shown above (with the addition of the 9V power supply connected to the barrel jack) ensure that you are able to properly set the DIRA input. You can check this with a multimeter on pin 7 of the LN298 whenever you are trying to switch directions.
2. If you are getting the expected results on pin 7 (pin goes HI when your code wants the motor to go one direction and pin goes LOW for the other) - check pin 9… the value here should always be the opposite of what you are setting for DIRA on pin 7. If that is all good, then you probably have an issue with the LN298.
3. If you are not getting the expected results at pin 9, but pin 7 looks good, then you are having a problem with the 74HC1G04 IC (although I would doubt this since you are having problems with both the A and B motor outputs).
4. If you are not getting the expected results at pin 7, then double check the code to make sure you have set and cleared the correct output pin. I would also check the solder joints of the connectors on the shield - you can set a multimeter to do a resistance check from the shield’s connector (pin 12) to the direction input on the LN298 (pin 7). If the continuity is measuring close to 0 ohms then there is probably an issue with your code.
5. If your code looks good, I have also had issues with bad outputs on the Arduino - you can check to see if you are properly toggling the direction pin (pin 12) on the
Arduino without the shield connected. Then, if the pin is not toggling, there is probably an issue with your Arduino - try toggling a different (unused) pin and jumpering it.

Tags: arduino-uno (Prev Q) (Next Q), motor (Prev Q) (Next Q)

**Q: Is it wise to use analog input pins to read digital buttons?**

**Tags: arduino-uno (Prev Q) (Next Q), analogread (Prev Q) (Next Q)**

I am thinking about building a circuit that is literally going to take 11 of the 14 digital pins of an UNO to communicate with an SD card and an external EEPROM chip (EEPROM programmer).

Using 4 shift registers and the SD card, I will have 1 digital pin left. Well, I’m also reserving pins 0/1 for serial communication.

Anyway, I need to read some external buttons and I have the 6 analog pins that I haven’t touched.

I will only need about 4-5 push buttons in my design so they should fit nicely on A0+.

So, my question is, will I run into any issues doing this? Or, are the analog inputs better left for true analog reading?

Thanks.

**Tags: arduino-uno (Prev Q) (Next Q), analogread (Prev Q) (Next Q)**

**User: cbmeeks**

**Answer** by russell-mcmahon

The Analog pins are essentially identical in functionality to digital pins when used as digital I/O.

If desired you could “be clever” and use an analog input in analog mode to read multiple keys using one pin. eg using 10k, 22k, 39k, 82k, 150k in series with 5 buttons from V+ to pin and a say 4k7 to ground would result in 5 voltages which were easily distinguished using an analog read. This is a very very old technique indeed and allows far more than 5 keys on one input pin. THere are limitations which can be discussed if required.

R1 … R5 correspond to buttons 1…5

\[ V_{out} = \frac{V+ \times R_b}{R_b + R_{button}} \]

Example only

\[ R_b = 39k \]
\[ R1 = 10k \]
\[ R2 = 22k \]
\[ R3 = 39k \]
\[ R4 = 82k \]
R5 = 150k

Gives

- The % of V+ in column 2 when a button is pressed.
- The voltage in column 3 with V+ = 5V

Delta-V is the difference between that key and the one below - very ample clearance.

<table>
<thead>
<tr>
<th>Rbutton</th>
<th>% of V+</th>
<th>V at 5V in</th>
<th>Delta V</th>
</tr>
</thead>
<tbody>
<tr>
<td>10k</td>
<td>80%</td>
<td>4.0</td>
<td>0.8</td>
</tr>
<tr>
<td>22k</td>
<td>64%</td>
<td>3.2</td>
<td>0.7</td>
</tr>
<tr>
<td>39k</td>
<td>50%</td>
<td>2.5</td>
<td>0.9</td>
</tr>
<tr>
<td>82k</td>
<td>32%</td>
<td>1.6</td>
<td>0.6</td>
</tr>
<tr>
<td>150k</td>
<td>21%</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

Answer by milliways

If you have no use for analog pins there is no problem. A5/A6 are used for I²C (on the Uno).

Tags: arduino-uno (Prev Q) (Next Q), analogread (Prev Q) (Next Q)

Q: RTClib library dependency on Wire library

Tags: arduino-uno (Prev Q) (Next Q)
I downloaded the RTClib library from https://github.com/adafruit/RTClib. In every provided example, #include <Wire.h> is written immediately above #include "RTClib.h" at the top of the sketch.

Why is this necessary? I know the RTClib library requires the functionality of the Wire library but, if this is the case, why can't I just #include <Wire.h> at the top of RTClib.h? I have tried the latter, even using just the following sketch, but my sketch failed to compile.

RTClib.h:

```cpp
// Code by JeeLabs http://news.jeelabs.org/code/
// Released to the public domain! Enjoy!

#ifndef _RTCLIB_H_
#define _RTCLIB_H_

#include <Wire.h>
...
#endif // _RTCLIB_H_
```

Sketch:

```cpp
#include "RTClib.h"
void setup() {}
void loop() {};
```

Tags: arduino-uno (Prev Q) (Next Q)

User: zilliput

Answer by brettam

Not getting too formal here, The arduino IDE looks at which libraries are included in the top level sketch when deciding what libraries to include in the compilation.

It would make sense for only RTClib.h to include Wire.h yes, but the IDE does not get the information to add Wire to the compilation list that way. It is a quirk of the IDE that makes implementing libraries that use other libraries a bit more painful.

Tags: arduino-uno (Prev Q) (Next Q)

Q: Using Arduino for Industrial process

Tags: arduino-uno (Prev Q) (Next Q)

I was wondering if the Arduino is applicable for Industrial monitoring, I want to make a real time status of the presses in our company. I haven’t seen or read much about people using Arduino at an Industrial level, most of what I have seen or read are prototype projects. I’m new in electronics and don’t know if there is a better way of doing this rather than using Arduino. I was thinking of using Arduino Uno and Arduino Ethernet Shield or Wifi Shield.

Any help on some information on how to start on this is helpful.
I wouldn’t use the Arduino in an industrial environment, as I wouldn’t trust its PCB to be able to handle much noise. Use the Arduino and shields to prototype your application, and then move to a circuit board that uses proper techniques suited to an industrial environment.

I don’t see any reason not to use Arduino in an industrial, engineering, or laboratory environment, if it meets your needs and does the job well. It is highly recommended, however, that you provide some sort of pin protection to keep industrial noise or voltage spikes from frying pins. Minimum pin protection could consist of using current-limiting resistors (or better yet, the other, more robust techniques shown here) on pins used for output and input signals. Optical isolation is also a good option, especially if potentially dangerous voltages are being used. Lastly, after you prototype and test your Arduino setup you should solder the wires to the Arduino and provide a protective case. You don’t want to accidentally fry the Arduino or a pin via accidental static discharge (hence the case, and pin protection above), and you don’t want jumper wires vibrating loose and connections slowly oxidizing over time and loosing electrical connectivity in any type of industrial or laboratory application that is expected to be used regularly (hence replacing your jumper wires with soldered wires). Shields, however, are prob. ok for low-humidity and room-temperature environments, as they are generally much more secure and solidly connected than individual jumper wires.

Note that you can solder wires to any Arduino, even the Uno, Leonardo, or Mega, for example, simply by soldering to the exposed portions of the pins on the bottom of the board. You can also cut off and desolder the female headers and either solder directly to the holes, or solder on male headers that you then solder to other PCBs. I especially like the Arduino Nano, Pro Mini, and Micro since they are small, have exposed holes or male pin headers, and can easily be soldered onto a professionally-manufactured circuit board or cheap protoboard, for better mounting in a final application. Keep reading below for more details on this.

I am an aerospace research engineer and use Arduino in my line of work regularly to solve problems. Here are some examples of current applications in use today, that I implemented, that are based on and using Arduino boards:

1. **sensorless brushless motor RPM sensor, motor commander, and data acquisition device** for bench-top testing
2. **linear actuator closed-loop controller** (linear stepping resolution is 0.01mm [10um], full moving range is 0 to 15mm); the Arduino commands the actuator driver board via a custom PWM signal using the Timer1 library ([http://playground.arduino.cc/code/timer1](http://playground.arduino.cc/code/timer1)); position is read via a custom quadrature encoder reader algorithm on the arduino; the actuator is used for very
precise aiming of certain instruments; the $30 Arduino is taking the place of a $500
board that the actuator manufacturer makes, and the Arduino gives us much more
functionality than the $500 board would give us. The Arduino and actuator will be
used to precisely aim a laser for experiments in a wind tunnel.

3. **flight controller interfaces** between MATLAB and off-the-shelf Radio Control
transmitter (read and write RC PWM & PPM communication signals)

4. **Professional film studio communication interface on robotic, drivable camera system** (info sheet & video).

Those are just a few cases. If it works, do it. Once you prototype with a solderless
breadboard, however, it is recommended that you at least hand-solder the final
configuration to a permanent prototyping board, to make it more durable and noise-
resistant (ie—primarily: so that it will have solid connections that can’t corrode, break,
come apart, or vibrate free easily).

Here’s a soldering article I wrote
(http://electricrcaircraftguy.blogspot.com/2014/08/recommended-soldering-kit.html#.VP-JCPzF-So). see the links at the bottom for recommended soldering tutorials. The one
about how to make “solder tracks” was especially useful to me once I got past the
breadboard stage and started prototyping things on double-sided perf board.

Here are some double-sided prototyping boards (or “perf boards”) to look into once you
get to this stage: (Ebay search for “double sided pcb”) http://www.ebay.com/sch/i.html?_from=R40&ghostText=&_sacat=0&_nkw=double+sided+pcb&rt=nc&LH_BIN=1 —
get the ones with all the little holes in them. You can put the whole thing in a plastic case
if you want too. You can permanently solder an Arduino (esp. the Nanos or Micros more
easily) to one of these perf boards.

Here’s another example of double-sized perf board. (image source:
http://www.elecfreaks.com/store/doublesided-protoboard-8cm-x-12cm-p-410.html)
Professional PCBs:

If you need to make more than a few of some device, hand-soldering wires and pieces all over a perfboard can get to be VERY tedious. In this case, you need to design a more professional PCB with custom traces and pads and layout and all for your project. If you don’t have PCB-designing skills yet with a professional tool like Eagle or KiCad, you can use Fritzing (http://fritzing.org/home/) to design a PCB, and then send it off to a company like OshPark (https://oshpark.com/) for high-quality, reasonably-priced, low-quantity US-based manufacturing—or send it off to the Fritzing Fab lab in Germany for high-quality PCB manufacturing that supports the Fritzing project. Lastly, for a couple really inexpensive Chinese options, look at ITEAD Studio and Seeedstudio. Fritzing and any other PCB CAD software tool can output the necessary Gerber files for manufacturing.

Quick tip: when designing a PCB, don’t use any 90 deg corners for your traces. They can act as antennas (sending or receiving) and cause electromagnetic (including radio) interference and noise. Use multiple 45 deg edges and corners instead.

Once you get your boards back, solder your Arduino to them, and solder any extra parts or hardware to them you need as well. Now you have quite a professional product, even if it has an Arduino soldered right to it.

Summary:

In summary, yes I think Arduino is great. Without it, I wouldn’t know how to even blink an LED or load a single code to a microcontroller, because getting into using
microcontrollers was just to enigmatic, difficult, and complicated for someone not professionally trained in it, such as an Aeronautical engineer like myself. With Arduino, however, I learned to build up my skills and solve real-world problems in professional environments.

I am very grateful to Massimo Banzi, David A. Mellis, and others who have made Arduino possible. It has changed my life and my ability to be useful in this life. My favorite part about it is that even though it is a tool meant to get the non-programmer or lay-person to do basic stuff like blink a light using poorly-written blocking code, it has opened up my world beyond what I ever imagined before, and it is a fully-capable microcontroller that doesn’t permanently hide away the low-level stuff. The low-level AVR functions, registers, etc are ALL STILL THERE AND ACCESSIBLE once you get to a level where you need them!

Cheers,
Gabriel Staples

PS: To see many of the very helpful links which I’ve referenced as I’ve built up my Arduino and ATmega328 microcontroller skillset, please see this article here, esp. all of the links at the bottom: http://electricrcaircraftguy.blogspot.com/2014/01/the-power-of-arduino.html

Tags: arduino-uno (Prev Q) (Next Q)

Q: What is the physical location of the U1A symbol on Arduino Uno?

I’m trying to find an easy way to allow an Arduino Uno to determine if it’s being powered from USB or the DC barrel jack. In this thread, one user recommends wiring the output of the on-board comparator op-amp (U5A pin 1) into the digital input pin.

I couldn’t find U5A on the current schematic, but I think he’s actually referring to U1A pin 1. Is this correct?

Unfortunately, I’m not sure where on the physical Arduino that device is located. I have one in front of me, and almost none of the parts are labeled on the silk screen. How do I find out where it is? I’ve googled for Arduino Uno part placement schematics, but all I’m finding are blurry unlabeled low-res images.

Tags: arduino-uno (Prev Q) (Next Q)

User: cerin

Answer by rslite

The op-amp is a LM358 so you can look at the ICs themselves. Also it’s a dual comparator with 8 pins. On my example image below you can see it below the crystal. I can’t comment if that’s the correct way to sense what you need, but it seems the comparator output should be 1 if there is external power applied. You could check with a
multimeter.
I would like to make a makefile for compiling C programs for the Arduino. I am somewhat familiar with make but have never used it with avr-gcc. What is the simplest way I could put the commands below in a makefile?

```
$ avr-gcc -Os -DF_CPU=16000000UL -mmcu=atmega328p -c -o led.o led.c
$ avr-gcc -mmcu=atmega328p led.o -o led
$ avr-objcopy -O ihex -R .eeprom led.elf led.hex
$ avrdude -F -V -c arduino -p ATMEGA328P -P /dev/ttyACM0 -b 115200 -U flash:w:led.hex
```

It’s no different to working with Make and any other form of GCC. Just set your CC variable and CFLAGS variable accordingly and work as per normal.

For instance, I just knocked this one up:

```
CC=avr-gcc
OBJCOPY=avr-objcopy
CFLAGS=-Os -DF_CPU=16000000UL -mmcu=atmega328p
PORT=/dev/ttyACM0

led.hex: led.elf
  ${OBJCOPY} -O ihex -R .eeprom led.elf led.hex

led.elf: led.o
  ${CC} -o led.elf led.o

install: led.hex
  avrdude -F -V -c arduino -p ATMEGA328P -P ${PORT} -b 115200 -U flash:w:led.hex
```

That’s saying that any automatic C compilation will be done with avr-gcc and the flags specified in CFLAGS. It will by default make the hex file using OBJCOPY, which is set to the avr one, which relies on the file led.elf - so to get that file it runs the led.elf target, which links the object file led.o with the default libraries using whatever was set in CC. To do that it needs led.o, and it makes that automatically using the program specified in CC and the flags in CFLAGS. You can then optionally to make install which will run avrdude to install the hex file into the chip.

You can make it even more generic so you can copy it in to other projects and make the minimum changes necessary:

```
BIN=led
OBJS=led.o test.o

CC=avr-gcc
OBJCOPY=avr-objcopy
CFLAGS=-Os -DF_CPU=16000000UL -mmcu=atmega328p
PORT=/dev/ttyACM0
```

```
```
That uses “automatic variables” and simple name replacement. BIN contains the “base” of your binary files, OBJS contains the list of object files. $@$ is the name of the current target, $<$ is the name of the first prerequisite, and $^@$ is the list of all the prerequisites. Just change BIN and OBJS to suit. As a bonus I have thrown in make clean to remove the compiled files and just leave you with the source.

Tags: arduino-uno (Prev Q) (Next Q), atmega328 (Prev Q) (Next Q), compile (Prev Q) (Next Q), c (Prev Q) (Next Q)

Q: How to increment and decrement an output voltage by using two buttons?

Tags: arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q), pwm (Prev Q) (Next Q), analogread (Prev Q) (Next Q), hardware (Prev Q) (Next Q)

I’m trying to create a code using an Arduino Uno board to increment and decrement the output voltage of the Arduino Uno which is 5 volts and I need to step it up to 10 volts which I have done below.

The things that are missing on the schematic are two buttons. I want to be able to increment 0.5 volts with each press of Button-A until the Arduino is at 5 volts. With Button-B I want to be able to lower the voltage by 0.5 volts with each press of the button till value is 0.

I can do the calculations and the circuit design but I’m not much of a programmer. I just want to know if anyone can point me in a good direction of where I can get the right code. I have already looked at all the tutorials on the Arduino Website. I do know I have to work with the PWM functions as well as attachinterrupt() and Debounce() functions. Also, how do I set the increments so that I don’t go over, say it takes 10 presses to get to 5 volts on the 11th press have it not do anything so I don’t break anything. So can anyone please help me with any suggestions?

I have a low pass filter on the circuit for noise handling and like I said I need to add in the buttons to my schematic.
Here is my code I have made. I’m using an LED to test the $V_{out}$ aspect. So far I can only get my LED to flicker from 2 to 4 volts and my buttons A and B don’t work. Can someone look over my code and see how bad I made this?

```cpp
// Skip code block
```

```cpp
int PWMPin = 6; // output pin supporting PWM
int buttonApin = 9; // buttonA to pin 9 PWM
int buttonBpin = 10; // buttonB to pin 10 PWM
float value = 0; // read the value at 0
int fadeValue = value;

void setup()
{
  Serial.begin(9600); //initialize serial communication at 9600 baud
  pinMode(buttonApin, INPUT_PULLUP);
  pinMode(buttonBpin, INPUT_PULLUP);
  pinMode(PWMPin, OUTPUT);
}

void loop()
{
  {
    int port = analogRead(0);
    port = map(port, 0, 10, 0, 255);
    analogWrite(6, port);
  }

  if (digitalRead(buttonApin) == LOW, fadeValue)
  {
    // fade from min to max in increments of 25.5 points: basically (0.5 volts)
    for(int fadeValue = 0 ; fadeValue <= 255; fadeValue += 25.5)
    {
      digitalWrite(PWMPin, fadeValue);
      // sets the value (range from 0 to 255):
      Serial.println(PWMPin);
      analogWrite(PWMPin, fadeValue);
    }
  }

  if (digitalRead(buttonBpin) == LOW, fadeValue)
  {
    // fade from max to min in increments of 25.5 points:
    for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=25.5)
    {
      digitalWrite(PWMPin, fadeValue);
      // sets the value (range from 0 to 255): basically (0.5 volts)
    }
  }
```

```cpp
// Skip code block
```
I suggest you to use debounce to detect if a pin has been definitely pressed. You could also test using interrupts (hint: interrupts can happen multiple times even if you “only” pressed once). So either use Arduino’s example of debounce or take a look at the Bounce2 library found on their website.

Now that you can check whether or not either button A or B has been pressed, how do you keep track of the voltage to be output? You should use a counter variable that will be incremented when button A is pressed and decremented when button B is pressed. Here is a pseudocode:

```
if(ButtonA == pressed)
    counter++;  
else if(ButtonB == pressed)
    counter--; 
```

This counter variable can be used to control the output voltage. Remember that the argument for analogWrite() function on the Arduino Uno is the duty cycle which takes in values from 0 to 255. So analogWrite(0) ≡ 0 Volt and analogWrite(255) ≡ 5 Volt. So at the moment, you couldn’t just do analogWrite(counter) because you would need to press 255 times button A!

Instead, you would need a way to convert your counter to reflect the increment you need. You mentioned that it should require 10 presses to reach 5 Volt at the Arduino output, and the easiest way to accomplish this is to use Arduino’s Map function. The pseudocode would look like this:

```
Output = map(counter, 0, 10, 0, 255); 
analogWrite(Output);  
```

Now there is one last step! The map function will limit the output voltage from 0 to 5 Volt so everything will be safe; however, the counter is not limited to 0 to 10. This means that even if the counter is at 15 (because you pressed button A too many times) the voltage will be 5 Volt and you would need to press button B at least 6 times to see the voltage go down! So to limit the counter you could easily implement your own function:

```
if(counter >= 10) 
    counter = 10;  
else if (counter<= 0)
    counter = 0;  
```

You could also use Arduino’s constrain() function which more information can be found on their website.

Hopefully this points you in the right direction.
If all you have to do is to read those 2 buttons, you could just forgo the interrupts and simply poll in the main loop. The Bounce2 library does basically the same.

Also, about the PWM, why not an R-2R?

10V/0.5V + 1 => 21 steps < 32 => 5 bits => 6 resistors.

But it could be more precise, as you would not depend on the oscillator, rather on the precision of the resistors.

Q: Have I bricked my Arduino Uno? Problems with uploading to board

I can’t upload sketches to my Arduino Uno.

- Have I “bricked” it?
- What steps can I take to work out what is wrong?
- What can I do to fix it?

An example for stopping the upload:

```c
char* test = "This will stop the upload!!!";
```

See Google Code issue.
**Q: Arduino Uno not able to handle 2-dimensional array**

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q)

Now the code I’m writing in Arduino (Using Arduino) uses multiple 2-dimensional arrays. Now when I print some thing using the Serial Monitor it prints it correctly but when I declare and initialize the 2-dimentional array it won’t print it.

**Code:**

```
void setup()
{
  Serial.begin(9600);
  int image_width = 56;
  int image_height = 96;

  int image_result[image_width][image_height];
  for (int i=0; i<image_height; i++)
  {
    for (int j=0; j<image_width; j++)
    {
      image_result[j][i] = 5;
    }
  }
  Serial.print("code works");
}
```

Now in this case “code works” does not print but when I remove the array declaration and initialization code works is printed. What is the problem?

Do 2 dimensional arrays work differently in Arduino or is it a space issue?

**Tags:** arduino-uno (Prev Q) (Next Q), serial (Prev Q) (Next Q)

**User:** george-adams

---

**Answer** by brettam

2D arrays work fine on arduino, but you run out of space very quickly.
An uno has 2 kilobytes of ram; your array has 56*96 = 5376 2-byte elements.
I would guess that you are writing over some critical memory value with a 5 at some point causing the program to fail.

---

**Answer** by bot3663369

You are trying to use more memory than you have available: 96 * 56 * 2 = 10.752KBytes. Not much that the MCU can do when this happens.
If you look into the datasheet, you’ll see that your microcontroller (ATmega328p) only has 2KBytes of RAM.
I guess the question then, is to ask yourself if you really need that large an array. You may consider getting an external flash chip or an SDcard shield. If you only need it for read-only purposes (such as a lookup table), you can use some of your flash program memory (you have 32KBytes).
**Q: What happens when code is uploaded using the bootloader?**

When I upload a new sketch to my Arduino Uno using the Optiboot bootloader, what really happens?

- What is sent to the Arduino?
- How does it respond?
- What does “not in sync mean”?
- What is “in sync” anyway?

---

**Note: This is intended as a “reference question”**.

---

**User: nick-gammon**

**Answer by nick-gammon**

When you reset a Uno running the Optiboot loader, the bootloader first flashes pin 13 three times.

![Diagram showing pin 13 flashing three times](image)

Top line (gray) is sent **to** the Arduino, middle line (orange) is sent **from** the Arduino.

While that is happening, the program **avrdude** running on your computer is sending a query to the device:

```
STK_GET_SYNC / CRC_EOP (0x30/0x20)
```

The Arduino doesn’t notice the first “get sync” because it is busy flashing pin 13. Once it is done it notices the “get sync” (it would be buffered by the serial hardware) and replies:

```
STK_INSYNC / STK_OK (0x14/0x10)
```

It looks like avrdude got a bit impatient, and timed out, because it tries again with the “get sync” query. This time Optiboot responds immediately.
The rest of the uploading is described in the next image. Example produced uploading the stock “Blink” program.

The steps are:

- Query: Get Sync? Reply: In Sync.
- Set device parameters. The following device parameters are sent to the chip:

```plaintext
0x42 // STK_SETDEVICE
0x86 // device code
0x00 // revision
0x00 // proctype: “0” - Both Parallel/High-voltage and Serial mode
0x01 // parmode: “1” - Full parallel interface
0x01 // polling: “1” - Polling may be used
0x01 // selftimed: “1” - Self timed
0x01 // lockbytes: Number of Lock bytes.
0x03 // fusebytes: Number of Fuse bytes
0xFF // flashpollval1
0xFF // flashpollval2
0xFF // eeprompollval1
0xFF // eeprompollval2
0x00 // pagesizehigh
0x80 // pagesizelow
0x04 // eepromsizehigh
0x00 // eepromsizelow
0x00 // flashsize4
0x00 // flashsize3
0x00 // flashsize2
0x00 // flashsize1
0x20 // Sync_CRC_EOP
```

Optiboot ignores all those and replies with In Sync/OK. :)

- Set extended device parameters:
Optiboot ignores all those as well and replies with In Sync/OK.

- Enter program mode. Reply: In Sync/OK.
- Read signature. Optiboot replies with 0x1E 0x95 0x0F without actually reading the signature.
- Write fuses (four times). Optiboot does not write the fuse but just replies In Sync/OK.
- Load address (initially 0x0000). The address is in words (ie. a word is two bytes). This sets the address for where the next page of data will be written.
- Program page (up to 128 bytes are sent). Optiboot replies “In Sync” immediately. Then there is a pause of about 4 ms while it actually programs the page. Then it replies “OK”.
- Load address (now 0x0040). This is address 64 in decimal, ie. 128 bytes from the start of program memory.
- Another page is written. This sequence continues until all the pages are written.
- Load address (back to 0x0000). This is for verifying the write.
- Read page (up to 128 bytes are read). This is for verifying. Note that even if the verify fails, the bad data is still written to the chip.
- Leave programming mode.

What does “not in sync” mean?

As you can see from the above, every step through the programming sequence the Arduino is expected to reply with “In Sync” (0x14), possibly followed by some data, followed by “OK” (0x10).

If it is “not in sync” that means that avrdude did not get the “in sync” response. Possible reasons could be:

- Wrong baud rate used
- Wrong serial port selected in IDE
- Wrong board type selected in IDE
- No bootloader installed
- Wrong bootloader installed
- Board not configured to use the bootloader (in the fuses)
- Some device plugged into pins D0 and D1 on the Arduino, interfering with serial communications
The USB interface chip (ATmega16U2) not working properly  
Wrong clock for the board  
Wrong fuse settings on the Atmega328P (eg. “divide clock by 8”)  
Board/chip damaged  
Faulty USB cable (some USB cables provide power only, and are not for data, eg. cheap cables for USB fans)

---

What is “in sync”?

As mentioned above, the response “In sync” means that the Arduino (bootloader) is synchronised with the uploading program.

---

What protocol is being used?

The protocol is the STK500 protocol as documented by Atmel. See the references below.

---

References

- [Optiboot source code - GitHub](#)  
- [AVR061: STK500 Communication Protocol](#)  
- [AVR068: STK500 Communication Protocol version 2](#)  
- [Have I bricked my Arduino Uno? Problems with uploading to board](#)  
- [AVR Downloader/UploadeR - avrdude](#)

Note: STK500 Version 2 is not used in Optiboot, but it is included for information in case you are using boards like the Mega2560.

---

STK500 constants

```
/* STK500 constants list, from AVRDUDE */
#define STK_OK 0x10
#define STK_FAILED 0x11 // Not used
#define STK_UNKNOWN 0x12 // Not used
#define STK_NODEVICE 0x13 // Not used
#define STK_INSYNC 0x14 // '
#define STK_NOSYNC 0x15 // Not used
#define ADC_CHANNEL_ERROR 0x16 // Not used
#define ADC_MEASURE_OK 0x17 // Not used
#define PWM_CHANNEL_ERROR 0x18 // Not used
#define PWM_ADJUST_OK 0x19 // Not used
#define CRC_EOP 0x20 // 'SPACE'
#define STK_GET_SYNC 0x30 // '0'
#define STK_GET_SIGN_ON 0x31 // '1'
#define STK_SET_PARAMETER 0x40 // '@'
#define STK_GET_PARAMETER 0x41 // 'A'
#define STK_SET_DEVICE 0x42 // 'B'
#define STK_SET_DEVICE_EXT 0x45 // 'E'
#define STK_ENTER_PROGMODE 0x50 // 'P'
#define STK_LEAVE_PROGMODE 0x51 // 'Q'
```
Q: Reasons why it is not OK to connect a relay directly from an Arduino digital pin

After looking into various schematics on controlling a relay via an Arduino, I have noticed that, most of the time, transistors are used to switch the separate supply into the relay coil rather than directly supplying the relay with the 5-Volt output from the I/O pin of the Arduino. For example, I have a 5-volt DPDT relay and also a small dc motor. I want to drive both directly from my Arduino UNO (SMD clone) with an Atmel328 processor? Would it be advisable to proceed?

If not (most likely):

1. Can someone give a detailed explanation and might as well important cautions on current limits and stuff?
2. How can one control such components without risking the Arduino? What are some common means to accomplish this?
3. What other devices can generally harm an Arduino board (or any microcontroller unit) in a similar manner?

I’m just a beginner who wants to be extremely cautious. Thanks.

Answer by nick-gammon

The recommended output (source or sink) from an I/O pin is 20 mA. The absolute maximum is 40 mA. Your relay coil is likely to consume more than that, particularly when it initially energizes. This will damage your output pin. Then it will eventually fail.
Nothing seems to be amiss by far.

No, not yet. :)

How can one control such components without risking the Arduino? What are some common means to accomplish this?

Use transistors or MOSFETs.

What other devices can generally harm an Arduino board (or any microcontroller unit) in a similar manner?

Anything that exceeds the maximum voltage or current limits as documented in the datasheet. Coils (eg. in relays and motors) in particular are likely to have a high reverse voltage when they are turned off, which is why you need a snubber diode.

Would it be advisable to proceed?

It would be advisable to take heed of what I wrote above, and to read the many, many articles on the Web about how to drive motors and relays from an Arduino. You are not the first person to attempt this.

---

**Answer** by andy

Have a look at the simple circuit on this PDF from the Arduino Playground. It shows a single transistor to drive a small relay.

As Russell says in his answer, a ULN2803 or similar is a chip which will allow you to drive several small relays, which is neater than using several transistors, if that’s what you want.

(Also note the diode “D1” in the circuit I linked - you need this, it’s to protect the transistor from eventually being damaged by inductive spikes generated when the relay turns off. Some of the ULN-style chips have this diode built in, which is why you don’t always see it in schematics.)

---

**Answer** by user14178

To really protect your arduino it’s worth putting a photo-coupler on the pin and driving your circuit that way. Then no stray inductive charge or short can affect the arduino.

They are also called OptoIsolators or Optocouplers.

This is admittedly a cross-post from LED fade malfunction (random flash) but I can’t get an answer on the Arduino forum.

I was mucking around with some very basic code and I noticed that when repeatedly holding an LED at 0 brightness for 1 second and then fading in to full brightness, a small flash would occasionally happen at the beginning of each fade (seemingly random).

```
int led = 11;
int brightness = 0;

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
  if(brightness >= 256) //checks if brightness has passed 255, resets to 0
  {
    analogWrite(led, 255);
    brightness = 0;
  }
  analogWrite(led, brightness);
  if(brightness == 0)
  {
    delay(1000); //LED off for 1 second
  }
  brightness+=1; //increment brightness
  delay(20);
}
```

So, the thing that has me completely perplexed is that I can use a different piece of code (below) and the flash goes away!

```
int i = 0;
int led = 11;

void setup()
{
  pinMode(led,OUTPUT);
}

void loop()
{
  analogWrite(led, i);
  delay(6);
  if(i%256 == 0)
  {
    i = 0 ;
    delay(1000);
  }
  i++;
}
```

Has anyone got any clue as to why this would happen? Both programs have basically the same code, except for that i is reset to 0 in the first program whereas in the second, i keeps incrementing past 255 so that analogWrite ‘overflows.’ I think it must be a firmware, (or maybe a software?) problem.

There is a video on youtube of it happening here, Arduino - fading LED problem.

Tags: arduino uno (Prev Q) (Next Q), led (Prev Q) (Next Q), pwm (Prev Q) (Next Q)
According to the links you provide from the arduino.cc forum the question is more or less answered. If not that important, just avoid \texttt{analogWrite(led, 0)} and make it \texttt{analogWrite(led, 1)}

if you still want that \texttt{analogWrite(led, 0)}, I’ve tested your code with the advice and it seems to work OK when changing the register manually:

\begin{verbatim}
#include "wiring_private.h"

int led = 11;
int brightness = 0;

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
  if (brightness >= 256) //checks if brightness has passed 255, resets to 0
  {
    brightness = 0;
    sbi(TCCR2A, COM2A1);
    OCR2A = 0; // set pwm duty
  }

  analogWrite(led, brightness);
  if (brightness == 0)
  {
    delay(1000); //LED off for 1 second
  }

  brightness += 1; //increment brightness
  delay(20);
}
\end{verbatim}

\textbf{Edit: to explain what those “strange codes” are}

I’m not very expert in this field to explain how exactly this works but basically \texttt{sbi} is function defined in a macro by Atmel (?), it stands for “set bit in” and is used to change registers of ATmega chips. So basically what I did was change register defined by the macro \texttt{TCCR2A} (why? because it is the register that controls PWM in pin 11) and pass the bitmask \texttt{COM2A1} (this is a mode of compare defined in datasheet) and \texttt{OCR2A} is a register used to store the compare value that defines duty cycle. Imagine a counter that every time it receives a tick from a signal clock it compares the value of the counter with the value stored in \texttt{OCR2A} and it sets the pin high or low if that value as been passed or not (more or less like this, depending on the mode stored on \texttt{TCCR2A}).

But in fact I’ve not done some black magic. I just looked into the code of \texttt{analogWrite()} and that is the way they use to set a value of PWM in pin 11:

\begin{verbatim}
void analogWrite(uint8_t pin, int val)
{
  // We need to make sure the PWM output is enabled for those pins
  // that support it, as we turn it off when digitally reading or
\end{verbatim}
// writing with them. Also, make sure the pin is in output mode
// for consistency with Wiring, which doesn't require a pinMode
// call for the analog output pins.
pinMode(pin, OUTPUT);
if (val == 0)
{
    digitalWrite(pin, LOW);
}
else if (val == 255)
{
    digitalWrite(pin, HIGH);
}
else
{
    switch(digitalPinToTimer(pin))
    {
    ...
    
    #if defined(TCCR2A) && defined(COM2A1)
    case TIMER2A:
      // connect pwm to pin on timer 2, channel A
      sbi(TCCR2A, COM2A1); OCR2A = val; // set pwm duty
      break;
    #endif...
    }

So basically I just used that info to set the register to zero, as suggested from the
arduino.cc forum link that you have.

If you want to know more about PWM in Arduino, this site has a lot of info and
explains a lot of the modes of registers also.

Tags: arduino-uno (Prev Q) (Next Q), led (Prev Q) (Next Q), pwm (Prev Q) (Next Q)

Q: Arduino DC motor causes disturbance. What can cause it?

Tags: arduino-uno (Prev Q) (Next Q), c (Prev Q) (Next Q), motor (Prev Q) (Next Q), lcd (Next Q)

I tried to fix this problem the whole weekend but, after no success I decided to post it here.
I would really appreciate any help.

The problem

The Wi-Fi module activates the DC motor and lets it run for 3 seconds, but when the DC
motor stops after that 3 seconds, the Wi-Fi module and display don’t respond anymore. It
only works once. (After I push the reset button the same thing happens.)

Question

What causes this disturbance? Any advice on my circuit? (I included a Fritzing
Diagram because I am bad at drawing schematics)
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial ESP8266(9, 8); // RX = 8 en TX = 9

#define DEBUG true

boolean FAIL_8266 = false;
int LED = 13; // led op 13
int secondeAan = 3000;

String my_AP_SSID = "myID";
String my_AP_Pass = "myPass";

void setup()
{
    // stel led in
    pinMode(LED, OUTPUT);

    // lcd scherm
    lcd.begin(16, 2);
    lcd.clear();
    lcd.setCursor(0, 1);

    do {
        ESP8266.begin(115200); // start communicatie met esp8266

        // Wait Serial Monitor to start
        while (!Serial);

        lcd.clear();
        lcd.print("--- Start ---");
        ESP8266.print("AT\r\n");
        delay(500);
    }
if (ESP8266.find("OK"))
{
  FAIL_8266 = false;
  sendData("AT+RST\r\n", 4000, DEBUG);
  sendData("AT+CWMODE=3\r\n", 2000, DEBUG);
  sendData("AT+CJAP=" + my_AP_SSID + "," + my_AP_Pass + ",9,4\r\n", 2000, DEBUG);
  sendData("AT+CIFSR\r\n", 2000, DEBUG);
  sendData("AT+CIPMUX=1\r\n", 2000, DEBUG);
  sendData("AT+CIPSERVER=1,80\r\n", 2000, DEBUG);
}
else {
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("Module have no response.");
  delay(500);
  FAIL_8266 = true;
}
} while (FAIL_8266);

void loop()
{
  // Get the number of bytes (characters) available for reading from the serial port
  if (ESP8266.find("+IPD,"))
  {
    lcd.clear();
    lcd.print("Nieuwe connectie");
    //—sluit connectie—
    motorToggle();
    sendData("AT+CIPCLOSE=0\r\n", 100, DEBUG);
    sendData("AT+CIPCLOSE=1\r\n", 100, DEBUG);
    sendData("AT+CIPCLOSE=2\r\n", 100, DEBUG);
  }
}

void motorToggle ()
{
  digitalWrite(LED, HIGH);
  lcd.clear();
  lcd.print("eten gegeven");
  delay(secondeAan);
  digitalWrite(LED, LOW);
  delay(200);
}

String sendData(String command, const int timeout, boolean debug)
{
  String response = "";
  ESP8266.print(command); // send the read character to the esp8266
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (ESP8266.available())
    {
      char c = ESP8266.read();
      response += c;
    }
  }
  if (debug)
  {
    lcd.clear();
    lcd.print(response);
  }
  return response;
}
Thanks for the help everyone. I fixed it by using Capacitors. They suppress the noise that the dc motor produces. I found my information here.
There are a couple of things that could be causing your problems:

- The current draw of the motor, especially at start up. The motor may be drawing enough power to pull down the voltage at the processor causing it to crash, or
- You may be seeing transients as the motor starts and/or stops that are causing the processor to crash.

One way you could detect problems like this is to use the on-board LED as a heart beat or run indicator. You could either blink it in loop() or do something like turn it off just before starting or stopping the motor and then back on afterwards. If the blink stops or if the LED goes out then you know that the processor has stopped running.

As a general rule, except with the tiniest of motors, I’d suggest using the Arduino to control a transistor that controls the motor from its own power supply line (it could be the same supply as the one the feeds the Arduino, just don’t run the motor power through the Arduino). That and the suggestions in the comments will go a long way towards isolating the Arduino from the motor and power disturbances it will cause.

**Tags:** arduino-uno (Prev Q) (Next Q), c (Prev Q) (Next Q), motor (Prev Q) (Next Q), lcd (Next Q)

---

**Q: I2C LCD displaying weird characters**

I have connected an LCD with an I2C backpack to my Arduino Uno but it prints the wrong characters. The weird thing is that it worked fine for a while and when I updated the code (but didn’t change anything to do with the LCD) it started displaying the wrong characters. This happened earlier today too. That time, reinstalling the LiquidCrystal library worked, but it doesn’t now.

Does anyone know what might be the problem so I can fix it permanently?

The LCD’s SDA and SCL are connected to A4 and A5 respectively and I’m running this code:

```c
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Hello, world!");
}

void loop()
```

and this is what’s displayed on the screen:

If you look up the ASCII codes for what you sent vs what was displayed, and also look at “Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)” in an HD44780U spec sheet, you will find hexadecimal values like the following.

- e 65, V 56
- l 6C, ニ C6
- o 6F, θ F2
- o 6F, π F7
- r 72, & 26
- d 64, B 24
- and so forth.

It appears that the high nibble of what you sent is being used as the low nibble of what is displayed, and the low nibble is being used as the high, but with some bits sometimes
going wrong. Note, a space is 0x20, and the reverse of that is 0x02, which would access a character from the character-generator RAM, which will contain random bits and produce a random blob as in your picture.

If your mis-specification of display size (20, 4 vs 16, 2) isn’t responsible for the problem, then maybe there is a hardware problem with the backpack you’ve soldered on to the LCD. Look at it under a bright light, using a binocular parts-examination microscope if you’ve got one, and see if there are loose solder particles, extra solder flux, or loose solder connections on that board. If you have an oscilloscope, check if your I2C signals are clean and have full swing.

**Answer** by tisteandii

From what I can see, it seems, in this line:

```cpp
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

wrong parameters have been passed to the constructor. The last two arguments are supposed to indicate the number of columns and rows on the screen. Your screen has 16 columns and 2 rows, whereas you passed 20 columns and 4 rows to the constructor. This could be the problem.

**Tags:** arduino-uno (Prev Q) (Next Q), lcd (Prev Q) (Next Q), i2c (Prev Q) (Next Q)

---

**Q: Precision voltmeter using Arduino**

**Tags:** arduino-uno (Prev Q) (Next Q), voltage-level (Prev Q) (Next Q), arduino-nano (Prev Q) (Next Q), analogread (Prev Q) (Next Q)

I am newbie, I need to measure precise voltage in range 0.00-0.80V, two digits precision.

The voltage will not be greater than 1V-1.2V.

I wonder if I need divider or not, I am sure that voltage won’t rise above 1.2V.

Can I connect the point to measure directly to analog input, or I still need to use divider? Please explain why.

**Tags:** arduino-uno (Prev Q) (Next Q), voltage-level (Prev Q) (Next Q), arduino-nano (Prev Q) (Next Q), analogread (Prev Q) (Next Q)

**User:** solderingiron

**Answer** by majenko

A divider is used to (as the name suggests) reduce a higher voltage to a lower voltage. Using a divider on a small voltage will only make it smaller and harder to measure.

To get the most out of your measuring you need to have the *reference* voltage as close to the highest voltage you want to measure as possible. The highest voltage that can be applied is not dependent on the reference voltage, but the supply voltage, and as 1.2V is way less than 5V (or 3.3V on a 3.3V board) you have no worries there.
So you need to change the reference voltage - the voltage that the ADC uses as its upper *measurement* voltage. That’s simple enough, since many of the Arduino boards include an internal 1.1V analog reference voltage generator that you can switch to (see: `analogReference()`).

So when you have switched to the 1.1V analog reference the ADC is capable of measuring 0-1.1V in 1024 steps. That’s 1.1/1023=0.00107V per bit of precision.

The formula is:

\[ V = \frac{ADC}{1023} \times V_{ref} \]

So a reading of 1 would give you 0.00107V - a reading of 2 would be 0.00215V, etc. That’s plenty of resolution to meet your 2 decimal places requirement.

*One thing to note* though is that the internal 1.1V reference isn’t that stable or accurate (or not even there on some chips). If you need better stability and accuracy then it could pay to use an external voltage reference chip. These are like a linear voltage regulator but are far more accurate and stable (and can’t provide lots of current). Pick one that is greater than your upper measurement voltage but as close to it as you can get (so just above 0.8V) and feed the output into the external `ARef` pin and switch the analog reference to the **EXTERNAL** mode.

You *could* use a voltage divider on the external ARef pin to provide a low reference voltage, but the stability and accuracy will be no better (and probably worse) than the internal 1.1V voltage reference, so there really is no point.

---

**Q: What’s the difference between setting a variable outside a function and setting a variable inside a function?**

**Tags:** arduino-uno (Prev Q) (Next Q), voltage-level (Prev Q) (Next Q), arduino-nano (Prev Q) (Next Q), analogread (Prev Q) (Next Q)

I’m programming C++ for arduino and today I have seen an example like this

```cpp
void loop() {
    byte variable = 0; // << This variable is the question.
    //more stuff here
}
```

Is the same as this?:

```cpp
byte variable = 0;
void loop() {
    //stuff here
}
```

If the answer is “no”, my question is “why?”.

Thanks for your time!

**Tags:** arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q)
The difference is not only about where you set the variable, but about where you defined it as well.

Defined outside the function, it will have global scope and lifetime, meaning it is visible everywhere and it exists for the entire duration of the program. If you then place an initializer inside the function, whatever value it had just before function-entry will be replaced by the initializer value.

If you define it inside the function (and without a “static” specifier) it only exists while that function is executing, and is only visible to code inside the function. Because it only comes into existence when the function is entered, you must initialize it within the function, somehow.

If you define it inside the function, but with a static specifier,

```
static byte variable = 0;
```

then it will exist for the entire execution of the program but will only be visible to code within the function. In addition, an initializer on its definition (as opposed to other assignments to the variable) will only be executed once, just before the program begins executing, and will not be executed within function.

Using local and static (as opposed to global) variables is considered good program-variable hygiene because it hides those variables (to a greater or lesser degree) from code that needn’t and shouldn’t be able see or alter them.

No, it’s not the same. Instructions within the loop() repeat every time the program gets to the end. If you set your variable at the start of the loop, it will automatically get reset to the default when you might not want it to be reset. The author of the example wants to reset to default every time the loop completes.

Tags: arduino-uno (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

Q: Change Servo Speed With RedBearLabs BLE Shield

Tags: arduino-uno (Prev Q) (Next Q)

I recently got the RedBearLabs BLE shield to use it to control two high speed continuous rotation servos. I have the BLE example sketch loaded and the ios controller app works. The problem is that the servos are too fast to control and I want to slow them down. I cannot use the app to slow them down. I was wondering how I could modify the example sketch or servo library to change the ax speed or something. Thanks in advance.

BLE Shield Homepage: http://redbearlab.com/bleshield/
BLE Libraries and Example Sketch: [https://github.com/RedBearLab/nRF8001](https://github.com/RedBearLab/nRF8001)

Tags: [arduino-uno](https://www.arduino.cc) (Prev Q) (Next Q)

User: snapdraggen

---

**Answer** by jwpat7

The bi-directional Parallax [High Speed Continuous Rotation Servo](https://www.parallax.com/) responds to “simple pulse-width modulation”, according to its web page. That is, pulse widths in the range from about 500 μs to 2500 μs will set the speed and direction, with some value near 1500 μs corresponding to zero RPM.

The code in the [BLEControllerSketch](https://www.parallax.com/) uses the following bits of code, among others, to initialize servo-control data structures and to control them:

```cpp
#include <Servo.h>
...
byte pin_servo[TOTAL_PINS];
Servo servos[MAX_SERVOS];
...
case 'O': // set Servo
{  
    byte pin = ble_read();
    byte value = ble_read();

    if (IS_PIN_SERVO(pin))
        servos[PIN_TO_SERVO(pin)].write(value);
    pin_servo[pin] = value;
    reportPinServoData(pin);
}
break;
```

Thus, you will need to send an `O` message from your iOS app, together with servo pin number and control value, to set a servo’s speed.

As noted on the [ServoWrite](https://www.parallax.com/) page at arduino.cc, the parameter to Servo’s `.write()` method is a number:

```
void Servo::write(int value)
{
    if(value < MIN_PULSE_WIDTH)
    {  // treat values less than 544 as angles in degrees (valid values in microseconds are handled as
        if(value < 0) value = 0;
        if(value > 180) value = 180;
        value = map(value, 0, 180, SERVO_MIN(), SERVO_MAX());
    }
    this->writeMicroseconds(value);
}
```

`SERVO_MIN()` and `SERVO_MAX()` are macros that access min and max fields in a `Servo` object and convert them to microsecond values. Because the code in BLEControllerSketch.ino
lets those fields default, the range evidently will be from 544 to 2400 μs (see the values
given in Servo.h).

Note that you can get better resolution – hence, better speed control – by sending values in
that range of microseconds to your servos, instead of trying to control it using values in
degrees. The .write() method will interpret such numbers as microseconds instead of as
degrees. Servo.cpp appears to use half-microsecond timer ticks; and each degree
corresponds to about 10 μs of pulse width; so resolution using microsecond values should
be about 10 times better than using degrees.

In short, it appears that you should do some experimenting with 0 messages that send
values around 1500, to find the values that correspond to slowest forward and backward
speeds.

---

**Q: Sending colours to vga monitor**

Tags: arduino-uno (Prev Q) (Next Q)

---

I’m looking for a simple way to send a signal from my arduino to a vga monitor.

What would be the easiest method, and what components would be needed?

I don’t need to display an image, just a solid colour.

Tags: arduino-uno (Prev Q)

User: terry

---

**Answer** by giordano-bruno

A quick Google search for “Arduino VGA” will give you a lot of information. There are a
few variations on both circuits and programming, which also vary in resolution and color
depth.

I was searching for this a few days ago, and these are my favorite (so far):

- [https://forum.arduino.cc/index.php?topic=320238.0](https://forum.arduino.cc/index.php?topic=320238.0) (read the whole discussion, pretty nice results)

If using a TV is also a plausible option, check the Arduino TV out library. It can be
installed directly from the Arduino IDE and has a good demo.

---

**Answer** by chris-stratton

Not needing to display an actual image simplifies things substantially, as an Arduino lacks
the memory and (except in a crude sense) the bandwidth to do so.
However, you cannot simply apply a steady analog voltage to the R, G, and B lines. Not only do you have to drive horizontal and vertical sync signals, you have to blank the RGB signals when not on the active portion of the screen, otherwise the monitor will assume their steady voltage means “black” and your colors will last only as a brief flash when your device is first connected or enabled.

Generating a large rectangular field of color from an Arduino is likely to be rather challenging, but probably not impossible. You may be able to use hardware PWM channels for the horizontal and a “color enable”, and tightly coded software counters for the vertical aspect. You can then use the “color enable” to gate a network of potentially variable resistors to establish the single color of particular interest.

Tags: arduino-uno (Prev Q)
Use this tag for questions regarding programming (writing code) for Arduino. This can include problems with code you’ve written for Arduino, or questions about how/why Arduino code acts in certain ways.

Where a question relates to a particular language, you can also use language-specific tags, such as c++.

Programming questions which are not specific to Arduino (e.g. general questions about a programming language) should be asked on Stack Overflow instead.

For questions regarding uploading sketches to Arduino devices, please use the uploading tag.

Note: With this tag, syntax highlighting will be enabled. See the syntax highlighting help section for more information.
Questions

Q: Programming an Arduino using Python, rather than C/C++

Tags: programming (Prev Q) (Next Q)

I am not very skilled with the C Language and I was wondering if there is a way in which python could be used to program an Arduino. This would most likely require a different IDE in order to be able to debug the scripts them self.

Tags: programming (Prev Q) (Next Q)

User: jvarhol

Answer by p peter-r.-bloomfield

It’s going to be extremely difficult to get any kind of Python script running directly on the Arduino. The reason is that it’s an interpreted language, so you would need the interpreter on-board in addition to the plain text script. There’s probably not going to be enough memory for all of that.

Your best bet would probably be finding a way to compile a Python script to native machine code (which is how C/C++ works). I believe there are projects around to do something like that for other platforms, but (as far as I know) none which does it successfully for Arduino yet.

You might find some more useful information on this question at Stack Overflow: Is there a way to “compile” Python code onto an Arduino (Uno).

Answer by asheesh

Yes, it is (somewhat) possible to program the Arduino using Python. One such project on Github is the Python Arduino Prototyping API v2. It provides very basic functionality such as digital I/O and analog I/O.

This can be used for very simple projects.

*This project is a bit of a hack at “programming” the board using the serial connection. It passes the commands over the serial connection to a sketch running on the board which then “executes” the Python command.

Answer by matthew-g.

There is a project which brings a Python virtual machine to micro-controllers, including the Arduino Mega.

Here’s a quote from the Arduinio Mega README, which gives a feel for what this could be like (though, I’ve not tested this!):

The following is an example session using ipm::

```
ipm> import avr, sys
```
Q: Programming options other than C++

Can I program my Arduino using any other programming languages than C++? If so, which programming languages. What software would I need to compile and load code to the Arduino.

Answer by hardik-thaker

Some commonly known ones:

- Ada
- Basic
- Forth
- Java
- Pascal
- Python
- C (mac) and C (Windows)

In theory you should be able to extend avr-gcc to support other languages, though this is no small undertaking. I should also caveat that support for languages other than C typically comes with some fairly weighty restrictions on language components — for instance, even with C++ it’s discouraged to be instantiating new objects as malloc and free are extremely expensive in both memory space and cycles.

Besides using Google, the following references are applicable Click Here.

Answer by matthew-g.

In theory, any compiler which outputs AVR assembly could be used (modulo limitations on code size, ram usage etc).

AVR backends exist for both LLVM and GCC — so in theory, with some effort, most languages which are supported by LLVM and GCC’s front-ends could be compiled to AVR. A big question will be library support, etc.

Never the less, here’s a couple of odd choices for the spirit of the question, rather than its letter [1].
• AMForth, a forth dialect can be flashed onto an Arduino.
• AVR-Ada: Ada for AVR micro-controllers.
• BASIC: Bring back the Goto!

Oh, and you could write raw AVR Assembly!

[1]- I Haven’t tried these though!

---

**Tags:** programming (Prev Q) (Next Q)

---

**Q: Code for one-time execution in Arduino**

**Tags:** programming (Prev Q) (Next Q)

**Question:** Where do I put code if I want it to execute only once per Arduino startup?

**Background:** I tend to use my Arduino to perform tasks that should be executed once per power cycle. (For example, my robot should only drive forward *once* every time I turn it on—more than that, and it might fall off the table.)

The standard Arduino IDE “template” consists of a `loop()` and a `setup()` function. The `setup` code is run once per power cycle, and the `loop` is re-started every time it finishes.

I see two approaches to code “1 time tasks:”

**Method 1: “Loop once”**

```c
void setup() {
    //do setup stuff
}

void loop() {
    //do task
    while(1); //Repeat forever, preventing function from re-starting
}
```

**Method 2: “Setup only”**

```c
void setup() {
    //do setup stuff
    //do task
}

void loop() {
    //do nothing
}
```

**More detailed form of question:** Is one of these methods “better” from a hardware/software sense? (e.g. is there anything “special” about the `setup` function, like it can only run for so much time?)

And, if they are the same from a hardware/software perspective, is there a “best practice” or standard way of writing code for these situations in the Arduino community?
I’m kind of confused by your question. You ask where you want to put once-per-startup setup functions, and then discuss the setup function. That’s what the setup function is for. As such, one-time setup functionality goes in the setup function.

FWIW, if you look in the file that calls the setup and loop functions:

```cpp
#include <Arduino.h>

int main(void)
{
    init();
    #if defined(USBCON)
        USBDevice.attach();
    #endif
    setup();
    for (; ; ) {
        loop();
        if (serialEventRun) serialEventRun();
    }
    return 0;
}
```

For all intents and purposes, the two options are completely identical. Either way, you get a empty busy-wait loop. Frankly, I’d expect the two different options to probably emit the same machine code anyways, so the whole thing is a non-issue.

Note:
if (serialEventRun) serialEventRun(); appears to be a facility to allow you to attach a function that is called upon reception of serial data, but if you do not define a function void serialEvent(){} in your code, it will compile out completely and not be present in the produced machine code.

I usually go with Method 2, but end up doing this:

```cpp
void setup() {
    //do setup stuff
    init(); //Do start-up initialization steps
}

void init() {
    // do tasks on startup
}

void loop() {
    //do looping code
}
```

With the above setup it allows my code to be even more organized.
I would strongly prefer Method 2. If you ever plan to add code to handle input, output, etc, it’s easy with Method 2 — just fill in loop(), but requires reworking/refactoring in Method 1.

**Q: Why can’t I declare a class in another tab in Arduino IDE?**

I wanted to move some of my code out into a second tab in the Arduino IDE, to keep things better organised. At first, I only tried moving a function, and it seemed to work fine. I could call the function from the setup() function in my main tab, and there were no problems compiling or uploading.

However, I tried putting a whole class into the second tab, and suddenly it didn’t work any more. For example:

**Tab 1:**

```cpp
TestClass obj;
void setup()
{
    obj.init();
}
void loop()
{
    //...
}
```

**Tab 2:**

```cpp
class TestClass
{
    public:
        void init()
        {
            //...
        }
};
```

When I tried to compile this, it gave me the following errors:

```plaintext
tab1:1: error: 'TestClass' does not name a type
tab1.ino: In function 'void setup()':
tab1:5: error: 'obj' was not declared in this scope
```

Why does it recognise a function in another tab, but not a class? Is there a way to make it work in the Arduino IDE, or do I need to use an alternative like Eclipse?
It is not possible to declare and use classes declared in one .pde file in another .pde file from within the Arduino IDE.

One workaround is to make the second file into a C++ source file (.cpp) and then add a #include "<filename>" directive in the beginning of the first file.

This code compiles correctly:

Tab 1:

```cpp
#include "test.cpp"
TestClass obj;
void setup()
{
    obj.init();
}
void loop()
{
    //...
}
```

test.cpp :

```cpp
class TestClass
{
    public:
        void init()
        {
            //...
        }
};
```

Answer by jamolnng

The way the Arduino IDE works is that it compiles your code (the code you write in the IDE) as the “main” code. Then it pulls code from all of the libraries you have imported and compiles that along with the main code. To do what you are suggesting would require you to create a library for Arduino.

Here is some more information on Arduino libraries:


And here is some on how to create a library:

http://arduino.cc/en/Hacking/LibraryTutorial //this one is the easiest to understand imo
http://playground.arduino.cc/Code/Library
http://www.divilabs.com/2013/03/write-your-own-arduino-library.html#

Here is an example library I wrote
https://github.com/jamolnng/Arduino/tree/master/libraries/ShiftRegister as you can see there is no problem with calling Arduino functions from the .cpp file (I know, I’ve tested the library)
**Q: What will happen when I use an invalid pin number?**

**Tags:** programming (Prev Q) (Next Q)

This question is similar to the one above, however this is an alternate situation:

```c
int pin = 999;
pinMode(pin, OUTPUT);
digitalWrite(pin, HIGH);
```

What would happen in this instance? The compiler might catch it but if you used a random number would the IDE catch it?

**Tags:** programming (Prev Q) (Next Q)

**User:** anonymous-penguin

---

**Answer** by asheeshr

The compiler will not detect any error and the code will compile and execute. Hence, to see what happens we need to explore the behind-the-scenes magic. *For a summary, skip to end.*

The second line in your code is where the magic will happen and thats where we need to focus.

```c
pinMode(pin, OUTPUT);
```

The portion of pinMode relevant to this discussion is:

```
void pinMode(uint8_t pin, uint8_t mode)
{
    uint8_t bit = digitalPinToBitMask(pin); //The first instance where pin is used
    uint8_t port = digitalPinToPort(pin);
    if (port == NOT_A_PIN) return;
    //Do something
}
```

(The complete implementation can be found in wiring_digital.c)

So, here, digitalPinToBitMask seems to be using pin to compute an intermediate bit. Exploring further, digitalPinToBitMask is a macro defined in Arduino.h whose definition is this one-liner:

```c
#define digitalPinToBitMask(P) ( pgm_read_byte( digital_pin_to_bit_mask_PGM + (P) ) )
```

This weird looking one liner does a very simple task. It indexes the Pth element in the array digital_pin_to_bit_mask_PGM and returns it. This array digital_pin_to_bit_mask_PGM is defined in pins_arduino.h or the pin map for the
specific board being used.

```c
const uint8_t PROGMEM digital_pin_to_bit_mask_PGM[] = {
  _BV(0), /* 0, port D */
  _BV(1),
  _BV(2),
  _BV(3),
  _BV(4),
  _BV(5),
  _BV(6),
  _BV(7),
...};
```

This array has 20 elements in total, so we are out of luck. 999 will index a memory location in the flash memory outside of this array, thereby leading to unpredictable behavior. *Or will it?*

We still have another line of defense against runtime anarchy. Its the next line of the function pinMode:

```c
uint8_t port = digitalPinToPort(pin);
```

digitalPinToPort takes us along a similar path. It is defined as a macro along with digitalPinToBitMask. Its definition is:

```c
#define digitalPinToPort(P) ( pgm_read_byte( digital_pin_to_port_PGM + (P) ) )
```

Now, we index the P\textsuperscript{th} element of digital_pin_to_port_PGM which is an array defined in the pin map:

```c
const uint8_t PROGMEM digital_pin_to_port_PGM[] = {
  PD, /* 0 */
  PD,
  .... PC,
  PC,
};
```

This array contains 20 elements, so 999 is again out of range. Again, this command reads and returns a value from flash memory of whose value we cannot be certain. This will again lead to unpredictable behavior from here on.

There is still one last line of defense. That is the if check in pinMode on the return value of digitalPinToPort:

```c
if (port == NOT_A_PIN) return;
```

NOT_A_PIN is defined as 0 in Arduino.h. So, if the returned byte from digitalPinToPort happens to be zero, then pinMode will silently fail and return.

In any case, pinMode cannot save us from anarchy. 999 is destined to result in doom.

**TL;DR, the code will execute and the result of this will be unpredictable.** Most likely, no pin will be set to OUTPUT, and digitalWrite will fail. If you happen to have exceptionally bad luck, then a random pin may get set to OUTPUT, and digitalWrite may set it to HIGH.

---

*Answer by thedoctor*
In the standard libraries, there are macros designed for converting pins to ports, which are used in assembly. Here they are for the Uno from Arduino 1.0.5:

```
#define digitalPinToPCICR(p) (((p) >= 0 && (p) <= 21) ? (&PCICR) : ((uint8_t *)(0))
#define digitalPinToPCICRbit(p) (((p) <= 7) ? 2 : (((p) <= 13) ? 0 : 1))
#define digitalPinToPCMSK(p) (((p) <= 7) ? (&PCMSK2) : (((p) <= 13) ? (&PCMSK0) : (((p) <= 21) ? (&PCMSK1) : ((uint8_t *)(0)))))
#define digitalPinToPCMSKbit(p) (((p) <= 7) ? (p) : (((p) <= 13) ? ((p) - 8) : ((p) - 14)))
```

There are more, but I won’t show them here.

I believe your program would subtract 14 from 999, which would still be too big for the program. It would then try to point to the 985th element of the digital_pn_to_bit_mask_PGM array, which only contains 20 elements. This would most likely end up screwing the Arduino by pointing to a random spot in progmem.

Tags: [programming](#) (Prev Q) (Next Q)

**Q: Why is int only 2 bytes?**

Tags: [programming](#) (Prev Q) (Next Q), [c++](#) (Prev Q) (Next Q)

When using C/C++ on other platforms, the `int` type is typically 4 bytes (or potentially more). However, on Arduino, it’s only 2 bytes.

Why is it different? Does it affect performance if I always use the 4 byte `long` instead?

Tags: [programming](#) (Prev Q) (Next Q), [c++](#) (Prev Q) (Next Q)

User: [peter-r.-bloomfield](#)

**Answer** by [cybergibbons](#)

The ATmega328 used in a lot of Arduinos is a 8-bit microcontroller. This means that registers are 8-bits, the data bus is 8-bits, the ports are 8-bits. There are some minimal 16-bit aspects to the system (e.g. one of the timers), but nearly everything is 8-bits.

Therefore, most operations handle 8-bits at a time. Working on anything except 8-bits (i.e. 16-bit or 32-bit integers and floating point numbers) requires what could essentially be described as software emulation, where the compiler uses multiple instructions to work on these larger variables.

8-bits is obviously adequate to address a 8-bit port. It’s also enough to deal with many loop counters, return values, and ASCII characters. It isn’t really enough though when dealing with numbers. A signed 8-bit int (int8_t) can only represent -128 -> +127. Unsigned (uint8_t) can only represent 0 -> 255.

8-bit integers are quite limiting. C/C++ `int` must represent at least -32,678 -> +32,767 so maps to int16_t - the smallest size that will do so. This gives a good balance of range and efficiency. This is especially important when beginners are learning - overflow is not really something that non-programmers understand.

There is a performance impact of doing this however, because most 16-bit operations take at least twice as long as an 8-bit operation, and use twice as many registers. This may or
may not make a difference to you.
Many of us switch to the native types such as int8_t and uint8_t as it gives you far more control.

Answer by jfpoilpret

One important fact about C and C++ languages is that their respective standards do not define the size (in bytes) of integral and floating point number types.
They just define minimal ranges and relation between these ranges, e.g.

\[
\text{range(short)} \leq \text{range(int)} < \text{range(long)}
\]

So the size of e.g. an int will typically depend on:

- the target platform (processor)
- the compiler itself

Tags: programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

Q: Would an infinite loop inside loop() perform faster?

Tags: programming (Prev Q) (Next Q)

When you’re writing a typical sketch, you usually rely on loop() being called repeatedly for as long as the Arduino is running. Moving in and out of the loop() function must introduce a small overhead though.

To avoid that, you could presumably create your own infinite loop, like this:

```c
void loop()
{
    while (true)
    {
        // do stuff...
    }
}
```

Is that a viable way to improve performance? Will it cause other problems if loop() never returns?

Tags: programming (Prev Q) (Next Q)

User: peter-r.-bloomfield

Answer by cybergibbons

The part of the code on an ATmega core that does setup() and loop() is at follows:

```c
#include <Arduino.h>
int main(void)
{
    init();

```
Pretty simple, but there is the overhead of the `serialEventRun();` in there.

Let’s compare two simple sketches:

```c
void setup()
{
}

volatile uint8_t x;

void loop()
{
    x = 1;
}
```

and

```c
void setup()
{
}

volatile uint8_t x;

void loop()
{
    while(true)
    {
        x = 1;
    }
}
```

The x and volatile is just to ensure it isn’t optimised out.

In the ASM produced, you get different results:
You can see the while(true) just performs a rjmp (relative jump) back a few instructions, whereas loop() performs a subtraction, comparison and call. This is 4 instructions vs 1 instruction.

To generate ASM as above, you need to use a tool called avr-objdump. This is included with avr-gcc. Location varies depending on OS so it is easiest to search for it by name.

avr-objdump can operate on .hex files, but these are missing the original source and comments. If you have just built code, you will have a .elf file that does contain this data. Again, the location of these files varies by OS - the easiest way to locate them is to turn on verbose compilation in preferences and see where the output files are being stored.

Run the command as follows:

```
avr-objdump -S output.elf > asm.txt
```

And examine the output in a text editor.

---

**Tags:** programming (Prev Q) (Next Q)

---

**Q: Why do sketches take up so much space and memory?**

**Tags:** programming (Prev Q) (Next Q), compile (Prev Q) (Next Q)

When I compile this sketch for the Yún:

```c
int led = 7;
```
 void setup() {
  pinMode(led, OUTPUT);
}

 void loop() {
  digitalWrite(led, HIGH);
}

I get:

Sketch uses 5,098 bytes (17%) of program storage space.
Maximum is 28,672 bytes. Global variables use 153 bytes (5%) of dynamic memory, leaving 2,407 bytes for local variables. Maximum is 2,560 bytes.

Even when I compile the BareMinimum sketch:

 void setup() {
  // setup
 }
 void loop() {
  // loop
 }

I get:

Sketch uses 4,548 bytes (15%) of program storage space.
Maximum is 28,672 bytes. Global variables use 151 bytes (5%) of dynamic memory, leaving 2,409 bytes for local variables. Maximum is 2,560 bytes.

Why does a bare minimum sketch take up 15% of the program storage space allocated? And why does a very simple sketch take up 17% of the program storage space? According to the Arduino website:

It’s easy to use it all up by having lots of strings in your program. For example, a declaration like: char message[] = "I support the Cape Wind project."); puts 33 bytes into SRAM (each character takes a byte, plus the ‘\0’ terminator).

However, there aren’t any strings declared in either of these sketches.

It seems as if they might import or use other libraries/classes that I don’t specify. Maybe it imports a system default library? Or is it something else?

Tags: programming (Prev Q) (Next Q), compile (Prev Q) (Next Q)

User: hichris123

Answer by mpflaga

The YUN is a combo. Part Arduino and Part OpenWRT(Linux). Your question is in reference to the Arduino. Where this is actually a ATmega32u4 similar to a Leonardo and not an UNO(ATmega328p). The 32u4(Leo) communicates via Virtual Serial Ports over the USB (short answer: this needs to be supported), where the UNO has a real Serial Port (aka UART). Below are builds statistics of the different boards types for the AVR.
processors.

Note on the UNO there is an external chip that converts USB to the Serial port’s DTR pin which toggles the ATmega328’s reset pin when connected causing a reboot to the bootloader. In contrast the Leo/Yun’s USB to Serial is in implemented in the firmware of the 32u4. Hence in order to remotely reboot the Leo or YUN’s 32u4 chip the firmware loaded must always support the USB client side driver. Which consumes approximately 4K.

If the USB was NOT needed and no other library resources were called as in the case of BareMinimum.ino on an UNO, only approximately 466 bytes are needed for the core Arduino Library.

### compile stats of BareMinimum.ino on a UNO(ATmega328p)

| Sketch uses 466 bytes (1%) of program storage space. Maximum is 32,256 bytes. |
| Global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes for local variables. Maximum |

### compile stats of BareMinimum.ino on a Leonardo(ATmega32u4)

| Sketch uses 4,554 bytes (15%) of program storage space. Maximum is 28,672 bytes. |
| Global variables use 151 bytes (5%) of dynamic memory, leaving 2,409 bytes for local variables. Maximum |

### compile stats of BareMinimum.ino on a Yun(ATmega32u4)

| Sketch uses 4,548 bytes (15%) of program storage space. Maximum is 28,672 bytes. |
| Global variables use 151 bytes (5%) of dynamic memory, leaving 2,409 bytes for local variables. Maximum |

---

**Answer** by **jippie**

Arduino compiles in a lot of standard libraries, interrupts, … etc. For example the pinMode and digitalWrite functions use a lookup table to figure out at run time which GPIO registers to write data to. Another example is that Arduino keeps track of time, it defines some interrupts by default and all this functionality requires some space. You’ll notice that if you extend the program, the footprint will only change slightly.

I personally like to program controllers with bare minimum, without “bloat”, but you’ll quickly enter the world of EE.SE and SO because several easy to use functions will no longer work out of the box. There are some alternative libraries for pinMode and digitalWrite that compile into a smaller footprint, but come with other disadvantages like for example static compiled pins (where led cannot be a variable, but is a constant).

---

**Answer** by **edgar-bonet**

You already have some perfectly good answers. I am posting this only to share some stats I did one day I asked myself the same sort of questions: What is taking so much space on a minimal sketch? What is the minimum needed to achieve the same functionality?

Below are three versions of a minimal blinky program that toggles the LED on pin 13 every second. All three versions have been compiled for an Uno (no USB involved) using avr-gcc 4.8.2, avr-libc 1.8.0 and arduino-core 1.0.5 (I do not use the Arduino IDE).

First, the standard Arduino way:

```c
const uint8_t ledPin = 13;
```

Skip code block
void setup()
{
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}

This compiles to 1018 bytes. Using both `avr-nm` and disassembly, I broke down that size into individual functions. From largest to smallest:

Skip code block

<table>
<thead>
<tr>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>148 A</td>
<td>ISR(TIMER0_OVF_vect)</td>
</tr>
<tr>
<td>118 A</td>
<td>init</td>
</tr>
<tr>
<td>114 A</td>
<td>pinMode</td>
</tr>
<tr>
<td>108 A</td>
<td>digitalWrite</td>
</tr>
<tr>
<td>104 C</td>
<td>vector table</td>
</tr>
<tr>
<td>82 A</td>
<td>turnOffPWM</td>
</tr>
<tr>
<td>70 A</td>
<td>delay</td>
</tr>
<tr>
<td>40 U</td>
<td>loop</td>
</tr>
<tr>
<td>26 A</td>
<td>main</td>
</tr>
<tr>
<td>20 A</td>
<td>digital_pin_to_timer_PGM</td>
</tr>
<tr>
<td>20 A</td>
<td>digital_pin_to_port_PGM</td>
</tr>
<tr>
<td>16 C</td>
<td>__do_clear_bss</td>
</tr>
<tr>
<td>12 C</td>
<td>__init</td>
</tr>
<tr>
<td>10 A</td>
<td>port_to_output_PGM</td>
</tr>
<tr>
<td>10 A</td>
<td>port_to_mode_PGM</td>
</tr>
<tr>
<td>8 U</td>
<td>setup</td>
</tr>
<tr>
<td>8 C</td>
<td>.init9 (call main, jmp exit)</td>
</tr>
<tr>
<td>4 C</td>
<td>__bad_interrupt</td>
</tr>
<tr>
<td>4 C</td>
<td>_exit</td>
</tr>
</tbody>
</table>

1018 TOTAL

In the list above, the first column is the size in bytes, and the second column tells whether the code comes from the Arduino core library (A, 822 bytes total), the C runtime (C, 148 bytes) or the user (U, 48 bytes).

As can be seen in this list, the largest function is the routine servicing the timer 0 overflow interrupt. This routine is responsible of tracking time, and is needed by `millis()`, `micros()` and `delay()`. The second largest function is `init()`, which sets the hardware timers for PWM, enables TIMER0_OVF interrupt and disconnects the USART (that was used by the bootloader). Both this and the previous function are defined in `<Arduino directory>/hardware/arduino/cores/arduino/wiring.c`.

Next is the C + avr-libc version:

Skip code block

```c
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRB |= _BV(PB5); /* set pin PB5 as output */
    for (; ; ) { /*
        PINB = _BV(PB5); /* toggle PB5 */
        _delay_ms(1000);
    }
}
```

The break-down of the individual sizes:
This is 132 bytes for the C runtime and 26 bytes of user code, including the inlined function _delay_ms().

It could be noted that, since this program does not use interrupts, the interrupt vector table is not needed, and regular user code could be put in its place. The following assembly version does exactly that:

```assembly
Skip code block
#include <avr/io.h>
#define io(reg) _SFR_IO_ADDR(reg)
sbi io(DDRB), 5 ; set PB5 as output
loop:
  sbi io(PINB), 5 ; toggle PB5
  ldi r26, 49 ; delay for 49 * 2^16 * 5 cycles
delay:
  sbiw r24, 1
  sbci r26, 0
  brne delay
rjmp loop
```

This is assembled (with avr-gcc -nostdlib) into only 14 bytes, most of which are used to delay the toggles so that the blink is visible. If you remove that delay loop, you end up with a 6-byte program that blinks too fast to be seen (at 2 MHz):

```assembly
  sbi io(DDRB), 5 ; set PB5 as output
loop:
  sbi io(PINB), 5 ; toggle PB5
rjmp loop
```

Tags: programming (Prev Q) (Next Q), compile (Prev Q) (Next Q)
Q: How much can I recurse? How much can I recurse? How much can I recurse? How much can I recurse? How much can I recurse?

Tags: programming (Prev Q) (Next Q), sram (Next Q)

The Arduino Uno board has limited RAM which means it has a limited call stack available. Sometimes, recursion is the only quick option to implement a certain algorithm. So, given that the call stack is severely limited, what would be a way to find out that given a certain program running on the board, exactly how many recursive calls can you afford before there is a stack overflow (and bad things happen)?

Tags: programming (Prev Q) (Next Q), sram (Next Q)
User: asheeshr

Answer by jfpoilpret

If you really want to recurse (and as @jippie said it is a bad idea; subliminal message: don’t do it) and want to know how much you can recurse, then you will have to perform some calculation and experiments; also you generally will have only an approximation of it as it depends a lot on the memory state at the time your recursive function will be called.

For this, you should first know how SRAM is organized inside AVR-based Arduino (it won’t apply to e.g. the Arduino Galileo by Intel). The following diagram from Adafruit shows it clearly:
Then you need to know the total size of your SRAM (depends on Atmel MCU, hence what kind of Arduino board you have).

On this diagram, it is easy to find out the size of Static Data block as it is known at compile-time and won’t change later on.

The Heap size can be more difficult to know as it can vary at runtime, depending on dynamic memory allocations (malloc or new) performed by your sketch or the libraries it uses. Using dynamic memory is quite rare on Arduino, but some standard functions do it (type String uses it, I think).

For the Stack size, it will also vary during runtime, based on the current depth of function calls (each function call takes 2 bytes on the Stack to store the address of the caller) and the number and size of local variables including passed arguments (that are also stored on the Stack) for all the functions called until now.

So let’s suppose your recurse() function uses 12 bytes for its local variables and arguments, then each call to this function (the first one from an external caller and the recursive ones) will use 12+2 bytes.

If we suppose that:

- you are on Arduino UNO (SRAM = 2K)
- your sketch does not use dynamic memory allocation (no Heap)
- you know the size of your Static Data (let’s say 132 bytes)
- when your recurse() function is called from your sketch, the current Stack is 128 bytes long

Then you are left with 2048 - 132 - 128 = 1788 available bytes on the Stack. The number of recursive calls to your function is thus 1788 / 14 = 127, including the initial call (which is not a recursive one).

As you can see, this is very difficult, but not impossible to find what you want.

A simpler way to get the stack size available before recurse() is called would be to use the following function (found on Adafruit learning center; I have not tested it myself):

```c
int freeRam ()
{
    extern int __heap_start, *__brkval;
    int v;
    return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}
```

I strongly encourage you to read this article at Adafruit learning center.

---

**Answer** by jippie

Recursion is bad practice on a microcontroller as you already stated yourself and you probably want to avoid it whenever possible. On the Arduino site there are some examples and libraries available for checking free RAM size. You can for example use this to figure out when to break recursion or a bit trickier/riskier to profile your sketch and hard code the limit in it. This profile would be required for every change in your program and for every change in Arduino tool chain.
It depends on the function.

Every time a function is called, a new frame is pushed onto the stack. It will usually contain various critical items, potentially including:

- Return address (the point in the code from which the function was called).
- The local instance pointer (`this`) if calling a member function.
- Parameters passed into the function.
- Register values which need to be restored when the function ends.
- Space for local variables inside the called function.

As you can see, the stack space required for a given call depends on the function. For example, if you write a recursive function which only takes an `int` parameter and uses no local variables, it won’t need much more than a few bytes on the stack. That means you can recursively call it far more than a function which takes several parameters and uses a lot of local variables (which will eat up the stack much quicker).

Obviously the state of the stack depends on what else is going on in the code. If you start a recursion directly within the standard `loop()` function, then there probably won’t be a lot on the stack already. However, if you start it nested several levels deep in other functions, then there won’t be as much room. That will affect how many times you can recurse without exhausting the stack.

It’s worth noting that tail recursion optimisation exists on some compilers (although I’m not sure if avr-gcc supports it). If the recursive call is the very last thing in a function, it means it is sometimes possible to avoid altering the stack frame at all. The compiler can just re-use the existing frame, since the ‘parent’ call (so to speak) is finished using it. That will mean you can theoretically keep recursing as much as you like, so long as your function doesn’t call anything else.

---

**Tags:** programming (Prev Q) (Next Q), sram (Next Q)

---

**Q:** Could someone explain this weird looking code, used to setup timers?

**Tags:** programming (Prev Q) (Next Q), timers (Prev Q) (Next Q)

While looking at sketches other people have written, I occasionally come across code that looks somewhat like this:

```c
TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 34286;
TCCR1B |= (1 << CS12);
TIMSK1 |= (1 << TOIE1);
```

All I know is that is has something to with timing/timers (I think). How can I decipher—
and create—code like this? What are TCCR1A, TCCR1B, TCNT1, CS12, TIMSK1, and TOIE1?

**Tags:** programming (Prev Q) (Next Q), timers (Prev Q) (Next Q)

**User:** the-guy-with-the-hat

---

**Answer**

This isn’t weird looking. It’s what normal MCU code actually looks like.

What you have here is an example of the concept of *memory-mapped peripherals*. Basically, the MCU hardware has special locations in the SRAM address space of the MCU assigned to it. If you write to these addresses, the bits of the byte written to address \( n \) control the behaviour of peripheral \( m \).

Basically, certain banks of memory literally have little wires running from the SRAM cell to the hardware. If you write a “1” to this bit in that byte, it sets this SRAM cell to a logical high, which then turns on some portion of the hardware.

If you look into the headers for the MCU, there are great big tables of keyword<->address mappings. This is how things like \( \text{TCCR1B} \) etc… are resolved at compile time.

This memory-mapping mechanism is extremely broadly used in MCUs. The ATmega MCU in the arduino use it, as do PIC, ARM, MSP430, STM32 and STM8 MCU series, as well as lots of MCUs I’m not immediately familiar with.

*Arduino* code is the weird stuff, with functions that access the MCU control registers indirectly. While this is somewhat “nicer” looking, it’s also much slower, and uses a lot more program space.

The mysterious constants are all described in great detail in the [ATmega328P datasheet](https://ws3.sparkfun.com/datasheets/ProductDatasheets/Atmel-328P-Datasheet.pdf), which you really should read if you’re interested in doing anything more then occasionally toggling pins on an arduino.

Select excerpts from the datasheet linked above:
Figure 16-1. 16-bit Timer/Counter Block Diagram

Note:  1. Refer to Figure 1-1 on page 2, Table 14-3 on page 83 and Table 14-9 on page 89 for Timer/Counter1 pin placement and description.

Registers
The Timer/Counter (TCNT1), Output Compare Registers (OCR1A/B), and Input Capture Register (ICR1) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in the section "Accessing 16-bit Registers" on page 114. The Timer/Counter Control Registers (TCCR1A/B) are 8-bit registers and have no CPU access restrictions. Interrupt requests (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR1). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK1). TIFR1 and TIMSK1 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T1 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_T1).

The double buffered Output Compare Registers (OCR1A/B) are compared with the Timer/Counter value at all time. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OC1A/B). See "Output Compare Units" on page 120. The compare match event will also set the Compare Match Flag (OCF1A/B) which can be used to generate an Output Compare interrupt request.
16.11.1 TCCR1A – Timer/Counter1 Control Register A

<table>
<thead>
<tr>
<th>Bit</th>
<th>COM1A1</th>
<th>COM1A0</th>
<th>COM1B1</th>
<th>COM1B0</th>
<th>–</th>
<th>–</th>
<th>WGM11</th>
<th>WGM10</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:6</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R</td>
<td>R</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>5:4</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R</td>
<td>R</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A
- Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. Table 16-1 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a Normal or a CTC mode (non-PWM).

**Table 16-1. Compare Output Mode, non-PWM**

<table>
<thead>
<tr>
<th>COM1A1/COM1B1</th>
<th>COM1A0/COM1B0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal port operation, OC1A/OC1B disconnected.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Toggle OC1A/OC1B on Compare Match.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clear OC1A/OC1B on Compare Match (Set output to low level).</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Set OC1A/OC1B on Compare Match (Set output to high level).</td>
</tr>
</tbody>
</table>

Table 16-2 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

**Table 16-2. Compare Output Mode, Fast PWM**

<table>
<thead>
<tr>
<th>COM1A1/COM1B1</th>
<th>COM1A0/COM1B0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal port operation, OC1A/OC1B disconnected.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)</td>
</tr>
</tbody>
</table>

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 124 for more details.
So, for example, \texttt{TIMSK1 |\(=(1<<\text{TOIE1});\)} sets the bit \texttt{TOIE1} in \texttt{TIMSK1}. This is achieved by shifting binary 1 (0b00000001) to the left by \texttt{TOIE1} bits, with \texttt{TOIE1} being defined in a header file as 0. This is then bitwise ORed into the current value of \texttt{TIMSK1}, which effectively sets this one bit high.

Looking at the documentation for bit 0 of \texttt{TIMSK1}, we can see it is described as

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow interrupt is enabled. The corresponding Interrupt Vector (see “Interrupts” on page 57) is executed when the TOV1 Flag, located in TIFR1, is set.

All the other lines should be interpreted in the same manner.

Some notes:

You may also see things like \texttt{TIMSK1 |\(=(1<<\text{TOIE1});\)}; \texttt{\_BV()} is a \texttt{commonly used macro} originally from the \texttt{AVR libc implementation}. \texttt{\_BV(TOIE1)} is functionally identical to \((1<<\text{TOIE1}),\) with the benefit of better readability.

Also, you may also see lines such as: \texttt{TIMSK1 &\(=(1<<\text{TOIE1});\)) \texttt{or TIMSK1 &\(=(1<<\text{TOIE1});\)) These has the opposite function of \texttt{TIMSK1 |\(=(1<<\text{TOIE1});\)}, in that it \textit{unsets}
the bit TOIE1 in TIMSK1. This is achieved by taking the bit-mask produced by _BV(TOIE1), performing a bitwise NOT operation on it (~), and then ANDing TIMSK1 by this NOTed value (which is 0b11111110).

Note that in all these cases, the value of things like (1 << TOIE1) or _BV(TOIE1) are fully resolved at compile time, so they functionally reduce to a simple constant, and therefore take no execution time to compute at runtime.

Properly written code will generally have comments inline with the code that detail what the registers being assigned to do. Here is a fairly simple soft-SPI routine I wrote recently:

```c
uint8_t transactByteADC(uint8_t outByte)
{
    // Transfers one byte to the ADC, and receives one byte at the same time
    // does nothing with the chip-select
    // MSB first, data clocked on the rising edge
    uint8_t loopCnt;
    uint8_t retDat = 0;
    for (loopCnt = 0; loopCnt < 8; loopCnt++)
    {
        if (outByte & 0x80) // if current bit is high
            PORTC |= _BV(ADC_MOSI); // set data line
        else
            PORTC &= ~(_BV(ADC_MOSI)); // else unset it
        outByte <<= 1; // and shift the output data over for the next iteration
        retDat <<= 1; // shift over the data read back
        PORTC |= _BV(ADC_SCK); // Set the clock high
        if (PINC & _BV(ADC_MISO)) // sample the input line
            retDat |= 0x01; // and set the bit in the retval if the input is high
        PORTC &= ~(_BV(ADC_SCK)); // set clock low
    }
    return retDat;
}
```

PORTC is the register that controls the value of output pins within PORTC of the ATmega328P. PINC is the register where the input values of PORTC are available. Fundamentally, things like this are what happen internally when you use the digitalWrite or digitalRead functions. However, there is a look-up operation that converts the arduino “pin numbers” into actual hardware pin numbers, which takes somewhere in the realm of 50 clock cycles. As you can probably guess, if you’re trying to go fast, wasting 50 clock cycles on an operation that should only require 1 is a bit ridiculous.

The above function probably takes somewhere in the realm of 100-200 clock cycles to transfer 8 bits. This entails 24 pin-writes, and 8 reads. This is many, many times faster then using the digitalWrite functions.

---

**Answer** by [thedoctor](https://www.example.com)

- **TCCR1A** is timer/counter 1 control register A
- **TCCR1B** is timer/counter 1 control register B
- **TCNT1** is timer/counter 1’s counter value
CS12 is the 3rd clock select bit for timer/counter 1
TIMSK1 is timer/counter 1’s interrupt mask register
T0IE1 is the timer/counter 1 overflow interrupt enable

So, the code enables timer/counter 1 at 62.5 kHz and sets the value to 34286. Then it enable the overflow interrupt so when it reaches 65535, it will trigger the interrupt function, most likely labelled as ISR(timer0_overflow_vect)

Tags: programming (Prev Q) (Next Q), timers (Prev Q) (Next Q)

Q: Is it better to use #define or const int for constants?

Arduino is an odd hybrid, where some C++ functionality is used in the embedded world—traditionally a C environment. Indeed, a lot of Arduino code is very C like though.

C has traditionally used #defines for constants. There are a number of reasons for this:

1. You can’t set array sizes using const int.
2. You can’t use const int as case statement labels (though this does work in some compilers)
3. You can’t initialize a const with another const.

You can check this question on StackOverflow for more reasoning.

So, what should we use for Arduino? I tend towards #define, but I see some code using const and some using a blend.

Q: Is it better to use #define or const int for constants?

Tags: programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

User: cybergibbons

Answer by microtherion

It’s important to note that const int does not behave identically in C and in C++, so in fact several of the objections against it that have been alluded to in the original question and in Peter Bloomfields’s extensive answer are not valid:

- In C++, const int constants are compile time values and can be used to set array limits, as case labels, etc.
- const int constants do not necessarily occupy any storage. Unless you take their address or declare them extern, they will generally just have a compile time existence.

However, for integer constants, it might often be preferable to use a (named or anonymous) enum. I often like this because:

- It’s backward compatible with C.
- It’s nearly as type safe as `const int` (every bit as type safe in C++11).
- It provides a natural way of grouping related constants.
- You can even use them for some amount of namespace control.

So in an idiomatic C++ program, there is no reason whatsoever to use `#define` to define an integer constant. Even if you want to remain C compatible (because of technical requirements, because you’re kickin’ it old school, or because people you work with prefer it that way), you can still use `enum` and should do so, rather than use `#define`.

Answer by peter-r.-bloomfield

EDIT: microtherion gives an excellent answer which corrects some of my points here, particularly about memory usage.

As you’ve identified, there are certain situations where you’re forced to use a `#define`, because the compiler won’t allow a `const` variable. Similarly, in some situations you’re forced to use variables, such as when you need an array of values (i.e. you can’t have an array of `#define`).

However, there are many other situations where there isn’t necessarily a single ‘correct’ answer. Here are some guidelines which I would follow:

**Type safety**
From a general programming point-of-view, `const` variables are usually preferable (where possible). The main reason for that is type-safety.

A `#define` (preprocessor macro) directly copies the literal value into each location in code, making every usage independent. This can hypothetically result in ambiguities, because the type may end up being resolved differently depending on how/where it’s used.

A `const` variable is only ever one type, which is determined by its declaration, and resolved during initialisation. It will often require an explicit cast before it will behave differently (although there are various situations where it can safely be implicitly type-promoted). At the very least, the compiler can (if configured correctly) emit a more reliable warning when a type issue occurs.

A possible workaround for this is to include an explicit cast or a type-suffix within a `#define`. For example:

```c
#define THE_ANSWER (int8_t)42
#define NOT_QUITE_PI 3.14f
```

That approach can potentially cause syntax problems in some cases though, depending on how it’s used.

**Memory usage**
Unlike general purpose computing, memory is obviously at a premium when dealing with something like an Arduino. Using a `const` variable vs. a `#define` can affect where the data is stored in memory, which may force you to use one or the other.

- `const` variables will (usually) be stored in SRAM, along with all other variables.
- Literal values used in `#define` will often be stored in program space (Flash memory),
alongside the sketch itself.

(Note that there are various things which can affect exactly how and where something is stored, such as compiler configuration and optimisation.)

SRAM and Flash have different limitations (e.g. 2 KB and 32 KB respectively for the Uno). For some applications, it’s quite easy to run out of SRAM, so it can be helpful to shift some things into Flash. The reverse is also possible, although probably less common.

**PROGMEM**

It’s possible to get the benefits of type-safety while also storing the data in program space (Flash). This is done using the `PROGMEM` keyword. It doesn’t work for all types, but it’s commonly used for arrays of integers or strings.

The general form given in the documentation is as follows:

```plaintext
dataType variableName[] PROGMEM = {dataInt0, dataInt1, dataInt3...};
```

String tables are a bit more complicated, but the documentation has full details.

---

**Q: What is the difference between declaring a variable outside of loop and declaring static inside loop?**

**Tags: programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)**

These are two ways I can hold a variable outside of loop (or any function).

First, I can declare it with global scope outside of loop:

```c
void setup()
{
    Serial.begin(9600);
}

int count = 0;

void loop()
{
    Serial.println(count);
    count++;
    delay(250);
}
```

I can also declare it static inside the loop:

```c
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    static int count = 0;
    Serial.println(count);
    count++;
}
What difference, if any, will this make?

**Tags:** programming (Prev Q) (Next Q)

**User:** cybergibbons

---

**Answer** by asheesh

The most basic difference is of scope.

In the first case, you are declaring a global variable. It is a variable that is accessible in every scope after its definition.

```cpp
void setup()
{
  Serial.begin(9600);
}

void inc();
int count = 0;

void loop()
{
  Serial.println(count);
  count++;
  inc();
  delay(500);
}

void inc() //Can edit the value of count
{
  count = count + 1;
}
```

In the second case, you are declaring a static variable with local scope. The variable will persist for the entire program run similar to global variables, but will be accessible only in the code block it is declared in. This is the same example, with only one change. `count` is now declared as a static variable inside `loop`.

```cpp
void inc();

void loop()
{
  static int count = 0;
  Serial.println(count);
  count++;
  inc();
  delay(500);
}
```

This will not compile as the function `inc()` does not have access to `count`.

Global variables, however seemingly useful, come with some pitfalls. These can even cause damage when it comes to writing programs that can interact with the physical surroundings. This a very basic example of something that is quite likely to happen, as soon as programs start getting larger. A function may inadvertently change the state of a
global variable.

```cpp
void setup()
{
  Serial.begin(9600);
}
void another_function();
int state=0;
void loop()
{
  //Keep toggling the state
  Serial.println(state);
  delay(250);
  state=state?0:1;
  //Some unrelated function call
  another_function();
}
void another_function()
{
  //Inadvertently changes state
  state=1;
}
```

Such cases are very hard to debug. *This* type of problem however, can easily be detected, by simply using a static variable.

```cpp
void setup()
{
  Serial.begin(9600);
}
void another_function();
void loop()
{
  static int state=0;
  //Keep toggling the state
  Serial.println(state);
  delay(250);
  state=state?0:1;
  //Some unrelated function call
  another_function();
}
void another_function()
{
  //Results in a compile time error. Saves time.
  state=1;
}
```

*Answer* by philip-allgaier

From a functional perspective, both versions generate the same result, since in both cases the value of `count` is stored between the executions of the `loop()` (either because it is a global variable or because it is marked as `static` and therefore keeps its value).

So the decision which to choose come down to following arguments:

1. Generally, in computer science, it is encouraged to keep your variables as local as possible in terms of *scope*. This usually results in much clearer code with less side-
effects and reduces chances of someone else using that global variable screwing up your logic). E.g. in your first example, other logic areas might change the count value, whereas in the second, only that particular function loop( )can do so).

2. Global and static variables always occupy memory, where as locals only do when they are in scope. In your above examples that makes no difference (since in one you use a global, in the other a static variable), but in bigger and more complex programs it might and you could save memory using non-static locals. However: If you have a variable in a logic area that is executed very often, consider making it either static or global, since otherwise you loose a tiny bit of performance each time that logic area is entered, since it takes a bit of time to allocate the memory for that new variable instance. You need to find a balance between memory load and performance.

3. Other points such as better layout for static analysis or optimization by the compiler might also come into play.

4. In some special scenarios, there might be issues with the unpredictable initialization order of static elements (not sure about that point, compare this [link](https://example.com) though).

Source: [Similar thread](https://example.com) on arduino.cc

---

**Q: Digital RGB LED animation**

Tags: programming (Prev Q) (Next Q)

I’ve been trying to get colours fade into each other for a project I’m working on. I have achieved this with the rainbow effect that some from Adafruit’s example code, however I want to be able to choose the colours (eg. dark blue into light blue).

I’ve got the colours changing and fading, however the fade turns off all the LEDs and starts to increase the brightness of the new colour. I need the colours to blend rather than fade out and increase in brightness.

Is anyone able to point me in the right direction?

```cpp
#include "LPD8806.h"
#include "SPI.h"
define stripSize 64

int nLEDs = 160;
int dataPin = 2;
int clockPin = 3;

// First parameter is the number of LEDs in the strand. The LED strips
// are 32 LEDs per meter but you can extend or cut the strip. Next two
// parameters are SPI data and clock pins:
LPD8806 strip = LPD8806(stripSize, dataPin, clockPin);

// You can optionally use hardware SPI for faster writes, just leave out
// the data and clock pin parameters. But this does limit use to very
// specific pins on the Arduino. For "classic" Arduinos (Uno, Duemilanove,
// etc.), data = pin 11, clock = pin 13. For Arduino Mega, data = pin 51,
// clock = pin 52. For 32u4 Breakout Board+ and Teensy, data = pin B2,
// clock = pin B1. For Leonardo, this can ONLY be done on the ICSP pins.
//LPD8806 strip = LPD8806(nLEDs);
```
void setup()
{
    // Start up the LED strip
    strip.begin();

    // Update the strip, to start they are all 'off'
    strip.show();
}

void loop()
{
    //turnAllOn(strip.Color(30,30,30),4000);
    fade(0, 127, 0, 100); //red, green, blue, delay - fade up all pixels one color
    //turnAllOn(strip.Color(30,100,30),4000);
    fade(50, 127, 02,100); //red, green, blue, delay - fade up all pixels one color
    //turnAllOn(strip.Color(100,30,100),4000);
    fade(50, 127, 50, 100); //red, green, blue, delay - fade up all pixels one color
}

void fade(uint32_t r, uint32_t g, uint32_t b, uint32_t wait)
{
    int i, j;
    for (j=0; j < 384; j++)
    {
        for (i=0; i < strip.numPixels(); i++)
        {
            strip.setPixelColor(i, strip.Color((r*j)/1000,(g*j)/1000,(b*j)/1000));
        }
        strip.show();
        delay(wait);
    }
}

void turnAllOn(uint32_t c, uint32_t wait)
{
    int i;
    for (i=0; i < strip.numPixels(); i++)
    {
        strip.setPixelColor(i, c); // turn all pixels on
    }
    strip.show(); // write all the pixels out
    delay(wait);
}

uint32_t Wheel(uint16_t WheelPos)
{
    byte r, g, b;
    switch(WheelPos / 128)
    {
        case 0:
            r = 127 - WheelPos % 128; //Red down
            g = WheelPos % 128; // Green up
            b = 0; //blue off
            break;
        case 1:
            g = 127 - WheelPos % 128; //green down
            b = WheelPos % 128; //blue up
            r = 0; //red off
            break;
        case 2:
            b = 127 - WheelPos % 128; //blue down
            r = WheelPos % 128; //red up
            g = 0; //green off
            break;
    }
    return(strip.Color(r,g,b));
}

Tags: programming (Prev Q) (Next Q), led (Prev Q) (Next Q)

User: rhys-edwards

Answer by peter-r.-bloomfield

Currently, your fade function is starting at 0 and effectively interpolating up to the desired colour. To fade between colours, you would need the sketch to remember the previous colour it used, and start fading from that instead of from 0.
The approach I’d use is to start by calculating how much each component needs to change by on each step. For example, if you want to fade from 100 to 200 across 50 steps, then it needs to change by +2 on every step. If you wanted to do the same in reverse (200 to 100) then it would have to change by -2.

One of the problems is that each component will probably be changing by a different amount (red might go from 0 to 200, but blue might only go from 50 to 70). If you’re using integers all the time then it could result in some uneven transitions, so I’d recommend using floating point instead. It’s technically slower (less efficient), but probably not enough to worry about.

This is how I’d probably write it:

```c
void fade(uint8_t oldR, uint8_t oldG, uint8_t oldB, uint8_t newR, uint8_t newG, uint8_t newB, uint32_t numSteps, uint32_t waitPerStep)
{
    // Guard against division by zero
    if (numSteps == 0) numSteps = 1;

    // Calculate how much each colour needs to change on each step
    const float stepR = (newR - oldR) / (float)numSteps,
    stepG = (newG - oldG) / (float)numSteps,
    stepB = (newB - oldB) / (float)numSteps;

    // These values will store our colours on the way along
    float r = oldR, g = oldG, b = oldB;
    uint8_t byteR = oldR, byteG = oldG, byteB = oldB;

    // Go through each fade step
    const uint16_t numPixels = strip.numPixels();
    for (uint32_t step = 0; step < numSteps; ++step) {
        // Move one step towards the target colour
        r += stepR;
        g += stepG;
        b += stepB;

        // Round the colours to integers here so we don't have to do it repeatedly in the loop below
        byteR = (uint8_t)(r + 0.5f);
        byteG = (uint8_t)(g + 0.5f);
        byteB = (uint8_t)(b + 0.5f);

        // Apply the colour to each pixel
        for (uint16_t pixel = 0; pixel < numPixels; ++pixel) {
            strip.setPixelColor(pixel, byteR, byteG, byteB);
        }
    }
    strip.show();
    delay(waitPerStep);
}
```

As you can see, you pass it the old colour (which you’re fading from) and the new colour (which you’re fading to). As I mentioned above, this means your sketch has to remember which colour it previously used, as I don’t think the library provides a way to read the current colour back.

I’ve included some optimisations in there to make it run faster. There’s more you could do to optimise it even further, if necessary.

To use this, you’d do something like this:

```c
// Fade from black to red and pause briefly
fade(0, 0, 0, 255, 0, 0, 100, 10);
```
I’ve made a couple of other changes compared to your own fade function function. First of all, I’ve made it so that you can set the number of steps to fade over. This can be quite useful because bigger colour changes will need more steps to look smooth.

I’ve also modified the wait parameter. In your code, you put the delay after the entire fade was complete, which seems like an odd approach. It makes more sense to allow a small delay between each step of the fade, so you can control how fast it goes.

In my example above, you can see the 100, 10 parameters at the end of each call to fade(). That means it will divide the colour change into 100 steps, with a delay of 10ms between each step. The result is that each fade will take roughly 1 second (not counting the time taken to actually update the LED strip).

---

**Q: Arduino Nano uploading gives error: avrdude: stk500_recv(): programmer is not responding**

I have a Arduino Nano (Sainsmart) that I’m trying to upload a sketch to. Under the Arduino IDE, the device selected was Arduino Nano w/ ATmega328.

However uploading the sketch gives me the error

```
avrdude: stk500_recv(): programmer is not responding
```

I tried both USB ports (/dev/tty.usbserial & /dev/cu.usbserial) but the same error persist. The Arduino is connected to a Macbook Air via the USB cable, and the PWR LED indicator light on the Arduino is turned on and the L indicator LED blinks. There was no problem uploading to a Arduino Uno.

Retried after installing the latest FTDI drivers (MAC OSX, x64, v2.2.18, FTDIUSBSerialDriver_10_4_10_5_10_6_10_7.mpkg) from http://www.ftdichip.com/Drivers/VCP.htm. However that did not help.

What could have gone wrong?
Having a similar issue with my Sainsmart Nano.

Under Preferences check “Show verbose output during [x] upload, then when uploading, you should get a red trace like this:

```
Skip code block
```

```
avrdude: Version 5.11, compiled on Sep 2 2011 at 18:52:52
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2009 Joerg Wunsch
System wide configuration file is "/Applications/Arduino.app/Contents/Resources/Java/hardware/tools/avr/etc/avrdude.conf"
User configuration file is "/Users/johannes/.avrduderc"
User configuration file does not exist or is not a regular file, skipping
Using Port : /dev/tty.usbserial-A403MRTP
Using Programmer : arduino
Overriding Baud Rate : 57600
avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]
```

You’ll notice a little “pause” just before the first line of `avrdude: Send: 0 [30] [20]`. Try uploading again and have you finger ready on the physical reset button of your Nano. **Just the moment before the first `avrdude: ...` appears, press the reset button.**

When you succeed, you’ll see the screen filling with all sort of gibberish, and ending something like this after a few seconds:

```
Skip code block
```

```
avrdude: Recv: . [10]
# | 100% 3.71s
avrdude: verifying...
avrdude: 8824 bytes of flash verified
avrdude: Send: Q [51] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]
avrdude done. Thank you.
```

---

**Answer** by electriccello

I was having the same problem and got the same error message. Turns out these boards don’t come with a bootloader preinstalled. If you have some jumper wires and another working arduino you can use this [tutorial](#) to install the bootloader and it should work great, mine did at least! :)

---

**Tags:** programming (Prev Q) (Next Q), serial (Prev Q) (Next Q), uploading (Prev Q) (Next Q), arduino-nano (Prev Q) (Next Q), sketch (Prev Q) (Next Q)

---

**Q: Should I try to make my sketches as small as possible, even when I have enough room?**

**Tags:** programming (Prev Q) (Next Q), sketch (Prev Q) (Next Q)

There has been a lot of talk about shrinking sketches recently, but if you don’t need the room, should it be done? Will it speed up my program?
Take this code:

```cpp
int led = 13;
int val;

void setup() {
    pinMode(led, OUTPUT);
}

void loop() {
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second

    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second

    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second

    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second

    val = digitalRead(10);
}
```

1,396 bytes on Arduino Uno. Now let’s shrink it a bit:

```cpp
int led = 13;
int val;

void setup() {
    pinMode(led, OUTPUT);
}

void loop() {
    blink();
    val = digitalRead(10);
}

void blink() {
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

1,270 bytes. A 10% decrease! It could be shrunk even more. I have the space... is it more efficient (as far as speed) to make it the most compact I can or leave it “uncompressed”? I would imagine that it would be a little more work (not much) calling `blink()`; therefore slowing down my code. Is this true? Are there other advantages/disadvantages of making it as small as possible (besides storage/distribution of C++ files)?

Tags: programming (Prev Q) (Next Q), sketch (Prev Q) (Next Q)

User: anonymous-penguin

Answer by caseys
Generally speaking, smaller is better. However, there is a point where too small actually makes the program run slower.

My suggestion is if you are working on a sketch and it is blatantly obvious that you are repeating code over and over again, I would rip that out and put it into a function, not only does it make the program smaller, it makes it easier to read.

There is also some optimization going on behind the scenes as well during compilation, so even though the C/C++ code looks to be large, chances are the compiler/linker are determining that something are repeats and consolidating them… Or at least the AVR compiler I used previously did, cannot remember if Arduino does this.

---

**Answer** by alexan_e

You second code may be smaller in size but due to the function call overhead the max execution speed is reduced. Does this matter in your case? No, because you have huge delays anyway, but if the code was an actual series of repeated calculations that should be executed as fast as possible then it would make a difference.

As a general rule, smaller code is not necessarily faster too.

For code optimization (either size or speed) you can refer to this application note from Atmel: [AVR4027: Tips and Tricks to Optimize Your C Code for 8-bit AVR Microcontrollers](http://www.atmel.com/designresources/doc.cfm?docId=17904&FAM=AVR 📲)

I could repeat everything it says here but I don’t think there is a point, you can read the original article directly.

---

Regarding the compiler optimization level. Arduino IDE uses the avr-gcc compiler with a setting for **Os** optimization. The avr-gcc available optimization levels are: [source1](http://www.atmel.com/designresources/doc.cfm?docId=17904&FAM=AVR 📲) [source2](http://www.atmel.com/designresources/doc.cfm?docId=17904&FAM=AVR 📲)

- **O0 or no -O option**
  At this optimization level GCC does not perform any optimization and compiles the source code in the most straightforward way possible. Each command in the source code is converted directly to the corresponding instructions in the executable file, without rearrangement. This is the best option to use when debugging a program and is the default if no optimization level option is specified.

- **O1 or -O**
  This level turns on the most common forms of optimization that do not require any speed-space tradeoffs. With this option the resulting executables should be smaller and faster than with -O0. The more expensive optimizations, such as instruction scheduling, are not used at this level. Compiling with the option -O1 can often take less time than compiling with -O0, due to the reduced amounts of data that need to be processed after simple optimizations.

- **O2**
  This option turns on further optimizations, in addition to those used by -O1. These additional optimizations include instruction scheduling. Only optimizations that do
not require any speed-space tradeoffs are used, so the executable should not increase in size. The compiler will take longer to compile programs and require more memory than with -O1. This option is generally the best choice for deployment of a program, because it provides maximum optimization without increasing the executable size. It is the default optimization level for releases of GNU packages.

- **O3**
  This option turns on more expensive optimizations, such as function inlining, in addition to all the optimizations of the lower levels -O2 and -O1. The -O3 optimization level may increase the speed of the resulting executable, but can also increase its size. Under some circumstances where these optimizations are not favorable, this option might actually make a program slower.

- **Os**
  This option selects optimizations which reduce the size of an executable. The aim of this option is to produce the smallest possible executable, for systems constrained by memory or disk space. In some cases a smaller executable will also run faster, due to better cache usage.

---

**Answer** by amadanon-inc

When considering optimization, think carefully about what resource is most valuable.

You can optimize for code size; this will let you put more code onto your processor. What will the rest of the space be used for? If your code doesn’t fit onto your arduino, and your alternative is buying a larger chip, then this is worth the effort.

You can optimize for speed - you can run more code in the same amount of time. If you are sampling data once per second, and sleeping the rest of the time, then this won’t help.

You can optimize for power usage - very important if you’re on a battery, less important if you are on mains. This might also not just be power consumed by the arduino, but also sensors & motors - can they be turned off for a while?

There are two optimizations that a lot of programmers forget - optimizing for development time, and optimizing for readability.

Any optimization for one thing will tend to de-optimize for at least one (often several) other things. If you don’t know which optimization is going to be most useful, then go for readability, then development time (because today’s readability = tomorrow’s development time x 10!)

Usually, the most expensive thing is the development time (i.e. YOUR time, especially if you are on the clock for a customer). Most other optimization has a threshold - so long as you are over (under?) a specific threshold, further optimization doesn’t help. Code size and execution speed (except when shared between competing tasks) are like this - if your code fits on the chip, then it is small enough. If it needs to do something else later, you can optimize it then. If your program can do all it needs to (assuming it is real-time; calculating Pi is a different case), then that’s as fast as it needs to be.

---

**Tags:** programming (Prev Q) (Next Q), sketch (Prev Q) (Next Q)
Q: Arduino as ISP without a computer

Tags: programming (Prev Q) (Next Q), uploading (Prev Q) (Next Q)

Has anyone seen or know of a method of preloading (using a computer or whatever) a firmware as data onto one arduino set up to operate as an isp and then using that arduino plug the isp interface into another and upload the firmware.

To be clear - i’m looking for a way to make uploading arduino sketches possible without a direct connection to a full blown computer.

Ideally the steps involved would be:

1. Write new sketch/firmware
2. Compile firmware
3. Write a copy firmware onto Arduino A as data (Arduino A is set up to work as ISP)
4. Detach Arduino A from the computer
5. In a separate location attach Arduino A to Arduino B using ISP headers
6. Using some method (maybe a pushbutton) trigger Arduino A to upload the preloaded firmware onto Arduino B.

I’m assuming that theoretically it could work because uploading firmware is just an exchange of bytes across the SPI interface but i’m not sure about the reality.

The scenario for which I’m considering is if I were to install an arduino (permanently) in a location that is remote from my computer (I don’t have a laptop nor do I want to get one) it would be troublesome to reprogram it.

The other method I have considered is using the DIL packaged ATMEGA which I can then remove from the board and program at the computer and return to the board - however I would much prefer not to risk bending of pins and the hassle of having to remove the chip each time I want to reprogram.

Tags: programming (Prev Q) (Next Q), uploading (Prev Q) (Next Q)

User: m3z

Answer by alexan_e

I think µProg – tiny, fast, portable AVR programmer with SD does what you want. You can store multiple files in the SD card (hex, eep etc) and select from the device menu which one to write to the target MCU without any need for a PC.

The best part is that it’s available for free (PCB, schematic, firmware etc).

One of a kind, portable AVR programmer! Helps wherever you need to update the device firmware, where target device is in a hard-to-reach location and you can’t (or don’t want to) bring your laptop with a bunch of wires with you. Trivially easy to use, super cheap to make, super small, super fast, uses SD cards…

Features:
- super small – dimensions 44 x 39 x 5,5 mm
- super fast – write up to 12,5kB/s, read up to 14,5kB/s
- uses cheap storage medium – small MICRO SD cards
- supports FAT16 and FAT32 file systems
- can read, write, verify flash and eeprom memory
- can read, write, verify fusebits and lockbits
- write and read to BIN, HEX, and TXT files
- can set default values of fusebits, erase memories
- cheap, easy to obtain, LPH7779 graphic display
- shows funny animations after every operation
- standard programming header – Atmel 6-PIN ISP
- has a function of auto-update its own firmware (from SD)
- very simple to use, 4 buttons navigation
- user-setting menu
- programming speed auto-selection (up to 4MHz)
- Operates at 3V, programs chips supplied from 3V to 5V

Supported chip list
I have no direct solution for what you want, but yes it is possible to do it. You can hack a bootloader firmware so that it reads the firmware flash code and upload it over ISP.

Though you obviously can’t upload 32k of firmware on the target AVR if both arduinos have only 32k of total flash in their MCUs, you’ll need to have the flasher have more flash than the target if you want to do so (a 1284p or 2560 Mega will be far enough, or a
328/32u4 will be good to flash a 168 or an attiny). The only exception would be to have the firmware to be flashed way smaller than the total memory (like for a blink sketch).

In the end, that won’t be easy, it’s 100% pure AVR code, but you can do it. Have a look at:

- [The bootloader FAQ](#)
- [The Optiboot Project](#) which goal is to make the smallest bootloader possible and
- [AVRISP](#) source code

The way I’d implement that would be to either make everything fit in the bootloader of the flasher device, so it reads the firmware (everything from 0x00 to the bootloader start address) and flashes it through ISP by merging AVRISP with Optiboot.

Anyway, that’s an interesting project for a workshop about bootloaders and AVRs I may suggest at my local hackerspace! ;-)

HTH

---

**Tags:** [programming](#) ([Prev Q](#)) ([Next Q](#)), [uploading](#) ([Prev Q](#)) ([Next Q](#))
Q: strange behaviour of dtostrf()

Tags: programming (Prev Q) (Next Q), sketch (Prev Q) (Next Q)

I’m trying to program a little bit in Arduino, but I’m stuck with probably something trivial.

This is what I have:

```c
char ang[3], lat[9];
dtostrf(GPS.angle, 3, 0, ang);
dtostrf(GPS.latitude, 9, 5, lat);
Serial.println(lat);
Serial.println(ang);
Serial.println("-------");
```

I would expect the following in the serial monitor:

```
5111.60160
267
-------
```

But instead, I’m getting this:

```
5111.60160
2675111.60160
-------
```

So it looks like the `ang` holds both the angle and the latitude….

Why is this happening? And how can I solve this?

My goal is to make one big string, comma separated, from the data stored in `GPS`

Tags: programming (Prev Q) (Next Q), sketch (Prev Q) (Next Q)

User: sturb

---

Answer by jrobert

Your arrays are both too short for the strings they’re meant to hold. In particular, `ang` has three digits and only three bytes. The string-terminator, NUL, ends up in the 1st byte of `lat`. Since you generated the `ang` string first, `lat` over-wrote `ang`’s NUL character, effectively getting appended to `ang`.

The `lat` string will need at least (10+1) bytes; `ang` will need (3+1) bytes, counting only the actual data in your question.

I make it a habit to declare string arrays just as I wrote the sums above, to make it clear that I’ve counted both the contents and the NUL byte, so:

```c
char ang[3+1], lat[10+1];
```

I doubt that you need to specifically add the NUL terminator; it would quite unusual (counter-conventional) for a C/C++ function that generates a string to leave off the terminator (except in a few special cases).
Try increasing just your array sizes first; it'll probably work.

**Q: Manchester Library Won’t Compile for Attiny85**

**Tags:** programming (Prev Q) (Next Q), sketch (Prev Q) (Next Q)

I am creating a wireless sensor using an Attiny85. I want to send the data to an arduino uno, so I purchased the 315mhz rf link kit from spark fun. Since the Attiny85 does not have a TX I decided to use the Manchester library however it won’t compile on the Attiny85.

I followed the steps from this blog: [http://mchr3k-arduino.blogspot.mx/2012/01/wireless-sensor-node-part-2.html?showComment=1338749638806#c853067277980266192](http://mchr3k-arduino.blogspot.mx/2012/01/wireless-sensor-node-part-2.html?showComment=1338749638806#c853067277980266192)

Here is the code I am using:

```cpp
#include <WProgram.h> //otherwise it says it can't find Arduino.h
#include <Manchester.h> //include the library to communicate
#define TxPin 2 //the pin that is used to send data
int sensorPin = 4;
int ledPin = 3;
int count = 50;

void setup(){
  pinMode(ledPin, OUTPUT);
  man.workAround1MhzTinyCore(); //add this in order for transmitter to work with 1Mhz Attiny85/84
  man.setupTransmit(TxPin, MAN_1200); //set transmit pin
}

void loop(){
  if (count == 50){
    digitalWrite(ledPin, HIGH);
    count = 0;
  }
  int data = analogRead(sensorPin);
  man.transmit(data); //transmits and reads the data
  delay(100);
  count ++;
}
```

Here is the error message:

```
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp: In function 'void MANRX_SetupReceive(uint8_t)':
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:366: error: 'TCCR2A' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:366: error: 'WGM21' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:368: error: 'TCCR2B' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:368: error: 'CS21' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:369: error: 'OCR2A' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:379: error: 'TIMSK2' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:379: error: 'OCIE2A' was not declared in this scope
/Users/joelsimonoff/Documents/Arduino/libraries/MANCHESTER/Manchester.cpp:380: error: 'TCNT2' was not declared in this scope
```

**Tags:** programming (Prev Q) (Next Q), compile (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q), attiny (Next Q)

**User:** joel
**Answer** by *ignacio-vazquez-abrams* ▪

**attiny** is missing an option required to allow the Manchester library to function properly on ATtinyX5 devices, specifically definition of `__AVR_ATtinyX5__` when a device is selected. In fact, it’s missing quite a few things.

The package I use for ATtinyX5 support is [arduino-tiny](#). I have verified that it defines that symbol properly. I recommend that you dump your current support package and install arduino-tiny instead.

---

**Answer** by *robert-jenkins* ▪

Having struggled through this myself, I can confirm that Joel’s solution works.

There are quite a lot of posts around that suggests that you can’t get the Manchester to work with Arduino1.0x and you need 0020. But you can.

The key is to use the arduino-tiny from the link above, put the tiny folder which you get from there in `/hardware` and then rename it to attiny and “prospective boards” to boards.

I realise that this doesn’t say any more than Joel already has, but there is so much conflicting and contradictory information around I thought it would be worth adding in my experience.

---

**Tags:** programming (Prev Q) (Next Q), compile (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q), attiny (Next Q)

---

**Q: Allocate object memory statically; initialize it dynamically?**

**Tags:** programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

I have an object whose constructor gets passed a parameter. If I know the parameter value at compile time, I can construct the object statically:

```
static FOOBJ foo(3);
```

(I understand that it isn’t really done statically, i.e. by the compiler, but is actually done during setup).

But if I don’t know the parameter value at compile time, I’d still like to pre-allocate space for the object but construct the object in that space at run time. Can it be done without a separate `.initialize()` method?

**Tags:** programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

**User:** *jrobert* ▪

---

**Answer** by *jfpoilpret* ▪

Using an `initialize()` method to a class is contrary to the principle of a class constructor, i.e. once a class instance has been **constructed**, it should be **“ready to use“**.

As suggested by Ignacio’s answer, C++ placement syntax is much better for your purpose.
However, with Arduino libraries, placement syntax is not supported “out of the box”, so you have to implement it yourself; don’t fear, that is quite straightforward:

```cpp
void* operator new(size_t size, void* ptr)
{
    return ptr;
}
```

Placement syntax can be a complex beast in C++, but for your specific purpose, its usage can be rather simple:

```cpp
static char buffer[sizeof FOOBJ];
static FOOBJ* foo;
void setup() {
    ...
    foo = new (buffer) FOOBJ(3);
    ...
}
```

The difference with your current code is that `foo` is now a pointer, hence any method call will use `->` instead of `.`.

If you absolutely want to keep using `foo` as an instance and not a pointer, then you can do it (but I don’t advise it as explained later) by using a `reference` instead:

```cpp
static char buffer[sizeof FOOBJ];
static FOOBJ& foo = *((FOOBBJ*)buffer);
void setup() {
    ...
    new (buffer) FOOBJ(3);
    ...
}
```

The problem with this code, is that you cannot know if `foo` has already been constructed with a real `FOOBBJ` instance or not; using a pointer, you can always check if it is `0` or not.

Using placement syntax, you must be aware that you cannot delete the `foo` instance above. If you want to destroy `foo` (i.e. ensure that its destructor is called), then you have to explicitly call the destructor:

```cpp
foo->~FOOBBJ();
```

Answer by ignacio-vazquez-abrams

You can use placement syntax to specify an existing allocation in which to instantiate the class.

```cpp
FOOBBJ foo(0);
...
FOOBBJ *f = new (foo) FOOBJ(3);
```

Tags: programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)
I want to do something along the lines of

```cpp
for (int i = 0; i < 4; i++) {
    analogRead(i);
}
```

Which appears to work, but the following does not:

```cpp
for (int i = 0; i < 4; i++) {
    pinMode(i, INPUT);
    pinMode(i + 4, OUTPUT); // should make Analog Pin (i + 4) into an output
    digitalWrite(i + 4, LOW);
    analogRead(i);
}
```

Instead, it appears to treat the pin addressed by `digitalWrite(i + 4, LOW);` as one of the digital pins.

Do I really have to explicitly specify A0, A1, A2, … anytime I want to loop over the analog pins?

---

**Answer** by **craig**

Yes, the analog pins must be addressed using A0, A1,… when using them for digital I/O.

Depending on the board you are using A0,A1,etc. are mapped to different values (for instance it looks like A0 is 18 on some boards but 14 on others.

One solution for looping over the analog pins would be this:

```cpp
static const uint8_t analog_pins[] = {A0,A1,A2,A3,A4};
// Setup pins for input
for (int i = 0; i < 5; i++) { //or i <= 4
    digitalWrite(analog_pins[i], LOW);
}
```

If you are using the analog pins only with the `analogRead()` call you can use 0,1,… instead of A0,A1,…

---

**Answer** by **brettam**

At least an Uno/Megas/leonardos, all the values mapped to analog pin numbers are consecutive, so

```cpp
for (int i = A0; i < A4; i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
}
```

will set A0, A1, A2, and A3 to OUTPUT, and then LOW.

---

**Tags:** programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q), analogread (Prev Q) (Next Q)
**Q: How do I convert code between Arduino platforms?**

**Tags:** programming (Prev Q) (Next Q)

What documentation is available regarding the differences between the different Arduino platforms?

For example, I have some code written for an Arduino Teensy that I would like to run on an Arduino Mega.

Is there documentation on the capabilities and equivalencies between each of the platforms? e.g. memory, number of PWM outputs, etc?

**Tags:** programming (Prev Q) (Next Q)

**User:** mark-harrison

---

**Answer** by ignacio-vazquez-abrams

The most detailed one seems to be Baldengineer’s, but Arduino, SparkFun, and Adafruit have their own as well, although more limited.

As always, the best source for information about them is the datasheets for both the MCU(s) used as well as for the boards themselves.

And note that different macros are predefined for each MCU and feature (e.g. __AVR_ATmega328__ and __AVR_HAVE_MUL__); it is possible to test for their existence in order to customize behavior.

---

**Q: Arduino Remote controlled RGB LED strip, having issues with brightness/dimming**

**Tags:** programming (Prev Q) (Next Q), led (Prev Q) (Next Q)

I have this sketch:

```c
#include <TimerOne.h>
#include <IRremote.h>
#include <RGBMood.h>

int RECV_PIN = 2;  // IR-Receiver PIN
int led = 13;  // Status-LED PIN
int modus;  // Modus for Interrupt-Query
int ledr = 11;  // RGB LED red PIN
int ledg = 12;  // RGB LED green PIN
int ledb = 13;  // RGB LED blue PIN
int SerialBuffer = 0;

RGBMood m(ledr, ledg, ledb);

int timerwert = 20;  // Timer time for Interrupt in ms

String readString;

// Color arrays
```

---
int black[3] = { 0, 0, 0 };  
int white[3] = { 100, 100, 100 }; 
int red[3] = { 100, 0, 0 }; 
int green[3] = { 0, 100, 0 }; 
int blue[3] = { 0, 0, 100 };  
int yellow[3] = { 40, 95, 0 }; 
int dimWhite[3] = { 30, 30, 30 }; 

int brightness = 0; // how bright the LED is 
int fadeAmount = 5; // how many points to fade the LED by 

// etc. 

// Set initial color 
int redVal = black[0]; 
int grnVal = black[1]; 
int bluVal = black[2];

int wait = 10; // 10ms internal crossFade delay; increase for slower fades 
int hold = 0; // Optional hold when a color is complete, before the next crossFade 
int DEBUG = 1; // DEBUG counter; if set to 1, will write values back via serial 
int loopCount = 60; // How often should DEBUG report? 
int repeat = 3; // How many times should we loop before stopping? (0 for no stop) 
int j = 0; // Loop counter for repeat 

// Initialize color variables 
int prevR = redVal; 
int prevG = grnVal; 
int prevB = bluVal;

#define ON 0xF4F37A66 
#define OFF 0x1363ADB4  
#define BRIGHTNESS_UP 0xE6721691 
#define BRIGHTNESS_DOWN 0xE9721848 
#define FLASH 0xFFF00F  
#define STROBE 0xFFF00F9 
#define FADE 0x30ED1255  
#define SMOOTH 0xFFC837 

#define RED 0x9D561314 
#define GREEN 0xCB8E93A5 
#define BLUE 0xC88E8EEC 
#define WHITE 0x16DBBEE3 

#define ORANGE 0xFFF00F4F 
#define YELLOW_DARK 0xFFF00F57 
#define YELLOW_MEDIUM 0xFFF00F67 
#define YELLOW_LIGHT 0xFFF00F77 

#define GREEN_LIGHT 0xFFF00F877 
#define GREEN_BLUE1 0xFFF00F9D 
#define GREEN_BLUE2 0xFFF00FA7 
#define GREEN_BLUE3 0xFFF00FB7 

#define BLUE_RED 0xFFF00F68F 
#define PURPLE_DARK 0xFFF00F6897 
#define PURPLE_LIGHT 0xFFF00F68A8 
#define PINK 0xFFF00F48B7 

IRrecv irrecv(RECV_PIN); 
decode_results results; 

void setup() 
{
  pinMode(ledr, OUTPUT); // Set RGB LED Pins as Output 
  pinMode(ledg, OUTPUT); // Set RGB LED Pins as Output 
  pinMode(ledb, OUTPUT); // Set RGB LED Pins as Output 
  pinMode(led, OUTPUT); // set Status-LED as Output 
  m.setMode(RGBMood::RANDOM_HUE_MODE); // Automatic random fade. 
  m.setHoldingTime(4000); // Keep the same color for 4 seconds before fading again. 
  m.setFadingSteps(150); // Fade with 150 steps. 
  m.setFadingSpeed(50); // Each step last 50ms. A complete fade takes 50*150 = 7.5 seconds 
  m.setHSB((random(359), 255, 255)); 
  Serial.begin(9600); 
  }
irrecv.enableIRIn(); // Start of IR-Receive

Timer1.initialize(timerwert); // Initialisation of Timer-Interrupts
Timer1.attachInterrupt(leseIR); // IR-Read from Interrupt
}

void leseIR(){
  if (irrecv.decode(&results)){
    irrecv.resume(); // Receive the next value
    switch (results.value) {
      case FADE: // Modus Fade (DIY 4)
        modus = 1;
        break;

      case 0xFF906F: // Modus pcambi (DIY 5)
        modus = 2;
        break;

      case ON: //Power
        modus = 0;
        crossFade(white); // RGB LEDs Off
        break;

      case OFF: //Power
        modus = 0;
        crossFade(black); // RGB LEDs Off
        break;

      case BLUE: // Blau 0,0,255
        modus = 0;
        crossFade(blue);
        break;

      case RED: // Rot
        modus = 0;
        crossFade(red);
        break;

      case GREEN: // Grün
        modus = 0;
        crossFade(green);
        break;

      case WHITE: // Weiss
        modus = 0;
        crossFade(white);
        break;

      case BRIGHTNESS_UP: // DIMMING
        modus = 0;
        // fade in from min to max in increments of 5 points:
        for(int fadeValue = 0 ; fadeValue <= 255 ; fadeValue += 5) {
          // sets the value (range from 0 to 255):
          analogWrite(ledg, fadeValue);
        }
        break;

      case BRIGHTNESS_DOWN: // orange
        modus = 0;
        for(int fadeValue = 255 ; fadeValue >= 0 ; fadeValue -= 5) {
          // sets the value (range from 0 to 255):
          analogWrite(ledg, fadeValue);
        }
        break;

      case 0xFFAA55: // Grün mittel
        modus = 0;
        break;

      case 0xFF926D: // blau mittel
        modus = 0;
        break;
    }
  }
}
break;

  case 0xFF12ED: //rosa
     modus = 0;
     break;

  }													//	Switch	END
}

void loop() {
  if(modus==1) {      // Query pb Modus:1
    m.tick();
    Serial.println(results.value, HEX);
    Serial.println(DEC);
    Serial.println(DEC);
    Serial.println(DEC);
  }

  int calculateStep(int prevValue, int endValue) {
    int step = endValue - prevValue; // What's the overall gap?
    if (step) {
      step = 1020/step; // divide by 1020
    }
    return step;
  }

  /* The next function is calculateVal. When the loop value, i,
  reaches the step size appropriate for one of the
  colors, it increases or decreases the value of that color by 1.
  (R, G, and B are each calculated separately.)
  */
  int calculateVal(int step, int val, int i) {
    if (((step) && i % step == 0) { // If step is non-zero and its time to change a value,
      if (step > 0) {           // increment the value if step is positive...
        val += 1;
      }
      else if (step < 0) {      // ...or decrement it if step is negative
        val -= 1;
      }
    }
    // Defensive driving: make sure val stays in the range 0-255
    if (val > 255) {
      val = 255;
    }
    else if (val < 0) {
      val = 0;
    }
    return val;
  }

  /* crossFade() converts the percentage colors to a
  0-255 range, then loops 1020 times, checking to see if
  the value needs to be updated each time, then writing
  the color values to the correct pins.
  */
  void crossFade(int color[3]) {
    // Convert to 0-255
    int R = (color[0] * 255) / 100;
    int G = (color[1] * 255) / 100;
    int B = (color[2] * 255) / 100;

    int stepR = calculateStep(prevR, R);
    int stepG = calculateStep(prevG, G);
    int stepB = calculateStep(prevB, B);
for (int i = 0; i <= 1020; i++) {
    redVal = calculateVal(stepR, redVal, i);
    grnVal = calculateVal(stepG, grnVal, i);
    bluVal = calculateVal(stepB, bluVal, i);
    
    analogWrite(ledr, redVal); // Write current values to LED pins
    analogWrite(ledg, grnVal);
    analogWrite(ledb, bluVal);
    delay(wait); // Pause for 'wait' milliseconds before resuming the loop
}

// Update current values for next loop
prevR = redVal;
prevG = grnVal;
prevB = bluVal;
delay(hold); // Pause for optional 'wait' milliseconds before resuming the loop
}

It’s working like a charm, the fade operates smoothly using timer1, the colors appear correct on led, but I can’t fix the brightness part. Can someone provide sample code to do this?

If anyone else wants to use the code, I’m using an Arduino Mega 2560, please use the same PINs as mine since timers works only on 11,12,13 on this Arduino. (link with pin info regarding several Arduino boards)

- RGB mood library
- TimerOne library
- IR Library

I’m using a small Xbox 360 IR control (no numbers keypad), some HEX codes are left intact from the original sketch found on internet.

original topic

Tags: programming (Prev Q) (Next Q), led (Prev Q) (Next Q)

User: stathis-ntonas

---

**Answer** by geometrikal

You don’t have a delay in the fade loop, this goes from 0 to 255 almost instantly:

```c
for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledg, fadeValue);
}
```

But I’m not sure you want to fade from off to on in one go. If you want to just increment and decrement the brightness:

First notice that prevR, prevG, prevB hold the current led values. So we shall use them.

Change the switch case to:

Skip code block

Case BRIGHTNESS_UP:
    modus = 0;
    int c[3];
    // decrease by 0.9*value but also take off 2 so we get to zero
\[
c[0] = (\text{prevR}-2) * \frac{90}{100};
\]
\[
c[1] = (\text{prevG}-2) * \frac{90}{100};
\]
\[
c[2] = (\text{prevB}-2) * \frac{90}{100};
\]
\[
\text{for( int } j = 0; j < 3; j++ ) \{ \\
\quad \text{if } (c[j] < 0) c[j] = 0;
\}
\]
\[
crossFade(c);
\]
\[
\text{break;}
\]
\[
\text{case BRIGHTNESS_DOWN:}
\quad \text{modus} = 0;
\quad \text{int } c[3];
\quad // \text{increase value by } 1/0.9 \text{ but also add } 2 \text{ so we get off zero}
\quad c[0] = (\text{prevR}+2) * \frac{100}{90} + 2;
\quad c[1] = (\text{prevG}+2) * \frac{100}{90} + 2;
\quad c[2] = (\text{prevB}+2) * \frac{100}{90} + 2;
\quad \text{for( int } j = 0; j < 3; j++ ) \{ \\
\quad \quad \text{if } (c[j] > 100) c[j] = 100;
\quad \}
\quad \text{crossFade(c);}
\quad \text{break;}
\]

I can’t check this to see if it works see please test and see.

**Edit**

Different approach:

Global variable for brightness:

```c
int brightness = 100;
int c[3];
```

Change this value each time:

**Skip code block**

**Case BRIGHTNESS_UP:**

```c
modus = 0;
brightness += 5;
if (brightness > 255) brightness = 255;
crossFade(c);
break;
```

**Case BRIGHTNESS_DOWN:**

```c
modus = 0;
brightness -= 5;
if (brightness < 0) brightness = 0;
crossFade(c);
break;
```

Then in `crossFade` we change the last few lines:

```c
analogWrite(ledr, redVal * brightness / 255); // Write current values to LED pins
analogWrite(ledg, grnVal * brightness / 255);
analogWrite(ledb, bluVal * brightness / 255);
```

**Edit 2:**

I think that what might be happening is that the color is reduced by brightness, then that reduced value is saved in `prevR/G/B` when actually it should just be the full brightness value. Try changing last bit of `crossFade` to:

```c
prevR = color[0];
prevG = color[1];
prevB = color[2];
```

**Tags:** programming (Prev Q) (Next Q), led (Prev Q) (Next Q)
Q: How can I declare an array of variable size (Globally)

Tags: programming (Prev Q) (Next Q), variables (Next Q)

I’d like to make three arrays of the same length. According to the documentation, Arrays must be defined as `int myArray[10];` where 10 can be substituted for a known length (another integer), or filled with an array `{2, 3, 5, 6, 7}`.

However, when I attempted to declare a value `int arrSize = 10;` and then an array based on that size `int myArray[arrSize];`, I get the following: error: array bound is not an integer constant.

Is there a way to varyably determine array sizes, or do I just need to hardcode them? (I was taught hardcoded is bad and something to avoid at all costs.)

Tags: programming (Prev Q) (Next Q), variables (Next Q)

User: user3.1415927

Answer by jfpoilpret

Your question has 2 parts actually.

1/ How can I declare the constant size of an array outside the array?

You can either use a macro

```
#define ARRAY_SIZE 10...
int myArray[ARRAY_SIZE];
```

or use a constant

```
const int ARRAY_SIZE = 10;
...
int myArray[ARRAY_SIZE];
```

if you initialized the array and you need to know its size then you can do:

```
int myArray[] = {1, 2, 3, 4, 5};
const int ARRAY_SIZE = sizeof(myArray) / sizeof(int);
```

the second `sizeof` is on the type of each element of your array, here `int`.

2/ How can I have an array which size is dynamic (i.e. not known until runtime)?

For that you will need dynamic allocation, which works on Arduino, but is generally not advised as this can cause the “heap” to become fragmented.

You can do (C way):

```
// Declaration
int* myArray = 0;
int myArraySize = 0;

// Allocation (let's suppose size contains some value discovered at runtime, // e.g. obtained from some external source)
if (myArray != 0) {
    myArray = (int*) realloc(myArray, size * sizeof(int));
} else {
    myArray = (int*) malloc(size * sizeof(int));
```

```
Or (C++ way):

Skip code block

```cpp
// Declaration
int* myArray = 0;
int myArraySize = 0;

// Allocation (let's suppose size contains some value discovered at runtime,
// e.g. obtained from some external source or through other program logic)
if (myArray != 0) {
    delete [] myArray;
}
myArray = new int [size];
```

For more about problems with heap fragmentation, you can refer to this question [link to question].

Tags: programming (Prev Q) (Next Q), variables (Next Q)
Q: Why use .h files with .cpp?

At school, our teacher always taught us to use .hpp (C++ Header files) with our .cpp (C++ files). But Arduino seems to use .h files (with .cpp links?).

When someone is programming in C (without C++ compiler) they have the chance to get linked to a .cpp file through a .h file?

- So why isn’t a .cpp file always with a .hpp file?
- What are the drawbacks of using .hpp files with your .cpp files?

This is not to start a discussion on what’s best, please provide an answer with arguments that are not based on “what you like to do more”.

Answer by peter-r.-bloomfield

The extension you use for a header file doesn’t technically matter because the compiler never sees it. You could name it with the extension .this-is-a-header-file and it would probably still work.

The compiler only handles the C/C++ source files, typically named .c and .cpp. However, before that happens, the preprocessor goes through and looks for any #include lines. When it sees one, it basically copies the entire contents of the included file into the source file before it gets compiled.

This is normally used to include header files. However, the preprocessor doesn’t know or care about different programming languages. All it does is process the raw text so there’s nothing stopping you from including any text file you like, with any name or extension.

Choosing to use .h or .hpp is mostly just personal preference. You could certainly argue that it lets you (as a programmer) distinguish between C/C++ code more easily.

However, the reason why .h is more common is probably that C and C++ code is mixed so freely. You can generally compile C code as C++ without any problems, in which case it’s the context (rather than the file itself) that tells you if it’s C or C++. It’s also possible to write a library in C++ which only exposes a C interface, meaning it could be used by C programs.

You could definitely include a C++-specific header file in a C program by mistake. All that would happen is that the compiler wouldn’t understand the contents of that header file. Sensible development environments don’t link to a project/library when it sees a #include directive (the Arduino IDE is the only one which does this, as far as I’m aware).
Q: What is the best way to define an I/O pin?

Tags: programming (Prev Q) (Next Q)

I read definitions like

```cpp
const int ledPin = 9;
```

and also

```cpp
#define ledPin 9
```

I know that a definition like

```cpp
int ledPin = 9;
```

is a bad practice if you’re not going to change it (which you usually won’t), though I’ve see this several times in Arduino programs. Which one of the other two is preferred?

Tags: programming (Prev Q) (Next Q)

User: joris-groosman

Answer by srinag-rao

#define ledPin 9 is preferred. By doing `int ledPin = 9;` you will be allocating an int memory whose value is used every time you use `ledPin`. `#define` is different in the sense it doesn’t allocate memory. there is no memory called `ledPin`. Before compiling all “ledPin”s in the code(other than strings) are replaced by 9. So basically

```cpp
digitalWrite(ledPin);
```

becomes

```cpp
digitalWrite(9);
```

Advantages of `#define`: Saves memory and since all `ledPin` are replaced by 9 before execution, it saves processor time.

Doesn’t really matter in small codes...

Answer by peter-r.-bloomfield

Strictly speaking, the `#define` approach will use slightly less memory. The difference is usually tiny though. If you need to reduce memory usage, then other optimisations would probably be far more effective.

An argument in favour of using `const int` is type safety. Wherever you refer to that pin number by variable, you know exactly what data type you’re getting. It might be promoted/converted implicitly or explicitly by the code which uses it, but it should behave in very clear ways.

By contrast, the value in a `#define` is open to interpretation. The vast majority of the time, it probably won’t cause you any problems at all. You just need to be a little careful if you have code which makes assumptions about the type or size of the value.

Personally, I almost always prefer type safety unless I have a very serious need to save
probably the best way would be
const uint8_t LED_PIN = 9; // may require to #include <stdint.h>
or
const byte LED_PIN = 9; // with no include necessary
const unsigned char LED_PIN = 9; // similarly

the name is in caps as per general practice in C++ (and others) to name constants. This
should not use any RAM in itself, and use about 1 byte of program memory per use.
however, there might be problems when the number is higher than 127 and is sign-
extended while getting promoted to larger signed integers (not entirely sure on this),
although that is unlikely to happen with pin numbers.

Tags: programming (Prev Q) (Next Q)

Q: Why doesn’t my code work in a function, yet works inline?

Tags: programming (Prev Q) (Next Q), c (Prev Q) (Next Q)

I’m trying to write a faster shiftOut function, which does not use the slow digitalWrite.
in my code i have the original shiftout, the new shiftOutFast, and the same as
shiftOutFast, but inline. the three blocks are separated by delays to tell them apart on my
scope. Here’s the code:

```c
#define CLR(x,y) (x &= ~(1 << y))
#define SET(x,y) (x |= (1 << y))

void shiftOutFast(uint8_t dataPort, uint8_t dataBit, uint8_t clkPort,
         uint8_t clkBit, uint8_t bitOrder, uint8_t val) {
    for (uint8_t i = 0; i < 8; i++) {
        if (bitOrder == LSBFIRST) {
            if (val & 0x01) {
                SET(dataPort, dataBit);
            } else {
                CLR(dataPort, dataBit);
            }
            val >>= 1;
        } else {
            if (val & 0x80) {
                SET(dataPort, dataBit);
            } else {
                CLR(dataPort, dataBit);
            }
            val <<= 1;
        }
        CLR(clkPort, clkBit);
        SET(clkPort, clkBit);
    }
}
#define dataPin 4
#define clockPin 5
#define dataPort PORTD
#define dataBit PORTD4
#define clockPort PORTD
#define clockBit PORTD5
```
void setup() {
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
}

void loop() {
    shiftOut(dataPin, clockPin, LSBFIRST, 0x55);
    delay(1);
    shiftOutFast(dataPort, dataBit, clockPort, clockBit, LSBFIRST, 0x5C);
    delay(1);
    bool bitOrder = LSBFIRST;
    uint8_t val = 0x5C;
    for (uint8_t i = 0; i < 8; i++) {
        if (bitOrder == LSBFIRST) {
            if (val & 0x01) {
                SET(dataPort, dataBit);
            } else {
                CLR(dataPort, dataBit);
            }
            val >>= 1;
        } else {
            if (val & 0x80) {
                SET(dataPort, dataBit);
            } else {
                CLR(dataPort, dataBit);
            }
            val <<= 1;
        }
    }
    CLR(clockPort, clockBit);
    SET(clockPort, clockBit);
    delay(1);
}

The problem is that calling shiftOutFast doesn’t do zilch. Absolutely nothing. It doesn’t seem to be the logic, because when I do the same inline it works. Any ideas?

Tags: programming (Prev Q) (Next Q), c (Prev Q) (Next Q)

User: joris-groosman

Answer by majenko

The problem is that you are passing the value of the ports to the function, not the address of the ports.

You need to either pass “by reference” or as a pointer, so that modifying the port variable modifies the actual port variable not the copy of the value that you pass.

The simplest way is to modify your function to be:

```c
void shiftOutFast(volatile uint8_t &dataPort, uint8_t dataBit,
    volatile uint8_t &clkPort, uint8_t clkBit,
    uint8_t bitOrder, uint8_t val) {
    ....
}
```

That way, instead of passing whatever is in the port at the time to the function (pass by value) it instead creates a new variable which is located at the same address as the variable that you are passing (pass by reference) so modifications of one variable modifies the other.
Note that only the “port” variables need to be passed by reference, all the others remain as pass by value. Also they should be, as Edgar says, flagged as volatile.

**Answer** by edgar-bonet

You should declare volatile the port parameters of your function. Otherwise, to the compiler, their write accesses seem useless, and they get optimized out.

**Tags:** programming (Prev Q) (Next Q), c (Prev Q) (Next Q)

---

**Q: How can I handle the millis() rollover?**

**Tags:** programming (Prev Q) (Next Q), time (Next Q)

I need to read a sensor every five minutes, but since my sketch also has other tasks to do, I cannot just `delay()` between the readings. There is the [Blink without delay](https://www.arduino.cc/en/Tutorial/Blinkwithoutdelay) tutorial suggesting I code along these lines:

```c
void loop()
{
    unsigned long currentMillis = millis();

    // Read the sensor when needed.
    if (currentMillis - previousMillis > interval) {
        previousMillis = currentMillis;
        readSensor();
    }

    // Do other stuff...
}
```

The problem is that `millis()` is going to roll over back to zero after roughly 49.7 days. Since my sketch is intended to run for longer than that, I need to make sure the rollover does not make my sketch fail. I can easily detect the rollover condition (`currentMillis < previousMillis`), but I am not sure what to do then.

Thus my question: what would be the proper/simplest way to handle the `millis()` rollover?

**Tags:** programming (Prev Q) (Next Q), time (Next Q)

**User:** edgar-bonet

---

**Answer** by edgar-bonet

Short answer: do not try to “handle” the millis rollover, write rollover-safe code instead. Your example code from the tutorial is fine. If you try to detect the rollover in order to implement corrective measures, chances are you are doing something wrong. Most Arduino programs only have to manage events that span relatively short durations, like debouncing a button for 50 ms, or turning a heater on for 12 hours… Then, and even if the program is meant to run for years at a time, the millis rollover should not be a concern.

The correct way to manage (or rather, avoid having to manage) the rollover problem is to think of the unsigned long number returned by `millis()` in terms of [modular](https://www.example.com/modular arithmetic).
For the mathematically inclined, some familiarity with this concept is very useful when programming. For the others, I offer here an alternative (hopefully simpler) way of thinking about it. It is based on the simple distinction between *instants* and *durations*. As long as your tests only involve comparing durations, you should be fine.

**Note on micros():** Everything said here about `millis()` applies equally to `micros()`, except for the fact that `micros()` rolls over every 71.6 minutes, and the `setMillis()` function provided below does not affect `micros()`.

**Instants, timestamps and durations**

When dealing with time, we have to make the distinction between at least two different concepts: *instants* and *durations*. An instant is a point on the time axis. A duration is the length of a time interval, i.e. the distance in time between the instants that define the start and the end of the interval. The distinction between these concepts is not always very sharp in everyday language. For example, if I say “I will be back in five minutes”, then “five minutes” is the estimated *duration* of my absence, whereas “in five minutes” is the *instant* of my predicted coming back. Keeping the distinction in mind is important, because it is the simplest way to entirely avoid the rollover problem.

The return value of `millis()` could be interpreted as a duration: the time elapsed from the start of the program until now. This interpretation, however, breaks down as soon as `millis()` overflows. It is generally far more useful to think of `millis()` as returning a *timestamp*, i.e. a “label” identifying a particular instant. It could be argued that this interpretation suffers from these labels being ambiguous, as they are reused every 49.7 days. This is, however, seldom a problem: in most embedded applications, anything that happened 49.7 days ago is ancient history we do not care about. Thus, recycling the old labels should not be an issue.

**Do not compare timestamps**

Trying to find out which among two timestamps is greater than the other does not make sense. Example:

```c
unsigned long t1 = millis();
delay(3000);
unsigned long t2 = millis();
if (t2 > t1) { ... }
```

Naively, one would expect the condition of the `if()` to be always true. But it will actually be false if `millis()` overflows during `delay(3000)`. Thinking of `t1` and `t2` as recyclable labels is the simplest way to avoid the error: the label `t1` has clearly been assigned to an instant prior to `t2`, but in 49.7 days it will be reassigned to a future instant. Thus, `t1` happens both before and after `t2`. This should make clear that the expression `t2 > t1` makes no sense.

But, if these are mere labels, the obvious question is: how can we do any useful time calculations with them? The answer is: by restricting ourselves to the only two calculations that make sense for timestamps:
1. *later_timestamp - earlier_timestamp* yields a duration, namely the amount of time elapsed between the earlier instant and the later instant. This is the most useful arithmetic operation involving timestamps.

2. *timestamp ± duration* yields a timestamp which is some time after (if using +) or before (if −) the initial timestamp. Not as useful as it sounds, since the resulting timestamp can be used in only two kinds of calculations...

Thanks to modular arithmetics, both of these are guaranteed to work fine across the millis rollover, at least as long as the delays involved are shorter than 49.7 days.

**Comparing durations is fine**

A duration is just the amount of milliseconds elapsed during some time interval. As long as we do not need to handle durations longer than 49.7 days, any operation that physically makes sense should also make sense computationally. We can, for example, multiply a duration by a frequency to get a number of periods. Or we can compare two durations to know which one is longer. For example, here are two alternative implementations of delay(). First, the buggy one:

```plaintext
void myDelay(unsigned long ms) {
    unsigned long start = millis(); // start: timestamp
    unsigned long finished = start + ms; // finished: timestamp
    for (;;) {
        unsigned long now = millis(); // now: timestamp
        if (now >= finished) // comparing timestamps: BUG!
            return;
    }
}
```

And here is the correct one:

```plaintext
void myDelay(unsigned long ms) {
    unsigned long start = millis(); // start: timestamp
    for (;;) {
        unsigned long now = millis(); // now: timestamp
        unsigned long elapsed = now - start; // elapsed: duration
        if (elapsed >= ms) // comparing durations: OK
            return;
    }
}
```

Most C programmers would write the above loops in a terser form, like

```plaintext
while (millis() < start + ms); // BUGGY version
```

and

```plaintext
while (millis() - start < ms); // CORRECT version
```

Although they look deceptively similar, the timestamp/duration distinction should make clear which one is buggy and which one is correct.

**What if I really need to compare timestamps?**

Better try to avoid the situation. If it is unavoidable, there is still hope if it is known that the respective instants are close enough: closer than 24.85 days. Yes, our maximum manageable delay of 49.7 days just got cut in half.
The obvious solution is to convert our timestamp comparison problem into a duration comparison problem. Say we need to know whether instant t1 is before or after t2. We choose some reference instant in their common past, and compare the durations from this reference until both t1 and t2. The reference instant is obtained by subtracting a long enough duration from either t1 or t2:

```c
unsigned long reference_instant = t2 - LONG_ENOUGH_DURATION;
unsigned long from_reference_until_t1 = t1 - reference_instant;
unsigned long from_reference_until_t2 = t2 - reference_instant;
if (from_reference_until_t1 < from_reference_until_t2)
  // t1 is before t2
```

This can be simplified as:

```c
if (t1 - t2 + LONG_ENOUGH_DURATION < LONG_ENOUGH_DURATION)
  // t1 is before t2
```

It is tempting to simplify further into if (t1 - t2 < 0). Obviously, this does not work, because t1 - t2, being computed as an unsigned number, cannot be negative. This, however, although not portable, does work:

```c
if ((signed long)(t1 - t2) < 0) // works with gcc
  // t1 is before t2
```

The keyword signed above is redundant (a plain long is always signed), but it helps make the intent clear. Converting to a signed long is equivalent to setting LONG_ENOUGH_DURATION equal to 24.85 days. The trick is not portable because, according to the C standard, the result is implementation defined. But since the gcc compiler promises to do the right thing, it works reliably on Arduino. If we wish to avoid implementation defined behavior, the above signed comparison is mathematically equivalent to this:

```c
#include <limits.h>
if (t1 - t2 > LONG_MAX) // too big to be believed
  // t1 is before t2
```

with the only problem that the comparison looks backwards. It is also equivalent, as long as longs are 32-bits, to this single-bit test:

```c
if ((t1 - t2) & 0x80000000) // test the "sign" bit
  // t1 is before t2
```

The last three tests are actually compiled by gcc into the exact same machine code.

**How do I test my sketch against the millis rollover**

If you follow the precepts above, you should be all good. If you nevertheless want to test, add this function to your sketch:

```c
#include <util/atomic.h>
void setMillis(unsigned long ms)
{
  extern unsigned long timer0_millis;
  ATOMIC_BLOCK (ATOMIC_RESTORESTATE) {
    timer0_millis = ms;
  }
}
```
and you can now time-travel your program by calling `setMillis(destination)`. If you want it to go through the millis overflow over and over again, like Phil Connors reliving Groundhog Day, you can put this inside `loop()`:

```plaintext
// 6-second time loop starting at rollover - 3 seconds
if (millis() - (-3000) >= 6000)  
    setMillis(-3000);
```

The negative timestamp above (-3000) is implicitly converted by the compiler to an unsigned long corresponding to 3000 milliseconds before the rollover (it is converted to 4294964296).

**What if I really need to track very long durations?**

If you need to turn a relay on and turn it off three months later, then you really need to track the millis overflows. There are many ways to do so. The most straightforward solution may be to simply extend `millis()` to 64 bits:

```plaintext
uint64_t millis64() {
    static uint32_t low32, high32;
    uint32_t new_low32 = millis();
    if (new_low32 < low32) high32++;
    low32  = new_low32;
    return (uint64_t) high32 << 32  |  low32;
}
```

Keep in mind, though, that 64 bit arithmetic is expensive on the Arduino. It may be worth to reduce the time resolution in order to stay at 32 bits.

---

**Answer** by gerben

**TL;DR** Short version:

An unsigned long is 0 to 4,294,967,295 ($2^{32} - 1$).

So lets say previousMillis is 4,294,967,290 (5 ms before rollover), and currentMillis is 10 (10ms after rollover). Then `currentMillis - previousMillis` is actual 15 (not -4,294,967,280) since the result will be calculated as an **unsigned** long (which can’t be negative, so itself will roll around).

So the above code will work perfectly fine. The trick is to always calculate the time difference, and not compare the two time values.

---

**Tags:** programming (Prev Q) (Next Q), time (Next Q)

---

**Q: OOP vs Inline with Arduino**

**Tags:** programming (Prev Q) (Next Q), arduino-mega (Next Q), memory-usage (Next Q)

I have been programming for quite a while now but I am new to Arduino and AVR Programming. The main question I have about programming these Micro-controllers is are there major differences in designing code in Object Orientated Classes vs the more traditional inline programming I have seen in many examples?
In other words in the world of Arduino/AVR Controllers are there any savings with Memory and Performance utilizing classes or visa versa?

Say for example we have a Class:

```cpp
Skip code block

#include<iostream>
using namespace std;

class SomeClass{
    private:
        int x;
        int y;
    public:
        void foo();
        void bar();
    }

SomeClass thisClass;
thisClass.foo();
thisClass.bar();
```

Would there be any performance or memory gains designing the program in a more inline fashion like:

```c
int x;
int y;

void foo(){ /* Do something */};
void bar(){ /* Do more stuff */};
```

I tried doing some searches on Stack Exchange and Google but couldn’t find quite the answer I am looking for the closest thing I was able to find was this Stack Exchange Question.

The reason I am asking about this is I have a project that needs to be as light as possible and I am not clear how I should be designing my program in this environment.

**Edit**

Thank you for the answers, this has shed light on things. There is one thing that I am not quite clear about.

Say you have a class that you are designing that utilizes the u8glib as follows:

```cpp
class UserInterface{
    private:
        U8GLIB_ST7920_128X64 Display;
    public:
        UserInterface();
    }
```

How would you get around using “Dynamic Memory” like:

```c
UserInterface::UserInterface(){
    UserInterface::Display = U8GLIB_ST7920_128X64(LCD_E_PIN, LCD_RW_PIN, LCD_RS_PIN, U8G_PIN_NONE);
}
```

**Tags:** programming (Prev Q) (Next Q), arduino-mega (Next Q), memory-usage (Next Q)

**User:** andy-braham

**Answer** by mikael-patel
The main question I have about programming these Micro-controllers is are there major differences in designing code in Object Orientated Classes vs the more traditional inline programming I have seen in many examples?

... In other words in the world of Arduino/AVR Controllers are there any savings with Memory and Performance utilizing classes or visa versa?

Yes, there is a great difference between using C or C++ for small scale embedded systems such as Arduino/AVR. C++ allows more information to be provide for compiler optimizations.

If you are implementing an OOP framework, platform or runtime C++ and classes can also help with software architecture and reuse. In Cosa a number of OOP design patterns are used to achieve interfaces for both application programmers and device driver programmers. The most common is delegation.

Using abstract classes, virtual member functions, inlining, and templates can help achieve lower foot-print and higher performance than traditional implementations. As an example the Cosa Pin classes are X5-X10 faster than the Arduino core and at the same time smaller in foot-print. Please see the benchmarks.

One thing to “unlearn” from traditional C++ programming is the usage of new/delete (malloc/free). With SRAM size of only a few Kbyte using a heap is very much a risk. The answer is static classes and stack based data.

There is much more to be said about OOP framework architecture but I hope this helps answer your initial questions.

Cheers!

---

Answer by majenko

The reason you can’t find the answer is because the answer is both Yes and No.

For basic class stuff - defining your class with methods etc and instantiating objects from it - there’s little difference in the end result compared with “vanilla” C. The compiler’s optimizations are so good now that performance is just the same. Yes, there may be slight increases in memory use since you’re passing an extra pointer with each method call (instead of foo(int x) you have foo(MyClass *this, int x)) but that’s so small as to not be noticeable.

The big differences come when you start playing with polymorphism and other advanced topics. When start doing these complex programs the compiler is not always unable to work out which functions are required and which aren’t, and it’s not able to cut out the unused functions (“garbage collection”). So you may well end up with bigger code.

That doesn’t mean slower code, just chunks of code that are hanging around that never do anything.

Of bigger importance is managing your dynamic memory better than you are used to. Since there is such a small amount of memory the heap is very small, and as a result it gets...
fragmented very easily. Dynamic creation and destruction of objects (\texttt{new\ myClass, delete\ myClassObject, etc}) is very bad. Class objects really need to either be statically defined (in global scope is the most common) or temporarily allocated on the stack (local instances). Otherwise you’re asking for trouble - and the first you will know of it is strange things happening (no error reports or exceptions, you see…).

Q: Simple URL decoding

How can I do a readable, small, smart and good URL decoder in Arduino?

Now I’m using this code:

```cpp
String GetRidOfurlCharacters(String urlChars) {
    urlChars.replace("%0D%0A", string(\"\n\"));
    urlChars.replace("+", \"\n\");
    urlChars.replace("%20", \"\n\");
    urlChars.replace("%21", \"!\" Español en Wikipedia);
    urlChars.replace("%22", string(char(\"" Español en Wikipedia));
    urlChars.replace("%23", \"#\" Español en Wikipedia);
    urlChars.replace("%24", \"$\" Español en Wikipedia);
    urlChars.replace("%25", \"%\" Español en Wikipedia);
    urlChars.replace("%26", \"&\" Español en Wikipedia);
    urlChars.replace("%27", string(char(39)) Español en Wikipedia);
    urlChars.replace("%28", \"(\" Español en Wikipedia);
    urlChars.replace("%29", \")\" Español en Wikipedia);
    urlChars.replace("%2A", \"\*\" Español en Wikipedia);
    urlChars.replace("%2B", \"+\" Español en Wikipedia);
    urlChars.replace("%2C", \"\,\" Español en Wikipedia);
    urlChars.replace("%2D", \"-\" Español en Wikipedia);
    urlChars.replace("%2E", \".\" Español en Wikipedia);
    urlChars.replace("%2F", \"/\" Español en Wikipedia);
    urlChars.replace("%30", \"0\" Español en Wikipedia);
    urlChars.replace("%31", \"1\" Español en Wikipedia);
    urlChars.replace("%32", \"2\" Español en Wikipedia);
    urlChars.replace("%33", \"3\" Español en Wikipedia);
    urlChars.replace("%34", \"4\" Español en Wikipedia);
    urlChars.replace("%35", \"5\" Español en Wikipedia);
    urlChars.replace("%36", \"6\" Español en Wikipedia);
    urlChars.replace("%37", \"7\" Español en Wikipedia);
    urlChars.replace("%38", \"8\" Español en Wikipedia);
    urlChars.replace("%39", \"9\" Español en Wikipedia);
    urlChars.replace("%40", \"@\" Español en Wikipedia);
    urlChars.replace("%41", \"A\" Español en Wikipedia);
    urlChars.replace("%42", \"B\" Español en Wikipedia);
    urlChars.replace("%43", \"C\" Español en Wikipedia);
    urlChars.replace("%44", \"D\" Español en Wikipedia);
    urlChars.replace("%45", \"E\" Español en Wikipedia);
    urlChars.replace("%46", \"F\" Español en Wikipedia);
    urlChars.replace("%47", \"G\" Español en Wikipedia);
    urlChars.replace("%48", \"H\" Español en Wikipedia);
    urlChars.replace("%49", \"I\" Español en Wikipedia);
    urlChars.replace("%4A", \"J\" Español en Wikipedia);
    urlChars.replace("%4B", \"K\" Español en Wikipedia);
However I want to make a loop and only replace a url escaping if it exists, not trying to replace all the (possible) occurrences.

Tags: programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

User: helio

Answer by majenko

The fact that you’re using a String (urgh, shudder shudder… you should read this: http://hacking.majenko.co.uk/the-evils-of-arduino-strings) actually makes it a little harder to work with. Instead you should have your String data in a character array (“C string”). Then a parser is much easier to write using a pair of character pointers.

Basically the theory is you scan from the start of the string to the end of the string using both pointers copying characters as you go, and as soon as you hit a % you read ahead 2 characters using the first pointer (the second pointer stays where it is) and you convert the two read characters as hex into the ASCII equivalent and use the second pointer to replace the %.

Then both pointers keep moving forward, but they stay out of step. As each % crops up the pointers get further and further out of step until you reach the end of the string and all the %xx have been replaced with a single character.

For instance, this code I have just knocked up does the job:

Skip code block

```c
char data[50] = "http://foo/bar?x=hello%20there%3A%20majenko";

// Create two pointers that point to the start of the data
char *leader = data;
char *follower = leader;

// While we're not at the end of the string (current character not NULL)
```
while (*leader) {
    // Check to see if the current character is a %
    if (*leader == '%') {
        // Grab the next two characters and move leader forwards
        leader++;
        char high = *leader;
        leader++;
        char low = *leader;

        // Convert ASCII 0-9A-F to a value 0-15
        if (high > 0x39) high -= 7;
        high &= 0x0f;

        // Same again for the low byte:
        if (low > 0x39) low -= 7;
        low &= 0x0f;

        // Combine the two into a single byte and store in follower:
        *follower = (high << 4) | low;
    } else {
        // All other characters copy verbatim
        *follower = *leader;
    }

    // Move both pointers to the next character:
    leader++;
    follower++;
} // Terminate the new string with a NULL character to trim it off
*follower = 0;

The data array now contains:

[http://foo/bar?x=hello there: majenko]

Answer by nick-gammon

I wrote a Tiny web server for the Arduino which has a similar thing in it. You might find it useful for your project. Inside there is a function that decodes the URL codes on-the-fly (ie. in a state machine). The code is a bit different to Majenko’s because he assumes that you have the string in memory. In my case I am decoding a byte at a time (so you don’t have to hold a long string in memory before you decode it).

The snippet that does the decoding is this:

```c
// percent-encoding: possible states
enum EncodePhaseType {
    ENCODE_NONE,
    ENCODE_GOT_PERCENT,
    ENCODE_GOT_FIRST_CHAR,
};

// percent-encoding: current state
EncodePhaseType encodePhase;
byte encodeByte; // encoded byte being assembled (first nybble)
```

...
// look for stuff like "foo+bar" (turn the "+" into a space)
// and also "foo%21bar" (turn %21 into one character)
if (percentEncoded)
{
    switch (encodePhase)
    {

        // if in "normal" mode, turn a "+" into a space, and look for "%"
        case ENCODE_NONE:
            if (inByte == '+')
                inByte = ' ';
            else if (inByte == '%')
                {
                    encodePhase = ENCODE_GOT_PERCENT;
                    return; // no addition to buffer yet
                }
            break;

        // we had the "%" last time, this should be the first hex digit
        case ENCODE_GOT_PERCENT:
            if (isxdigit (inByte))
                {
                    byte c = toupper (inByte) - '0';
                    if (c > 9)
                        c -= 7; // Fix A-F
                    encodeByte = c << 4;
                    encodePhase = ENCODE_GOT_FIRST_CHAR;
                    return; // no addition to buffer yet
                }
            // not a hex digit, give up
            encodePhase = ENCODE_NONE;
            flags |= FLAG_ENCODING_ERROR;
            break;

        // this should be the second hex digit
        case ENCODE_GOT_FIRST_CHAR:
            if (isxdigit (inByte))
                {
                    byte c = toupper (inByte) - '0';
                    if (c > 9)
                        c -= 7; // Fix A-F
                    inByte = encodeByte | c;
                }
            else
                flags |= FLAG_ENCODING_ERROR;

            // done with encoding it, or not a hex digit
            encodePhase = ENCODE_NONE;
    } // end of switch on encodePhase
} // end of percent-encoded

// add to value buffer, encoding has been dealt with
valueBuffer [valueBufferPos++] = inByte;
valueBuffer [valueBufferPos] = 0; // trailing null-terminator
} // end of HTTPserver::addToValueBuffer

Tags: programming (Prev Q) (Next Q), c++ (Prev Q) (Next Q)

Q: Combine bits from two ports into a single byte

Tags: programming (Prev Q) (Next Q), c (Prev Q) (Next Q)

I’m attempting to read the values of 8 pins at a time into a byte.

The obvious answer to this seems to be the Port registers, however with the way the Arduino is wired, I’d have to split my read across two registers.
In this case, I need a way to combine the lower 5 bits of PORTB with the upper 3 bits of PORTD into a single byte.

For example, this is the physical layout with the data lines I'm attempt to read/write to and their corresponding port assignments.

```
<table>
<thead>
<tr>
<th>SCSI</th>
<th>PB4</th>
<th>PB3</th>
<th>PB2</th>
<th>PB1</th>
<th>PB0</th>
<th>PD7</th>
<th>PD6</th>
<th>PD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB7</td>
<td>DB6</td>
<td>DB5</td>
<td>DB4</td>
<td>DB3</td>
<td>DB2</td>
<td>DB1</td>
<td>DB0</td>
<td></td>
</tr>
</tbody>
</table>
```

From what I can gather I need to use some shift operators to achieve this, perhaps something like:

```
hi = PORTB & 3;
lo = PORTD >> 3;
```

But I'm unsure if that's correct and then how I would combine them.

Is it possible to write a macro to make getting and setting this byte easier?

Is there a good learning resource on bit manipulation like this?

**Tags:** programming (Prev Q) (Next Q), c (Prev Q) (Next Q)

**User:** peter-godwin

---

**Answer** by gbulmer

Assuming PORTB and PORTD are correctly giving unsigned char values:

```
result = (PORTB << 3) | (PORTD>>5)
```

would do it.

As a macro:

```
#define SCSIBITS ((PORTB << 3) | (PORTD>>5))
```

Then use like this:

```
result = SCSIBITS
```

To make it clearer which specific bits are being used, write:

```
#define SCSIBITS (((PORTB & B00011111) << 3) | ((PORTD & B11100000) >> 5))
```

It should be easier to see that the lower 5 bits of PORTB, and the upper 3 bits of PORTD are being used.

In the general case, where the values being manipulated are not unsigned char (should not be a problem in this specific case, but may be helpful to know), you might write:

```
#define SCSIBITS ((unsigned char)((PORTB & B00011111) << 3) | ((PORTD & B11100000) >> 5))
```

or:
```
#define SCSIBITS (((unsigned char)PORTB & B00011111) << 3) | (((unsigned char)PORTD & B11100000) >> 5)
```

Here are a few links to bit-manipulation tutorials:
[https://www.hackerearth.com/notes/bit-manipulation/](https://www.hackerearth.com/notes/bit-manipulation/)
Q: Return value of 0 if value is less than zero

Tags: programming (Prev Q), sensors (Prev Q) (Next Q)

Background: I am using an analog pressure sensor with a range from 0-100PSI returning voltage values .44v to 4.44v. It’s open air (or 0 PSI) voltage on the signal pin is .47 volts. I have a function that subtracts that .47v (to make zero) and then multiplies it by 25 (25 PSI per volt). Due to signal noise or other factors, sometimes my function returns a negative number, which wreaks havoc on my math. I would like to return a value of 0 when my offset math returns a negative number, but I’m having a problem with it.

The code that I wrote that doesn’t work:

```c
double EngineOilPressure() {
    //insert oil pressure code here
    int reading = analogRead(sensorPin3);
    float voltage = reading * 5.0;
    voltage /= 1024.0;
    Serial.print(voltage); Serial.println(" volts");
    if (voltage-.47 < 0) {
        float voltadj = 0;
    } else {
        float voltadj = (voltage-.47);
    }
    float pressureP = voltadj * 25;
    return PSIToPascal(pressureP);
}
```

The error that I get is

```
In function 'double EngineOilPressure()':
dynamic_engine.pde:41:23: error: 'voltadj' was not declared in this scope
Error compiling.
```

The code that works but sometimes returns a very wrong value:

```c
double EngineOilPressure() {
    int reading = analogRead(sensorPin3);
    float voltage = reading * 5.0;
    voltage /= 1024.0;
    Serial.print(voltage); Serial.println(" volts");
    float pressureP = (voltage -.47) * 25;
    return PSIToPascal(pressureP);
}
```

Tags: programming (Prev Q), sensors (Prev Q) (Next Q)

User: jason-stewart
The scope of each of the two different and unrelated voltadj variables that your code (as below) declares is limited to within its enclosing set of braces.

```c
if (voltage-.47 < 0) {
    float voltadj = 0;
} else {
    float voltadj = (voltage-.47);
}
```

Instead perhaps say

```c
float voltadj = voltage-.47;
if (voltadj < 0) voltadj = 0;
```

or perhaps

```c
float voltadj = voltage < .47? 0 : voltage-.47;
```

or

```c
float voltadj = constrain(voltage-.47, 0.0, 5.0);
```

or

```c
float voltadj = max(0.0, voltage-.47);
```

Also see `max()` and `constrain()` reference pages at arduino.cc.

**Tags:** programming (Prev Q), sensors (Prev Q) (Next Q)
Serial

Serial communication is the standard USB connection between the Arduino and a computer with a standard USB A to B cable or through the TX/RX pins using a USB to serial converter. It can also refer to the serial library.

[Serial is] used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART): Serial. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output. You can use the Arduino environment’s built-in serial monitor to communicate with an Arduino board. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin().

Excerpt from the Arduino Page about Serial.

Most boards have only one serial port. Exceptions:

- Leonardo has two: one connected to USB and the other connected to pins 0 and 1 for connection to UART items.
- The Mega/Mega2560 has 4 serial ports, one connected to a UART to USB converter.
- The DUE also has 4 serial ports, one connected to a UART to USB converter.

The serial is outputted through UART. Then, for the primary serial connection on pins 0 and 1*, the data goes to a UART to serial chip** and then transferred to/from the computer.

*On some boards, such as the Leonardo, it doesn’t use the pins 0 and 1.

**On some boards, such as the Leonardo, there is no USB chip because it is built into the chip itself, or it has no USB connection.

The serial library:

Arduino includes a library for inferencing with the serial ports. Here is some example code:

```cpp
void setup() {
  // open the serial port at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello World!");
  delay(1000);
}
```
**Questions**

**Q: Custom Arduino board beeps and process freezes while I’m uploading a sketch to it. Why?**

**Tags:** serial (Prev Q) (Next Q)

The strangest thing is happening to a standalone Arduino board that I designed and built. The board (whose schematics are below) has the following features:

1. It has a ATmega328P with a 5V voltage regulator and usual circuitry when set up as a standalone controller.
2. It controls a scoreboard with several 7-display digits linked through the connectors on the right (JP1 through JP12).
3. It has cursor buttons decoded using a voltage ladder through ANALOG_0 (A0).
4. It has a Real Time Clock to keep time when it’s turned off.
5. It has an RF receiver module.
6. It has a UART header (JP17) so I can program the board using a serial port.
7. It has a speaker attached to digital pin 3 (D3).

I upload sketches to it using a RS232-to-TTL adapter that I’ve also built (schematics also below) and a Serial-to-USB cable. When programming it, the board behaves much like a [Severino board](#).
What’s strange is that, when I upload a sketch, the process is paused in the middle and then the speaker starts to beep continually. It pauses with the following avrdude messages:

```
```

System wide configuration file is "C:\arduino-1.0.3\hardware\tools\avr\etc\avrdude.conf"

```
Using Port : \\.\COM1
Using Programmer : arduino
Overriding Baud Rate : 115200
```

When I press reset on the board, avrdude continues outputting its messages (such as below), the beep stops, the upload continues and the sketch is uploaded successfully to the board.

```
avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]
avrdude: Send: 0 [30] [20]
... messages and upload continue and completes successfully.
```

I don’t have the rest of the messages handy here, but I hope you get the idea.

So, my questions are:

1. **What is making the upload process pause?**
2. **Why is the buzzer beeping when the process pauses?**

*Tags: serial  (Prev Q)  (Next Q)*

*User: ricardo  *

---

**Answer** by zmo

What is making the upload process pause?

Well, given all the comments made under your question, this is definitely a tricky one. You might want to try to add a decoupling capacitor as suggested by @jfpoilpret, though your schematics look alright to me… (maybe ask a review of it on EE SE).

I’d also advice you to copy the `avrdude` command from Arduino IDE output (you’ll find it at the top), and run it in the command shell with lower bitrate:

```
avrdude -v -v -v -v -C "C:\arduino-1.0.3\hardware\tools\avr\etc\avrdude.conf" -P COM1 -patmega328 -U...
```

the higher the `-B` value is the lower your bitrate is. and `-b` to the correct baudrate for your bootloader (are you sure this is 115200? it’s more common to see 57600).

Did you also set fuses correctly? You might as well want to check the values of the fuses to see if the clocks are correctly set up:

```
avrdude -U lfuse:r::1 -U hfuse:r::1 -U efuse:r::1
```

which will output the values of the fuses. Then to make the values readable, put the values at the bottom of the form on the [fuse calculator site](https://www.fusecalculator.com), after choosing the Atmega328P MCU from the dropdown list.

Double check that:

- you have enough size for the bootloader (usually the more you give it the better it is);
- you have correctly set up the cristals configuration for your design (which may not be compatible with default Uno settings as setup by the Arduino IDE), checking if you correctly set up the external crystal at 16MHz…

Why is the buzzer beeping when the process pauses?

It’s because when the processor does not have the output set up, the values within the registers are in an undefined and unpredictable state. Which means that usually it keeps the last value it had before a reset, or changed state because of some electrostatic weirdness, or is giving some weird PWM output because of the influence of a clock nearby.

So basically, there’s something wrong happening when you’re uploading your code and that wrong thing has an indirect effect on the audio. I wouldn’t worry much about that stuff, but more about the cause of the pausing.

Sadly, I don’t have a direct answer for you, but at best a few paths you should investigate more. HTH
Q: Connecting breakout board to Arduino with UART

I’m using a DigiX (Arduino Due based 3.3V) board to connect to a GPRS/GSM SIM908 breakout board. The SIM908 EVB board has the following pins on its RS232:

I’ve got all three UART TX lines hooked up to the RX Lines (and RX to TX) on my Arduino but the board keeps resetting, which I would guess is some sort of voltage issue (I turned the board off immediately). However there are three MAX3232s on the underside of the board, which I thought would have made it so I could just connect the lines directly to the Arduino. Does anyone know why this might be happening or how to properly connect these two devices?

Additional info:
Pinout of the DigiX.
Schematic of the DigiX.
(The below links are in Chinese):
Here is a SIM908 Hardware guide.
The full schematic of the breakout board.

User: user2757902

Answer by cybergibbons

Both the DigiX and the SIM908 are natively 3.3V boards. They should be able to connect to each other directly.

For some reason, the SIM908 breakout seems to only offer RS232 level serial outputs for the GPRS and GPS. On U3 (the SIM908 module itself) Pins 68/71 offer 3.3V levels for the GPRS (GPRS-TXD and GPRS-RXD) and 15/16 offer 3.3V levels for the GPS (GPS-
TXD and GPS-RXD).

These are converted to RS232 levels by the MAX3232 and presented on J3 and J13. I can't see any point where the 3.3V signals are presented on a connector.

You could either:

- Breakout the 3.3V signals onto connectors by modifying the SIM908 breakout board.
- Build a level converter to take the 3.3V signals from the DigiX to RS232 and then connect them as before.

You may have damaged the DigiX board connecting RS232 levels directly to it however. The ARM based Arduinos are far less tolerant of voltage abuse than the ATmega based Arduinos.

---

**Q: How do I print multiple variables in a string?**

**Tags:** serial  (Prev Q)  (Next Q)  ,  arduino-due  (Prev Q)  (Next Q)

Say I have some variables that I want to print out to the terminal, what’s the easiest way to print them in a string?

Currently I do something like this:

```cpp
Serial.print("Var 1:");Serial.println(var1);
Serial.print(" Var 2:");Serial.println(var2);
Serial.print(" Var 3:");Serial.println(var3);
```

Is there a better way to do this?

**Tags:** serial  (Prev Q)  (Next Q)

**User:** sachleen

**Answer** by asheesh

ardprintf is a function that I hacked together which simulates printf over the serial connection. This function (given at the bottom) can be pasted in the beginning of the files where the function is needed. It should not create any conflicts.

It can be called similar to printf. See it in action in this example:

```
Skip code block
```
The output as expected is:

```
test 2 123456789 g test 2.30
```

The function prototype is:

```
int ardprintf(char *, ...);
```

It returns the number of arguments detected in the function call.

This is the function definition:

```c
#include <stdarg.h>
#include <Arduino.h>

int ardprintf(char *str, ...)
{
  int i, count=0, j=0, flag=0;
  char temp[ARDBUFFER+1];
  for(i=0; str[i]!='\0';i++) if(str[i]=='%') count++;
  va_list argv;
  va_start(argv, count);
  for(i=0,j=0; str[i]!='\0';i++)
  {
    if(str[i]=='%')
    {
      temp[j] = '\0';
      Serial.print(temp);
    }
    else
    {
      temp[j] = str[i];
      j = (j+1)%ARDBUFFER;
      if(j==0)
      {
        temp[ARDBUFFER] = '\0';
        Serial.print(temp);
        temp[0]='\0';
      }
    }
  }
  switch(str[++i])
  {
    case 'd': Serial.print(va_arg(argv, int));
    break;
    case 'l': Serial.print(va_arg(argv, long));
    break;
    case 'f': Serial.print(va_arg(argv, double));
    break;
    case 'c': Serial.print((char)va_arg(argv, int));
    break;
    case 's': Serial.print(va_arg(argv, char *));
    break;
    default: ;
  }
  if(str[i]=='%')
  {
    temp[j] = '\0';
    Serial.print(temp);
    temp[0]='\0';
  }
  return count + 1;
}
```
**To print the % character, use %%.*

Now, available on [Github gists](https://github.com).

**Answer** by madivad

I wouldn’t normally put two answers to a question, but I only just found [this](https://github.com) today, where you can use printf without any buffer.

```c
Skip code block
// Function that printf and related will use to print
int serial_putchar(char c, FILE* f) {
  if (c == '\n') serial_putchar('\r', f);
  return Serial.write(c) == 1? 0 : 1;
}
FILE serial_stdout;
void setup(){
  Serial.begin(9600);
  // Set up stdout
  fdev_setup_stream(&serial_stdout, serial_putchar, NULL, _FDEV_SETUP_WRITE);
  stdout = &serial_stdout;
  printf("My favorite number is %6d!\n", 12);
}
void loop(){
  static long counter = 0;
  if (millis() %300==0){
    printf("millis(): %ld counter: %ld (%02X)\n", millis(), counter, counter++);
    delay(1);
  }
}
```

This still has the floating point limitation.

edit: I thought I would do a little testing on this, and it works quite well. I added a better test to the loop with formatted output.

**Answer** by klaus-warzecha

This is probably not better, just different. You can use the `String` object for output. These objects allow concatenation and support automatic typecasting.

```c
Serial.begin(9600);
String label = "Var";
const byte nValues = 3;
int var[nValues] = {36, 72, 49};
for (int i = 0; i < nValues; i++) {
  String stuff = label + i + ": ";
  Serial.println(stuff + var[i]);
}
```

**Tags:** serial (Prev Q) (Next Q)

**Q:** [Sending Large Amounts of Serial Data](https://www.arduino.cc/en/Reference/Serial)
So in the fields of robotics sometimes you need multiple boards and or computers linked together to share information or save statistical data. Currently I need to send a few different variables over a serial connection and was wondering what was the best way to go about doing such?

So far I’ve determined that sending structures would probably be the easier way to send data. Does anyone know of any other way that might be more efficient?

Please keep in mind that I will essentially be having to send the data for 4 motors, compressor, different temperatures, random things, and the 3 sections of the arm.

---

**Answer** by *steven10172*

With my personal thoughts on Structs being the most efficient way to send many different variables I’ve constructed a library to help make it easier to send structs and variables over serial. [Source Code](#)

In this library it makes sending through serial easily. I’ve used with with hardware and software serial. Usually this is used in conjunct with xbee’s so I can wirelessly send the data to and from the robot.

When sending data it make it simple as it allows you to either send a variable or a struct (it doesn’t care).

**Here is an example of sending a simple char over the serial:**

```cpp
// Send the variable charVariable over the serial.
// To send the variable you need to pass an instance of the Serial to use,
// a reference to the variable to send, and the size of the variable being sent.
// If you would like you can specify 2 extra arguments at the end which change the
// default prefix and suffix character used when attempting to reconstruct the variable
// on the receiving end. If prefix and suffix character are specified they'll need to
// match on the receiving end otherwise data won't properly be sent across

char charVariable = 'c'; // Define the variable to be sent over the serial
StreamSend::sendObject(Serial, &charVariable, sizeof(charVariable));

// Specify a prefix and suffix character
StreamSend::sendObject(Serial, &charVariable, sizeof(charVariable), 'a', 'z');
```

**Example of sending a simple int over the serial:**

```cpp
int intVariable = 13496; // Define the int to be sent over the serial
StreamSend::sendObject(xbeeSerial, &intVariable, sizeof(intVariable));

// Specify a prefix and suffix character
StreamSend::sendObject(xbeeSerial, &intVariable, sizeof(intVariable), 'j', 'p');
```

**Example of sending a struct over serial:**

```cpp
// Define the struct to be sent over the serial
struct SIMPLE_STRUCT {
    char charVariable;
    int intVariable[7];
```
Receiving Examples:

Receiving a char that was send via StreamSend:

```cpp
char charVariable;  // Define the variable on where the data will be put

// Read the data from the Serial object an save it into charVariable once
// the data has been received
byte packetResults = StreamSend::receiveObject(Serial, &charVariable, sizeof(charVariable));

// Reconstruct the char coming from the Serial into charVariable that has a custom
// suffix of a and a prefix of z
byte packetResults = StreamSend::receiveObject(Serial, &charVariable, sizeof(charVariable), 'a', 'z');
```

Receiving an int that was sent via StreamSend:

```cpp
int intVariable;  // Define the variable on where the data will be put

// Reconstruct the int from xbeeSerial into the variable intVariable
byte packetResults = StreamSend::receiveObject(xbeeSerial, &intVariable, sizeof(intVariable));

// Reconstruct the data into intVariable that was send with a custom prefix
// of j and a suffix of p
byte packetResults = StreamSend::receiveObject(xbeeSerial, &intVariable, sizeof(intVariable), 'j', 'p');
```

Receiving a Struct that was sent via StreamSend:

```cpp
Skip code block
```

Once you read the data using `StreamSend::receiveObject()` you need to know if the data was GOOD, Not Found, or BAD.

**Good** = Successful

**Not Found** = No prefix character was found in the specified ostream
Bad = Somehow there was a prefix character found, but the data isn’t intact. Usually it means there was no suffix character found or the data wasn’t the correct size.

Testing Validity of Data:

```cpp
// Once you call StreamSend::receiveObject() it returns a byte of the status of
// how things went. If you run that though some of the testing functions it'll
// let you know how the transaction went
if(SteamSend::isPacketGood(packetResults)) {
    // The Packet was Good
} else {
    // The Packet was Bad
}

if(SteamSend::isPacketCorrupt(packetResults)) {
    // The Packet was Corrupt
} else {
    // The Packet wasn't found or it was Good
}

if(SteamSend::isPacketNotFound(packetResults)) {
    // The Packet was not found after Max # of Tries
} else {
    // The Packet was Found, but can be corrupt
}
```

SteamSend Class:

```cpp
#include "Arduino.h"

#ifndef STREAMSEND_H
#define STREAMSEND_H

#define PACKET_NOT_FOUND 0
#define BAD_PACKET 1
#define GOOD_PACKET 2

// Set the Max size of the Serial Buffer or the amount of data you want to send+2
// You need to add 2 to allow the prefix and suffix character space to send.
#define MAX_SIZE 64

class StreamSend {
private:
    static int getWrapperSize() { return sizeof(char)*2; }
    static byte receiveObject(Stream &ostream, void* ptr, unsigned int objSize, unsigned int loopSize);
    static byte receiveObject(Stream &ostream, void* ptr, unsigned int objSize, unsigned int loopSize, char prefixChar, char suffixChar);
    static char _prefixChar; // Default value is s
    static char _suffixChar; // Default value is e
    static int _maxLoopsToWait;

public:
    static void sendObject(Stream &ostream, void* ptr, unsigned int objSize);
    static void sendObject(Stream &ostream, void* ptr, unsigned int objSize, char prefixChar, char suffixChar);
    static byte receiveObject(Stream &ostream, void* ptr, unsigned int objSize, char prefixChar, char suffixChar);
    static boolean isPacketNotFound(const byte packetStatus);
    static boolean isPacketCorrupt(const byte packetStatus);
    static boolean isPacketGood(const byte packetStatus);
    static void setPrefixChar(const char value) {
        _prefixChar = value;
    }
    static void setSuffixChar(const char value) {
        _suffixChar = value;
    }
    static void setMaxLoopsToWait(const int value) {
        _maxLoopsToWait = value;
    }
    static const char getPrefixChar() { return _prefixChar; }
    static const char getSuffixChar() { return _suffixChar; }
    static const int getMaxLoopsToWait() { return _maxLoopsToWait; }
};

// Preset Some Default Variables
```
Can be modified when seen fit

char StreamSend::_prefixChar = 's'; // Starting Character before sending any data across the Serial
char StreamSend::_suffixChar = 'e'; // Ending character after all the data is sent
int StreamSend::_maxLoopsToWait = -1; // Set to -1 for size of current Object and wrapper

/**
 * sendObject
 *
 * Converts the Object to bytes and sends it to the stream
 *
 * @param Stream to send data to
 * @param ptr to struct to fill
 * @param size of struct
 * @param character to send before the data stream (optional)
 * @param character to send after the data stream (optional)
 */
void StreamSend::sendObject(Stream &ostream, void* ptr, unsigned int objSize) {
    sendObject(ostream, ptr, objSize, _prefixChar, _suffixChar);
}

void StreamSend::sendObject(Stream &ostream, void* ptr, unsigned int objSize, char prefixChar, char suffixChar) {
    if(MAX_SIZE >= objSize+getWrapperSize()) {
        //make sure the object isn't too large
        byte * b = (byte *)ptr; // Create a ptr array of the bytes to send
        ostream.write((byte)prefixChar); // Write the suffix character to signify the start of a stream
        // Loop through all the bytes being send and write them to the stream
        for(unsigned int i = 0; i<objSize; i++) {
            ostream.write(b[i]); // Write each byte to the stream
        }
        ostream.write((byte)suffixChar); // Write the prefix character to signify the end of a stream
    }
}

/**
 * receiveObject
 *
 * Gets the data from the stream and stores to supplied object
 *
 * @param Stream to read data from
 * @param ptr to struct to fill
 * @param size of struct
 * @param character to send before the data stream (optional)
 * @param character to send after the data stream (optional)
 */
byte StreamSend::receiveObject(Stream &ostream, void* ptr, unsigned int objSize) {
    return receiveObject(ostream, ptr, objSize, _prefixChar, _suffixChar);
}

byte StreamSend::receiveObject(Stream &ostream, void* ptr, unsigned int objSize, char prefixChar, char suffixChar) {
    return receiveObject(ostream, ptr, objSize, 0, prefixChar, suffixChar);
}

byte StreamSend::receiveObject(Stream &ostream, void* ptr, unsigned int objSize, unsigned int loopSize, char prefixChar, char suffixChar) {
    int maxLoops = (_maxLoopsToWait == -1) ? (objSize+getWrapperSize()) : _maxLoopsToWait;
    if(loopSize >= maxLoops) {
        return PACKET_NOT_FOUND;
    }
    if(ostream.available() >= (objSize+getWrapperSize())) { // Packet meets minimum size requirement
        if(ostream.read() != (byte)prefixChar) {
            // Prefix character is not found
            // Loop through the code again reading the next char
            return receiveObject(ostream, ptr, objSize, loopSize+1, prefixChar, suffixChar);
        }
        char data[objSize]; //Create a tmp char array of the data from Stream
        ostream.readBytes(data, objSize); //Read the # of bytes
        memcpy(ptr, data, objSize); //Copy the bytes into the struct
        if(ostream.read() != (byte)suffixChar) {
            //Suffix character is not found
            return BAD_PACKET;
        }
        return GOOD_PACKET;
    }
    return PACKET_NOT_FOUND; //Prefix character wasn't found so no packet detected
Q: Serial sensors and the Yun

I am involved in a project that is going to measure PH and chlorine in a swimming pool. The plan is to use a Arduino Yun. The sensors we are going to use are connected using serial. Since we may be using multiple sensors, is there any difference in reliability, latency, known issues (bugs) of using software serial versus pin0 and pin1? Is it any way to avoid software serial pitfalls (if there are any)?

Answer by cybergibbons

The Yun is a bit different to other Arduinos. The main microcontroller is a ATmega32U4 with a build in USB interface. This is used for the connection to the PC so the single USART is not used for this purpose.

However, the single USART is used for connecting to the built in wireless processor. This means that you can’t use the wireless and hardware serial at the same time. I presume you are using the Yun for wireless, so disabling this is not desirable.

SoftwareSerial is relatively good, but does consume a lot of cycles. You can only listen on a single port at a time. It is possible to use pin change interrupts to listen for activity on multiple ports and switch to the one required, but this requires that the other end can wait for a response.

SoftwareSerial will tie up the microcontroller whilst sending, which can lead to problems when supporting low baud rates. Conversely, because it is CPU heavy, it can also have problems with high baud rates. There is a window, between 9600 and 38400 baud, where it seems to work best.

There is another software serial version called AltSoftSerial. This works faster than SoftwareSerial and can transmit and receive at the same time.
You can use the USART, SoftwareSerial and AltSoftSerial all at the same time. Be warned that because all three of these use interrupts and some use timers, you are highly likely to end up with complex, hard to debug interactions.

Note than in older versions of Arduino (pre v1), SoftwareSerial was very, very poor. It was replaced entirely with “NewSoftSerial”.

Sensors are often have very basic USART hardware with limited buffer sizes, and drop data when they aren’t serviced in a timely manner.

I am not a massive fan of the Yun. It isn’t popular, the wireless is still relatively hard to use (compared to the TI CC3000), and the wireless interface uses valuable pins. Personally, if I was to work on a project that required multiple serial interfaces, I would either

1. Use a board with multiple USART, such as a ATmega1280 (2 USARTs) or ATmega2560 (4 USARTs) and a CC3000 (SPI connected) wireless board.
2. Use SPI/I2C->USART bridges. This allows a single SPI/I2C port to control multiple USARTs. Sparkfun sell one, there are many others available.

Be cautious of falling into the trap of using hardware because it is what you have available.

---

**Answer** by sachleen

I don’t have any experience with the Yun so I can’t comment on that specifically, but the only problem you’ll encounter with using SoftwareSerial is that you can only communicate with one device at a time.

This means you’ll have to make a call to `listen()` before communicating with sensor1. Then when you want to communicate with another sensor, you have to do the same.

---

**Tags:** serial, sensors, arduino-yun

**Q: How does this code line detect if a serial interface is connected?**

**Tags:** serial, library

I am working with some code for remote temperature sensor nodes. They turn the debug interface on and off depending on if anything is conneced to the serial port or not.

The code is on Github here. Line 111 is the line of interest:

```
if (Serial) debug = 1; else debug=0; //if serial UART to USB is connected show debug O/P.
```

So, a simple question: On a hardware and software level, how does this work? I can’t see anything in HardwareSerial or Stream that would return true/false if anything was connected or not.
Note that I am not confident that it actually works. This simple test indicates it doesn’t work as the LED flashes regardless of the presence of a serial adapter or connection:

```cpp
Skip_code_block
boolean debug;
const int toggle = 6;

void setup()
{
  if (Serial) debug = 1; else debug = 0;
  pinMode(toggle, OUTPUT);
  digitalWrite(toggle, LOW);
  if (debug)
    {
      Serial.begin(9600);
      Serial.println("Setup");
      digitalWrite(toggle, HIGH);
      delay(1000);
      digitalWrite(toggle, LOW);
    }
}

void loop()
{
  if (debug)
    {
      Serial.println("Loop");
      digitalWrite(toggle, HIGH);
      delay(1000);
      digitalWrite(toggle, LOW);
      delay(1000);
    }
}
```

**Tags:** serial (Prev Q) (Next Q), library (Prev Q) (Next Q)

**User:** cybergibbons

**Answer** by peter-r.-bloomfield

You’re right — it doesn’t work in most cases, and will almost always return true. The one board where it’s actually functional is the Leonardo. According to the official documentation:

> On the Leonardo, **if (Serial)** indicates wether or not the USB CDC serial connection is open. For all other instances, including **if (Serial1)** on the Leonardo, this will always returns true.

Basic serial connections (used by most Arduinos) usually don’t care if anything is actually listening. It’s not uncommon for embedded devices to send debug info by serial even when nothing is receiving it. This has the advantage that the code’s timing and behaviour won’t change when debugging, which could cause all sorts of problems if you’re trying to diagnose a problem reported in the field.

**Answer** by asheeshr

While it may not be possible to detect whether a device is connected to the Arduino serial connection or not, it is possible to **enable the debug messages over the serial connection**
based on the presence of a device on the serial connection.

Considering that you will be using the debug interface of your sketch only when connected to a computer capable of serial communication, it is possible to incorporate a simple test into the sketch based on which debug mode is enabled or not.

When the sketch starts, you can check to see if any data is available on the Serial connection or not. If there is, then a device is present and debugging can be enabled. On the computer side, everytime you want to start the debugging mode on the Arduino, simply send a byte over the serial connection during the setup phase and sit back.

Here is a sample sketch showing the same:

```
int debug = 0;
void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(9600);

  //Wait for four seconds or till data is available on serial,
  //whichever occurs first.
  while(Serial.available()==0 && millis()<4000);

  //On timeout or availability of data, we come here.
  if(Serial.available()>0)
  {
    //If data is available, we enter here.
    int test = Serial.read(); //We then clear the input buffer
    Serial.println("DEBUG"); //Give feedback indicating mode
    debug = 1; //Enable debug
  }
}
void loop()
{
  if(debug==1) Serial.println("ON");
  digitalWrite(13,HIGH);
  delay(1000);
  if(debug==1) Serial.println("OFF");
  digitalWrite(13,LOW);
  delay(1000);
}
```

**Tags:** serial (Prev Q) (Next Q), library (Prev Q) (Next Q)

---

**Q: Why does starting the serial monitor restart the sketch?**

**Tags:** serial (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q)

If I upload any sketch that sends serial data, I immediately see the TX/RX LEDs flash once the sketch is uploaded. If I then start the serial monitor, the sketch appears to restart.

A bare minimum sketch that shows this behaviour:

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Setup");
}
```
void loop()
{
    Serial.println("Loop");
    delay(1000);
}

Tested with several boards and Mac and Windows versions of the IDE.

Example output - it goes back to “Setup” when I open the serial monitor:
Why is this?

**Tags:** serial (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q)

**User:** cybergibbons

---

**Answer** by sachleen

The Arduino uses the **RTS (Request To Send)** (and I think **DTR (Data Terminal Ready)**) signals to auto-reset. If you get a serial terminal that allows you to change the flow control settings you can change this functionality.

The Arduino terminal doesn’t give you a lot of options and that’s the default. Others will allow you to configure a lot more. Setting the flow control to none will allow you to connect/disconnect from the serial without resetting your board. It’s quite useful for debugging when you want to be able to just reset the board without having to start the sketch over.

Another way to disable the auto reset is to put a pull up resistor on the reset pin.

**Disabling Auto Reset On Serial Connection**

---

**Tags:** serial (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q)
I have done quite a few projects with Arduino, but have never done more than a “Hello World” in C. I am writing my first library, and a lot of the object oriented stuff is going over my head. Anyway, my library is working as it is supposed to, I pass a float to my function and it formats it properly and sends it out over serial. However, I implemented it in a clunky way. The Arduino API Style Guide mentions a way that seems much better:

When using serial communication, allow the user to specify any Stream object, rather than hard-coding “Serial”. This will make your library compatible all serial ports on Mega and the Due, and can also use alternate interfaces like SoftwareSerial. The Stream object can be passed to your library’s constructor or to a begin() function (as a reference, not a pointer). See Firmata 2.3 or XBee 0.4 for examples of each approach.

Can anybody elaborate on how to do this, or show me some code?

Here is a simple example (built and tested with a mega2560), with a class that can be passed a Stream object, and sends a Hello over this generic stream object. When constructing the object, you can pass the Stream object you want to actually communicate:

```cpp
#include "Arduino.h"

class MyProtocol
{
public:
    MyProtocol(Stream& s):serial(s){}
    void send(){
        serial.println("Hello");
    }

private:
    Stream& serial;
};

MyProtocol p(Serial);

void setup() {
    Serial.begin(9600);
}

void loop(){
    delay(1000);
    p.send();
}
```

NOTE: The serial.println() is not the Serial global object, note the lower case, it is the internal Stream class variable.

If you are confused with the OO and classes, take into account (while you gain more knowledge about it) that this approach can be also used with a simple function, you can
pass the serial parameter. Furthermore, it is not necessary to hardwire the Serial interface for ever, you can in fact switch in run time and use a different channel depending on some condition:

```
#include "Arduino.h"

void send(Stream& serial){
    serial.println("Hello");
}
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
}
void loop() {
    delay(1000);
    if(someCondition)
        send(Serial);
    else
        send(Serial1);
}
```

**Tags:** serial (Prev Q) (Next Q), c++ (Prev Q) (Next Q), library (Prev Q) (Next Q)

---

**Q: How do I split an incoming string?**

**Tags:** serial (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

I am sending a list of servo positions via the serial connection to the arduino in the following format

```
1:90&2:80&3:180
```

Which would be parsed as:

```
servoId : Position & servoId : Position & servoId : Position
```

How would I split these values up, and convert them to an integer?

**Tags:** serial (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

**User:** valrikrobot

**Answer**

Contrarily to other answers, I’d rather stay away from String for the following reasons:

- dynamic memory usage (that may quickly lead to heap fragmentation and memory exhaustion)
- quite slow due to construction/destruction/assignment operators

In an embedded environment like Arduino (even for a Mega that has more SRAM), I’d rather use standard C functions:

- `strchr()` : search for a character in a C string (i.e. char *)
- `strtok()` : splits a C string into substrings, based on a separator character
• **atoi()**: converts a C string to an int

That would lead to the following code sample:

```c
// Calculate based on max input size expected for one command
#define INPUT_SIZE 30...

// Get next command from Serial (add 1 for final 0)
char input[INPUT_SIZE + 1];
byte size = Serial.readBytes(input, INPUT_SIZE);
// Add the final 0 to end the C string
input[size] = 0;

// Read each command pair
char* command = strtok(input, "&");
while (command != 0) {
    // Split the command in two values
    char* separator = strchr(command, ':');
    if (separator != 0) {
        // Actually split the string in 2: replace ':' with 0
        *separator = 0;
        int servoid = atoi(command);
        ++separator;
        int position = atoi(separator);
        // Do something with servoid and position
    }
    // Find the next command in input string
    command = strtok(0, "&");
}
```

The advantage here is that no dynamic memory allocation takes place; you can even declare `input` as a local variable inside a function that would read the commands and execute them; once the function is returned the size occupied by `input` (in the stack) is recovered.

**Answer**

This function can be used to separate a string into pieces based on what the separating character is.

```c
String xval = getValue(myString, ':', 0);
String yval = getValue(myString, ':', 1);
Serial.println("Y:" + yval);
Serial.print("X:" + xval);
```

**Convert String to int**

```c
int xvalue = stringToNumber(xval);
int yvalue = stringToNumber(yval);
```

This Chunk of code takes a string and separates it based on a given character and returns the item between the separating character:

```c
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {
        0, -1
    };
    int maxIndex = data.length()-1;
    for(int i=0; i<=maxIndex && found<=index; i++){
        if(data.charAt(i)==separator || i==maxIndex){
            strIndex[found] = i;
            found = found + (i>0 ? 1 : 0);
        }
    }
    return strIndex[index];
}
```
found++;
strIndex[0] = strIndex[1]+1;
strIndex[1] = (i == maxIndex) ? i+1 : i;
}
return found>index ? data.substring(strIndex[0], strIndex[1]) : "";

Answer by federico-fissore

You can use `Stream.readStringUntil(terminator)` passing a different terminator for each part.

On each part you then call `String.toInt`

Tags:serial (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

Q: How does the Arduino handle serial buffer overflow?

Tags: serial (Prev Q) (Next Q)

How does the Arduino handle serial buffer overflow? Does it throw away the newest incoming data or the oldest? How many bytes can the buffer hold?

Tags: serial (Prev Q) (Next Q)

User: highlife

Answer by peterj

For hardware serial ports you can see in `HardwareSerial.cpp` that the buffer size varies depending on the amount of RAM available on the particular AVR:

```
#if (RAMEND < 1000)
  #define SERIAL_BUFFER_SIZE 16
#else
  #define SERIAL_BUFFER_SIZE 64
#endif
```

For a software serial port in `SoftwareSerial.h` the receiver buffer size `_SS_MAX_RX_BUFF` is defined as 64 bytes. In both cases it stops attempting to insert received data into the queue when it is full, so you could get a mix to old and new data depending on how you’re retrieving data from the queue.

Ideally it would be best to ensure the buffer always gets emptied in a prompt manner to avoid the buffer filling. Maybe take a look at timers and implementing a simple state machine if your problem is related to other code blocking the main loop.

Tags: serial (Prev Q) (Next Q)
I’m implementing a filter for my IMU Sensor and thus I want close to real time data visualized on the computer. I use binary serial communication to facilitate the sending part for the arduino (as far as I know the serial.print is pretty slow). So I split my int16_t in two bytes and send it, like:

```
Serial.write((uint8_t)(gx >> 8)); Serial.write((uint8_t)(gx & 0xFF));
```

After that i directly send the next number (3 in total by now, maybe up to 7 2byte numbers in the future). I read the thing in matlab with:

```
dt(k) = toc;
tic;
bindata([1:6],k) = fread(s,[6,1],'int8');
time = cumsum(dt(1:k));
```

Which reads 6 bytes (3 numbers) and then I recalculate the binary representation, concatenate them and get the original number (if someone can suggest an easier way.. I found matlab pretty unhandy here).

The problem is, that the numbers get mixed by the time. Somewhen one byte isn’t read or anything, so the bytes get messed up and a nonsense number is produced. One full number (2byte) is skipped for one sample exactly. Instead of this number one number is there twice. The next sample the order is messed up (shifted, such that the first number is second). This process appears after maybe 30 sec, sometimes a few minutes. After the first time it keeps shifting and jumping around.

Can someone tell me, what to do here? Can I include some ‘breakpoint’/line terminator, where the reader (matlab) knows, that we are at the start of the first number? Or how is this done actually?

I guess I have to add my main goal: I want to make the sending as fast for the arduino as possible. No extra calculations should be necessary (if possible). And: the reason for these shifts seems to be the some time delay (slowness). I suspect it to be a slow matlab reading, since I saw fluent processing scripts in HIL reading. The errors have stopped however, since I turned the baud rate down. Only wrong numbers are the problem still.

May there be a possibility to loop around the fread and read just store the values after an added ‘header’? So let’s say a wrong ordering occurs. Then I discard everything until the next ‘a’ char/byte and use the following 6 bytes to produce my 3 values. Then I wait for an ‘a’ again. For that I would have to loop fread(s,[1,1],'int8'); and search for the header.

Full arduino code:

```
Skip_code_block
```

```
// Program to send the gyro/accel data via serialport
// corresponding matlab programs: sensing.m and sensing_binary.m
// 2 security loops to guarantee a constant sampling time

// #define DEBUG
#include "GY86.h"
#include "Wire.h"
```
GY86 gy86;
int16_t ax, ay, az;
int16_t gx, gy, gz;

uint32_t currenttime = 0;
uint32_t starttime = 0;
uint32_t starttime2 = 0;

// #define OUTPUT_ACCEL_COUNTS
#define OUTPUT_GYRO_COUNTS

// #define OUTPUT_ACCEL_BINARY
// #define OUTPUT_GYRO_BINARY

void setup () {
    Serial.begin(9600);
    gy86.setup();
}

void loop () {
    currenttime = millis();
    if (currenttime-starttime > 9) {
        while (micros()-starttime2 < 9000) {}  
        starttime2 = micros();
        // read raw accel/gyro measurements from device
        gy86.getSensorValues(&ax, &ay, &az, &gx, &gy, &gz);
        // testing constants
        // gx = -29;
        // gy = 245;
        // gz = 17;

        #ifdef OUTPUT_GYRO_COUNTS
            Serial.print((int)gx);	Serial.print(F("\t"));
            Serial.print((int)gy);	Serial.print(F("\t"));
            Serial.print((int)gz);	Serial.print(F("\t"));
        #endif
        #ifdef OUTPUT_ACCEL_COUNTS
            Serial.print(ax);		Serial.print(F("\t"));
            Serial.print(ay);		Serial.print(F("\t"));
            Serial.println(az);	Serial.print(F("\t"));
        #endif
        #if defined(OUTPUT_ACCEL_COUNTS) || defined(OUTPUT_GYRO_COUNTS)
            Serial.print(F("\n"));
        #endif
        #ifdef OUTPUT_ACCEL_BINARY
            Serial.write((uint8_t)(ax >> 8));	Serial.write((uint8_t)(ax & 0xFF));
            Serial.write((uint8_t)(ay >> 8));	Serial.write((uint8_t)(ay & 0xFF));
            Serial.write((uint8_t)(az >> 8));	Serial.write((uint8_t)(az & 0xFF));
        #endif
        #ifdef OUTPUT_GYRO_BINARY
            Serial.write((uint8_t)(gx >> 8));	Serial.write((uint8_t)(gx & 0xFF));
            Serial.write((uint8_t)(gy >> 8));	Serial.write((uint8_t)(gy & 0xFF));
            Serial.write((uint8_t)(gz >> 8));	Serial.write((uint8_t)(gz & 0xFF));
        #endif
        starttime = currenttime;
    }  
}  

Tags: serial (Prev Q) (Next Q), sensors (Prev Q) (Next Q)
User: mike

Answer by anonymous-penguin

Edit: It seems as the problem is overflow. That means that the USB connection isn’t keeping up with the data trying to be sent. To fix this you need to do one of the following
Higher baud rate. The baud rate is the frequency that data is sent. From what I’ve heard, anything above 500,000 as a baud rate isn’t helpful with the Arduino libraries.

- The Arduino IDE only goes up so high. Try an application such as PuTTY to get higher baud rates on the serial monitor.
- Doing very high rates like this are best suited on as short of a cable as you can manage. I’d say 4 feet max, although it depends on many factors, including cable quality. A shorter cable has less resistance (thus less errors).
- The rest of the answer still applies. You might want to add a simple parity bit to make sure the data doesn’t get corrupted while sent. Adding two or three characters greatly reduces the risk of corruption, but at the expense of cutting your sampling rate in half and it doesn’t verify data integrity. I don’t know your exact situation, so adding a bit might not be possible.

Lower sampling rate: you’re sending too much, so a simple solution is just to add delay(250); at the end of the loop so you don’t overload the port.

Original Answer:

The only thing that comes to mind that is very efficient is a parity bit with another bit that’s always the opposite of the parity bit. Why? Having an accidental thing where there the last two bits are opposite of each other and they all add up to a even number (ignoring the last bit) would be really odd.

A parity bit is an extra bit so all the bits added up with the parity bit equals an even number. If it isn’t quite right, then you know there’s a problem. It works only for odd numbers of bits changed, so it isn’t foolproof. An example is you have the bits 10010110. There are four 1s, so it’s an even number, thus the parity bit will be 0. If it was an odd number, it would be 1 to make the total count an even number. If the computer calculates it doesn’t add up right (excuse my lame pun) then it’s corrupted and the computer can discard it.

To implement this you’d need to convert the number(s) and the parity bit to ASCII and then count the 0s and 1s. You can use a remainder function and divide by two so there will be a remainder of 1 if it’s odd, thus it’s corrupted. I’d personally take the latest bit and the x number before it, and just keep looping until you find a combination that satisfies the whole parity bit thing and is within a reasonable range that you’ve specified in the code.

A line-break would suffice, but it takes up more bits and discovers only missing data, not corrupt data.

Maybe you should be looking into why there’s a problem. Could you reduce the length of the USB cable? Upgrade your cable/try a different one? Slightly reduce the baud rate?

Answer by brettam

You could add some kind of packet header/footer to the serial output. For example, if you sent the characters “ab” before the bytes for your number you could reject packets that only had 2 bytes in them. The serial would look like
“ab123ab123ab12ab123”

The matlab code could see another “ab” come in before the previous packet finished and ignore the packet that ended up as only “ab12”, and then get itself back on track again. You should use 2 bytes for your header to avoid collisions with the actual data. Adding a checksum at the end of the packet would also allow you to detect bit swap errors. CRC or fletcher16 would work perfectly in this scenario, but in your case a checksum may be more than you need.

Tags: serial (Prev Q) (Next Q), sensors (Prev Q) (Next Q)

Q: How to retrieve the data type of a variable?

Tags: serial (Prev Q) (Next Q), variables (Prev Q) (Next Q)

I am using Arduino and I would like to know if there is a function that returns the data type of a variable. That is, I would like to run something as like the following:

```c
// Note: 'typeof' is a sample function that should return the data type.
Serial.println(typeof(myVar));
```

Tags: serial (Prev Q) (Next Q), variables (Prev Q) (Next Q)

User: user502052

Answer by peter-r.-bloomfield

In a typical C++ program, you would use the typeid operator, like this:

```c
std::cout << typeid(myVar).name();
```

However, that requires a compiler feature called Runtime Type Information (RTTI). It’s disabled in the Arduino IDE, presumably because it tends to increase the runtime memory requirements of the program.

You can get more information about the resource cost here:
http://stackoverflow.com/questions/579887/how-expensive-is-rtti

Any fully-featured C++ compiler will definitely support RTTI though. If you want to try using a 3rd party IDE (such as Eclipse with the Arduino plugin) then you could easily enable it. It’s probably not worth the hassle just for that though.

Alternative

A higher-performing (but less flexible) solution would be to use a traits class approach. This involves some funky template meta-programming:

```c
// Generic catch-all implementation.
template<typename T_ty> struct TypeInfo { static const char * name; };
template<typename T_ty> const char * TypeInfo<T_ty>::name = "unknown";

// Handy macro to make querying stuff easier.
#define TYPE_NAME(var) TypeInfo< typeof(var) >::name

// Handy macro to make defining stuff easier.
```
You can add `MAKE_TYPE_INFO( . . . )` lines for any type you want, including the names of custom classes. You could then use it like this:

```cpp
int myVar = 17;
Serial.println( TYPE_NAME(myVar) );
```

Anything you don’t define using `MAKE_TYPE_INFO( . . . )` will show up as "unknown".

That’s some fairly advanced stuff in there, so I won’t try to explain how it all works here. There are various tutorials on the web though about C++ template programming if you’re interested.

EDIT: It’s worth noting that the `typeof` operator isn’t standard C++, but is supported by a few compilers, such as GCC. It’s basically an older equivalent of `decltype`, which appears in the C++11 standard.

**Q: For which type of serial communication is Arduino Severino designed?**

**Tags:** serial (Prev Q) (Next Q), variables (Prev Q) (Next Q)

Arduino has become quite costly where I come from (2/3 of the price of Raspberry PI) so I have created an [Arduino Severino clone](https://www.arduino-severino.com/).

I had used the device with the RS-232 Usb-Serial converter. But after a (long) while, the device stopped working. My friend came up with a point that RS-232 communication ranges between +6 to -6 V (with thresholds at +3 and -3 volts).

So I have tried FT232 which uses 0 to 5 volts instead. I have a little Chinese [FTDI](https://www.ftdichip.com/) chip designed to do that:
However, after connecting it, it behaves unexpectedly. (it had crashed my PC, one of the communication LEDs is on all the time…) I realised that the Arduino board might get broken for a different reason (an accidental short-circuit is very likely).

So for which type of communication is this “Arduino” designed?

Tags: serial (Prev Q) (Next Q)

User: tomáš-zato

Answer by majenko
Looking at the schematic of that board it uses a very crude signal inversion for the RS-232. This allows it to work of a standard RS-232 signal direct from a computer - just.

It’s far from ideal and really should use a MAX232 (or similar) chip for the interface.

Your FT232 adaptor should bypass that chunk of the circuit and connect direct to pins 0 and 1 of the Arduino clone. The DTR pin of the FT232 should connect to the RESET pin of the Arduino clone via a 100nF capacitor to allow the IDE to reset the board to enter the bootloader. Oh, and of course, connect the ground to your Arduino clone too (something Arduino users often forget).

You can follow the serial breakout board portion of the Arduino breadboard tutorial and apply it to your board.

---

**Answer** by wouter-van-ooijen

The arduino you linked to is designed for RS232-level serial signals, which can be guessed from the use of a DB09 connector.

The FT232 print you show is for 5V TTL (or maybe 3.3V ‘TTL’) signaling. Not compatible with your arduino. (BTW: that chip is not Chinese at all, it is made by www.ftdichip.com IMO they make the best usb-to-serial chip/driver products.)

---

**Answer** by ricardo

The Arduino Severino (Single Sided Serial version 3 - S3V3rino) is designed to work with RS-232 serial interface, not with the TTL serial signal levels of your FT232 board.

So, to use your FT232 board to connect your Severino to a USB port, you’ll need connections like those in the schematics below:
So, basically you need these connections:

<table>
<thead>
<tr>
<th>FTDI</th>
<th>ATmega328</th>
<th>Arduino Severino</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTR</td>
<td>Pin 1 (thru cap C6)</td>
<td>RESET (thru cap C6)</td>
</tr>
<tr>
<td>RXI</td>
<td>Pin 3</td>
<td>TX (D1)</td>
</tr>
<tr>
<td>TXO</td>
<td>Pin 2</td>
<td>RX (D8)</td>
</tr>
<tr>
<td>VCC</td>
<td>Not connected</td>
<td>Not connected</td>
</tr>
<tr>
<td>CTS</td>
<td>Not connected</td>
<td>Not connected</td>
</tr>
<tr>
<td>GND</td>
<td>Pin 8 &amp; 22</td>
<td>Ground</td>
</tr>
</tbody>
</table>

That’s basically the same Majenko suggested (but with a picture added).

I’m not sure you can make these connections directly into the Severino without damaging the board or anything else. You’ll probably need to cut the TX/RX traces that go from the ATmega328 to the DB9 female connector on the Severino and wire them to the FTDI header pins.

Best would be to modify the Severino EagleCAD schematics to remove all the RS-232 circuitry and replace it by TTL connector as shown in my schematics above and, with that, etch and assembly a new board. It’s a lot of work, but can be done.

I hope this helps.
**Q: Can I use an external power supply and still communicate with my computer over Serial?**

Tags: serial (Prev Q) (Next Q), power (Prev Q) (Next Q)

I read the following online: If you plug in both USB power (say from a pc) and external power via the 2.1mm jack, the Arduino chooses power source automatically.

If the external 2.1mm DC is providing greater than 6.6V, the Arduino will take power from it instead of the USB.

Firstly, is this accurate?

Secondly, my real question: If I am powering my Arduino via external DC, can I still communicate with my Raspberry Pi using serial via the USB?

Tags: serial (Prev Q) (Next Q), power (Prev Q) (Next Q)

User: c_breeez

Answer by anonymous-penguin

Yes you can. It still leaves the data/ground lines intact, it just doesn’t bridge the 5V line.

**Arduino Uno Schematic v.3**: 

![Arduino Uno Schematic](image)

The USBVCC is directly from the 5V on the USB jack (well, technically there’s a fuse there). It then goes through a transistor (T1) that only allows power through when there is
no power coming from the 5V regulator (that regulates the VIN voltage).

The USB jack:

You can see pin 1 of the USB jack (5V) goes through the fuse to USBVCC. The next ones D+/D- are the data lines that go through resistors to the ATMEGA16U (used for USB to UART). The last one just ties into ground.

Tags: serial (Prev Q) (Next Q), power (Prev Q) (Next Q)

Q: Serial.begin(): Why not always use 28800?

Tags: serial (Prev Q) (Next Q)

In a lot of the sample code online people add the line `Serial.begin(9600)` in the setup block.

When I look up what `Serial.begin()` is on the official documentation, it says that it controls the bit per second data transfer.

So the obvious question is, why not use 28800, the highest transfer rate? Why do people settle for 9600? What is the limitation here?

Tags: serial (Prev Q) (Next Q)

User: look-behind-you

Answer by jdr5ca

Why do people settle?

People settle because it is more than fast enough. The most common use is just to print some stuff on a terminal for debuggin. 9600 baud is 960 characters per second, or 12 x 80 character lines per second. How fast can you read? :)

If your program is using the serial port for bulk data transfer, you would choose not to settle.
What is the limitation…

The limits on serial are high. Directly you can use 115200 baud in your programs and it will just work. The Arduino terminal will allow a max of 115200, but other programs such as RealTerm would let you run higher.

Hardware serial will run to 1 M baud. If you read around you will see people have used up to 1 M by directly controlling the UART. You might get benefit of high baud rates for uses such as transmitting via a bluetooth chip. If you are using the hardware serial interface to exchange from chip to chip with just a short distance, then 1 M baud is completely feasible. Think of all the SPI and I2C devices that operate just fine at 1 MHz clock rate.

Over larger distances, you will start to have problems with noise when using logic level (plain 0 to 5V) signalling. To use larger distances, you would add a transceiver to provide robust signalling, commonly RS-232 and less commonly RS-485. With RS-232 you could run a mega bit at distances of 10’s of feet.

The microprocessor clock speed will be the real limit. With a hardware UART, the processor must load one byte to the UART every 10 bits (for N81). So when you get to 1 M baud it will be a challenge for the 16 MHz processor to keep the UART supplied with data. A new byte will be sent every 160 clock ticks, which is very few lines of code. For a short burst of data, you might achieve that rate. The message is, the processor will run out of speed before the UART is the limit.

Note, this all applies to HardwareSerial, software serial is very different.

---

**Answer** by sylvain-leroux

In addition to all the interesting answers, it worth mentioning that setting the serial speed to ***XXX*** bits/s does not necessary imply ***XXX*** bits/s on the hardware.

Clocks — even quartz based — are imperfect and subject to drift. In addition, as the serial clock is usually generated through a power-of-two pre-divider and (integer) counter, all value cannot be accurately obtained given a base clock frequency. With the help of the start/stop bits, asynchronous serial communication may be tolerant to some clock drift. But this has limits.

For example, if your ATmega328PA is running at 1MHz, you can achieve 9600b/s at 0.2% of error. But at 14400b/s the error is -3.5% (actually communicating at 13900b/s). And at 28800b/s, the error is +8.5% (actually communicating at 31200b/s). All those figures are from **ATmega48PA-88PA-168PA-328PA datasheet, p200**.

This is *not* an issue when two identical devices communicate together (as there are in fact communicating at the *same* speed). It *might* be an issue when communicating between different devices.

Increasing the base frequency does not necessary improve significantly the accuracy. For example, running the same ATmega328PA as above at 2MHz does not really give better results as those are mostly due to rounding errors. But running it 1.8432MHz give very accurate bps from 2400b/s up to 57.6kHz.
I think it is a kind of tradition to use a transfer rate that is not the slowest one (300) but also not one that could eventually cause problems in some setups (28800 or even 115200). The PC serial port (most often a FTDI232 USB adapter) can cope with higher rates but your DIY hardware may not. So 9600 bps has established itself as some kind of standard transfer rate for code examples.

**Q: How to reset Arduino from software**

As you know, the Arduino IDE resets the Arduino when the Serial Monitor is opened. I want also my program to do it. But I don’t know how.

I examined the source code of the Arduino IDE (Serial.java & SerialMonitor.java), but I can’t find where it sends the reset command. Where is it?

I use C# to communicate with serial.

**Q: Arduino Nano no serial port for MacBook Air 2013**

I have an Arduino Nano connected via a USB cable (Type A to Mini Type B) to my MacBook Air (Mid 2013 Model). The PWR LED is on while the arduino is connected. Opening the Arduino software, pulling down the Tools > Serial Port menu does not show me a valid serial port. All I see is:
What I’ve tried and did not work

- I have installed the correct FTDI driver (v2.2.18). x64 for my MacBook Air’s Corei5, file FTDIUSBSerialDriver_10_4_10_5_10_6_10_7.mpkg. Restarted after installation.
- Tried connecting arduino to both USB ports on my Air.

More Info

- Running Mac OS X 10.10 Yosemite
- Same arduino and cable work when connected to a Windows machine or a Raspberry Pi.

If it helps, I can see the USB Serial in my Apple Logo > About This Mac > System Report under Hardware > USB

```
USB2.0-Serial:
  Product ID:     0x7523
  Vendor ID:     0x1a86
  Version:       2.54
  Speed:         Up to 12 Mb/sec
  Location ID:   0x14100000 / 8
  Current Available (mA): 500
  Current Required (mA):  Unknown (Device has not been configured)
```

I’m out of ideas and options. Any help would be greatly appreciated.

Thanks in advance :)

**Tags:** serial (Prev Q) (Next Q), arduino-nano (Prev Q) (Next Q)

**User:** mekku

---

**Answer** by mekku

This solved it for me.

- Downloaded [this driver](#)
- Installed it
- Ran `sudo nvram boot-args="kext-dev-mode=1"
- Rebooted
- Serial ports are shown in Arduino software and even when I `ls -l /dev/tty.*`

Hopes this helps someone.

Reference: [This thread](#)

---

**Tags:** serial (Prev Q) (Next Q), arduino-nano (Prev Q) (Next Q)

---

**Q:** Is SoftwareSerial left out for the ATTiny85/84?
I want to use SoftwareSerial with my ATTiny85 and when I google it several projects claim that they are using it… I downloaded the hardware from https://code.google.com/p/arduino-tiny/, but when I try to add SoftwareSerial and compile I get:

```
fatal error: SoftwareSerial.h: No such file or directory
```

Switching the board to UNO makes the problem go away, so did they leave it out? How can I fix this?

The link you provided is not hardware, but software called the ATtiny core files. It contains the necessary files that will help you compile and upload code from Arduino IDE to the ATtiny (even when using ArduinoISP as programmer).

The SoftwareSerial is a library that comes with the Arduino IDE zip/installation. If you have downloaded the latest version of Arduino IDE from the Arduino Website, you should be able to find the SoftwareSerial library files located at:

```
[Path to Arduino installation folder]\libraries\SoftwareSerial
```

Example for SoftwareSerial library files you should expect to find there are SoftwareSerial.cpp and SoftwareSerial.h.

```
fatal error: SoftwareSerial.h: No such file or directory
```

The compiler error you receive seems to imply that the compiler cannot find the header file (SoftwareSerial.h) for SoftwareSerial library. This indeed seems quite odd, since this library does not require any special installation, and as mentioned, is part of the libraries that are built into the Arduino IDE installation.

So, to mitigate this, I would recommend that you will:

- Make sure the library folder exists in the location mentioned above
- If it doesn’t, then you should probably re-download the Arduino IDE installation, and after that install the ATtiny core files as per the instructions (important!). Installation instructions are located in the README file inside the ATTiny core zip file (tiny\avr\README).
- If it does exist, and you still get this compiler error, then you must have a corrupted installation (perhaps the compiler include folders is messed up). I would still
recommend following the re-installation as mentioned in the previous bullet.

- Make sure that you can find the **ATtiny 85** in the board list, at Tools->Board and that it is selected.
- Another angle that might shed more light on this error, would be to turn on the verbose output for the compiler (you can do it in the Arduino IDE, by choosing File->Preferences-> “Show verbose output during” and checking “compilation”). Verbose output can help you track down any compiler command line errors, like missing include folders.
- Last but not least, if the SoftwareSerial folder exist, you could use the Arduino IDE to check if the IDE can identify its existance, by going to Sketch->Import Library, and looking for SoftwareSerial. If it’s not there, then the folder is not located properly under the ‘libraries’, verify the path as mentioned above (did you move it?)

Finally, when you will be able to compile your code, try using the SoftwareSerial example that comes with the library to check it. Note that it is not going to work ‘out-of-the-box’ as with the Arduino Uno; The ATtiny does not have the same FTDI hardware for communicating directly with USB, so for sending and receiving serial messages using the ATtiny you will have to either use the Arduino Uno as a proxy (using the Uno’s pins 0 & 1) or to use dedicated USB to Serial hardware similar to [this one](http://www.instructables.com/id/Easy-ATTiny-Serial-Communication-with-Tiny-AVR-Pro/).

Switching the board to UNO makes the problem go away

As a side note, switching to the Uno will make most ATtiny problems ‘go away’ since the Uno is much more equipped than ATtiny85/4, both on the MCU level and on the breakout board level, which you get with the Uno (and which you don’t get when working directly with a chip like ATtiny). Also, most of the code base out there is targeting the Uno and not ATTiny. So if you are doing your first steps with the ATtiny, be prepared that many things that ‘just worked’ with the Uno will require extra/different code before they could work with the ATtiny (and some won’t work at all).

You can use my simple Analog to Serial code:

```
#include <SoftwareSerial.h>
// Definitions
#define rxPin 4
#define txPin 3
SoftwareSerial mySerial(rxPin, txPin);
int sensorPin = 7; //ACTUALLY PIN LABELED AS "2" on the HLT tutorial
int sensorVal = -1;

// the setup routine runs once when you press reset:
void setup() {
  pinMode(sensorPin, INPUT);
  mySerial.begin(9600);
```

```
.Skip_code_block

Takes an input on the ATtiny85 from the VERY TRICKY analogue input pin 2 (as labeled in HLT tutorial) and outputs these with Software serial to the arduino uno, or Duemillanove

ATTINY85 pin 3 -> Arduino Uno pin 0
ATTINY85 pin 4 -> Arduino Uno pin 1

```

/*
analogWrite(0,1); // This is kind a heart beat to show the ATTINY is "alive"
}

// the loop routine runs over and over asensorpingain forever:
void loop() {
    sensorVal = analogRead(sensorPin);
    mySerial.print("Input Val: ");
    mySerial.print(sensorVal);
    mySerial.println();
}

Tags: serial (Prev Q) (Next Q), attiny (Prev Q) (Next Q)

**Q: Serial.write and Serial.println return different values? Cannot use correct ones**

Tags: serial (Prev Q) (Next Q)

I have a text file on an SD card, and I am trying to print the lines to the serial monitor.

This Code Works:
```c
#include <SPI.h>
#include <SD.h>

File myFile;

void setup()
{
    Serial.begin(9600);

    Serial.print("Initializing SD card...");

    pinMode(10, OUTPUT);

    if (!SD.begin(4)) {
        Serial.println("initialization failed!");
        return;
    }
    Serial.println("initialization done.");

    myFile = SD.open("c.txt");
    if (myFile) {
        Serial.println("c.txt:");

        while (myFile.available()) {
            Serial.write(myFile.read()); //THIS LINE RIGHT HERE
        }
        myFile.close();
    } else {
        // if the file didn't open, print an error:
        Serial.println("error opening c.txt");
    }
}

void loop()
{
}
```

However, if I replace the line `Serial.write(myFile.read());` with `Serial.println(myFile.read());`, then the code doesn’t return the correct lines.

It should return numbers similar to this (albeit 3000 of them):
If I use `Serial.println` it returns numbers like this (and it doesn’t stop at 3000 lines):

```
Skip code block
13
10
54
52
13
10
53
54
52
```

And if I try to do `int(myFile.read())` then the integer becomes the wrong numbers (returned by `println`), not the right ones (returned by `write`).

How do I get a number returned by `Serial.write(myFile.read());` to be an integer?

**Tags:** serial (Prev Q) (Next Q)

**User:** user3151828

**Answer** by brettam

The text file contains the sequential ascii values that spell out the data, not binary data. `println` in printing the numerical value of each byte. For example, the 10’s `println` prints are ‘\n’ characters. `write` is printing the ascii character associated with the binary value (the new line). You need to parse the ascii characters into numbers.

Luckily for you the SD library inherits from the `Stream` class, so it will have the same `parseInt` function most often used to parse ints from the serial monitor.

You probably want to set `myFile.setTimeout(0)` first because the file isn’t slowly getting characters from the serial bus, so the timeout would just slow you down.

Then you should be able to parse and print out the ascii integers with something like this:

```
int val = myFile.parseInt();
Serial.println(val);
```

**Tags:** serial (Prev Q) (Next Q)
Q: what is the use of “<<=” operator? □

Tags: serial (Prev Q) (Next Q)

I recently read this piece of code and I don't know what is the use of “<<=” in the for loop:

```
for (mask = 0x01; mask>0; mask <<= 1) {
    if (data & mask) { // choose bit
```

This is the entire code:

```
// Created August 23 2006
// Heather Dewey-Hagborg
// http://www.arduino.cc

#include <ctype.h>
#define bit9600Delay 100
#define halfBit9600Delay 500
#define bit4800Delay 188
#define halfBit4800Delay 94

byte rx = 6;
byte tx = 7;
byte SWval;

void setup() {
    pinMode(rx, INPUT);
    pinMode(tx, OUTPUT);
    digitalWrite(tx, HIGH);
    delay(2);  
    digitalWrite(13, HIGH);  // turn on debugging LED
    SWprint('h');  // debugging hello
    SWprint('i');
    SWprint(10);  // carriage return
}

void SWprint(int data) {
    byte mask;
    // startbit
    digitalWrite(tx, LOW);
    delayMicroseconds(bit9600Delay);
    for (mask = 0x01; mask>0; mask <<= 1) {
        if (data & mask) {  // choose bit
            digitalWrite(tx, HIGH);  // send 1
        } else{
            digitalWrite(tx, LOW);  // send 0
        }
    }
    delayMicroseconds(bit9600Delay);
    // stop bit
    digitalWrite(tx, HIGH);
    delayMicroseconds(bit9600Delay);
}

int SWread() {
    byte val = 0;
    while (digitalRead(rx));
    // wait for start bit
    if (digitalRead(rx) == LOW) {
        delayMicroseconds(halfBit9600Delay);
        for (int offset = 0; offset < 8; offset++) {
            delayMicroseconds(bit9600Delay);
            val |= digitalRead(rx) << offset;
        }
        // wait for stop bit + extra
        delayMicroseconds(bit9600Delay);
```
As stated in the other answers, the <<= operator shifts the byte to the left; however, the number ‘1’ after it specifies how far to shift it - so, 2 <<= 2 == 8. Possibly clearer in binary:

00000001 <<= 1 == 00000010
00000010 <<= 1 == 00000100
00000001 <<= 2 == 00000100

In simpler terms mask <<= 1 is shorthand for mask = mask <<= 1

In general

variable operator= operand

is the augmented assignment operator which means

variable = variable operator operand

<< is the left bit-shift operator. That means it takes the binary contents of a number and moves everything left by a certain number of binary places (1 place in this case). <<= simply combines the left bit-shift with an assignment, so it takes the current value of mask, shifts it left, and then stores the result back in mask.

In this situation, the mask variable starts with a value of 1. The second time round the loop, it will be 10 (binary). Next time, it will be 100, and so on. Since mask is a byte (which is 8 bits), it will go all the way up to 10000000 and then stop. After that, the 1 effectively falls-off the left and mask ends up with a value of 0.

The purpose of this in the code you provided is to test each individual bit within the data parameter. It basically goes through from least- to most-significant and checks if each bit is 1 or 0. It will then pull an output pin HIGH or LOW (respectively) to match. This allows it to send the contents of the variable digitally to some other device.

Doing low-level binary (aka bit-wise) operations like this is very common in microcontroller programming. If you’re not familiar with it then I’d definitely recommend...
looking up some tutorials online. It opens up a huge range of new possibilities for your projects.

**Q: Can you query the current serial port’s speed?**

**Tags:** serial (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

Is there some code I can use to find out what the a chosen serial port’s speed is running at?

**Tags:** serial (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

**User:** hayden-thring

**Answer** by brettam

There is not a top level, easy to use way. Sorry. It is probably easier to just store your chosen baud rate in a variable when you setup the serial port.

Anyway, What you can try to do for the AVR-based, hardware UARTs is undo what Serial.begin does to set up a particular baud rate, but its a bit of a pain. This is the code for setting the baud rate:

```cpp
// Try u2x mode first
uint16_t baud_setting = (F_CPU / 4 / baud - 1) / 2;
*_ucsra = 1 << U2X0;

// hardcoded exception for 57600 for compatibility with the bootloader
// shipped with the Duemilanove and previous boards and the firmware
// on the 8U2 on the Uno and Mega 2560. Also, The baud_setting cannot
// be > 4095, so switch back to non-u2x mode if the baud rate is too
// low.
if ((((F_CPU == 16000000UL) && (baud == 57600)) || (baud_setting >4095))
{
    *_ucsra = 0;
    baud_setting = (F_CPU / 8 / baud - 1) / 2;
}

// assign the baud_setting, a.k.a. ubrr (USART Baud Rate Register)
*_ubrrh = baud_setting >> 8;
*_ubrrl = baud_setting;
```

You can find the results of this by reading the right UCSRA, UBRRH, and UBRL registers. On an uno those are the correct register names, on an mega its UCSRA0, UBRR0H, UBRR0L for Serial, UCSRA1A … for serial1 and so on. Non-avr boards (and Serial on a leonardo) will be totally different.

There will be a single state for those three registers on an AVR hardware serial port for a particular board (and frequency of that board) at each baud rate. You can try and make an equation to get the original baud rate out, but I recommend just comparing the direct values with a lookup of some kind because the integer arithmetic rounding errors will make it a huge mess.

For example, on my mega UBRR0H, UBRR0L, and UCSRA0 are 0, 207, 2 at 9600 baud,
but they are 0, 51, 2 at 38400 baud and 0, 16, 0 at 57600 baud.

---

**Q: Can I make the Arduino ignore serial print**

**Tags:** serial (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

---

I like having serial communication for debugging and testing purposes but after a while it takes away too much speed from the sketch.

Is it possible to have the Arduino ignore serial.print and serial.println throughout my code, without turning it into a comment or placing every serial printing inside for example “if(debug == true)” statements?

Thanks in advance.

**Tags:** serial (Prev Q) (Next Q)

**User:** kevin-kroon

---

**Answer** by fuenfundachtzig

If you insist on top performance, the best thing would be to use a macro for that:

```
#define Sprintln(a) (Serial.println(a))
```

Then instead of

```
Serial.println(F("Hello world!"));
```

write

```
Sprintln(F("Hello world!"));
```

etc. To deactivate the Serial printing, define the macro empty:

```
#define Sprintln(a)
```

This will have the preprocessor remove all debugging code defined with println from your code.

(Of course, there’s a huge number of variations on this theme.)

---

**Answer** by christopher-creutzig

You could, for example, use the preprocessor to change all Serial in your code.

---

**Skip code block**

```cpp
#ifndef ENABLE_PRINT
// disable Serial output
#define Serial SomeOtherwiseUnusedName
static class {
  public:
    void begin(...) {} 
    void print(...) {} 
    void println(...) {}
} Serial;
#undef
```
Q: Put ATmega328 in very deep sleep and listen to serial?

I have investigated the sleep options of the ATmega328, and read a few articles about it, and I would like to understand if there are more options.

So I would like to get as low current as possible, so that anything less that 100uA would be good - as long as I can listen to uart and interrupts for waking up.

I am using a custom PCB (not the UNO), with ATmega328p.

Setting the chip to deep sleep:

```c
set_sleep_mode(SLEEP_MODE_PWR_DOWN);
sleep_enable();
sleep_cpu();
```

would not wake it up with serial communication, according to this.

You will need to put it in IDLE mode, to listen to serial, but this would consume a few mA - bad.

I have found this link where you can connect in hardware the serial to the interrupt - which is dangerous so you can lose data, and moreover, I need these 2 interrupts pins.

I also read this article of Gammon, where you can disable some things, so you can get IDLE sleep with much lower power - but he didn’t mention how exactly you get from this:

```c
power_adc_disable();
power_spi_disable();
power_timer0_disable();
power_timer1_disable();
power_timer2_disable();
power_twi_disable();
```

So, bottom line, is there any option out there, to get less than 0.25mA at least, and also listen to serial port, without any hardware manipulation? For example, waking up with long serial data input?

A board we make does this.

- The RX pin is wired to INT0
- INT0 pin set to input or input pullup depending on how the RX line is driven
- On sleep, INT0 low level interrupt is enabled

```c
//Clear software flag for rx interrupt
rx_interrupt_flag = 0;
//Clear hardware flag for rx interrupt
```
• INTO interrupt service routine sets a flag and disables the interrupt

```c
void rx_interrupt()
{
  detachInterrupt(INT_RX);
  rx_interrupt_flag = 1;
}
```

• On wakeup, we check for the flag (there are other interrupt sources)

On the comms side of things we use a message protocol that has a start character > and end character \r e.g. >setrtc, 2015, 07, 05, 20, 58, 09\r. This give some basic protection against losing messages, as incoming characters are not processed until a > is received. To wake up the device we send a dummy message before transmission. A single character would do it, but we send >wakeup\r hehe.

The device stays awake for 30 seconds after the last message is received in case of new messages. If a new message is receive the 30 second timer is reset. The PC interface software sends a dummy message every second to keep the device awake while the user has it connected for configuration etc.

This method gives absolutely no problems at all. The board with a few peripherals uses about 40uA when sleeping. The actual current consumed by the ATMega328P is probably around 4uA.

**Update**

At look at the datasheet shows that the RX pin is also pin change interrupt pin 16 (PCINT16)

Thus another method without wires may be

• Before sleep: Set the port change interrupt mask bit in PCMSK2 for PCINT16, clear the pin change port 2 flag in PCIFR, enable the pin change port 2 interrupt (PCINT16-PCINT23) by setting PCIE2 in PCICR.

• Setup an ISR for the pin change port 2 interrupt and continue as before.

The only caveat with the port change interrupt is that the interrupt is shared across all the 8 pins that are enabled for that port. So if you have more than one pin change enabled for the port, you have to determine which triggered the interrupt in the ISR. This is not a problem if you are not using any other pin change interrupts on that port (PCINT16-PCINT23 in this case)

Ideally this is how I would have designed our board but what we have works.

---

**Answer** by nick-gammon

The code below achieves what you are asking:

```c
#include <avr/sleep.h>
#include <avr/power.h>
```
const byte AWAKE_LED = 8;
const byte GREEN_LED = 9;
const unsigned long WAIT_TIME = 5000;

ISR (PCINT2_vect)
{
    // handle pin change interrupt for D0 to D7 here
} // end of PCINT2_vect

void setup()
{
    pinMode (GREEN_LED, OUTPUT);
    pinMode (AWAKE_LED, OUTPUT);
    digitalWrite (AWAKE_LED, HIGH);
    Serial.begin (9600);
} // end of setup

unsigned long lastSleep;

void loop()
{
    if (millis () - lastSleep >= WAIT_TIME)
    {
        lastSleep = millis ();
        noInterrupts ();

        byte old_ADCSRA = ADCCRA;
        // disable ADC
        ADCCRA = 0;
        // pin change interrupt (example for D0)
        PCMSK2 |= bit (PCINT16); // want pin 0
        PCIFR |= bit (PCIF2); // clear any outstanding interrupts
        PCICR |= bit (PCIE2); // enable pin change interrupts for D0 to D7

        set_sleep_mode (SLEEP_MODE_PWR_DOWN);
        power_adc_disable();
        power_spi_disable();
        power_timer0_disable();
        power_timer1_disable();
        power_timer2_disable();
        power_twi_disable();
        UCSR0B &= ~bit (RXEN0); // disable receiver
        UCSR0B &= ~bit (TXEN0); // disable transmitter

        sleep_enable();
        digitalWrite (AWAKE_LED, LOW);
        interrupts ();
        sleep_cpu ();
        digitalWrite (AWAKE_LED, HIGH);
        sleep_disable();
        power_all_enable();

        ADCCRA = old_ADCSRA;
        PCICR &= ~bit (PCIE2); // disable pin change interrupts for D0 to D7
        UCSR0B |= bit (RXEN0); // enable receiver
        UCSR0B |= bit (TXEN0); // enable transmitter
    } // end of time to sleep

    if (Serial.available () > 0)
    {
        byte flashes = Serial.read () - '0';
        if (flashes > 0 && flashes < 10)
        {
            // flash LED x times
            for (byte i = 0; i < flashes; i++)
            {
                digitalWrite (GREEN_LED, HIGH);
                delay (200);
                digitalWrite (GREEN_LED, LOW);
                delay (200);
            }
        }
    } // end of if
I used a pin-change interrupt on the Rx pin to notice when serial data arrives. In this test the board goes to sleep if there is no activity after 5 seconds (the “awake” LED goes out). Incoming serial data causes the pin-change interrupt to wake the board. It looks for a number and flashes the “green” LED that number of times.

**Measured current**

Running at 5 V, I measured about 120 nA of current when asleep (0.120 µA).

**Awakening message**

A problem however is that the first arriving byte is lost due to the fact that the serial hardware expects a falling level on Rx (the start bit) which has already arrived by the time it is fully awake.

I suggest (as in geometrikal’s answer) that you first send an “awake” message, and then **pause** for a short time. The pause is to make sure the hardware does not interpret the next byte as part of the awake message. After that it should work fine.

Since this uses a pin-change interrupt no other hardware is required.

---

**Tags:** serial (Prev Q) (Next Q), atmega328 (Prev Q) (Next Q)

---

**Q: Low Power library messing up serial text**

**Tags:** serial (Prev Q) (Next Q), power (Prev Q) (Next Q)

I noticed something unusual using the Low Power library today. When printing text in the loop and using `LowPower.powerDown(SLEEP_1S, ADC_OFF, BOD_OFF);` to sleep for one second, all the text in the loop gets messed up in the Serial Monitor as if you chose the wrong baud rate.

Example code:

```
#include "LowPower.h"

void setup() {
  Serial.begin(9600);
  Serial.println(F("HELLO"));
  Serial.println(F("HELLO"));
}

void loop() {
  Serial.println(F("HELLO"));
  LowPower.powerDown(SLEEP_1S, ADC_OFF, BOD_OFF);
}
```

This code just prints

```
Hello
Hello
Hello
Hello
```
even when using different baud rates. (I used the same baud rate on both the arduino and the monitor). Replacing the low power command with delay(1000); works fine

**Tags:** serial (Prev Q) (Next Q), power (Prev Q) (Next Q)

**User:** qwertz

**Answer** by majenko

The problem is most likely that the system is going to sleep *while* it’s still sending the serial data.

Forcing all the serial data to be sent before you go to sleep should fix the problem (serial data is sent in the background using an interrupt so as to keep sketch slowdown to a minimum):

```cpp
Serial.flush();
LowPower.powerDown(SLEEP_1S, ADC_OFF, BOD_OFF);
```

**Tags:** serial (Prev Q) (Next Q), power (Prev Q) (Next Q)

**Q: HardwareSerial - check for overflow**

**Tags:** serial (Prev Q)

The SoftwareSerial library has a built-in **Overflow** function, but the **Serial** library does not. Is there a simple way to check for a data buffer overflow on a hardware serial port?

**Tags:** serial (Prev Q)

**User:** joel-m.

**Answer** by tisteandii

A look at `HardwareSerial_private.h` shows incoming bytes are always read (interrupt-based), regardless of what your code is doing. So we can’t set up a hardware interrupt to notify us when the chip’s RX register has overflowed, because such an interrupt would depend on overflow flags but these flags never become set because, like I said, bytes are always read. It also shows that while all incoming bytes are read, not all are added to the 64-byte Serial buffer i.e. a byte may be read but if the buffer is full, nothing else happens; it is simply read to clear hardware flags. So the only (direct) way I can think of is to constantly poll `Serial.available()` in `loop()` to check if the number of bytes available for reading is equal to 64. If this condition yields TRUE, then you may safely conclude that the Serial buffer is full and about to overflow. This should suffice:

```cpp
void loop(){
  //your code
  if (Serial.available() == SERIAL_RX_BUFFER_SIZE){ // Just to be general, 64 bytes in most cases
    //buffer is overrun, do what u will
  }
}
```

You could make some changes to `HardwareSerial_private.h` to get something like the
overflow() method of SoftwareSerial. The relevant lines are:

```c
if (i != _rx_buffer_tail) {
    _rx_buffer[_rx_buffer_head] = c;
    _rx_buffer_head = i;
}
```

The snippet above ensures that the current write position is not equal to the tail of the ring buffer (i.e. if we are about to overwrite a character in the buffer), before writing the incoming byte to the buffer. An else block could be added to set some global variable to indicate overflow. This global variable would have to be cleared when Serial.read() is called.

Tags: serial (Prev Q)
C++ is the standard language used to program the Arduino IDE. Most Arduino code online is written in this language.

C++ is one of the most popular coding languages available, being highly cross-platform. However, transferring Arduino code to PC code (and vice versa) is very hard, even when ignoring the fact that your computer can’t run a motor. For example, in C++ code, the first thing to be called is `main`. In Arduino C++, `main` is still called, however, **the Arduino libraries are called, not your sketch.** Those libraries then call `setup` and then `loop` repeatedly after finishing all the initialization tasks they must perform before the Arduino can run the code of your sketch.

It is pronounced “C plus plus.”
Questions

**Q: What are the other IDEs for Arduino?**

Tags: c++ (Prev Q) (Next Q), ide (Next Q)

The basic Arduino IDE lacks a lot of the sophistication present in other IDEs such as code completion, code collapsing, folder organisation, and more. Are there other IDEs that allow programming in C/C++ and improve on these aspects?

Tags: c++ (Prev Q) (Next Q), ide (Next Q)

User: dlj

---

**Answer** by hardik-thaker

There is an [Arduino Eclipse plugin](http)! And Eclipse is an awesome cross-plateform open-source IDE!

Stino is good. Requires [Sublime Text 2](http) which has an indefinite free trial.

For More Go to [The Official Arduino Site](http)

For development on Windows, there is a special edition from Arduino official IDE called [arduino-erw](http), This edition much better the last one because it fixed a lot of lagging and stability issues!

---

**Answer** by anders-r.-bystrup

Expanding on Fake Name’s answer, there is also [Visual Micro’s Arduino plugin](http) for Atmel Studio (built off Visual Studio). It shares the tools, sources and libraries with the Arduino IDE but provides all the code completion and other features of Visual Studio.

---

**Answer** by connor-wolf

You can use [AVR Studio](http) to write programs for Arduinos.

There is a plugin that lets you link against and use the arduino libraries from [within Atmel Studio](http).

If you’ve used Microsoft Visual Studio, Atmel Studio is basically a reskin of Visual studio with different compiler backends that target the ATmega series of devices, so you’ll feel right at home.

---

**Q: Why do I need to include libraries used in other libraries?**

Tags: c++ (Prev Q) (Next Q), library (Prev Q) (Next Q)

I have a simple library which uses another library.
Here is the header:

```c
#ifndef __DERIVEDCLASS_H__
#define __DERIVEDCLASS_H__
#include <HardwareSerial.h>

class DerivedClass {
    private:
        HardwareSerial* serial;
    public:
        DerivedClass();
    }
#endif
```

And the CPP:

```c
#include "DerivedClass.h"
DerivedClass::DerivedClass()
{
}
```

And the sketch:

```c
#include <DerivedClass.h>

void setup()
{
}

void loop()
{
}
```

This works fine.

However, if I change the library header to this:

```c
#ifndef __DERIVEDCLASS_H__
#define __DERIVEDCLASS_H__
#include <SoftwareSerial.h>

class DerivedClass {
    private:
        SoftwareSerial* serial;
    public:
        DerivedClass();
    }
#endif
```

The sketch fails to compile:

```
/Users/andrew/Documents/Arduino/libraries/DerivedClass/DerivedClass.h:9: error: ISO C++ forbids declaration of ‘SoftwareSerial’ with no type
/Users/andrew/Documents/Arduino/libraries/DerivedClass/DerivedClass.h:9: error: expected ‘;’ before ‘*’ token
```

If I change the sketch to:

```c
#include <DerivedClass.h>

void setup()
{
}

void loop()
{
}
```
i.e. including the SoftwareSerial library in the top level sketch, it compiles fine.

If I expand the library out so it actually performs actions, then if it compiles, it works.

Why is this? HardwareSerial is a built-in part of the Arduino, located in `hardware/arduino/cores/arduino` and SoftwareSerial is in `libraries`, but why would the build process differentiate between the two.

**Tags:** c++ ([Prev Q](#)) ([Next Q](#)), library ([Prev Q](#)) ([Next Q](#))

**User:** cybergibbons

---

**Answer** by jfpoilpret

If I remember correctly this is in fact a limitation of the Arduino IDE.

If you use a better IDE (personally I use Eclipse with the Arduino plugin) it works as it should, i.e. if you include a library A that includes another library B, then in the end you won’t get compile-time errors and both B and A will be included in the final binary.

**Tags:** c++ ([Prev Q](#)) ([Next Q](#)), library ([Prev Q](#)) ([Next Q](#))

---

**Q: C++ vs. The Arduino Language?**

**Tags:** c++ ([Prev Q](#)) ([Next Q](#)), arduino-ide ([Prev Q](#)) ([Next Q](#))

What are the advantages of each language when using the Arduino?

I’m thinking this is a good general question, but I’ll add a bit about why I’m asking if anyone wants to give me a tip.

I’m experienced in preprocessed languages like JavaScript, PHP, and have fiddled with languages like Java and Visual Basic. In other words I know programming techniques and both classical and prototypal object orientation, but nothing about communicating directly with hardware.

I’m making an octocopter, and am thinking that an object oriented approach will be the easiest. (The software will have very many features…) However I have never written in C++.

Since this is a Q&A site that’s supposed to help others, only the general question presented at the beginning is of much importance, but I’d appreciate any comments on my situation.

**Tags:** c++ ([Prev Q](#)) ([Next Q](#)), arduino-ide ([Prev Q](#)) ([Next Q](#))
My personal experience as professor (programming, mechatronics) is that if you have previous programming experience and you are aware of concepts as OOP, it is better to go for C/C++. The arduino language is really great for beginners, but have some limitations (e.g. you must have all your files in the same folder). And it is basically a simplification of C/C++ (you can practically copy & paste arduino code to a C/C++ file, and it will work). Also it makes sense that you can go and use a full well known IDE as eclipse:

http://playground.arduino.cc/Code/Eclipse

Initially it is required a bit more of setup and configuration of your dev environment, but IMHO it is worth it for programmers with experience in any other language.

In any case, it won’t harm you to start using the arduino language and the arduino IDE for a few days to get familiar with the arduino hardware and then move to C/C++ with Eclipse for really developing your project.

In theory...
There isn’t really an Arduino language as such. It’s really just C++ with some domain-specific libraries. These add on various features, such as functions you can call to control the hardware. If you didn’t have those functions, you’d need to fiddle directly with special registers to control everything. That’s how embedded programming is usually done. It’s fast, but it can be quite hard to learn and understand.

In addition to the functions, the libraries add alternative names for some types. For example, `boolean` and `byte` are not in the C++ standard. However, they are directly equivalent to `bool` and `unsigned char`.

All of these things mean you can probably port general C++ code directly to Arduino without difficulty. However, going back the other way may require some minor editing.

In practice...
Having said all of that, programming for Arduino isn’t exactly the same as general C++ programming. A lot of the differences are common to all embedded programming though (such as limited memory and processing power).

It’s also worth noting that if you’re using the official Arduino IDE then there are all sorts of annoying quirks and limitations on how you setup your code. There are workarounds in all cases (as far as I’m aware), but they are sometimes quite frustrating.

For full flexibility, use a third-party IDE (such as Eclipse) with a plug-in to support Arduino. That should give you all the advantages of C++, along with the Arduino libraries.

What are the advantages of C++ vs the Arduino language when using Arduino? I’m
experienced in preprocessed languages like JavaScript, PHP, and have fiddled with languages like Java and Visual Basic.

First, the Arduino compiler/IDE accepts C and C++ as-is. In fact many of the libraries are written in C++. Much of the underlying system is not object oriented, but it could be. Thus, “The arduino language” is C++ or C.

C++ is not garbage collected. It does manage variables in scope - if you write:

```c
Skip code block
int led = 13;
void blinkTimes(int value)
{
    int i;
    for(i=0;i<value;i++)
    {
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        delay(1000);
    }
}
```

Then you’ll find that led and i don’t grow or leak, no matter how many times you call blinkTimes.

If i were a class, it would similarly be disposed of once the function ended. So as long as you aren’t using new or similar memory allocation functions to create new objects, then you won’t have to worry about leaks.

You may still run out of memory, if you create huge classes and use a lot of them in deeply nested functions, but in general you aren’t going to run into trouble until you start dealing with new and free functions.

If you are using new, then you’ll have to call delete at appropriate times. C++, and by extension Arduino, has no automatic garbage collection, you have to explicitly manage your own memory.

---

**Tags:** c++ (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q)

---

**Q:** How can I convert Arduino String to C string type?

**Tags:** c++ (Prev Q) (Next Q)

I got a String which as I understand is an Arduino object, and got some C++ code:

```c
Skip code block
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <string.h>

LiquidCrystal_I2C lcd(0x20,16,2);
boolean borrar = false;
String IP;
void setup()
```

{  
    lcd.init();
    lcd.backlight();
    pinMode(13, OUTPUT);
    Serial.begin(9600);
    Serial1.begin(9600);
}

void loop() {
    while (Serial1.available()) {
        char caracter = Serial1.read(); //Comprobamos el caracter
        switch(caracter) {
            default:
                if (borrar) {
                    IP = "";
                    lcd.clear();
                }
                lcd.print(caracter);
                delay(125);
                borrar = false;
                IP.concat(caracter);
                break;
            case '\r':
            case 0x0F:
            case 0x0A:
                String res = "";
                borrar = true;
                int num = atoi(IP.c_str());
                if (num < 127)
                    res="Clase A";
                if (num == 127)
                    res="Direccion reservada";
                if (num > 127 && num < 192)
                    res="Clase B ";
                if (num >= 192 && num < 224)
                    res="Clase C ";
                if (num >= 224 && num < 240)
                    res="Clase D ";
                if (num >= 240 && num < 255)
                    res="Clase E ";
                break;
        }
    } //fin switch
} //fin programa

} //fin programa

However, this won’t compile because of this line:

int num = atoi(IP.c_str())

As IP is a String and such method works for string. How can I make it compatible (convert it)?

Tags: c++ (Prev Q) (Next Q)

User: diegoaguilar

Answer by jfpoilpret

Your code could be improved by removing the use of IP string altogether, and directly
calculating its numeric value while characters come in through Serial1:

### Skip code block

```cpp
boolean borrar = false;
int IP = 0;
...

void loop() {
  while (Serial1.available()) {
    char caracter = Serial1.read(); //Comprobamos el caracter
    switch(caracter) {
      // NOTE it is better to replace default by the list of all digits...
      case '0':
      case '1':
      case '2':
      case '3':
      case '4':
      case '5':
      case '6':
      case '7':
      case '8':
      case '9':
        if (borrar) {
          IP = 0;
          lcd.clear();
        }
        lcd.print(caracter);
        delay(125);
        borrar = false;
        IP *= 10;
        IP += (int)(caracter - '0');
        break;
    }
  //fin switch
  }
//serial disponible
}
```

This way would bring you 2 advantages:

1. a bit faster than working with String
2. no dynamic memory allocation/deallocation (String does a lot of these) which might lead your program to heap fragmentation and eventually crash.

Note that I have not further refactored your code as I guessed it was just a snippet, not the
complete code for your program. Otherwise, I would have performed further refinement like:

- remove num variable since it is the same as IP now
- replace res from String to const char* (to further reduce heap fragmentation due to String usage)

Answer by rmi

Try

```c
unsigned char z[100];
IP.getBytes(z, 100);
z[IP.length()] = 0;
int n = atoi(z);
```

To retrieve bytes inside the string. This assumes IP string’s length is shorter than 100.

Answer by peter-r.-bloomfield

Try updating to the latest version of the Arduino libraries/IDE.
The `c_str()` method was introduced to the Arduino String class quite recently, I believe. That line of code certainly works fine for me on Arduino IDE 1.0.5.

Tags: c++ (Prev Q) (Next Q)

Q: Arduino `sizeof Servo` array objects is.. wrong?

Tags: c++ (Prev Q) (Next Q)

After searching for a quite long while over the internet, I have no choice but to try asking someone if they can explain me this apparently strange situation.

I’m doing some tests using some servo motors, trying to move them almost together using `millis()` and Servo object.

It is, of course, working either for single Servos and multiple servos.

The case, more detailed, is about this:

1. Move one, two or more servo motors together.
2. Force them to take ALMOST the same time to accomplish any action.
3. Have a dynamic number of servos.

About the first two points I had no problems to solve them, I was easily able to make a function and then make a library to accomplish such a job.

However, since I’m NOT that used to C++ and I was wondering if it actually was possible to create an array of Servo objects, like you can actually do, on arduino, with strings:

```c
String stringArray[] = {"string1", "string2", "and so on, I love it."};
```
So, after wondering a while, I’ve tried this:

```cpp
Servo servObject;
Servo servObject_2;
Servo servObject_3;
Servo servos[] = {servObject, servObject_2, servObject_3};
```

And you know what’s cool? my function is actually working correctly by forcing the amount of elements of the servos array.

What I mean by that is that if I force my function to know that it is going to have 3 elements into the array it will, of course, work; However, for some reason, if I use `sizeof(servos);` the value returned, surprisingly, is 2…. Moreover, if I only push two servo objects instead of three into the servo array, its `sizeof` returns 6.

By googling, I’ve found out a couple of discussions about the fact that, for some reasons, in some cases, `sizeof` returns an incorrect value (if I’m not wrong it was the correct value -1), but in my case it apparently is not following any logic at all.

The board I’m using is an Arduino nano w/ ATMEGA 328.

Any idea of why it is not returning the correct value? Am I actually going totally wrong because I shouldn’t make an array of servo objects?

ps: I didn’t post the whole code because it would’ve been useless, since the `sizeof` is the first function called in the `loop()`

Regards,

briosheje

**Tags:** c++ *(Prev Q)* *(Next Q)*

**User:** briosheje

---

**Answer** by jfpoilpret

What you describe seems to be a common issue that many C or C++ developers can get (or have got) with `sizeof`.

Take a look at the following code:

```cpp
char a[3];
int size(char b[]) {  
    return sizeof b;
}

void setup() {  
    int size1 = size(a);  
    int size2 = size(a);  
}
```

In that code, `size1` will be evaluated, by the compiler, to 3 which is the real size of a variable, whereas `size2` will be 2.

**Why so?**
This is just because sizeof a is evaluated by the compiler, and the compiler knows that a is an array of 3 char; but when a is passed as an argument to function size(), it is converted to a pointer (to the first element of the array), hence its size is the size of a pointer, which is 2 bytes on AVR ATmega MCU.

This subtlety of sizeof may explain strange values as you describe.

**How to fix the issue?**

Most C functions that take an array as argument will require another argument that tells the size of the array; that’s the way (the only one I think) to deal with this problem.

Here is a sample code:

```c
Skip code block
Servo allServos[] = {...};
#define NUM_SERVOS (sizeof(allServos) / sizeof(Servo))
...
void handleServos(Servo *servos, int numServos) {
    for (int i = 0; i < numServos; i++) {
        servos[i].write(...);
    }
}
void loop() {
    handleServos(allServos, NUM_SERVOS);
}
```

It is common practice in C to **declare an array** and immediately after, **define its size** as above.

**Tags: c++ (Prev Q) (Next Q)**
Q: Advisability of Using Bit Fields in Structs

Tags: c++ (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

I have to track a large amount of data (for an Arduino) in a program while taking care of a fair amount of other business.

I started with a struct like this:

```c
struct MyStruct
{
  // note: these names might as well be foo bar baz
  uint8_t color;
  boolean state;
  uint8_t area;
  uint8_t number;
  uint8_t len;
};
```

I have an array of 800 of these. At 5 bytes each, that’s 4Kb, or half the ram on the Arduino Mega.

I changed to a struct like this:

```c
struct MyStruct
{
  uint8_t color : 3; // max value 7
  // uint8_t state : 1; *** moved, thanks Ignacio ***
  uint8_t area : 5; // m.v. 31
  uint8_t number;
  uint8_t len : 5; // m.v. 31
  uint8_t state : 1; // m.v. 1
};
```

I understand this reduces the size of each instance to 3 bytes, for a total of 2.4Kb in my array.

I have read that in some implementations / chipsets using bit fields in structs can lead to less efficient execution. Obviously this is a more efficient use of memory, but my question is:

How does the Arduino handle bit fields? Will it be better, worse, or negligibly different in terms or speed when iterating through an array of bit field structs? Is there a good way to test this?

Tags: c++ (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

User: anthropomo

Answer by jdr5ca

What you are facing is the classic time-memory tradeoff. The bit fields will be smaller in memory, but will take more time to operate on. You can count that no matter what processor, the bit fields will be slower.

You use the word efficient, but that word has no certain meaning without a metric for what is good or bad. When you only have 8 k of RAM, using memory is bad, time may be cheap. If you have real time constraints, then using time is bad and memory may be cheap. In general, you can only buy your way out of this tradeoff. In other words, when you find
both time and memory bad, spend cash and use a bigger chip. There is no single answer for what is good or bad. This is part of why there are so many choices for microcontrollers, people fit the chip to the application and application to the chip.

Populating the bits of a bit field will be slower than filling of complete bytes. Take for example

```c
x = 5;
...
assimplestruct.len = x;
// vs
abitfield.len = x;
```

The first simple case will just:

- load value of `x` to a register
- store it to the byte for `len`

The second does something like:

- loads the current value of `abitfield`
- loads a mask
- clears the bits for `len`
- loads current value of `x`
- loads a mask
- clears off unused bits of `x`
- shifts the bits of `x`
- or's `x` with the `abitfield`
- stores the current value back to memory

If all your operations are packing data into the bit field, or unpacking out of the bitfield, you should expect slower execution. Bit fields are a type of compression - they cost ticks. But moving around bit fields will be faster because there are less bytes to load and store. If you where sorting this array, the smaller byte count could be an advantage. If you where transferring them over a serial port, the compressed size of the bitfield might be a winner.

So with respect to your question:

```
Is there a good way to test this?
```

The only good way to test this is to write test cases for both approaches using a pattern that closely matches your application. It really matters what mix of operations you perform to decide whether the difference is negligible or significant.

When doing this sort of optimization experiment, definitely use source control on your project. You can create a local GIT or Mercurial repository with just a few clicks. Keeping a chain of checkpoints allows you to tear up your code exploring the effects of different implementations. If you take a wrong turn, the repository allows you to simply go back to the last good point and try another path.

(Side note: This time memory tradeoff exists in the opposite direction also. If you look
through the compiler options for desktop class processors, you will find something called
structure packing. This option allow you to add blank bytes between single byte fields
such that they stay aligned at word or double word boundaries. This may seem crazy to
throw away RAM on purpose, but on processors with 16 or 32 bit wide registers and
buses, so called word or dword aligned memory operations can be faster than byte wise
operations.)

---

**Answer** by jrobert

A discussion of bit-fields wouldn’t be complete without mentioning that:

Structures with bit-fields may be used as a portable way of attempting to reduce the
storage required for a structure (with the probable cost of increasing the instruction
space, and time, needed to access the fields), or as a non-portable way to describe a
storage layout known at the bit level.


Other posters, including yourself, have addressed the space/time tradeoff. But what is
sometimes missed (possibly not relevant to your application - ?) is the complete
implementation-dependence of the storage layout. This matters when applications share
this data with another process: Reading or writing hardware, whose register bit-
assignments are necessarily fixed; communicating the data to another system (via network
or storage device doesn’t matter); or to another application on the same system, that may
have been compiled with a different compiler or version of the compiler.

---

**Tags:** c++ (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

---

**Q:** How can I pass an object as a parameter to a function?

**Tags:** c++ (Prev Q) (Next Q)

I am making a tiny game, and want to store high-scores. I made a Highscore-class in the
top of the sketch file, like this:

```cpp
class Highscore {
public:
  String name;
  int score;

  String toString() {
    return this->name + " - " + (String)score;
  }

  Highscore(String name, int score) {
    this->name = name;
    this->score = score;
  }
};
```

Now. Near the bottom of the sketch, I have a function I want to take a Highscore-object as
a parameter, but the compiler will not let me. Example:

```cpp
void writeHighscore(Highscore hs) {
    File f = SD.open("high.txt", FILE_WRITE);
    // Do stuff
}
```

The compilation error is as follows:

```
sketch.ino:19:21: error: variable or field ‘writeHighscore’ declared void
sketch.ino:19:21: error: ‘Highscore’ was not declared in this scope
sketch.ino: In function ‘void setup()’:
sketch.ino:64:20: error: ‘writeHighscore’ was not declared in this scope
```

After changing parameter type to String, int or whatever, all the errors go away. I am able to create Highscore-objects INSIDE the `writeHighscore()` function, but how can I pass objects to it?

**Edit 1 - Object creation**

```cpp
Highscore hs = Highscore("John", 975);
Serial.println(hs.toString()); // Prints "John - 975" as expected
writeHighscore(hs); // Error
```

**Tags:** c++  (Prev Q)  (Next Q)

**User:** krystah

---

**Answer** by brettam

You are doing everything right, but you need to move the class outside of your .ino file for this to work. I’m not sure why the arduino IDE makes you do this (see edit), but it is something I’ve run into before.

I just tested this, and moving the class declaration outside the ino and into a header file in the project folder makes it compile correctly. Just remember to include the header you make in the ino and to `#include "Arduino.h"` in the header to get access to “String” again.

Technically, only a forward declaration of the class needs to be in a different file.

**Edit:**

I’m relatively sure what is happening is the IDE is generating forward declarations for all your functions but not the class, which are placed before your class definition. This means that breaking the function regex the IDE uses with a dummy “throw()” fixes your problem. for example:

```cpp
void writeHighscore(Highscore hs) throw() { ... }
```

should work just fine (it worked for me)

---

**Tags:** c++  (Prev Q)  (Next Q)

---

**Q: Classes and objects: how many and which file types do I actually need**
I have no previous experience with C++ or C, but know how to program C# and am learning Arduino. I just want to organize my sketches and am quite comfortable with the Arduino language even with its limitations, but I really would like to have an object-oriented approach to my Arduino programming.

So I have seen that you can have the following ways (not exhaustive list) to organize code:

1. A single .ino file;
2. Multiple .ino files in the same folder (what the IDE calls and displays like “tabs”);
3. An .ino file with an included .h and .cpp file in the same folder;
4. Same as above, but the files are an installed library inside Arduino program folder.

I have also heard of the following ways, but have not got them working yet:

- Declaring a C++-style class in the same, single .ino file (have heard of, but never seen working - is that even possible?);
- [preferred approach] Including a .cpp file where a class is declared, but without using a .h file (would like this approach, should that work?);

Note that I only want to use classes so that code is more partitioned, my applications should be very simple, only involving buttons, leds and buzzers mostly.

How the IDE organizes things

First thing, this is how the IDE organizes your “sketch”:

- The main .ino file is the one of the same name as the folder it is in. So, for foobar.ino in foobar folder - the main file is foobar.ino.
- Any other .ino files in that folder are concatenated together, in alphabetic order, at the end of the main file (regardless of where the main file is, alphabetically).
- This concatenated file becomes a .cpp file (eg. foobar.cpp) - it is placed in a temporary compilation folder.
- The preprocessor “helpfully” generates function prototypes for functions it finds in that file.
- The main file is scanned for #include <libraryname> directives. This triggers the IDE to also copy all relevant files from each (mentioned) library into the temporary folder, and generate instructions to compile them.
- Any .c, .cpp or .asm files in the sketch folder are added to the build process as
separate compilation units (that is, they are compiled in the usual way as separate files)

- Any .h files are also copied into the temporary compilation folder, so they can be referred to by your .c or .cpp files.
- The compiler adds into the build process standard files (like main.cpp)
- The build process then compiles all the above files into object files.
- If the compilation phase succeeds they are linked together along with the AVR standard libraries (eg. giving you strcpy etc.)

A side-effect of all this is that you can consider the main sketch (the .ino files) to be C++ to all intents and purposes. The function prototype generation however can lead to obscure error messages if you are not careful.

### Avoiding the pre-processor quirks

The simplest way of avoiding these idiosyncrasies is to leave your main sketch blank (and not use any other .ino files). Then make another tab (a .cpp file) and put your stuff into it like this:

```cpp
#include <Arduino.h>

// put your sketch here...

void setup ()
{
    // end of setup
}

void loop ()
{
    // end of loop
}
```

Note that you need to include Arduino.h. The IDE does that automatically for the main sketch, but for other compilation units, you have to do it. Otherwise it won’t know about things like String, the hardware registers, etc.

### Avoiding the setup/main paradigm

You don’t have to run with the setup/loop concept. For example, your .cpp file can be:

```cpp
#include <Arduino.h>

int main ()
{
    init (); // initialize timers
    Serial.begin (115200);
    Serial.println ("Hello, world");
    Serial.flush (); // let serial printing finish
} // end of main
```

### Force library inclusion
If you run with the “empty sketch” concept you still need to include libraries used elsewhere in the project, for example in your main .ino file:

```c
#include <Wire.h>
#include <SPI.h>
#include <EEPROM.h>
```

This is because the IDE only scans the main file for library usage. Effectively you can consider the main file as a “project” file which nominates which external libraries are in use.

### Naming issues

- Don’t name your main sketch “main.cpp” - the IDE includes its own main.cpp so you will have a duplicate if you do that.
- Don’t name your .cpp file with the same name as your main .ino file. Since the .ino file effectively becomes a .cpp file this also would give you a name clash.

### Declaring a C++-style class in the same, single .ino file (have heard of, but never seen working - is that even possible?)

Yes, this compiles OK:

```c
class foo {
  public:
  
  foo bar;

  void setup () {}
  void loop () {}
```

However you are probably best off to follow normal practice: Put your declarations in .h files and your definitions (implementations) in .cpp (or .c) files.

### Why “probably”?

As my example shows you can put everything together in one file. For larger projects it is better to be more organized. Eventually you get to the stage in a medium to large-size project where you want to separate out things into “black boxes” - that is, a class that does one thing, does it well, is tested, and is self-contained (as far as possible).

If this class is then used in multiple other files in your project this is where the separate .h and .cpp files come into play.

- The .h file declares the class - that is, it provides enough detail for other files to know what it does, what functions it has, and how they are called.
- The .cpp file defines (implements) the class - that is, it actually provides the functions, and static class members, that make the class do its thing. Since you only want to implement it once, this is in a separate file.
The .h file is what gets included into other files. The .cpp file is compiled once by the IDE to implement the class functions.

Libraries

If you follow this paradigm, then you are ready to move the entire class (the .h and .cpp files) into a library very easily. Then it can be shared between multiple projects. All that is required is to make a folder (eg. myLibrary) and put the .h and .cpp files into it (eg. myLibrary.h and myLibrary.cpp) and then put this folder inside your libraries folder in the folder where your sketches are kept (the sketchbook folder).

Restart the IDE and it now knows about this library. This is really trivially simple, and now you can share this library over multiple projects. I do this a lot.

A bit more detail here.

My advice is to stick to the typical C++ way of doing things: separate interface and implementation into .h and .cpp files for each class.

There are a few catches:

- you need at least one .ino file - I use a symlink to the .cpp file where I instantiate the classes.
- you must provide the callbacks that the Arduino environment expects (setup, loop, etc.)
- in some cases you will be surprised by the non-standard weird things that differentiate the Arduino IDE from a normal one, like automatic inclusion of certain libraries, but not others.

Or, you could ditch the Arduino IDE and try with Eclipse. As i mentioned, some of the things that are supposed to help beginners, tend to come in the way of more experienced developers.

I’m posting an answer just for completeness, after finding out and testing a way of declaring and implementing a class in the same .cpp file, without using a header. So, regarding the exact phrasing of my question “how many file types do I need to use classes”, the present answer uses two files: one .ino with an include, setup and loop, and the .cpp containing the whole (rather minimalistic) class, representing the turning signals of a toy vehicle.

Blinker.ino

```
#include <TurnSignals.cpp>

TurnSignals turnSignals(2, 4, 8);

void setup() { }
```
void loop() {
    turnSignals.run();
}

#include "Arduino.h"

class TurnSignals
{
    int _left,
    _right,
    _buzzer;

    const int
    amberPeriod = 300,
    beepInFrequency = 600,
    beepOutFrequency = 500,
    beepDuration = 20;

    boolean
    lightsOn = false;

    public : TurnSignals(int leftPin, int rightPin, int buzzerPin)
    {
        _left = leftPin;
        _right = rightPin;
        _buzzer = buzzerPin;

        pinMode(_left, OUTPUT);
        pinMode(_right, OUTPUT);
        pinMode(_buzzer, OUTPUT);
    }

    public : void run()
    {
        blinkAll();
    }

    void blinkAll()
    {
        static long lastMillis = 0;
        long currentMillis = millis();
        long elapsed = currentMillis - lastMillis;
        if (elapsed > amberPeriod) {
            if (lightsOn)
                turnLightsOff();
            else
                turnLightsOn();
        } else
        {
            lastMillis = currentMillis;
        }
    }

    void turnLightsOn()
    {
        tone(_buzzer, beepInFrequency, beepDuration);
        digitalWrite(_left, HIGH);
        digitalWrite(_right, HIGH);
        lightsOn = true;
    }

    void turnLightsOff()
    {
        tone(_buzzer, beepOutFrequency, beepDuration);
        digitalWrite(_left, LOW);
        digitalWrite(_right, LOW);
        lightsOn = false;
    }
Q: Why does this code execute?

After experiencing failures of my Arduino projects due to low memory, I decided to do some research into it so I could understand better where the problems were. I eventually came to this code:

```c
void setup()
{
  pinMode(13, OUTPUT);
  int len = 5000;
  byte *data = (byte *)malloc(len * sizeof(*data));
}

void loop()
{
  int timing = 1000;
  delay(timing);
  digitalWrite(13, HIGH);
  delay(timing);
  digitalWrite(13, LOW);
}
```

I expected that since my Arduino Uno does not have enough RAM to hold an array of 5000 bytes (Atmel’s information on the Atmega328P shows us it only has 2KB of SRAM), the code in `void loop()` would not be able to run due to lack of memory. As far as my understanding goes, once `malloc` has allocated 5000 bytes (or as many as it could of the 5000 bytes), there would physically not be enough space left in memory for the variable `timing` and the LED I attached on pin 13 would not flash on and off at intervals of 1 second.

However, my LED toggles at perfect intervals of 1 second. Why would this occur? Isn’t memory allocated by `malloc` unavailable for use by anything else until `free` is called on it?

Tags: c++ (Prev Q) (Next Q), compile (Prev Q) (Next Q), memory-usage (Prev Q) (Next Q)

User: mc-δt

Answer by nick-gammon

If you add a debugging print you will see what is happening:

```c
void setup()
{
  Serial.begin (115200);
  Serial.println ();
  pinMode(13, OUTPUT);

  int len = 5000;
  byte *data = (byte *)malloc(len * sizeof(*data));
  Serial.print ("data = ");
}
The malloc failed, it returned NULL, the rest of the program proceeded normally.

Once malloc has allocated 5000 bytes (or as many as it could of the 5000 bytes) malloc does not return with a partial allocation. It either allocates the amount you requested or none at all.

**Tags:** c++ (Prev Q) (Next Q), compile (Prev Q) (Next Q), memory-usage (Prev Q) (Next Q)

**Q:** Works with gcc, not with Arduino. error: taking address of temporary array

**Tags:** c++ (Prev Q)

I need to hard code 8 byte addresses into char arrays of length 8. This must be done many places in my code (in function scope), so I have tried to come up with a one-liner. The following works perfectly in C when compiling with gcc.

```
Skip code block
char a[8];
void cpaddr(char target[], char *source) {
    int i;
    for (i=0; i<8; i++)
        target[i] = source[i];
}
int main() {
    char b[] = {0x00, 0x10, 0xFF, 0xCA, 0x00, 0x00, 0xA2, 0x7D};
    cpaddr(a, b);
    // line below does not compile with Arduino IDE
    cpaddr(a, (char[]){0x00, 0x10, 0xFF, 0xCA, 0x00, 0x00, 0xA2, 0x7D});
}
```

When compiling on Arduino the last line, which is the one-liner I was aiming for, does not compile. It gives:

```
/home/bob/Desktop/ate/Ate.ino: In function 'int main()':
Ate:101: error: taking address of temporary array
cpaddr(a, (char[]){0x00, 0x10, 0xFF, 0xCA, 0x00, 0x00, 0xA2, 0x7D});
^
exit status 1
taking address of temporary array
```

What is the problem here, that apparently is not a problem with gcc?

How do I fix it?

**Tags:** c++ (Prev Q)
It is quite right, using that kind of syntax is not allowed. It’s a bit of a pain, but it’s ok since there is an alternative method - kind of a “trick” if you will.

That trick is to use a string, not an array. After all, a string is just an array, it’s just handled slightly differently by the compiler.

Instead of using {...} use "..." and use the hexadecimal character escape sequence \xNN, such as:

```c
cpaddr(a, "\x00\x10\xFF\xCA\x00\x00\xA2\x7D");
```

You could even lose your custom function and use a standard library function - memcpy():

```c
memcpy(a, "\x00\x10\xFF\xCA\x00\x00\xA2\x7D", 8);
```

On the 8-bit AVRs you can save RAM by using the progmem variant and the F() macro:

```c
memcpy_P(a, F("\x00\x10\xFF\xCA\x00\x00\xA2\x7D"), 8);
```

See [Using array init list as temporary in C++11?](#)

You can solve it by using const. This compiles:

```c
char a[8];

void cpaddr(char target[], const byte *source) {
    int i;
    for (i=0; i<8; i++)
        target[i] = source[i];
}

int main() {
    byte b[] = {0x00, 0x10, 0xFF, 0xCA, 0x00, 0x00, 0xA2, 0x7D};
    cpaddr(a, b);
    cpaddr(a, (const byte[]) {0x00, 0x10, 0xFF, 0xCA, 0x00, 0x00, 0xA2, 0x7D});
}
```

Note I had to change your array from char to byte because I was getting (valid) warnings that things like 0xCA don’t fit into a char.
Sensors

Questions

**Q: Best way to determine if a 5 gallon water jug is getting empty**

Tags: sensors (Prev Q) (Next Q)

I have an espresso machine that gets its water from one of those 5 gallon blue water bottles (I would plumb it, but our water here is WAAAAY too hard).

If I ever accidentally let the bottle run out of water, the rotary pump in the espresso machine will be ruined, which would be a costly mistake.

I know that I can install some sort of magnetic float sensor inside the water bottle, but since I have to change it out once every few weeks, having another thing inside the bottle would make changing it out harder, and microbial contamination more likely. The pump that pumps water out of the bottle forms a tight seal at the lid, and anything running under it or through it would compromise that seal.

Is there something I could use to detect if the water level is low from the outside of the bottle? Like a laser sensor?

Tags: sensors (Prev Q) (Next Q)

User: synic

---

**Answer** by jlbnjmn

**Using a Force Sensor to Measure a Quantity of Water**

To satisfy your requirement of having the sensor on the exterior of the water container, I would recommend using a force sensor to weigh the bottle at set intervals. A force sensor, such as the FlexiForce A401, is a type of resistor that changes resistance in response to the amount of force applied. While not as accurate as a scale, it should provide enough accuracy (+-3% according to the datasheet) for the application described.

The basic procedure would be to hook up the force sensor to the Arduino and first measure the resistance with an empty bottle, then measure again with a full bottle. This will give you an idea of the force that your specific platform is actually applying to the sensor, and you can then fine tune the voltage (as per the datasheet) for your installation.

A basic tutorial using a similar force sensor with an Arduino can be found [here](#).

**Other Possibilities**

- Optical Liquid Level Sensors
- Ultrasonic Level Monitoring
- Thermistors and Heating Coils at various levels to Measure the Thermal Mass (i.e.
Q: Sensor for underground water tank level

Tags: sensors (Prev Q) (Next Q)

In a similar way to the question “best way to determine if a 5 gallon water jug is getting empty“, I’m looking for a way to measure water in a container. This is so it can be connected to an Arduino for logging and reporting. But it’s a very different container.

I have a home with the household water supplied from an underground concrete tank (around 3000 US gallons, 11000 litres). Although there is a float switch cutout when it gets empty (to protect the pump), I’d like to know it’s getting low long before that (as it takes a while before a tanker delivers a fresh load). Ideally something that measures exactly or in steps (at least in eights of a load), so software can predict when it’ll run out based on average usage, and it can be monitored online when away.

The tank has a heavy concrete hatch, and a small hole that has an existing primitive dipstick, so access is limited. It’s in a harsh desert environment, but the pump room and power are only a few feet away.

Some ideas I had:

- suspend an array of float switches with strings of different lengths from a bar inside the tank, and wire them all back to a controller. Hard to install, ugly, low tech. Or something similar on a vertical pole.
- contacts (like a US 2 pin plug) on a vertical pole that again are connected back to a
controller (probably using a ribbon cable). Corrosion/anodes might be a problem, although power could just be a short pulse at times. Condensation, spiders webs and dead bugs may give false readings.

- run a fully insulated cable vertically down it, and use as an antenna to send a pulse through, and use the difference between how it works as an antenna in air, and in water, (like an SWR meter) to calculate how far down the water is.
- ultrasonic (or light?) distance finder mounted under the hatch. Range needs to be up to 2 metres (6 feet).
- depth finder (sonar) on a float. Biggest challenge here is getting the signal out (long cable might get tangled when the tank fills).

The last two would be even better if they were wireless (to save running a cable that might get damaged) but I don’t think I’d get the battery life I’d like (1 year or more).

Any other ideas? Has anybody ever done this?

Tags: sensors (Prev Q) (Next Q)
User: rob-hoare

---

**Answer** by cybergibbons

I have a background working on ships where robust and workable tank level gauges are a huge part of instrumentation available to you. Water is the most forgiving and easiest liquid to measure - it’s cool, not too viscous, not corrosive in itself, and easy to clean off. There are a lot of options available to you.

Personally I would avoid any system which involves immersing anything in liquid if at all possible. Waterproofing isn’t easy. I would also try to stay with a tried and tested system.

So to run through your solutions:

1. Multiple float switches - has poor resolution. Would be reliable. You can buy assemblies with LowLow, LowHigh, HighLow, HighHigh float limit switches already on them for maintaining tanks levels, but these are industrial quality with industrial prices.

2. This would work, but corrosion is going to be an issue. Resolution limited by how many contacts you have.

3. Interesting idea, but would likely require a lot of DSP work. Not a tried and tested method.

4. An ultrasonic sensor would be my preferred method. Non-contact, high resolution, range of 2m is easy. Tried and tested system.

5. Interesting idea, would require a lot of work. Easier to measure depth of air as per 4 and do the simple maths.

A few other methods come to mind:

1. A pressure sensor at the bottom of the tank. This is often used on ships, works well
but needs immersion which can be avoided.

2. Inflow/outflow monitoring. A flow meter on the output can tell you how much you have used. Inferring level like this is prone to errors though.

There are a million other methods, but they get more and more complex.

The ultrasound would be a great way to do this. The commonly available Parallax Ping sensor would do the job. Realistically, you would only need to make measurements at most once every hour, so you could achieve great battery life. The concrete is likely to present a wireless range challenge. Most RF transceivers will get through concrete though, so place the receiver nearby.

If you want ideas on how it is done, google for ultrasonic oil level monitoring - there are loads of wireless commercial solutions.

---

**Answer** by jlodge

It’s tricky when you only have a small hole to work with, so trying to get inside to install a series of floats or contact pins on the side is not ideal. The ultrasonic is a good idea, but I don’t know how that would be achieved. Personally, I would try the cable, as it is simple enough to drop it in the hole, and wait for your results.

If you wanted to try running a cable through it, you might want to use this Water Level Sensor, or at least base your own project on it.

It is non-corrosive, and has no moving parts, which is supposed to lead to a heightened accuracy, and less chance of it failing or breaking down. The enclosure is water-proof, and you can buy one with a relay attached to fit your monitoring needs. The only issue that I see is that it isn’t wireless, but other than that, it seems to be pretty functional.

---

**Answer** by rickmeasham

I know you asked this a few months ago .. but sometimes with something technical the project never gets completed, or at least not in short order.

So while you’re building your new-world-order with an Arduino, can I suggest you put a pipe down the dip hole and insert a floating dip stick? 1.5” pipe. 1/4” dowel glued to a ping-pong ball. Flag on top of the dowel. Now you’ll see the level any time you walk past.

**Tags:** sensors (Prev Q) (Next Q)

---

**Q: How can I sense when someone is sleeping with my Arduino?**

**Tags:** sensors (Prev Q) (Next Q), arduino-uno-smd (Next Q)

Ignore the stalker-like title.

I am thinking about creating a project with Arduino that can sense you sleep habits, when you’re asleep, and how much you sleep. It would be convenient to not have to turn it
on/off at night; it should know how to do this.

Some ideas that I have:

- Motion sensor
- Breathing patterns (if an Arduino microphone could even pick that out... I would be surprised if it could)
- A RTC clock could assist in deciding *It's midnight. Usual, Steve isn’t asleep at this time on Fridays. He’s probably not sleeping now.* It won’t always be there to help you, but you might be able to try to get more accurate results this way

**How can I accomplish this?** (Extra points for some sort of algorithm that can tell dozing off from a deep sleep.)

*Tags: sensors (Prev Q) (Next Q), arduino-uno-smd (Next Q)*

*User: anonymous-penguin*

---

**Answer** by sachleen

I actually did something like this, using a PIR Motion sensor.

The motion sensor is good enough to detect a person turning in their sleep, and my simple algorithm just counts movements per minute and when its over a threshold (found by trial and error), it sounds the alarm to wake you up.

![Image](image-url)

The image above represents my body movements during a night. The spikes represent a high movements/minute.

Here we see after about 30 minutes body motion was minimal. This suggests I was in
a deeper stage of sleep during that time.

The spike at 250 minutes (around 4:30AM) is when I woke up only to realize I could sleep for couple of hours longer. I recall the time being 4:38 when I looked at my clock and I was already facing it. This means I turned around to face the clock but actually “woke up” 8 minutes later to see the time, and then turned back around (indicated by the shorter spike at 265).

The data suggests I was not able to sleep very well (compared to the time period between 30 and 240 minutes) after waking up once and finally got up at 400 minutes (7:00AM), when I shut everything down.

Here are a couple of posts from my blog that have a lot more information.

- [Body Movement During Sleep](#)
- [Intelligent Alarm Clock Hardware](#)

---

**Q: Grove sensors without a Grove shield**

There exists a large variety of sensors in the [Grove System](#). Usually these are accessed via a special shield (Grove base shield).

Can I access these sensor directly without a shield. In particular I am interested in using them with an Arduino Pro Mini. My guess is that one definitely needs the 5V version of the Mini, but other than this? How fiddly is it to use these sensors directly?

I am interested in using the [Grove dust sensor](#), but I am also would like to operate a [Grove 4 digit display](#).

**Tags:** sensors (Prev Q) (Next Q)

**User:** a.schulz

**Answer** by anonymous-penguin
Above is a schematic of a Groove shield. You can see that each of the connectors has two digital pins connected to it and then V (5V) and G (GND). The last two you can connect normally to an Arduino. The first two you connect to the pins that are used in your sketch. Say it says to connect the connector to the top left DIGITAL pin on the board (D6/D7). You’ll want to connect the pins as followed: Black → GND, Red → 5V, White → D7, and Yellow → D6.

Notes:

- The shield overlaps the pins (i.e. 3-4, 4-5), so that might make things confusing. I think that is a design flaw. If it needs two pins, it should have two connectors. In the current state, you can connect two things together. I don’t really know how to figure out if it uses all of the ports attached or not. If you don’t have to, I would try not to overlap any two grooves. You would do this by using two digital ports for every groove “module.”
- You’ll also need to have a adapter cable (Link from TheDoctor’s Answer.)
- For analog connections, it applies the same way as digital. G is ground, V is 5v, and A# is an analog pin.

Inferencing it:

There’s no real cookie cutter way to connect to them. However, the wiki seems to be a decent resource once you figure out how to navigate it. The dust sensor you mentioned
seems to ONLY use digital pin 8. It was connected to the D7/D8 connector on the Groove shield, so the yellow wire could be ignored since it was never used in the sketch.

It’s hard to do much with this system: it’s poorly documented and not uniform at all. Plus, some boards could cause problems that could even lead to damaging components. I would say look at something else.

- This dust sensor seems pretty good, minus the documentation. On this blog, they say to do this:

  Sharp pin 1 (V-LED) => 5V (connected to 150ohm resister) Sharp pin 2 (LED-GND) => Arduino GND pin Sharp pin 3 (LED) => Arduino pin 2 Sharp pin 4 (S-GND) => Arduino GND pin Sharp pin 5 (Vo) => Arduino A0 pin Sharp pin 6 (Vcc) => 5V

I can’t seem to find the cord for it, but I’m sure there are many others like this online. (It lists the cable needed: a 6-pin, 1.5mm pitch connector. That doesn’t seem too hard to find).

- Here is a 4x7 segment display for much cheaper. It does use a lot of pins and need resistors, but multiplexing and resistors aren’t that hard to figure out. That’s outside of the scope of this question. There might be some other board with a built in IC for controlling a display like that, but I can’t seem to find one now.

---

**Answer** by thedoctor

Grove sensors use a special socket, a 4 pin JST with two power lines and two data. you would have to make your own board/terminal to get them to connect to a mini.

What you could do is use a Grove proto board, which converts the JST terminals onto .1” pins suitable for connecting to the Mini. There is also a little proto area if you want it.

You could also use an adapter cable, which would not have a bulky protoboard along with it. The terminals could be soldered directly to the Mini.

---

**Tags:** sensors (Prev Q) (Next Q)

---

**Q: How can I sense very high temperatures with Arduino?**

**Tags:** sensors (Prev Q) (Next Q), temperature-sensor (Next Q)

I’ve seen many temperature sensors, but the highest temperature they go up to is 125 degrees C (257 degrees F). I want to reach higher temperatures than that. I’m thinking about rigging up a closed loop sensor for my soldering iron (for a convenient LCD display), but I can’t find out how to sense the temperature of the tip due to the higher operating temperature than most sensors.
One of the options to measure high temperatures would be to use thermocouples. This table provides a basic description of the types and the corresponding temperatures that they can sense.

A simple search led me to these components.

- Thermocouple Type-K Glass Braid Insulated - A mid range thermocouple
- Thermocouple Amplifier MAX31855 breakout board - Board to interface the thermocouple with the Arduino. This is needed as the voltage changes in the thermocouple are extremely small, and the rise is non-linear.

This is how the board is connected to the thermocouple.

There is also a tutorial on Adafruit that deals with using and interfacing thermocouples.

This is an image of the thermocouple interfaced with the Arduino.
Q: My DS18B20 is reading high, how can I get it to return the correct temperature?

I am working on a heating control system for my home, but having problems with the temperature sensor reading higher than I expect.

I have a 5V Arduino Mini Pro clone hooked up to a relay module, an HD47780 display and a Tiny RTC module, with a DS18B20 attached to it.

The DS18B20 is powered from the 5v Vcc applied to the Tiny RTC module rather than using parasitic power on the DQ line (as I wasn’t sure if the Onewire libraries supported...
this mode of operation) and the DS18B20 datasheet says that Vcc may be between 3.0v and 5.5v. I measured the regulated power from the Mini Pro’s being at 5.2v.

I am comparing the temperature from the DS with the existing room thermostat (bi-metallic), an RS desktop temperature monitor with two sensors, a small ‘thermo-hygrometer’ and a bi-metallic fridge thermometer (in the upper quarter of it’s temperature range). The readout from the DS is always around 1-2 degrees higher than the readout from the other sensors, which are within a degree of each other.

† Not the actual one, but something similar.

Reading various forums I have heard that these sensors are quite accurate but can suffer from localised heating from components near them, so I moved the sensor onto a 20cm cable and positioned it well away from the rest of the electronics. This made no difference.

I then read that the DS can become self heating if it is supplied with too high a Vcc voltage, so I added a Red LED with 1.7v drop and measured the new voltage over the DS as 3.5v, but again, after waiting for the temperatures to stabilise once more, this made no difference.

Can anyone suggest why the DS may be reading high and what I can can do to prevent this?

Tags: sensors (Prev Q) (Next Q), temperature-sensor (Prev Q) (Next Q)

User: mark-booth

---

Answer by cybergibbons

I’ve put a lot of DS18B20 sensors in place now on various sensor nodes - maybe 50 or so.

I’ve learnt a bit about the readings returned from them. I know you have isolated some of these as not being the cause, but it is good to confirm that they are issues.

I don’t know if you are using Celsius or Fahrenheit, so the degree of the problem isn’t clear.

**Localised heating**

I don’t think of Arduino as a “hot” chip, but putting a DS18B20 in an enclosure with an Arduino and LCD caused me to see readings 2-3°C higher than expected.

Moving it outside the enclosure fixed this.

**Heating from high Vcc/frequent conversions**

If you supply the chip with a high Vcc and perform frequent conversions (i.e. as frequently as you can), the chip does warm up. I have seen this increase readings by 1-2°C.

I haven’t seen the problem with high Vcc alone, but I operate most DS18B20 in parasitic mode and convert at most once every 10s. The ones operated in active mode generally have the powered turned off when not being used.

OneWire and DallasTemperature do support parasitic mode.

**Not all temperature sensors are created equal**
I had a number of temperature sensors before I started playing around with DS18B20. Almost without exception, these under-read. Most of them are based on thermistors and calibration is poor. I thought the DS18B20 were over reading, but when I used a good thermocouple and meter, I could validate that they were correct.

This was about learning to trust the DS18B20.

**Heat conducted along wire**

If you have the sensors on a wire, the wire conducts a surprising amount of heat. This can be an issue when running wiring along pipes.

**Fake DS18B20**

These sensors are very popular, and this seems to have caused fakes to appear. I have a batch of stainless steel waterproof sensors that over-read and the timings were out. They worked with the OneWire library, but not a DS2482 OneWire master. I have heard of the normal TO-92 cased ones being fake as well.

---

**Tags:** sensors (Prev Q) (Next Q), temperature-sensor (Prev Q) (Next Q)
Q: How do I know if a board fits my requirements?

I’m new to Arduino and I saw on there website that there are many different kinds of boards. What are the differences, or to be more specific, how would I know which board is meet my requirements? I understand that the shields can be connected to the board, but are some boards that do not support certain shields?

Another thing, is it possible to connect external sensors to an Arduino board or there are special sensors that can communicate with the board? I want to use weight sensors.

Answer by anonymous-penguin

Things to consider when picking an Arduino board:

- Do I want native keyboard/mouse support? If yes: you’ll need the Leonardo (or it’s breadboard mountable equivalent) or the TRE (not released, dual MCU board).
- Do I want a small form factor and/or have it breadboard mountable? If yes, choose a board like the Mini/Micro/Nano/etc.
  - If you want to make it even smaller, either get a board without USB built in or try using an ATTiny
- Am I computing large numbers with it? If yes: use the Due, Galileo, or TRE (not released yet)
- Do I want video/audio? If yes: TRE is the only way to go as of now (but it’s not released yet).
- Do I need a USB host (and not want to have an external shield)? If yes, use the Mega ADK (limited support), Yun, TRE (again, not released yet), or Galileo.
- Do I need WiFi/ethernet built in? If yes; use the Yun, TRE (still, not released yet), Ethernet board, or Galileo.
- How many pins do I need (and how many of each type)?
  - Digital: simple ON or OFF. Use these for buttons, simple lights, some communication, etc.
    - Stay away from the Due if you don’t want to mess with 3.3V logic. Most other boards are 5V. You can work around this with either voltage dividers (doesn’t always work) or level shifters. As far as I know, you can’t really do much if a sensor provides max 5V; level shifters AFAIK only work with digital signals.
  - UART Pins: these are the pins that you can use the serial library on. (Note: this doesn’t include the software-serial library). One of these on most boards is connected to a USB chip.
  - SPI Pins: These are pins which are used by a form of communication (like on the Ethernet shield).
• Interrupt Pins: These can trigger an event when the signal getting sent to the board changes. There are a couple of different settings, but this can be used for near-instant response time when low latency is critical.

• PWM Pins: pins that go on and off to simulate a lower voltage. It does this by turning it on and off very rapidly. This is used often for dimming LEDs and controlling the speed of a motor.

• Analog pins: pins that can read a voltage and convert it into a number.
  - The Arduino Due has a much higher precision rate, as it has more possible values and it has less voltage, thus much more accuracy.

All of these types of pins can act as a digital pin. All of these, except the analog pins, are under the “digital pin” section of your Arduino board. I didn’t go into the technical aspects of the pins; I explained them in the simplest way I could.

I left a lot of boards out of this list, but this should give you some things to think about when choosing a board. There’s a lot of weird quirks with this list, so it’s incomplete, especially with the types of pins. You’ll just have to do research and figure out what you need and if a board can meet your needs. Just make sure to not plan two things on the same pin, especially when mixing different shields and sensors! Shields do take up pins.

Shields: they are pretty much compatible with every board except for if the board is 3.3V (newer shields can adapt automatically) OR if it’s a breadboard mounted board, like the Nano. There are ways to work around this problem, but it’s very difficult.

---

**Answer** by electropepper

To start with arduino you need to have some factors in consideration, for example the board you will chose, arduino UNO being the starting board for most beginners, if you are already an experienced programmer in C++ you might want to try more powerful boards. You have a pretty good overview of all of them in this link: [http://arduino.cc/en/Main/Products](http://arduino.cc/en/Main/Products) Also there are hundreds of shields out there, you will find some in the link above. Adafruit also has some interesting ones, [https://www.adafruit.com/search?q=arduino](https://www.adafruit.com/search?q=arduino), you can even find them on ebay, and you can always buy individual sensors, LEDs, buttons etc, and plug them in directly or in a breadboard. Sparkfun has a big variety of very interesting sensors. There is also a lot of open source shields out there, you can check a very simple one i designed myself: [http://electropepper.org/projects/prototyping/item/proto-i-os-arduino-v1-0](http://electropepper.org/projects/prototyping/item/proto-i-os-arduino-v1-0)

**Answer** by federico-fissore

The usual suggestion I give to people new to Arduino is to get a Starter Kit (either from the store or from a local reseller): it contains an UNO, plenty of sensors and actuators and a projects book.

**Tags:** sensors (Prev Q) (Next Q), shields (Prev Q) (Next Q)

---

**Q: Strange values from LSM303 Magnetometer**
I am having problems with values I get from Pololu MinIMU-9 board’s magnetometer. I am using Arduino-micro and the arduino library from Pololu. The sensor on the board is 3-axis acceler- and magnetometer LSM303. The problem is - the highest value always seems to be in vertical axis, not horizontal, as I would expect.

For example, here is the data I got when I put sensor flat on the table and made 360 clockwise turn in horizontal plane: (values are X Y Z)

```
Skip code block

-6 102 -516
-11 103 -515
-11 103 -515
-21 102 -514
-31 109 -515
-41 114 -512
-57 116 -509
-57 116 -509
-77 111 -507
-93 109 -506
-104 106 -507
-104 106 -507
-114 99 -508
-126 88 -504
-131 74 -507
-142 52 -507
-142 52 -507
-151 31 -506
-150 4 -509
-147 -27 -513
-147 -27 -513
-133 -60 -513
-119 -80 -519
-112 -87 -517
-102 -97 -521
-102 -97 -521
-89 -101 -522
-77 -101 -524
-50 -104 -527
-50 -104 -527
-43 -105 -529
-14 -98 -532
2 -92 -535
23 -92 -535
23 -92 -535
36 -81 -534
86 -54 -534
76 -32 -531
76 -32 -531
76 -21 -532
78 -7 -531
85 10 -528
91 39 -524
91 39 -524
94 50 -526
91 63 -524
76 75 -525
76 75 -525
63 82 -526
47 93 -525
44 98 -522
44 98 -522
39 104 -521
26 118 -521
26 112 -518
21 116 -519
21 116 -519
18 116 -518
12 120 -517
11 120 -515
11 120 -515
```
Turning the sensor “upside down” and then rotating it in horizontal plane again gives similar results, only Z axis has positive 500+ value.

Shouldn’t the magnetometer always point in the direction of the magnetic field - in X or Y direction? I have tried this in different rooms and buildings and got the same results. Rotating the sensor on other axes yields the same results - vertical axis has the largest (and constant) value.

For reference, here is the code i used to get the values.

```cpp
#include <Wire.h>
#include <LSM303.h>
LSM303 compass;

void setup()
{
  Serial.begin(115200);
  Wire.begin();
  compass.init();
  compass.enableDefault();
}

void loop(){
  compass.readMag();
  Serial.print(compass.m.x);
  Serial.print(" ");
  Serial.print(compass.m.y);
  Serial.print(" ");
  Serial.print(compass.m.z);
  Serial.print("\n");
  delay(100);
}
```

The earth’s magnetic field is not parallel with the ground. In some locations it can be pointing more down than across.

NOAA has a magnetic field calculator where you can enter your latitude and longitude. For example, I am located at approx 19deg S 147degE and the field is

<table>
<thead>
<tr>
<th>North Comp</th>
<th>East Comp</th>
<th>Vertical Comp</th>
<th>Total magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>32,434.4 nT</td>
<td>4,280.6 nT</td>
<td>-36,035.1 nT</td>
<td>48,670.7 nT</td>
</tr>
</tbody>
</table>

So the downwards strength is more than the other two combined for my location. I’m guessing the calculator will also show a very strong down component for your location.

Some other things to be aware of are that the LSM303DLHC magnetometer can have quite large offsets, up to the equivalent of 2 earth magnetic fields, and therefore requires calibration. Also hard and soft iron distortions can affect the readings.

Edit: Please see my answer here for methods to calibrate the sensor.

Tags: sensors (Prev Q) (Next Q)
Q: Which reflectance sensor array for a line following robot?

Tags: sensors (Prev Q) (Next Q)

I’m working on a line-following robot with my son. We’ve found a sensor array on Robot Shop that will help us but .. there’s two and I’m not sure which will be the best for us.

The difference between them, as described in this PDF is “The QTR-8A uses a voltage divider for each sensor’s output while the QTR-8RC uses an RC circuit.”

What I want to do is detect which sensor the followed line is triggering. All I need is a yes/no. But I’m too new to this to decide which of these two sensors would be best for me.

Tags: sensors (Prev Q) (Next Q)

User: rickmeasham

Answer by omer

The main difference between the two models, as stated in the PDF you provided, is the output they provide to the MCU.

The QTR-8A will provide an Analog output, requiring a corresponding analog pin on the microcontroller. The QTR-8RC will provide a Digital output, but will allow the same accuracy of the QTA-8A by measuring the discharge time of the capacitor (HIGH to LOW time), with the advantage of higher accuracy due to the use of RC circuit instead of a voltage divider.

Both seems pretty straight forward, but for your needs, the QTR-8RC might seem more compatible, considering the Arduino Uno has 6 analog I/O pins (you didn’t mention which Arduino is going to be used, the Arduino Nano has 8 and the Arduino Mega has 16), and that the QTR sensor provides 8 outputs. The way you could accomplish that with QTR-8RC and 8 digital I/O pins is described in the PDF:

1. Turn on IR LEDs (optional)
2. Set the I/O line to an output and drive it high
3. Allow at least 10 us for the 10 nF capacitor to charge
4. Make the I/O line an input (high impedance)
5. Measure the time for the capacitor to discharge by waiting for the I/O line to go low
6. Turn off IR LEDs (optional)

Tags: sensors (Prev Q) (Next Q)

Q: Why is the constrain function used after the map function?

Tags: sensors (Prev Q) (Next Q)

I am reading some sample code and they use this method of mapping data from IR
sensors:

```c
sensor1 = analogRead(0);
adj_1 = map(sensor1, s1_min, s1_max, 0, 255);
adj_1 = constrain(adj_1, 0, 255);
```

What is the point of using `constrain` here if `adj_1` is already getting a value 0-255 from the `map` function?

**Tags:** sensors *(Prev Q) (Next Q)*

**User:** michael-rader

**Answer** by loganblades

From the official Arduino site:

> [The map function] does not constrain values to within the range, because out-of-range values are sometimes intended and useful. The constrain() function may be used either before or after this function, if limits to the ranges are desired.

EDIT: Example.

You can try this by yourself with this code:

```c
int val = 20;
val = map(val, 0, 10, 0, 100);
```

Although you set the upper bound of the value’s range to 10, you passed an higher value than that and the function will linearly map it accordingly, resulting in an output of 200.

**Tags:** sensors *(Prev Q) (Next Q)*

**Q: Compute yaw from magnetometer and accelerometer**

**Tags:** sensors *(Prev Q) (Next Q)*

I use an Arduino and a 9 DOF sensor (gyroscope, accelerometer and magnetometer) and I’m trying to use the pitch, roll and yaw that the sensor gives me to rotate an object in unity.

I managed to correctly compute pitch and roll (the z and x axis in unity) from the accelerometer but it seems that I can’t get the Yaw right. By that I mean that when I rotate my sensor pitch or roll it rotates my yaw too, in a weird way.

**Code in the arduino to get the heading**

```c
void getHeading(void)
{
    heading=180*atan2(Mxyz[0],Mxyz[1])/PI;
    if(heading <0) heading +=360;
}

void getTiltHeading(void)
{
    //float pitch = asin(-Axyz[0]);
}
Here is the code snippet that calculates pitch and roll:

```c
float roll = asin(Axyz[1]/cos(pitch));

float xh = Mxyz[0] * cos(pitch) + Mxyz[2] * sin(pitch);
float yh = Mxyz[0] * sin(roll) * sin(pitch) + Mxyz[1] * cos(roll) - Mxyz[2] * sin(roll) * cos(pitch);
tiltheading = 180 * atan2(yh, xh)/PI;
if(yh<0) tiltheading +=360;
```

I don’t hesitate to ask for details.

**Tags:** sensors (Prev Q) (Next Q)

**User:** dnotol

---

**Answer** by brettam

I don’t know how all your sensor’s axis are lined up, but it looks like you should level the magnetometer readings by rotating them by -pitch and -roll to get back to the global reference frame, instead of rotating the magnetometer readings by the positive angles.

However, The pitch and roll angles in the code are both relative the horizon, not a set of euler angles that define a rotation between two states; because of this, they do not compose to rotate the magnetometer back into the global xy frame like I assume the equations are trying to do. The code is also only using atan for pitch and roll instead of atan2, so their range is only 180 degrees.

You need to make sure the rotation matrix you multiply your magnetometer reading by is based on the same conventions as the equations you use to get pitch and roll from the accelerometer, and that it cancels the observed rotation.

To do this correctly, I recommend you read into euler angles. You might also try searching around for a good library that handles 9-DOF sensor fusion for you.

---

**Tags:** sensors (Prev Q) (Next Q)

---

**Q: ArduinoMEGA: 64 digital inputs cause random digitalRead values**

**Tags:** sensors (Prev Q), arduino-mega (Prev Q) (Next Q)

I have 64 hall effect sensors (magnetic field sensors) (DRV5023AJQLP) connected to an Arduino MEGA 2560. For that, I use 48 digital pins and 16 analog pins as digital input pins. The program in the Arduino continuously reads out those 64 digital input pins (in a for-loop) and sends out a serial byte, to a laptop via a usb cable, whenever one of those input pins changes between LOW and HIGH. I also use the ATmega2560’s internal pullup resistors to prevent irregular readouts of the input pins when the sensor is not giving a 0v output (i.e. when it isn’t measuring a strong enough magnetic field).
Everything is working fine when I only physically connect up to 16 of the sensors. The laptop then only receives bytes that correctly tell when a sensor measures a state-change. However, when I start connecting more of those 64 sensors, then the laptop starts receiving irregular streams of bytes, erroneously suggesting that all the connected sensors measure changing magnetic fields (HIGH and LOW readouts).

I can think of two reasons why these irregular readouts occur: 1. Either, the ATmega2560’s internal pullup resistors aren’t working as I think they should. 2. Or, the arduino board and >16 hall effect sensors combined are pulling to much current.

1. **Pullup resistors not working?** When testing with only 16 sensors connected, it is a stable, properly working setup. When I explicitly *don’t* use the pullup resistors in the program then the setup *does* start behaving irregular, also with just 16 sensors. So the internal pullup resistors seem to be properly activated by the program.

2. **Arduino + 64 sensors draw to much current?** Besides powering the Arduino board and sensors via the usb cable (providing 500mA), I’ve also tried external power sources that provide 1 or 2A. I’ve also powered the Arduino board via an external 5v, +1A voltage regulator, bypassing the Arduino’s onboard voltage regulator. This all doesn’t seem to make a difference in the system’s irregular behaviour. When measuring those setup’s current usage (with amperage meter) it didn’t go above 250mA.

Calculating the combined current usage (using datasheet info), I came to this:

<table>
<thead>
<tr>
<th>Description</th>
<th>Current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- sensor operating current:</td>
<td>2,7</td>
</tr>
<tr>
<td>- sensor output current when in ‘active’ state:</td>
<td>0,25</td>
</tr>
<tr>
<td>- TOTAL current of 64 sensors (3mA*64)</td>
<td>192</td>
</tr>
<tr>
<td>- Arduino MEGA current (approx)</td>
<td>150</td>
</tr>
<tr>
<td>- TOTAL current of 64 sensors + Arduino MEGA:</td>
<td><strong>342</strong></td>
</tr>
</tbody>
</table>

So, both the calculated and measured current usage seem within the range of what can be handled by mentioned power sources.

I’m kinda stuck now. Can anyone suggest anything that might cause this problem? Or point to incorrect thinking of me? Thanks for any responses.

‘Schematic’ (only showing first 8 sensors connected. The Arduino MEGA is connected to a computer via a usb cable with disabled power line):
Top-side: the hall effect sensors and Vcc an GND rails:

Bottom-side: ArduinoMEGA + the 64 red sensor output lines to the input pins:
Below is the Arduino code:

/*
Arduino to Pd protocol

Placing or removing a chess piece sends only one byte.
There’s no other serial data being communicated.

event: byte value:
placing a chess piece on a square: 0-63
removing a chess piece from a square: 64-127

Arduino MEGA pin to byte value mapping:

<table>
<thead>
<tr>
<th>byte</th>
<th>pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>46</td>
<td>52</td>
</tr>
<tr>
<td>47</td>
<td>53</td>
</tr>
<tr>
<td>48</td>
<td>A0 (analog 0)</td>
</tr>
<tr>
<td>49</td>
<td>A1</td>
</tr>
<tr>
<td>50</td>
<td>A2</td>
</tr>
<tr>
<td>62</td>
<td>A14</td>
</tr>
<tr>
<td>63</td>
<td>A15</td>
</tr>
</tbody>
</table>
*/

byte prevSquares[64];  // array that holds previous readouts
byte i;                // variable used in for-loops
byte sensorValue;      // variable temp storing of digitalread
```c
void setup() {
  Serial.begin(9600);
  for (i = 0; i < 64 ; i++) {
    pinMode(i+6, INPUT_PULLUP);  // activate pullup resistor
  }
  for (i = 0; i < 64 ; i++) {
    prevSquares[i] = digitalRead(i+6);
  }
}

void loop() {
  for (i = 0; i < 64 ; i++) {
    sensorValue = digitalRead(i+6);
    if (sensorValue != prevSquares[i]) {
      // chess piece was placed or removed from square
      if (sensorValue == 0) {
        // chess piece was placed
        // send serial byte (0-63):
        Serial.write(i);
      }
      else {
        // chess piece was removed
        // send serial byte (64-127):
        Serial.write(i + 64);
      }
    }
  //update prevSquares array:
  prevSquares[i] = sensorValue;
  }
}
```

**Tags:** sensors (Prev Q), arduino-mega (Prev Q) (Next Q)

**User:** edo-paulus

**Answer** by jake-c  

After seeing your schematic, I think the general problem here is that you need to more closely read and follow the datasheet for this part. A lot of times, when you are hобbling a project together at a small scale there are a lot of things you can ignore and get away with, but as you scale up, all the rounded corners start to add up. In this case, you can get away with 16, but once you start adding too more, it becomes too much. Let’s go over a few:

**Pull up resistors**

Like I mentioned in the comments, relying on the internal pull ups is probably not a good idea. In my personal experience, they are only really good for adding a quick button input for debugging purposes. I’m not sure specifically what would cause issues only after 16 sensors. It could be that collectively internal pullups are subject to a different and lower current limit, but I’m not familiar enough with the internal architecture of the ATMega2560 to do more than just speculate.

Whatever the case, the best practice here would be to use separate external pull up resistors, within the specifications of the datasheet. (See equation 1 on page 13 of the datasheet.)

**Decoupling Capacitor**

Per the datasheet, page 17:

A 0.01-µF (minimum) ceramic capacitor rated for VCC must be placed as close to the DRV5023 device as possible.
This capacitor is referred to as C1 in the datasheet figures and is required. This type of capacitor is called a decoupling capacitor, and one of the things it does is provide a local energy storage, so that when the sensor starts pulling a little extra current, it doesn’t have to wait for the regulator to respond, it has some local energy already.

Your schematic indicates a single 1uF capacitor on the output of the voltage regulator, however, there are no capacitors for the sensors. The one capacitor is only for output filtering for the regulator, and will not satisfy any decoupling needs of the sensor.

**Filtering Capacitor**

Referred to as C2 in the datasheet, this an optional capacitor to filter out high frequency noise. This isn’t needed in most cases, in fact, per the datasheet, page 13:

> Most applications do no require this C2 filtering capacitor.

However, your project might not fall into “most applications” and there are a few things specifically that worry me.

1. You have long wires that are basically acting as antennas that will capture any nearby noise.
2. You have long power lines running parallel to your signal lines which can cause stray inductance which leads to other interesting things. (With the thickness of the plywood separating them, this might not be an issue, but it is something to be aware of).

I would definitely try the first two ideas above to begin with, and only pursue this as a last resort. The only way to know for sure if this is something you need is to stick an oscilloscope on the line and see how noisy it is, but that is not something you probably have handy.

**Other Concerns**

I’m just a little bit worried about voltage drop. Depending on the gauge of wire used, with all the sensors connected, at the end of a strand you might have a non-negligible voltage drop.

With that many wires running together into the Mega, there might be cross talk problems. Another common problem with scaling up is that there is a higher probability of having faulty components. This is especially with this part which per the datasheet indicates that is particularly sensitive to ESD. It could be that there is a dead sensor that has failed in such a way that it is causing problems for the other sensors. Maybe go through and test all 64 sensors, 8~16 at a time to make sure that separately they all work, and that the issue only occurs when all of them are connected.

**Other Ideas**

Many other comments on the question have suggested a multiplexer and/or multiplexing and I will just briefly echo that here. Multiplexers are typically used in situations where you don’t have enough pins and you want to split a single pin to multiple other pins. While that isn’t an issue here with the Mega (it appears you have enough pins to do what
you are wanting to do), there are other benefits that can be derived from multiplexing this.

For example, one thing you could do is split up the board into 8 - 2 x 4 chunks and then have a multiplexer (such as a 4051 or a 74151) local to each chunk. Then you would have just 8 short wires going into the multiplexer, as opposed to having long antenna-like wires heading to the Mega. This could improve overall signal integrity.

**Tags:** sensors (Prev Q), arduino-mega (Prev Q) (Next Q)
Pins

Questions

Q: What happens to the Arduino if you don’t use a resistor with an LED?

Tags: pins (Prev Q) (Next Q), led (Prev Q) (Next Q), safety (Prev Q) (Next Q)

Making a blinking light seems to be one of the most common beginner projects with Arduino. A typical approach is to connect an LED + resistor (in series) to an IO pin, and connect the other end to ground. In code, you make the pin an output, and bring it HIGH and LOW repeatedly to flash the LED.

What happens if you leave that resistor out? Obviously you’re going to wreck the LED, but will it also damage the Arduino in some way?

Tags: pins (Prev Q) (Next Q), led (Prev Q) (Next Q), safety (Prev Q) (Next Q)

User: peter-r.-bloomfield

Answer by cybergibbons

From page 411 of the ATmega328P data sheet, we have a graph of the output current vs voltage.

Figure 29-162. ATmega328P: I/O Pin Output Voltage vs. Source Current ($V_{CC} = 5$ V)
We can see over the the 20mA range of output currents, we drop ~0.5V, giving an approximate internal resistance of 25Ω.

A typical LED has a forward voltage of about 2V. Therefore, we end up in the situation where we will drop 3V over the 25Ω internal resistance of the ATmega328P, giving an output current of 120mA.

\[ I = \frac{(5-2)V}{25\Omega} = 120mA \]

This is a simplification, but this 120mA is far larger than the recommended maximum current per pin of 40mA of the microcontroller, and far in excess of what a normal LED should take.

That said, most LEDs and the ATmega328P seem to survive this kind of abuse, especially for short periods. In general, if the pin continues to work after this is done, it should be fine.

Answer by jippie

Leaving series resistors out will definitely greatly (exponentially) shorten lifetime of both the controller and the LED. Absolute maximum ratings for most AVRs are (a.o.):

- max. 40mA per GPIO pin
- max. 200mA per package.

A new Arduino sets you back $20 or more, a fist full 220 ohm resistors sets you back 1$. When treated within spec, an Arduino (and LEDs for that matter) can survive a lifetime.

Tags: pins (Prev Q) (Next Q), led (Prev Q) (Next Q), safety (Prev Q) (Next Q)

Q: Can a function be called automatically when an input changes?

Tags: pins (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

Currently, my sketch is checking an input pin every time round the main loop. If it detects a change, it calls a custom function to respond to it. Here’s the code (trimmed down to the essentials):

```c
int pinValue = LOW;

void pinChanged()
{
    //...
}

void setup()
{
    pinMode(2, INPUT);
}

void loop()
{
    // Read current input
    int newValue = digitalRead(2);
    // Has the input changed?
```
if (newValue != pinValue) {
    pinValue = newValue;
    pinChanged();
}

Unfortunately, this doesn’t always work properly for very short changes on the input (e.g. brief pulses), especially if loop() is running a bit slowly.

Is there a way to make the Arduino detect the input change and call my function automatically?

Tags: pins (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

User: peter-r.-bloomfield

Answer by peter-r.-bloomfield

You can do this using external interrupts. Most Arduinos only support this on a limited number of pins though. For full details, see the documentation on attachInterrupt().

Assuming you’re using an Uno, you could do it like this:

Skip code block

```cpp
void pinChanged()
{
    //...
}

void setup()
{
    pinMode(2, INPUT);
    attachInterrupt(0, pinChanged, CHANGE);
}

void loop()
{
}
```

This will call pinChanged() whenever a change is detected on external interrupt 0. On the Uno, that corresponds to GPIO pin 2. The external interrupt numbering is different on other boards, so it’s important to check the relevant documentation.

There are limitations to this approach though. The custom pinChanged() function is being used as an Interrupt Service Routine (ISR). That means the rest of the code (everything in loop()) is temporarily stopped while the call is executing. In order to prevent disrupting any important timing, you should aim to make ISRs as fast as possible.

It’s also important to note that no other interrupts will run during your ISR. That means anything relying on interrupts (such as the core delay() and millis() functions) may not work properly inside it.

Lastly, if your ISR needs to change any global variables in the sketch, they should usually be declared as volatile, e.g.:

```cpp
volatile int someNumber;
```

That’s important because it tells the compiler that the value could change unexpectedly, so it should be careful not to use any out-of-date copies/caches of it.
Any state of change on any pin configured as digital input can create an interrupt. Unlike the unique vectors for the interrupts causes by INT1 or INT2 the PinChangeInt feature uses a common vector and then Interrupt Service Routine (aka ISR) for this vector needs to then determine which pin changed.

Fortunately PinChangeInt Library makes this easy.

```cpp
PCintPort::attachInterrupt(PIN, burpcount, RISING); // attach a PinChange Interrupt to our pin on the RISING, FALLING and CHANGE all work with this library
// and execute the function burpcount when that pin changes
```

**Q: How can higher current devices (motors, solenoids, lights, etc.) be controlled by an Arduino?**

I’m looking for a broadly applicable solution, one that can be adapted to a variety of projects.

I’m currently working on several projects that each require controlling devices ranging from 800mA to 2A from an Arduino Uno. One controls stepper motors, one controls 12vdc solenoid actuators, and one controls 12vdc pneumatic valves.

For example:

The Arduino monitors a button, and each time the button is pressed it triggers the solenoid actuator. Because the Arduino is unable to source the current required by the solenoid, a separate power supply is required with the Arduino controlling a switch (relay, transistor, etc.) that allows the higher current to pass. For the stepper motor, the layout is more complex as there would need to be four pins controlling four separate switches (to maintain interoperability of the circuit). The relay controls an air valve and requires 12vdc as well.

I’m trying to figure out how to use a single circuit that can be used in each of these applications (and any future projects) that involve controlling higher current devices than the Arduino pins can handle.

Prototyping speed, standardized components, and low cost are the driving factors. Switching speed, useful life, and noise are also important.

Is there a breakout board, circuit, or component that can be connected to an Arduino pin and used to control a high current device? Ideally with a software controlled potentiometer so that the resistance for different projects could be set in the sketch itself.
To drive such high currents, you may have to cascade several transistors (you can also use a Darlington transistor). There are arrays of Darlington transistors mounted in a chip (e.g. the ULN2803A has 8 darlington transistors, but is limited to 500mA).

You probably will have to deal with higher power transistors; as an example I have found STMicroelectronics TIP110 which can support switching a 2A current (4A peak), but it would probably need a heatsink to dissipate heat.

Note that I wonder if your steppers really need 2A current (are they that big?). For steppers, you can generally find IC that can drive them easily, e.g. the L293D but this one can drive “only” 600mA).

As a conclusion, I am afraid you will not find a “one size fits all” solution, as all your devices are different and should be driven by the appropriate circuit.

**Edit:**

Since oversizing is not an issue in your prototyping case, then you could go with a MOSFET instead of usual bipolar transistors. A MOSFET will be able to drive higher currents and voltages than standard transistors.

The downside is you can use it as a switch only (like e.g. a relay) and thus can’t really drive the exact power for your devices. I guess that does not matter for a Stepper motor, or a Solenoid, but that may be important for driving lights for instance.

However, the good point is that you can still use PWM for that as the MOSFET switching speed is good enough for such purposes.

Regarding price, there are many different kinds of MOSFET out there, but I guess you can find one that fits your needs (12V, 2A) for less than $1.

I advise you to take a look at this great article about this topic.

---

**Answer** by peter-r.-bloomfield

There are lots of ways to switch higher loads, and jfpoilpret has described some good options. I’ll summarise a couple of relay-based solutions, which are mainly appropriate for comparatively slow switching speeds (i.e. not usually suitable for PWM).

**Solid State Relays**

Solid State Relays (SSRs) are effectively semiconductor-based switches. They come in a wide variety of configurations, depending on your requirements, but the key factor is that they have no moving parts. This means they can be very reliable in the long run if used properly.

Internally, they are usually comprised of MOSFETs and thyristors or similar. This can let them achieve fairly high switching speeds in theory. In practice though, the more power it’s designed for, the harder it is to switch quickly. That means high speed + high power can get quite expensive.

A critical factor to bear in mind is that you’ll usually need a different type of SSR if you intended to switch AC instead of DC. It’s also good to note that some will come with a built-in opto-isolator or similar to keep your power supplies separate.
Electro Mechanical Relays
This is the more ‘traditional’ approach. An Electro Mechanical Relay (EMR) is a fairly simple component, containing a mechanical switch, controlled by an electromagnetic coil. If the switch is normally open, the coil pulls it closed when a control current is applied. In contrast, a normally-closed switch would be pulled open when a control current is applied.

There are a number of advantages of EMRs over things like SSRs. The most obvious is cost — their simplicity makes them quite cheap, and the cost doesn’t rise so sharply for higher-power versions. Additionally, the control and load are inherently isolated, and they don’t care whether you’re switching AC or DC.

There are several disadvantages though. The mechanical aspect means that EMRs are usually much slower than non-mechanical switching solutions, and can suffer from contact bounce. Additionally, they can physically wear out, and they can be affected by things like shocks, vibrations, and (potentially) other magnetic fields.

When designing a circuit to use an EMR, it’s essential to be aware of back-EMF (electromotive force). When a control current is applied, the coil acts as an inductor, storing charge electromagnetically. When the control current is stopped, the stored charge can surge back through the control circuit, creating a large negative voltage spike (potentially much larger than what was originally applied).

This spike can unfortunately damage/destroy any attached components or microcontroller pins. It is typically prevented/mitigated by putting a diode in reverse across the relay’s control contacts. In this context, it’s sometimes known as a flyback diode, and it allows the EMF to dissipate safely.

---

**Answer** by david-cary

As jfpoilpret already said, a power MOSFET is great for turning ON and OFF the 12 VDC power to devices that pull up to 44 A. There are dozens of such power MOSFETs for under $1 each. More expensive MOSFETs are available that can handle much higher current and voltage.

In principle it is possible to drive a stepper motor with a microcontroller and a handful of transistors and a few other little parts. However, many people prefer to use a “stepper driver chip”, so it is impossible for a software bug to accidentally turn on transistors in such a way as to short the power supply to ground (typically destroying at least 2 transistors). Many recent stepper driver chips also handle microstepping, current limiting, thermal overload protection, and other nice features.

All the stepper driver chips that I’ve ever heard of, and a few off-the-shelf breakout boards using those chips, are listed at [http://reprap.org/wiki/stepper_motor_driver](http://reprap.org/wiki/stepper_motor_driver).

In particular, many of the RepRap 3D printers I’ve seen connect an Arduino to four Pololu stepper drivers (under $15 each) to drive five stepper motors.

---

**Tags:** pins (Prev Q) (Next Q), voltage-level (Prev Q) (Next Q)
Q: Why does my sketch report too many button presses?

Tags: pins (Prev Q) (Next Q), button (Prev Q) (Next Q)

I’ve been using a button to interact with a sketch (on my Uno), and it’s mostly working fine. However, sometimes it seems to be detecting multiple presses when it’s only pressed once.

It’s a small circuit-mounted momentary button, which is normally open. It’s connected from +5V directly to an input pin. I’ve also added a 10K pull-down resistor on the pin to make sure the input isn’t left floating.

Here’s a test sketch I’ve been using, which reports the number of presses to the serial monitor:

```cpp
int oldState = LOW;
int numPresses = 0;

void setup() {
    pinMode(5, INPUT);
    Serial.begin(9600);
}

void loop() {
    const int newState = digitalRead(5);
    if (newState != oldState) {
        if (newState == HIGH) Serial.println(++numPresses);
        oldState = newState;
    }
}
```

Sometimes when I press the button once, the serial monitor shows two presses in a row. It sometimes even reports a press when I release the button.

Is it possible to fix this? Or do I just have a faulty button or pin or something?

Tags: pins (Prev Q) (Next Q), button (Prev Q) (Next Q)

User: peter-r.-bloomfield

Answer by jlbnjmn

The button contacts themselves are probably bouncing in and out of contact, which is causing the circuit to open and close multiple times, even though the button position isn’t changing.

It’s a common problem with buttons and would be the first thing I would check. In order to fix this, you need to de-bounce the button. There are various ways in which to accomplish this, some of which can be done in the sketch itself.

A good solution is code that measures the duration of each button state and only registers a change in state if the current duration is greater than the desired duration. That option should be used for more complex projects. The code for it will be project specific and likely use sketch defined variables, but an example of what needs to be done can be found
If the sketch is very basic or you just want to make sure that bounce is the problem, you can add a short delay after the button state changes. For example:

```cpp
void loop()
{
    const int newState = digitalRead(5);
    if (newState != oldState) {
        if (newState == HIGH) Serial.println(++numPresses);
        oldState = newState;
        delay(10) //Adjust the delay length to be as short as possible without triggering errors
    }
}
```

Answer by jfpoilpret

This is your button **bouncing** which is quite usual behavior; it has nothing to do with button quality.

There are 2 ways to “debounce” a button: by hardware or by software.

Personally, I prefer by hardware; just google for button debouncing circuit and you should find many examples.

If you don’t have the necessary components available, you can use software debouncing.

By software, you have an example here on Arduino web site.

Tags: pins (Prev Q) (Next Q), button (Prev Q) (Next Q)

---

**Q: A digital I/O pin as ground**

Tags: pins (Prev Q) (Next Q), arduino-due (Prev Q) (Next Q)

I want an I/O pin to return current in a circuit, similarly to the function of ground.

Which of these will achieve what I require?

```cpp
pinMode(pinnumber, OUTPUT)
digitalWrite(pinnumber, LOW)
```

or

```cpp
pinMode(pinnumber, input)
```

Tags: pins (Prev Q) (Next Q), arduino-due (Prev Q) (Next Q)

User: rajat

Answer by fredp

Your first snippet is the correct solution.

```cpp
pinMode(pinnumber, OUTPUT)
digitalWrite(pinnumber, LOW)
```
Be careful though, you can only sink up to 6mA or 9mA per pin (as per the documentation [“Input and Output” section], I never tried more). If you need more current, use a transistor (this looks like a good example, you just need to invert the output pin to HIGH).

When the pin is configured as input it is supposed to be high impedance, not what you want.

Q: Difference between PWM and regular output port for servos?

I have been controlling my servomotors for a while now only using pin 0. After some research, it has become apparent to me that the ports with ~ are the PWM pins.

I thought that all pins sent out PWM signals, therefore the motor was able to move back and forth. Now I’m really confused as to how the motor was able to turn without using a ~ pin like pin 5.

Can someone explain why I should or should not use PWM signal when controlling a motor?

Servos does not use PWM, it uses PPM:

- PWM = Pulse width modulation, The pulse start at 0ms, and is high for the percentage of the time compared to the percentage of the voltage you want, 5v signal, if you want 1v, the pulse is 20ms High and 80ms Low.
- PPM = Pulse position modulation, The pulse length is always 5ms High and 95ms low, but the interesting part is if high starts after 5ms or 50ms (or any other).

So PWM is to lower voltage, PPM is a signal pulse.

This means that no matter what port you are using on the arduino you are able to control a servo since you only need High or Low, and not a lower voltage.

What you need to be aware of in this case is on the arduino is a PPM signal is using up a timer. The arduino has 3 timers where of 2 is available, one is used internally. Though there is code where you can attach up to 3 servos on the same internal timer.

Q: ATMega328P-PU and 328P-AU
im planing to make my very own arduino design based on the Arduino PRO mini. I picked this board instead of the UNO mostly because it’s way more simpler compared with the UNO since the USB-to-Serial part is missing. As many of you know the MINI uses the 328P-AU which is the SMD version of the 328P however since the AU version is too small for me i would like to use the PU (UNO’s chip) instead. From the datasheets the only difference is a small change in the pins position. And the one million dollar question is: Is there any difference on those 2 chips? If i try to use the same design just with the PU model is it gonna work?

Answer by duskwuff

The -AU and -PU suffixes on the part name indicate different packages. The former is used for the “32A” package (32-lead TQFP), and the latter is used for the “28P3” package (28-lead DIP). In plain English, the -AU is a surface-mount part, and the -PU is a breadboard-friendly chip.

As far as functionality, there is virtually no difference. The 32A package exposes pins for ADC6 and ADC7, which are not available on the 28P3 package, but they are otherwise identical. (Note that these pins are only used for the ADCs; they are not a member of any port, and thus cannot be used for digital I/O.)

Answer by vicatcu

There is no difference as far as Arduino is concerned. I think the AU has a couple extra pins (analog ins), that you might want to route to header locations in a surface mount design, but if your design is based on an Uno, them you can also just leave those pins disconnected with no impact.
Uploading

Uploading is the process of updating or changing the sketch on the Arduino board. It does not refer to uploading to a server or a website.

Upload process:

Sketches are uploaded by avrdude.

The upload process is also controlled by variables in the boards and main preferences files. Those in the boards file include:

<BOARD>.upload.protocol: the protocol that avrdude should use to talk to the board (typically “stk500”).

<BOARD>.upload.speed: the speed (baud rate) avrdude should use when uploading sketches (typically “19200”).
<BOARD>.upload.maximum_size: the maximum size for a sketch on the board (dependent on the chip and the size of the bootloader).

And in the main preferences file:

upload.verbose: whether or not to dump debugging messages while upload a sketch to a board (defaults to “false”).

http://arduino.cc/en/Hacking/BuildProcess

Once the code is transferred to the board, then [if the board has auto-reset], the board will reset and the sketch will be run.
Questions

**Q:** Some dude named avr won’t let me upload my program, aka “avrdude: stk500_getsync(): not in sync: resp=0x00”

**Tags:** uploading (Prev Q) (Next Q), avrdude (Next Q)

I made an awesome program the other day, and I wanted to upload it to my Arduino. After clicking the upload button, some really mean dude named avr came along and stopped me. He told me:

```
 avrdude: stk500_getsync(): not in sync: resp=0x00
```

All I want to do is just upload my program, but avr won’t let me. He’s even unintelligible, so can someone tell me what the heck he’s trying to say? Is there any way I can ban him, arrest him, or simply get rid of him and his stupid message?

---

**Translation:**

Whenever I try to upload a program to my Arduino, I get this error message:

```
 avrdude: stk500_getsync(): not in sync: resp=0x00
```

What does this mean, and how can I fix it?

**Tags:** uploading (Prev Q) (Next Q), avrdude (Next Q)

**User:** the-guy-with-the-hat

---

**Answer** by the-guy-with-the-hat

This is caused by a connection error between your computer and the Arduino, and can result from many different specific problems. Here are some things that can fix this error:

- Disconnect and reconnect the USB cable.
- Use a different USB cable.
- Press the reset button on the board.
- Restart the Arduino IDE.
- Make sure you select the right board in Tools ► Board ►, e.g. If you are using the Duemilanove 328, select that instead of Duemilanove 128. The board should say what version it is on the microchip.
- Make sure you selected the right port in Tools ► Serial Port ►. One way to figure out which port it is on is by following these steps:
  1. Disconnect the USB cable.
  2. Go to Tools ► Serial Port ► and see which ports are listed (e.g. COM4 COM5 COM14).
  3. Reconnect the USB cable.
  4. Go back to Tools ► Serial Port ►, and see which port appeared that wasn’t there before.
In extreme cases, you may need to burn the bootloader.

Make sure digital pins 0 and 1 do not have any parts connected, including any shields.

**Answer** by ricardo

Unfortunately, it can also mean that you burned your microcontroller. Were you doing anything dangerous right before you tried to upload a new sketch? Is the microcontroller still working, with its previously loaded sketch?

**Answer** by thedoctor

If none of the above steps work, try reinstalling Avrdude and the avr-gcc compiler. I had to do this after my Mega started suffering from this.

If you installed Avrdude and avr-gcc separately, simply uninstall them. if not, you may have to reinstall the Arduino IDE. These steps will depend on your operating system.

**Tags:** uploading (Prev Q) (Next Q), avrdude (Next Q)

**Q: Can I corrupt my Arduino if it’s unplugged during an upload?**

**Tags:** uploading (Prev Q) (Next Q), avrdude (Prev Q) (Next Q), safety (Prev Q) (Next Q)

When messing around with code and uploading a sketch to my Arduino board, I almost accidentally pulled out the USB cord. Everything is fine, but then I started thinking what would’ve happened if I had pulled it out.

How safe is avrdude? Could it damage or brick the board? I’d imagine that you would just have to upload code to it again… but what would happen before you upload the new code? I’ve seen something about how you can always upload new code if the bootloader is intact, but I would imagine that there would be some problems. If this were to happen, should I unplug everything connected in case Arduino writes a HIGH to an input (or similar) that would damage the component?

**Tags:** uploading (Prev Q) (Next Q), avrdude (Prev Q) (Next Q), safety (Prev Q) (Next Q)

**User:** anonymous-penguin

**Answer** by ignacio-vazquez-abrams

AVRDUDE does not actually perform the write, it merely directs it. The actual writes are performed either by the bootloader or by the ISP firmware/hardware within the MCU itself.

With the bootloader it depends on how itself is coded, but at worst it would finish off the flash byte or page or EEPROM page zero- or one-filled and then fail to communicate further with AVRDUDE, aborting programming.

With ISP, the page write command would never be received and the programming would
be aborted at the last full page. Fuse bytes are done singly and so programming a single fuse can be viewed as an atomic operation.

In either case, a chip erase operation should reset the MCU to a known working state, fuse changes excepted.

---

**Answer** by **microtherion**

One safety feature that hasn’t been mentioned so far is that the ATmega328 has a set of lock bits that prevent read/write access to certain areas. To the best of my knowledge, the way Arduinos are set up, those bits are always set to prevent writing to the boot loader area unless the “Burn Bootloader” command is currently active, and since you can only use that command with an external programmer, you can always write a new bootloader if something goes wrong.

I think the only way of bricking the board (in the sense of leaving the MCU in a state that is functional, yet almost irreparably unresponsive) is writing bad fuse settings.

---

**Tags:** [uploading](Next Q) [avrdude](Prev Q) [safety](Prev Q) [Next Q)
I am new to the arduino world and wanted to start playing with the micro controllers, I am a .NET developer by trade so not completely new to this kind of world.

I purchased a seeeduino v3 and a Grove started kit (No soldering skills here) and can’t even get the basics to work.

I am setting up a simple button to light up the LED program. I have had it run once or twice but it seems really hit or miss on whether or not the Arduino IDE actually gets the program onto the seeeduino.

**Arduino IDE (1.5.6-r2) set-up:** Followed setup instructions found here: [SeeedDuino v3](#)

- Board: Arduino Duemilanove or Diecimila
- Processor: ATmega328
- Port: COM4 [this is what popped up once device was plugged in]
- Button is installed on D3
- LED is installed on D7

**Program:**

```
int button = 3;
int LED = 7;

void setup()
{
    pinMode(button, INPUT); //define button on INPUT devices
    pinMode(LED, OUTPUT); //define LED on OUTPUT device
}

void loop()
{
    int buttonState = digitalRead(button); //read the status of the button

    if(buttonState == HIGH) //also used (buttonState == 1)
    {
        digitalWrite(LED, HIGH); //also used digitalWrite(LED, 1)
    }
    else
    {
        digitalWrite(LED, LOW); //also used digitalWrite(LED, 0)
    }
}
```

**Upload Verbose Message:**

```
avrdude: verifying...
avrdude: 1070 bytes of flash verified
avrdude: Send: Q [51] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]

avrdude done. Thank you.
```

I have only had the LED from this program turn on twice, then reloading it ruins it.
SIDE NOTE

If I try and run Burn BootLoader I get this message-

```
avrdude: Version 5.11, compiled on Sep 2 2011 at 19:38:36
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2009 Joerg Wunsch

System wide configuration file is "C:\Program Files (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf"
Using Port : usb
Using Programmer : stk500v2
avrdude: usbdev_open(): did not find any USB device "usb"
```

Tags: uploading (Prev Q) (Next Q), bootloader (Prev Q) (Next Q)

User: cubicle.jockey

Answer by cubicle.jockey

OK mystery solved!

I ended up getting Visual Micro Debugger plug-in for my Visual Studio 2013 since I am native .NET programmer. I could see that the code was indeed loaded onto the board and when I pressed the button it was indeed getting to the line of code for setting the digital LED to HIGH.

It was a very unfortunate case of trying different components and learning I actually had 3 dead LEDs and one bad cable. After getting functioning ones it’s all good. :) unlucky me just had 4 bad parts in my kit.

Output from Visual Studio Debugging:

```
Skip code block
Legend:
line 12 = button state check
line 16 = button pressed turn on LED
line 20 = button not pressed turn off LED

=================OUTPUT====================
14:01:04.372 SeeedDuinoPlayground.ino, line 12 loop()
14:01:04.372 SeeedDuinoPlayground.ino, line 16 loop()
14:01:04.372 SeeedDuinoPlayground.ino, line 10
14:01:04.481 SeeedDuinoPlayground.ino, line 12 loop()
14:01:04.592 SeeedDuinoPlayground.ino, line 16 loop()
14:01:04.592 SeeedDuinoPlayground.ino, line 10
14:01:04.696 SeeedDuinoPlayground.ino, line 12 loop()
14:01:04.697 SeeedDuinoPlayground.ino, line 20 loop()
14:01:04.804 SeeedDuinoPlayground.ino, line 10
14:01:04.912 SeeedDuinoPlayground.ino, line 20 loop()
```

Tags: uploading (Prev Q) (Next Q), bootloader (Prev Q) (Next Q)

Q: Arduino compatible's serial port not showing Mac OSX

Tags: uploading (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q), avrdude (Prev Q) (Next Q)

I bought an Arduino-compatible Freaduino board atmega8 (I selected the board type as Arduino ng or older atmega8). I installed the Arduino IDE on Mac OS X and the FTDI
drivers. But my serial port looks like this when I connect the board:

And when I try to upload the program, this is the error I get:

Can someone tell me what is the mistake? I am clueless.

**UPDATE:** the problem was I had not installed the driver properly (I guess). I uninstalled and reinstalled them and it worked perfectly. Also made a YouTube tutorial regarding the same.  [http://youtu.be/BwatajQO_88](http://youtu.be/BwatajQO_88)

**Tags:** uploading (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q), avrdude (Prev Q)
From the OSX System Report we learn that your board is based on an SiLabs CP2102 converter (or workalike), but according to your question you installed the drivers for an entirely different product from FTDI

(Arduino.cc has used FTDI in the past and CDC/ACM today, I don’t believe they have ever used SiLabs so their instructions probably don’t cover that, but other vendors obviously have)

Install the CP2102 drivers and it should work if you use the /dev/tty.XX device which then appears when you plug in the board.

SiLabs appears to provide the drivers at


Just for sake of completeness, there are at least 5 distinct protocols/chips in use for serial-like USB communication with embedded devices:

- FTDI
- CDC/ACM Virtual Com Port
- SiLabs CP210x
- Prolific PL-2303
- ch340

(Additional possibilities could include actual RS232 serial or serial-over-bluetooth)

just to add to the list - some cheaper boards use CH340 chipset

here is a blogpost on how to install the drivers on OSX


From that post:
You can find drivers for this chip on the web site of the chinese manufacturer, here:

http://www.wch.cn/downloads.php?name=pro&proid=5

---

**Q: How do send sketch updates to devices without the editor?**

I’d like to update the sketch of a device that a friend out of state is testing. I could have them install the full editor and send the source code but I’d rather not require them to do
something so technical.

Is there a way to compile the firmware binary and push that out to a device over the COM port more directly?

**Tags:** uploading (Prev Q) (Next Q)

**User:** asawyer

---

**Answer** by ignacio-vazquez-abrams

In general, yes.

The compile routine creates a .hex file in a temporary directory that contains the flash image for the device. One can use AVRDUDE to upload this .hex file to the Arduino from the command line. It is simply a matter of getting the command arguments correct in order to tell it where and how to upload the image. Enabling verbose output in the IDE will tell you both exactly where the .hex file is located and what arguments are required for the board selected.

---

**Q: ERROR: Timed out waiting for Arduino 101 - anything I can do?**

**Tags:** uploading (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q)

I just received an Arduino 101 so I did the following in a wave of excitement…

1. Connected the USB cable between it and my Macbook Pro
2. Started the Arduino IDE version 1.6.7
3. Wrote a little bit of code
4. Selected Tools | Boards | Boards Manager… and installed the Intel Curie Boards as per Getting Started
5. Selected Tools | Board: "Arduino 101"
7. Selected Sketch | Upload

I was expecting something amazing, but I got the error:

```
Starting upload script
Payload: /var/folders/jr/5rmqcrqj58103d4p1f6_3sjr0900gn/T/build7c6ce067b9e8a57f9917c651ec494989.tmp/orientation.ino.bin
Waiting for device...
ERROR: Timed out waiting for Arduino 101.
```

Is there anything I can do to debug this or did I miss a step perhaps?

(I tried the good ol’ turn it off and on again, including shutting the IDE down).

**Tags:** uploading (Prev Q) (Next Q), arduino-ide (Prev Q) (Next Q)

**User:** kmp

---

**Answer** by nuke
Press the reset button next to the USB port. (either one, the master reset is between the DC jack and USB).

This will cause the Arduino 101 to reboot and enter DFU mode for 5 seconds if it is connected on USB. The download will proceed when the Curie enters DFU.

The IDE uses a baud-rate signal over USB. It is a virtual serial port, so baud rate doesn’t matter. But when the host side (Arduino IDE) switches the baud rate to 1200, the board will reboot and enter DFU mode for download. If it gets stuck for some reason and misses the message to reboot, you can just hit the reset button and accomplish the same thing.

---

**Q: Is it possible to upload previously compiled binary?**

**User:** lefty

**Answer** by 25mhz

Yes! Its possible that you can upload a previously complied binary, using

AVRDUDE - a command line utility which is actually used to upload binaries, behind the arduino IDE.

You can find AVRDUDE in Arduino folder > hardware > tools > avr > bin

This some documentation for AVRDUDE [http://www.nongnu.org/avrdude/user-manual/avrdude.html](http://www.nongnu.org/avrdude/user-manual/avrdude.html)

else simplest thing you can do is turn on verbose output during upload and watch for a command in the end with AVRDUDE in it along with the path of hex file, just copy paste & run the same command in CommandLine; your task is accomplished.

---
A library is a file that can be included into the code to add new instructions by combining many existing instructions and creating one easy to use instruction to call.

Here is the list of libraries:


**Standard Libraries:**

- EEPROM - reading and writing to “permanent” storage
- Ethernet - for connecting to the internet using the Arduino Ethernet Shield
- Firmata - for communicating with applications on the computer using a standard serial protocol.
- GSM - for connecting to a GSM/GRPS network with the GSM shield.
- LiquidCrystal - for controlling liquid crystal displays (LCDs)
- SD - for reading and writing SD cards
- Servo - for controlling servo motors
- SPI - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
- Stepper - for controlling stepper motors
- TFT - for drawing text, images, and shapes on the Arduino TFT screen
- WiFi - for connecting to the internet using the Arduino WiFi shield
- Wire - Two Wire Interface (TWI/I2C) for sending and receiving data over a network of devices or sensors.
- The Matrix and Sprite libraries are no longer part of the core distribution.

**Due Only Libraries:**

- Audio - Play audio files from a SD card.
- Scheduler - Manage multiple non-blocking tasks.
- USBHost - Communicate with USB peripherals like mice and keyboards.

**Esplora Only Libraries:**

- Esplora - this library enable you to easily access to various sensors and actuators mounted on the Esplora board.
Arduino Robot Library:

- Robot - this library enables easy access to the functions of the Arduino Robot

Arduino Yún Bridge Library:

- Bridge Library - Enables communication between the Linux processor and the Arduino on the Yún.

USB Libraries (Leonardo, Micro, Due, and Esplora)

- Keyboard - Send keystrokes to an attached computer.
- Mouse - Control cursor movement on a connected computer.

Contributed Libraries

If you’re using one of these libraries, you need to install it first. See these instructions for details on installation. There’s also a tutorial on writing your own libraries.

Communication (networking and protocols):

- Messenger - for processing text-based messages from the computer
- NewSoftSerial - an improved version of the SoftwareSerial library
- OneWire - control devices (from Dallas Semiconductor) that use the One Wire protocol.
- PS2Keyboard - read characters from a PS2 keyboard.
- Simple Message System - send messages between Arduino and the computer
- SSerial2Mobile - send text messages or emails using a cell phone (via AT commands over software serial)
- Webduino - extensible web server library (for use with the Arduino Ethernet Shield)
- X10 - Sending X10 signals over AC power lines
- XBee - for communicating with XBees in API mode
- SerialControl - Remote control other Arduinos over a serial connection

Sensing:

- Capacitive Sensing - turn two or more pins into capacitive sensors
- Debounce - for reading noisy digital inputs (e.g. from buttons)

Displays and LEDs:

- GFX - base class with standard graphics routines (by Adafruit Industries)
- GLCD - graphics routines for LCD based on the KS0108 or equivalent chipset.
- Improved LCD library fixes LCD initialization bugs in official Arduino LCD library
- LedControl - for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- LedControl - an alternative to the Matrix library for driving multiple LEDs with Maxim chips.
- LedDisplay - control of a HCMS-29xx scrolling LED display.
- Matrix - Basic LED Matrix display manipulation library
- PCD8544 - for the LCD controller on Nokia 55100-like displays (by Adafruit Industries)
- Sprite - Basic image sprite manipulation library for use in animations with an LED matrix
- ST7735 - for the LCD controller on a 1.8”, 128x160 TFT screen (by Adafruit Industries)

**Audio and Waveforms:** FFT - frequency analysis of audio or other analog signals
Tone - generate audio frequency square waves in the background on any microcontroller pin
Motors and PWM: TLC5940 - 16 channel 12 bit PWM controller.

**Timing:**
- DateTime - a library for keeping track of the current date and time in software.
- Metro - help you time actions at regular intervals
- MsTimer2 - uses the timer 2 interrupt to trigger an action every N milliseconds.

**Utilities:**
- PString - a lightweight class for printing to buffers
- Streaming - a method to simplify print statements
**Questions**

**Q: What are the steps involved in porting an Arduino library?**

I want to use the Adafruit CC3000 and its Arduino libraries on another platform. In this case it's the Arduino pin compatible PSoC4 Pioneer Kit. This is an ARM Cortex M0 based platform.

My question is: How would I go about porting a library like this? What are the steps, key files etc?

One obvious issue is that PSoC Creator does not readily support C++.

One positive might be that the Arduino Due board is based on the Atmel SAM3X8E ARM Cortex-M3 CPU.

NOTE: If you think this question is off topic; please head over to this question on Arduino Meta to voice your opinion on whether questions on Arduino Compatible devices are appropriate for this list.

---

**Answer** by chris-k

I did essentially the same thing with getting the Adafruit ST7735 library working in chipKIT. I started a support thread here which can lead you through my process.

I tackled it the same way I do general programming:

1-» Copy over Libraries, get it so your compiler can actually SEE them.
2-» Start trying to compile and then triage errors from the “top level” down:
3-» Look for stlib functions you recognize as should be working and fix those.
4-» Look for AVR specific code and comment those out. I normally also add my initials and a comment: ckck - bitbanging SPI, so I can search for these later.
Repeat 3 and 4 until it works or you quit. :)

Reach out to the forums for your target; they might have been there done that — or like in my case, you introduce them to something that they want, so they dive in and help.

---

**Q: Are there any signal smoothing libraries for Arduino?**

I am working on a mobile robot controlled via a wireless 2.4 GHz link. The receiver is connected to the Arduino Uno which serves onboard as the main controller. The most critical (and main) input channel coming from the receiver produces a very noisy signal,
which leads to lots of minor changes in the output of the actuators, even though these are not needed.

Plot of the input of the Arduino in a 30 second interval.

I am looking for libraries that can perform efficient smoothing. Are there any signal smoothing libraries available for the Arduino (Uno)?

**Tags:** library (Prev Q) (Next Q)

**User:** asheesh

---

**Answer**

*Microsmooth* is a lightweight signal smoothing library currently being developed by me. It is still being worked on and the aim is to make it lightweight in terms of memory and fast. The library provides multiple filters for smoothing:

- Simple Moving Average
- Exponential Moving Average
- Cumulative Moving Average
- Savitzky Golay Filter
- Ramer Douglas Pecker Algorithm
- Kalmogorov Zurbenko Filter

To use the library, download and add it to the source directory. Also, add the following line to your source file:
Answer by david-cary

I think I see a lot of single-sample noise spikes in your noisy signal.

The median filter does better at getting rid of single-sample noise spikes than any linear filter. (It is better than any low pass filter, moving average, weighted moving average, etc. in terms of its response time and its ability to ignore such single-sample noise spike outliers).

There are, in fact, many signal-smoothing libraries for the Arduino, many of which include a median filter.

signal-smoothing libraries at arduino.cc:

- Paul Badger: smooth digital low-pass filter
- Paul Badger: digitalSmooth digital low-pass filter with outlier rejection
- David A. Mellis and Tom Igoe: Smoothing tutorial
- Majenki: Average Library

signal-smoothing libraries at github:

- AsheeshR / Microsmooth
- jeroendoggen: Arduino-signal-filtering-library
- karlward: Arduino data filtering library
- sebnil: FIR-filter-Arduino-Library
- daphoosa: MedianFilter
- arc12: A Collection of Digital Signal Filters (intended for use with Arduino)

Would something like this work in your robot? (The median-of-3 requires very little CPU power, and hence fast):

```
#include "microsmooth.h"

/*
median_filter.ino
2014-03-25: started by David Cary
*/

int median_of_3( int a, int b, int c ){
    int the_max = max( max( a, b ), c );
    int the_min = min( min( a, b ), c );
    // unnecessarily clever code
    int the_median = the_max ^ the_min ^ a ^ b ^ c;
    return( the_median );
}

int newest = 0;
int recent = 0;
int oldest = 0;

void setup() {
    Serial.begin(9600);
    // read first value, initialize with it.
    oldest = random(200);
    newest = oldest;
    recent = oldest;
```
```c
recent = oldest;
newest = recent;
Serial.println("median filter example: ");
}

void loop()
{
    // drop oldest value and shift in latest value
    oldest = recent;
    recent = newest;
    newest = random(200);
    Serial.print("new value: ");
    Serial.print(newest, DEC);
    int median = median_of_3( oldest, recent, newest );
    Serial.print("smoothed value: ");
    Serial.print(median, DEC);
    Serial.println(" ");
    delay(5000);
}
```

**Answer** by **thedoctor**

Have you tried a low pass filter? I found an example [here](#) an another one [here](#).

Both of these libraries have a list of data being read from the analog sensor of your choice which is averaged. Every new sensor value is added to the list, and the last one is thrown out, like this:

| List: 3 4 3 3 4 3 5 3 2 3 4 3 |
| new reading added. old one thrown out |
|/--|
| List: 5 3 4 3 3 4 3 5 3 2 3 4 |
| list averaged |

**Tags:** library (Prev Q) (Next Q)

**Q:** How can I adapt the DallasTemperature library to accept another 1-Wire class?

**Tags:** library (Prev Q) (Next Q)

A commonly used library is the [DallasTemperature](#) library, used to access DS18B20 sensors on a 1-Wire bus. When creating a DallasTemperature object you pass a reference to a [OneWire](#) object. OneWire can create a 1-Wire bus on pretty much any port of an Arduino.

An example of this is like so:

```c
// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
```

However, there are some issues with using OneWire to drive a 1-Wire bus, mainly around driving longer bus lengths, dealing with shorts, and other errors.

There is a commonly available chip called the DS2482-100 that can drive longer busses
(hundreds of feet), deals with shorts and other errors. It is also less CPU heavy and it deals with a number of the slower aspects of 1-Wire.

For this reason, I would like to be able to pass a reference to a library that I have created instead. Currently, I am doing this by:

- Calling the library OneWire
- Replicating all the methods called by the DallasTemperature library
- Swapping out the OneWire library for mine

This is relatively messy. I would ideally like to be able to pass either a reference to my library or a 1-Wire library, so that people can chose which interface to use.

How would I go about doing this?

**Tags:** library (Prev Q) (Next Q)

**User:** cybergibbons

---

**Answer** by jfpoilpret

The typical way to achieve what you want would normally be to create your own **subclass** of OneWire class and **override** the methods you need to change, then pass a pointer to an instance of your class to the DallasTemperature constructor.

However, for this to work, the OneWire class must have been developed with extensibility (subclassing) in mind, i.e. use **virtual** methods that you can override.

Unfortunately, a quick look at OneWire.h shows that no method is virtual in there.

So that means in the end you will have to perform some hacks, just as you do now.

What you could potentially do (that’s still a hack, but maybe a bit less dirty) is:

- replace OneWire.h to make all methods virtual (at least those you need to override)
- keep the original OneWire.cpp that should still compile with the newly adapted header file
- create a new MyOneWire.h / MyOneWire.cpp to define your own OneWire subclass that overrides all necessary methods as you want

Then you can pass either a OneWire or a MyOneWire instance to DallasTemperature constructor.

NB: when I see my answer, this makes me think your question is more a general C++ question than an Arduino-related one.

**Tags:** library (Prev Q) (Next Q)

---

**Q:** What are the FLASH_ARRAY/FLASH_TABLE macros in this code?
In this code for heating monitor, there appears to be macros called `FLASH_ARRAY` and `FLASH_TABLE`. I can't find any reference to them either in this code, Arduino, or on the Internet. What do they do?

```
FLASH_ARRAY(byte, VARIABLEDEVICE_TABLE, 
  DEVICE_LADDNING_RETUR, 
  DEVICE_LADDNING_UT, 
  DEVICE_TANK1, 
  DEVICE_TANK2, 
  ... 
  VARIABLE_ID_LOAD, 
  VARIABLE_ID_TIME_EMPTY);
```

and

```
FLASH_TABLE(byte, NUMBER_TEMP_TABLE, 7, 
  {DEVICE_LADDNING_RETUR, 99, 5, 10, 7, MAIN_TYPE_TEXT_CELCIUS, 2}, 
  {DEVICE_LADDNING_UT, 99, 43, 10, 7, MAIN_TYPE_TEXT_CELCIUS, 2}, 
  ... 
  {DEVICE_UTE, 0, 55, 15, 7, MAIN_TYPE_TEXT_CELCIUS_WITH_NEGATIVE, 103}, 
  {VARIABLE_ID_EFFECT, 114, 29, 10, 7, MAIN_TYPE_TEXT_VARIABLE, 2});
```

The reason this library was developed is:

Storing static program data in flash/PROGMEM is a tricky part of Arduino programming. To save precious RAM, a novice user already at odds with unfamiliar C++ syntax must digest such daunting concepts as prog_char, PSTR(), PROGMEM, pgm_read_word(), etc. Even seasoned users get tripped up by the indirection and typecasting that are required to retrieve valid PROGMEM data. Add to that a couple of apparent bugs in the implementation, and it’s clear that PROGMEM is a complicated mess.

I have written a new library, Flash, which abstracts away most of this complexity. It provides new String, Array, Table, and String Array types that make ROM-based data collections as easy to use as “normal” types. Each overrides the C++ [] operator, so to extract individual elements one uses familiar array access syntax:

Specifically, these are macros and a simple helper class to make it easier to create arrays and tables of values in program memory.

The two macros are declared as so:

```c
#define FLASH_ARRAY(type, name, values...) 
  static const type name##_flash[] PROGMEM = { values }; 
  _FLASH_ARRAY<type> name(name##_flash, sizeof(name##_flash) / sizeof(type));

#define FLASH_TABLE(type, name, cols, values...) 
  static const type name##_flash[][cols] PROGMEM = { values }; 
  _FLASH_TABLE<type> name((const PROGMEM type *)name##_flash, sizeof(name##_flash) / sizeof(name##_flash[0]));
```

After further digging, this appears to be from Mikal Hart’s Flash library.
The helper functions include `size` and override the operator `[]` so that they can easily be accessed like so:

```c
FLASH_ARRAY[9]
```

I disagree with zmo - this is more than boilerplate. The area of PROGMEM causes a lot of questions from Arduino users, and it is very easy to do things wrong when creating arrays in PROGMEM.

---

**Answer** by **zmo**

The flash array is from Mikal Hart’s [flash library](https://github.com/MikalHart/flash) as you show, and what it does is that they declare an array or a matrix (table) object in the PROGMEM memory:

```c
#define FLASH_ARRAY(type, name, values...) 
static const type name##_flash[] PROGMEM = { values }; 
_FLASH_ARRAY<type>(name##_flash, sizeof(name##_flash) / sizeof(type));
```

which is an instance of the `_FLASH_ARRAY` class defined in that same file, which is a facility to access PROGMEM/Flash memory.

Basically, it’s a lot of boilerplate for what already does the `PSTR()` macro amongst other from `pgmspace.h`. I personally would avoid using that and prefer to use the original macros along with my own index and pointers.

BTW, except for some arduino IDE magic, I don’t think that can compile, as the `Flash.h` is not included.

EDIT:

To make another point against the `FLASH_TABLE/FLASH_ARRAY` boilerplate/overhead, since avc-gcc 4.8 has been introduced the `__flash qualifier` which is a replacement for the `PROGMEM` macro, simplifying the whole stuff:

```c
// to use string literals without having to cast
#define FSTR(X) ((const __flash char[]) { X })
// create string pstr in the .pgm.data address space
const __flash char* pstr = FSTR("foo");
```

and no more use of the `pgm_read_*` functions, you can now access the content of the flash memory using `*pstr`, as shown is [this thread](https://forum.arduino.cc/t/how-can-i-get-the-source-files-for-arduino-libraries/). In case you want to keep a code compatible with older version of avr-gcc, [here’s a post](https://forum.arduino.cc/t/how-can-i-get-the-source-files-for-arduino-libraries/12331) that offers macros to deal with both systems.

---

**Tags:** library (Prev Q) (Next Q), flash (Prev Q) (Next Q)

---

**Q: How can I get the source files for Arduino libraries?**

**Tags:** library (Prev Q) (Next Q)

I’ve been looking for a while inside Arduino’s files to find the core library. However, I haven’t found them yet. **Where on my hard drive can I obtain the core library files (.cpp and .h) necessary for all Arduino code?**
Looking around on Google, I couldn’t find a way to find the above files. I did see something about getting an .a file while Arduino is compiling, however, that isn’t what I want. I want to browse the code and look at functions such as delay to see how they would react when given the values not expected. (i.e. Will my code stall if I use `delay(-200);` by not being to exit the loop, or will it create a black hole and disrupt the space-time continuum?)

**Tags:** library (Prev Q) (Next Q)

**User:** anonymous-penguin

---

**Answer by ignacio-vazquez-abrams**

The base Arduino source files are in the Arduino installation, under `hardware/*/cores/`. Source for additional libraries included are in `libraries/`. Additional files may be found in the sketchbook under its `hardware/*/cores/` and `libraries/` directories.

---

**Answer by cybergibbons**

The code is also available online for those times you don’t have the IDE installed:

https://github.com/arduino/Arduino

---

**Answer by jrobert**

On a Mac, the path to the source code folder is:

`Arduino.app/Contents/Resources/Java/Hardware/Arduino/Cores/Arduino`.

I’d expect it would be the same from `/Java/` on down in other OSes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Date Modified</th>
<th>Size</th>
<th>Kind</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>_CodeSignature</td>
<td>May 17, 2013 9:40 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>MacOS</td>
<td>May 21, 2013 4:51 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>Resources</td>
<td>May 17, 2013 8:19 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>examples</td>
<td>May 17, 2013 8:20 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>hardware</td>
<td>May 17, 2013 8:20 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>arduino</td>
<td>May 17, 2013 8:20 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>bootloaders</td>
<td>May 17, 2013 8:20 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>cores</td>
<td>May 17, 2013 8:20 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>avr-lttc</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>File</td>
<td>--</td>
</tr>
<tr>
<td>Arduino.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>12 KB</td>
<td>C File</td>
<td>--</td>
</tr>
<tr>
<td>binary.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>CDC.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C File</td>
<td>--</td>
</tr>
<tr>
<td>Client.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>16 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>HardwareSerial.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>HID.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>16 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>IPAddress.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>IPAddress.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C File</td>
<td>--</td>
</tr>
<tr>
<td>main.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>new.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>new.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C File</td>
<td>--</td>
</tr>
<tr>
<td>Platform.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>Print.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>Print.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C File</td>
<td>--</td>
</tr>
<tr>
<td>Printable.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>Serial.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C File</td>
<td>--</td>
</tr>
<tr>
<td>Stream.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>Stream.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>Time.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>16 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>Udp.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>USBAPI.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>USBCore.cpp</td>
<td>Oct 16, 2013 3:09 pm</td>
<td>16 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>USBCore.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>USCore.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>WCharacter.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>WInterrupts.c</td>
<td>May 17, 2013 8:20 pm</td>
<td>12 KB</td>
<td>C Source File</td>
<td>--</td>
</tr>
<tr>
<td>wiring_analog.c</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Source File</td>
<td>--</td>
</tr>
<tr>
<td>wiring_digital.c</td>
<td>May 17, 2013 8:20 pm</td>
<td>8 KB</td>
<td>C Source File</td>
<td>--</td>
</tr>
<tr>
<td>wiring_private.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>wiring_pulse.c</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>wiring_shift.c</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C Source File</td>
<td>--</td>
</tr>
<tr>
<td>WMath.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>4 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>WString.cpp</td>
<td>May 17, 2013 8:20 pm</td>
<td>16 KB</td>
<td>C++ File</td>
<td>--</td>
</tr>
<tr>
<td>WString.h</td>
<td>May 17, 2013 8:20 pm</td>
<td>12 KB</td>
<td>C Hea... File</td>
<td>--</td>
</tr>
<tr>
<td>firmware</td>
<td>May 17, 2013 8:20 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
<tr>
<td>variants</td>
<td>Oct 16, 2013 3:11 pm</td>
<td>--</td>
<td>Folder</td>
<td>--</td>
</tr>
</tbody>
</table>

Tags: [library](Prev_Q) [Next_Q]
Q: How Do I Send An Email WITHOUT An Ethernet or WiFi Shield

Tags: library (Prev Q) (Next Q)

I can only find info relating to the wifi shield, but I want to send an email without the wifi shield. For now, I just plan on leaving my arduino plugged into a computer, so I just want to use the computer’s network connection. Is there a library or something I can grab for this?

Tags: library (Prev Q) (Next Q)

User: rys

Answer by peter-r.-bloomfield

If you mean plugged-in via USB, then no. There’s nothing you can do on the Arduino that would let you use the computer’s network connection directly.

You would need to have a program running on the computer which would communicate with the Arduino via serial. It would effectively act as an intermediary, sending the email on behalf of the Arduino.

Unfortunately, I suspect you’d have to write the program yourself. It’s possible that something similar already exists, but at the very least you’d probably have to modify it to suit your needs.

Tags: library (Prev Q) (Next Q)

Q: Problem importing a library in .h file

Tags: library (Prev Q) (Next Q), c (Prev Q) (Next Q)

I have following code:

MyApp.ino:

```c
#include "DfRobotLcdShield.h"

void setup() {
}

void loop() {
}
```

DfRobotLcdShield.h:

```c
#include <liquidCrystal.h>
```

When I try to compile, I’m getting the error:

```c
In file included from MyApp.ino:1:8:
c:\temp\build58561818774637812172.tmp\DfRobotLcdShield.h:4:27: fatal error: LiquidCrystal.h: No such file or directory
```

^  
compilation terminated. 
Error compiling.
When I add

```c
#include <LiquidCrystal.h>
```
directly to the .ino file, it works. I’m using the latest Arduino-IDE 1.6.3.

**Tags**: library (Prev Q) (Next Q), c (Prev Q) (Next Q)

**User**: thomas-s.

**Answer** by fuenfundachtzig

This seems to be a general problem of the Arduino IDE: It only recognizes libraries that are included in the (primary) .ino file.

If you look at the invocation of the compiler, the path to any given library is only added if this library’s header file is included in the .ino file.

I tested this with the EEPROM library. If this library is included in the .ino file, the compiler is called with

```
avr-g++ -c -g -Os -Wall -fno-exceptions -ffunction-sections -fdata-sections -mmcu=atmega328p -DF_CPU=16000000L ...
```

where the important part is the `-I/usr/share/arduino/libraries/EEPROM`

If I now remove `#include <EEPROM.h>` (but still have it in another header file which in turn is included in the .ino as in your case), the compiler call changes to

```
avr-g++ -c -g -Os -Wall -fno-exceptions -ffunction-sections -fdata-sections -mmcu=atmega328p -DF_CPU=16000000L ...
```

where now the include path to the library is missing.

**TL;DR**

If you need to use a library, just (also) add an include to the .ino file. It doesn’t cost you anything and will solve your problem ;)

---

**#include <file> vs. #include "file"**

From the [gcc documentation](https://gcc.gnu.org/onlinedocs/gcc/Include.html):

- `#include <file>` is used for system header files. It searches for a file named file in a standard list of system directories. […]
- `#include "file"` is used for header files of your own program. It searches for a file named file first in the directory containing the current file, then in the quote directories and then the same directories used for `<file>`. […]

---

**Tags**: library (Prev Q) (Next Q), c (Prev Q) (Next Q)

**Q**: Writing C Program outside of Arduino IDE?
I like the idea of using the Arduino IDE for simple projects and for getting started with Arduino, but the consensus I’ve gotten so far is that it is for those who are new to Arduino and/or programming in general. My understanding is that it is possible to write a C program from scratch, completely outside of the Arduino IDE, and then use a tool like AVRDUDE to upload it to an Arduino MCU. This option, albeit appealing and interesting, leaves me with a few concerns:

- What Arduino libraries would need to be imported/linked by such a “raw C” program? I assume that when an Arduino IDE-based program executes a `digitalWrite(...)` it is really calling a C lib, probably provided by Arduino, under the hood. I am concerned about making sure all these “underlying libs” get included with my C program. Thoughts?
- Is anything else “lost” by flying solo and venturing outside the Arduino IDE? Any capabilities/features that I would now have to “roll my own”?

Here you are two nice articles how Arduino actually works in background. How it removed friction from microcontroller programming.

1. A Tour of the Arduino Internals: How does Hello World actually work?
2. Arduino and GCC, compiling and uploading programs using only makefiles

It was also discussed here Is there a way or tutorial for converting Arduino code to C code?

You’re absolutely right about needing to link against other libraries. However, you don’t necessarily need to worry about all of the details yourself, unless you want/need to do something unusual. Several alternative IDEs already support Arduino, typically via plugins which handle a lot of the details automatically.

I normally recommend Eclipse as it’s a very widely used IDE for various purposes. However, there are several other possibilities. See this question for more information:

- What are the other IDEs for Arduino?
Apologies if this is the wrong sub-forum, the involvement of ATTiny and arduino places this into a strange category. Please correct me if I should post elsewhere.

It is important for this to be a library as small kids will be using this program and hardware for their own projects. Its kind of a rip on the LOLshield but it involves a lot more steps to integrate more lessons all at once.

My whole program takes 8 bytes of information and converts it to data for a shift register driving two multiplexers all driving an 8x8 LED matrix. It can also use multiple of these packets to make a slide show. I have it in a working state right now and would like to turn it into a library for easy use. I am a bit lost in this step as I had to create an interrupt timer and this makes it more complicated than the tutorial on the arduino website covers.

Also if someone could point me to a library for ATTiny85 that makes calling a timer interrupt easy then the rest of this project will likewise become easier.

Any help would be great! I will be continuing to look on online resources and will update should I figure it out.

---

**Code**

In the interests of answering the question though, this works:

```plaintext
ISR (TIMER1_COMPA_vect)
{
  digitalWrite (4, !digitalRead (4));  //toggle D4 (pin 3 on chip)
}

void setup()
{
  pinMode (4, OUTPUT);  // chip pin 3

  // Timer 1
  TCCR1 = bit (CTC1);     // Clear Timer/Counter on Compare Match
  TCCR1 |= bit (CS10) | bit (CS13); // prescaler of 256
```

---
OCR1C = 123;  // what to count to (zero-relative)
TIMSK = bit (OCIE1A);  // interrupt on compare
}
// end of setup
void loop()
{
  // other code here
}

You can tweak the time delay by changing both the prescaler and the counter in OCR1C. The prescaler gives you coarse tuning, which gets you into the ballpark of the delay time. The timer then counts up to give you the final delay.

**Prescaler values**

From the datasheet:

<table>
<thead>
<tr>
<th>CS13</th>
<th>CS12</th>
<th>CS11</th>
<th>CS10</th>
<th>Asynchronous Clocking Mode</th>
<th>Synchronous Clocking Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>T/C1 stopped</td>
<td>T/C1 stopped</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>PCK</td>
<td>CK</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>PCK/2</td>
<td>CK/2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>PCK/4</td>
<td>CK/4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>PCK/8</td>
<td>CK/8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>PCK/16</td>
<td>CK/16</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>PCK/32</td>
<td>CK/32</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>PCK/64</td>
<td>CK/64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>PCK/128</td>
<td>CK/128</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>PCK/256</td>
<td>CK/256</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>PCK/512</td>
<td>CK/512</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>PCK/1024</td>
<td>CK/1024</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>PCK/2048</td>
<td>CK/2048</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>PCK/4096</td>
<td>CK/4096</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>PCK/8192</td>
<td>CK/8192</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>PCK/16384</td>
<td>CK/16384</td>
</tr>
</tbody>
</table>

The Stop condition provides a Timer Enable/Disable function.

In my case I chose a prescaler of 256 which is or’ing together CS10 and CS13. The delay is then:

\[
125 \text{ ns} \times 256 \times 124 = 3.968 \text{ ms}
\]

Where 256 is the prescaler and 124 is what we are counting to. This is assuming an 8 MHz clock which gives a clock period of 125 ns.
Flashing more slowly

If you put an LED on pin 3 and want to see the flashing, you need it much slower. For example:

```
TCCR1 |= bit (CS10) | bit (CS11) | bit (CS12) | bit (CS13); // prescaler: 16384
```

That toggles the pin every 254 ms, which is visible.

Warning

On this chip Timer 1 is an 8-bit timer, so you cannot put more than 255 into OCR1C.

More information

- [Timers](#)
- [Interrupts](#)

Tags: library (Prev Q), attiny (Prev Q) (Next Q)
The Arduino processor chips have a number of timers/counters. Basically they are hardware that **count** things, however if they count pulses from the system clock (which runs at a known speed) they can also be used to **time** things.

Timers have an **input** which can be an external signal, or the system clock. If counting an external signal they can be used to count something like how many pulses are detected on a pin. If counting the system clock they can be used as an elapsed-time timer.

The input can go through a **prescaler** which divides down the input (eg. by 64). This lets you have the counter “tick over” more slowly than it otherwise might.

Timers are usually described as 8-bit, 10-bit, 16-bit and so on. The number of bits indicates the maximum the counter/timer can count to before overflowing. For example, an 8-bit counter can count to 256.

Timers can be configured to raise **interrupts** on various events, for example:

- Timer overflow
- A certain count has been reached

The timers can also be used to generate **PWM** pulses. PWM (pulse-width-modulation) can be configured to have a certain frequency, and a certain duty cycle, depending on the way the timer registers are set.
Questions

**Q: Does millis() conflict with the PWM pins associated with timer 0?**

I’ve read that the millis() function uses the same timer as a couple of PWM pins. If you’re using those PWM pins, will millis() still return the correct value?

**Answer** by ignacio-vazquez-abrams

They do not conflict as millis() strictly reads the immediate value in TCNT0 whereas PWM via timer 0 uses the hardware’s ability to compare the value of TCNT0 with the values in OCR0x without affecting the value of any of them.

**Answer** by madivad

Just to add to @Ignacio’s answer which has directly answered your question. The “conflict” you speak of is in relation to Timer0’s prescaler.

For the most part, you can use those pins (incidentally pins 5 and 6 on the UNO) with PWM without an issue, and read the correct value of millis() (as well as get the expected delay from delay)

Where you run into problems is if you want to change the prescaler (usually to modify the frequency of the PWM signal). If you change this, then you directly affect the calculation and reporting of millis() and the length of delay(x).

**Q: Is volatile needed when variable is accessed from > 1 ISRs, but not shared outside ISRs?**

It’s clearly documented that when global data is shared with an ISR and the main program, the data needs to be declared volatile in order to guarantee memory visibility (and that only suffices for 1-byte data; anything bigger needs special arrangements to guarantee also atomicity). Here we have good rules:

- Variables only used outside an ISR should not be volatile.
- Variables only used inside an ISR should not be volatile.
- Variables used both inside and outside an ISR should be volatile.

But is volatile needed when the variable is accessed from > 1 ISRs, but not shared
outside ISRs? For example, I have a function that maintains internal state using a static variable:

```c
void func()
{
    static volatile long counter; // volatile or not?
    // Do stuff with counter etc.
}
```

That function is called in two ways: from pin interrupt, and from TimerOne library:

1. `attachInterrupt(0, func, CHANGE);`
2. `Timer1.attachInterrupt(func);`

There are no atomicity problems, since *when an ISR is entered, interrupts are automatically disabled*, but this volatile is more of a compiler question: what is cached and what is not.

Better safe than sorry, of course...

**Tags:** timers (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

**User:** joonas-pulakka

---

**Answer** by jrobert

`volatile` only informs the compiler’s code generator that the variable may be modified by something other than the code being generated, so not to assume any copy of it remains accurate.

ISR code must be written/generated under the assumption that it has no context at entry, and preserve the CPU’s context around its (ISR’s) own operation. So, as with the indivisibility of non-atomic operations, volatility depends, *in this case*, on whether or not interrupts are allowed to nest. If non-nesting is guaranteed, the shared variable can not change by other than this ISR during its own execution. If your ISR might someday be used in an environment where interrupts are allowed to nest, then that constraint will no longer hold.

* *in this case*:
I’m assuming a software-maintained variable here. If we’re talking about a variable that can be updated by a hardware event, a timer register for example, all bets are off: the variable is volatile no matter what.

**Tags:** timers (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

---

**Q: Are function pointer assignments atomic in Arduino?**

**Tags:** timers (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

The following snippets are from TimerOne library source code:

Skip code block

```c
// TimerOne.h:
void (*isrCallback)();
```
TimerOne.cpp:
ISR(TIMER1_OVF_vect) // interrupt service routine that wraps a user defined function supplied by attachInterrupt
{
    Timer1 isrCallback();
}

// TimerOne.cpp:
void TimerOne::attachInterrupt(void (*isr)(void), long microseconds)
{
    if(microseconds > 0) setPeriod(microseconds);
    isrCallback = isr; // register the user’s callback with the real ISR
    TIMSK1 = _BV(TOIE1); // sets the timer overflow interrupt enable bit
    resume();
}

The question: if the timer is already running, and the main program calls attachInterrupt(), could the timer interrupt occur there during the function pointer assignment isrCallback = isr;? Then, with lucky timing, Timer1 isrCallback(); function pointer would consist partly of the old and partly of the new address, causing the ISR jump to a bogus location?

I suppose this might be the case, since function pointers are certainly wider than 1 byte, and accessing > 1 byte data isn’t atomic. Possible workarounds could be:

- Always call detachInterrupt() to make sure the timer isn’t running, before calling attachInterrupt(), i.e. clarify Timer1 docs.
- Or, modify Timer1, disabling timer overflow interrupts temporarily just before isrCallback = isr;

Does this make sense, or is there something in Timer1 sources or function pointer assignments that I’ve missed?

Tags: timers (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

User: joonas-pulakka

Answer by jrobert

Have a look at the code for attachInterrupt() and detachInterrupt() in /Applications/Arduino.app/Contents/Resources/Java/hardware/arduino/cores/arduino/WInterrupts.c (well, that’s where they are on a Mac, anyway. Arduino file structure on other OSes’ probably looks similar in the lower levels of the path).

It appears that attachInterrupt() assumes that the interrupt in question is not yet enabled because it writes the function pointer without taking any precautions. Note that detachInterrupts() disables the target interrupt before writing a NULL-pointer to its vector. So I’d at least use a detachInterrupt() / attachInterrupt() pair

I’d want to run any such code in a critical section, myself. It appears your first way (detach, then attach) would work, though I can’t be sure it couldn’t miss an unfortunately timed interrupt. The datasheet for your MCU might have more to say on that. But neither am I sure at this point, that a global cli()/sei() wouldn’t miss it either. The ATMega2560 datasheet, section 6.8, does say “When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example”, seeming to imply that it can buffer an interrupt while the interrupts are off.
**Q: When setting hardware timers as interrupts, should I prefer a lower prescaler value or a lower CTC?**

I’m currently playing with Arduino’s hardware timers, and a question came to my mind. Let me explain it a bit.

Let’s suppose I want a certain function to execute every 1024 clock ticks. AFAIK, I could achieve this in several ways, playing with the prescaler value for Timer2, and its CTC:

1. Setting the prescaler to 1 and the CTC to 1024
2. Setting the prescaler to 8 and the CTC to 128
3. …
4. Setting the prescaler to 1024 and the CTC to 1.

All these ways would achieve what I want, but, which one is more efficient? Or it doesn’t matter at all?

Thanks in advance for sharing your knowledge :P

**Tags:** timers (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

---

**Answer** by janakiram

If ‘clock tick’ count (here 1024), timer width and MCU clock used are constant (a fixed value), I would not go for last option i.e. prescalar 1024 and CTC of 1. Here are my reasons (specially for Arduino):

1) When CTC is close to zero (like 1-> 10), sometimes timer behaves as if count is zero and outputs a square wave (with 50% duty ratio) with half the demanded frequency. So make CTC reasonably big. But again selecting a high clock frequency (pre scalar 1) may also show reduced accuracy. A intermediate option I think is better.

2) When count CTC changes dynamically in a process, timer pre scalar should be selected such that CTC count will always be less than or equal to timer width over whole process. i.e. the maximum ‘timer tick count’ should be comfortably accommodated by the timer with it’s given width. Even here same dilemma again occurs, then selection based on point 1 is better.

If this is not enough, then only way I think is to implement each case physically and check which combination gives you the best results. Practical results I think are more reliable (may be you will get a board specific efficient combination!)

---

**Answer** by nick-gammon

There is a trade-off between resolution and achievable frequencies. You mention Timer 2, but on the Atmega328P Timer 2 is an 8-bit timer, thus you would not be able to set the
CTC to 1024.

Let’s assume we are talking about an 16-bit timer, like Timer 1 on the Atmega328P. With a prescaler of one, you can time (assuming a 16 MHz clock) from 1 to 65536 “ticks”, that is 62.5 ns up to 4096 µs.

This would be the most precise measurement because you are using one (processor) clock tick per timer tick (a prescaler of one).

However if you plan to time for more than 4.096 ms then you need to bump up the prescaler. The next prescaler up on Timer 1 is 8, so now you can time for an interval 8 times as long (32768 µs) however your accuracy (precision) has now decreased by a factor of 8. The granularity of the timer has increased from 62.5 ns to 62.5 * 8 ns, which is 500 ns.

If you need to time longer than 32.768 ms then the prescaler has to be larger again, the next one being 64. So now you can time up to 262144 µs, but with a granularity of 62.5 * 64, which is 4000 ns (4 µs).

My suggestion would be to use the lowest prescaler that you can, but still get the interval you want. So obviously you can’t use a prescaler of one to time 10 ms.


On that page is a chart which helps visualize the effects of different prescalers:
<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>16,000,000</td>
<td>62.5</td>
<td>0.0625</td>
<td>0.0000625</td>
<td>0.0000000625</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2,000,000</td>
<td>500</td>
<td>0.5</td>
<td>0.0005</td>
<td>0.000005</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>500,000</td>
<td>2,000</td>
<td>2</td>
<td>0.02</td>
<td>0.00002</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>250,000</td>
<td>4,000</td>
<td>4</td>
<td>0.04</td>
<td>0.00004</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>125,000</td>
<td>8,000</td>
<td>8</td>
<td>0.08</td>
<td>0.00008</td>
</tr>
<tr>
<td>256</td>
<td>1</td>
<td>62,500</td>
<td>16,000</td>
<td>16</td>
<td>0.16</td>
<td>0.0016</td>
</tr>
<tr>
<td>1024</td>
<td>1</td>
<td>15,625</td>
<td>64,000</td>
<td>64</td>
<td>0.64</td>
<td>0.0064</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>1,600,000</td>
<td>625</td>
<td>0.625</td>
<td>0.000625</td>
<td>0.000000625</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>200,000</td>
<td>5,000</td>
<td>5</td>
<td>0.005</td>
<td>0.00005</td>
</tr>
<tr>
<td>32</td>
<td>10</td>
<td>50,000</td>
<td>20,000</td>
<td>20</td>
<td>0.2</td>
<td>0.0002</td>
</tr>
<tr>
<td>64</td>
<td>10</td>
<td>25,000</td>
<td>40,000</td>
<td>40</td>
<td>0.4</td>
<td>0.0004</td>
</tr>
<tr>
<td>128</td>
<td>10</td>
<td>12,500</td>
<td>80,000</td>
<td>80</td>
<td>0.8</td>
<td>0.0008</td>
</tr>
<tr>
<td>256</td>
<td>10</td>
<td>6,250</td>
<td>160,000</td>
<td>160</td>
<td>1.6</td>
<td>0.016</td>
</tr>
<tr>
<td>1024</td>
<td>10</td>
<td>1,563</td>
<td>640,000</td>
<td>640</td>
<td>6.4</td>
<td>0.064</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>320,000</td>
<td>3,125</td>
<td>3.125</td>
<td>0.003125</td>
<td>0.00003125</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>40,000</td>
<td>25,000</td>
<td>25</td>
<td>0.25</td>
<td>0.00025</td>
</tr>
<tr>
<td>32</td>
<td>50</td>
<td>10,000</td>
<td>100,000</td>
<td>100</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>64</td>
<td>50</td>
<td>5,000</td>
<td>200,000</td>
<td>200</td>
<td>2</td>
<td>0.02</td>
</tr>
<tr>
<td>128</td>
<td>50</td>
<td>2,500</td>
<td>400,000</td>
<td>400</td>
<td>4</td>
<td>0.04</td>
</tr>
<tr>
<td>256</td>
<td>50</td>
<td>1,250</td>
<td>800,000</td>
<td>800</td>
<td>8</td>
<td>0.08</td>
</tr>
<tr>
<td>1024</td>
<td>50</td>
<td>625</td>
<td>3,200,000</td>
<td>3,200</td>
<td>3.2</td>
<td>0.032</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>160,000</td>
<td>6,250</td>
<td>6.25</td>
<td>0.00625</td>
<td>0.0000625</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>20,000</td>
<td>50,000</td>
<td>50</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td>32</td>
<td>100</td>
<td>5,000</td>
<td>200,000</td>
<td>200</td>
<td>2</td>
<td>0.02</td>
</tr>
<tr>
<td>64</td>
<td>100</td>
<td>2,500</td>
<td>400,000</td>
<td>400</td>
<td>4</td>
<td>0.04</td>
</tr>
<tr>
<td>128</td>
<td>100</td>
<td>1,250</td>
<td>800,000</td>
<td>800</td>
<td>8</td>
<td>0.08</td>
</tr>
<tr>
<td>256</td>
<td>100</td>
<td>625</td>
<td>1,600,000</td>
<td>1,600</td>
<td>1.6</td>
<td>0.016</td>
</tr>
<tr>
<td>1024</td>
<td>100</td>
<td>313</td>
<td>6,400,000</td>
<td>6,400</td>
<td>6.4</td>
<td>0.064</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>80,000</td>
<td>12,500</td>
<td>12.5</td>
<td>0.125</td>
<td>0.00125</td>
</tr>
<tr>
<td>8</td>
<td>200</td>
<td>10,000</td>
<td>100,000</td>
<td>100</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>32</td>
<td>200</td>
<td>2,500</td>
<td>400,000</td>
<td>400</td>
<td>4</td>
<td>0.04</td>
</tr>
<tr>
<td>64</td>
<td>200</td>
<td>1,250</td>
<td>800,000</td>
<td>800</td>
<td>8</td>
<td>0.08</td>
</tr>
<tr>
<td>128</td>
<td>200</td>
<td>625</td>
<td>1,600,000</td>
<td>1,600</td>
<td>1.6</td>
<td>0.016</td>
</tr>
<tr>
<td>256</td>
<td>200</td>
<td>313</td>
<td>3,200,000</td>
<td>3,200</td>
<td>3.2</td>
<td>0.032</td>
</tr>
<tr>
<td>1024</td>
<td>200</td>
<td>78</td>
<td>12,800,000</td>
<td>12,800</td>
<td>12.8</td>
<td>0.128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>256</td>
<td>62,500</td>
<td>16,000</td>
<td>16</td>
<td>0.16</td>
<td>0.00016</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>7,813</td>
<td>128,000</td>
<td>128</td>
<td>1.28</td>
<td>0.0128</td>
</tr>
<tr>
<td>32</td>
<td>256</td>
<td>1,953</td>
<td>512,000</td>
<td>512</td>
<td>5.12</td>
<td>0.00512</td>
</tr>
<tr>
<td>64</td>
<td>256</td>
<td>977</td>
<td>1,024,000</td>
<td>1,024</td>
<td>10.24</td>
<td>0.1024</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>488</td>
<td>2,048,000</td>
<td>2,048</td>
<td>20.48</td>
<td>0.2048</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
<td>244</td>
<td>4,096,000</td>
<td>4,096</td>
<td>40.96</td>
<td>0.4096</td>
</tr>
<tr>
<td>1024</td>
<td>256</td>
<td>61</td>
<td>16,384,000</td>
<td>16,384</td>
<td>163.84</td>
<td>1.6384</td>
</tr>
</tbody>
</table>

**Timer 1 only:**

<table>
<thead>
<tr>
<th>Pre-scalar</th>
<th>Count</th>
<th>Frequency Hz</th>
<th>Period nS</th>
<th>Period µS</th>
<th>Period mS</th>
<th>Period (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>16,000</td>
<td>62,500</td>
<td>62.5</td>
<td>0.0625</td>
<td>0.000625</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>2,000</td>
<td>500,000</td>
<td>500</td>
<td>5</td>
<td>0.005</td>
</tr>
<tr>
<td>64</td>
<td>1000</td>
<td>250</td>
<td>4,000,000</td>
<td>4,000</td>
<td>4</td>
<td>0.04</td>
</tr>
<tr>
<td>256</td>
<td>1000</td>
<td>63</td>
<td>16,000,000</td>
<td>16,000</td>
<td>16</td>
<td>0.16</td>
</tr>
<tr>
<td>1024</td>
<td>1000</td>
<td>16</td>
<td>64,000,000</td>
<td>64,000</td>
<td>64</td>
<td>0.64</td>
</tr>
</tbody>
</table>
The top part (count of one) effectively gives you the granularity of each prescaler. For example, a prescaler of 256 has a granularity of 16,000 ns (16 µs). Certain frequencies (powers of 2) will lend themselves to combinations (eg. prescaler of 1 with a count of 256, or prescaler of 256 with a count of 1).

However for frequencies that don’t have that property, the smaller prescaler will (if it can be used) give a finer granularity.

---

**Q: Setting timer3 in CTC mode - conflict with servo library**

I would like to set up a timer in order to call a function 800 times per second. I’m using the Arduino Mega and Timer3 with a prescaler of 1024. To choose the prescaler factor I’ve considered the following steps:

- CPU freq: 16MHz
- Timer resolution: 65536 (16 bits)
- Divide CPU freq by the chosen prescaler: 16x10^6/1024 = 15625
- Divide the rest through the desired freq 62500/800 = 19.
- Put the result + 1 in OCR3 register.

I’ve used the following table to set registers of TCCR3B:

<table>
<thead>
<tr>
<th>CSn2</th>
<th>CSn1</th>
<th>CSn0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No clock source. (Timer/Counter stopped)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>clk[OC]1 (No prescaling)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>clk[OC]8 (From prescaler)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>clk[OC]64 (From prescaler)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>clk[OC]256 (From prescaler)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>clk[OC]1024 (From prescaler)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>External clock source on Tn pin. Clock on falling edge</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>External clock source on Tn pin. Clock on rising edge</td>
</tr>
</tbody>
</table>

**The error**

It is impossible to compile the code. This is the error returned by the compiler:

Servo\Servo.cpp.o: In function ‘__vector_32’: C:\Program Files (x86)\Arduino\libraries\Servo\Servo.cpp:110: multiple definition of ‘__vector_32’
AccelPart1_35.cpp.o:C:\Program Files (x86)\Arduino\AccelPart1_35.ino:457: first
The code

```c
volatile int cont = 0;
unsigned long aCont = 0;
void setup()
{
    // initialize Timer3
    cli();  // disable global interrupts
    TCCR3A = 0;  // set entire TCCR3A register to 0
    TCCR3B = 0;  // same for TCCR3B

    // set compare match register to desired timer count: 800 Hz
    OCR3A = 20;
    // turn on CTC mode:
    TCCR3B |= (1 << WGM12);
    // Set CS10 and CS12 bits for 1024 prescaler:
    TCCR3B |= (1 << CS30) | (1 << CS32);
    // enable timer compare interrupt:
    TIMSK3 |= (1 << OCIE3A);
    // enable global interrupts:
    sei();
}

void loop()
{
    // Print every second the number of ISR invoked -> should be 100
    if (millis() % 1000 == 0)
    {
        Serial.println();
        Serial.print("tick: ");
        Serial.println(contatore);
        contatore = 0;
    }
}

ISR(TIMER3_COMPA_vect)
{
    accRoutine();
    contatore++;
}

void accRoutine()
{
    // reads analog values
}
```

How to solve the conflict with the servo library?

**SOLUTION**

Conflict solved using the following code. It compiles but the counter associated with the 800Hz timer doesn’t increment its value.

```c
volatile int cont = 0;
void setup()
{
    Serial.begin(9600);
    // Initialize Timer
    cli();  // disable global interrupts
}
TCCR3A = 0; // set entire TCCR3A register to 0
TCCR3B = 0; // same for TCCR3B

// set compare match register to desired timer count: 800 Hz
OCR3B = 20;
// turn on CTC mode:
TCCR3B |= (1 << WGM12);
// Set CS10 and CS12 bits for 1024 prescaler:
TCCR3B |= (1 << CS30) | (1 << CS32);
// enable timer compare interrupt:
TIMSK3 |= (1 << OCIE3B);
// enable global interrupts:
sei();

Serial.println("Setup completed");
}

void loop()
{
  if (millis() % 1000 == 0)
  {
    Serial.print(" tick: ");
    Serial.println(cont);
    cont = 0;
  }
}

ISR(TIMER3_COMPB_vect)
{
  cont++;
}

Since the main problem has been solved, I’ve created another question here related to the problem of the counter incrementation.

Tags: timers (Prev Q) (Next Q), compile (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q)

User: userk

Answer by brettam

Unfortunately, The Servo library reserves output compare A (OCR*A) on timers 1,3,4, and 5 when loaded on an arduino mega. Each can only have one ISR, so you will not be able to define your own TIMER*_COMPA_vect while you use Servo without modifying the library.

However, each hardware timer is equipped with 2 output compare registers. Servo does not claim any TIMER*_COMPB_vect interrupts, so those are free to use, and work exactly the same.

You should watch out for the Servo libraries activities, it could change the configuration of your timer. The default order is on megas is 5,1,3,4 and to give each 12 servos. It only configures the timer once it needs it, so you should be fine using timer 3 until you add that 25th servo.

To change your code, use OCR3B instead of OCR3A (the Output Compare Registers) and set bit OCIE3B instead of OCIE3A in TIMSK3 (the Output Compare Interrupt Enable bits). Then you to change your ISR function to ISR(TIMER3_COMPB_vect) {} 

CTC mode only works with OCR3A, but if you set TCNT3 to 0 in your interrupt function you can get similar behavior. Remember to remove the line that turns CTC mode on using
I’m trying to check the frequency of Timer3 using a counter. The value of the counter, declared as volatile, is incremented in the ISR and every second the sum is shown in the main loop and the value reset to zero.

The timer has been set up correctly. (If I choose a 3Hz timer I can see the led blinking)

The problem

The counter isn’t incremented. Here is the output:

Setup Completed
tick: 1
tick: 0
tick: 0
tick: 0

CODE

```
volatile int cont = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  // Initialize Timer
  cli(); // disable global interrupts
  TCCR3A = 0; // set entire TCCR3A register to 0
  TCCR3B = 0; // same for TCCR3B
  // set compare match register to desired timer count: 800 Hz
  OCR3B = 20; // 800Hz 5; // 3 Hz
  // turn on CTC mode:
  TCCR3B |= (1 << WGM12);
  // Set CS10 and CS12 bits for 1024 prescaler:
  TCCR3B |= (1 << CS30) | (1 << CS32);
  // enable timer compare interrupt:
  TIMSK3 |= (1 << OCIE3B);
  // enable global interrupts:
  sei();
  Serial.println("Setup completed");
}

void loop()
{
  if (millis() % 1000 == 0)
  {
    Serial.print(" tick: ");
    Serial.println(cont);
    cont = 0;
  }
}
ISR(TIMER3_COMPB_vect)
{
   //digitalWrite(13, digitalRead(13) ^ 1);
   cont++;
}

EDIT This timer is used to read an analog value from an accelerometer and store it in an array of float. But at the moment I’m stuck on this update issue.

SOLUTION 1 Thanks to Gerben

Skip code block
volatile int cont = 0;
void setup()
{
   Serial.begin(9600);
   pinMode(13, OUTPUT);

   // Initialize Timer
   cli(); // disable global interrupts
   TCCR3A = 0; // set entire TCCR3A register to 0
   TCCR3B = 0; // same for TCCR3B

   // set compare match register to desired timer count: 800 Hz
   OCR3A = 20; // 20; //800Hz 5; // 3 Hz
   // turn on CTC mode:
   TCCR3B |= (1 << WGM32);
   // Set CS10 and CS12 bits for 1024 prescaler:
   TCCR3B |= (1 << CS10) | (1 << CS12);
   // enable timer compare interrupt:
   TIMSK3 |= (1 << OCIE3B);
   // enable global interrupts:
   sei();
   Serial.println("Setup completed");
}
void loop()
{
   delay(1000);
   Serial.println(cont);
   cont = 0;
}
ISR(TIMER3_COMPB_vect)
{
   cont++;
}

SOLUTION 2 Thanks to BrettM

Skip code block
volatile int cont = 0;
void setup()
{
   Serial.begin(9600);
   pinMode(13, OUTPUT);

   // Initialize Timer
   cli(); // disable global interrupts
   TCCR3A = 0; // set entire TCCR3A register to 0
   TCCR3B = 0; // same for TCCR3B

   // set compare match register to desired timer count: 800 Hz
   OCR3B = 20; //800Hz 5; // 3 Hz
   // turn on CTC mode:
   //TCCR3B |= (1 << WGM32);
   // Set CS10 and CS12 bits for 1024 prescaler:
   TCCR3B |= (1 << CS10) | (1 << CS12);
   // enable timer compare interrupt:
   TIMSK3 |= (1 << OCIE3B);
```cpp
    // enable global interrupts:
    sei();
    Serial.println("Setup completed");
}

void loop()
{
    Serial.println(cont);
    cont = 0;
    delay(1000);
}

ISR(TIMER3_COMPB_vect)
{
    TCNT3 = 0;
    cont++;
}
```

**Tags:** timers (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q), interrupt (Prev Q) (Next Q), variables (Prev Q) (Next Q)

**User:** userk

---

**Answer** by gerben

In CTC mode the top is OCR3A, not OCR3B!

After that TIMSK3 |= (1 << OCIE3B); should also be changed to TIMSK3 |= (1 << OCIE3A);, and ISR(TIMER3_COMPB_vect) to ISR(TIMER3_COMPA_vect)

For 3Hz, OCR3A should be 5208, not 20.

Technically TCCR3B |= (1 << WGM12); should be TCCR3B |= (1 << WGM32);

---

**Answer** by brettam

It seems my answer to [this question](#) was previously incomplete, thanks for pointing out that CTC mode only works with OCR3A Gerben. I apologize for not testing an answer before I post it.

Given the information only in this question Gerben’s answer is complete, but since your other question implies that you cannot use OCR3A due to the Servo library I’ll add a bit. (I’ve also edited that answer)

you can emulate the behavior of CTC mode by setting TCNT3 to 0 in your interrupt routine. Remember to remove the line that turns on CTC mode in your code.

I’ve tested your code with this ISR:

```cpp
ISR(TIMER3_COMPB_vect)
{
    TCNT3 = 0;
    cont++;
}
```

and this configuration of the timer registers

```
OCR3B = 5208; // 800Hz 5; // 3 Hz
// Set CS10 and CS12 bits for 1024 prescaler:
TCCR3B |= (1 << CS30) | (1 << CS32);
// enable timer compare interrupt:
TIMSK3 |= (1 << OCIE3B);
```
This might be a bit less accurate at high frequencies than CTC, I’m not sure, but at 3Hz it worked perfectly. Notice that 5208 was the correct OCR value, not 20 (again thanks to Gerben).

Q: Adjust time calculation after Timer0 frequency change

Tags: timers (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q), interrupt (Prev Q) (Next Q), variables (Prev Q) (Next Q)

I have an Arduino Nano with an 328P and need all 6 PWM pins. Thus, I had to adjust the prescaler and WGM Mode of Timer0. It is now in phase correct PWM mode with a prescaler of 1.

```
TCCR0A = _BV(COM0A1) | _BV(COM0B1) | _BV(WGM00);
TCCR0B = _BV(CS00);
```

Now I need a working time calculation for other libraries, but since Timer0 had that duty everything is out of order now.

I tried adjusting the wiring.c

```c
// the prescaler is set so that timer0 ticks every 64 clock cycles, and the
// the overflow handler is called every 256 ticks.
#define MICROSECONDS_PER_TIMER0_OVERFLOW (clockCyclesToMicroseconds(64 * 256))
```

to this

```
#define MICROSECONDS_PER_TIMER0_OVERFLOW (clockCyclesToMicroseconds(1 * 510))
```

But it’s like I didn’t change anything. (tested other settings that were changed so it was compiled anew)

Whole Code:

```c
Skip code block
```
Fixing the timekeeping functions with your PWM settings is not so simple. You should at least try to rewrite ISR(TIMER0_OVF_vect), micros(), and probably delay(). Here is why:

First, there is a rounding problem. Time is kept using two global variables:

```c
volatile unsigned long timer0_millis;
static unsigned char timer0_fract;
```

The first one is what millis() returns. The second one keeps track of how much time has passed since the last full millisecond, and it does so in units of 8 µs. The two variables are incremented by ISR(TIMER0_OVF_vect) like this:

```c
m += MILLIS_INC;  // temporary copy of timer0_millis
f += FRACT_INC;   // temporary copy of timer0_fract
```

On a normal Uno configuration, the ISR is called every 1024 µs. Then MILLIS_INC is 1 and FRACT_INC is 3. With your timer configuration, the ISR is called every 31.875 µs (510 cycles), then MILLIS_INC should be 0 and FRACT_INC should be 3.984375. But since we are dealing with integers, it will be rounded down to 3, and your millis() will tick about 25% too slow.

A simple fix would be to

```c
#define MICROSECONDS_PER_TIMER0_OVERFLOW (clockCyclesToMicroseconds(1 * 512))
```

in order for FRACT_INC to be 4 and millis() to be 0.4% too fast. Or you could make timer0_fract a 16-bit variable and have it count clock cycles, just to avoid this error. Either option should fix millis(), but you still have a problem with micros().

micros() works by reading both timer0_overflow_count (incremented by 1 in the ISR) and the actual counter value. Since your counter is now going alternatively up and down, it will be harder to compute a microsecond count from these readings. Maybe you could take two consecutive readings of the counter, just to know whether it is going up or down...

And then there is delay(), which relies on micros(). If you fix micros(), delay() should work fine. If not, You could rewrite delay() to use millis() instead, which should be easy but you will lose some accuracy.
I’m taking my first wobbly steps outside the Arduino IDE and I’m not having much success with timers / interrupts. I can set a pin to output and light an LED with registers fine but I cannot for the life of me get it to blink using a timer. I have tried numerous tutorials and followed the Atmel ATmega328 datasheet very closely.

Using an Arduino Uno R3 & Atmel ICE (ISP.) My dev system is Raspbian (Debian) with the GNU AVR toolchain (avr-gcc, avr-objcopy, avrdude.) Apart from not having a bootloader it’s a bog standard board (including fuses.)

Here’s my current code:

```
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(TIMER0_COMPA_vect)
{
    PORTB ^= (1 << PB5); // Toggle Arduino Pin #13
}

int main (void)
{
    DDRB = (1 << DDB5); // Arduino Pin #13 is Output
    cli(); // Clear Interrupts
    OCR0A = (unsigned char)0xFF; // Compare Register A = 255
    TIMSK0 = (1 << OCIE0A); // Enable Interrupt for Comp. Reg. A
    TCCR0A = (1 << WGM01); // CTC Mode
    sei(); // Set Interrupts
    TCCR0B = (1 << CS02) | (1 << CS00); // Divide by 1024 Prescaler (GO!)
    return 0;
}
```

I don’t know if I’m just not seeing something or if I’ve set registers in the wrong order but it’s driving me crazy.

Tags: timers (Prev Q), atmega328 (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)

User: ashlyn-black

---

**Answer** by gerben

return 0 resets the device. Use while(1); instead, to make it wait indefinitely.

---

**Answer** by nick-gammon

Returning from main does not reset the device (it would start up again and do it all over in that case). It calls exit which turns interrupts off and loops indefinitely.
You can see at address 0x74 it calls main, and then jumps to exit. exit turns off interrupts. With interrupts off your ISR will not toggle pin 13.

**Tags:** timers (Prev Q), atmega328 (Prev Q) (Next Q), interrupt (Prev Q) (Next Q)
Power

Questions

Q: What are (or how do I use) the power saving options of the Arduino to extend battery life?

Tags: power (Prev Q) (Next Q), battery (Prev Q) (Next Q)

For solar & battery powered projects, decreasing power consumption is a necessity.

I know a little bit about using a timer and interrupt to put the microcontroller into sleep when it isn’t doing anything.

I have also read that you can disable some peripherals to further save power.

Here is a page explaining these: Power Saving Techniques

My question:

- What other techniques are there to save power?
- Are there any libraries that make using these AVR features easier?

Tags: power (Prev Q) (Next Q), battery (Prev Q) (Next Q)

User: sachleen

Answer by ricardo

Note: I wanted to point out that, even though my answer was accepted, the reader (you) should really read Anindo’s answer first if you really want to save energy with any Arduino board. Once you address those items in your design, then you should read my answer and worry about how to set the MCU into low power mode.

Having said that, there are several techniques to make an Arduino save power. I’ll list a few and then point you to a page that explains them all in more detail.

1. While the controller isn’t doing anything important (between one read of a sensor and the next, for example), you can put the controller into one of the sleep modes below, with the command set_sleep_mode (SLEEP_MODE_PWR_DOWN). Next to each mode is the approximate power consumption of each mode.

   - SLEEP_MODE_IDLE: 15 mA
   - SLEEP_MODE_ADC: 6.5 mA
   - SLEEP_MODE_PWR_SAVE: 1.62 mA
   - SLEEP_MODE_EXT_STANDBY: 1.62 mA
   - SLEEP_MODE_STANDBY: 0.84 mA
   - SLEEP_MODE_PWR_DOWN: 0.36 mA
2. Disable **brown-out detection** (the circuitry that turns off the controller when low voltage is detected).

3. Turn off ADC (analog to digital conversion)

4. Use the internal clock

Then, when you put the controller to sleep, you need to use one or more mechanisms below to wake up the controller and do something with it:

- Wake up with a signal
- Wake up with a timer

This is a summary I made from -


That article applies mostly to ATmega328P, but the technique applies to other Arduino compatible controllers as well. As TheDoctor said well, you will need to check the datasheet to make sure your controller supports any of those techniques and how to do it more precisely.

---

**Answer** by [anindo-ghosh](https://www.instructables.com/member/anindo-ghosh)

Realistically, the biggest power waster on an Arduino board is the linear voltage regulator on it.

1. As long as the microcontroller on the board, the LEDs, or any other peripherals, draw current, the linear regulator wastes power equal to difference between supply and board voltage x current drawn.

   So, a first fix would be to disconnect the power indicator LED on the board, and not use any of the other LEDs as far as possible. Second, supply the board with as low a voltage as possible that is just sufficient to power the on-board regulator.

2. On the original designs, the voltage regulators used do not boast low quiescent current. This means even with nothing drawing power within the board, the regulator itself wastes a fair bit of power all the time it is on.

   An easy fix is to replace the on-board regulator with an LDO (low drop-out linear regulator) rated for extremely low quiescent current. Parametric searches on various vendor sites will yield likely substitutes.

3. Even with the above steps, the Arduino board does not provide a mechanism to set the LDO into low-power mode, if the LDO chosen supports this. Power efficient designs typically use a “sleep mode” pin on voltage regulators to save a fair bit of energy - not an option here.

Even with all possible power saving modes and tricks applied at the microcontroller level, the Arduino simply is not designed to be an ultra-low-power device out of the box. Having
experimented with several non-trivial applications, I have found that the best that is realistically achievable is some 10% to 30% power saving depending on application, since the voltage regulator and the LEDs are going to consume the rest anyway.

---

**Answer** by **thedocto**

You could read through the 200-page datasheet, and then mess with some confusing bit shifts and registers, but I recommend this library: [http://playground.arduino.cc/Code/Enerlib](http://playground.arduino.cc/Code/Enerlib)

Also, if you’re using an Uno or any one with a usb-to-serial chip, you could disable that or remove it.

---

**Tags**: power (Prev Q) (Next Q), battery (Prev Q) (Next Q)

---

**Q: Why is the regulator very hot?**

---

**Tags**: power (Prev Q) (Next Q)

We have an LED strip connected to output 6 and is powered by the Uno board itself. Shouldn’t be drawing too much current, only have 10 elements on the LED strip. While this LED strip is connected to the Uno, I noticed that the regulator is getting very hot.

One thing we haven’t done yet is to power the LED strips from an external source, which will be necessary since there will be many strips. At the moment, the students are simply trying to debug their program with the small strip.

---

**Tags**: power (Prev Q) (Next Q)

**User**: chris-o

---

**Answer** by **ricardo**

When a circuit is hot, it’s very likely that it is drawing too much current. It may be a component that’s getting more current than it’s designed for, or a PCB track, or a wire.

So, before you do anything else, I strongly suggest that you get a DMM and measure the current drawn by the board when powering the led strip. Google for ways to measure current with a DMM, as it can be tricky if you never done it before.

Then, compare the value to those specified by the voltage regulator datasheet. The current should be within those values shown in the “Electrical Characteristics”, and well below those from “Absolute Maximum Ratings”.

---

**Answer** by **jippie**

If you power the Arduino with an external power supply the on board regulator has to *dissipate* the voltage difference between input and 5VDC at the total current you draw from the board.

As Klaus Warzecha wrote in an [answer](http://playground.arduino.cc/Code/Enerlib) to [another question](http://playground.arduino.cc/Code/Enerlib)
The Arduino is just the brain, not the power station.

If the output voltage of your power supply is 12V, then the regulator will be dissipated 12-5V = 7V at the total current drawn. Total dissipated power for eg. 150mA (just an estimate) would be P = V × I = 7 × 0.150 = 1W. The regulator will be pretty hot from dissipating 1W.

Mind you that the current is not limited to 150mA, enough LEDs can easily increase current and power. If the regulator is stressed (often) enough, sooner or later it will fail. It is better practice to power any load directly from the external power supply, taking the higher voltage into account.

---

**Answer** by butzke

A regulator normally get hot when is driven a current near to the maximum that could be driven, and its a normal thing the regulator keep warm.

A normal regulator like LM7805 could work until 125 C°.

If you try to drive more current that it can probably it will burn, so, maybe its time to you change your power supply to an external one to keep safe your Arduino board.

---

**Q: Powering Arduino with car batteries**

**Tags:** power (Prev Q) (Next Q)

Presumably, one powers an Arduino board with car batteries for long-term operation away from the power grid.

How can this be done, and what are the most (1) economic, and (2) reliable ways to do this? Specifically, do any additional risks to short the board with an extraneous discharge exist?

**Tags:** power (Prev Q) (Next Q), battery (Prev Q) (Next Q)

**User:** 0xf2

**Answer** by alexan_e

Use a switching regulator such as the LM2596. You can get a ready board from eBay for about US$1.
Just set the output voltage to 5 V and feed the $V_{cc}$ input of Arduino; this bypasses the onboard regulator.

The benefit is that unlike the linear onboard regulator, a switching regulator is very efficient which means less consumption and less generated heat on the regulator. An additional benefit is that it can withstand a higher input voltage (about 35 V for the specified regulator), just in case the circuit is used in a vehicle that has 24 V batteries.

The regulator has already several protections like for short circuiting, overheating, etc.

---

I’m a big fan of USB car chargers for things like this, or USB buck converter boards that you get from ebay.
There is also the higher-end Akafugu breadboard power supply - takes input from 7V all the way to 35V, but costs 11$ apiece in the bargain.

Q: Powering Arduino with Solar Panels

Clearly, an Arduino can be powered with solar panels.

1. What is the most optimal configuration?
2. Can you recharge and discharge at the same time?
3. Is LiPO the best battery cache technology, or is there something more durable out there, for fielded deployment?
4. What are the most economic, and reliable ways to do this?
First of all you want to buy some solar panels or cells. Although they can be found on eBay for very cheap, but they not be as efficient as they say. I would say try to buy from a reputable dealer to see reviews on the product or trust that they work efficiently. Sparkfun datasheets are generally accurate and can be trusted.

- Most optimal configuration: **Do not use a regulator and some sort of step-up booster.** Try the TPS61200 or TPS61202 chips. They will not regulate the voltage, they boost the voltage dynamically. This is much more efficient and therefore generates less heat. They work like a reverse voltage regulator, taking .3V to 5.5V and turning it into anywhere from 1.8V to 5V (xxx0) or 5V exactly (xxx2). Hack a USB B end and just connect the power lines to the Arduino.

- Discharge/charge simultaneously: Yes, however I’m not sure how to. I believe that if you hook up a charger IC and a load, the voltage won’t go through the battery, it will go through the path of least resistance. However, your battery wouldn’t get charged then. This is outside the scope of the question, so I would recommend EE on how exactly to do this.

- Is LiPO the best: **Try a Lithium-Ion Battery.** (A.K.A. Li-ion) According to this site:

> Although the lithium-polymer battery is sleeker and thinner, lithium-ion batteries have a higher energy density and cost less to manufacture.

Therefore, it seems like Li-ion is the best choice for you. **Note: Since you are dealing with very dense batteries that if overvolted can go up into flames [Note: please mute your speakers!]**, make sure to go with Sparkfun/Adafruit/etc…

- Economic way: If you don’t mind a big “homemade” solar panel, go this way. Buy through eBay or a similar site solar cells. They are small plates that are connected to create a solar panel. Try looking for “grade B” or similar solar cells that work fine, they just may have minor imperfections. It won’t be as pretty, but you’ll save some cash.

See this EE.SE post for more information. One cool thing that I saw was:

This is really cool. You can connect the Li-Ion voltage through an op-amp to generate a pulse wave, which can be picked up by your MPU to tell your bot to go charge – Raaj Oct 18 ‘13 at 18:44

I’m not exactly sure how to do this, but it might be useful. I couldn’t find anything online about it.

---

**Answer** by eternityforest

I would actually suggest LiFePo4 for anything DIY. Not only is it safer so one false move won’t cause an inferno, but it will last two to four times as long at the CONST of a little extra size and weight.

Since you are just running an arduino, I would use only 1 single cell, and avoid cell
balance woes.

I would use a boost regulator to step up the voltage from the cell to the required 5v.

To actually charge the cell, I would probably use a cheap ebay CC-CV charger board, and set the voltage for about 0.15V below the max for your lifepo(you only lose 10% of the energy but You make the battery last much longer), and set the current as appropriate for your panel and battery.

If your circuit is drawing power while the charger is charging, your circuit will take what it needs and any extra the charger puts out will go into the battery.

You might need a diode, to make the battery not discharge back through the charger.

Use a protection pcb like you can get from batteryspace, or better yet a protected lifepo with the pcb built it.

Remember to size your solar panel so that there is enough energy to keep the circuit running. Even with the protected cells overdischarge kills.

What I’d really like to do is design a lifepo4 charge control board that had all this in one unit since everyone seems to want it…

Tags: power (Prev Q) (Next Q)
I’m working on a medium sized robot project with multiple arduinos to control stuff. I’ve been planning on using a lead-acid battery pack, but then I was given 10 working and identical Asus laptop battery packs rated at 14.8 V 4400 mAh, and one old Asus laptop.

I opened up one of the battery packs, because it was bad and it has a 4S2P battery configuration and a protection board.

Can I hook all of these battery packs in parallel, or will power leak between the battery packs, and is that be a problem?

Is there alternatively some 4S20P or 3S25P battery pack with charger I can purchase to put these batteries in?

(I do have a step-down converter that will provide the Arduinos with the right voltage, regardless of input voltage up to 36 V)

I tried to ask a similar question on electrical engineering, but ot was deemed off topic for the site.

In case you didn’t know it already you can hook any batteries in parallel by adding a diode on every battery to avoid current from any other battery flowing back to it. Like this:

```
+----||--->|---+
|    |          |
+----||--->|---+----o V++
|
V GND
```

The drawback is that you have a small drop of voltage on the diodes and of course a small power loss too. If for example you’re drawing 0.5 Amp from the batteries and you’re using an 1N4001 diode which has a voltage drop of about 0.7 Volts then expect 0.350 Watts of power loss (0.5A x 0.7V). So choose the diode with the lowest voltage drop that can handle the current you need. There are solutions to lower the power that is wasted but they’re more complex.
[multipack parallel] … batteries are assembled by a company that is experienced and certified to test and assemble battery packs. The individual batteries are tested and sorted by machine so that each pack has matching batteries with the same capacity and internal resistance. Individuals do not have this equipment, which is why you should not try to make your own packs.

Are you certain all your batteries are the same age and have the same usage history? A brand-spanking new one will try to charge an old worn battery to its own fresh voltage.

---

**Q: Arduino powering from 9V battery**

Yesterday afternoon I have left my Arduino running on a fresh Alkaline 9V battery (+ connected to Vin, - connected to GND). Today morning (16 hours later) the LCD display was no longer visible, and when I measured the volatage, I saw the battery was depleted to my surprise - its idle voltage was only 7 V.

My device is Arduino Uno with LCD 2x16 with LCD backlight and 2 DS18B20 sensors. Is it normal for such device to deplete 9V battery this quick?

I have read an answer to What are (or how do I use) the power saving options of the Arduino to extend battery life? question and I think what I see might be related to this, however I am not sure. If it is, is there some common schematics how to connect Arduino to be powered by a battery efficiently?

---

**User:** suma

---

**Answer** by geometrikal

**Power consumption**

The Arduino boards use a fair bit of power compared to other embedded systems with similar functionality.

There are three main factors:

- The NCP1117 (datasheet) 5V linear regulator in the Arduino UNO R3 (schematic) has a quiescent current of around 6mA.
- The ATMega328P (datasheet) draws around 5mA @ 8MHz and 5V, and probably more than double that at 16MHz.
- user2973: The ATMega16U2 used for USB communications also draws approximately 13mA.

LEDs and other peripherals also draw some current. In your circuit, the LCD backlight is
probably drawing 4mA as well.

When dropping 9V to 5V via a linear regulator, almost half of the power is lost by the regulator due to its 4V drop. Duncan comments that this nearly doubles the quiescent power draw from 9V as well as the power needed for every mA of 5V, since 4/9ths of the power gets wasted as heat by the voltage regulator. An efficient switching regulator would put out 5V with little wasted energy, effectively reducing the current draw seen by the battery by 4/9ths.

A duracell 9V battery (data sheet) drops from 9V to 7V in around 7.5 hours with a 50mA current draw. Therefore, a rough guess is that your circuit draws around 25mA, which sounds about right based on the description of your circuit.

Note, alkaline battery life is non-linear with respect to current. For very small currents (<1mA) the life of an Alkaline approaches that of a lithium battery.

**Getting current down**

Here are some tips to get the current consumption down:

- **Regulator:** Replace the regulator with one with a low quiescent current, or better yet, a switching regulator (also with low quiescent current). A switching regulator uses ‘pulses’ of current and some external inductor and capacitors to give a reasonable steady voltage output. It doesn’t waste energy like with the voltage drop of linear regulator and efficiency in the high 90% are possible.
  - There are buck (step down) converters that take the battery as input, then connect directly to 5V and GND, bypassing VIN and the regulator. This one from Pololu both steps up and down, and has 0.1mA quiescent current.
  - Alternatively, you could use some 1.5V alkaline batteries, and a boost (step up) converter to get the voltage up to 5V (e.g. this product from Sparkfun). It seems boost converters are more commonly stocked.
  - Finally, you could buy a rechargeable lithium battery with charging shield. This advantage of this is not having to buy new batteries, and for a tiny bit bigger than a 9V battery the lithium one has much greater capacity. A really cool product is the seeeduino stalker waterproof kit that includes a charging circuit, battery, solar panel and other goodies.

- **ATMega328P:** Rather than using delay for timing and spinning in loop endlessly waiting for something to happen, re-write your code so that it goes to sleep in between sensor reads, etc. There are a few low power libraries out there that use the watchdog timer for periodic wakeup from sleep which are handy. You can get the current consumption of the ATMega328P below 0.1mA during sleep.

- **LCD:** Turn off the backlight, or even the entire LCD. Add a button to the design that the user can push to activate the LCD and have it turn off after a set amount of inactivity.

- **Peripherals:** Most peripheral chips also have a sleep mode that drastically reduces their power consumption. Remove power LEDs and other indicators that are not necessary.
• **ATMegu16U2: user2973** comments It seems this chip is quite power hungry (*user2973*). It could be removed to save power and just use the UART instead, but that seems like overkill. There are Arduino Pro boards that are just the bare bones Arduino without the USB interface that could be used instead of the UNO.

• **Batteries**: Other alkaline cells have much greater capacity. For example, a 1.5V AA has over 2000mAh for low currents. Using AA cells plus a boost converter can increase the time before battery replacement. Use D cells (16000mAh) and it will run for quite a while. :D

**Summary**

With appropriate power supply and coding you can get a reasonable amount of life out of a battery. Using the above principles, I have made an Arduino derivative board that measures a few sensors and stores the readings to an SD card every half a second. It can last for about 4 months on 2 AA batteries, so it is definently possible to have low power and remain in the Arduino ecosystem.

The chip I used for the low power in my board is the LTC3525-3.3V. It takes an input voltage as low as 0.8V and boosts up to 3.3V and a 5V version is also available. I designed a PCB for this chip as there was no ready-made breakout, and in the datasheet there are reference designs. The main criteria for choosing this chip was it still had high efficiency at very low currents. Some other converters need a small minimum current draw.

The biggest consumer of power on the board ended up being the micro SD card. It can vary between 0.1mA and 1.5mA of idle current depending on the manufacturer. I have found Verbatim and Lexar cards to consume the least amount of power. I will [this EE.SE question](https://electronics.stackexchange.com/questions/30453/how-can-i-decrease-battery-consumption) up to date with the results of my micro SD card power consumption tests.

---

**Answer** by [franklin](https://electronics.stackexchange.com/users/2352/franklin)

I think an Arduino Uno is not suitable for such projects. Some components on the board draw too many amps, as @geometrikal correctly points out. If you’re up to a challenge, I would advise you to take your project to another level and go barebones.

Sparkfun has a good article about how to increase battery life, using an ATmega328 that you also have on your Arduino Uno: [https://www.sparkfun.com/tutorials/309](https://www.sparkfun.com/tutorials/309)

Look for ways to create a circuit with only the components that you really need, and have your microcontroller sleep as much as possible.

---

**Tags**: [power](https://electronics.stackexchange.com/questions/tagged/power) *(Prev Q)*, [battery](https://electronics.stackexchange.com/questions/tagged/battery) *(Prev Q)* *(Next Q)*
ATMega328

Questions

Q: I got a missignatured ATMega328-PU. How can I fix that?

Tags: atmega328 (Prev Q) (Next Q)

Sometime in the past, I was burning bootloaders on a fresh batch of four ATMega328-PU using the Arduino IDE (notice there’s no P after 328 - it’s the slightly cheaper non-picopower version of the MCU, not to be confused with the ATMega328P-PU with a P), and was surprised with the following message from avrdude:

```
avrdude: Device signature = 0x1e950F
avrdude: Expected signature for ATMEGA328 is 1E 95 14
Double check chip, or use -F to override this check.
```

That means avrdude thought the chip was not what its label said. Then I changed the chip type on my Arduino IDE to ATMega328P-PU and avrdude burned the bootloader without complaints. What that means is that the chip was labelled as one MCU and internally it responded as another, slightly different.

What I would like to know is:

- How rare is this event? Has anyone had a similar experience? (Original question, off-topic)
- Is it possible to fix this? How can I fix the signature so that avrdude recognize the chip correctly?

This is a cross-post from EE.SE. I posted this question there but didn’t attract much attention, so I wanted to see if anyone from our community had a similar experience.

Tags: atmega328 (Prev Q) (Next Q)

User: ricardo

Answer by jippie

Not the preferred way to fix things and certainly not the first solution to consider, but you can consider programming the signature bytes. Before attempting this be absolutely sure you really want to do this and you have investigated what is involved to undo this. This may involve changing configuration files on your computer ...

Anyways the way to set the controller’s signature bytes is as follows (untested, I don’t have a spare AVR lying around):

```
avrdude -p atmega328 -c arduino -P /dev/ttyUSB003 -b 19200 -v -U signature:w:0x1E,0x95,0x14:m
```

Answer by thedoctor
While browsing sparkfun, I have found several news posts that show their struggle with mislabeled chips. Here are a few:


Sparkfun had received a questionable shipment if ICs from a new seller in China. They decided to test them before sending them into production, and none of their test boards worked. Using nitric acid, they were able to remove the case of the ICs and foul a hunk of metal that looked like copper.

In another article, they dissected some suspicious ICs from Atmel, and found an ON semiconductor silicon wafer inside. The chips were not functional ATmegas, but they did have silicon, unlike the other ones.

---

**Q: Can I use an ATmega328 alone without anything and what is the max voltage I can power the atmega328 with?**

**Tags:** atmega328 (Prev Q) (Next Q)

Okay 2 questions. If I am to replace the Arduino with an ATmega328 programmed with a standard blink on pin 13, is this configuration correct, do I still need oscillators and whatnot?
Second question is, what is the max voltage I can power the ATmega328 with without frying it? I have seen conflicting answers of this on the net. Some have said 5V while some say 6V can someone clarify?

**Tags:** atmega328 (Prev Q) (Next Q), hardware (Prev Q) (Next Q)

**User:** suckms

---

**Answer** by ignacio-vazquez-abrams

The [datasheet](#) describes the electrical and clock requirements of the chip, and [AVR042](#) describes the external hardware configuration required. Decoupling, analog voltages, external crystal or oscillator if required.

---

**Answer** by greenonline

If you take a look at the [Standalone Arduino](#) on the Arduino website, then this is what you require:
Note that there is one slight error in the guide, which I have highlighted here, Arduino Standalone - photo shows incorrect pin wired to MOSI.

There is a video on how to achieve, more or less, the same thing, 1-Day Project: Build Your Own Arduino Uno for $5.

Answer by nick-gammon

I have a page describing How to make an Arduino-compatible minimal board. The minimal setup would be:

That excludes the circuitry for uploading your code.

By the way, in the image in the question you have an LED with no resistor. That will damage both the LED and the Atmega328. You need a current-limiting resistor.

```

  do I still need oscillators

Not if you program the fuse to use the internal oscillator.

  Some have said 5v while some say 6v can someone clarify?

The datasheet says 1.8 to 5.5 V operating range with an absolute maximum rating of 6 V.

This is what the datasheet says about absolute maximum ratings:

  This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is
not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Page 2 of the datasheet says:

- Operating Voltage:
  - 1.8 - 5.5V

---

**Q: What are the implications of running an ATmega328P at 8 MHz and 3.3 V with the Arduino system?**

**Tags:** atmega328 (Prev Q) (Next Q), hardware (Prev Q) (Next Q)

I’m working on a custom arduino that will run at 3.3 V with no regulators or USB chips (for power consumption). I’ve heard that overclocking the chip at 16 MHz/3.3 V is generally fine, but I’d like to stay within spec anyway.

First: to run at 8 MHz, do I just need to replace the crystal? Or is there anything else I have to do?

Second: Do I need to make any code changes to reflect the new speed? I’m using serial, SPI, and I2C connections in this project, along with the NilRTOS operating system and millis/delay functions. Will any of these be broken? Will the code run noticeably slower?

Third: How can I program the chip? Is it ok to program it in an Arduino Uno at 16 MHz, then transplant it to the custom arduino?

Fourth: Will power consumption change at 8 MHz?

Thanks!

**Tags:** atmega328 (Prev Q) (Next Q)

**User:** vulcan

---

**Answer** by amadanon-inc.

The formula seems to be volt*5.9-6.6=mhz (valid between 1.8v and 4.5v), so at 3.3v, this would give 12.8mhz. Be aware that, if you are running at 12.8mhz, ANY drop below 3.3v MAY cause problems (and very hard to diagnose, too!) or you MAY get away with it.

However, Arduino Atmega328p chips have a “brown-out” detection set at 4.3v - any drop below that, the chip shuts down. You can get an ISP (AKA ICSP) to change the fuses - go to [http://www.engbedded.com/fusecalc/](http://www.engbedded.com/fusecalc/) to work out what to set fuses to. You can also use the ISP to read the current fuses, or indeed reprogram the chip without an Arduino board - all my projects have a 6-pin header. If you do this, before reprogramming using the Arduino board, you will have to “Burn Bootloader” (in the tools menu). Note that programming can still be done via Arduino IDE.
Alternatively, if you want to go “light” (and have an ISP), you can change the fuses to use the internal 8mhz clock - this frees up 2 extra pins, too! Be aware, however, that the internal clock may drift 1.2 hours per day (=5%), compared to 1.7 seconds per day (=20 parts per million) on a typical quartz crystal. If you don’t use it for timekeeping, this probably doesn’t matter.

You will need to make changes to the timing, for your code - you need to add a boards.txt with correct settings for your chip/clockspeed etc - you can copy the settings from the entry for the board, change the name, and change the setting “build.f_cpu”; restart (or start) your Arduino IDE, go to “Tools->Board” and select the board you added.

You can move the chip on/off the board, and reprogram it there (note that the board selected in Tools->Board is the FINAL board, not the programming board). Be careful plugging/unplugging it - eventually, you WILL bend the pins! Alternatively, you can use ISP (ICSP), as mentioned above - much recommended. ISP programming will also work if the chip is factory default, so long as it has a working clock.

How much power is used by the chip depends hugely on what you do with it - not just clock speeds, clock source, etc but also sleep modes. If you are sleeping a lot, it may draw less when running faster than running slower - if almost all your time is asleep, then the power consumption should be counted per clock cycle, not per second - if your clock is half the speed, and half current, then you draw the same current per instruction. Since the rest of your time is spent asleep, you get no benefit. See http://www.gammon.com.au/power for a very good breakdown of power usage, tips on how to save power, etc.

If you slow your clock down too much, serial and i2c may become unreliable (unless you slow down the speed), but I would expect to be able to go down to 1mhz before this to be a problem - a 1mhz chip trying to do 9600bps has 104 instructions per bit of data transferred. i2c runs at 100khz, so you get 10 instructions per bit (at 1mhz) - possibly pushing it. 8mhz should be fine. Test, test, test.

UPDATE: There are 3 possible settings for the brown-out detection - there are 3 possible values for the AtMega328p: disabled, 1.8v, 2.7v, and 4.3v. I was under the impression (above) that Arduinos were normally set to 4.3v; apparently that is not the case (see comment below). I recall seeing somewhere that there was brownout detection. This is something to keep in mind, if your arduino resets, especially when putting larger loads on the battery (running motors, multiple leds etc).

Tags: atmega328 (Prev Q) (Next Q)
**Q: Usefulness of measuring 0V using ADC**

I’m looking at the datasheet of the ATMega328, and I see the different channels you can select for AD conversion. ADC0..7, ADC8 (temperature), 1.1V \(V_{BG}\), and lastly 0V (GND).

- What would be the use of measuring 0V?
- Would it ever result in a reading other than 0? I find it very intriguing.

The datasheet doesn’t seem to give any additional information about it.

**AVR126** seems to give a hint:

configure 0V (GND) in ADC by setting Mux bitfield (MUX3:0) equal to 1111. This is done to discharge the capacitor of ADC.

Though I find the document rather strange (incomplete, lacking certain info, running the ADC at 500kHz while the datasheet says to not exceed 200kHz).

Is it to protect the ADC from to high a voltage in the ADC capacitor, when changing to a different (lower) reference voltage? Rather odd you need software to protect the IC, and it’s not even mentioned in the datasheet. Also I’ve never seen anyone do this (e.g. when measuring the internal temperature of the ATMega).

**Tags: atmega328 (Prev Q) (Next Q), analogread (Prev Q) (Next Q)**

**User: gerben**

**Answer** by **majenko**

That quote you mention says it all really. It’s purpose is to discharge the Sample And Hold capacitor.

That capacitor is used (as I’m sure you are aware) to store the incoming voltage while it’s being sampled by the ADC. The ADC operated in two main phases:

1. Acquisition
2. Sampling

During acquisition the capacitor is connected to the incoming voltage. During sampling it is disconnected from the incoming voltage and the voltage it stores is read into the internal registers.

It takes a certain amount of time for the voltage in the capacitor to equal that of the incoming voltage. If you try reading at higher speeds then what voltage was in the capacitor from the previous sample can affect the new reading. However, if you do each sample always from a known voltage in the capacitor (0V) then you get much more stable results.
Another good use for it is when using the ADC as a Charge Time Measurement Unit. This is the proper way of performing capacitative sensing. You apply a constant (tiny) current to an external capacitor (the sensing plates) for a specific period of time. When that time is up you disconnect the constant current source and measure the voltage that has developed across that capacitance. The capacitances and currents are small, and if there is a charge in your S&H capacitor at the time you do the sampling that will be back-fed into the touch sensor capacitative plates changing the voltage. By ensuring that the S&H capacitor is at 0V before you sample the plates you won’t be adversely affecting the voltage on the plates with successive readings (well, you will, but it will be consistently in the same way every time so it won’t affect your overall results).

Answer by nick-gammon

Though I find the document rather strange (incomplete, lacking certain info, running the ADC at 500kHz while the datasheet says to not exceed 200kHz).

The document actually says:

The ADC can prescale the system clock to provide an ADC clock that is between 50kHz and 200kHz to get maximum resolution.

(My emphasis)

It goes on to say:

If ADC resolution of less than 10 bits required, then the ADC clock frequency can be higher than 200kHz. At 1MHz it is possible to achieve eight bits of resolution maximum.

I did some testing on this page with different prescalers. The results showed that reasonable results could be obtained even with a 1 MHz ADC clock:

Skip code block

Prescaler 2
Analog port = 0, average result = 1023
Analog port = 1, average result = 1023
Analog port = 2, average result = 1023
Analog port = 3, average result = 1022
Time taken = 26220

Prescaler 4
Analog port = 0, average result = 673
Analog port = 1, average result = 718
Analog port = 2, average result = 512
Analog port = 3, average result = 193
Time taken = 32780

Prescaler 8
Analog port = 0, average result = 842
Analog port = 1, average result = 677
Analog port = 2, average result = 509
Analog port = 3, average result = 34
Time taken = 46840

Prescaler 16
Analog port = 0, average result = 1022
Analog port = 1, average result = 672
Analog port = 2, average result = 509
Analog port = 3, average result = 0
Time taken = 73164

Prescaler 32

Analog port = 0, average result = 1022
Analog port = 1, average result = 672
Analog port = 2, average result = 508
Analog port = 3, average result = 0
Time taken = 128040

Prescaler 64

Analog port = 0, average result = 1022
Analog port = 1, average result = 672
Analog port = 2, average result = 508
Analog port = 3, average result = 0
Time taken = 240972

Prescaler 128

Analog port = 0, average result = 1022
Analog port = 1, average result = 672
Analog port = 2, average result = 508
Analog port = 3, average result = 0
Time taken = 448108

The four test voltages were, and their results should have been:

- 5V (should return 1023)
- 3.3V (should return 674)
- 2.5V (should return 511)
- 0V (should return 0)

The tests were done in rapid succession, no allowing for the ADC to recharge or anything:

```
Skip code block
unsigned long startTime = micros ();
for (int i = 0; i < ITERATIONS; i++)
{
    for (int whichPort = lowPort; whichPort <= highPort; whichPort++)
    {
        int result = analogRead (whichPort);
        totals [whichPort - lowPort] += result;
    }
}
unsigned long endTime = micros ();
```

Tags: atmega328 (Prev Q) (Next Q), analogread (Prev Q) (Next Q)

Q: Why were Atmel MCUs used for Arduino?

Tags: atmega328 (Prev Q)

Atmel microcontrollers are not cheap compared to STM and Microchip. Why did Arduino select Atmel MCUs such as atmega328 and atmega2560 on their board out of so many alternatives?
Simply because the Wiring system that Arduino stole was written for the Atmel chips. Arduino didn’t “invent” the Arduino system - they just took an existing system called Wiring and adapted it slightly and branded it Arduino.
Questions

**Q: Is there a preprocessor constant for memory size**

I would like my compiles to exclude some code depending upon the flash/program space available. To be used in the demo code of a library I support. Whereas my current solution is to use a #ifdef of processor type. But I would like to be more general and not have to specify all the various chips. Rather it would be nice if there was a pre-processor constant that stated the available size.

Where I am not fluent enough in gcc and or avr-gcc I might expect to find some constants similar those that define the beginning and end of the heap.

On case example. Is that both the UNO and Leo have 32K of Flash. But the Leo’s core library uses 4K for USB support, resulting in only 28K available. My Library’s demo is near max’ed out on the UNO and I would like to automatically trim out based on available program space.

**Answer** by **jippie**

Find the io__.h file for your microcontroller, on Linux it is located in /usr/lib/avr/include/avr, on Windows it will be in a somewhat similar location.

Scroll down to the part that says /* Constants */. There are couple interesting macros defined there, FLASHEND being the one you should be interested in. You can use it for example as follows:

```c
#if FLASHEND > 0x8000
/* Include some extra code when sufficient flash is available. */
#endif
```

**Q: Can I compile c/c++ code on the linux part of the Arduino Yun?**

Can I compile c/c++ code on the linux part of the Arduino Yun?
How complete is the linux part of the Arduino Yun? Can I scp some c/c++ code onto the Atheros AR9331 chip compile it?

Or must I first crosscompile all software and then put them onto the Atheros?

**Tags:** compile (Prev Q) (Next Q), arduino-yun (Prev Q) (Next Q), linux (Prev Q) (Next Q)

**User:** johan

**Answer** by twod

The Yun’s OS (Linino) is based on OpenWRT, and the official toolchain from OpenWRT does not appear to have changed much. It could probably be done, the question is if you’d really want to though.

The limited resources (storage, RAM, CPU) means you would most likely not fit all the parts of the toolchain for C/C++ compilation (perhaps with the exception of very basic programs without many dependencies). The YUN’s package manager (opkg) should contain pre-built package if they exist, but given that even the list of packages is only kept in RAM to conserve space, I doubt the entire toolchain with libraries needed to compile useful stuff would fit.

The official way to build things for OpenWRT appears to be cross-compilation on a much beefier machine, if only because it takes a fraction of the time needed by a native compiler. All in all, you’re also very likely to save time and resources by cross-compiling and have more resources left on the YUN itself.

**Tags:** compile (Prev Q) (Next Q), arduino-yun (Prev Q) (Next Q), linux (Prev Q) (Next Q)
Q: Arduino IDE #ifdef

Tags: compile (Prev Q), arduino-ide (Prev Q) (Next Q)

I use my Arduino IDE to either upload my sketch to a Arduino or ATTiny or ATmega328. As you know each device can have a different pinout. Does the Arduino compiler support ifdef, depending on the board I am connected to?

For example

```c
#ifdef Attiny85
    a=0; b=1; c=2;
#else
    // arduino
    a=9; b=10; c=11;
#endif
```

Answer by anonymous-penguin

Yes. Here is the syntax:

```c
#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)  
    //Code here
#endif
```

You can also do something like this for the Mega:

```c
#elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)  
    //Code here
#endif
```

Assuming the implementation for the ATtiny is correct, your code should be like this:

```c
#if defined (__AVR_ATtiny85__)  
    a=0; b=1; c=2;
#else
    //Arduino
    a=9; b=10; c=11
#endif
```

Tags: compile (Prev Q), arduino-ide (Prev Q) (Next Q)
Shields

Questions

Q: Is there a color LCD display for an Arduino?

I currently have a Mega2560 with the LCD4884 Shield. The Shield is good for putting out text and simple pixel images. But I am wondering if there is something that has a full range of colours that would work with the Arduino? The screen size would only have to be a few inches in size.

Answer by jvarhol

There are actually several shields that have color screens on them. Many of the shields are just regular mini TFT displays, however there are also touch screen ones.

Here is one from Adafruit that includes a MicroSD holder and a Joystick.

Here is one from Radio Shack that features a 2.8 inch screen with touch capabilities

As far as screen size, many of the screens are only about an 1 inch across, however the biggest I have ever been able to find are 2.8 in.

Answer by akellyirl

There’s also a 2.2” Adafruit TFT.

The nice things about the Adafruit displays are: i) library support for both the display and the graphics core library. So it’s easy to get it up and running.

The built in frame buffer reduces resource requirements on Arduino.

Answer by hichris123

Another very cool screen is the official Arduino TFT LCD Screen. (Datasheet if you’re curious.)

It’s a 1.77” screen, with a resolution of 160x128. It has a micro-SD slot in it, which is easily accessible. It is a color screen, with up to 18 bit per pixel.

The extremely nice thing with this screen is that since it’s officially supported, you don’t need to do anything extra to support it. The official Arduino TFT library works perfectly with it. However, according to Jameco:
Library Update: There is a known issue with the TFT library supplied in IDE versions 1.0.5 and 1.5.4 Beta. You will need to replace the library. We have zipped up the updated TFT library for both versions available here. Just delete the existing library and copy in the replacement for the appropriate IDE version and restart the IDE. (IDE v1.0.5 or v1.5.4)

So beside that minor fix, everything should be easy and quick to work with. This page has some code samples which should make it easy to work with.

Tags: shields (Prev Q) (Next Q), lcd (Prev Q) (Next Q)

Q: 2.4” TFT LCD Shield isn’t working on Arduino Mega

Tags: shields (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q), lcd (Prev Q) (Next Q)

Even on ebay’s website it is mentioned that I can’t use 2.4” TFT LCD Shield display on attach to Arduino Mega. The problem is that I bought this shield by mistake. I want to put this shield onto Arduino Mega 2560. Is there a way to combine Mega and 2.4” Display Shield?

note: I tried on my friend’s Arduino Uno. Shield is working very good.

note: The photo below is determining my question. The display not runs my Arduino’s code. It only runs its LED.

```
// UTFT_Demo_320x240 (C)2012 Henning Karlsen
// web: http://www.henningkarlsen.com/electronics

// This program is a demo of how to use most of the functions
// of the library with a supported display modules.
// This demo was made for modules with a screen resolution
// of 320x240 pixels.
// This program requires the UTFT library.

#include <UTFT.h>
define ILI9320_16 18
```
// Declare which fonts we will be using
extern uint8_t SmallFont[];

// Uncomment the next line for Arduino 2009/Uno
//UTFT myGLCD(UNO_24,A2,A1,A3,A4);  // Remember to change the model parameter to suit your display

// Uncomment the next line for Arduino Mega
UTFT myGLCD(ILI9320_16,38,39,40,41);  // Remember to change the model parameter to suit your display

void setup()
{
    randomSeed(analogRead(0));
    // Setup the LCD
    pinMode(A0,OUTPUT);  // for the UNO_SHIELD_1IN1
    digitalWrite(A0,HIGH);  // the RD pin must be set high
    myGLCD.InitLCD();
    myGLCD.setFont(SmallFont);
}

void loop()
{
    int buf[318];
    int x, x2;
    int y, y2;
    int r;

    // Clear the screen and draw the frame
    myGLCD.clrScr();
    myGLCD.setColor(255, 0, 0);
    myGLCD.fillRect(0, 0, 319, 13);
    myGLCD.setColor(64, 64, 64);
    myGLCD.fillRect(0, 226, 319, 239);
    myGLCD.setColor(255, 255, 255);
    myGLCD.setBackColor(255, 0, 0);
    myGLCD.print("*	Universal Color TFT Display Library *	", CENTER, 1);
    myGLCD.setBackColor(64, 64, 64);
    myGLCD.setColor(255,255,0);
    myGLCD.print("<http://electronics.henningkarlsen.com>", CENTER, 227);
    myGLCD.setColor(0,0,255);
    myGLCD.drawRect(0, 14, 319, 225);

    // Draw crosshairs
    myGLCD.setColor(0, 0, 255);
    myGLCD.setBackColor(0, 0, 0);
    myGLCD.drawLine(159, 15, 159, 224);
    myGLCD.drawLine(1, 119, 318, 119);
    for (int i=9; i<310; i+=10)
    {
        myGLCD.drawLine(i, 117, i, 121);
    }
    for (int i=19; i<220; i+=10)
    {
        myGLCD.drawLine(157, i, 161, i);
    }

    // Draw sin-, cos- and tan-lines
    myGLCD.setColor(0,255,255);
    myGLCD.print("Sin", 5, 15);
    for (int i=1; i<318; i++)
    {
        myGLCD.drawPixel(i,119+(sin(((i*1.13)*3.14)/180)*95));
    }
    myGLCD.setColor(255,0,0);
    myGLCD.print("Cos", 5, 27);
    for (int i=1; i<318; i++)
    {
        myGLCD.drawPixel(i,119+(cos(((i*1.13)*3.14)/180)*95));
    }
    myGLCD.setColor(255,255,0);
    myGLCD.print("Tan", 5, 39);
    for (int i=1; i<318; i++)
    {
        myGLCD.drawPixel(i,119+(tan(((i*1.13)*3.14)/180)));
    }
}
delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);
myGLCD.setColor(0, 0, 255);
myGLCD.setBackColor(0, 0, 0);
myGLCD.drawLine(159, 15, 159, 224);
myGLCD.drawLine(1, 119, 318, 119);

// Draw a moving sinewave
x=1;
for (int i=1; i<(318*20); i++)
{
    x++;
    if (x==319)
        x=1;
    if (i>319)
    {
        if ((x==159)||(buf[x-1]==119))
            myGLCD.setColor(0,0,255);
        else
            myGLCD.setColor(0,0,0);
        myGLCD.drawPixel(x,buf[x-1]);
    }
    myGLCD.setColor(0,255,255);
    y=119+(sin(((i*1.1)*3.14)/180)*(90-(i / 100)));
    myGLCD.drawPixel(x,y);
    buf[x-1]=y;
}
delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled rectangles
for (int i=1; i<6; i++)
{
    switch (i)
    {
        case 1:
            myGLCD.setColor(255,0,255);
            break;
        case 2:
            myGLCD.setColor(255,0,0);
            break;
        case 3:
            myGLCD.setColor(0,255,0);
            break;
        case 4:
            myGLCD.setColor(0,0,255);
            break;
        case 5:
            myGLCD.setColor(255,255,0);
            break;
    }
    myGLCD.fillRect(70+(i*20), 30+(i*20), 130+(i*20), 90+(i*20));
}
delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled, rounded rectangles
for (int i=1; i<6; i++)
{
    switch (i)
    {
        case 1:
            myGLCD.setColor(255,0,255);
            break;
        case 2:
            myGLCD.setColor(255,0,0);
            break;
        case 3:
myGLCD.setColor(0,255,0);
break;
case 4:
myGLCD.setColor(0,0,255);
break;
case 5:
myGLCD.setColor(255,255,0);
break;
}
myGLCD.fillRoundRect(190-(i*20), 30+(i*20), 250-(i*20), 90+(i*20));
}
delay(2000);
myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled circles
for (int i=1; i<6; i++)
{
    switch (i)
    {
    case 1:
        myGLCD.setColor(255,0,255);
        break;
    case 2:
        myGLCD.setColor(255,0,0);
        break;
    case 3:
        myGLCD.setColor(0,255,0);
        break;
    case 4:
        myGLCD.setColor(0,0,255);
        break;
    case 5:
        myGLCD.setColor(255,255,0);
        break;
    }
    myGLCD.fillCircle(100+(i*20),60+(i*20), 30);
}
delay(2000);
myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some lines in a pattern
myGLCD.setColor (255,0,0);
for (int i=15; i<224; i+=5)
{
    myGLCD.drawLine(1, i, (i*1.44)-10, 224);
}
myGLCD.setColor (255,0,0);
for (int i=224; i>15; i-=5)
{
    myGLCD.drawLine(318, i, (i*1.44)-11, 15);
}
myGLCD.setColor (0,255,255);
for (int i=224; i>15; i-=5)
{
    myGLCD.drawLine(1, i, 331-(i*1.44), 15);
}
myGLCD.setColor (0,255,255);
for (int i=15; i<224; i+=5)
{
    myGLCD.drawLine(318, i, 330-(i*1.44), 224);
}
delay(2000);
myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some random circles
for (int i=0; i<100; i++)
{
myGLCD.setColor(random(255), random(255), random(255));
x=32+random(256);
y=45+random(146);
r=random(30);
myGLCD.drawCircle(x, y, r);
}
delay(2000);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRect(1, 15, 318, 224);

// Draw some random rectangles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
y=16+random(207);
x2=2+random(316);
y2=16+random(207);
    myGLCD.drawRect(x, y, x2, y2);
}
delay(2000);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRect(1, 15, 318, 224);

// Draw some random rounded rectangles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
y=16+random(209);
x2=2+random(316);
y2=16+random(209);
    myGLCD.drawRoundRect(x, y, x2, y2);
}
delay(2000);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRect(1, 15, 318, 224);
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
y=16+random(209);
x2=2+random(316);
y2=16+random(209);
    myGLCD.drawLine(x, y, x2, y2);
}
delay(2000);
myGLCD.setColor(0, 0, 0);
myGLCD.fillRect(1, 15, 318, 224);
for (int i=0; i<10000; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    myGLCD.drawPixel(2+random(316), 16+random(209));
}
delay(2000);
myGLCD.fillScr(0, 0, 255);
myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect(80, 70, 239, 169);

myGLCD.setColor(255, 255, 255);
myGLCD.setBackColor(255, 0, 0);
myGLCD.print("That's it!", CENTER, 93);
myGLCD.print("Restarting in a", CENTER, 119);
myGLCD.print("few seconds...", CENTER, 132);
```c
myGLCD.setColor(0, 255, 0);
myGLCD.setBackColor(0, 0, 255);
myGLCD.print("Runtime: (msecs)", CENTER, 210);
myGLCD.printNumI(millis(), CENTER, 225);

delay (10000);
}
```

**Tags:** shields (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q), lcd (Prev Q) (Next Q)

**User:** bay

---

**Answer** by user3473830

I just happened to buy the same LCD Shields a few days ago, looking for a library to use it with a MEGA 2560 board I found [https://github.com/Smoke-And-Wires/TFT-Shield-Example-Code](https://github.com/Smoke-And-Wires/TFT-Shield-Example-Code) which supports both UNO and MEGA boards.

Usage is very, simple if we want to use it for MEGA we should change the header `#include "uno_24_shield.h"` in SWTFT.cpp to `#include "mega_24_shield.h"

**Description (useful for adding support for the shield in other libraries):**

Incompatibility comes from different Port mappings for Arduino pin-out between Mega and UNO.

**in UNO LCD shield will be connected through:**

<table>
<thead>
<tr>
<th>LCD Data Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital pin #</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Uno port/pin</td>
<td>PD7</td>
<td>PD6</td>
<td>PD5</td>
<td>PD4</td>
<td>PD3</td>
<td>PD2</td>
<td>PB1</td>
<td>PB0</td>
</tr>
</tbody>
</table>

**in MEGA it will be connected through:**

<table>
<thead>
<tr>
<th>LCD Data Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital pin #</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>MEGA port/pin</td>
<td>PH4</td>
<td>PH3</td>
<td>PE3</td>
<td>PE5</td>
<td>PE5</td>
<td>PE4</td>
<td>PH6</td>
<td>PH5</td>
</tr>
</tbody>
</table>

---

**Answer** by gwideman

A way to proceed is to create a spreadsheet showing the pin positions used by this board, and the Arduino shield signals they plug into. Next to these, you need columns showing the actual signals on the ATMega2560 (for Mega2560) and ATMega328 (for Uno) that these shield pins attach to. You can get this info from the Uno and Mega2560 schematic drawings.

In a quick look, it seems that the Arduino shield pin names for Uno and Mega are the same: for example, shield pin ‘0’ (digital zero) is in the same location on both boards, and likewise for other pins.

However, on the Uno digital-0 attaches to ATMega328 Port D bit 0, while on the Mega2560, it attaches to ATMega2560 Port E bit 0. And things get more obtuse with
digital 2..7.

Now, when twiddling bits individually using `digitalWrite(pin, value)`, the Arduino library no doubt takes care of translating to the appropriate port/bits that need to be set for the ATMega chip and Arduino board that’s in use. However, libraries that use lower-level functions (especially if they need to write entire bytes to ports, as a fast LCD library might) will need to take their own steps to make this translation.

So… first step is to determine whether there is a separate LCD driver library for Mega2560.

Next, investigate whether the library you have has initialization code that is supposed to determine what board it’s running on (and is your board included?), or requires you to set some flag to tell it what board is in use.

Failing that, you could create a mess of jumpers or some other wiring scheme to jumper the Mega’s ATMega2560’s signals so that it’s wired up like a Uno would be. It’s not clear that this is possible, since some of ATMega2560’s Port D is not even wired to a header.

Or you could look at the source code for the library and see what it’s actually doing, and what it would need to do different to operate the ATMega 2560 pins that the shield does connect to.

__Answer___

Have you checked out the Library homepage? [Henning Karlsen’s library page](http://www.hackster.io/hk1280/arduino-gprs-shield)

He has made a user manual for the library. There is also a reference to what pin goes where in the requirements document.

---

**Tags:** shields (Prev Q) (Next Q), arduino-mega (Prev Q) (Next Q), lcd (Prev Q) (Next Q)

---

**Q: Which shield to use for GPRS?**

**Tags:** shields (Prev Q) (Next Q), gsm (Next Q), tcpip (Next Q)

I'm planning to build a device that would read some sensor data and send it via GPRS, eg. once per day. (Not that original, yeah.) But my problem is choosing a GSM/GPRS shield.

The official shield has a [nice interface for doing an HTTP POST/GET](http://www.hackster.io/hk1280/arduino-gprs-shield). OTOH, the shield seems to be sold out (and would be quite expensive anyway).

There seem to be other shields available, but their code examples tend to look quite hacky: the device waits for an arbitrary moment and then hopes that the server is done. That's IMO both inefficient or unreliable.

My question: which GSM/GPRS shield would you recommend, with these features:

- a sane library w/ examples, such as the official one
- an external antenna
- bonus: a competitive price
- another bonus: hopefully a “real” shield, so no soldering required and a few pins still
easily usable for the sensors.

Tags: shields (Prev Q) (Next Q), gsm (Next Q), tcpip (Next Q)

User: tuomassalo

Answer by sachleen

That library should work with pretty much anything that has the M10 module on it. I only have experience with the SIM900 modules. Found the cheapest one on EBay. While interfacing with these things can be a challenge at first, you really just need to read the manual for all the AT commands and execute them. I’ve written a couple of functions that may help:

Note: you may safely replace all instances of DEBUG_PRINT and DEBUG_PRINTLN with Serial.print and Serial.println.

```cpp
SoftwareSerial SIM900(7, 8);

/*
 * Sends AT commands to SIM900 module.

 Parameter      Description
  command       String containing the AT command to send to the module
  timeout       A timeout, in milliseconds, to wait for the response

 Returns a string containing the response. Returns NULL on timeout.
 */
String SIMCommunication::sendCommand(String command, int timeout) {
  SIM900.listen();    // Clear read buffer before sending new command
  while(SIM900.available()) { SIM900.read(); }

  SIM900.println(command);
  if (responseTimedOut(timeout)) {
    DEBUG_PRINT(F("sendCommand Timed Out: "));DEBUG_PRINTLN(command);
    return NULL;
  }

  String response = "";
  while(SIM900.available()) {
    response.concat((char)SIM900.read());
    delayMicroseconds(500);
  }
  return response;
}

/*
 * Waits for a response from SIM900 for <ms> milliseconds

 Returns true if timed out without response. False otherwise.
 */
bool SIMCommunication::responseTimedOut(int ms) {
  SIM900.listen();

  int counter = 0;
  while(!SIM900.available() && counter < ms) {
    counter++;
    delay(1);
  }

  // Timed out, return null
```
if (counter >= ms) {
    return true;
} else {
    counter = 0;
    return false;
}
**Q: How to use components that don’t have a shield?**

**Tags:** shields (Prev Q) (Next Q)

I recently started messing around with an Arduino and I have made a few “gadgets” using shields. I then wondered how I would go about using things I can’t find shields for.

One specific thing I want to do is to be able to use the larger e-paper displays from Pervasive Displays (7” and 10”) but a brief consultation with Google found no Arduino kits or shields. I then looked through the documentation of the timing controller for the 7” display and found the information required to be able to use it (initiation, sending data to be drawn, opcodes, etc.). I then decided to check the Github repo for the examples for the 2” displays and they seem to just do what the documentation says to do.

Is my observation naive here? Is it really as straightforward as following the documentation? No magic sauce?

Note: I don’t have a clue how complex using the 7” display actually is, I just used it as an example because I had just looked it up…

**Tags:** shields (Prev Q) (Next Q)

**User:** user2272296

**Answer** by sachleen

Your observation is correct. Everything breaks down to some number of physical connections and some protocol for communicating with the device. Shields and libraries associated with them do most of the work for you, but not everything comes in a shield.

There are some common protocols for interfacing with devices, including SPI and I2C. From a quick peek at the driver interface sheet, it looks like they’re using SPI. Well, your Arduino already supports that so you can hit the ground running with this display. You just need to know what commands to send to it, which you can find from the display’s datasheet.

**Note:** It looks like the 2” screen requires 3.3V while the 7” one supports up to 7V.

**Tags:** shields (Prev Q) (Next Q)

**Q: If I put a shield on an Arduino, can I use the Arduino for anything else?**

**Tags:** shields (Prev Q) (Next Q)

I did some basic Arduino examples in the recent past, but I have never actually seen a shield for Arduino firsthand.

So I wonder: If I put a shield on an Arduino (for example, the Adafruit Motor Shield), does it “block” the entire Arduino, including all input and output pins?
Is it possible to use a shield, and still connect additional components that would usually connect directly to the Arduino (say, a potentiometer, or anything similar)? Does this depend on the type of Arduino, or on the type of the shield, or is it not possible at all?

Tags: shields (Prev Q) (Next Q)

User: majiy

---

**Answer** by chris-stratton

There are several distinct concerns to consider:

- **Physical interference**: are headers provided to make it possible to stack the two shields on top of one another? Do any protruding components block stacking? Do any components have metal shield cans (for example the USB connector on the Uno itself) which can short out an adjacent PCB?

- **Pin assignments**: generally you cannot use the same pin for two different purposes. Many shield provide a way to alter the pins used in order to avoid those used by another shield. Sometimes you can share a pin, for example two SPI devices can usually share their clock and data lines provided they have separate selects, and you may be able to even use those signals for other purposes while the SPI select is de-asserted. Sometimes you can move a part from an unavailable hardware peripheral to a software emulated one, for example using software serial, bitbang SPI, or interrupt based PWM.

- **MCU resources**: Not specifically the shields themselves, but driver libraries may end up consuming large amounts of program memory or RAM, using up timer channels, needing interrupts frequently services with low latency, etc, so combinations may not be possible or may require alterations to library code.

---

**Answer** by ignacio-vazquez-abrams

As long as the shield uses stackable headers (or provides alternate headers) it doesn’t block any (broken out) pins, either input or output. It is possible to add more shields or components, but the circuit must still be “valid” (all inputs must be either be connected to an output or have the pullup enabled, no more than one push-pull output connected together, etc.), otherwise components may be damaged.

---

**Q: What is the advantage of using a motor shield if I want to use a stepper motor?**

Tags: shields (Prev Q) (Next Q), motor (Prev Q) (Next Q)

I know that it is possible to connect a stepper motor directly to an arduino (as displayed here). I know that another option is to use a motor shield (for example the Arduino Motor Shield or the Adafruit Motor Shield).
What I would like to know: What are the actual advantages of using a motor shield?

Is it just a question of convenience? Or does a motor shield do something that could not easily performed without a shield? Does it allow me to connect more motors than I could connect directly without a shield (power supply comes to mind)?

Tags: shields (Prev Q) (Next Q), motor (Prev Q) (Next Q)

User: majiy

---

**Answer** by peter-r.-bloomfield

The example you linked to isn’t actually connecting the Arduino directly to a stepper motor. It’s going via a ULN2003A driver. That’s a very common Darlington transistor array, which basically just lets you use a small current to switch a larger load.

That’s necessary because the Arduino pins can’t safely source enough current to drive the stepper motor directly. It might work for a while, but it would damage your Arduino sooner or later.

Motor shields usually have a similar driver in them, but can also provide additional functionality which you might find useful. For example, the Adafruit shield you linked to is controlled by the I2C bus, and has its own on-board PWM. That means you can control several motors without needing any extra output pins on your Arduino.

In short, motor shields aren’t necessary. They can be helpful though, especially if you aren’t familiar with the underlying electronics, or you don’t want to build it yourself.

---

**Answer** by jippie

Mechanical stability of the whole thing is probably the main advantage. I’d personally opt for a generic module or loose components. The main disadvantage of shields is pin conflicts when using more than one shield.

---

**Answer** by darellon

I’m using a adafruit motorshield v2 for my solar tracker project with two stepper motors atm. i’m quite a new to arduino and coding and i think the shield made it a lot easier for me to get a hang of it - easy to code and nicely stack able on the arduino (little soldering to assemble but not too difficult).

---

**Q: Are all Arduino shields compatible with all Arduino boards?**

Tags: shields (Prev Q) (Next Q), motor (Prev Q) (Next Q)

Are all Arduino shields compatible with all Arduino boards? Like is the Wi-Fi one compatible with the Arduino Nano & Micro as well as the Uno, etc., or do I need to be careful when buying shields that they’ll be compatible with my board? Also, the shields just plug into the boards, don’t they?
When you want to choose a shield for your board you should check:

1- **Pin-out**, this one is the least concerning one, Most arduino boards are Shield pin-out compatible and you can verify it visually, for others, normally there are some converter boards (e.g for NANO), and in the worst case, building a converter board or wiring the shield to board is trivial in most situations.

2- **Shield operating voltage should match your board’s**, most shields operate at 5V, 3.3V or both. Normally AVR-based boards (e.g. UNO, Mega, …) operate at 5V and ARM based ones (e.g. Due, …) operate at 3.3V, So we should choose the shield accordingly (*just keep in mind that most shield are designed with Arduino UNO in mind so it’s more likely that your shields operate at 5V*).

3- **Library**, Lots of Libraries for Arduino shields bypass Arduino library abstraction (mostly for performance reasons) and use hardware specific features (e.g directly write on specific ports), here is source of some incompatibilities between your board and the shield, for example there are handful of TFT LCD shields in the market which their library only supports UNO boards, in these cases you should either port it to your board’s (based on micro-controller model and it’s architecture, sometimes its as easy as changing some pin/port definitions and sometimes as hard as writing the library from scratch).

---

**Q: H-Bridge vs Motor shield for the arduino**

Assuming we wish to control a small toy car with an arduino - meaning spinning the right and left wheels forward and backward and being able to break and control the speed, where each wheel has it’s own engine (but of course two engines on the same side will run on the same speed and direction).

We saw we have two main options - Buying and wiring up a H-Bridge or buying a motor shield (which is based on one).

Are there any advantages for using a motor shield?

Thanks in advance.

**User:** avenger

**Answer** by ladislas

If I were a beginner, I’d go with the Motor shield.
I’m even personally using the one from DFRobot, which as @Gerben said, is just plug and play.

It also allows you to select the input power for the motors, which can be either from the 5V Arduino pin or from an external battery. This is pretty convenient if you need more than 5V to power up your motors. It can also support 2A of current draw, which is more than enough for small Arduino toy cards.


Tags: shields (Prev Q), motor (Prev Q) (Next Q)
**Arduino Mega**

**Questions**

**Q: What is the cheapest way to add wifi to Arduino?**

Tags: arduino-mega (Prev Q) (Next Q)

I am trying to find some sort of wifi adapter for Arduino, it needs to be as small as possible (not a shield) and as cheap as possible.

So far I can only seem to find shields that cost about £10+ but I figured that if you can get a replacement wifi adapter for a laptop for next to nothing that I should be able to get a wifi adapter for Arduino much cheaper than this.

It’s possible I just don’t know what to search for.

Tags: arduino-mega (Prev Q) (Next Q)

User: connersz

---

**Answer** by yeti

Some months have passed since my previous answer and now the ESP8266 based WiFi modules are available.

They get connected to your project serially (3.3V) and are controlled in a modem like way using AT commands.

With a pricing of round about $5 they fit the question better than my previous OpenWrt based answer.

For more information about ESP8266 ask your favorite finding machine or list news with tag ESP8266 at Hackaday.

*(Old answer from the pre-ESP8266-days:)*

I vote for a small router reflashed with OpenWrt like “TP-Link TL-WR703N”.

It is like building an own Yûn with an Arduino of your choice…

The WR703N:

It has Ethernet, 150MBit WiFi, USB and if you want to solder, a serial console and 2 GPIO-Pins which can be turned into I2C.

This sure is not a chip- or stamp-sized solution but often cheaper than the alternatives, more powerfull and extendable.

Search the net with your favourite find’o’bot for “arduino wr703n” and you’ll find lots of examples…
Q: How do I specify a working directory when saving to SD card?

Tags: arduino-mega (Prev Q) (Next Q)

I’m trying to datalog to an EyeFi SD card by writing to a file with a jpg extension. Here’s that portion of my code so far:

```c
Skip code block
// see if the directory exists, create it if not.
if(!SD.exists("/DCIM/100NIKON") )
{
  if( SD.mkdir("/DCIM/100NIKON") )
  {
    Serial.print("File directory created: ");
  }
  else
  {
    error("File directory not created");
  }
}
else
{
  Serial.print("File directory exists: ");
}

// Test file
char filename[] = "DSCN0000.JPG";
if (!SD.exists(filename))
{
  logfile = SD.open(filename, FILE_WRITE); // only open a new file if it doesn't exist
}
if (!logfile)
{
  error("Couldnt create file");
}
```

The Arduino creates the directory, but the file is still saved to the root. I would love some help or tips on this.

Tags: arduino-mega (Prev Q) (Next Q)

User: user2218339

Answer by connor-wolf

You’re creating a directory, and then not writing the file into it.

You need to change the filename to `/DCIM/100NIKON/DSCN0000.JPG`;

Note that `SD.open()` takes the **full filepath to the file**, not just the file*name*.

From the Arduino Docs:

The file names passed to the SD library functions can include paths separated by forward-slashes, /, e.g. “directory/filename.txt”. Because the **working directory is always the root of the SD card**, a name refers to the same file whether or not it includes a leading slash (e.g. “/file.txt” is equivalent to “file.txt”). As of version 1.0, the library supports opening multiple files.
Q: How can I indicate that I am using ATmega2560 so that Arduino libraries understand it?

Tags: arduino-mega (Prev Q)

I am developing AVR microcontrollers in Atmel Studio and sometimes I am using Arduino libraries(such as LiquidCrystal).

When I initialize 

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

I don’t think Arduino Libraries use registers and ports of ATmega2560, although I am building my code with the device selection ATmega2560 option of Atmel Studio. Atmel Studio works in order to program 2560 but how can I tell Arduino libraries that I am using 2560?

Tags: arduino-mega (Prev Q)

User: zgrknl

---

Answer by majenko

There are two things that have to be considered here.

One is telling the code what board you are using, so it can compile in the right optional parts, and the other is getting the right pin mappings.

The first can be covered simply with a compilation flag:

```
-DARDUINO_MEGA2560
```

That defines the macro ARDUINO_MEGA2560 which is used in some places to know which board you are using.

Second is the pin mappings. These are stored in the file pins_arduino.h and each board variant has its own copy. You need to ensure that this file (and be sure to get the right one) is included in your code in the right places to ensure that the pins can be mapped right. If you are including the whole Arduino core (which it sounds like you must be in order for things to even have a hope of working) then you just need to ensure that you have the compiler set up to include the right copy of the pins_arduino.h file.

That means making sure you have an include path in your compile command line which points to where the pins_arduino.h file is before anything else Arduino core related.

```
-I'C:\Path\to\arduino\mega2560\variant
```

or

```
-I/path/to/arduino/mega2560/variant
```

Tags: arduino-mega (Prev Q)
**Arduino Ide**

**Questions**

**Q: Why does opening the Tools menu take so long?**

Tags: arduino-ide (Prev Q) (Next Q)

Every time I open the Tools menu, the IDE freezes for a few seconds. Why is that? It happens on both of my computers, and nothing I do makes a difference in how long it takes. I’ve tried restarting the IDE, unplugging my Uno, and many other things, but nothing helps.

I’m on windows, running IDE version 1.0.2.

Tags: arduino-ide (Prev Q) (Next Q)

User: the-guy-with-the-hat

---

**Answer** by federico-fissore

It’s an RXTX issue. RXTX is the native serial library used to make the IDE talk to the boards.

On some Windows installation, with some Bluetooth things (both dongles or integrated), when enumerating serial devices, these dongles scan the air for remote serial devices. Hence, the IDE stays stuck, waiting for RXTX serial device enumeration to end. This issue has been acknowledged a lot of times, see for example #1375, #532, #200 and #143

As far as I understand, this issue does not affect linux or mac because of the different way the scan occurs. On linux it’s a plain filtered file listing of /dev. Similarly it’s on mac. On windows, it uses native API calls.

Some fixes are available in the 1.5 series but I suggest switching to the latest 1.5.6 release as RXTX was replaced by JSSC, a more modern and simpler library.

---

**Answer** by philip-allgaier

Frederico Fissore already pointed out that this delay is due to the serial libraries. From a functional perspective: The Arduino IDE tries to enumerate all available COM ports which simply takes some time.

Quote from the official troubleshooting guide:

**Why does the Arduino software and the Tools menu take a long time to open (on Windows)?**

If the Arduino software takes a long time to start up and appears to freeze when you
try to open the Tools menu, there by a conflict with another device on your system. The Arduino software, on startup and when you open the Tools menu, tries to get a list of all the COM ports on your computer. It’s possible that a COM port created by one of the devices on your computer slows down this process. Take a look in the Device Manager. Try disabling the devices that provide COM ports (e.g. Bluetooth devices).

---

**Answer** by johnny-norre

I was having the same problem under Windows 7 x64 and had already removed all com ports and unknown devices as suggested above. For me, the solution was to rename “ListComPorts.exe” in the `hardware\tools` subfolder to “ListComPorts.exe.xxx” - this prevents the Arduino IDE from using this file and apparently it reverts to using the internal COM port detection mechanism (using DEVICE_ARRIVAL/DEVICE_REMOVAL messages??) . The tool menu now opens immediately every time.

---

**Tags:** arduino-ide (Prev Q) (Next Q)

---

**Q:** Can I change font Color and/or Type for IDE 1.5.6r2?

**Tags:** arduino-ide (Prev Q)

The program becomes harder to read (at least on my particular monitor) after upgrading. Any way to change font **Color to darker** color and/or **Type to thicker stroke** (like B in this question editor) for IDE 1.5.6r2?

**Tags:** arduino-ide (Prev Q)

**User:** john-williams

---

**Answer** by mpflaga

Yes.

Under the file menu select the preferences. From this you can find and than view/edit the preferences.txt file, which has the color settings.

Example [preferences.txt](#)

---

**Tags:** arduino-ide (Prev Q)
Interrupt

Skip to questions.

Wiki by user peter-r.-bloomfield.

For more information see the isr tag.
Questions

**Q: How precise can I get Arduino using rotary encoders?**

Tags: interrupt (Prev Q) (Next Q)

Stepper motors are often pricey for a large motor. However, with a powerful, standard DC motor and a rotary encoder, you can “simulate” a stepper motor. **How accurate are rotary encoders with Arduino just with a basic loop and not much other code?** Is it going to “miss a pulse” and not record enough rotation often? How can I handle this with much more code? Is there a better way to do this? Would interrupts help? Is there some sort of circuit that would help with this problem?

Note: It will probably not exceed 4 rotations a second… *(That may be a lot but if you’re doing a lead screw type mechanism that would only be 24 inches a minute if there is 10 rotations per inch of linear motion.)*

Tags: interrupt (Prev Q) (Next Q)

User: anonymous-penguin

---

**Answer** by matt-clark

Interrupts are what you would use in this situation.

The rates that you have mentioned are slow enough where you *probably* would be able to count it inside of a loop, however this is not recommended, as a typical loop will take many clock cycles to complete, and depending on the number of instrucitons per loop, you may miss some counts.

Interrupts are made for this purpose, as they can be triggered at any time, regardless of the state of the loop.

There is a function named `attachInterrupt()` which will take three arguments.

1. Interrupt Pin
2. Interrupt Service Routine
3. Mode

Below is an example for counting an interrupt

```c
volatile int myCounter = 0;

void setup(){
    attachInterrupt(0, count, RISING);
}

void count(){
    myCounter++;
}
```

The Modes are as follows:

    LOW, CHANGE, RISING, FALLING
More information about using interrupts can be found here.

Depending on the encoder you use, you will need to tailor the code to your needs, and do some more calculations to determine your position, but this should be a good starting point.

**Edit** [Here](#) is some example code from Arduino Sandbox for using a rotary encoder.

---

**Answer** [Here](#) by gwideman

Adding some references to already-written libraries and examples, to enable comparison between different approaches, and experiences with speed versus susceptibility to missing steps.

Reading rotary encoders: [http://playground.arduino.cc/Main/RotaryEncoders](http://playground.arduino.cc/Main/RotaryEncoders)


Teensy Encoder library: [https://www.pjrc.com/teensy/td_libs_Encoder.html](https://www.pjrc.com/teensy/td_libs_Encoder.html)


---

**Tags:** interrupt (Prev Q) (Next Q)

---

**Q: Multiple independent LED patterns**

**Tags:** interrupt (Prev Q), led (Prev Q) (Next Q)

I have a problem, which at first thoughts (and being new to Arduino) I thought was a perfect application for an Arduino. However, after trying and failing to implement it I am doubting myself!

Simply - I need to control many LEDs independently, many of which will have their own individual patterns - for example “5 seconds on - 5 seconds off”, “continuous flashes” - or sequences such as “2 flashes, pause, 1 flash”. Obviously without the luxury of threads I am becoming a little unstuck. Be great to hear if a) Arduino is the best choice and b) if it is - how can I go about this!

Thanks in advance :)

**Tags:** interrupt (Prev Q), led (Prev Q) (Next Q)

**User:** nickos

---

**Answer** [Here](#) by peter-r.-bloomfield

Handling multiple patterns at the same time is certainly possible with a platform like Arduino, and there are a number of ways you could go about it.

One method I would consider is writing functions which effectively represent each pattern mathematically. You’d just pass it the total time that’s elapsed in your program so far, and
it will do the appropriate action for that specific moment in time. It will return immediately afterwards (no delays or anything).

To do that, you’ll first need to know how long a single cycle of the pattern lasts. You can then use the modulo operator to figure out how far through the current cycle you are. From there, all you need to do is have some if conditions to determine what to do at any given time.

Here’s what it might look like for your “5 seconds on, 5 seconds off” pattern:

```cpp
function pattern5on5off(unsigned long totalTime)
{
    // Calculate how far through the current cycle we are
    const unsigned long cycleTime = totalTime % 10000;

    // If we're in the first 5 seconds of the cycle then turn the light on.
    // Otherwise, turn it off.
    if (cycleTime < 5000)
        digitalWrite(3, HIGH);
    else
        digitalWrite(3, LOW);
}
```

Admittedly, constantly calling `digitalWrite()` when you don’t technically need to isn’t very efficient. It shouldn’t cause any harm though, and is fairly easy to optimise if necessary.

To use the above example in a sketch, you’d just need to call it in `loop()`, and pass the number you get from `millis();` e.g.:

```cpp
void loop()
{
    const unsigned long totalTime = millis();
    pattern5on5off(totalTime);
    // call other patterns here...
}
```

Other patterns will be more complex, but follow the same principle. You’d just to use appropriate if statements to express your logic.

The vital thing to remember is that the function represents a specific moment in time. It should never pause or delay the program, otherwise it will prevent the other patterns from running.

**Edit: Timing on the first cycle**

As jfpoilpret noted in the comments, the very first cycle will start at a random point. This is because the first time you call `millis()` in `loop()`, it won’t start at 0 (the device will already have been running for a short time before `loop()` gets called). It’s easy to resolve though, if necessary.

You would do it by offsetting the `totalTime` value by the whatever value you got on the very first time around `loop()`. For example:

```cpp
unsigned long g_startTime = 0;
void loop()
{
    unsigned long totalTime = 0;
```
if (g_startTime == 0) {
    // This is the first cycle.
    // Store the start time so we can compensate later.
    g_startTime = millis();
} else {
    // This is not the first cycle.
    // Compensate for the start time.
    totalTime = millis() - g_startTime;
}

pattern5on5off(totalTime);
// etc..

---

**Answer** by user2973

Arduino is a fine choice for the task - easy to get started with. The key is to write non-blocking code. You could take a look at the BlinkWithoutDelay example.

I made a suggestion for your task:

**Skip code block**

```cpp
// Timing sequences for the LED's in milliseconds
// First value is on time, second value is off time,
// third value on time and so on (up to 10 values)
// One row for each LED
unsigned int led_timing[][10] = {
    {5000, 5000},
    {100, 1000},
    {100, 100, 100, 1500, 100, 1500}
};

// The pins the LED's are connected to
byte led_pins[] = {11, 12, 13};

// Keep track of timing sequence
// Array size taken from led_pins
unsigned long last_change[sizeof(led_pins)/sizeof(led_pins[0])];
byte timing_i[sizeof(led_pins)/sizeof(led_pins[0])];

void setup()
{
    // Initialize LED's as output
    for (byte i = 0; i < sizeof(led_pins)/sizeof(led_pins[0]); i++)
    {
        pinMode(led_pins[i], OUTPUT);
        digitalWrite(led_pins[i], HIGH);
    }
}

void loop()
{
    // Current timestamp
    unsigned long now = millis();

    // Keep track of sequence for each LED
    for (byte i = 0; i < sizeof(led_pins)/sizeof(led_pins[0]); i++)
    {
        if (now - last_change[i] >= led_timing[i][timing_i[i]])
        {
            digitalWrite(led_pins[i], !digitalRead(led_pins[i]));
            timing_i[i]++;
            // Start over at the end of timing sequence
            timing_i[i] %= sizeof(led_timing[i])/sizeof(led_timing[i][0]);
        }
        last_change[i] = now;
    }
}
```
An LED (Light Emitting Diode) is a semiconductor component which emits light (via electro-luminescence) when its forward voltage is sufficiently high.

In Arduino projects, small LEDs are commonly used as indicators. For example, it could be illuminated to show that a system is active/powered-on. LEDs typically operate at around 2-3 volts, requiring a current of 10-20 mA, meaning they can by powered directly from the digital output pins.

For safety, an LED should normally be used with a current-limiting resistor in series. LEDs can also be used for other purposes, including general illumination, and in opto-isolators.
Questions

Q: Can two LEDs attached to different pins share their resistor?

Tags: led (Prev Q) (Next Q)

I’m designing an Arduino style board with built-in LEDs for several of the pins. To simplify assembly of the board and save on components, I was wondering whether it would be safe for those LEDs to share a single resistor, as in this diagram:

![Diagram of LEDs sharing a resistor](image_url)

Obviously, the power dissipated through the resistor (3mA per pin @ 3V) is not a real problem. The reason I was hesitating is that LEDs in a parallel configuration should not share resistors, but is it safe to think that separate microcontroller pins don’t count as a common anode, even if both pins are high?
Can two LEDs attached to different pins share their resistor?

- **can**: yes
- **should**: probably not

Your resistor act here as a current limiter, say to 3mA

If you light one LED at a time, you give 3mA to *that* LED If you light 2 LEDs, you give 3mA for both of them (more of less 1.5mA each) If you light 3 LEDs, you give 3mA for all of them (*on average* 1mA each)

Things are even a little bit more subtle, as different LED might have different forward voltage, so one LED might very well drain 10% more current than an other.

As “Amps made LED bright”, less Amp is less light. On LED light with that setup will give “full light on”, and more LED you add, less would bright each one.

As an edit to that answer, there is one case I might consider that arrangement for “multi-LED lighting”. This is if you are [multiplexing your LEDs](https://microtherion.github.io/)

That is lighting only one at a time, but cycling fast enough from one LED to the next so that the human eye would see them all lit at once.

Your circuit is safe from the electronic viewpoint, so long as the LEDs used can withstand a back-emf of 4v or so.

If only one LED is on at a time, all will be sweet. But if two or more are on, the brightness will drop and the brightness between LEDs will probably be inconsistent. The forward voltage drop of individual LEDs tend to vary a little, so one is likely to hog most of the available current.

A separate resistor per LED is probably best.

**Q: Why are LED 0 and 1 on as well?**

**Tags**: led (Prev Q) (Next Q)
int pinCount = sizeof(ledPins);

void setup(){
    for (int i = 0; i < pinCount; i++){
        pinMode(ledPins[i], OUTPUT);
    }
}

void loop(){
    for (int i = 0; i < pinCount; i++){
        digitalWrite(ledPins[i], 1);
    }
}

Tags: led (Prev Q) (Next Q)

User: kees-bakker

Answer by peter-r.-bloomfield

The behaviour of your program is undefined because you’re reading past the end of your array.

In C/C++, the sizeof operator always gives you the size of something in bytes. It doesn’t directly tell you the number of items in an array. Since int is usually 2 bytes for Arduino programming, the pinCount variable is actually being initialised to 10 (5 items x 2 bytes each).

The result is that each for loop is actually doing 10 iterations. The first 5 iterations are OK, reading correctly from the ledPins array. However, the rest of the iterations are reading whatever data happens to be in memory after that. This means it could be trying to initialise and switch on any pin, potentially including 0 and 1.

I can’t guarantee that this is the source of the problem, but it’s certainly very important to fix.

The solution is quite easy though. Divide the total size of the array by the size of a single element:

```
int pinCount = sizeof(ledPins) / sizeof(int);
```

Answer by mark-williams

You haven’t initialised the other pins at all; or included your circuit, but clearly you have LEDs on more pins than those defined (5 defined, 7 on).

The other LEDs are, therefore, in an undefined state - and may well be attached to pins set ‘on’ by default in your circuit. You probably need to either move 0/1 off the serial interface or explicitly set them ‘off’.

Tags: led (Prev Q) (Next Q)

Q: Why does an RGB LED between VCC and PWM work?

Tags: led (Prev Q), pwm (Prev Q) (Next Q)
I’m an Arduino beginner and I recently bought a cheap starter kit on eBay. One of the lessons in the starter kit is an RGB LED controlled by 3 PWM output pins. However, the arrangement of the connections seems a bit funny to me:

To my surprise, it worked. I could see the three LEDs inside the plastic change colors. Why does this work? Do the PWM pins accept current whilst low? Is this a common arrangement? See the above link for the code provided in the lesson.

Tags: led (Prev Q), pwm (Prev Q) (Next Q)
User: nobruked
The RGB LED is common anode. This is why it is connected to VCC and the PWM pins pulls down to light resp LED.

A common cathode RGB LED would be connect to GND and the PWM pins would pull up to light

Cheers!

Tags:  led (Prev Q),  pwm (Prev Q) (Next Q)
PWM

A Pulse Width Modulation (PWM) signal is a series of digital pulses. Each pulse makes the signal go high (on) for a short period, and it then drops back to low (off) for a short period before the next pulse. The result can be referred to as a square or rectangular wave, because of its appearance if it is graphed as voltage over time.

There are two important attributes which describe a PWM signal. Firstly, the frequency determines how often the pulses happen. Frequency is expressed in Hertz (Hz), which is the number of pulses per second. PWM signals generated by the Arduino usually have a frequency of 490 Hz (or 980 Hz in some cases).

The second attribute of a PWM signal is the duty cycle. For human-readable purposes, it is often expressed as a percentage. This determines how long a pulse lasts compared to the space between pulses. For example, a pulse could last 5 milliseconds, and be followed by a 15 millisecond gap before the next pulse. This is a 25% duty cycle, because the pulse is high for a quarter of the total time. A 10ms pulse followed by a 10ms gap would be a 50% duty cycle, and so on.

PWM is very useful for controlling the speed of a DC motor. Raising or lowering the duty cycle will cause the motor to speed up or slow down. A conventional analog approach to motor control would raise or lower the voltage instead, which can result in poorer performance.

Similarly, a PWM signal can effectively control the brightness of an LED. Raising and lowering the duty cycle will increase and decrease the apparent brightness of the light.

The most common way to produce a PWM signal on the Arduino is using the `analogWrite()` function. The frequency is fixed, but it allows specification of the duty cycle in the range 0 to 255 (where 255 represents 100%).

The `tone()` function also allows a PWM output to be generated. It has a fixed 50% duty cycle, but allows specification of the frequency in Hz. It is primarily intended for generating simple sounds on a speaker, but could be used for other purposes as well.
Questions

**Q: How precise is the timing of pulseIn()?**

**Tags:** pwm (Prev Q) (Next Q)

I’ve been using the `pulseIn()` function for processing PWM-based binary data encoding. It works well for distinguishing pulses which are significantly different lengths, e.g. 500us vs. 1500us. That makes it more than sufficient for handling typical IR remotes.

However, I want to make my own IR system which can use more than 2 pulse lengths, so that data transfer can occur faster. Ideally, I’d like to use 8 different pulse lengths for octal encoding (e.g. 200us, 400us, 600us, etc.).

I’ve noticed quite significant variations in the values returned by `pulseIn()` though (+/- 10%). I expect at least some of it is introduced by the IR transmitter and receiver modules, but I don’t have good enough equipment to verify that.

Assuming I can mitigate that external error, is `pulseIn()` likely to be precise enough to distinguish such similar pulses?

**Tags:** pwm (Prev Q) (Next Q)

**User:** peter-r.-bloomfield

---

**Answer** by mpflaga

The `pulseIn()` function is very lossy, in that it is a hard loop and returns a number * the assumed clock cycles it takes for per loop

```
// wait for the pulse to stop
while (((*portInputRegister(port) & bit) == stateMask) {
  if (numloops++ == maxloops)  
    return 0;
  width++;
}

// convert the reading to microseconds. The loop has been determined
// to be 20 clock cycles long and have about 16 clocks between the edge
// and the start of the loop. There will be some error introduced by
// the interrupt handlers.
return clockCyclesToMicroseconds(width * 21 + 16);
```

The most accurate method for capturing the timing of a PIN is use the INPUT CAPTURE FEATURE. Look at the [this example](https://example.com). It enables the input capture at 1x of CPU for max resolution and each edge of the input pin captures the timer’s clock value for reading from the generated Interrupt service. It also enables the timer overflow interrupt to as to maintain large absolute time, to be capture. Since the 1x will roll rather quickly. The captures store the time into an array for reading by the main loop.

Where for signals over IR the typical library to use is [shirriff/Arduino-IRremote](https://github.com/shirriff/Arduino-IRremote) library. Where it has several demo’s that will read and send the IR from a demodulated signal. To allow one to build a sketch of their own design. This code originally creates a timer
interrupt that polls the input pin at a rate determine by

```c
#define USECPERTICK 50 // microseconds per clock interrupt tick
```

in the IRremote.h file. For my purposes I have change it to 25 us. Where I find this still can be intermittently missing pulse streams.

Note the demodulation is best accomplished within the IR receiver, which in turn outputs this signal of interest. Where to add some background. Using the typical 38KHz modulation, equates to a minimum resolution of 26.3uS per pulse’s cycle. sherriff’s library shows typically most bauds or bits are in the order of 10+ pulses. Which appear to meet your desired timings.

`microtherion/Arduino-IRremote` fork of shirriff’s work improves the reception by replacing the timer interrupt polling of the pin with the use of PinChangeInterrupts. Which I have merged into my own `mpflaga/Arduino-IRremote` fork, that adds several other features.

So you could use any of the above libraries. Or create your own app that uses either of the below to capture edges.

1. polls on a Timer event (e.g. 50uS)
2. captures the micros() on a PinChangeInterrupt
3. uses Input Capture Interrupts to grab the exact time

---

**Answer** by `thedoctor`

Here is some test data of a `pulseIn` test. One Arduino sent what were supposed to be 14us pulses, and the other spat out this data:

```
18,18,18,12,18,18,18,18,18,18,18,18,18,18,18,18,18,24,19,18,18,18,18,18,24,18,18,18,19,18,18,12,18,18,19,18,18,18,18,18,18 ...
```

As you can see, the pulses are by no means accurate. The time would be more accurate if the sending and receiving ends were written in assembly, or even offloaded to their own processors.

---

**Tags:** [pwm](#) ([Prev Q](#)) ([Next Q](#))

---

**Q:** What is the frequency of PWM output on Arduino

**Tags:** [pwm](#) ([Prev Q](#))

What frequency do Arduinos use for normal PWM when you use `analogWrite()`?

And is it different for different Arduino model? I’m interested specifically in the Mega 2560, but also want to know if it’s consistent between models.

I’ve seen a passing references to the Arduino using 500 Hz, which seems really slow.

**Tags:** [pwm](#) ([Prev Q](#))
The PWM signal is generated by timers on the AVR chips. Each timer can generate a PWM signal on two or three different pins. Each pin can have its own duty cycle, but they share the PWM frequency. You can change the frequency of the PWM by changing the clock source for the timers. By default they use the CPU clock divided by 64, i.e., they have their prescaler set to 64 by Arduino initialization code. To confuse things more there are two different PWM modes: fast PWM and phase correct PWM. In fast PWM the timer counts to 255, then overflows and starts over from 0 (256 different states). In phase correct PWM the timer counts to 255, then changes direction and counts downwards to zero, changes direction and so on (510 different states).

The Arduino Mega has 5 timers, timer0 - timer4. Because timer0 is also used for the millis and micros functions it uses fast PWM, whereas the other timers are configured for phase correct PWM. This results in different frequencies on different pins:

- Pin 4 and 13, controlled by timer0:
  16 Mhz / 64 / 256 = 976.56 Hz

- Other PWM pins, controlled by timer1-4:
  16 Mhz / 64 / 510 = 490.20 Hz

The calculation is: Clock / Prescaler / PWM mode number of states

The situation is the same for all other Arduino boards I know of, except they have less timers that connect to different pins.

You can change the PWM frequency by changing the timer prescaler. See this page: http://playground.arduino.cc/Main/TimerPWMCheatsheet

It is also possible to make the timers count to a different value than 255. On the 8 bit timers you will loose an output pin, but on the 16 bit timers you can use the Input Capture Register to define a TOP value. The input capture functionality is a feature rarely used in the Arduino community so you likely won’t miss it.

The Arduino libraries only allow you to use 8 bit resolution, even on the 16 bit timers. If you want the higher resolution, you will have to write your own analogWrite, or use a library made for the purpose. On the Arduino Mega timer0 and timer2 are 8 bit, while the rest are 16 bit.

By changing the mode of the 16 bit timers to make use of the full resolution, in combination with changing the prescaler and the TOP value, you can achieve a very wide range of PWM frequencies.

The maximum frequency you can achieve is clock / 4, by setting the prescaler to 1 and TOP to 3 in fast PWM mode - a lower value isn’t allowed. This will give you a 4 MHz PWM with 2 bit resolution. It can be on 0%, 25%, 50%, or 75% of the time. A higher TOP value will give you a higher resolution at a lower frequency.

For a longer explanation read this article or refer to the datasheet.
Changing the prescaler, PWM mode, or TOP value for timer0 will mess with \texttt{millis()} and \texttt{micros()}.

Tags: \texttt{pwm} (Prev Q)
The C programming language is the (direct or indirect) predecessor of many modern programming languages, although it is still actively used and developed. It is very similar to C++, except that C is not object oriented. That is, it will only support simple data structures which have no member functions.

**Note:** Arduino is usually programmed in C++, not C.

Please **do not** use this tag to indicate a general programming question. You may want to consider using the [programming](https://example.com/programming) tag instead.
Questions

**Q: Chain of, or one big if sentence**

**Tags: c (Prev Q) (Next Q)**

(This might be more of a common C question)

I want to test several conditions in an IF-sentence

```
if(variable) {
    // fastest check
    if(register) {
        // a bit slower
        if(read_peripheral) {
            // very slow check
            // [do something]
        }
    } else {
        // [do something]
    }
}
```

It seems obvious that I want to start with the fastest check, and move on to the slower ones afterwards, potentially avoiding to check them if the above was false.

Now since I need to check quite a few conditions, I’d like to know if the following is the same, or if all conditions will be evaluated prior to making a decision?

```
if(variable && register && read_peripheral) {
    // [do something]
}
```

**Tags: c (Prev Q) (Next Q)**

**User: allan-nørgaard**

**Answer by peter-r.-bloomfield**

Putting them all in a single condition like that is absolutely fine. C/C++ uses “short circuit evaluation” which means it will stop as soon as it’s determined the outcome. If everything is combined using logical AND operators (&&) then that means it will stop as soon as one of them evaluates to false.

For example:

```
if (blah() && foo()) { ... }
```

If `blah()` returns false, it will never call `foo()` at all because it already knows the overall outcome is false.

The reverse happens with the logical OR operator (||):

```
if (blah() || foo()) { ... }
```

If `blah()` returns `true` in this case, it won’t call `foo()`. That’s because it already knows the overall outcome is true.

**Tags: c (Prev Q) (Next Q)**

**Q: SAM3X8E (Arduino Due) Pin IO registers**
How does IO registers of Arduino Due work? On Arduino Uno just set DDRx, then PINx to read, PORTx to write, I’d like to do the same thing with an Arduino Due, but it has many more registers, such as PIO_OWER, PIO_OSER, PIO_CODR, PIO_SODR, etc. I find no correspondence between Arduino Uno and Arduino Due registers.

There are also some useful functions such as pio_clear, pio_set, pio_get, and others, all explained here: [http://asf.atmel.com/docs/3.19.0/sam3x/html/group__sam__drivers__pio__group.html](http://asf.atmel.com/docs/3.19.0/sam3x/html/group__sam__drivers__pio__group.html)

Now, I think I’ve understood what the three mentioned functions do, but not others, for example pio_configure (Pio *p_pio, const pio_type_t ul_type, const uint32_t ul_mask, const uint32_t ul_attribute), I can’t figure out what ul_attribute and ul_type are.

Thank you in advance

If you have a read of section 31 of the Datasheet, available from [here](http://asf.atmel.com/docs/3.19.0/sam3x/html/group__sam__drivers__pio__group.html), things may come a little clearer for you.

Here’s a summary of what I know:

PIO stands for Parallel Input/Output and offers the functionality to read and write multiple register ports at a time. Where the datasheet mentions a register, for example PIO_OWER, the Arduino library has macros for accessing them in this format REG_PIO?_OWER where ? is either A, B, C or D for the different ports available.

I tend to still use the slow Arduino pinMode() function to set input/output on the pins as it makes the code more readable than the acronym based registers calls such as REG_PIOC_OWER = 0xdeadbeef, but then use the direct registers to set the pins for performance/synchronisation. As yet, I haven’t done anything with input, so my examples are all output based.

For basic usage, you would use REG_PIO?_SODR to set output lines high and REG_PIO?_CODR to set them low. For example REG_PIOC_SODR = 0x00000002 would set bit 1 (numbered from zero) on PORTC (this is Due digital pin 33) high. All other pins on PORTC remain unchanged. REG_POIC_CODR = 0x00000002 would set bit 1 on PORTC low. Again all other pins would be unchanged.

As this is still not optimal, or synchronised if you are working with parallel data, there is a register that allows you to write all 32 bits of a port with a single call. These are the REG_PIO?_ODSR, so REG_PIOC_ODSR = 0x00000002 would now set bit 1 on PORTC high and all other bits on PORTC would be set low instantly in a single CPU instruction.

Because it is unlikely that you would ever be in a situation where you need to set all 32 bits of a port at the same time, you would need to store the current value of the pins, perform an AND operation to mask out the ones you want to alter, perform an OR operation to set the ones you want set high then perform your write and again, and this is
not optimal. To overcome this, the CPU itself will perform the masking for you. There is a register called OWSR (output write status register) that will mask out any bits that you write to ODSRs that don’t match bits set in the OWSR.

So, now if we call REG_PIOC_OWER = 0x00000002 (this sets bit 1 of the OWSR high) and REG_PIOC_OWDR = 0xffffffff (this clears all bits except bit 1 of the OWSR) and then call REG_PIOC_ODSR = 0x00000002 again, this time it would only change bit 1 of PORTC and all other bits remain unchanged. Pay attention to the fact that **OWER enables** any bits **that are set to 1** in the value you write and that **OWDR disables** any bits **that are set to 1** in the value you write. Even though I understood this when I read it, I still managed to make a code mistake when writing my first test code thinking that OWDR disabled bits that weren’t set to 1 in the value I wrote.

I hope this has at least given you a bit of a start in understanding the PIO of the Due CPU. Have a read and a play and if you have any further questions, I’ll try to answer them.

Edit: One more thing…

How do you know which bits of the PORTs correspond to which digital lines of the Due? Check this out: [Due Pinout](#)
Avrdude

Skip to questions.

Wiki by user jhuebsch

Official site: http://www.nongnu.org/avrdude/
**Questions**

**Q: Change Options for the AVR Programmer via Arduino IDE**

**Tags:** avrdude (Prev Q) (Next Q)

I want to program an ATtiny through the Arduino IDE. I have a somehow strange programmer. But I can upload via avrdude by

```
  avrdude -p attiny13 -P /dev/cu.usbmodemfa131 -c stk500v2 -F -B20 -U flash:w:programm.hex
```

It took a while to figure out that I had to slow the baud rate via the -B20 switch. If I try to upload via the arduino 1.0.5 IDE avrdude is executed as follows.

```
/Applications/Adafruit Arduino
1.0.5.app/Contents/Resources/Java/hardware/tools/avr/bin/avrdude -
C/Applications/Adafruit Arduino
1.0.5.app/Contents/Resources/Java/hardware/tools/avr/etc/avrdude.conf -v -v -v -v -p attiny13 -c stk500v1 -P /dev/tty.usbmodemfa131 -U flash:w:/var/folders/9t/5jldfq752fs1x74_rhn2plz80000gn/T/build76802016134264975-
```

My question is, how can I change the preferences such that I have control over the avrdude parameters? Say I want to add a -B20 switch and change stk500v1 to stk500v2.

**Tags:** avrdude (Prev Q) (Next Q)

**User:** a.schulz

**Answer** by jfpoilpret

**Short answer:** I’m afraid you can’t do it with Arduino IDE 1.0.5.

**Long answer:**

First of all, be aware that -B20 does not specify the baud rate but the bitclock period (in us); this is specific to stk500v2 programmer.

Normally, enabling your programmer should only be a matter of adding it to the list of programmers known by Arduino IDE; that list can be found in `hardware/arduino/programmers.txt`. You would then append the following lines at the end of this file:

```
strangeprogrammer.name=Somehow Strange Programmer
strangeprogrammer.communication=serial
strangeprogrammer.protocol=stk500v2
strangeprogrammer.speed=?????
```

However, the problem here is that Arduino IDE does not seem to be able to use flag -B (bitclock period) but only flag -b (baud rate) which value is set to whatever you will put to `strangeprogrammer.speed`. 
At least that’s what I could find out by inspecting Arduino IDE source code: `AvrdudeUploader.java` never adds that flag to the `avrdude` command-line :-(

That means your options are:

1. Rebuild the Arduino IDE 1.0 on your own after modifying `AvrdudeUploader.java` to support `-B` flag; code should be quite easy (about 2 more lines of Java code).
2. Switch to Arduino IDE 1.5 and follow Federico’s answer :-)  
3. Use another IDE, something that deserves the IDE name. I use Eclipse with this Arduino plugin and it works fine for me. As a bonus, avrdude support seems better as you can see on the screenshot below:

![Screenshot of Arduino IDE settings](image)

**Tags:** avrdude (Prev Q) (Next Q)

**Q: Understanding the compilation/linking/upload process (so I don’t have to use the IDE) ![ ]**

**Tags:** avrdude (Prev Q) (Next Q), ide (Prev Q) (Next Q)

I have started to play with and arduino UNO quite recently (without any prior experience with micro-controllers). I would like to use emacs instead of the IDE, and I’d also like to know what the IDE does under the hood, in order to be able to write my own makefile. The tutorials I’ve found are either outdated, or are presented as a series of steps without any explanation. I’d appreciate it if someone could explain to me how the whole
compilation/linking/upload process works using gcc-avr and avr-dude, and how it is used by the IDE.

Tags: avrdude (Prev Q) (Next Q), ide (Prev Q) (Next Q)

User: ash

Answer by ignacio-vazquez-abrams

If you want an exact duplication of what the IDE does but want it driven from the command line, that’s what Ino is for. The full Arduino build process involves copying a lot of files from a lot of places, and is generally not really trivial to duplicate.

If you’re ready to let go of .ino files and the Arduino libraries, you get a much simpler toolset. avr-gcc compiles, avrdude uploads, and you’re done. Here’s one of my makefiles from a simple project:

```makefile
Skip code block
CC=avr-gcc
CXX=avr-c++
CXXFLAGS=-Wall -Wextra -mmcu=atmega1284p -Os
CFLAGS=$(CXXFLAGS)
BINARY=ledmatrix
OBJECTS=

all: $(BINARY)
	@avr-size $<

$(BINARY): $(OBJECTS)

clean:
	@rm -f $(BINARY) $(BINARY).hex $(OBJECTS)

upload: $(BINARY).hex
	@avrdude -c usbasp -p m1284p -U flash:w:$<:i

%.hex: %
	@avr-objcopy -j .text -j .data -O ihex $< $@

.PHONY: all clean upload
```

If copying-and-pasting, be sure to replace all “aceous characters.

EDIT:

I have created a [repository with my buildsystem on Github](https://github.com/).

Tags: avrdude (Prev Q) (Next Q), ide (Prev Q) (Next Q)

Q: When using Arduino Uno as ISP does “Yikes! Invalid device signature” mean a bad connection, bad config, or bad version of avrdude?

Tags: avrdude (Prev Q) (Next Q), attiny (Prev Q) (Next Q)

I’m using an Arduino UNO to program a pre-compiled hex image to an ATTINY45, using the avrdude in the Arduino IDE directory, on Windows 7. The Uno has the ISP sketch loaded from the examples directory and that works- the heartbeat LED pulses nicely.
On the hardware side I have this setup, except the ATTINY is surface-mount, soldered to a surfboard with all its pins verified-soldered with a voltmeter. I use a header-pin strip, held by hand to get the pins touching the surfboard, to get the signals from the Uno to the ATTINY.

Here are the command lines, which I took from looking at the IDE’s debugging output:

```
REM set the fuse for 8MHz, so the ISP programmer can work
C:\Program Files\Arduino\hardware\tools\avr\bin\avrdude -CC:\Program Files\Arduino\hardware\tools\avr/etc/avrdude.conf -v -v -v -v -p attiny45 -c stk500v1 -P .\COM7 -b 19200 -e -Uefuse:w:0xff:m -Uhfuse:w:0xdf:m -Ulfuse:w:0xe2:m
REM load the program
C:\Program Files\Arduino\hardware\tools\avr\bin\avrdude -CC:\Program Files\Arduino\hardware\tools\avr/etc/avrdude.conf -v -p attiny45 -c stk500v1 -P .\COM7 -b 19200 -Uflash:w:firefly.hex:i
REM set fuse for 1MHz, as the project requires
C:\Program Files\Arduino\hardware\tools\avr\bin\avrdude -CC:\Program Files\Arduino\hardware\tools\avr/etc/avrdude.conf -v -p attiny45 -c stk500v1 -P .\COM7 -b 19200 -e -Uefuse:w:0xff:m -Uhfuse:w:0xdf:m -Ulfuse:w:0x62:m
```

-I try running those separately, but always get this error:

```
avrdude: Device signature = 0x000000
avrdude: Yikes! Invalid device signature.
   Double check connections and try again, or use -F to override this check.
```

but sometimes the number is ff0000 or ffff00 or ffffff

I read that “Arduino uses a slightly modified version of avrdude to upload sketches to the Arduino board. The standard version queries for the board’s device signature in a way not understood by the bootloader, resulting in this error.” Does this mean that using the Arduino avrdude with a new ATTINY chip, no bootloader on it, would also cause that same error? In other words, is the modified avrdude incapable of querying a non-Arduino-bootloaded AVR chip?

Or does that error simply mean I don’t have a good contact between all my programmer pins and the ATTINY?

And, are the fuse settings truly needed, can the Uno program the flash into an attiny running at 1MHz and thus save me some steps?

(I would just buy a “real” programmer, but need to get this code in the next two days, and am in a rural part of Nova Scotia)

**Tags:** [avrdude](https://www.arduino.cc) (Prev Q) ([Next Q](https://www.arduino.cc)), [attiny](https://www.arduino.cc) (Prev Q) ([Next Q](https://www.arduino.cc))

**User:** yary

---

**Answer** by yary

I soldered wires to pads to be sure the connection was good, and used those wires for programming from the Uno. Then the programmer worked, without any edits to the command lines as posted in the question. I ran a few variations and was able to answer all my questions, and a few more besides:

1. The “Yikes! Invalid device signature” error was being caused by poor contact between the programmer (Uno) and the ATTINY pins.
2. It had nothing to do with the bootloader not being on the new microcontroller. The new ATTINY45 chip doesn’t need a bootloader for the in-circuit programmer to work.
3. The Uno can program the ATTINY45 configured to either of its speeds, 1MHz or 8MHz. The factory settings are fine, and don’t require changing the fuses.

4. Both avrdude versions worked for programming- the 6.1 from the author’s site, and the modified one from the Arduino IDE.

5. avrdude can program via the Uno using any of -c avrisp, -c arduino, or -c stk500v1, it doesn’t matter which.

One little question I wasn’t able to answer, was what speed the Uno programs at (communication speed between the Arduino I/O pins & microcontroller being flashed). Is it related to the serial port speed? But it turned out to be not pertinent to answering this problem. Someone else can ask it if curious enough.

---

**Q: Extract .hex from ATtiny45 using Arduino**

**Tags**: avrdude (Prev Q) (Next Q), attiny (Prev Q) (Next Q)

As it is possible to program a ATtiny45 using Arduino as ISP Programmer, is there a way to backup an ATtiny45 program to a .hex file using Arduino?

**Tags**: avrdude (Prev Q) (Next Q), attiny (Prev Q) (Next Q)

**User**: alfonso-jiménez

**Answer** by ignacio-vazquez-abrams

Assuming the lock bits on the device haven’t been set, sure. Simply prepare the Arduino for ArduinoISP, hook up the ATtiny45, and perform a read using AVRDUDE instead of a write (-U flash:r:foo.hex:i).

**Tags**: avrdude (Prev Q) (Next Q), attiny (Prev Q) (Next Q)

---

**Q: Problem programming Mini Pro via Arduino as ISP**

**Tags**: avrdude (Prev Q) (Next Q), clones (Next Q)

I just picked up a pair of Arduino Mini Pro 5v clone boards (Inland, sold at Microcenter, currently $4 USD) and I wanted to use my Uno as an ISP like I’ve done with ATmega328 and ATtiny85 chips before. I’ve programmed the Uno with the instructions at the ArduinoISP tutorial page and I’ve connected up the Uno to the Mini Pro using the instructions at this page. I cannot get the Arduino IDE or avrdude from the command line to successfully program these boards no matter what I try. Except that I can use the Arduino IDE to burn a new bootloader to the Mini Pro.

I’ve connected up the status LEDs as well (pins 7, 8 and 9), and I can see that the programmer has an error condition (pin 8) when I attempt to program anything besides the
Here is my avrdude command line where I tried to set the fuses on the Mini Pro:

```
avrdude -P /dev/tty.usbmodema0121 -b 57600 -c avrisp -p atmega328p -U lfuse:w:0xff:m -U hfuse:w:0xdb:m -U efuse:w:0xfd:m
```

I get the following back from avrdude:

```
Skip code block
avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 2 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 3 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 4 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 5 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 6 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 7 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 8 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 9 of 10: not in sync: resp=0x1c
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x1c
avrdude done. Thank you.
```

I’ve not had any problems with using pretty much this exact same setup, but with a ATmega328 or ATtiny85 chip on a breadboard with avrdude in the past, so I’m at a loss as to what is happening here.

I’ve also taken a look at the ArduinoISP code, and I’m not really sure what the resp=0x1c is. I don’t see a 0x1c value defined in the STK Definitions section of the code, so I’m not sure what might be causing this.

Any pointers here would be appreciated. And yes, I do know that I can use the Uno board with the ATmega328 chip pulled to use as a programmer as well. But I want to be able to overwrite the bootloader area with my own code like I can with a regular ATmega328 chip.

Thanks!

**Tags:** avrdude (Prev Q) (Next Q), clones (Next Q)

**User:** dennis-munsie

---

**Answer** by dennis-munsie

After poking around for a bit more, it turns out my problem was the baud rate. I could make it work by setting it to 19200, just a little slower. Alternatively, you can modify the ArduinoISP sketch to enable higher baud rates. I used this post to enable 115200 baud and I was able to successfully use avrdude at the higher rates.

Not sure how I missed this before, but it’s an easy to solve at least.

**Tags:** avrdude (Prev Q) (Next Q), clones (Next Q)

---

**Q: Remove Bootloader on Arduinos**

**Tags:** avrdude (Prev Q)

I am trying to move from arduinos to AVR C. Would somebody know how to remove the
arduino bootloader from the microcontroller? Is there a different process for the different atmega microcontrollers like the 32u4, 328, or 2560?

Thanks.

Tags: avrdude (Prev Q)

User: naan

---

Answer by majenko

You don’t “remove” the bootloader, you just ignore it. When you program your new code with the hardware programmer of your choice it will just overwrite the bootloader code with your software’s startup code.

Tags: avrdude (Prev Q)
Battery

Batteries are electrochemical devices. That is, they store energy in chemical form, and can convert it to electrical energy when required. They are available in a very wide variety of configurations, producing different voltages, with different power output capabilities.

Some are single-use devices which are disposed of when depleted. In consumer products, these are often referred to as alkaline batteries. Others types of battery may be recharged repeatedly for multiple uses, such as lithium ion or lead acid batteries.

This tag should be used for questions about using batteries with Arduino devices. This can include using batteries as a power supply, but may also cover projects where the Arduino controls the batteries in some way (e.g. as part of a recharging circuit).
Questions

Q: Bootup on lower power not functioning

Tags: battery (Prev Q) (Next Q)

I have a custom Arduino ATMega328 board that generally runs at 5V @ 8MHz (using the Arduino Pro 3.3V 8MHz profile and bootloader). The main reason I’m using this setup is so that I can put the board to sleep when main power is disconnected and it starts being run off of battery power (3V from a coin cell). The 5V and 3V sources are diode OR’ed together and the 5V input is tied to INT0. In code, when it detects that INT0 has fallen low, it initializes sleep mode and everything powers down with the exception of the watchdog timer that keeps a 1Hz cycle to keep an internal count and check if the chip should be woken back up. This works beautifully when 5V power is applied first, then the battery is inserted, then 5V is disconnected. It goes to sleep and when 5V is brought back it wakes up and I can see it hasn’t lost count.

However, the problem comes when 3V is applied first. I’m honestly not sure if it’s even booting. But what it is supposed to do is boot, check if INT0 (Digital 2) is low and, if so, go right to sleep. By watching the current draw I see that it powers up to a few mA for a couple seconds, then drops to about 0.3mA (still higher than it should be in sleep mode). But when I re-apply 5V, nothing. The power draw goes back up but it is unresponsive (over FTDI serial).

Is there maybe something I’m missing that it can’t be booted on 3V… in theory it should run just fine.

Update: I dropped an LED onto D13 and tried the blink sketch. Works fine when starting from 3V or 5V. However, when I run my firmware and start it from 3V, the LED just starts flashing wildly. I have no idea what’s causing it since I never even setup D13 as anything in my code. But it makes me thing it’s something to do with the bootloader…

Tags: battery (Prev Q) (Next Q)

User: adam-haile

---

Answer by adam-haile

So… turned out what was happening was all in code. Turns out that I was initializing an interrupt on INT0, when it was low, at the very beginning of my code. Problem was that when it started up on 3V backup power, INT0 was always low because INT0 is tied to the 5V line (it’s how it knows to go to sleep). Because INT0 was low and the interrupt was triggering on low it was causing a constant interrupt to occur, never giving the rest of the program any time to run. Switched to enabling that interrupt only once I’ve entered full waking mode and it works fine now.

---

Answer by imjosh

Exactly what is happening and why cannot be determined based on the amount of info provided. However, I see at least one potential problem which would at least partially
explain the symptoms described.

You said that you are using diodes to select the voltage supply, and one supply is a 3V battery. If you are using standard diodes that drop ~0.6V then the supply voltage to the MCU is only ~2.4V. If you are using Schottky diodes with a voltage drop between 0.15-0.45, the supply voltage is potentially as low as 2.5V. You have the BOD voltage set to 2.7 volts, so in theory the MCU will never boot with the battery.

As for why you can start it at 5v, drop to 3v, and bring it back up again- I’m not sure. You could be disabling the BOD in code…maybe… Not sure why it works, but it’s likely not guaranteed to work.

I setup a diode switch circuit with 5v and 3.3v to see what it looks like on my oscilloscope when the voltages switch. When at 3.3v switching up to 5v, the voltage oscillates quite a bit initially. This may potentially cause some problems when the MCU tries to come out of sleep. Putting a cap between VCC and GND smoothed the signal very nicely. When switching from 5v to 3.3v, there really wasn’t any oscillating, just a clean drop.

From this information, it seems that you should lower the BOD threshold or turn off the BOD, and put a decoupling cap between VCC and GND. You probably also should make sure you have a pull-down resistor on INTO, and read the MCU datasheet sections explaining the various sleep modes all the considerations for sleeping and waking – it’s pretty involved. Cheers

---

Answer by thedoctor

I think the problem is with your bootloader, but not in the way you think.

When an Arduino starts up, a bootloader profiting several tasks, such as looking for a serial host or loading program data from flash is going to take a lot more energy than a sleeping Arduino.

I think the reason why your Arduino will not start is because the bootloader requires an accurate clock source, but applying 3v to it will mess up that clock and it will perhaps crash or wait for 3.3v to be applied for a stable source.

Looking at the datasheet:

\[ T_A = -40^\circ C \text{ to } 85^\circ C, \ V_{CC} = 1.8V \text{ to } 5.5V \text{ (unless otherwise noted)} \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Condition</th>
<th>Min.</th>
<th>Typ.(^{(2)})</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I_{CC}</td>
<td>Power Supply Current(^{(1)})</td>
<td>Active 1 MHz, (V_{CC} = 2V)</td>
<td>0.2</td>
<td>0.5</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Active 4 MHz, (V_{CC} = 3V)</td>
<td>1.2</td>
<td>2.5</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Active 8 MHz, (V_{CC} = 5V)</td>
<td>4.2</td>
<td>9</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle 1 MHz, (V_{CC} = 2V)</td>
<td>0.03</td>
<td>0.15</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle 4 MHz, (V_{CC} = 3V)</td>
<td>0.2</td>
<td>0.7</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle 8 MHz, (V_{CC} = 5V)</td>
<td>0.9</td>
<td>2.7</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power-save mode(^{(3)})</td>
<td>32 kHz TOSC enabled, (V_{CC} = 1.8V)</td>
<td>0.75</td>
<td>(\mu A)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>32 kHz TOSC enabled, (V_{CC} = 3V)</td>
<td>0.83</td>
<td>(\mu A)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power-down mode(^{(3)})</td>
<td>WDT enabled, (V_{CC} = 3V)</td>
<td>4.1</td>
<td>8</td>
<td>(\mu A)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WDT disabled, (V_{CC} = 3V)</td>
<td>0.1</td>
<td>2</td>
<td>(\mu A)</td>
<td></td>
</tr>
</tbody>
</table>
We see that current draw while waking up (running the bootloader) is probably going to be around 2mA, while sleeping current is about 0.8 microamps. This could certainly lead to a blackout on startup without the proper current.

Perhaps you should only turn it on with 5v power, or maybe you need a higher current, higher voltage battery.

---

**Q: Most power efficient way to constantly save 2 kilobytes of data per second**

Tags: battery (Prev Q) (Next Q)

I would like to constantly save data coming from sensors (for an activity tracker) while consuming as less battery as possible (I’d like to have a 4/5 days autonomy on a 3000 mAh battery) but I can’t find the best way to save about 2 kBytes/s on a memory large enough to contain several days of data.

Sensors would be read at a rate of 20 times per second and produce about 100 bytes each time.

What options do I have in order to:

- store such an amount of data (hardware)?
- ensure the system can work properly with required autonomy?

Tags: battery (Prev Q) (Next Q)

User: trevör-anne-denise

Answer by cybergibbons

There are two constraints here, and I think it might back you into a corner. Size and then power.

2Kbyte/s is 2*60*60*24/1024 = 168.75Mbyte/day. This is a lot. The only readily available technology that can be used with a small microprocessor that can store this volume of data is an SD card. There are no serial EEPROM or flash chips this large, and the Arduino, as far as I know, can’t deal with the NAND flash used in USB sticks and SD cards.

However, SD cards can be relatively power hungry. They draw between 5mA and 100mA when writing, typically around 30-50mA. The SD spec says the maximum is 200mA. Writing is also quite lengthy as they operate on blocks of data rather than individual cells. I’ve never done a full analysis of how much power they use, but it is not insignificant and much, much more than serial EEPROM or flash.

So your only choice, if you have to store that much, is to go with SD cards.

I would investigate compressing the data. Even very basic schemes can make a huge difference.
**Q: Day-night powered Arduino**

What should I do to have an Arduino Pro Mini 5V (clone) powered by a 5V solar panel during the day and by battery during night which will be charged during the day by the solar panels.

- is there any guide out there?
- is it possible with the 5V version of Arduino Pro Mini or will I have to buy the 3.3V
- I prefer to make the circuits by myself rather than buying them (for example: the charging circuit.)

**User: christos-mitsis**

**Answer** by russell-mcmahon

This would be better answered on Stack Exchange Electrical Engineering.

Also see this question and answers. This is not quite the same and the answers there are OK but less than complete or correct, but it adds to the resource available.

You need a battery charging controller, a PV (solar) panel capable of producing somewhat more than maximum battery voltage and enough energy to operate your circuit and a battery with enough voltage to operate the device and enough capacity to store the energy.

Current drain will vary with total circuit. You can measure this will a milliammeter. I’ll assume that 50 mA continuous will be enough. Less or much less may suffice. I have a pro mini clone operating at 20 mA on 5V at present and with due care you may be able to operate one with less or much less current depending on application and what your circuit needs are.

Looking at solar current needs at present, the average solar current needed is about \( I_{\text{circuit}} \times 24/\text{SSH} \) mA where \( \text{SSH} \) = average sunshine hours per day. eg if you had 4 hours of sunshine per day but want to run the circuit for 24 hours a day the sun must provide \( (24 \text{ hours per day)/(4 \text{ hours of sun})} = 6 \) as much current as the circuit needs on average. So if your circuit needs 50 mA continuous the PV panel must provide \( 6 \times 50 \text{ mA} = 300 \text{ mA} \) for 4 hours per day. You will see below that this is not a trivial amount.

Sunshine hours per day may be found for sites worldwide using the excellent www.gaisma.com The average sunshine hours per day for Iráklion may be found at that link and other locations in Greece can be found here but will probably be similar. For Iráklion the SSH may be found in the 1st line of the 4th chart.

Average hours/day of equivalent full sunshine = kWh/m^2/day = a peak of 8.53 hours in June and a low of 2.08 hours in December.

Wow! I think that is about as sunny a place in Summer as anywhere on earth not in the
polar regions. In Winter it gets about the same SSH as in my city. You can calculate PV sizing for peak summer - but if you want year round operation you should use the worst case winter month and add a bit for bad days.

For Iraklion we say 2 SSH per day so the panels needs to provide $24/2 = 12 \times$ winter average load. So per 10 mA of load you need 120 mA of panel and if the mean load is 50 mA you need $12 \times 50 = 300$ mA panel output.

Battery voltage needs to be $> 5$V at all times.

An easy cheap available starting solution is to use a 6B lead acid battery. These are easyish to charge and provide a relatively stable operating voltage.

Vmax_charge is about 7V so an 8V or 9V PV panel is “about right”. 9V is a common voltage for hobby applications. An alternative is to use a 12V panel. this is probably more suitable as they are widely available and cost per energy out is often lower. This wastes MOST of the panel output but still MAY be the cheapest option in many cases. (A 12V x 600 mA panel is rated at about 18V x 0.6A ~ 10 Watts. Actual Arduino load = 50 mA x 5V x 12:1 = 3 Watts)

At 50 mA load you need $24/2 \times 50 \text{mA} = 600 \text{mA}$ PV output in full sun.

PV panel sizing is not as may be expected. It is easiest to get a panel that makes at least enough voltage and enough current. This is because panel RATED voltages are usually higher than the voltages they are used at. eg 12V system PV panels are usually rated at 18V max power voltage- so an 18W panel provide 18V at 1 A BUT if attached directly to a 12V battery the input to the battery = 12V x 1A = 12 Watts. There are ways around this but at first, just concentrate on panel mA.

So, you “need” a 600 mA panel for a 50 mA load.
Or a 60 mA panel for a 5 mA load.
Or a 6 mA panel for a 0.5 mA load.

Obviously, reducing load current is going to help panel sizing.

So knowing what your application really needs is essential.

Batery is easier.

If you use 6V lead acid then for one day you need.

Hours x load = mAh.

Here load=24 hours. mA = 50 in this example.

mAh = $24 \times 50 \text{mA} = 1200 \text{mAh} = 1.2 \text{Ah}$. This is the capacity of about the smallest LA battery commonly available. In practice you want to discharge your battery by only a small % of its actual capacity so use a 2.4 Ah or 3.5 Ah or larger. 7Ah is not vastly dearer and will last far longer and will have several days of reserve capacity for bad weather.

Charging can now be discussed - but I’ll stop here and see what is said about the above. We can add charging and other detail if it seems useful to do so.
An easy battery to use is 2 x LiIon or LiPo cells in series.

Tags: battery (Prev Q)
The process of programming consists of several stages, including:

- editing the source code
- managing libraries and build events
- compiling/linking the program (or sketch)
- debugging the program when it is running

In the case of Arduino, it typically also involves uploading the sketch to a device.

Each part of this process is often managed by a separate program. These are often command-line based, and can be quite difficult to learn. However, an Integrated Development Environment (IDE) combines everything into one application, and often automates several common tasks. This simplifies the programmer’s workflow, allowing him/her to focus more on the project at hand.

The official Arduino software is a very light-weight IDE, based on the Processing Development Environment. It includes all of the essential functionality for writing and uploading Arduino sketches, but lacks several features which are found in more sophisticated applications.

Some users prefer more mainstream IDEs, which can have plug-ins or extensions to support Arduino. These include Eclipse, Visual Studio, and XCode. More information about alternative IDEs can be found on this question.

If the question is about the official Arduino IDE use the arduino-ide tag.
Questions

Q: How do I debug on-chip with Arduino?

Tags: ide (Prev Q) (Next Q), emulation (Next Q)

Though an emulator is quite useful for debugging code that has no side effects, most real-world applications include some circuitry besides the Arduino. If a microcontroller on an Arduino had a JTAG interface, it would be pretty straight-forward, but what if it’s e.g. an Arduino Uno?

Tags: ide (Prev Q) (Next Q), emulation (Next Q)

User: polkovnikov.ph

Answer by adam-davis

The Arduino Uno uses the Atmel Atmega328p microcontroller, which only has one debug option, DebugWire. There are a few tools that can work with it, but the least expensive and most commonly available is the AVR Dragon.

You can use AVR Studio 6 for this. You can use the .elf file file produced during the Arduino build process directly, but setting up the debugger is a little easier if you port the Arduino program to the AVR Studio. It’s not difficult, but there are a lot of little pieces to deal with. There are guides online to help you through this.

Tags: ide (Prev Q) (Next Q), emulation (Next Q)

Q: How to debug an Arduino sketch on a Mac?

Tags: ide (Prev Q) (Next Q)

I have 2 Lowpowerlab WaterMote’s talking to each other using the ‘out of the box’ pre-installed sketch that just confirm they’re talking to each other. I verified this by opening a COM connection and saw traffic of ‘receiving packets…’. That was just to show that they are functioning and talking to each other.

Now, using Arduino IDE on Mac OS X 1.9.x (Lion?) I have loaded the WaterMote sketch (found on Github) on the Moteino with the sensor (not the gateway) as it should be. The traffic looks like it’s working, except the text being transmitted (number of Gallons) stays the same at 0.02 Gallons.

So… I want to debug the WaterMote sketch, but I do not see an option in my Arduino IDE on Mac OS X. Is there a way to debug sketches using Macs? Searching online shows various and preferred options, but mostly on Windows (of which I don’t have any at home).

My current setup

- Moteino Gateway (as out of the box) connected to a Raspberry Pi (with ssh
enabled)
• Moteino with WaterMote sketch loaded
  • Connected via USB to the MacBook Pro
  • Connected via USB to the RasPi
  • Arduino IDE running, but I see strange “[][][]” square output in the Serial connection.
  • I do not know what to make of this since that text is not in the WaterMote sketch.

Is there a way to debug sketches loaded on Arduino clones using Arduino IDE on Macs?
I found a debugging article:

• Visual Micro

Tags: ide (Prev Q) (Next Q)
User: jose-leon

Answer by zmo

So... I want to debug the WaterMote sketch, but I do not see an option in my Arduino IDE on Mac OS X. Is there a way to debug sketches using Macs? Searching online shows various and preferred options, but mostly on Windows (of which I don’t have any at home).

It’s definitely possible to debug a sketch with OS X. But it’s definitely an advanced AVR topic, and if you’re not experienced enough with embedded programming you should first try to avoid it, except if it is really necessary.

Debugging a sketch for an ATmega328 implies:

• Running a flasher such as the AVR Dragon, the JTAG ICE 3, the STK500 or the AVR ONE; no cheap flasher implements the DebugWire protocol;
• Setting up the debug fuse, which is an operation that can brick your MCU.

If you’re using an ATmega32U4 (or alike), you won’t be able to do it out of the box, as the 32U4 needs a JTAG connection for on-chip debugging.

Finally, once you got through that, you can bind a gdb to the AVR using AVaRICE. I’m not getting you through all that, because I really, really, strongly advise you not to do it until you’re experienced enough to know what you’re doing.

Moteino with WaterMote sketch loaded connected via USB to the MacBook Pro Arduino IDE running, but I see strange “[][][]” square output in the Serial connection. I do not know what to make of this since that text is not in the WaterMote sketch. Is there a way to debug sketches loaded on Arduino clones using Arduino IDE on Macs?
It is very likely you did not setup the correct baudrate for the serial outputs. Arduino IDE defaults to 9600 bits/s connections, whereas the WaterMote source code defines a 115,200 bits/s connection. When you open the Serial Monitor change the setting at the right of the status bar.

And finally, be sure to uncomment the `#define SERIAL_EN line` to enable the serial communication and debug outputs.

**Q: Can I switch from Arduino IDE to Atmel studio or Visual Micro but still use Arduino library?**

If switch from Arduino IDE to Atmel studio or Visual Micro, how to use existing Arduino library?

Library serves important function of ‘packaging’ real and complex hardware (like NRF24L01 radio board) into functions of ‘relatively standard format’, like radio.init, radio.write, radio.read, radio.on, etc.

This allows user to easily use the hardware without reading the data sheet.

How can the same be achieved under Atmel studio?

User: john-williams

Answer by majenko

At the end of the day, an Arduino library and the Arduino core are just files full of code. There is nothing to stop you creating a project in Atmel studio (or wherever) and importing all those files into your project.

You will need all the core files (hardware/arduino/core/arduino/*) plus all the files for the libraries you want to use.

**Q: Arduino screws up trivial program involving enum**

Behold this insanity:

```c
enum WhatArduinoIs { Easy, But, Insane, Obsolete, And, Far, Worse, Than, mBed };

void TellMe(WhatArduinoIs pls) { }
```
I’m pretty good at C++, so I’m fairly sure that should compile. However I get this error:

```
void setup() {}
void loop() {}
```

```
void setup()
{
}
```

```
void loop()
{
}
```

```
I’m pretty good at C++, so I’m fairly sure that should compile. However I get this error:
```
sketch_jan21b.ino:3:13: error: variable or field 'TellMe' declared void
sketch_jan21b.ino:3:13: error: 'WhatArduinoIs' was not declared in this scope
Error compiling.
```

I’ve encountered the Arduino “IDE”‘s propensity for insane and totally unrobust modification of programs before. Specifically when you `#include` libraries Arduino only magically adds the relevant C++ files if the `#include` is in your main sketch. `#include <LiquidCrystal.h>` in another C++ file? Link errors.

I suspect in this case it is Arduino magically trying to add function declarations to the start of the file so their order doesn’t matter. I.e. it turns the file into this:

```
namespace {
    enum WhatArduinoIs {
        Easy, But, Insane, Obsolete, And, Far, Worse, Than, mBed
    };
    void TellMe(WhatArduinoIs pls) {}
}
```

That obviously doesn’t work. How can I tell Arduino to leave my damn program alone?

**Tags:** ide (Prev Q) (Next Q)

**User:** timmmm

**Answer** by peter-r.-bloomfield

You can workaround the issue by putting your `enum` and function into a namespace. You can even take advantage of C++ unnamed namespaces if you like:

```
namespace {
    enum WhatArduinoIs {
        Easy, But, Insane, Obsolete, And, Far, Worse, Than, mBed
    };
    void TellMe(WhatArduinoIs pls) {}
}
```

```
void setup() {}
void loop() {}
```

It’s annoying (as are many things in the Arduino IDE), but it seems to work.

If you’re more experienced with programming then you might want to consider using a 3rd party IDE with an Arduino plugin, such as Eclipse. The Arduino IDE is really only designed for beginners.

**Answer** by brettam

You seem to be correct about what the arduino IDE is doing to your sketch.
Arduino will not preprocess any .h or .cpp files, so you could use a second file. you could also break the IDE’s function locating regex with a dummy `throw()` statement like this:

```c
void TellMe(WhatArduinoIs pls) throw() { }
```

The dummy throw allows it to compile for me on 1.5.8

Tags: ide (Prev Q) (Next Q)

**Q: List of Arduino board preprocessor #defines**

Tags: ide (Prev Q)

When one selects a board within Arduino IDE, a preprocessor definition is added to one of the behind-the-scenes files.

After a lot of hunting and some good fortune I found that the format of this definition is:

```
#define ARDUINO_<PROCESSOR-DESCRIPTOR>_<BOARDNAME>
```

Some are easy to guess (ARDUINO_AVR_UNO, for example), but others are less so. The Pro Micro has ‘16’ or ‘8’ appended to the definition depending on the speed. I do not know if the definition is different for 5V or 3.3V. I haven’t managed to guess the definition for the Mega2560, but it isn’t anything obvious.

**Question 1:** Is there a list in existence of the possible definitions?

**Question 2:** Is there any distinction, as far as compilation and preprocessor involvement is concerned, between BoardX-5V and BoardX-3.3V, and how is this distinction defined?

Tags: ide (Prev Q)

User: charliehanson

**Answer** by mikael-patel

The list of board symbols can be generated by this shell command:

```
$ grep board= boards.txt | cut -f2 -d= | sort -u
AVR_ADK
AVR_BT
AVR_DUEMILANO
AVR_ESPLO
AVR_ETHERN
AVR_FIO
AVR_GEMMA
AVR_LEONARDO
AVR_LILY
AVR_LILYPAD
AVR_LILYPAD_USB
AVR_MEGA
AVR_MEGA2560
AVR_MICRO
AVR_MINI
AVR_NANO
AVR_NG
AVR_PRO
AVR_ROBOT_CONTROL
AVR_ROBOT_MOTOR
AVR_UNO
AVR_YUN
```
The boards are defined by the “build.board” property in the `boards.txt` file.

```
mini.build.board=AVR_MINI
```

This property is used by the build recipe together with ARDUINO_-prefix.

```
-DARDUINO_{build.board}
```

Cheers!

---

**Answer** by [ignacio-vazquez-abrams](https://www.example.com).

The first blank is the *platform*. This is “AVR” for AVRs, “SAM” for SAM-based Arduinos, etc. This is derived from the platform directory containing the core.

The second blank is the *board*. This comes from the entry in `boards.txt` in the core itself, and is the identifier before the first period.

There is no difference between compilation processes with regards to voltages; any speed difference is given in `F_CPU` and the board itself should not be checked for this.

So there is no definitive list, since the list is of arbitrary size due to its source.

---

**Tags:** ide (Prev Q)
Motor Questions

**Q: Can a stepper motor be controlled directly from an Arduino?**

Tags: motor (Prev Q) (Next Q)

I’m planning to build a simple XY plotter, and have been looking at the possibility of using a pair of small 5V 4-phase steppers to provide the movement. In my research, I’ve often seen stepper driver boards advertised, ranging from very big and expensive, through to quite small and cheap. Some of them actually look like little more than a darlington array on a PCB.

Do I really need one of those external driver boards, or can I control the steppers directly from my Arduino?

Tags: motor (Prev Q) (Next Q)

User: peter-r.-bloomfield

---

**Answer** by thedoctor

Generally, stepper motors cannot be controlled by an Arduino pin. They have a current draw of over the 20mA any single pin can supply.

Kickback is also a problem. Because the motors have moving magnetic parts inside, they will continue to generate electricity after power has been cut. this will almost certainly cause enough negative voltage that is enough to fry your Arduino.

A solution is to use a motor driver chip or shield. This one is a good example. These have transistor based chips that can drive enough current to power a motor, and diodes to mitigate the kickback from the motors.

---

**Q: Control/count rotations of a DC motor**

Tags: motor (Prev Q)

Using an Arduino and an L293D IC, can I control the number of rotations a DC motor makes? Or can I only control the direction and speed of the motor?


Each wheel is controlled by a DC motor. I followed Adafruit’s tutorial...
(https://learn.adafruit.com/adafruit-arduino-lesson-15-dc-motor-reversing) to control the motors with the help of an L293D IC.

Now I realize that I can’t directly control the rotations of the motor. I can control the direction, voltage, and time of the rotation. For example: turn forward at 50% voltage for 500 milliseconds.

But that’s difficult to translate into actual rotations. The speed of the motor varies according to voltage (like if I switch from 2AA batteries to 4AA’s) and weight (adding sensors slows the motors down). Every time I change voltage or weight, I have to guess how much voltage/time causes a single rotation.

I think I should just buy stepper motors. Before I do that, I’ll ask the community: Is there a way to control DC motors by rotations rather than time?

**Tags**: motor (Prev Q)

**User**: michael-cornn

---

**Answer** by karl-bielefeldt

As you’ve discovered, there are a lot of variables involved, so you need some sort of feedback. A popular way to do this is with an encoder, but depending on your needs, other kinds of sensors can make do. For example, if your problem is keeping the robot going in a straight line, an electronic compass can help. For a line following robot, the line sensors are usually enough. Range sensors can track your distance. You get the idea.

---

**Answer** by gwideman

Stepper would be heavy and battery consuming, as Ignacio says.

Pololu makes an encoder for a wheel: [http://www.pololu.com/product/1217], but obviously you can implement this in other ways too.

---

**Tags**: motor (Prev Q)
Voltage Level

Questions

Q: Arduino Pro Mini (3.3V version) input voltage range / tolerance

Tags: voltage-level (Prev Q) (Next Q), clones (Prev Q) (Next Q)

I have a few Arduino Pro Mini clones (cheap Chinese stuff) and would like to power them with 12V power supply (same as fan voltage). According to the Arduino Pro Mini spec the RAW pin can take 3.35-12V (3.3V model). In practice this means a 12V PSU cannot be used as they are almost always over 12V with low load. I read that the voltage regulator in this board is capable up to 16V input. I tried plugging in a cheap 12V power adapter that read 15.1V with no load but a component on the arduino clone board actually exploded instantly. The board seems to work still when powered from usb programmer. Component that burned out is just above VCC, RST and GND pins.

Why did this happen? What did I just damage and most importantly what is a safe voltage level to use? I already have a few 12V switching power supplies it would be a shame if I could not use them.

Solution:

In the end it seems to have been a faulty arduino clone, faulty or poor quality power adapter or the fact that I powered the power supply before plugging it to the board’s RAW input pin. The cheap 12V switching power supplies work just fine even though my multimeter registers a voltage spike as high as 30V during power up.

Tags: voltage-level (Prev Q) (Next Q), clones (Prev Q) (Next Q)

User: dominicm

Answer by jfpoilpret

The component that exploded during your test is a capacitor.

According to the position on the board you mentioned, it is one of the 2 electrolytic capacitors used on both sides of the 3.3V regulator (I would say that’s probably the upstream capcitor).

There are a few reasons why a capacitor may explode:

- apply an inverted voltage to it (polarized capacitors only)
- apply a voltage above its rating
- apply a current above its rating

Your situation is probably the 2nd one: too high voltage applied.

According to Sparkfun original schematics, this capacitor must be polarized with 10uF
capacity. Unfortunately the schematics do not mention the maximum voltage.

According to the [specification for the original board](#), the maximum voltage for this cap should be at least 12V, but there is no guarantee that it is anthing above than 12V. Hence, you have to consider 12V is the maximum voltage applicable to your board (on the RAW pin).

In addition, you have mentioned you are using a cheap chinese clone, it is perfectly possible (although not expected, as it would make the board operate differently than the original specs) that this one uses a lower voltage cap (eg 10V) which would be a bit cheaper.

If you do want to reuse your 12V PSU, you will have to ensure that its output voltage never exceeds 12V, for this you are left with a few options:

- use a [voltage divider](#) made of 2 resistors: you must first know the max voltage output of your PSU and then calculate resistance values to get 12V based on that max voltage. That’s easy but it can waste a lot of energy
- use a 12V [Zener diode](#) outside your PSU
- use a 12V [voltage regulator](#) circuit after your PSU: that one is the safest, but also the most expensive solution

**Edit:**

Of course, if you have already done it and are equipped for it, you can find replacements for the capacitor that has exploded and buy a higher-voltage one, then replace it on your boards; that will reauire good soldering/desoldering equipement usable for surface-mounted components.

**Q: Level detection of 3.3V from 5V arduino**

**Tags:** voltage-level (Prev Q) (Next Q), clones (Prev Q) (Next Q)

A quick question about microcontroller digital level on I/O pins. I have a photo interrupter which is powered at 3.3V which is interfaced to Arduino UNO running at 5V.

For microcontroller to detect high or low level is determined by signal higher than 5V*(2/3) = 3.33V is high and signal lower than 5V*(1/3) is low. What I don’t understand is that arduino is able to detect the level change from the photo interrupter. Doesn’t the supply on pins have to be greater than (2/3) or Vcc in order for it to detect logic high? The circuit works and I am able to count pulses from the interrupter but I want to know why that works considering the photo interrupt only gets about 3.23V.

Please clarify

**Tags:** voltage-level (Prev Q) (Next Q)

**User:** david-norman
You’ve… misread the datasheet. The ‘328 input high voltage (V_{IH}) for most pins is 0.6V_{CC} minimum for devices with a 2.4-5.5V supply. This means that a 5V device has a 3V threshold.

The designed threshold of the ‘328 logic is 45% of V_{cc} for all supply voltages, except the tolerance reduces from +/-25% of V_{cc} {<3V} to +/-15% of V_{cc}{>3} This tolerance is due to temperature and process variation.

Keep in mind that the smaller the difference between the “actual” threshold and your input is your immunity to noise and it will work if there is low noise. Thus 45+15% = 60%* 5V=3V is guaranteed. What is not guaranteed, is the accuracy of your 5V regulator which if say 10% high will increase the threshold 10% above 3V or 3.3V. So then 3.1 would not be guaranteed. So ensure accuracy of 5 V is below 5.04 minus your noise immunity requirements.

I would like to set up a project with an ATtiny chip, such as the ATtiny84 or ATtiny85, in which I interface the microcontroller directly with an XBee for both wireless serial communication, as well as monitoring a digital signal from the Xbee (digital input to the ATtiny). As the Xbee chips can handle only 3.3V and will be damaged at higher voltages (such as 5V), I was wondering whether I can simply power the entire project at 3.3V so that all digital inputs and outputs will be at that voltage?

Yes. As given in the datasheets, normal ATtiny4/X5 chips can run with a supply between 2.7 and 5.5V inclusive, whereas the low-power versions can run between 1.8V and 5.5V. Mind the speed restrictions when running at lower voltages though; see the “Electrical Characteristics” section of each datasheet to determine the maximum characterized speed for a given voltage.
I just ran into this article: Arduino: What adapter?

It says that the external power supply should be 9-12V. That really surprises me, since my Arduino Uno can also be powered via USB, which is 5V.

Later he talks about a 9V pin. But I can only see a 5V and 3.3V pin on mine. Is he talking about a different version of Arduino?

Judging from the picture, his Arduino looks slightly different from my Arduino Uno R3, but I can’t tell which version is on the picture.

---

**Answer** by sachleen

The board has a voltage regulator which converts 9-12V to the 5V that the chip uses.

Why does it need 9V if the board works at 5V? Well, it doesn’t need 9 exactly, but it needs something greater than 5. The reason for this is that the regulator has what’s called a dropout voltage.

Looking at the schematic of the Uno we see it uses a NCP1117, which has a dropout of 1.2V. So the minimum voltage you can give it to get 5V is 6.2.

The 9V pin is the Vin pin which is at the same level as the input voltage (before converting).

---

**Answer** by duncan-c

And the third part of your question: It is possible to bypass the voltage regulator and feed REGULATED 5 volts directly to the 5V pin.

That’s how the Arduino is powered using USB. The USB power is already regulated 5V, and is fed directly to the 5V pin.

Beware of doing this yourself however. You better be certain the 5V you’re feeding in is a clean, filtered and regulated 5V. If it’s not you run the risk of destroying your Arduino (and possibly anything connected to it’s 5V lines.)

---

**Q: How to identify Arduino Mini Pro 5v vs 3.3v**

**Tags: voltage-level (Prev Q) (Next Q)**

I have many Arduino Mini Pro, every from different seller. Few are 3.3v and most 5v.

I had to clean table for Xmas and now I do not have any idea how to identify 3.3v Arduinos.
They do not have any marks. I bought them on ebay.

I know 3.3v has 8mhz clock but only one my arduino has big crystal with 16.000-30.

**Tags:** [voltage-level](#) (Prev Q)

**User:** max

---

**Answer** by jon

The regulator should be marked K850(5.0V) or K833(3.3V).
The 16MHz resonator may be marked with “A1” or “A’N”
The 8MHz resonator may be marked with “80’0”
As others have indicated, you can apply up to 12V at the RAW pin, and measure the output of the regulator.

---

**Answer** by ignacio-vazquez-abrams

Plug it in and measure the voltage on pin 4 of the MCU.

---

**Answer** by xedret

Check the label of the voltage regulator, KB50 for 5V and KB33 for 3.3

---

**Tags:** [voltage-level](#) (Prev Q)
Aattiny

Questions

**Q: Is programming the ATTiny85 possible in 1.5?**

Tags: attiny (Prev Q) (Next Q)

I upgraded to Arduino 1.5.7 in hopes of using the command line interface for a larger project. However, the old package I was using to program the ATTiny85 chip with the Sparkfun Tiny AVR Programmer no longer works. I’m getting the following error:

```
Third-party platform.txt does not define compiler.path. Please report this to the third-party hardware maintainer.
```

Does a package exist that will allow me to program the chip in 1.5.7?

Tags: attiny (Prev Q) (Next Q)

User: batman

**Answer** by user2973

You can download a version of arduino-tiny for IDE 1.5.x from [here](#).

Tags: attiny (Prev Q) (Next Q)

---

**Q: Ideas for setting I2C device IDs**

Tags: attiny (Prev Q), i2c (Prev Q) (Next Q)

I am using I2C to communicate between a collection of ATtiny85s that are acting as controllers for various parts of a system and the “Main Brain” — a RaspberryPi. Most of the ATtiny85s only have a couple of free pins left (usually Pins 1 and 2 — the reset and the pin that can’t be used for PWM) so I can’t use something like a DIP switch to directly set device IDs. I’d rather not hardcode the device IDs when I push code to the tinys but I can’t think of a good, simple, and reliable way to do it otherwise. Anyone here have a
Put the address in EEPROM. It’s still “on the chip” but it can be programmed independent of the flash, and can be changed on the fly if deemed appropriate.
Arduino Yun

Questions

Q: How to get HTTPS on Arduino? [ ]

Tags: arduino-yun (Prev Q) (Next Q)

Put plainly: is there a way to get an HTTPS connection on the Arduino?

I have been looking in to it, and I have found it is impossible with the standard library and the Ethernet shield, but is there a custom library that can do it?

What about a coprocessor, i.e. like the WiFi shield has? Anyone know if the Arduino yún has ssl?

Tags: arduino-yun (Prev Q) (Next Q)

User: thedoctor [ ]

Answer [ ] by federico-fissore [ ]

MCUs mounted on Arduinos don’t have the horse power to handle https connections.

Yún does handle https on the linux side, with software like curl, wget or python. Your sketch can just delegate the task to the linux side.

While curl works fine, despite having python preinstalled, you need to manually install python-openssl, since it’s not available out of the box (due to disk space constraints)

Answer [ ] by jvarhol [ ]

I don’t think it is possible due to the size and complexity of the SSL Library, because the Arduino would most likely be under powered. That being said you could make the requests to a regular server and use some sort of PHP script to proxy the request to the HTTPS server. Not sure how well it would work though.

You may be interested into looking at this [ ].

Answer [ ] by hichris123 [ ]

As Federico Fissore says above, the Yún can only handle HTTPS (or SSL, whatever you want to call it) on the Linux side of the Yún.

The first way to do it is using Python with Python OpenSSL. The way you do this is by issuing the following commands via SSH or YunSerialTerminal:

```
opkg update
opkg install python-openssl
```

opkg update will make sure the package list is up-to-date, and then opkg install installs
Python OpenSSL. And then you can talk to the Arduino using Python. This page on the Arduino website should help you in using Python with Arduino.

You could also use curl with the -k option in your sketch. For example:

```plaintext
Process process;
process.runShellCommand("curl -k http://example.net");
while(p.running()); // this waits for the command to be done before continuing
```

Also, if you want to download a file, you could use wget. This would require you to upgrade wget by again issuing these commands via SSH or YunSerialTerminal:

```plaintext
opkg update
opkg upgrade wget
```

And then you can just do this in your sketch:

```plaintext
Process process;
process.runShellCommand("wget http://example.net");
while(p.running()); // this waits for the command to be done before continuing
```

**Tags:** arduino-yun (Prev Q) (Next Q)

---

**Q: Yún boot from SD card**

**Tags:** arduino-yun (Prev Q) (Next Q), bootloader (Prev Q) (Next Q), linux (Prev Q) (Next Q)

I have an Arduino Yún, and it only has 16mb flash onboard for the Linux. I was wondering of it is possible to put a bootloader on the flash that uses the sd card as a rootfs? I want to be able to do more projects on it than those that can fit in 16 MB flash.

**Tags:** arduino-yun (Prev Q) (Next Q), bootloader (Prev Q) (Next Q), linux (Prev Q) (Next Q)

**User:** thedoctor

**Answer** by thedoctor

And the Arduino team finally figured it out:

[http://blog.arduino.cc/2014/05/06/time-to-expand-your-yun-disk-space-and-install-node-js/]()

**Answer** by lokers

I am using Yun as well for my projects. I don’t boot from SD card, but I store a lot of Python code there and it works for me. I’ve also seen this article that might be interesting for you. I haven’t tried to do it on my Yun but I think it should work… This is example how to install node.js on the Yun (SD card), so in theory you could use the same approach for all additional software you want to run.

[http://linino.org/doku.php?id=wiki:nodejs]

**Tags:** arduino-yun (Prev Q) (Next Q), bootloader (Prev Q) (Next Q), linux (Prev Q) (Next Q)
**Q: How do I use Arduino and node.js?**

Tags: arduino-yun (Prev Q) (Next Q)

I am new to Arduino, however I have experience in web development, lately I have been using, meteor js and the mean stack for different projects. However, I am open to trying any language for development.

What I am trying to do is build a simple application to control the LED on the Arduino over the web.

I am working with another person, and the Arduino will be behind a firewall so I am not sure how to access it via the internet. I do have access to an Arduino Uun that I can use for testing at home which does have linio connected via bridge, which I don’t quite understand what that does.

I’ve also heard of this, but would like to stay away from third party builds as much as possible if I can do it myself.

What’s the common way to do this with an Arduino behind a firewall? How do I do this with a Arduino Yun? Is it easier with a Yun?

Tags: arduino-yun (Prev Q) (Next Q)

User: anders-kitson

---

**Answer by lokers**

I am working on similar project that uses internet to communicate with external device (mobile). I am reading data from sensors and passing them over the web with web sockets. I am using Arduino Yun for this project and Spacebrew (for web sockets communication).

It’s very interesting approach but quite a challenge in some cases. You will probably run into asynchronous/synchronous problem, which in my case lead me to running some Python scripts on Yun triggered from Processing code. I am currently running Spacebrew server on my VPN, so I can connect my Yun left at home and use my iphone from everywhere to get the readings.

It’s also worth mentioning Yun comes with pre-installed Temboo library, which is great starting point for web communication. The configuration is really easy and quick. You will also find official documentation very handy and easy to understand. It’s really great starting point for experimenting with web-like applications integrated with Arduino Yun.

**Spacebrew**

- [http://docs.spacebrew.cc/](http://docs.spacebrew.cc/)

  Spacebrew is an open, dynamically re-routable software toolkit for choreographing interactive spaces. Or, in other words, a simple way to connect interactive things to one another. Every element you hook up to the system can subscribe to, and publish
data feeds.

- [http://github.com/Spacebrew/pySpacebrew](http://github.com/Spacebrew/pySpacebrew)

  This repo contains the Spacebrew Library for Python along with documentation and example apps.

- [http://github.com/julioterra/yunSpacebrew](http://github.com/julioterra/yunSpacebrew)

  The Spacebrew Yun library features two main components - a set of python scripts that run on the linino, and a library that runs on the atmel chip. You find there information how to install both of these components so that you can connect to Spacebrew from an Arduino sketch.

Temboo

Yun has also build in Temboo library, it’s a great way to start experimenting, once you prove the concept you can start narrowing your environment and moving away if you don’t want to use 3rd parties software.

- [http://www.temboo.com/arduino](http://www.temboo.com/arduino)

  The Temboo Library makes it easy to connect to over 100 web-based resources and services with Processing. Currently you can find libraries for Android, Arduino, iOS, Java, Node.js, PHP, Processing, Python, REST API, Ruby, Twyla

Node.js

You can also run node.js on the Yun itself, here’s good article and some packages ready to install:

- [http://giorgiocefaro.com/blog/installing-node-js-on-arduino-yun](http://giorgiocefaro.com/blog/installing-node-js-on-arduino-yun)

- [https://github.com/giorrrgio/nodejs-linino](https://github.com/giorrrgio/nodejs-linino)

  This is an openwrt package for node. It probably only works on mips (not mipsel, or arm). It runs an outdated v8 as well.

What is Bridge library for Arduino Yun?

You also asked about Bridge library. As you know Yun has two processors on board, one is running your Processing code, the other one has Linux on it (Linino). Bridge basically simplifies communication between them so they can “talk” to eachother, more information here:


  (Taken from arduino.cc documentation:)

  The Arduino Yún has two processors on board. One is an ATmega32U4 like on the Arduino Leonardo. The other is an Atheros 9331, running Linux and the OpenWRT wireless stack, which enables the board to connect to WiFi and Ethernet networks. It
is possible to call programs or custom scripts on the Linux system through the Arduino to connect with various internet services.

The Bridge library simplifies communication between the ATmega32U4 and the AR9331. It inherits from Stream, and many of the methods should be familiar from Serial and other derivatives of Stream.

Bridge commands from the 32U4 are interpreted by Python on the AR9331. Its role is to execute programs on the GNU/Linux side when asked by Arduino, provide a shared storage space for sharing data like sensor readings between the Arduino and the Internet, and receiving commands from the Internet and passing them directly to the Arduino.

Bridge allows communication in both directions, acting as an interface to the the Linino command line.

Where to start?

I would start with Temboo if I were you, it’s a great learning curve and quite simple step to start. You will also learn and understand how the board works and how you can communicate with “external world”. Once you have proof of concept, start experimenting with different libraries and improving the approach. Perhaps you will find Spacebrew more handy later or even discover different solutions.

---

**Tags:** arduino-yun (Prev Q) (Next Q)

---

**Q:** How to disable Arduino Yun to work as Access Point and set it to work as needed?

**Tags:** arduino-yun (Prev Q) (Next Q)

I am using Arduino Yun and I would like to disable it to work as Access Point. That is, I don’t want that it is displayed in the list of available networks.

Then I would like to set and use it (separately) in the following scenarios:

a) Arduino Yun connected just over a given Wi-Fi network (Ethernet connection disabled).

b) Arduino Yun connected just over Ethernet (Wi-Fi connection disabled).

How can I make it?

**Note:** In my previous attempts I tried to customize and/or disable someway the Wi-Fi connection (through the LuCI panel) in order to reach what I am looking for but in all of these cases I had to restore my Arduino Yun from scratch because it did not work anymore.

**Tags:** arduino-yun (Prev Q) (Next Q)

**User:** user502052
Yun expects wifi to be working and, if it’s not, it will reboot itself after 60 seconds and return in access point mode.

In order to disable this check, edit file `/etc/rc.local` and turn `wifi-live-or-reset` into `#wifi-live-or-reset` (with a starting `#`).

Once done, use LuCI (the advanced configuration panel) to configure network interfaces.

**Q: How many total digital pin outs does the Arduino Yun have**

I have read the details on the products page for Arduino Yun but I have also read, on an Arduino forum, that there are only 18 digital pins because of Tx, and Rx. Is this true? I am not worried about the PWM, or analog capabilities of the pins, just purely digital.

Tags: arduino-yun

User: jimmy-s.

Answer by peter-r.-bloomfield

According to the official specification, it has 20 digital IO pins which can be used with `digitalRead()` and `digitalWrite()` etc. As is typical on Arduino boards though, pins 0 and 1 also act as serial Rx and Tx (respectively).

You can program it to use them for either purpose.

The information you read might be referring to problems with uploading though. If you have something physically connected to pin 0 and/or 1, then you may have to disconnect it while you do an upload, otherwise it can interfere with the communication between your computer and the microcontroller.

If you’re doing a lot of tinkering, then sometimes it’s simply easier to avoid using pins 0 and 1 altogether, meaning you’d only have 18 IO pins left.

Tags: arduino-yun
Safety

The safety tag is for the well-being of the people and objects around you, and yourself.

**Some general precautions for your Arduino:**

- Don’t under/over power your Arduino
- Don’t pull more than 40 mA on most boards from one digital pin
- Don’t apply more than 5V to any pin (except VIN)
- Don’t change a circuit while the Arduino is powered

**Some general precautions for you/the people around you:**

- Don’t apply mains to your Arduino.
- When powering high voltages with a relay, make sure to cover and seal any wires that might be connected to the high voltages.
- Arduino not to be used as a flotation device. Really, don’t get your Arduino or any other part of your setup wet.
Questions

Q: What I should be aware before I touch my Arduino?

Tags: safety (Prev Q) (Next Q)

Before you start to mess with a computer there are basic guidelines to be aware of, such as turn the power off and watch out for static electricity. What guidelines should I be aware of when I work with an Arduino board if I don’t want to harm neither myself nor the board. The three kind of activity that requires touching the Arduino and what I’m asking about:

- Setting up wire, component layout or breadboard.
- Debugging a running setup.
- Moving, mounting up the board somewhere.

Tags: safety (Prev Q) (Next Q)

User: totymedli

Answer by karl-bielefeldt

Turning the power off and taking proper static precautions also applies to Arduinos. Anorton’s tip about not resting it on a conductive surface is also useful. Some other things to keep in mind:

- Double check the polarities of any connections you make.
- Keep a consistent wiring color code. Use red for power and black for ground.
- Calculate the expected current through all components before you apply power. Check the datasheet to make sure it’s within safe limits.
- Calculate the heat dissipation for higher power components like power supplies and motor drivers.
- Connect and test one small part at a time, instead of in one big bang.
- Make sure the parts you buy expect the same voltage, or perform the appropriate conversion.

For keeping people safe:

- Know where the fire extinguisher is.
- Don’t put cords where people can trip on them.
- Use a proper stand to hold your soldering iron, and keep pets and small children away.
- If your project has a propeller or something physically dangerous, build failsafes and kill switches into the system.
- Be careful to charge any batteries properly, and not let them overheat.
- Be careful what you touch while troubleshooting. Arduinos usually don’t deal with very high voltages, but inductors and capacitors can build up higher charges than you expect, and hold it after power is removed.
One of the most common bad things I have seen is connecting LEDs directly to addition pins without a current limiting resistor.

Basically, in electronics there are a few main sources of problems. Too much current, too much voltage, overheating, and physical damage.

Too much current usually comes from shorting things, too much voltage (or reversed polarity) usually comes from not paying attention to what is being connected, overheating comes from too much current or not using a heatsink when you should be.

Basically, just think before you connect things and you’ll be fine. Learn some basic circuit theory if you haven’t already, get in the habit of knowing the ratings of the parts you want to use, and double check stuff before turning it on.

I like this list here of things not to do: http://ruggedcircuits.com/html/ancp01.html

Tags: safety (Prev Q) (Next Q)

Q: How cold or hot can my Arduino Uno get?

Tags: safety (Prev Q), arduino-uno-smd (Prev Q) (Next Q)

The Arduino Uno Page doesn’t say the temperatures that it can operate in. I’m thinking about placing it outdoors. How can I make sure my Arduino Uno is safe in weather that can reach -20 to 105 degrees Fahrenheit? (-26 to 40 degrees Celsius).

Tags: safety (Prev Q), arduino-uno-smd (Prev Q) (Next Q)

User: anonymous-penguin

Answer by anonymous-penguin

It’s not that big of deal. The ATmega 328p datasheet states this:

Temperature range: -40 to 85 degrees celsius.

The same goes for the USB chip on the Uno (ATmega 16u2 for UNO R3).

That’s inside your limits. It probably could go a bit colder than mentioned, but it’ll shorten the length of the board a little bit.

However, there are some things that may go wrong:

- EEPROM might not be stored as long in extreme temperatures. Keep this in mind if you’re storing critical data.
- The voltage regulator may not perform as well in hot conditions.
- The crystal oscillator may not produce exact values. However, I would imagine that a few hertz more or less wouldn’t affect a 16 MHz processor. The tolerance is actually a little less than 1%. You may have some issues with serial (baud rate not correct). I
would look into any communications like I2C also. (I don’t know exactly how the clock line works… it might be fine for I2C.)
- Resistors/capacitors may not produce the exact values. I’d imagine that the tolerance won’t be more than 8% on resistors: most resistors are rated at 5% for normal temperatures. It does depend on the manufacturer. Capacitors do have a bigger tolerance, but their main purpose is to “smooth” a signal.
- Extreme cooling/warming may cause minor expansion problems. (Note: it’s fine once in a while, but not on an hourly basis of dropping 30 degrees.)
- Other components (LCDs, etc.) should be kept in mind also when looking at the viability of keeping it outdoors.

So as long as all the other components not on the board will happily run in the temperatures that you need, you should be fine. Also, as with all engineering, values often have added “padding” built in. (i.e. 5% tolerance is often 3-4%, maximum 12V it can run on 12.5V, etc.)*

*What I mean by that is your Arduino won’t explode when it’s -41 degrees C. It’s not great for it, but most likely you should be fine as long is it isn’t a regular occurrence.

Answer by user851

Like everyone mentions, as long as you’re in the shade, hot temperature probably doesn’t matter too much as it’s within the limits of the components.

I’m more worried about condensation in the mornings. Water vapors will condense on electronics just like it does on grass. You could try electrical epoxy to cover the circuit. The Arduino doesn’t run very hot, so the epoxy doesn’t do much in terms of preventing it from cooling down. But epoxy does keep water vapor condensation from being a problem.

Tags: safety (Prev Q), arduino-uno-smd (Prev Q) (Next Q)
Arduino Nano

Questions

**Q: Is there a semi-permanent way to attach wires to the Nano?**

At least for prototyping, is there a semi-permanent way to attach wires to the Nano posts?

And yes, usually the Nano is hanging in free space by its wires (yes I know, quick and dirty, should be using something cleaner, even for prototypes). So I’m hoping for the best of both worlds, sturdy “enough” to stay attached, yet possible to remove and switch around. Having no luck finding such advice on the internet, perhaps searching for the wrong terms.

**EDIT:** I neglected one important point, each wire is attached individually just like the SparkFun wires with the female header as mentioned in @TheDoctor’s answer below.

**Tags:** [arduino-nano](#) (Prev Q) (Next Q)

**User:** chris-o

---

**Answer** by chris-stratton
A wire wrap tool can work well for this - the wrappings tend to be fairly secure if done right, though the wire itself can break if stressed.

Tags: arduino-nano (Prev Q) (Next Q)

**Q: Are there any reasons to pick Nano over Micro?**

Tags: arduino-nano (Prev Q) (Next Q)

Nano is from 2008, and micro from 2012. Micro seems better on every parameter, in particular in having a better (I think) microcontroller, ATMega32u4, over ATMega328P. However, all the shops I’ve checked out, both local and online, sell the Nano, and only some sell Micro. And Nano is always more expensive than Micro. E.g. at RS-Online 26GBP for Nano ([link](#)) and 16GBP for Micro ([link](#)).

Why do many shops only sell the inferior product?

Why is the older and less capable Nano more expensive?

Why would anyone buy the Nano over the Micro?

Tags: arduino-nano (Prev Q) (Next Q)

User: mads-skjern

**Answer** by geometrikal

The main difference between the ATmega32u4 and the ATmega328P is that the 32u4 has onboard USB. When ATmega328P are used in an Arduino they are often coupled with an FTDI USB to serial chip. The FTDI chip is about $5 so this may be where the extra cost comes in.

Having the USB chip separate is actually not a bad thing:

- The Atmega328P consumes less power than the ATmega32u4
- FTDI chips are common, so the drivers are on most computers, whereas the 32u4 requires an .inf file to be loaded on Windows.
- The FTDI chip uses the FTDI vendor and product ID and thus can be used in large-quantity commercial projects whereas the ATmega32u4 would require purchase of a vendor id and product id from the USB group. (My conclusion after reading around the net a bit, please correct if wrong).

**Answer** by anonymous

I’d buy the NANO and STEER CLEAR of the micro with that 32u4 chip. The good ol’ 328p chip also has a FTDI chip to handle USB to serial while the other chip handles it on its own. The problem is that the Micro’s bootloader ONLY allows trying to program it with USB and requires a special driver. With Linux, only a few select versions have the needed driver. If your Linux version doesn’t have the driver, you’re out of luck. In that case, take it back or donate it to a hackerspace. You will NEVER make it work.
I was at the arduino.cc site about this and tried everything only to get increasingly annoyed about that lemon of a card. You know it’s time to give up when you get flamed.

**Q: How does this Motor Circuit not ruin the Microcontroller**

I came across this instructable about using no external hardware but two pwm pins to control a motor in both directions. Usually you would use an H bridge. However, ever since I have had my Arduino I have heard that you are not supposed to connect a pin to another pin. So how does this circuit works. It looks like he sends a signal from one pin and in my experience it would fry the board. So basically, my question is this circuit ok with an Arduino Nano and if so how does it work? Below is a picture of the setup he gave:
This is a bad idea, but not because two pins seem to be connected directly. They’re not; there is a load (the motor) between the pins, and that motor will take care of different levels between the pins. High and low will make the motor turn one way, low and high the other way, and the same level on both pins will stop the motor.

Why is this bad? An I/O pin can only supply a limited current, and even for a small motor that will peak rather high. If you run the motor for some time this way you will damage the I/O ports. An H-bridge, like you mention, is the solution. Note that electric motors cause a lot of noise in the power supply, which may cause the microcontroller to reset all the time, so that your program doesn’t work. (The motor may seem to start/stop all the time.) You need to decouple the power supply with capacitors.

Tags: arduino-nano (Prev Q)
Assigning analogRead(x) (where x is a pin number) allows the Arduino to detect the approximate voltage on its analog pins. This is performed using a 10-bit ADC (Analog to Digital Converter), which takes approximately 100 microseconds.

It returns an integer value between 0 and 1023, which will usually represent the range +0v to +5v. However, the voltage range will be different if the board runs at a different voltage, or if the program has selected a different range by calling analogReference().
Questions

**Q: Analog Reference and Power Voltage**

On the Arduino website it states that the analog pins read voltages from 0 to 5 volts (default) and return a value from 0 to 1023. One can also set the range with analogReference(). However, this is probably assuming the Arduino is running at 5 volts. If one powers the board with, say 4 volts, will the analog reference still be 5 volts default, or changed to the input voltage of 4 volts?

**Tags: analogread (Prev Q)**

**User:** mc-δt

---

**Answer by ignacio-vazquez-abrams**

Your suspicions are correct. Calling `analogReference()` with a value of `DEFAULT` sets the ADC reference to the voltage at AV_{CC} which at least on the Uno R3 is tied directly to V_{CC} and hence has the same voltage as whatever you’re supplying there.

**Tags: analogread (Prev Q)**
**LCD**

**Questions**

**Q: What is the point of D0-D3 on LCD?**

Tags: lcd (Prev Q) (Next Q)

The [Arduino LiquidCrystal library](https://www.arduino.cc/en/Tutorial/LiquidCrystal) has 4 constructors with different arguments:

```
LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)
```

The following comments are stated for the constructor arguments:

> [...] d0, d1, d2, and d3 are optional; if omitted, the LCD will be controlled using only the four data lines (d4, d5, d6, d7).

What is the point of d0-d3 if they are not needed? Will they decrease the response time?

Tags: lcd (Prev Q) (Next Q)

User: friend-of-kim

---

**Answer by peter-r.-bloomfield**

That type of LCD has two main modes of operation: 4-bit mode which uses 4 data pins (d4-d7), and 8-bit mode which uses all 8 data pins (d0-d7).

4-bit mode has the advantage of requiring fewer output pins on your Arduino. However, it means your sketch needs to send each command/character as two separate batches of 4-bits (one after the other). This is handled for you automatically by the LiquidCrystal library, so you won’t need any extra complexity in your code. However, it does mean that your sketch requires roughly double the number of processor cycles to send anything to the LCD.

In contrast, 8-bit mode sends each command/character as a single batch of 8-bits. That simply means it technically runs a little faster.

With that said, a person looking at the display probably won’t see a significant speed difference most of the time, as it still goes very quickly. The main impact will be on the microcontroller (the Arduino in this case). If it can use the faster 8-bit mode then it can devote a little more processing time to other things. That can be particularly helpful for timing-critical applications.

---

**Answer by jfpoilpret**

The answer is on the same page as the quote you mentioned (I have highlighted the
important statement):

Creating a variable of type LiquidCrystal. **The display can be controlled using 4 or 8 data lines.** If the former, omit the pin numbers for d0 to d3 and leave those lines unconnected. The RW pin can be tied to ground instead of connected to a pin on the Arduino; if so, omit it from this function’s parameters.

It turns out that the Hitachi HD44780 LCD display can work in 4-bit or 8-bit modes.

**Q: Does a graphical driver chip exist for this project?**

I need some expert advice. I’m using an Arduino Leonardo, which I have got working with a Nokia 3310 monochrome screen. I wrote some code to draw any graphics to the screen.

What I want to know is, can I buy a chip from somewhere that will offload most of the CPU usage (for graphics from the Leonardo) that draws numbers and letters to the screen and basic graphics (different types of polygons and lines)? So I’m looking for a chip designed to do the graphical work and a data sheet reference for the chip.

**User:** karlie-kloss

**Answer** by majenko

I have no idea if there is a chip or not, but I doubt it.

But then, why do you need one?

You already know how to communicate with the screen, so why not off-load those functions on to another Arduino (make your own small ATmega328P based board to do it) and then communicate with that through whatever means you like from your Leonardo?

You don’t *need* a dedicated chip to do what you can quite easily program an Arduino to do.
Arduino DUE

Questions

Q: Arduino Time Clock Accuracy

Tags: arduino-due (Prev Q) (Next Q), time (Prev Q) (Next Q)

I’m currently trying to create an Arduino time clock by using the PJRC Time library (http://www.pjrc.com/teensy/td_libs_Time.html). I know that since most of the Arduino boards are running with a 16MHz clock and a single resonator, the time can become “out-of-sync” after a certain period of time.

However, I was wondering if anyone has an idea about the accuracy of the Time library when used on the Arduino DUE with a 84MHz clock. I’ve been testing it out and so far, the clock has been kept in sync for a few hours. Thanks!

Tags: arduino-due (Prev Q) (Next Q), time (Prev Q) (Next Q)

User: kk6fsl

Answer by jamolnng

The limitations of accuracy of the library depend on accuracy of the crystal. When they make, or cook, the crystal they can only make it to a certain degree accurate, also the environment of the crystal (temperature, humidity, etc.) play a role in the accuracy of it. Let’s say you have a crystal that is off by .5 second every hour, great for short term, but if you expand that over a year it is over 1 hour off by that time. If you want something to keep an accurate time over a long period I suggest a real time clock (they still have inaccuracies), a GPS module, or an internet connect to sync with.

For further information look at the wikipedia article on quartz crystals.

The use of a 84MHz crystal versus a 16MHz crystal will not necessarily improve the accuracy of the Arduino clock since the frequency of the crystal is more an indicator of processor speed than accuracy. The accuracy of the Arduino clock is primarily dependent on the accuracy of the crystal oscillator.

EDIT: I am no expert on crystal oscillators so if you see anything wrong here please let me know.

Answer by jippie

Your average system clock crystal will be off by several tens ppm (parts per million. They are great for stable and accurate timing of signals, but dramatic for keeping accurate time. Without special provisions the system crystal may be off by several seconds per day.

The solution is to use a proper Real Time Clock, driven by what is commonly known as a 32768Hz watch crystal. These crystals are easily a factor 10 better in accuracy. You can
either set up your own oscillator that interrupts the main processor and keep count in your Arduino sketch or you find a RTC breakout board.

Two random examples that pop up in Google with search terms “RTC breakout”:

- https://www.sparkfun.com/products/10160

---

**Q: Why does the Arduino Due have a native and USB programming port?**

**Tags:** arduino-due (Prev Q) (Next Q)

According to the schematics, the Arduino Due has two USB inputs:

1. Native
2. Programming

Why is this? And when would I use the different ports?

**Tags:** arduino-due (Prev Q)

**User:** poseid

---

**Answer** by sachleen

Either port can be used for programming, but the native USB port lets you do other things:

- It also enables the Due to emulate a USB mouse or keyboard to an attached computer. To use these features, see the [Mouse and Keyboard library reference pages](http://arduino.cc/en/Main/arduinoBoardDue).

- The Native USB port can also act as a USB host for connected peripherals such as mice, keyboards, and smartphones. To use these features, see the [USBHost reference pages](http://arduino.cc/en/Main/arduinoBoardDue).

So you can use the Due to interface with USB devices or connect it to your computer and have it act like a USB device.

**Tags:** arduino-due (Prev Q)
This tag should be used when writing code for the Arduino specifically using the Arduino IDE.

A sketch has the following structure:

```c
void setup()
{
  ...
}

void loop()
{
  ...
}
```

The `setup` function is called once when the sketch starts. The `loop` function then is called repeatedly until the board is either shut down or restarted.

A sketch is saved with the `.ino` extension.

If you are used to more traditional C programming, effectively what is done by the above is:

```c
int main ()
{
  init ();  // set up hardware, particularly timers used by delay()
  setup ();  // your own code, to be run once only
  while (true)
  {
    loop ();  // do this repeatedly
    return 0;  // this is never executed
  }
}
```

It’s a bit more complex than that, but that is the basic idea. Also the IDE lets you take control by writing your own `main()` function like this:

```c
int main ()
{
  ...  // do whatever you want here
  return 0;
}
```

If you do that, the “normal” `main` is ignored.
Questions

**Q: Is there any way to download a sketch from an Arduino?**

**Tags:** sketch (Prev Q) (Next Q)

I made a sketch, but then I lost it. However, I uploaded it to the Arduino before losing it. Is there any way I can get it back?

**Tags:** sketch (Prev Q) (Next Q)

**User:** the-guy-with-the-hat

---

**Answer** by sachleen

It should be possible as long as the security bit isn’t set. This question was asked on EE a while back.

**Is it possible to extract code from an arduino board?**

But you won’t get the Arduino code you wrote back. The code is compiled into assembly and you’ll have to convert that back to C yourself.

---

**Answer** by cybergibbons

This answer doesn’t directly answer the question, but still will result in the same end result.

The Arduino IDE uses temporary directories to store build files, including the original sketch as well as the HEX and intermediate files.

On a Mac, these are in `/var/folders` by default, and on a Windows machine they are in … `\Local Settings\Temp\` (which depending on the version of Windows could be in several places).

I’ve found that especially in Windows, these build files don’t get deleted when you close the sketch or IDE, so they may exist on your machine for far longer than you would think.

Also, by default, Crashplan, Backblaze and Time Machine back-up these locations, so even if they have been deleted, they may still be in a backup.

The advantage here is that you will recover C code rather than ASM.

---

**Answer** by zathura

I think there is another way to do this, without extracting the code from the Arduino board. Remember the date you create the sketch, and look for it in

C:\Users\User\AppData\Local\Temp\ (For example :

C:\Users\User\AppData\Local\Temp\untitled4390292704786567977.tmp\sketch_jan19a ).

You could find your unsaved code unless it was deleted by some “freeing space program”. Good luck!

---

**Tags:** sketch (Prev Q) (Next Q)
Q: What happens if there is a runtime error?

Tags: sketch (Prev Q)

What happens if there is a runtime error in a program? Will execution of the program just stop? Is there some way I get the Arduino to tell me what the error is?

Tags: sketch (Prev Q)

User: the-guy-with-the-hat

---

Answer by ricardo

First, let’s see a few examples of what can go wrong.

Uninitialized variables

```cpp
int status;
pinMode(13, OUTPUT);
digitalWrite(13, status);
```

Although the variable status is not explicitly initialized by the code above, the C++ compiler does that for you. It sets it as zero. So, even though it’s not explicit in the code, we know that the led won’t be lit. However, it’s not considered good practice to rely on that behaviour. Try and assign values to your variables as it makes the code clearer.

Memory overflow

```cpp
int array[10];
int v = array[100];
array[-100] = 10;
```

The first problem here is that you don’t know what will be assigned to v, but worse is that you don’t know what you messed up with the assignment to position -100 of array. The good news is that you only messed up with RAM and not with the Flash memory, thus your program is safe.

Jump to an illegal instruction

```cpp
void doSomething( void ) {
    for (int i = 0; i < 1000; i++);
}

void setup ()
{
    void (*funcPtr)( void );
    funcPtr = &doSomething;
    funcPtr(); // calls doSomething();
    funcPtr = NULL;
    funcPtr(); // undefined behavior
}
```
The first call to funcPtr() will actually be a call to doSomething(). Calls like the second one may lead to undefined behavior. In this case, it will continue execution from memory address 0, which seems to be the start of the sketch (I just tested and that’s equivalent to a soft boot - it doesn’t even go through the bootloader).

**Other bad things that may happen**

Well, you can run out of RAM, for example. What else. In any case, I think your program will keep running, probably not the way you intended it to.

**Kinds of Protection**

In computer systems, problems like these are usually dealt with at various levels:

1. By the compiler
2. By the programming language runtime (as in Java for example).
3. By the operating system or the processor (if your memory access a position outside the boundaries of the address space reserved to your program, the OS or the processor may have safety mechanisms to prevent that)

Arduinos only have limited protection of the compiler, and probably nothing else. The good news is that they aren’t multi-tasked, so the only program being affected is yours. In any case, any of those bugs will lead to erratic behavior.

**The Answers**

The assumptions are the all of the problems I stated above are runtime problems.

   What happens if there is a runtime error in a program?

The program will continue and what happens will depend on the side-effects of the runtime error. A call to the null function pointer will probably make the program jump to an unknown location.

   Will execution of the program just stop?

No, it will keep going as if nothing extraordinary happened, probably doing what you didn’t intend it to do. It may reset or act erratically. It may turn some inputs into outputs and burn a sensor or two (but that’s *highly unlikely*).

   Is there some way I get the Arduino to tell me what the error is?

I don’t think so. As I said earlier, the protection mechanisms aren’t there. There’s no runtime support from the language, no OS, no hardware checks for out-of-bounds memory access (the bootloader doesn’t count as either). You just have to be careful with your program and probably set your own safety nets.
The reason for the lack of protection is probably because Arduino controllers are too cheap, have too little memory, and should not run anything too important (yes, there seems to be a disclaimer by AVR somewhere for you not to use the MCUs normally used by Arduino in life support systems).

There are no runtime exceptions. There is only undefined behaviour.

Really, there are no exceptions at all. If you try to perform an invalid operation, it’s results will be unknown.

There is no runtime checking at all, except what you implement. Your program is running on bare-metal hardware. It’s the Desktop equivalent of running in ring-0 all the time, because the ATmega doesn’t have rings.

There is one mechanism that can get MCU from erratic state and it’s the watchdog timer. If you’re implementing some code that will repeatedly run in a loop, that will not run anytime longer than some fixed time, you can set this time as watchdog period and enable the timer.

Then, you have to repeatedly reset the timer in the loop. If your code freezes at some condition loop that will never end, then the watchdog will count to zero and eventually reset the MCU.

This way you are losing data, but if you run the AVR WDT in interrupt mode, you can store some data before resetting the MCU.

So the watchdog timer can guard your code from occasional unintended endles loops.

Documentation: AVR132: Using the Enhanced Watchdog Timer

Tags: sketch (Prev Q)
The arduino has a USB port which can connect to a computer. The cable used to make this connection has a standard USB connector at one end (that connects to the computer) and a “larger” connector on the opposite end (that connects to the arduino). See picture below:

What is the name of this “larger” connector?

They are both standard USB. The end which goes into the computer is type A (or USB-A), and the end which goes into the Arduino is type B (or USB-B). The USB-B connector is
also commonly used on devices such as printers and scanners. The distinction mainly exists so it’s obvious which device is the host (i.e. the computer) and which device is the client (i.e. the Arduino).

As a side note, other types of USB connector exist as well. You will probably be familiar with mini and micro USB connectors, which can be found on various devices such as digital cameras and mobile phones.

---

**Q: Difference between Arduino.cc and Arduino.org**

**Tags:** hardware (Prev Q) (Next Q)

Recently I have noticed that there are two arduino sites, arduino.cc and arduino.org. They both have the Arduino logo and both sell what seems to be official Arduino boards. Also, arduino.org came out with the Arduino Zero board first. What is the deal here? Has Arduino partnered with another site? Any ideas appreciated.

**Tags:** hardware (Prev Q) (Next Q)

**User:** null

**Answer** by ignacio-vazquez-abrams

The short of it is that there was a falling out within the Arduino people and now there are two groups laying claim to the “Arduino” name.

Arduino LLC runs arduino.cc. They are the steward of the Arduino IDE and libraries and own the “Arduino” trademark outside of Italy. Arduino SRL (fka Smart Projects SRL) is the company that assembles (assembled) the majority of Arduino boards for Arduino LLC, runs arduino.org, and owns the “Arduino” trademark inside Italy.

Arduino SRL recently decided that they are no longer beholden to Arduino LLC and has stopped paying licensing fees for using the Arduino name. Whether this is justified or not has not been fully tested in court yet.

**Answer** by mick-waites

This article can explain it better than I could paraphrase: [http://hackaday.com/2015/02/25/arduino-v-arduino/](http://hackaday.com/2015/02/25/arduino-v-arduino/)

Relevant excerpt:

Arduino LLC [arduino.cc] is the company founded by Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe and Gianluca Martino in 2009 and is the owner of the Arduino trademark and gave us the designs, software, and community support that’s gotten the Arduino where it is. The boards were manufactured by a spinoff company, Smart Projects Srl, founded by the same Gianluca Martino. So far, so good.

Things got ugly in November when Martino … renamed Smart Projects to Arduino
Srl and registered **arduino.org**

According to this [Italian Wired article](http://www.wired.it/it/2013/01/banzi-martino-arduino-split/), the cause of the split is that Banzi and the other three wanted to internationalize the brand and license production to other firms freely, while Martino … [wanted] to list on the stock market and keep all production strictly in the Italian factory.

---

**Tags:** [hardware](http://www.arduino.org) (Prev Q) (Next Q)

---

**Q: Re-use smartphone/tablets parts like a camera**

**Tags:** [hardware](http://www.arduino.org) (Prev Q)

For learning and hacking purposes, I plan to study and use parts from smartphones and tablets like Galaxy or iPhone. They are really efficient and pretty cheap…

Did anybody do this before? If yes, can anybody tell me what kind of connector is this one on the galaxy camera, [Galaxy Tab S 10.5 Front Facing Camera (Wi-Fi)](http://www.arduino.org), and whether it can be found for attaching it to an Arduino project?

Edit: Someone else asked a similar question here, [How to reuse quality camera from old / broken phone](http://www.arduino.org).

**Tags:** [hardware](http://www.arduino.org) (Prev Q)

**User:** charaf

**Answer** by jwpat7

The connector on the flex circuit from that camera was made by Hirose (it has an HRS logo on it), and might be in the BM10, BM14, or BM20 series. You can measure the pin spacing, stacking height, etc. to check which series it’s in. For example, the [Hirose BM14 series](http://www.arduino.org) of FPC to board connectors has pins at 0.4 mm Pitch and 0.8 stacking height. [Mouser.com](http://www.arduino.org) carries a 24-pin BM14 connector that might mate or match with the pictured connector. For other photos of Hirose FPC connectors, see [google images for hirose fpc](http://www.arduino.org).

While you probably can find a matching connector (via careful measurement and perhaps some trial and error) and build an interface, it may be difficult to use the camera once it’s hooked up. Most Arduino models have too little RAM to support image processing. A Yun might do it, a Tre, or some other micro entirely, eg Beagle or Raspberry.

**Answer** by cortices

*(summary of comment thread)*

I **strongly recommend** against trying to use any of these smartphone camera replacement parts. While they are cheap, this is because the signal they output is nigh impossible for you to use, even if you knew its specifications, which is unlikely given proprietary parts often have proprietary, non-public protocols. Even if you did have it, you’d likely have to
implement a linux kernel extension for Raspberry Pi yourself to support it. Far more work than it’s worth.

If you actually want to do image processing, there is a clear recommendation: the **Raspberry Pi camera interface**. RasPis have a special flex socket on the board for a camera, and you can buy cameras which work with the `raspivid` etc. commands included in raspbian with no extra setup. This allows you to use OpenCV or similar image processing libraries, or write your own, without having to worry about bitbanging obscure camera protocols.

No-name camera boards can be bought from eBay or similar sites for less than US$20, and they support 30fps 5 MP image/video capture to the Raspberry Pi natively and easily. Just search for “Raspberry Pi camera”.

*However…*

If you’re *really* sure you’re up to the challenge of hacking a proprietary camera module, you’ll want to do thorough research finding the exact protocol specifications for these camera chips first, and try to find ones that are the most common, for the highest chance of compatibility, and likelihood of others having made progress on using them. Good luck!

---

**Tags:** hardware [Prev Q]
The **bootloader** is a small piece of code loaded into the high part of program memory (flash memory). It is optionally invoked by the processor after a reset (depending on **BOOTRST** fuse setting).

Its purpose is to allow self-programming of the flash, generally by checking the serial port for specific “programming” sequences within the first second or so after a reset. If such a sequence is found the bootloader responds to the programmer (usually **avrdude** running on the host computer), and in conjunction with the programmer accepts and acts upon instructions to erase and reprogram memory, change the EEPROM if desired, and change the fuses if desired.

The use of the bootloader avoids the need for special hardware to upload new code. However to change the bootloader itself (something not normally necessary) a special programmer is needed which plugs into the ICSP (In Circuit Serial Programming) header of the Arduino.

Once the bootloader has finished uploading new code (if required) it jumps to address 0x0000 which is the start of the actual code, so that the newly-uploaded sketch can now start.

If no programming instructions are detected by the bootloader after a short delay, it jumps to address 0x0000 to execute the previously-loaded code.
Questions

**Q: How do I burn the bootloader?**

Tags: [bootloader](#) ([Prev Q](#))

I was instructed to “burn the bootloader” to my Arduino to fix an error I was getting. How can I do this? (And what exactly is the bootloader?)

Tags: [bootloader](#) ([Prev Q](#))

User: [the-guy-with-the-hat](#)

---

**Answer** by [the-guy-with-the-hat](#)

The bootloader is a small program that is (almost) always on an Arduino. It manages uploading and starting the sketches you make, and blinks the pin 13 LED whenever the board resets.

To burn the bootloader:

1. Obtain an [AVR ISP](#), [USBtinyISP](#), [ParallelProgrammer](#), or [another Arduino board](#). This will be your ISP.
2. Unless otherwise instructed, connect the ISP to the ICSP pins on your board.
3. Power your board with either a USB cable or an external power supply.
4. Make sure you selected the correct board that you are burning to at Tools ▶ Board ▶ in the IDE. Double-check this even if you could upload programs correctly; uploading doesn’t always require the right board.
5. Select the appropriate programmer at Tools ▶ Programmer ▶.
6. Click Tools ▶ Burn Bootloader, and wait. It shouldn’t take more than a minute, and often takes only a few seconds.

---

Sources:


---

**Answer** by [nick-gammon](#)

What is a bootloader?

And what exactly is the bootloader?

The bootloader is a small piece of code residing in the “high” part of program memory. Normally it is not erased when you upload new sketches. Its purpose it to be activated immediately after a reset (the fuses are configured to commence code at the bootloader address rather than address 0x0000). The bootloader will wait a short time (such as one
second) to see if “programming” data is arriving on the serial port.

- If so, it reads that data, and decodes it to reprogram the rest of the flash memory. Working in conjunction with the uploading program (normally avrdude) the new sketch is sent via the serial port. The bootloader erases pages of flash memory, and replaces them with the new code. Once done, it executes the new code.
- If no instructions arrive, the bootloader jumps to address 0x0000 to execute the existing sketch on the board.

How to replace the bootloader

An alternative method to program a bootloader is to use a sketch I wrote a while ago which runs on a:

- Uno (or similar, such as Duemilanove)
- Leonardo
- Mega2560

The “programming” board runs the sketch which has bootloaders for a number of boards hard-coded into it. You connect the programming board to the target board (the board wanting the bootloader) with 6 hook-up wires as shown.
That particular photo illustrates one Uno programming another one.

You run the sketch (operating via the Serial Monitor to give it commands). First it identifies the target chip:

```
Skip code block
Atmega chip programmer.
Written by Nick Gammon.
Entered programming mode OK.
Signature = 0x1E 0x95 0x0F
Processor = ATmega328P
Flash memory size = 32768 bytes.
LFuse = 0xFF
HFuse = 0xDE
EFuse = 0xFD
Lock byte = 0xCF
Bootloader address = 0x7E00
Bootloader length = 512 bytes.
Type 'L' to use Lilypad (8 MHz) loader, or 'U' for Uno (16 MHz) loader...
Type 'G' to program the chip with the bootloader...
```

The you press **G** to program it.

```
Skip code block
```
Erasing chip...
Writing bootloader...
Committing page starting at 0x7E00
Committing page starting at 0x7E00
Committing page starting at 0x7F00
Committing page starting at 0x7F00
Written.
Verifying...
No errors found.
Writing fuses_
LFuse = 0xFF
HFuse = 0xDE
EFuse = 0xFD
Lock byte = 0xCF
 Done.
Type 'C' when ready to continue with another chip...

For a Uno this takes one second. To program a Mega2560 takes 3 seconds.

Detailed instructions at Atmega bootloader programmer

Code can be downloaded from Github: nickgammon/arduino_sketches

Tags: bootloader (Prev Q)
Use this tag for questions regarding how to get the time on an Arduino, how to create or use clocks with an Arduino, or anything that time would be a substantial part of the problem/question.

Also consider using programming or other tags specific to your problem.
Questions

**Q: How do I get an accurate time?**

**Tags:** time (Prev Q) (Next Q)

I’ve made a clock using an Arduino, but the time seems to drift. I am aware of the rollover issue; the clock seems to drift by about 15 minutes over the course of a week.

I’m using a custom PCB with this resonator from Digi-key. The code reads the millis() function at the beginning of each loop, and works from that value.

My question is: How can I measure time with an Arduino, accurately enough to make a passable desk clock?

**Tags:** time (Prev Q) (Next Q)

**User:** john-walthour

---

**Answer** by edgar-bonet

You do not need an RTC to build a clock: the ATmega chip has all the hardware needed to perform the duties of the RTC itself. Here is how:

1. Get a 32768 Hz watch crystal: either buy it or disassemble an old clock. These crystals, specifically designed for time keeping, have an extremely small temperature drift. You would also need one of those if you wanted to use an RTC chip.

2. Configure the fuses of your ATmega to run off the 8 MHz RC oscillator. This will make your millis() function horribly inaccurate, and also free the XTAL1 and XTAL2 pins.

3. Connect the watch crystal to the TOSC1 and TOSC2 pins. These are the same pins as XTAL1 and XTAL2 (9 and 10 on the 328P). The different names are used to mean different functions.

4. Configure the Timer/Counter 2 for asynchronous operation, normal counting mode, prescaler set to 128, and enable the timer overflow interrupt.

Now you will get a TIMER2_OVF interrupt at a very steady rate of once per second. You only need to advance the clock display by one second in the ISR. In between the interrupts, you can put the MCU in very deep sleep (power-save sleep mode: nothing runs but Timer/Counter 2) and run for years on a couple of AA cells. Unless the display is power-hungry, obviously.

I did exactly this to build my 24-hour one handed wall clock. This link points now to the English translation of the original documentation in French.

**Quartz calibration**

If you do not calibrate your quartz, you can expect a significant drift, typically a few
seconds per week. The drift rate depends on the stray capacitance of the traces that connect the crystal to the MCU. In principle, it could be removed by adding some extra, finely tuned capacitance. It is worth noting that you would have the same drift problem with an RTC.

If you are satisfied with this kind of accuracy, then live with it and be happy. However, if you care to measure the drift, you will notice that it is very stable. You can then easily compensate for it in software, and achieve an accuracy of a few seconds per year.

The algorithm for correcting the drift is very simple. From the measured drift, you figure out the precise delay between the interrupts, which should be very close to 10^9 nanoseconds, then:

```c
#define ONE_SECOND 1000000000 // in nanoseconds
#define ONE_INTERRUPT 999993482 // for example

ISR(TIMER2_OVF_vect)
{
  static uint32_t unaccounted_time;
  unaccounted_time += ONE_INTERRUPT;
  while (unaccounted_time >= ONE_SECOND) {
    advance_display_by_one_second();
    unaccounted_time -= ONE_SECOND;
  }
}
```

In the above example, the quartz is slightly too fast, and the software compensates by “missing” a tick every few days. If the quartz was too slow, the same code would instead double-tick once every few days.

This kind of calibration could also be done for an RTC, but it would be significantly more complex because the RTC reports the time in a broken-down form that does not naturally lend itself to arithmetic operations.

Answer by mpflaga

The resonator you specified has a 0.3% stability, where the crystal or crystal oscillator (as mentioned by Ricardo) is 50ppm. Many times more stable. Not to even mention the temperature drift of resonator is horrible. Heating by sunlight will change it. Hence a resonator should not be used for keeping time over long periods.

Hence using either a crystal or crystal oscillator will get what you want. Either using it on the ATmega and set the fuses respectively or us one connected to an RTC.

Answer by jippie

If you don’t want to use extra hardware like an Real Time Clock (eg. DSDS1307), you can significantly improve timing accuracy by disabling all unused interrupts. By default Arduino sketches come with various interrupt routines enabled and often they are not used for actually your sketch. Quickest way to find out if you can do without it to try and disable them by issuing `noInterrupts();`.

Tags: time (Prev Q) (Next Q)
Q: Is it possible to find the time taken by millis?

Tags: time (Prev Q) (Next Q)

The function millis would be running in the span of 100+ microseconds or less. Is there a reliable way to go about measuring the time taken by a single millis call?

One approach that comes to mind is using micros, however, a call to micros will include the time taken by the function call of micros itself as well, so depending on how long micros takes, the measurement for millis may be off.

I need to find this as an application I am working on requires accurate time measurements for every step taken in the code, including millis.

Tags: time (Prev Q) (Next Q)

User: asheesh

Answer by jippie

Write a sketch that millis 1000 times, not by making a loop, but by copy and paste. Measure that and compare it to the actual expected time. Mind you that the results may vary with different versions of the IDE (and its compiler in particular).

Another option is to toggle an IO pin before and after the millis call, then measure the time for a very small value and a somewhat larger value. Compare the measured timings and calculate the overhead.

The most accurate way is to take a look at the disassembly listing, the generated code. But that is not for the faint of heart. You’ll have to carefully study the datasheet how long each instruction cycle takes.

Answer by 80hd

I second calling millis repeatedly and then comparing actual vs. expected.
There will be some minimal overhead, but it will diminish in significance the more times you call millis().

If you look at

```
C:\Program Files (x86)\Arduino\Arduino ERW 1.0.5\hardware\arduino\cores\arduino\wiring.c
```

You can see that millis() is very tiny at only 4 instructions (cli is simply # define cli() __asm__ __volatile__ ("cli" ::)) and a return.

I’d call it about 10 million times using a FOR loop which has a volatile as the conditional. The volatile keyword will prevent the compiler from attempting any optimization on the loop itself.

I don’t guarantee the following to be syntactically perfect..

```c
int temp1,temp2;
temp1=millis();
for (volatile unsigned int j=0;j<1000000;++j){
temp2=millis();}
Serial.print("Execution time = ");
```
my guess is that takes ~900ms or about 56us per call to millis. (I don’t have an arduino handy ATM.)

**Q: Delay function without using a timer**

Is there a delay function available that does not use millis() which itself makes use of a timer? Does not need to be hyper-exact.

**User:** thomas-s.

**Answer** by gerben

```c
delayMicroseconds(1000); // maximum delay is 65535 uS
```

or

```c
#include <util/delay.h>
_delay_ms(1);
```

**Tags:** time (Prev Q)
Software

Questions

Q: Assembly on the Arduino: IO registers

Tags: software (Prev Q)

Question: What registers do I use to access the IO on the Arduino Uno when I am using assembly for programming?

Background: I was interested in practicing my assembly language skills on the Arduino (I thought it would be easier to turn LEDs on/off than to worry about console IO). From doing a preliminary search, I have found that the AVR toolchain can be used to program the Arduino. Yet, I don’t quite know how to find the corresponding registers for the IO pins.

Tags: software (Prev Q)

User: apnorton

Answer by jippie

The datasheets for the AVR microcontrollers are reasonably good to read. Just Google for the exact controller on your Arduino (eg. ATmege328) and find the “Complete” datasheet on the atmel.com website. Don’t download the “Summary”, it doesn’t include a lot of information. Download the datasheet from atmel.com, that is the only location with the most recent documentation.

There are three registers used in basic IO (check the chapter called “I/O-Ports”):

- **DDRn**, Data Direction Register: This basically configures a pin for either input (0) or output (1).
- **PORTn**, Port n Data Register: When a pin is set as output, the related bit toggles the output pin high (1) or low (0). When configured as input, this enables a weak pull up resistor on the output pin.
- **PINn**, Port n Input Register: Use this register to read the current level on a pin that is configured as input.

Tags: software (Prev Q)
I2c

Skip to questions.

Wiki by user chris-o

Inter-Integrated Circuit (I2C) is a bus for low speed peripherals. It was invented by Philips, but many companies have created devices using the I2C protocol.

Use this tag for questions regarding how to use I2C with an Arduino, or for problems with I2C and your Arduino.

See the Wikipedia article for more info.
Questions

Q: How do I use I2C devices with Arduino?

Tags: i2c (Prev Q)

I have a few useful I2C components, such as a 16-bit port expander (MCP23017), which I’d like to use in various projects.

What do I need to do to make these work with Arduino? Will it work with any Arduino, or do I need a specific board or shield?

Tags: i2c (Prev Q)

User: peter-r.-bloomfield

Answer by butzke

Arduino have the Wire Library that communicate with I2C.

This library allows you to communicate with I2C / TWI devices. On the Arduino boards with the R3 layout (1.0 pinout), the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. The Arduino Due has two I2C / TWI interfaces SDA1 and SCL1 are near to the AREF pin and the additional one is on pins 20 and 21. As a reference the table below shows where TWI pins are located on various Arduino boards.

Board I2C / TWI pins

- Uno, Ethernet A4 (SDA), A5 (SCL)
- Mega2560 20 (SDA), 21 (SCL)
- Leonardo 2 (SDA), 3 (SCL)
- Due 20 (SDA), 21 (SCL), SDA1, SCL1

You can take a look at this, this examples.

Like @TheDoctor said:

The I2C data lines both need resistors to 3.3V. This keeps them from drifting if they are disabled. Andy resistor value should work, but it is recommended to use one between 10k ohm and 47k ohm.

Answer by klaus-warzecha

Additional shields are not needed.

You only need the Wire library and a couple of wires to connect the device to SDA and SCL.

The position of these pins is different on the various Arduinos, please have a look here.
The purpose of a button is to change the state of a connection when it is pushed. The connection change may be momentary (push-to-make or push-to-break), or it may be toggled on/off by consecutive pushes.

A microcontroller can monitor the connection, allowing the button to act as a simple input mechanism.

Use this tag when asking about using button components in Arduino projects. This can include issues such as connections and de-bouncing.
I’ve used enough products with embedded microcontrollers and unresponsive UIs to know that it can make or break a product. Even a slight delay or lag between the button presses and the display updating, or button presses being ignored or double counted, can be extremely frustrating.

What design patterns can be used to deal with user input (from buttons) and updating displays without causing these issues?

I currently use the following kind of pattern (this is simplified down to the bare minimum):

```cpp
#include <Bounce.h>
#define GREEN_LED 6 // Pin for green LED
#define BUTTON_PIN 15 // Pin for button - uses external pull down so active high
#define DISPLAY_REFRESH_INT 100 // How many MS between display updates

Bounce button = Bounce(BUTTON_PIN, 5);
// To signal between the button read and the display update
bool ledState = false;
// Used to keep track of last display update
long displayUpdate = 0;

void setup()
{
  pinMode(GREEN_LED, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
}

void loop()
{
  // Read the button status
  if (button.update())
  {
    if (button.risingEdge())
    {
      ledState = !ledState;
    }
  }

  // Update the display periodically
  if (millis() - displayUpdate > DISPLAY_REFRESH_INT)
  {
    displayUpdate = millis();
    digitalWrite(GREEN_LED, ledState);
  }
}
```

What other options are there? Is it ever worth using pin change interrupts to read buttons (if we are not considering battery life!).
The answer depends on exactly how the sketch is meant to respond to the user interaction.

**External interrupts**
If the interaction depends on responding very accurately to the rising or falling edge of an input event (as in your example), then an *external interrupt* may be the way to go, if possible.

This is particularly important if the input event might be shorter that the duration of a call to `loop()`. If that happens, consecutive attempts to poll the input could completely miss the state change. It’s quite rare for a user input event to be that short though, unless `loop()` is quite slow to execute.

External hardware to get help that though, e.g. by having an input which latches until it is read and reset by the microcontroller.

**Improving polling frequency**
External interrupts often aren’t possible (e.g. because the chip only supports a limited number), or they are simply overkill/unnecessary for a given application.

In those cases, it may still be possible to improve the timing precision when using polling. One option is to split up all the different parts of the main loop into separate functions. On each successive iteration of `loop()`, it will call the next function, and then wrap round to the first. In between each of those functions, it can poll the button. Here’s a simple example:

```c
Skip code block
unsigned int g_section = 0;
void setup() {} 
void pollButton()
{   //...
}
void runSection0()
{   //...
}
void runSection1()
{   //...
}
void runSection2()
{   //...
}
void loop()
{
   // Poll the button on every pass:
pollButton();
   // Execute the next section of the main code:
   switch (g_section)
   {
```
One important issue with this approach is that it’s very difficult to make each of those runSectionX() functions take the same execution time. That can result in the button being polled inconsistently from one iteration to the next. If it’s fast enough though, then it shouldn’t be too big of a problem.

**Display update**
Updating a display is unfortunately often quite slow, compared to many other operations. That means you need to be quite careful where you do it. Inside an Interrupt Service Routine often wouldn’t be appropriate, as that could introduce other timing issues in your sketch.

Realistically, you’d have to do it in the main loop. If you absolutely have to run it quickly though, then it can be included in the faster polling example I’ve given above — i.e. update the display on every iteration, the same as polling. You have to be careful to avoid partial updates though.

---

**Q: How to avoid that my arduino clicks eternally?**

I have accidentally implemented a program that imitates the click of the mouse.

The problem is that now I can’t delete the program because each time that I connect the USB, the computer starts clicking everything and it is not possible to upload a new program.

Can I solve this problem?

---

**Answer** by **majenko**

Hold the RESET button as you plug the Arduino in and keep it pressed all the time it’s plugged into your computer. Load Blink.ino into the IDE and compile it. Press the UPLOAD button in the IDE and move the mouse to somewhere “safe”. Release the RESET button.

Do those last three steps in rapid succession and you should be fine.
Q: Choosing wireless tech for lowest possible lag

Tags: button (Prev Q)

I am drafting a pub quiz project, where the Referee gives a signal and multiple players push their buttons. The first to push wins the right to give an answer.

I want all buttons (referee’s + X players) to be wireless. Since the best quiz players can click a button within 10ms of the signal, it is very important that there is a very, very little variance in ping between different buttons. I cannot have players shouting “my button is laggy!” On the other hand, I don’t want to use more expensive/complex tech than needed.

SO, what would be the optimal wireless tech to use? Please share your experience from similar projects (or just theoretical knowledge :) Here’s what I researched so far (correct me if I’m wrong):

1. Bluetooth 4 LE (as in RFduino) Pros: 3-6ms latency (advertised), low energy Cons: cost, not more than 7 buttons to the device
2. Wi-fi Pros: 2ms latency (I justed pinged my wi-fi router), dozens of buttons if need be Cons: cost, power-hungry
3. RF data tranceiver Pros: dunno, cost seems to be a bit smaller Cons: multiple buttons at the same frequency will probably create a lot of noise
4. Simplest “Radio remote” Pros: zero lag since it’s all electromechanical, no data Cons: multiple buttons at the same frequency won’t work

So, did I miss anything? I appreciate any guidance.

Tags: button (Prev Q)

User: sergey-snegirev

Answer by cortices

I’ve used nRF24L01+ chipset 2.4GHz wireless modules with Arduino before, and found them to be great, and super cheap (~$10 for 10 of them on ebay!). They have 3 modes of transmission: 250kbps, 1Mbps, and 2Mbps. The range decreases accordingly with higher bitrate, but the time spent sending a message does too. There are multiple Arduino libraries (e.g. RF24, Mirf, RadioHead) and tutorials for using the modules (see http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01). They also have built-in mesh networking modes which may or may not be of use.

Versions with an RP-SMA antenna connector and an advertised range of up to 1000m are also available for about $5 each. I would recommend using one of these for the referee’s button at least, depending on the range you need.

According to Charles Hallard of hallard.me, The non-amplified chips can get 30m line-of-sight range in 250kbps mode, and I have verified this in my own testing.

According to user sporadic on diychristmas.org, sending a 32 byte payload in 250kbps mode takes 1432μs from the start of Tx mode on the transmitter to receiving the
full message on receiver. This decreases to **444µs on 1Mbps mode** and **283µs on 2Mbps mode**.

Given these statistics, they fall well within 2ms latency for triggering, suitable for your application, and they use very little power. It’s up to you to work out the best balance of latency, range and expense.

---

**Tags:** button (Prev Q)
There are three types of memory on an Arduino:

- EEPROM
- RAM
- PROGMEM (Flash Memory - where your program goes)

On an Arduino Uno you have 1 KB of EEPROM, 2 KB of RAM, and 32 KB of program memory.

---

EEPROM

*Electrically Erasable Programming Read Only Memory*

This retains its value after power-off, so it can be used to retain configuration parameters, serial numbers, and similar things, even while the device is powered down.

EEPROM has a limited erase/rewrite life-span. It is typically quoted as 100,000 erase/write cycles. This is still a lot for things like serial numbers, which would change rarely. It can be read an indefinite number of times without wearing out.

---

RAM

*Random Access Memory*

This memory is lost at power-down, and has undefined values when the chip is powered up. It can be read from, and written to, an indefinite number of times. Your program variables are stored in RAM.

---

PROGMEM

*Program Memory*

Otherwise known as “flash memory”. This is where your program code is stored. It has a separate address space to RAM (variables) due to the Arduino AVR range being Harvard Architecture. PROGMEM has a limited erase/rewrite life-span. It is typically quoted as 10,000 erase/write cycles.

During normal program operation PROGMEM is only read from (to read program
instructions) - this can be done an unlimited number of times.
The program memory is normally changed when “uploading” a new program, either by using the bootloader, or an ICSP programmer.
See also progmem bootloader icsp
Questions

Q: Can I write to Flash Memory using PROGMEM?

Tags: memory-usage (Prev Q) (Next Q), progmem (Next Q)

On the documentation of Arduino, I quote:

http://playground.arduino.cc/Learning/Memory Note: Flash (PROGMEM) memory can only be populated at program burn time. You can’t change > the values in the flash after the program has started running.

And on the PROGMEM description:

http://arduino.cc/en/Reference/PROGMEM Store data in flash (program) memory instead of SRAM. There’s a description of the various types of memory available on an Arduino board.

The PROGMEM keyword is a variable modifier, it should be used only with the datatypes defined in pgmspace.h. It tells the compiler “put this information into flash memory”, instead of into SRAM, where it would normally go.

So can we or can’t we? Or it’s not the same thing?

Tags: memory-usage (Prev Q) (Next Q), progmem (Next Q)

User: zzarbi

Answer by peter-r.-bloomfield

The short answer is no: PROGMEM data is read-only.

Flash memory limitations

The first thing to understand is that Flash memory (where program space lives) is designed for long-term fixed storage. Reading from it is very fast and precise. However, generally speaking, you can’t modify it on a byte-by-byte basis (e.g. changing a specific variable). You usually have to erase and re-write it in large blocks. That makes it completely impractical for run-time manipulation, because you’d have to store a lot of redundant information somewhere else while you do the erase and write cycle.

What PROGMEM actually does

Any literal data specified in your code (such as strings and numbers) always reside in program space at first (i.e. in Flash). However, when your sketch actually wants to use that data at runtime, it normally has to allocate some space for it in SRAM and copy it over. That means you end up with two copies: the fixed original in Flash, and the temporary copy in SRAM.

When you use the PROGMEM modifier, you’re telling it not to make that second copy in SRAM. Instead, your sketch will simply access the original in Flash. That’s very useful if you only ever have to read the data, as it avoids the allocation and copy operations.
However, copying it to SRAM is essential if you want to modify the data. Aside from the Flash limitations I mentioned above, it’s also a question of code safety.

If you’re able to modify the data stored in program space, then it follows logically that you could also modify the code stored in program space. That would mean that a simple mistake (or in theory a malicious attack) could result in your sketch being partially or fully rewritten at run-time. This could have very unpredictable results, ranging from simply ceasing to work, through to damaging/destroying any connected equipment.

More information
You can learn more about the low-level PROGMEM stuff from here:


An older version of the same PROGMEM tutorial is available here:


---

**Tags:** memory-usage (Prev Q) (Next Q), progmem (Next Q)

---

**Q: How is stack memory used for fuctions and local variables?**

I wanted to save some values to the EEPROM and also wanted to free up SRAM by avoiding some variable declarations, but EEPROM memory is byte wise.

If I want to store an int value, I have to use some expressions repeatedly. I thought I would make some functions for those. But I am concerned that, if I create a function, it would still occupy SRAM memory, better I declare an int variable instead of using EEPROM.

How are the functions and the local variables stored in SRAM? Does it only store the address of the function pointer from the flash memory or all the variables and commands are stored on the stack?

**Tags:** memory-usage (Prev Q) (Next Q), eeprom (Prev Q) (Next Q), sram (Prev Q) (Next Q)

**User:** nafis

---

**Answer** by jrobert

Only the function’s data is stored on the stack; its code stays in flash. You can’t really reduce SRAM use by using EEPROM instead because, as you have seen, EEPROM isn’t addressable in the same way. The code to read and store EEPROM also needs to use some SRAM - probably as much SRAM as you were trying to save! EEPROM is also slow to write, and has a limited lifetime (in number of writes to each byte), both of which make it impractical to use for storing the kind of temporary data we usually put on the stack. It is
better suited to saving infrequently changed data, like the unique device configuration for mass-produced devices, or capturing infrequent errors for later analysis.

**Edited:** There is no stack for that function *until* the function has been called, so yes, that is when any of the function’s data get’s put there. What happens after the function returns is that its stack-frame (its reserved area of SRAM) is no longer reserved. It will eventually be re-used by another function call. [Here is a diagram](https://example.com) of a C stack in memory. When a stack frame is no longer useful, it is simply released and its memory becomes available to be re-used.

---

**Answer** by duncan-c

Local variables and function parameters are stored on the stack. However, that is not a reason not to use them. Computers are designed to work that way.

Stack memory is only in use while a function is active. As soon as the function returns, the memory is freed. Stack memory is a GOOD Thing.

You don’t want to use recursive functions with lots of levels of recursion, or allocate lots of large structures on the stack. Normal use is fine however.

The 6502 stack is only 256 bytes, but the Apple II works just fine.

---

**Answer** by user2973

The AVR (the microcontroller family traditionally used on Arduino boards) is a [Harvard Architecture](https://example.com), meaning that executable code and variables are in two separate memories - in this case flash and SRAM. The executable code never leaves flash memory.

When you call a function the return address is usually pushed to the stack - the exception is when the function call happens at the end of the calling function. In this case the return address of the function that called the calling function will be used instead - it is already on the stack.

Whether any other data is put on the stack depends on the register pressure in the calling function and in the called function. Registers are the working area of the CPU, the AVR has 32 1-byte registers. The registers can be accessed directly by CPU instructions, whereas data in SRAM will first have to be stored in registers. Only if arguments or local variable are too big or too many to fit in registers will they be put on the stack. However, structures are always stored on the stack.

You can read the details of how the stack is used by the GCC compiler on the AVR platform here: [https://gcc.gnu.org/wiki/avr-gcc#Frame_Layout](https://gcc.gnu.org/wiki/avr-gcc#Frame_Layout)

Read the sections “Frame Layout” and “Calling Convention”.

---

**Tags:** memory-usage (Prev Q) (Next Q), eeprom (Prev Q) (Next Q), sram (Prev Q) (Next Q)

---

**Q:** **PROGMEM:** do I have to copy data from flash to RAM for reading?
I have got some difficulties understanding the memory management.

Arduino documentation says, it is possible to keep constants like strings or whatever I don’t want to change during runtime in program memory. I think of it as embedded somewhere in the code segment, which must be fairly possible inside a von-Neumann-architecture. I want to make use from that to make my UI menu on a LCD possible.

But I’m bewildered by those instructions to just read and print data from program memory:

```cpp
strcpy_P(buffer, (char*)pgm_read_word(&(string_table[i]))); // Necessary casts and dereferencing, just
Serial.println(buffer);
```

Why on earth do I have to copy the damn content to RAM before accessing it? And if this is true, what happens to all the code then? Is it also loaded to RAM before execution? How is the code (32kiB) handled then with only 2kiB RAM? Where are those little goblins carrying floppy disks?

And even more interesting: What happens to literal constants like in this expression:

```cpp
a = 5*(10+7)
```

are 5, 10 and 7 really copied to RAM before loading them into registers? I just can’t believe that.

---

**Answer**

AVR is a modified Harvard architecture family, so code is stored in flash only, whereas data exists primarily in RAM when being manipulated.

With that in mind, let’s address your questions.

Why on earth do I have to copy the damn content to RAM before accessing it?

You don’t need to per se, but by default code assumes that data is in RAM unless the code is modified to specifically look in flash for it (such as with `strcpy_P()`).

And if this is true, what happens to all the code then? Is it also loaded to RAM before execution?

Nope. Harvard architecture. See the Wikipedia page for the full details.

How is the code (32kiB) handled then with only 2kiB RAM?

The preamble generated by the compiler copies the data that should be modifiable/modified into SRAM before running the actual program.

Where are those little goblins carrying floppy disks?
Dunno. But if you happen to see them then there’s nothing I can do to help.

… are 5, 10 and 7 really copied to RAM before loading them into registers?

Nah. The compiler evaluates the expression at compile time. Whatever else happens depends on the lines of code around it.

---

**Answer** by **geometrikal**

This is how `Print::print` prints from program memory in the Arduino library:

```cpp
size_t Print::print(const __FlashStringHelper *ifsh)
{
  const char PROGMEM *p = (const char PROGMEM *)ifsh;
  size_t n = 0;
  while (1) {
    unsigned char c = pgm_read_byte(p);
    if (c == 0) break;
    n += write(c);
  }
  return n;
}
```

`__FlashStringHelper*` is an empty class that allows for overloaded functions like `print` to differentiate a pointer to program memory from one to normal memory, as both are seen as `const char*` by the compiler (see [http://stackoverflow.com/questions/16597437/arduino-f-what-does-it-actually-do](http://stackoverflow.com/questions/16597437/arduino-f-what-does-it-actually-do)).

So you could overload the `print` function for your LCD display so that it takes a `__FlashStringHelper*` argument, let’s call it `LCD::print`, and then use `lcd.print(F("this is a string in progmem"));` to call it. `F()` is a macro that ensures the string is in program memory.

To predefine the string (to be compatible with built-in Arduino `print`) I have used:

```cpp
const char firmware_version_s[] PROGMEM = {"1.0.2"};
__FlashStringHelper* firmware_version = (__FlashStringHelper*) firmware_version_s;
...
Serial.println(firmware_version);
```

I think an alternative would be something like

```cpp
size_t LCD::print_from_flash(const char *pgms)
{
  const char PROGMEM *p = (const char PROGMEM *) pgms;
  size_t n = 0;
  while (1) {
    unsigned char c = pgm_read_byte(p);
    if (c == 0) break;
    n += write(c);
  }
  return n;
}
```

which would avoid the `__FlashStringHelper` cast.

---

**Answer** by **nick-gammon**

Arduino documentation says, it is possible to keep constants like strings or whatever
I don’t want to change during runtime in program memory.

**All** constants are initially in program memory. Where else would they be when the power is off?

I think of it as embedded somewhere in the code segment, which must be fairly possible inside a von-Neumann-architecture.


Why on earth do I have to copy the damn content to RAM before accessing it?

You don’t. In fact there is a hardware instruction (LPM - Load Program Memory) which moves data directly from program memory into a register.

I have an example of this technique in [Arduino Uno output to VGA monitor](https://www.arduino.cc/en/Tutorial/VideoOoutputToVGA). In that code there is a bitmapped font stored in program memory. It is read from that on-the-fly and copied to the output like this:

```c
// blit pixel data to screen
while (i--)
    UDR0 = pgm_read_byte (linePtr + (* messagePtr++));
```

A disassembly of those lines shows (in part):

```
f1a: e4 91  lpm  r30, Z+
f1c: e0 93 c6 00  sts  0x00C6, r30
```

You can see that a byte of program memory was copied into R30, and then immediately stored into the USART register UDR0. No RAM involved.

However there is a complexity. For normal strings the compiler expects to find data in RAM not PROGMEM. They are different address spaces, and therefore 0x200 in RAM is something different from 0x200 in PROGMEM. Thus the compiler goes to the trouble of copying constants (like strings) into RAM at program startup, so it doesn’t have to worry about knowing the difference later on.

How is the code (32kiB) handled then with only 2kiB RAM?

Good question. You won’t get away with having more than 2 KB of constant strings, because there won’t be room to copy them all.

That’s why people who are writing things like menus and other wordy stuff, take extra steps to give the strings the PROGMEM attribute, which disables them being copied into RAM.

But I’m bewildered by those instructions to just read and print data from program memory:

If you add the PROGMEM attribute you have to take steps to let the compiler know that these strings are in a different address space. Making a complete (temporary) copy is one way. Or just print directly from PROGMEM, a byte at a time. An example of that is:
void printProgStr (const char * str)
{
    char c;
    if (!str)
        return;
    while ((c = pgm_read_byte(str++))
    
        Serial.print (c);
} // end of printProgStr

If you pass this function a pointer to a string in PROGMEM, it does the “special read” (pgm_read_byte) to pull the data from PROGMEM rather than RAM, and the prints it. Note that this takes one additional clock cycle, per byte.

And even more interesting: What happens to literal constants like in this expression a = 5*(10+7) are 5, 10 and 7 really copied to RAM before loading them into registers? I just can’t believe that.

No, because they don’t have to be. That would compile into a “load literal into register” instruction. That instruction is already in PROGMEM, so the literal is now dealt with. No need to copy it to RAM and then read it back.

I have a lengthy description of these things on the page Putting constant data into program memory (PROGMEM). That has example code for setting up strings, and arrays of strings, reasonably easily.

It also mentions the F() macro which is an easy way of simply printing from PROGMEM:

Serial.println (F("Hello, world"));

A bit of preprocessor complexity lets that compile into a helper function which pulls the bytes in the string from PROGMEM a byte at a time. No intermediate use of RAM is required.

It is easy enough to use that technique for things other than Serial (eg. your LCD) by deriving the LCD printing from the Print class.

As an example, in one of the LCD libraries I wrote, I did exactly that:

class I2C_graphical_LCD_display : public Print
{
    ...
    size_t write (uint8_t c);
};

The key point here is to derive from Print, and override the “write” function. Now your overridden function does whatever it needs to output a character. Since it is derived from Print you can now use the F() macro. eg.

lcd.println (F("Hello, world"));

Tags: memory-usage (Prev Q), flash (Prev Q) (Next Q), progmem (Prev Q) (Next Q)
Linux

Linux is an open source operating system (OS) that is lightweight and runs on many systems. It also runs on the Arduino Yún. There are different “flavors” or “distros” (distributions and changes made by different groups) of Linux such as Ubuntu, Debian, and Fedora. All versions of Linux use the Linux Kernel as the base of the OS. The kernel was released in late 1991.

Screenshot of Ubuntu, a popular distro.

The Yún runs a port of Linux called Open WRT that is designed for routers and other networking appliances.
Questions

**Q: How to differentiate between 2 Micros on Linux**

Tags: linux (Prev Q)

I have 2 Arduino Micro based devices that I want to use on my Ubuntu Linux machine. I need them to have unique serial ports. Unfortunately, both have the same serial number which makes identification through udev rules difficult (if not impossible).

This is the serial number that I get for both:

```
$ udevadm info -a -p $(udevadm info -q path ~/dev/ttyACM0) | grep -i serial
ATTRS{serial}=="0000:00:14.0"
```

Is there any way to manually set the serial number on an Arduino Micro? Or is there any other unique identifying properties I can use for udev rules?

Typically, I do something like this in my `/etc/udev/rules.d/99-usb-serial.rules` file:

```
SUBSYSTEM=="tty", ATTRS{product}=="Arduino Uno", ATTRS{serial}=="64936333936351911191", SYMLINK+="MyDevice"
```

Side note: I also have a Due, that has the same serial number as the Micros, but since ATTRS{product} is different, I can differentiate it.

Tags: linux (Prev Q)

User: ben

---

**Answer** by majenko

The USB VID, PID and Serial Number are all programmed into the chip when you upload your sketch. The VID and PID are tied to the board, and the serial number is always the same regardless of which board.

So no matter which board you program it will get the same details as any other board of the same kind.

There’s no option to change the serial number (there is, IIRC, actually no serial number defined) but the VID and PID are provided as a pair of macros on the compilation command line: `USB_VID` and `USB_PID`.

So you could (cheekily) create a new board definition (based on the existing Micro one) that is “The other Micro” and provide a different (unofficial) PID to that board. You can then differentiate the two boards in udev using the VID/PID combination.

Of course, you have to then ensure that you have the right board selected when you program them, and the act of entering the bootloader on the “other” board would revert back to the old PID which is programmed into the bootloader, but I’m sure you can cope with that.

Tags: linux (Prev Q)
A clone is a legal, Arduino-like board developed by a third party using the original schematics for the board (with a few minor adjustments). If significant changes are made, it is called a derivative.

All Arduino clones follow the [naming guidelines](#):

While unofficial products should not have “Arduino” in their name, it’s okay to describe your product in relation to the Arduino project and platform. Here are a few guidelines that explain which uses we consider reasonable. Not okay:

- Arduino Xxxxxx
- Xxxxxx Arduino
- Arduino Compatible Xxxxxx - use “Xxxxxx (Arduino-Compatible)” instead
Okay:

- Xxxxxx for Arduino - products that work with official Arduino boards (e.g. shields or kits)
- Xxxxxx (Arduino-Compatible) - variations and clones which are software and hardware compatible

Note that while we don’t attempt to restrict uses of the “duino” suffix, its use causes the Italians on the team to cringe (apparently it sounds terrible); you might want to avoid it. (It’s also trademarked by a Hungarian company.)
Questions

Q: Compatibility between Arduino and Arduino clones

Tags: clones (Prev Q) (Next Q)

In a project I am using a Seeeduino v3.0 as an Arduino clone. On the seeeduino website it says that

   It is based on the Duemilanove schematic, 100% compatible to its existing program, shield and IDEs.

In fact I never had any problems with shields or the IDE.

My question is now: Is the seeeduino and other clones really 100% compatible? The seeeduino board was considerable cheaper then an arduino.

Tags: clones (Prev Q) (Next Q)

User: a.schulz

Answer by anindo-ghosh

Over the years I have used original Arduino boards as well as a multitude of clones of varying provenance, from the excellent InduinoX that is listed on the Arduino site as an official compatible board, to the very well-made Chinese Meduino Nano Enhancement (3.3 / 5 Volt switchable), to dirt-cheap clones from eBay that are sometimes better constructed than the official boards.

There is nothing unique or sophisticated about the Arduino board: It’s basically just the microcontroller (atMEGA328, atMEGA1280, SAM3X8E or others) and its minimal support hardware. Not much that can go wrong with that little, really.

The two common elements that ensure compatibility across these boards are:

1. The Arduino bootloader
2. Support for the board within the official Arduino software distribution

The bootloader can be replaced with a better or more up-to-date version if one is available for the microcontroller on your board.

The support files can either be found online, or minor edits to the nearest similar board’s files can provide the requisite support for the Arduino environment.

In short: Yes, clones typically work just as well as the original Arduino, and clones with enhanced features often work far better while retaining compatibility (e.g. InduinoX)

Answer by jippie

If you closely look at the Arduino circuit diagram, You’ll notice that it is little more than power supply, a USB-serial interface and the microcontroller itself.
The ‘hard’ part for compatibility is getting the board layout right (which connector goes where on the circuit board), and to load a compatible bootloader in the chip (which is free/open software, so why bother making your own?).

I’ve used several el-cheapo Arduino clones and apart from apparent physical construction (one feels a bit flimsy), they all work just like the ‘real thing’. I guess the major difference between original and clone is the quality of the board, the used components and quality of assembly.

Answer by x4mer

Check twice!

Most clones are exact clones with copies of the flaws that original Arduino board had. Like for example Arduino Nano not having proper RESET and TEST pin connection of FT232RL USB-serial bridge. This flaw may lead to FT232RL not responding to PC and was eventually fixed in later revisions of Arduino Nano board. Some clones were not updated.

Another “not exactly Arduino clones” example are numerous boards with CP2102 USB to serial bridge instead of FT232RL. This may pass unnoticed, but may give frustration when one clone board connects and works with your PC, but another generates system complains on missing drivers.

So again - be prepared!

Tags: clones (Prev Q) (Next Q)

Q: What’s the difference between a clone and a counterfeit Arduino?

Tags: clones (Prev Q)

I’ve been hearing a lot about clones and counterfeits. What are the differences between the two?

Tags: clones (Prev Q)

User: anonymous-penguin

Answer by anonymous-penguin

The difference is simple:

Clones don’t say “Arduino,” where counterfeits do

If I were to create 200 boards all labeled “Arduino UNO R3,” it would be a counterfeit. If I were to call them “Happyduino,” it would be a clone.

Note: You can still write Happyduino (Arduino Compatible) on your product. Here’s a quote from the Arduino FAQ section
Not okay:

- Arduino Xxxxxx
- Xxxxxx Arduino
- Arduino Compatible Xxxxxx - use “Xxxxxx (Arduino-Compatible)” instead

Okay:

- Xxxxxx for Arduino - products that work with official Arduino boards (e.g. shields or kits)
- Xxxxxx (Arduino-Compatible) - variations and clones which are software and hardware compatible

Note that while we don’t attempt to restrict uses of the “duino” suffix, its use causes the Italians on the team to cringe (apparently it sounds terrible); you might want to avoid it. (It’s also trademarked by a Hungarian company.)

Answer by federico-fissore

A **clone** is an exact or almost exact replica of an original Arduino board, with a different branding.

A **derivative** is a board based or inspired by Arduino boards, with some specific addition or modification (different layouts, built-in sensors…)

A **counterfeit** is a clone of an Arduino board, with the same branding of an Arduino board.

More info on an [article](http://www.arduino.cc) on the Arduino blog.

**Tags:** clones (Prev Q)

Variables

A variable holds data in a sketch/program. For example, a program might store the result of an analog-to-digital conversion in an integer variable:

```c
int val = analogRead(2);
```

In this case, a variable named `val` is being declared with type `int` (integer). It is being initialised to contain the result of a call to `analogRead()`.

Other common variable types can include `float`, `char`, `long`, and pointer, among others.

Variables are technically distinct from objects (instances of a class/struct), but they are largely handled in the same way.

Use this tag for Arduino programming issues which specifically relate to the use of variables. This is likely to be quite rare, so please consider that there may be a more suitable tag, such as `programming`. 
**Questions**

**Q: Assuring an unsigned long int?**

**Tags:** variables (Prev Q)

Basic question: How far do I have to go to assure that integer math is done correctly? For example, this is probably overdone:

```c
unsigned long burnTime = 0UL;
unsigned long curBurnTime = 0UL;
// Do some stuff that sets the above variables to millis() at various times
// Later on...
unsigned long adjustedBurnTime = (unsigned long)((burnTime + curBurnTime) / 1000UL);
```

Would the math be done correctly if I went to a more minimal last statement (since all the elements of the equation are unsigned longs)? Like this:

```c
unsigned long adjustedBurnTime = (burnTime + curBurnTime) / 1000UL;
```

Or even:

```c
unsigned long adjustedBurnTime = (burnTime + curBurnTime) / 1000;
```

**Tags:** variables (Prev Q)

**User:** bluesmoke

---

**Answer** by jrobert

Any good C manual will tell you the result type for each basic math operation on given types. But briefly, the basic math operators return a result as wide as the widest operand, with the narrower operand being widened to match the wider one, if they differ. The same with assignment: assigning a result of a given type to a variable of the same type will not involve any conversion.

tl;dr: No, you’re not missing anything. You don’t need any casts in your example.

PS: One thing you do need to be aware of is whether any of your intermediate results will overflow the arithmetic type you’re using. For example, the final result may be not exceed an unsigned long but the expressions (burnTime + curBurnTime) must also not exceed it. This is a greater concern with smaller data-types, of course.

PPS: Good on you for tackling fixed point arithmetic rather than just reaching for the floating-point library!

**Tags:** variables (Prev Q)
Flash

Questions

Q: What can I do if I run out of Flash memory or SRAM?

Tags: flash (Prev Q)

According to the Arduino documentation, the ATmega328 has 32KB of Flash memory for the bootloader + uploaded sketch, and only 2KB SRAM for runtime data. The ATmega2560 has quite a bit more, totalling 256KB and 8KB respectively.

In either case, those limits seem rather small, especially when compared to similarly sized consumer devices, such as smartphones.

What can you do if you run out? E.g. if your sketch is too big, or you need to process a lot of data (such as strings) at runtime? Is there any way to expand the Flash or SRAM?

Tags: flash (Prev Q)

User: peter-r.-bloomfield

Answer by peter-r.-bloomfield

Optimisation

Low-level programming for embedded systems is quite different from programming for general purpose devices, such as computers and cell phones. Efficiency (in terms of speed and space) is far more important because resources are at a premium. That means the very first thing to do if you run out of space is to look at what parts of your code you can optimise.

In terms of reducing program space (Flash) usage, the code size can be quite difficult to optimise if you’re inexperienced, or if you’re more used to programming for desktop computers which don’t tend to need that skill. Unfortunately, there’s no ‘magic bullet’ approach which will work for all situations, although it helps if you consider seriously what your sketch really needs to have. If a feature isn’t needed, take it out.

Sometimes it’s also helpful to identify where multiple parts of your code are the same (or very similar). You may be able to condense them into reusable functions which can be called from multiple places. However, be aware that sometimes trying to make code too reusable actually ends up making it more verbose. It’s a tricky balance to strike that tends to come with practice. Spending some time looking at how code changes affect the compiler output can help.

Runtime data (SRAM) optimisation tends to be a bit easier when you’re used to it. A very common pitfall for beginner programmers is using too much global data. Anything declared at global scope will exist for the entire lifetime of the sketch, and that isn’t always necessary. If a variable is only used inside one function, and it doesn’t need to
persist between calls, then make it a local variable. If a value needs to be shared between functions, consider if you can pass it as a parameter instead of making it global. That way you’ll only use SRAM for those variables when you actually need it.

Another killer for SRAM usage is text processing (e.g. using the `String` class). Generally speaking, you should avoid doing String operations if possible. They are massive memory hogs. For example, if you’re outputting lots of text to serial, use multiple calls to `Serial.print()` instead of using string concatenation. Also try to reduce the number of string literals in your code if possible.

Avoid recursion if possible as well. Each time a recursive call is made, it takes the stack a level deeper. Refactor your recursive functions to be iterative instead.

**Use EEPROM**

EEPROM is used for long-term storage of things that only change occasionally. If you need to use large lists or look-up tables of fixed data, then consider storing it in EEPROM in advance, and only pulling out what you need when necessary.

Obviously EEPROM is quite limited in size and speed though, and has a limited number of write cycles. It’s not a great solution to data limitations, but it might be enough to ease the burden on Flash or SRAM. It’s also quite possible to interface with similar external storage, such as an SD card.

**Expansion**

If you’ve exhausted all other options, then expansion may be a possibility. Unfortunately, expanding Flash memory to increase program space isn’t possible. However, it is possible to expand SRAM. This means you may be able to refactor your sketch to reduce code size at the expense of increasing data size.

Getting more SRAM is actually fairly straightforward. One option is to use one or more 23K256 chips. They are accessed via SPI, and there is the [SpiRAM library](https://github.com/julienvincent/SpiRAM) to help you use them. Just beware that they operate at 3.3V **not** 5V!

If you’re using the Mega, you could alternatively get SRAM expansion shields from [Lagrangian Point](https://lagrangian-point.com/) or [Rugged Circuits](https://ruggedcircuits.com/).

---

**Answer** by sachleen

When you upload your code to your Arduino, say an Uno for example, it’ll tell you how many bytes it uses up out of the 32K available. That’s how much flash memory you have (think computer hard disk). While your program is running, it’s using what’s called SRAM, and there is much less of that available.

Sometimes you’ll notice your program behaving oddly at a point that you haven’t even touched in a while. It could be that your most recent changes cause it to run out of memory (SRAM). Here are a few tips on how to free up some SRAM.

**Storing strings in Flash instead of SRAM.**

One of the most common things I’ve seen is the chip running out of memory because there are too many long strings.
Use the \texttt{F()} function when using strings so they are stored in Flash instead of SRAM, since you have much more of that available.

\begin{verbatim}
Serial.println(F("This string will be stored in flash memory"));
\end{verbatim}

\section*{Use the right data types}

You can save a byte by switching from an \texttt{int} (2 bytes) to a \texttt{byte} (1 byte). An unsigned byte will give you 0-255 so if you have numbers that don’t go higher than 255, save a byte!

\section*{How do I know I’m running out of memory?}

Usually you’ll observe your program behaving strangely and wonder what went wrong… You didn’t change anything in the code near the point where it’s messing up, so what gives? It’s running out of memory.

There are a couple of functions to tell you how much available memory you have.

\section*{Available Memory}

\begin{verbatim}
Answer by jfpoilpret
\end{verbatim}

In addition to what others have said (on which I fully agree), I would advise to read \texttt{this adafruit article} about memory; it’s well written, explains a lot of things about memory and provide hints on how to optimize it.

At the end of the read, I think you would get a quite complete answer to your question.

To sum it up, you have 2 possible optimization targets (depending on where you memory problems are located):

- **Flash** (ie Program Memory); for this, you can:
  - remove dead code (e.g. any code that is included but not used) and unused variables (that one also helps with SRAM)
  - factor out duplicated code
  - remove the bootloader altogether (you can gain between 0.5K for a UNO and 2 or 4K for other Arduino models); this has some downsides though

- **SRAM** (ie stack, heap and static data); for this you can:
  - remove unused variables
  - optimize the size of each variable (eg don’t use long -4 bytes- if you need only int -2 bytes)
  - use the right scope for your variables (and prefer stack to static data when possible)
  - reduce buffers size to the strict minimum
  - move constant data to PROGMEM (ie your static data will stay in Flash memory and won’t get copied to SRAM at program start); that also applies to constant strings for which you can use \texttt{F()} macro
  - avoid dynamic allocation if it is not absolutely necessary; you will avoid a
fragmented heap that may not shrink even after freeing memory

An additional approach to reduce SRAM usage is also described (but seldom used, because it is a bit heavy when coding and not very efficient), it consists in using EEPROM to store data built by your program, but not used until later when some conditions occur, when data can be loaded back from EEPROM.

Tags: flash (Prev Q)
EEPROM

Questions

Q: What is the real lifetime of EEPROM?

Tags: eeprom (Prev Q) (Next Q)

ATMEL says the cell lifetime of an EEPROM cell is about 100,000 write cycle/cell. Is this actually how the EEPROM performs in the wild?

If I do not change the value of a cell, does this stress the lifetime? For example, if I write the value 0xFF to the same cell again and again, is this any different to writing 0x00, 0xFF, 0x00 etc.

Tags: eeprom (Prev Q) (Next Q)

User: marlon-abeykoon

---

Answer by cybergibbons

As you state, the internal EEPROM has a lifetime of 100,000 write cycles. This isn’t a guess - a very significant proportion of ATmega328 will reach this number with no issues. I have tested three processors before, and all reached 150,000 cycles with no issues.

It is important to note the failure mode of EEPROM. Most “EEPROM destroyer” projects repeatedly read/write until the data is not written at all. Before this point, the EEPROM will still be damaged. This would be manifested by data not being retained for a reasonable period. It is unwise to rely on anything more than 100,000 write cycles for this reason.

EEPROM is different to the RAM on an ATmega. Writing to it is not simple or quick, but it is wrapped up in a friendly Arduino library, hiding this complexity from the user.

The first level of indirection is the EEPROM library, which is trivially simple, just calling two other functions for read and write. This calls eeprom_write_byte, found here.

This function uses inline assembly, so might not be easily understood. There is a comment that is easily understood though:

Set programming mode: erase and write

This hints to one of the complexities of dealing with EEPROM - to write to it, you first need to erase it. This means that if you call EEPROM.write(), it will perform a write cycle regardless of the value you are writing.

This means that repeatedly writing 0xFF will likely have the same effect as writing 0xFF,0x00,0xFF,0x00 etc.
There are ways to work around this - you can try calling EEPROM.read() before EEPROM.write() to see if the value is already the same, but this takes additional time. There are other techniques to avoid excessive EEPROM wear, but their use depends on your application.

**Answer** by **thedoctoreseprom**

I once ran an experiment on an external EEPROM with 1 million max rated cycles. It took about 6 million cycles to become majorly corrupted, and before that it had progressed having sporadic amounts of corruption.

When you say you do not change the value, i am assuming you are writing the same data to an address multiple times. This almost certainly would stress the life, although it would probably not stress the surrounding cells.

**Tags:** eeprom (Prev Q) (Next Q)

**Q: Is EEPROM persisted across program flashes?**

**Tags:** eeprom (Prev Q)

If I use EEPROM.write(someAddr, someValue) to write a value to an address, how long will that value “stick” to that address before being lost?

If I were to power down the Arduino, flash a whole new program to it, and power it on, could I still retrieve that old/cached value written from the previous program?

**Tags:** eeprom (Prev Q)

**User:** smeeb

**Answer** by **ignacio-vazquez-abrams**

Only if the EESAVE fuse is programmed (it is unprogrammed by default). If not then the EEPROM will be erased when a chip erase command is performed.

**Tags:** eeprom (Prev Q)
Questions

Q: I’m using too much RAM. How can this be measured?

Tags: sram (Prev Q)

I would like to know how much RAM I am using in my project, as far as I can tell, there’s no way to actually work that out (other than going through and calculating it myself). I have got to a stage in a rather large project where I have determined that I am running out of RAM.

I have determined this because I can add a section and then all hell breaks loose somewhere else in my code for no apparent reason. If I #ifndef something else out, it works again. There is nothing programatically wrong with the new code.

I suspected for a while that I was getting to the end of available RAM. I don’t think I’m using too much stack (although it’s possible), what is the best way to determine how much RAM I am actually using?

Going through and trying to work it out, I have problems when I get to enums and structs; how much memory do they cost?

first edit: ALSO, I have edited my sketch so much since starting, these are not the actual results I initially got, but they are what I am getting now.

<table>
<thead>
<tr>
<th>text</th>
<th>data</th>
<th>bss</th>
<th>dec</th>
<th>hex</th>
<th>filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>17554</td>
<td>844</td>
<td>449</td>
<td>18847</td>
<td>499f</td>
<td>HA15_20140317w.cpp.elf</td>
</tr>
<tr>
<td>16316</td>
<td>694</td>
<td>499</td>
<td>17419</td>
<td>440b</td>
<td>HA15_20140317w.cpp.elf</td>
</tr>
<tr>
<td>17346</td>
<td>790</td>
<td>426</td>
<td>18562</td>
<td>4882</td>
<td>HA15_20140317w.cpp.elf</td>
</tr>
</tbody>
</table>

The first line (with text 17554) was not working, after much editing, the second line (with text 16316) is working as it should.

edit: the third line has everything working, serial reading, my new functions, etc. I essentially removed some global variables and duplicate code. I mention this because (as suspected) it’s not about this code per sae, it has to be about the RAM usage. Which brings me back to the original question, “how to best measure it” I’m still checking out some answers, thanks.

How do I actually interpret the above information?

So far my understanding is:

```
'TEXT' is program instruction memory
'DATA' is variables (uninitialised?) in program memory
'BSS' is variables occupying RAM
```

since BSS is considerably less than 1024 bytes, why does the second work, but the first doesn’t? If it’s DATA+BSS then both occupy more than 1024.
I edited the question to include the code, but now I’ve removed it because it really had nothing to do with the problem (other than maybe poor coding practices, variable declarations and the like). You can review the code by looking back through the edits if you really want to see it. I wanted to get back to the question at hand, which was more based around: How to measure RAM usage.

Tags: sram (Prev Q)
User: madivad

Answer by alexan_e

You can use the functions provided AVRGCC: Monitoring Stack Usage.

The function was intended to check the stack usage but what it reports is the actual RAM that has never been used (during execution). It does so by “painting” (filling) the RAM with a known value (0xC5), and then checking the RAM area counting how many bytes have still the same initial value.

The report will show the RAM that has not been used (minimum free RAM) and therefore you can calculate the max RAM that has been used (Total RAM - reported RAM).

There are two functions:

- **StackPaint** is executed automatically during initialization and “paints” the RAM with the value 0xC5 (can be changed if needed).
- **StackCount** can be called at any point to count the RAM that hasn’t been used.

Here is an example of usage. Doesn’t do much but is intended to show how to use the functions.

```
// -----------------------------------------------------------------------------
extern uint8_t _end;
extern uint8_t __stack;

void StackPaint(void) __attribute__((naked)) __attribute__((section(".init1")));

void StackPaint(void)
{
  #if 0
    uint8_t *p = &_end;
    while(p <= &_stack) {
      *p = 0xc5;
      p++;
    }
  #else
    __asm volatile("  
      ldi r30, lo8(_end)\n"  
      ldi r31, hi8(_end)\n"  
      ldi r24, lo8(0xc5) /* STACK_CANARY = 0xc5 */  
      ldi r25, hi8(__stack)\n"  
      rjmp .cmp\n"  
    .loop:\n"      st Z+, r24\n"  
    .cmp:\n"      cpi r30, lo8(__stack)\n"      cpc r31, r25\n"      brlo .loop\n"      breq .loop::);  
  #endif

```
uint16_t StackCount(void)
{
    const uint8_t *p = &_end;
    uint16_t c = 0;
    while(*p == 0xc5 && p <= &_stack)
    {
        p++;
        c++;
    }
    return c;
}

// -------------------------------------------------------------------------------
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    // put your main code here, to run repeatedly:
    Serial.println(StackCount(), DEC); // calls StackCount() to report the unused RAM
    delay(1000);
}

Answer by jippie

When you figure out how to locate the generated .elf file in your temporary directory, you can execute the command below to dump a SRAM usage, where project.elf is to be replaced with the generated .elf file. The advantage of this output is the ability to inspect how your SRAM is used. Do all the variables need to be global, are they really all required?

Skip code block

avr-objdump -S -j .bss project.elf

project.elf: file format elf32-avr

Disassembly of section .bss:
00800060 <__bss_start>:
   ...
00800070 <measurementReady>:
   ...
00800071 <cycles>:
   ...
00800073 <measurement>:
    00 00 00 00
    ....
00800077 <measurementStart>:
    00 00 00 00
    ....
0080007b <timerOverflows>:
    00 00 00 00
    80007b:

Notice that this doesn’t show stack or dynamic memory use as Ignacio Vazquez-Abrams noted in the comments below.

Additionally a avr-objdump -S -j .data project.elf can be checked, but none of my programs output anything with that so I can’t tell for sure if it is useful. It supposed to
The main issues you can have with memory usage at runtime are:

- no available memory in the heap for dynamic allocations (malloc or new)
- no room left on the stack when calling a function

Both are actually the same as the AVR SRAM (2K on Arduino) is used for both (in addition to static data which size never changes during program execution).

Generally, dynamic memory allocation is seldom used on MCUs, only a few libraries typically use it (one of them is String class, which you mentioned you don’t use, and that’s a good point).

The stack and the heap can be seen in the picture below (courtesy of Adafruit):

Hence, the most expected issue comes from stack overflow (i.e. when the stack grows towards the heap and overflows on it, and then -if the heap was not used at all- overflows on the static data zone of the SRAM. At that time, you have a high risk of either:
- data corruption (i.e. the stack overwrites heap or static data), giving you ununderstandable behavior
- stack corruption (i.e. the heap or static data overwrites stack content), generally leading to a crash

In order to know the amount of memory that’s left between the top of the heap and the top of the stack (actually, we might call it the bottom if we represent both the heap and the stack on the same image as depicted below), you can use the following function:

```c
int freeRam () {
    extern int __heap_start, *__brkval;
    int v;
    return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}
```

In the code above, *__brkval points to the top of the heap but is 0 when the heap has not been used, in which case we use &__heap_start which points to __heap_start, the first variable that marks the bottom of the heap; &v points of course to the top of the stack (this is the last variable pushed on the stack), hence the formula above returns the amount of memory available for the stack (or the heap if you use it) to grow.

You can use this function in various locations of your code to try and find out where this size is getting dramatically reduced.

Of course, if ever you see this function return a negative number then it is too late: you have already overflown the stack!

---

**Tags:**  [sram](/tag/sram)  (Prev Q)
Questions

**Q: How high of a baud rate can I go (without errors)?**

The standard is 9600 baud. That’s just the *standard*. **Using an Arduino Uno SMD R2, what is the highest practical baud rate I can achieve?**

Bonus points for the audacious: **How would you go about creating an error checking mechanism and then increasing the baud rate ridiculous high to get high transfer rates?**

---

**User:** anonymous-penguin

**Answer** by connor-wolf

There are several factors here:

- How high of a baud-rate can the ATmega328P MCU achieve?
- How high of a baud-rate can the USB-Serial interface achieve?
- What is the oscillator frequency on the ATmega328P?
- What is the oscillator frequency on the USB-serial interface (if it has one)?
- How tolerant is the USB-serial interface of baud-rate mismatch?

All of these factors are relevant to determining the maximum achievable baud rate. The ATmega328P uses a hardware divisor from it’s clock-rate to generate the base-clock for the serial interface. If there is no integer ratio from the main clock to the bit-time of the desired baud rate, the MCU will not be able to *exactly* produce the desired rate. This can lead to potential issues, as some devices are much more sensitive to baud-rate mismatch then others.

FTDI-based interfaces are quite tolerant of baud-rate mismatch, up to several percent error. However, I have worked with specialized embedded GPS modules that were unable to handle even a 0.5% baud rate error.

General serial interfaces are tolerant of ~5% baud-rate error. However, since each end can be off, a more common spec is +-2.5%. This way, if one end is 2.5% fast, and the other is 2.5% slow, your *overall* error is still only 5%.

Anyways. The Uno uses a ATmega328P as the primary MCU, and a ATmega16U2 as the USB-serial interface. We’re also fortunate here in that both these MCUs use similar harware USARTs, as well as 16 Mhz clocks.
Since both MCUs have the same hardware and clock-rate, they’ll both have the same baud-rate error in the same direction, so we can functionally ignore the baud error issue.

Anyways, the “proper” answer to this question would involve digging up the source for the ATmega16U2, and working out the possible baud-rates from there, but since I’m lazy, I figure simple, empirical testing will work.

A quick glance at the ATmega328P datasheet produces the following table:

<table>
<thead>
<tr>
<th>Baud Rate (bps)</th>
<th>f_{osc} = 16.0000 MHz</th>
<th>f_{osc} = 16.4320 MHz</th>
<th>f_{osc} = 20.0000 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U2Xn = 0</td>
<td>U2Xn = 1</td>
<td>U2Xn = 0</td>
</tr>
<tr>
<td>2400</td>
<td>416 -0.1%</td>
<td>832 0.0%</td>
<td>479 0.0%</td>
</tr>
<tr>
<td>4800</td>
<td>207 0.2%</td>
<td>416 -0.1%</td>
<td>239 0.0%</td>
</tr>
<tr>
<td>9600</td>
<td>103 0.2%</td>
<td>207 0.2%</td>
<td>119 0.0%</td>
</tr>
<tr>
<td>14.4k</td>
<td>68 0.6%</td>
<td>138 -0.1%</td>
<td>39 0.0%</td>
</tr>
<tr>
<td>19.2k</td>
<td>51 0.2%</td>
<td>103 0.2%</td>
<td>59 0.0%</td>
</tr>
<tr>
<td>28.8k</td>
<td>34 -0.8%</td>
<td>68 0.6%</td>
<td>39 0.0%</td>
</tr>
<tr>
<td>38.4k</td>
<td>25 0.2%</td>
<td>51 0.2%</td>
<td>29 0.0%</td>
</tr>
<tr>
<td>57.6k</td>
<td>16 2.1%</td>
<td>34 -0.8%</td>
<td>19 0.0%</td>
</tr>
<tr>
<td>76.8k</td>
<td>12 0.2%</td>
<td>25 0.2%</td>
<td>14 0.0%</td>
</tr>
<tr>
<td>115.2k</td>
<td>8 3.5%</td>
<td>16 2.1%</td>
<td>9 0.0%</td>
</tr>
<tr>
<td>230.4k</td>
<td>3 8.3%</td>
<td>8 3.5%</td>
<td>4 0.0%</td>
</tr>
<tr>
<td>250k</td>
<td>3 0.0%</td>
<td>7 0.0%</td>
<td>4 -7.8%</td>
</tr>
</tbody>
</table>

Max. \(^{1}\) 1 Mbps 2 Mbps 1.152 Mbps 2.304 Mbps 1.25 Mbps 2.5 Mbps

---

1. \(UBRRn = 0, \ Error = 0.0\%\)

So given the max stated baud-rate of 2 Mbps, I wrote a quick test program:

```c
void setup(){};
void loop(){
    delay(1000);
    Serial.begin(57600);
    Serial.println("\r\rBaud-rate = 57600");
    delay(1000);
    Serial.begin(76800);
    Serial.println("\r\rBaud-rate = 76800");
    delay(1000);
    Serial.begin(115200);
    Serial.println("\r\rBaud-rate = 115200");
    delay(1000);
    Serial.begin(230400);
    Serial.println("\r\rBaud-rate = 230400");
    delay(1000);
    Serial.begin(250000);
    Serial.println("\r\rBaud-rate = 250000");
    delay(1000);
    Serial.begin(500000);
    Serial.println("\r\rBaud-rate = 500000");
    delay(1000);
    Serial.begin(1000000);
    Serial.println("\r\rBaud-rate = 1000000");
```
delay(1000);
Serial.begin(2000000);
Serial.println("\r\rBaud-rate = 2000000");
);

And then looking at the relevant serial port with a serial terminal:

So it appears the hardware can run at 2,000,000 baud without problems.

Note that this baud rate only gives the MCU 64 80 clock-cycles per byte, so it would be very challenging to keep the serial interface busy. While the individual bytes may be transferred very rapidly, there is likely to be lots of time when the interface is simply idle.

Edit: Actual Testing!

The 2 Mbps is real:
each bit-time is 500 ns, which matches exactly with what is expected.

Performance issues! Overall packet length:
500 Kbaud:
1 Mbaud:
2 Mbaud:
Note: The noticeable overshoot is due to poor scope probe grounding practices, and is probably not real. I’m using the ground-clip-lead that’s part of my scope probe, and the lead-inductance is likely the cause of the majority of the overshoot.

As you can see, the overall transmission length is the same for 0.5, 1 and 2 Mbaud. This is because the code that is placing the bytes in the serial buffer is poorly optimized. As such, you will never achieve anything better than an effective 500 Kbaud, unless you write your own serial libraries. The Arduino libraries are very poorly optimized, so it probably wouldn’t be too hard to get a proper 2 Mbaud, at least for burst transmissions, if you spent a bit of time on it.

The Arduino Serial Monitor window limits you to 115200, but that’s not the highest baud rate capable. You can read the Atmel and the FT232 (or whatever you’re using) datasheets to find out the maximum but I am able to successfully use 230400 (twice as fast as the largest the Arduino Serial Monitor supports) with no issues.

If you want to see the results in your computer, you will need another serial monitor that supports other baud rate options. I like CoolTerm and Termite.

Do note that this heavily depends on your clock speed as well.

Here’s a calculator to help you with calculating what’s possible.
This is probably one of the few aspects where el-Cheapo boards differ from original boards. The maximum serial transfer rate is pretty much only limited by the quality of the board and its layout. Once the serial data enters either AVR or USB interface chip, the data will be processed differently from the serial UART protocol.

Keep in mind though that the microcontroller has some basic hardware to shift in/out serial data to/from the IO pins, but the absolute maximum rate is limited to the 16MHz clock (for AVRs). Once a byte is moved to the serial buffer, the UART hardware will take over and push out / pull in the bits on its own. An AVR at best reaches 16M instructions per second and the interrupts used to fill the serial buffer have some overhead (at least 8 clock ticks for the interrupt handling + instructions to save current state + several instructions for actually filling the buffer). At a given bitrate, the protocol will run at a whopping n bits per second, but your controller needs more time to fill the serial buffer than it needs to actually output the data, resulting in a lower average throughput than you expect and the UART idling for a relatively long time. Disadvantage is the increased probability for bit errors.

Another effect to remember is that all the overhead required to push data out onto UART (or pull it in) cannot be spent in your actual program, again affecting the average practical throughput. You can only use every instruction cycle once, either for filling the buffer or for computing the main loop.

The maximum throughput is therefore dependent on the application you use (how fast is data generated/computed/ready to move to/from the serial buffer) and the actual ‘physical’ bitrate is only a small part of the design decision.

Tags: arduino-uno-smd (Prev Q)
Temperature Sensor

Questions

Q: Temperature sensors TMP36 and LM35 have temperature diff. (offset) in the same circuit

Tags: temperature-sensor (Prev Q)

I am trying to use the two temperature sensors TMP36 (from Sparkfun Inventor’s kit) and KEYES LM35.

In read TMP36 Sensor and I convert the result to Celsius using the following code:

```cpp
// TMP36 input sensor -> degrees Celsius calculation
TM36reading = analogRead(TM36sensor);
TM36voltage = (TM36reading/1024.0)*5.0;
// converting from 10mV per degree with 500 mV offset
// (TMP36 voltage - 500mV) times 100
TM36degreesC = (TM36voltage - 0.5) * 100;
```

In read LM35 Sensor and I convert the result to Celsius using the following code:

```cpp
// LM35 input sensor -> degrees Celsius calculation
LM35reading=analogRead(LM35sensor); // reads the LM35 output
LM35voltage = (LM35reading/1024.0)*5.0;
LM35degreesC=LM35voltage*100.0;
```

I have connected also a LCD display to monitor visually the values of these two sensors. Here is my full code. I use also a serial log. Here is a sample from this serial log (as you can see there is a difference/offset between 0.5 to 0.9 Celsius).

Skip code block

<table>
<thead>
<tr>
<th>Skip code block</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM36: 17.38, LM35: 16.60 diff -&gt; 0.78</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.11 diff -&gt; 0.78</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 17.87, LM35: 16.00 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 17.09 diff -&gt; 0.26</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 17.09 diff -&gt; 0.20</td>
</tr>
<tr>
<td>mean difference between sensors: 0.48</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.60 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.60 diff -&gt; 0.78</td>
</tr>
<tr>
<td>TM36: 16.41, LM35: 17.58 diff -&gt; 1.17</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.11 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.11 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 17.09 diff -&gt; 0.29</td>
</tr>
<tr>
<td>mean difference between sensors: 0.71</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.60 diff -&gt; 0.78</td>
</tr>
<tr>
<td>TM36: 16.89, LM35: 16.60 diff -&gt; 0.29</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.11 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.11 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.11 diff -&gt; 1.27</td>
</tr>
<tr>
<td>TM36: 17.38, LM35: 16.60 diff -&gt; 0.78</td>
</tr>
</tbody>
</table>
I power the Arduino from the computer using the USB cable. When the computer is ON we have the offset that I already demonstrated above. When the computer is turned off (after computer shutdown), immediately the sensor of TMP36 falls around 1 Celsius.. And then both sensors have similar temperature.

Why there is this temperature difference between TMP36 and LM35 sensors? How can we “fine tune” these sensors to monitor the real temperature?

Tags: temperature-sensor (Prev Q)

User: ggia

Answer by cano64

There are many things, first different sensors have different accuracy or they may not have been calibrated from the factory. In addition both are analog and analog reading is susceptible to noise. My guess is there is a high frequency noise on power line coming from the computer interfering with your readings. Try putting a small (100nF) capacitor between AREF and GND pin. You can always just take median of the measured values as the actual temperature.
PROGMEM

Memory types on an Arduino

RAM is precious on a microcontroller, PROGMEM (program memory) not quite as much so. On the Arduino Uno (using an Atmega328P) you have:

- 1 KB of EEPROM
- 2 KB of RAM
- 32 KB of PROGMEM (Flash Memory - where your program goes)

Clearly, there is a lot more program memory than RAM.

Constants are normally copied from PROGMEM into RAM

Constants in your program are initially stored in PROGMEM (where else could they be stored when the power is off?) but copied into RAM by compiler-generated initialization code at the very start of executing your sketch (program).

For example:

```cpp
Serial.println("Hello, world!");
```

The constant string “Hello, world!” is initially in PROGMEM, but copied into RAM at startup. This means that room for it has to be allowed for in RAM as well as in the Flash memory.

This uses up valuable RAM for what is really just a copy of something that will never change.

Another example:

```cpp
const int NUMBER_OF_ELEMENTS = 10;
const float table[NUMBER_OF_ELEMENTS] =
{ 1.0, 34.234, 324.234, 23.1, 52.0, 3.6, 5.6, 42.42, 1908, 23.456 } ;
```

Although this table of floating-point values won’t change, it is also copied into RAM. If you have a large table, you will soon run out of RAM.

The PROGMEM attribute

You can prevent this copying into RAM by adding PROGMEM to a data declaration, like this:

```cpp
const int NUMBER_OF_ELEMENTS = 10;
```
const float table[NUMBER_OF_ELEMENTS] PROGMEM =
{ 1.0, 34.234, 324.234, 23.1, 52.0, 3.6, 5.6, 42.42, 1908, 23.456 } ;

Accessing PROGMEM constants

Unfortunately the compiler (on the AVR Arduinos anyway) expects all data to be in RAM and does not allow for accessing data directly from PROGMEM.

You can make a small library PROGMEM_readAnything.h:

```c
#include <Arduino.h>  // for type definitions

template <typename T> void PROGMEM_readAnything (const T * sce, T& dest)
{
    memcpy_P (&dest, sce, sizeof (T));
}

template <typename T> T PROGMEM_getAnything (const T * sce)
{
    static T temp;
    memcpy_P (&temp, sce, sizeof (T));
    return temp;
}
```

Copy the above into a text file, save it as PROGMEM_readAnything.h, and place that into a folder called PROGMEM_readAnything, and put that folder inside your Arduino libraries folder which itself is inside your sketchbook folder. Like this:

`{(sketchbook location)/libraries/PROGMEM_readAnything/PROGMEM_readAnything.h`

Now you can access a PROGMEM constant like this:

```c
float thisOne;
PROGMEM_readAnything (&table[3], thisOne);
```

Or:

```c
float thisOne = PROGMEM_getAnything (&table[3]);
```

The F() macro

For simple string constants, you can use the F() macro to easily keep strings in PROGMEM, like this:

```c
Serial.println (F("Hello, world!")));`
Questions

Q: Why can I not use pointers instead of array with PROGMEM?

Tags: progmem (Prev Q)

I’m currently changing some libraries to use flash instead of RAM for string storage so that I do not run out of SRAM on a project.

Some strings in the library are declared in this manner:

```c
const char *testStringA = "ABC";
```

This is different to how I normally see this done:

```c
const char testStringB[] = "DEF";
```

However, I think that these two are equivalent when declared const and initialised in the declaration. Both work fine in code.

I attempted to move these to flash:

```c
const prog_char *testStringC PROGMEM = "GHI";
```

I then found that this didn’t work. It was producing gobbledygook when printed.

However, following the more usual pattern of:

```c
const prog_char testStringD[] PROGMEM = "JKL";
```

works fine.

I can see in the disassembly:

```plaintext
00000068 <testStringC>:
   68:   04 01             ..
0000006a <_ZL11testStringD>:
   6a:   4a 4b 4c 00      JKL.
```

So it is clear the pointer and PROGMEM results in the string/array not being initialised.

Why is this?

Example code:

```c
#include <avr/pgmspace.h>
const int BUFFER_LEN = 20;

const char *testStringA = "ABC";
const char testStringB[] = "DEF";
const prog_char *testStringC PROGMEM = "GHI";
const prog_char testStringD[] PROGMEM = "JKL";

void setup()
{
   Serial.begin(9600);
}

void loop()
{
   char buffer[BUFFER_LEN];
}```
Serial.println(testStringA);
Serial.println(testStringB);
strncpy_P(buffer, testStringC, BUFFER_LEN);
Serial.println(buffer);
strncpy_P(buffer, testStringD, BUFFER_LEN);
Serial.println(buffer);
delay(1000);
}

**Tags:** progmem (Prev Q)

**User:** cybergibbons

**Answer** by zmo

Well, this question has been wisely answered in an answer to Stack Overflow question C: differences between char pointer and array.

Basically, what you’re declaring as PROGMEM with,

```c
const prog_char testStringD[] PROGMEM = "JKL";
```

is both the array and the memory it points to, that is, the elements of the array, both in current’s scope stack. Whereas with:

```c
const prog_char* testStringC PROGMEM = "GHI";
```

you declare a PROGMEM pointer to a constant string that may stay elsewhere in memory, but not declared as a PROGMEM string.

Though I did not test that, but you should try to declare:

```c
const prog_char* testStringC PROGMEM = F("GHI");
```

to actually allocate the pointed string within the PROGMEM space. I guess it should be working, using Arduino’s F() macro, which adds a lot of boilerplate code to actually have the same result as the array declaration.

As said in comments, if not in a global context, the PSTR() macro could be used instead of the F() macro.

Simpler is better: use the array declaration, not the pointer one!

Cf that other answer, the __flash qualifier is a third solution ;-)

---

**Answer** by ignacio-vazquez-abrams

What this line:

```c
const prog_char *testStringC PROGMEM = "GHI";
```

does is to write prologue code to copy the characters in the string to SRAM, and then initializes the pointer stored in flash to this SRAM location. You must load the pointer via normal means, and then dereference the pointer as usual.

```c
const char *str = pgm_read_word(&testStringC);
Serial.println(str);
```
This line:

```
const prog_char testStringD[] PROGMEM = "JKL";
```

creates the *array of characters* in flash, allowing you to access it as expected.

**Tags:** progmem (Prev Q)
Electricity

Questions

Q: How can Arduino control with a servo with only one wire?

Tags: electricity (Prev Q)

I was playing with getting my servos to be on dedicated power, separate form the Arduino entirely, but obviously controlled by the Arduino. I originally had the ground of the Arduino joined with the - of the battery, but to my surprise my Arduino continued to control the servos when I remove the ground wire entirely from the Arduino.

I was under the impression that all electricity must form a complete circuit, and be a closed loop. But my Arduino can obviously send a signal to a component that has no return path to itself.

I feel like I’m missing something fundamental about electricity. So my question is:

Why does this work?
The return signal is probably the other wire’s servo. This only works when the control pulses to both servos are not in sync. Pull one signal wire off the breadboard and both servos will stop working.

You need to interconnect grounds or you will get strange effects sooner or later.

**Tags:** electricity (Prev Q)
Global System for Mobile Communications (GSM) is part of the second generation (2G) of mobile phone network technology. It was originally designed for two-way voice communication, but was later expanded to support more general data transfer, via GPRS and EDGE. It has been superseded by other technologies, but remains widely available.

Arduino support for GSM is often achieved by using the Arduino GSM Shield, although generic components can be used as well. In either case, a working SIM card and (typically) a contract or pre-payment plan with a mobile network operator is also required. The GSM Library is available to facilitate programming.

Arduino projects involving GSM often focus on text-messaging. However, audio communication is also possible.
Questions

Q: SerialGSM for GSM shield without SoftwareSerial

Tags: gsm (Prev Q) (Next Q)

SerialGSM is a library for simplifying GSM shields.

This is sample code for sending SMS:

```c
#include <SerialGSM.h>
#include <SoftwareSerial.h>
SerialGSM cell(2,3);
void setup(){
  Serial.begin(9600);
  cell.begin(9600);
  cell.Verbose(true);
  cell.Boot();
  cell.FwdSMS2Serial();
  cell.Rcpt("+972123456789");
  cell.Message("hello world");
  cell.SendSMS();
}

void loop(){
  if (cell.ReceiveSMS()){  
    Serial.println(cell.Message());
    cell.DeleteAllSMS();
  }
}
```

As you can see, he uses software serial.

I intend to use this for my school thesis, but I am using a shield, so I won’t be using Software Serial.

What bothers me is this line

```c
SerialGSM cell(2,3);
```

In my case, do I declare it like this?:

```c
SerialGSM cell(0,1);
```

Or will there be conflicts with serial?

EDIT: In other words, how do I declare the SerialGSM object using hardware serial and not software serial?

Tags: gsm (Prev Q) (Next Q)

User: user1584421

Answer by peter-r.-bloomfield

SerialGSM is designed to work with software serial, as you can see that from the class declaration in the SerialGSM.h header:

```c
class SerialGSM : public SoftwareSerial {
```
Unfortunately, changing it to use hardware serial is more complicated than it looks. There are two potential ways you could go about it:

1. Derive from `HardwareSerial` instead of `SoftwareSerial`. This would give you flexibility, but you’d have to take care of passing all the right data to the `HardwareSerial` constructor, which is quite messy.

2. Don’t derive from anything. Use the global `Serial` object, and manually replace all serial function calls with their `Serial.xxx()` equivalents. This is probably easier, but it restricts you to a specific serial port.

Both approaches involve quite a lot of coding, so they aren’t trivial tasks. It’s also important to note that `SerialGSM` currently uses the global `Serial` object to output debug information. You’d have to remove or modify that before converting to hardware serial, or else it could interfere.

---

**Answer** by user1584421

Never mind, I went with the GSM SMS example from arduino 1.0.5 (Don’t know the version of arduino IDE that included this) and works fine.

If you are considering using the SerialGSM library there is no need to!

CAUTION: In the Arduino’s library, when you declare the phone number, you have to use international format

(Thanks bloomfield for the info you provided in the previous answer)

---

**Tags:** gsm (Prev Q) (Next Q)

---

**Q: Difference between SoftwareSerial and Serial**

**Tags:** gsm (Prev Q)

From the sample code

```c
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(57600);
  while (!Serial) {
    // wait for serial port to connect. Needed for Leonardo only
  }
  Serial.println("Goodnight moon!");
  // set the data rate for the SoftwareSerial port
  mySerial.begin(4800);
  mySerial.println("Hello, world!");
}

void loop() // run over and over
{
  if (mySerial.available())
    Serial.write(mySerial.read());
}
What is the difference between, why use 2 types of calls to make AT command calls.

    SoftwareSerial mySerial (10,11) mySerial.begin(10,11)

and

    Serial.begin(9600)

Edit, this answers my question in my own words based on @jfpoilpret response.

1. It is important to **initialize both Serial and SoftwareSerial**, since SoftwareSerial is used to send the actual commands to the GSM, and Serial is used to view all the commands and their responses on the debug port of the GSM
2. The pins on the GSM and Arduino are used to connect the the 2, and pass on actual commands. While the RS232 is only used for debugging. We can however connect a RS232 to USB connector so as to connect the GSM to the desktop(or laptop)
3. The code in the loop is essential to help convert all actual commands to the debugging RS232 port, and the other way around, so as to enable us to use a tool like cutecom (on ubuntu) to pass commands to the GSM serial input pins.

    Important Note : When using cutecom ensure that you select **CR line end** to send commands.

Tags: **gsm (Prev Q)**

User: **siddharth**

---

**Answer** by **jfpoilpret**

All Atmel MCU, on which Arduino models are based, include special hardware for serial communication; this part of the MCU is called **USART** (stands for *Universal Synchronous/Asynchronous Receiver/Transmitter*).

Based on the Arduino model you have, you may have only one (eg on UNO) or more USART (leonardo, mega…) on the board MCU.

Each MCU USART is mapped to specific pins of the board, you cannot change those pins.

On Arduino, if you use the monitor to send debug information to your PC, then one USART is used, this is when you used Serial1.

If you need more than one serial communication link (e.g. because you have some devices that communicate through serial) but your Arduino model has only one USART, then you have to use the library **SoftwareSerial** that “simulates” the job of an USART only by software. Hence it is much less optimized than the hardware USART.

One advantage of **SoftwareSerial** is that you can map it to any pair of pins you like.

Going to your code sample:
if (mySerial.available())
    Serial.write(mySerial.read());
if (Serial.available())
    mySerial.write(Serial.read());

Obviously here, Serial is used for debugging (sending info to the serial monitor of your PC), whereas mySerial is used to communicate with the GSM device.

So this sketch is used to allow you to type AT commands on the monitor, which will then be forwarded to the GSM device; conversely, any strings returned from the GSM device will be echoed to the serial monitor.

Tags: gsm (Prev Q)
Emulation

Questions

Q: Can I program for Arduino without having a real board?

Tags: emulation (Prev Q)

I would like to start the development of some basic Arduino projects but I don’t own an Arduino board yet. Is there a way I can write my code and emulate/test it using a desktop computer so after my board arrives I just have to upload and run my project on it?

Tags: emulation (Prev Q)

User: totymedli

Answer by anindo-ghosh

There are a whole slew of Arduino simulators out there, many free, and some paid products as well.

- The CodeBlocks Arduino development environment includes a free Arduino simulator, still under development but functional.
- Simuino simulates the Arduino Uno and Mega pins - not a pretty-looking realistic simulator, but it works.
- The Python based Arduino Simulator is another option, that plays well with the official IDE
- Virtronics Simulator for Arduino looks promising, but I don’t see why I would pay $14.99 for it, when I could buy one or more actual Arduino clones for that price

Many other Arduino simulators are out there if you search, and new ones are being announced, even crowdfunded, all the time.

Answer by akellyirl

I like to use 123dcircuits.io for this.

It’s cloud based and has some nice features including PCB design and collaboration.

IMO the Arduino simulation and capability is impressive and intuitive.

Answer by 123d-circuits

Plenty of options exist in the world of electronics simulators, but 123D Circuits Electronics Lab is probably the most versatile one out there.

- You can compose your circuit on a virtual breadboard which looks just like real.
- You can actually watch a LED blinking or press a button during real-time simulation...
rather than struggling with abstract waveforms.

- You can combine digital devices like a 555 timer or shift registers with (interactive) analog components like resistors and LEDs.
- You can program an Arduino and simulate the microcontroller together with your analog circuit around it.
- You can debug an Arduino: when the simulator reaches your breakpoint, the simulation is paused and you can easily check every variable in your code and every voltage or current in your circuit.
- You can use and observe different communication protocols like UART, I2C and SPI.
- You can pause the simulation in the middle of an I2C sequence by putting a breakpoint in the driver code (e.g. interrupt service routine for I2C events)
- It’s FREE!

This is an example circuit that has with an IR Remote and an LCD Screen.

Disclaimer: We are the makers of 123D Circuits

Tags: emulation (Prev Q)
I made a simple circuit controlled by the ATMEGA IC then tried to encase it in resin. It died. What is the thing that caused it to die? I’ve had other circuits working fine in resin. I want to try again, what steps can I take to prevent this?

I’ve looked for obvious shorts but can find none.

Resin heats up when it cures, could that be the issue?
Potting of circuits can lead to various issues. It can’t be told exactly what caused your device to fail until you uncover it like an archeologist.

Common problems are:

- Resin entering a spring loaded contact and canceling out contact forces
- Resin exsuding corrosive substances ruining contacts
- Resin expanding during curing and opening spring contacts or even breaking up solder contacts
- Same effect after cool down of resin

Later and often unexpected problems are:

- thin gaps around potted objects (i.e. no surface contact) literally sucking in water.
- potting material producing water itself during curing.

Many of these problems can be circumvented by applying conformal coating, as suggested by comments and apparently proven right by your experiments. If it is not necessary refrain from using sockets and solder connections directly to the components. You won’t be able to salvage them, anyway.

**Tags:** enclosure (Prev Q)
Questions

**Q: Ethernet shield and IPv6 support**

Tags: tcpip (Prev Q)

Have you heard about support for IPv6 protocol for any Ethernet shield? As far as I know only IPv4 is supported. Maybe some one would like to write library?

Tags: tcpip (Prev Q)

User: max

---

**Answer** by anonymous-penguin

**There is no library, and there physically cannot be one.** IPv4 is built into the Ethernet shield itself. (It has a Wiznet 510 chip.)

Shields based on the MicroChip ENC28J60 chip (such as those sold by Ekitszone and Nuelectronics) are suitable for the implementation of an IPv6 or dual IPv4/IPv6 stack. This approach requires the TCP state machine to be implemented in software and results in more complex arduino code.

Source: [this tutorial](#). *(You may find it useful, I’d recommended reading it.)*

There is [this library for IPv6 on other hardware, not for the official Ethernet shield](#).

---

Tags: tcpip (Prev Q)
Oscillator Clock

Questions

Q: Why is there a clock in my arduino?

Tags: oscillator-clock (Prev Q)

Of course I know it is usefull to control the time of a delay, but any other use?

With a 16Mhz clock, each line of my program will take 1/16000000 second, right?
even a very very big line will only take 1/16000000 of a second?

With a 8MHz clock, each line will take the double of the time, right? So the whole “main loop” will take the double time too, right?

Tags: oscillator-clock (Prev Q)

User: n0tis

Answer by majenko

Why is there a clock in my arduino?

Because that is how computers and microcontrollers, etc, work.

With a 16mhz clock, each line of my program will take 1/16000000 second, right?

No.

even a very very big line will only take 1/16000000 of a second?

No.
The clock defines at what speed the machine code instructions are fetched from memory and executed. Most instructions take 1 clock cycle, but some take more.

One line of C code may be compiled into any number of assembly instructions, which then get converted into machine code (raw numbers). That could be anywhere from 1 assembly instruction to thousands of assembly instructions, depending on what the line does.

With a 8mhz clock, each line will take the double of the time, right? So the whole “main loop” will take the double time too, right?

Every operation in the chip is governed by the clock. If the clock is half the speed the chip is running at half the speed, so every operation will take twice as long, yes.
Not answered in @Majenko’s post: Yes, with a 8MHz clock, each line will take the double of the time. Unless the line waits for something that is not clock driven - e.g. external input.

In addition to @Majenko’s answer, a CPU has a clock to ensure the instructions are complete, before the next step starts. A CPU is made out of lots of transistors (I found a reference that indicated the AtMega was in the range of low millions, give-or-take an order of magnitude).

During a cycle, the electricity flows through the chip, turning transistors on/off, the results of which will turn more transistors on/off, and so forth down the line. While this is happening, some parts of the chip have a “wrong” value - you can think of this as being half-way through a calculation (you’ve added the one’s column and the 10s column, and are about to start on the 100s column). You don’t want this to affect the outside world, so (for example) your output pins are latched - held at whatever value they are - until the instruction is complete. How long it takes to complete an instruction varies, but the manufacturer works out the slowest instruction under the worst circumstances.

For the AtMega (which is the chip on the Arduino), Atmel (who designed the chip) has declared this is 1/20,000,000 of a second - this is 20MHz.

Note that not all microprocessors run all their instructions at 1 instruction per cycle - some instructions might take 1, or 2, or 10 cycles. Pipelining makes things even more complicated - a processor might do some of the work (e.g. fetch the next instruction) in one cycle, execute it in the next - but while it is executing instruction 1, it can also fetch the next instruction. To do this, it might need to take a guess at what instruction comes next (in the case of the machine-code equivalent of a “goto” - such is used for loops), and if it guesses wrong, it has to cope with that; throw away the instruction it retrieved and retrieve the next one, losing a cycle.

The Wikipedia page on [instruction pipelining](https://en.wikipedia.org/wiki/Instruction_pipelining) shows an example of a RISC chip pipelining in 5 stages - instruction fetch, instruction decode, execute, memory access, and write-back. So, you can have 5 instructions at some stage of execution, overlapping. Until the “writeback” phase, the instructions have no real effect. You can think of this as an assembly line - it takes 7 minutes to put a widget together, but it can be broken down into 5 stages, the longest stage taking 2 minutes. Once every two minutes, each partly completed widget is moved by the assembly line to the next station. You get one widget out every two minutes - The “clock” can only tick as fast as the slowest step. If you push the widget out any faster, the “bottleneck” will get more and more widgets queued up.

**Tags:** oscillator-clock (Prev Q)
Use this tag for any question which involves infrared in an Arduino project. This can cover a broad range of topics, so please try to use other tags as well, where appropriate.
Questions

Q: Infrared TV Remote to Arduino

Tags: ir (Prev Q)

On the hardware side of things, what is involved in making the Arduino receive signals from a TV Remote? What kind of infrared receiver is needed? Do TV remotes use some kind of channels, encoding, or encryption requiring special receivers or else additional programming work to decrypt the signals?

Tags: ir (Prev Q)

User: hk1

---

Answer by sachleen

The most common (not sure what others are used) carrier frequency is 38kHz. You need an IR receiver tuned to that frequency to get the signal from the remove. See this page for way more information.

There is no encryption or special receivers needed as long as the carrier frequencies match.

That’s really all the hardware you need. On the software side of things, you read this signal and decode it. There’s a neat little library for doing exactly that. It’s called Arduino IRemote.

It even has an example sketch which tells you just what the decoded value is of whatever button you pressed so you can use that in your other programs.

---

Answer by peter-r.-bloomfield

Responding to TV remotes is surprisingly simple, because they typically all work on exactly the same principle, with no concept of channels or encryption.

What you need is an IR receiver module, which you can buy fairly cheaply from many electronics suppliers. Anything intended to work with remote controls should be suitable. The important part is that it will work with ~38 kHz signals.

It will typically be a 3 pin component. Many are able to operate at 5 volts, which means you can usually connect the IR module’s voltage and ground pins directly to the Arduino’s power supply. The data (or signal) pin can typically be connected straight to any GPIO pin. Always check the component’s data sheet though. It will often recommend using some additional components (e.g. resistor and capacitor) to smooth out power fluctuations. That shouldn’t be necessary for a small project though.

When an IR signal is detected, the module will usually output a series of pulses on the data pin. These will be digital pulses, so there’s no need to use an analog-to-digital converter. What matters is typically either the length of each pulse (Pulse Width Modulation), or the length of the gap between consecutive pulses (Pulse Distance Modulation).
The timings of the pulses/gaps will be very fast; usually in the region of 400 to 2000 microseconds. That means it’s probably too fast to try to detect them using repeated calls to `digitalRead()`. Instead, there is a core Arduino function called `pulseIn()`. When it’s called, it will wait for a pulse to arrive on the specified GPIO pin. It will then return the approximate length of the pulse. You can detect high pulses for PWM, or low pulses for PDM.

Precise timing values will vary considerably, so you can ignore small variations. The key is to categorise each pulse as short or long, corresponding to a binary 0 or 1. A good starting point is to consider anything below 1000 usec as short, and everything else as long. Check what timings you actually receive though, and adjust that value accordingly.

Call `pulseIn()` repeatedly, storing each result. Keep going until no more pulses arrive (you will need to give it a timeout after which it gives up waiting, e.g. 10,000 usec or more).

Your TV remote will output a unique sequence of short/long pulses for each button, typically 32 bits or less (possibly with extra pulses at the beginning/end). All you need to do is figure out which sequence corresponds to each button. You could store that information in a lookup table, and respond accordingly when a sequence is recognised.

---

**Tags:** ir (Prev Q)
Web Service

Questions

**Q: Post Data To a Web Service From Arduino**

Tags: web-service

If you want to submit sensor data such as temperature to a remote server/database somewhere you need to use some kind of call to a web server since it isn’t possible to connect directly to a database from the Arduino.

How do you post data to a JSON web service from an Arduino that’s connected to the Internet?

Tags: web-service

User: hk1

---

**Answer** by sachleen

There is the aJson library that allows you to work with JSON objects in Arduino. However, depending on the complexity of your program, I would just do it manually to save memory. You may be able to just copy the functions you want out of the library.

Then check out the WebClient library which has an example for making a POST request at the bottom.

---

Tags: web-service
Questions

**Q: How to organize my Arduino projects’ structure for easy source control?**

Tags: project

It’s been a long time I’ve been looking for a good answer to this question.

Typically, any Arduino project but the simplest one will include:

- The main source code file `MyProject.ino`
- Libraries specific to the project (`MyProjectLibrary1.h`, `MyProjectLibrary1.cpp`...)
- Third-party libraries (generally free open source, added manually to Arduino libraries directory)
- Schematics, PCB diagrams
- Documentation
- …

All this makes it hard to keep the whole code and doc of one project under Source Code Management (e.g. on Subversion, Git or GitHub).

Managing source control of your project means managing the version of all files used by the project including 3rd-party libraries.

Now for a single project, I need to define a directory structure that:

- Includes all project files as described above
- I can entirely commit to a Source Code Management tool (including 3rd-party dependencies)
- I can checkout anywhere on my hard drive and build the project from there (does it have to be a single location as imposed by Arduino IDE)
- I can zip into a self-contained archive that I can send to a friend for him to build as easily as possible (no extra manual download)

What I find particularly tricky with Arduino projects is the management of external libraries dependencies. Java projects developers have maven repositories for that and that helps a lot in managing all external deps. But we don’t have an equivalent system for Arduino libraries.

I would be interested to know how other Arduino project makers deal with these aspects in their own projects.

Also note that I am open to changing my development process, including my IDE (currently I use Eclipse with the Arduino plugin most of the time, and then I ensure my
projects can also work directly with the Arduino IDE).

**Tags:** project

**User:** jfpoilpret

---

**Answer** by sachleen

The simplest way to do this is to copy the header and code files of the library into your source directory and include them.

```
myproject/  
    myproject.ino  
    somelib.h  
    somelib.cpp  
```

In your code, you can do include "somelib.h"

The down side to this is that the libraries must be in the same folder, not sub folders, so it makes your directory look messy.

Regarding the directory structure of my entire project, including schematics and documentation, mine usually looks like this:

```
myproject/  
    schematics/ - eagle files or whatever you may have  
    docs/ - include relevant datasheets here  
    test/ - any test cases or other code to test parts of the system  
    myproject/ - since Arduino code must be in a directory of the same name  
    myproject.ino  
```

---

**Answer** by zmo

My way of organizing an arduino project is quite simple, all my project are git repositories so that there is at least the following:

- **project**
  - project.ino
  - **Makefile**

I have a preference using my favorite editor and a **Makefile** which I have crafted to work against most use cases (and I even improved on that one that I’m going to share soon).

For the libraries, I prefer to keep them as their own repositories and use git submodule to include them to the project. As many libraries written by the community are shared as git repositories, that’s a good generic solution. Then, within the Makefile, I just have to add the libraries path I want to include in the `LOCALLIBS` variable.

Though, for some projects, it makes sense to encapsulate the libraries into a hardware abstraction layer library crafted for the project, then I prefer to use a path such as:

- **project**
  - project.ino
  - **Makefile**
Though, with arduino 1.5.x a new way to specify libraries is offered, that will offer a way to create and build arduino projects the same way we already do with pipy and virtualenv in python, i.e. you define the set of libraries you need and they get downloaded.

Git submodules are extremely powerful when it comes to organizing multiple nested repositories. Handling multiple libraries from different sources, and even handling parts of your own project which may be stored at different sources becomes easy with git submodules.

**Directory Structure**

A way to organize your projects would be:

- **projectA** - Parent Directory
  - **projectA** - Source code directory containing Arduino code
    1. projectA.ino
    2. header.h
    3. implementation.cpp
  - **docs** - Your main documentation directory
  - **schematics** - these may be maintained separately on a separate Git repo or part of the same repo
  - **libs** - This will contain your third party libraries.
    1. libA - These may be maintained as third party repositories
    2. libC - …
  - **license**
  - **README**
  - **Makefile** - Necessary to handle dependencies across directories

**Workflow**

You would follow your normal cycle of make changes, add and commit as far as the main repository is concerned. Things get interesting with the sub-repositories.

You have the option of adding a repository into the parent directory of your main repository. This means that any part of you directory structure, i.e. docs, schematics,
etc. can be maintained as a separate repository and continuously updated from.

You can do this using the `git submodule add <repo.git>` command. To keep it up to date, you can use `git submodule update <path>`.

When it comes to maintaining multiple third party libraries within your repository such that each can be version controlled in itself or each can be kept up to date if need be, git submodule again saves your day!

To add a third party repo to **libs**, use the command `git submodule add <lib1.git> libs/lib1`. Then, to maintain the library at a fixed point in the release cycle, checkout the library and make a commit. To keep the library up to date, use the command `git submodule update <path>`.

Now, you can maintain multiple repositories within a main repository as well as multiple third party libraries in their independent stages of release.

**Versus Single Directory Approach**

While the **single directory approach** is the simplest, it is not possible to version control parts of a directory without a lot of pain. Hence, the simple approach fails to accomodate different repositories with varying states in the project.

This approach allows maintaining multiple repositories but brings in the need for a Makefile to handle the compilation and linking process.

Depending on the complexity of your project, the optimal approach can be selected.
RFID

Questions

Q: How do I use RFID-RC522 with an Arduino?

Tags: rfid

I am working on a project for my local Makerspace, we have limited budget so I was hoping to use the RFID-RC522 el’cheap’o RFID/NFC readers on eBay, I received a couple of SPI based boards, they work, however the code examples for them are limited.

I have found several different libraries and settled on this one:
https://github.com/ljos/MFRC522

The trouble is that all the code available online seem to spawn from some Chinese guys Python code that people have translated, and hacked into an Arduino library.

The code works, but Mifare cards are meant to have 4, 7 or 10 byte UIDs and the example/library is returning a 5 byte serial number.

There is no documentation and the NXP datasheet is incomprehensible… Additionally it seems to work with most cards, but it doesn’t work with Mastercard PayWave cards which conform to the ISO 14443 standard. The more expensive RDM880 reader which is based on the MFRC500 works fine and has a nice library but the cost makes the implementation impossible.

So, can someone help me to get this NXP MFRC522 based unit reading the UID from all ISO 14443 cards.

Skip code block

```c
#include <SPI.h>
#include <MFRC522.h>

#define RFID_SS 10
#define RFID_RST 5

MFRC522 rfid( RFID_SS, RFID_RST );

void setup(){
  SPI.begin();
  Serial.begin(115200);
  rfid.begin();
}

void loop(){
  byte data[MAX_LEN];
  byte uid[5];

  if( rfid.requestTag( MF1_REQIDL, data ) == MI_OK ){
    if( rfid.antiCollision( data ) == MI_OK ){
      memcpy( uid, data, 5 );
      for( int i = 0; i < 5; i++ ){
        Serial.print( uid[i], HEX );
        Serial.print( ' ' );
      }
      Serial.println();
    }
  }
}
```
Answer by staquur

Sorry, but I must disagree with your ‘NXP datasheet is incomprehensible’. From the FIRST PAGE of the MFRC522 datasheet:

‘Remark: The MFRC522 supports all variants of the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus RF identification protocols’

Here it says the MFRC522 covers just a part of ISO/IEC 14443. The following are missing from the NXP RC522:

*MIFARE DESFire EV1* (includes AES encryption)

*MIFARE DESFire EV2* (includes MIsmartApp, Transaction MAC, Unlimited Applications MIFARE Plus drop-in replacement for MIFARE Classic with certified security level (AES 128 based))

*MIFARE SAM AV2* (secure access module that provides the secure storage of cryptographic keys and cryptographic functions)*

These are probably the ones used in bank cards. Again, from the FIRST PAGE of the MFRC500 datasheet:

‘All protocol layers of the ISO/IEC 14443 A are supported.’

You’ll have to plow through all the different MIFARE / ISO/IEC 14443 specs to find out how many ID bytes there are for the different types (I suspect different types return a different quantity of bytes).

So, basically, you’re screwed. Spring for the expensive MFRC500-based reader. I assume that with the higher cost you also get a high-class API, documentation and examples or even (gasp!) tech support.

Cheers
Osx

Questions

Q: Rename device name (ch340 usb to serial) Mac OS

Tags: osx

Can’t upload sketches to Arduino Nano clone with USB to serial chip ch340.

After installing drivers and plugging in, device has the name /dev/tty.wch ch341 USB=>RS232 fa130 (with spaces), but Arduino can’t recognize the full name (only tty.wch).

Is it possible to rename device in Mac OS or change device identifier?


Tags: osx

User: sonique

Answer by gedson-faria

I used a terminal command:

```bash
sudo ln -s /dev/tty.wch\ ch341\ USB\=>RS232\ 1450 /dev/tty.USB0
```

where /dev/tty.wch\ ch341\ USB\=>RS232\ 1450 is the original name not recognized by Arduino; and /dev/tty.USB0 is the new port name.

Answer by sonique

I send email about this issue to WCH, and they change drivers to the latest version on their website, this is direct link http://www.wch.cn/downloads.php?name=pro&proid=178 (driver works and device has proper name in dev/)

Answer by sjunnesson

For Yosemite you seem to need to run this command in the Terminal and then restart the computer to get it to work.

```bash
sudo nvram boot-args="kext-dev-mode=1"
```

Tags: osx
Copyright

This book was compiled from arduino, using the latest data dump available at archive.org. A selection of the best questions about Arduino Stack Exchange, as voted by the site’s users have been selected for inclusion in this book.

I am not affiliated or endorsed by StackExchange Inc, although I do have a Stack Exchange user profile there under the username George Duckett.

All questions and content contained within this book are licensed under cc-wiki with attribution required as per Stack Exchange Inc’s requirements.

If you have or know of copyrighted content included in this book and want it to be removed please let me know at georgeduckett@hotmail.com.

The cover image was altered from it’s original source, found below.

Wikipedia - SparkFun Electronics / CC BY 2.0