Arduino Tutorial I

Updated: 1/12/2013

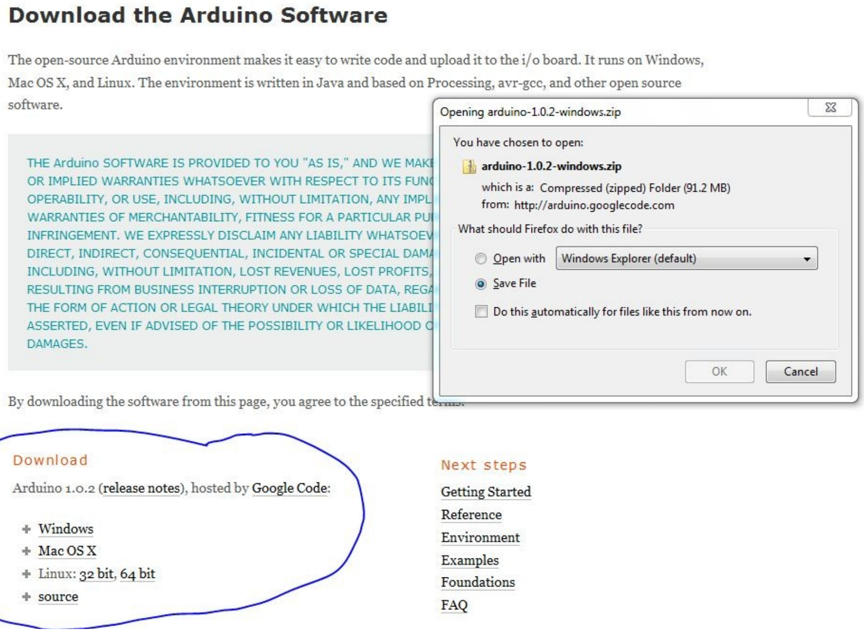
Introduction:

As a reminder, all posted safety guidelines must be followed at all times. The current safety code is posted around N3D an available at http://engineering.nd.edu/groups/n3d/safety.

In this tutorial you will be introduced to the Arduino Uno and the programming language associated with it. The major goals of this tutorial are to introduce you to the IDE and write two programs to illustrate important functions in the Arduino language. This first program will introduce digital reading and writing. You will then write a second program which will dim an LED based on the state of a potentiometer. The analog value read from the potentiometer will also be displayed in a serial communication created between your computer and the microcontroller. This second program will illustrate analog read/write and serial communication.

Download Arduino IDE and Getting Started:

1. Go to http://arduino.cc/en/Main/Software and download (save file) the appropriate developer for your computer.



Download the Arduino Software

The open-source Arduino environment makes it easy to write code and upload it to the i/o board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

THE Arduino SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNC OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPL WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PU INFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEV DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMA INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGA THE FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILI ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD O DAMAGES.

By downloading the software from this page, you agree to the specified terms.

Opening arduino-1.0.2-windows.zip

You have chosen to open:

arduino-1.0.2-windows.zip

which is a: Compressed (zipped) Folder (91.2 MB)
from: http://arduino.googlecode.com

What should Firefox do with this file?

○ Open with Windows Explorer (default)
● Save File
□ Do this automatically for files like this from now on.

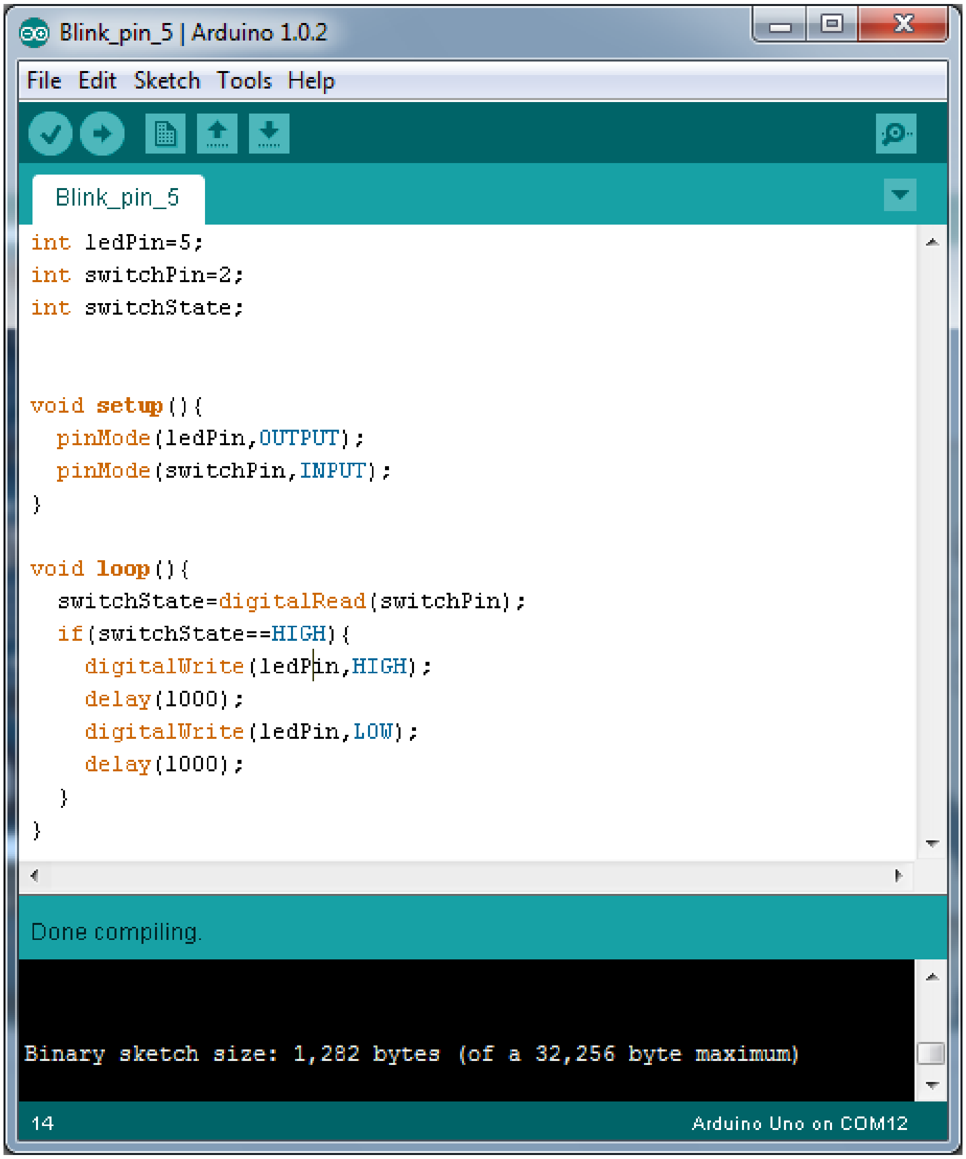
Download

Arduino 1.0.2 (release notes), hosted by Google Code:

Figure 1: Screenshot of download website.

2. Go to the getting started guides at this link http://arduino.cc/en/Guide/HomePage and follow the instructions to download and configure the IDE for your computer. Make sure to run the example provided called blink to make sure that you have configured everything correctly.

First Program:
Now you are going to write your first program which will blink an LED on pin 5 when pin 2 reads 5 volts and will not blink when it read 0 volts. This illustrates the basics of digital read, write and setting a pin. Figure 2 is a screenshot of the code and then a description of each line.



```
int ledPin=5;
int switchPin=2;
int switchState;


void setup(){
  pinMode(ledPin,OUTPUT);
  pinMode(switchPin,INPUT);
}

void loop(){
  switchState=digitalRead(switchPin);
  if(switchState==HIGH){
    digitalWrite(ledPin,HIGH);
    delay(1000);
    digitalWrite(ledPin,LOW);
    delay(1000);
  }
}
```

```
Done compiling.

Binary sketch size: 1,282 bytes (of a 32,256 byte maximum)
```

Figure 2: Code for first program

1. Create a new sketch by clicking on File->New. This should cause a new window with a new sketch to pop up.

2. Declare a new integer variable by typing “int ledPin = 5;”. This line declares a new variable ledPin of integer type equal to 5. The semicolon at the end of the statement is important you must end most lines not including conditional statements with

semicolons. Otherwise the code will not compile. After this you should declare a pin number for the switch("int switchPin= 2;"), and declare a variable to store the value read from pin 2 into ("int switchState;").

3. Now declare a new function called setup (). This is done by adding in a new line "void setup() {". This line declares the function setup which is a void function meaning it does not return a value. The setup function is important to any arduino code it is an initialization function built into the available libraries. The open squiggly bracket declares the beginning of the code inside the setup function. End the setup function with a closed squiggly bracket "}".

4. Inside the squiggly bracket, type "pinMode(ledPin, OUTPUT);". This function declares the pin numbered at the value ledPin to an output pin. (Remember to include the semicolon).

5. After the closed squiggly bracket on a new line start a 1oop function. This should be written like this "void loop() {". The loop function is essentially an infinite while loop which the uno executes continuously. End the loop function with a closed squiggly bracket "}".

6. Inside the brackets write a new line, "switchState=digitalRead(switchPin); ". This has read the state of pin numbered by the variable passed in the argument(whether it is LOW or HIGH) into the switchState variable.

7. Next is a conditional if statement written as "if(switchState==HIGH){". This evaluates true if the switchState value is HIGH and will execute any code inside the if's squiggly brackets. Be sure to close the if statement with squiggly brackets

8. On the next line write "digitalWrite(ledPin , HIGH); ". This line sets the digital pin at ledPin to high state.

9. Then on a new line inside the if statement, "delay(1000);". This delays the operation of the microcontroller by 1 second or 1000 milliseconds.

10. Still inside the if function write "digitalWrite(ledPin , LOW); ". This line sets the digital pin at ledPin to low.

11. Then on a new line inside the loop function, "delay(1000);". This delays the operation of the microcontroller again.

12. You have finished writing your first program. Now it is time to build the circuit before uploading the code to the uno.

Build the Circuit:

The circuit we are building will include an LED and a current limiting resistor (3.3 kOhm color code orange, orange, red) tied to 5 volts and pin 5. It also contains a 10kOhm potentiometer where the center tap is connected to pin 2 and the end taps are respectively connected to 5 volts and ground. When pin 5 goes low the voltage across the LED and resistor will be about 5 volts allowing current to flow and the LED to light. When pin 5 goes high the voltage across the LED and resistor will be about 0 volts causing no current to flow. The resistor is important as the current through a diode is related to the exponential of the voltage across the diode. If we biased the LED by 5 volts alone then it would cause extremely large current to flow and possibly destroy the Uno as well as the diode. It is important to consider when connecting circuits to the Uno how much current is being sourced or sunk. Figure 3 is a basic circuit diagram of what is being built.

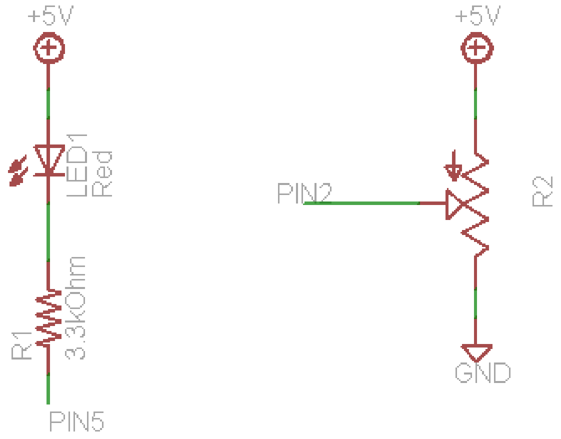


Figure 3: Circuit Diagram for First Circuit

Things to note when building this circuit.

1. The longer lead on the LED is usually the anode on the side the higher voltage is placed. The polarity of the diode can also be determined by which side has a flattened edge this is the cathode and should be oriented towards the lower voltage side of the component.

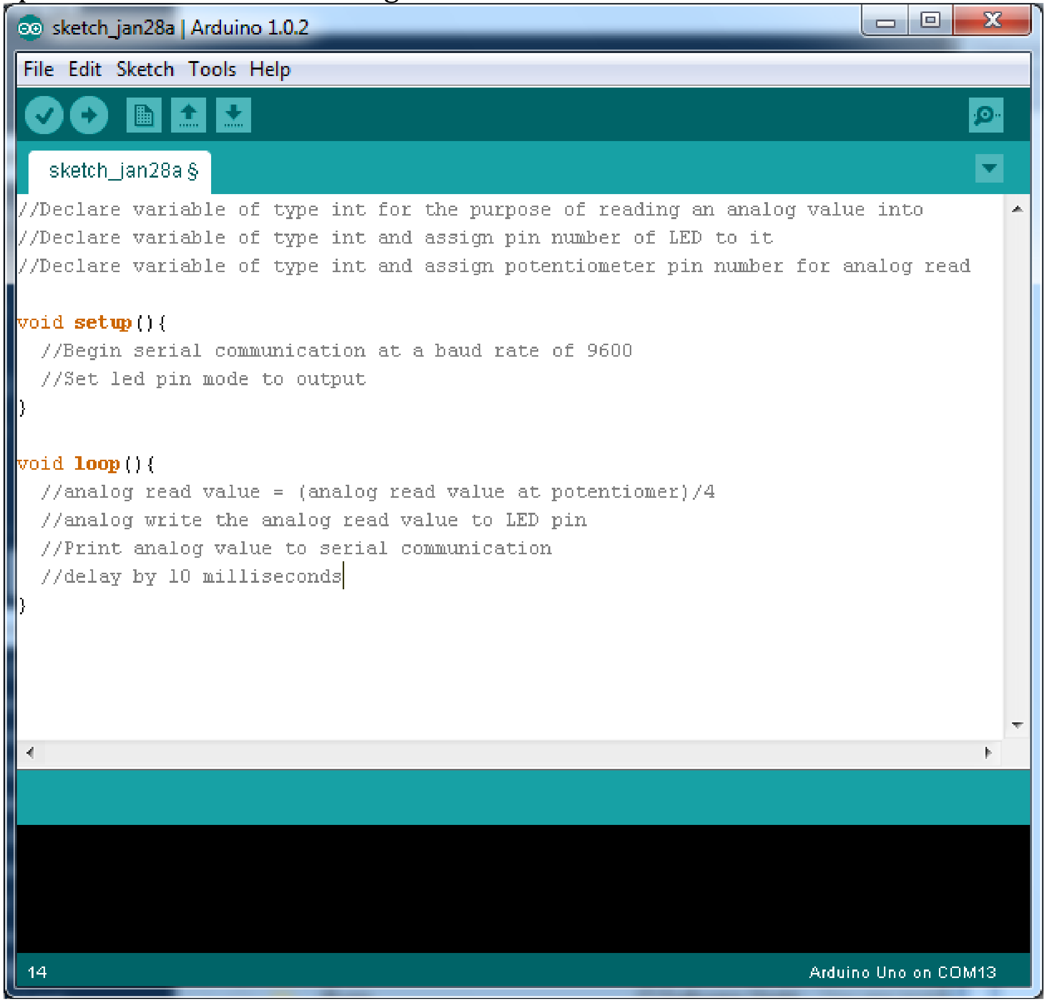
2. One can get 5 volts from the arduino board in the appropriately labeled pin.

3. It may be easiest to build this circuit by using a wire to connect 5 volts to your bread board and then another wire to connect pin 5 to a different portion of your bread board.

Ask an instructor to look at your circuit to confirm it is configured correctly. Then upload the sketch to your Uno and see if the circuit and program work as intended.

Dimming Circuit Program:

In this second program and circuit you will change the brightness of an LED by reading the voltage at an input pin. When the input pin (analog input pin 0) is close to 5 volts the LED will be at its dimmest, while when close to 0 volts the LED will be at its brightest. You will also write the analog value being used to perform an analog write to a serial port communication port between the Uno and your computer. One way to accomplish this is with the pseudocode as shown in Figure 4.



```
//Declare variable of type int for the purpose of reading an analog value into
//Declare variable of type int and assign pin number of LED to it
//Declare variable of type int and assign potentiometer pin number for analog read

void setup(){
  //Begin serial communication at a baud rate of 9600
  //Set led pin mode to output
}

void loop(){
  //analog read value = (analog read value at potentiomer)/4
  //analog write the analog read value to LED pin
  //Print analog value to serial communication
  //delay by 10 milliseconds
}
```

Figure 4: Psuedocode for Dimming Circuit Program

Here is a description of the new functions used in implementing this pseudocode. For more thorough descriptions of these functions and all Arduino functions consult this link http://arduino.cc/en/Reference/HomePage .

void **Serial.begin**(baud rate): Sets the data rate between the Uno and what it is communicating with.

void **Serial.print**(value): Prints string or value passed as an argument to serial comm.

void **Serial.println**(value): Prints string or value passed as an argument to serial comm. Followed by a new line.

integer **analogRead**(pin number):Returns an integer value from 0 to 1023 based on the voltage from 0 to 5 from the pin number provided.

void **analogWrite**(pin number, value): Writes an analog value (using a pulse width modulated wave) to the pin number provided. Acceptable values for the analog write are from 0 to 255.

Build the Dimming Circuit:

The circuit will again have an LED and 3.3 kOhm resistor tied to whichever pin you used with the capacity to analog write, these are the digital pins 3, 5, 6, 9, 10, 11 with a squiggly line/tilde symbol next to them. The circuit also requires a 10kOhm potentiometer. The center tap is connected to the analog read pin designated in the code, while the end pins of the potentiometer will be connected to 5 volts and ground. Figure 5 is a basic circuit diagram of the circuit being built.

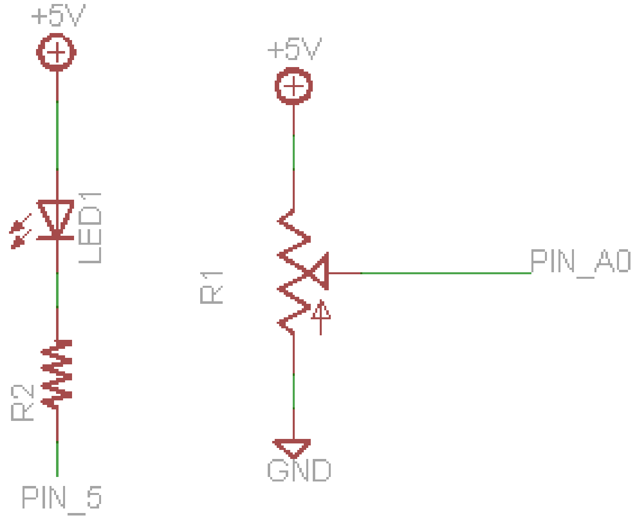


Figure 5: Dimming Circuit Schematic

After building the circuit, upload your code and see if it functions properly. Open a serial monitor by pressing the magnifying glass in the right corner. Shown on the next page in Figure 6.

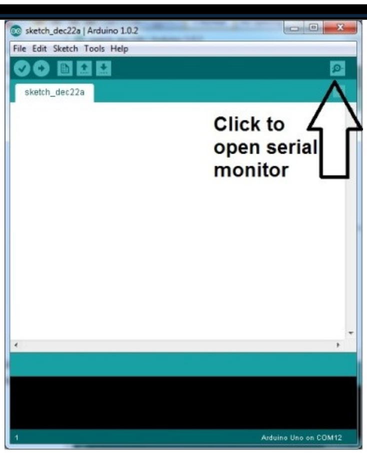

sketch_dec22a | Arduino 1.0.2
File Edit Sketch Tools Help
sketch_dec22a
Click to open serial monitor
1
Arduino Uno on COM12

Figure 6: Location of Serial Monitor Button

Vary the potentiometer to show that the LED changes in brightness and the values displayed in the serial monitor change.

Quiz Questions:

1. setup() serves which of these purposes?
   a) It is a function inside which code is continually looped through until the board is reset or reprogrammed.
   b) It is a function called at the beginning of the sketch and will run once after powerup or reset and is
   c) It is a function that begins serial communication between the Uno and computer.
   d) It is a function that frames its best friend function for a crime in an attempt to distract the authorities.

2. pinMode() affects whether a pin is designated for output or input?
   a) True
   b) False

3. analogRead() produces integer values from _________ to _________.

4. analogWrite() can be passed integer values from _________ to _________.

5. Building a circuit, while the power supply is on and attached to the board is a safe practice?
   a) True
   b) False