Micro . Computer System Lab.

Arduino Overview

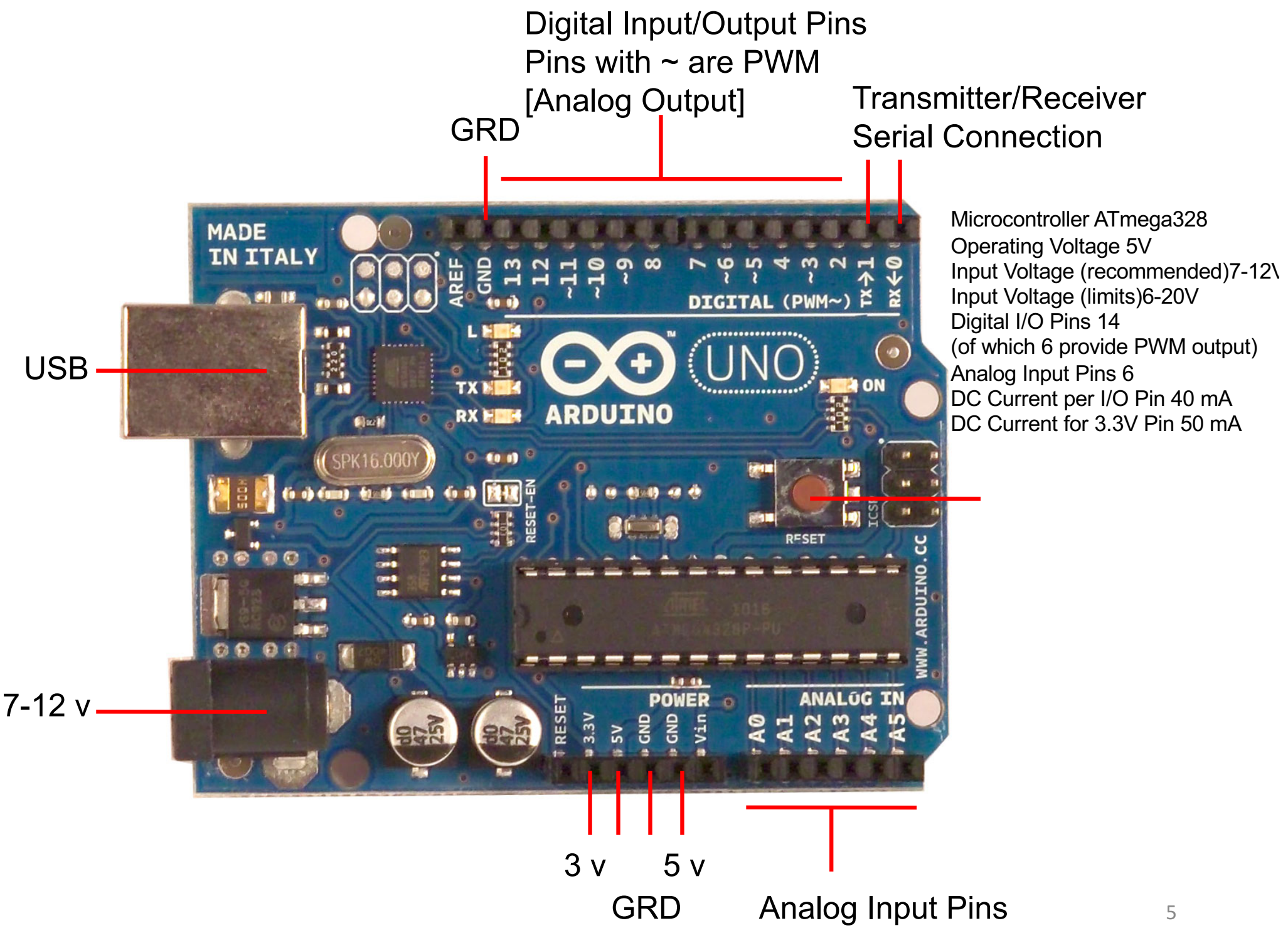
Outline

- Arduino Hello World (Blink)
- DFRduino UNO R3/Zero Pro
- More about Arduino

Noted

- Slides are adapted and modified from
  - Linz Craig, Nick Poole, Prashanta Aryal, Theo Simpson, Tai Johnson, and Eli Santistevan, “Intro to Arduino,” https://cdn.sparkfun.com/assets/3/9/d/9/e/Intro_to_Arduino_-_v30_1.pdf
  - http://isites.harvard.edu/fs/docs/icb.topic983682.files/Week%2008/Arduino%20Technical%20Session%201.ppt
  - http://www.ym.edu.tw/~yhkao/iedu/arduino_experiments.pdf

Arduino Hello World (Blink)

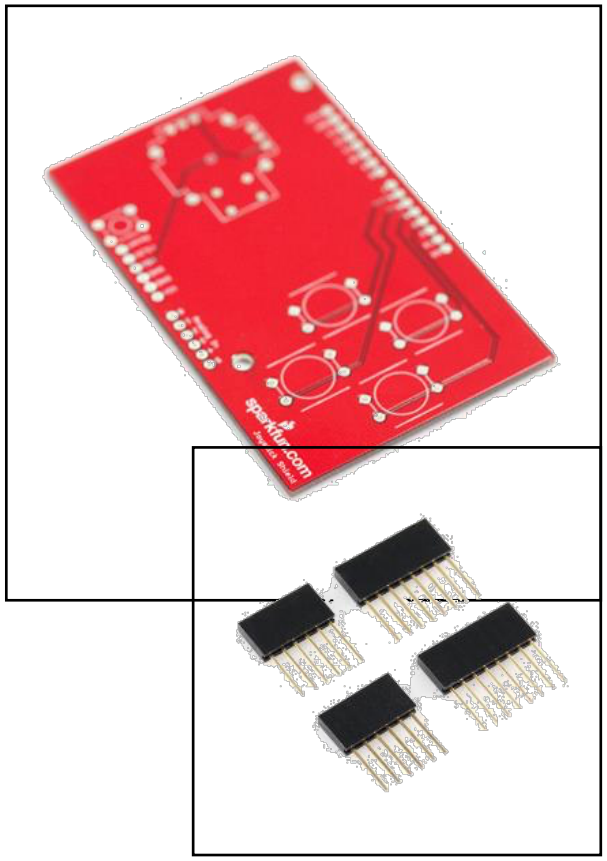

Digital Input/Output Pins
Pins with ~ are PWM
[Analog Output]
GRD
Transmitter/Receiver
Serial Connection
Microcontroller ATmega328
Operating Voltage 5V
Input Voltage (recommended)7-12V
Input Voltage (limits)6-20V
Digital I/O Pins 14
(of which 6 provide PWM output)
Analog Input Pins 6
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA
USB
7-12 v
3 v
5 v
GRD
Analog Input Pins
MADE IN ITALY
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
RESET
WWW.ARDUINO.CC

Why is Arduino so popular?

- Open source
  - Hardware (Arduino I/O board
  - Software (Arduino IDE)
- Fast prototyping
- Flexible and easy to use
  - Artists, designers, …

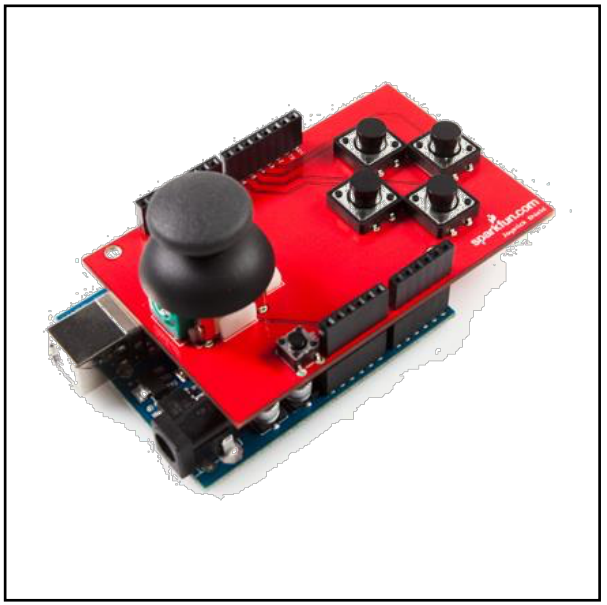
Arduino Shields

PCB



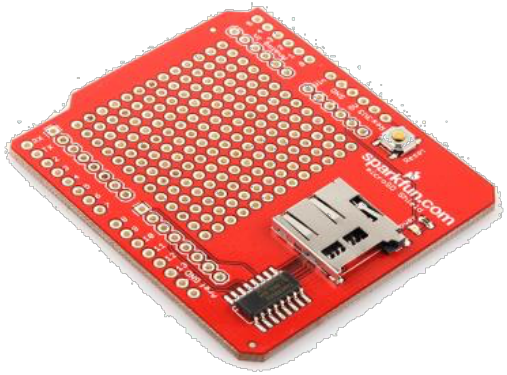
Built Shield

Inserted Shield

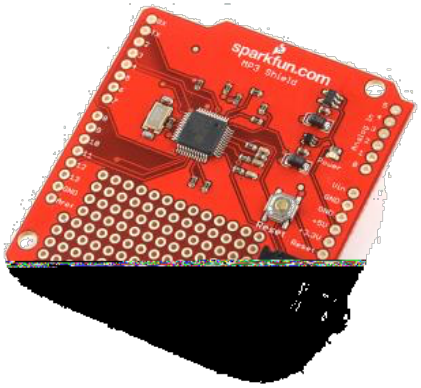


Arduino Shields

Micro SD

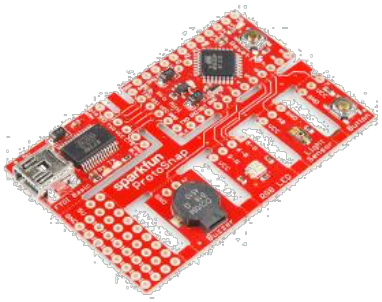


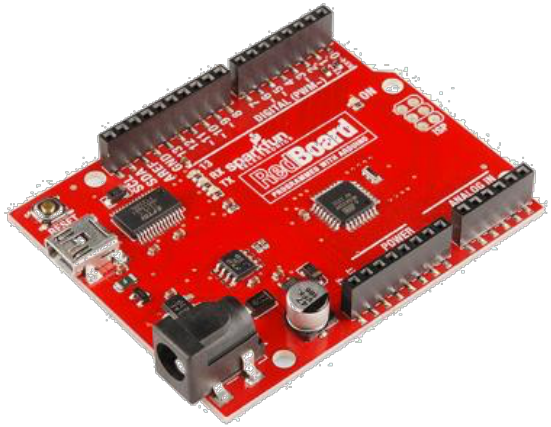
MP3 Trigger

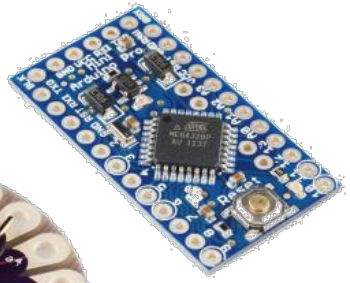


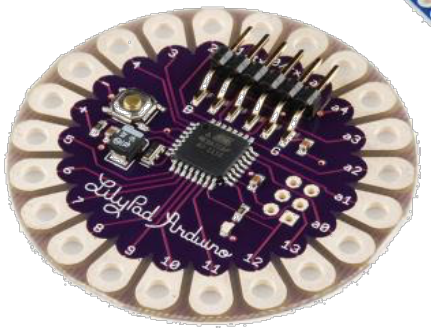
LCD

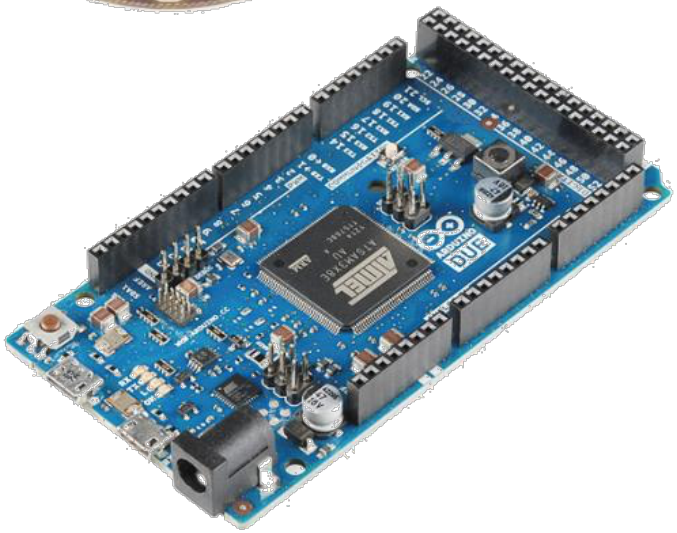
Arduino & Arduino Compatible Boards









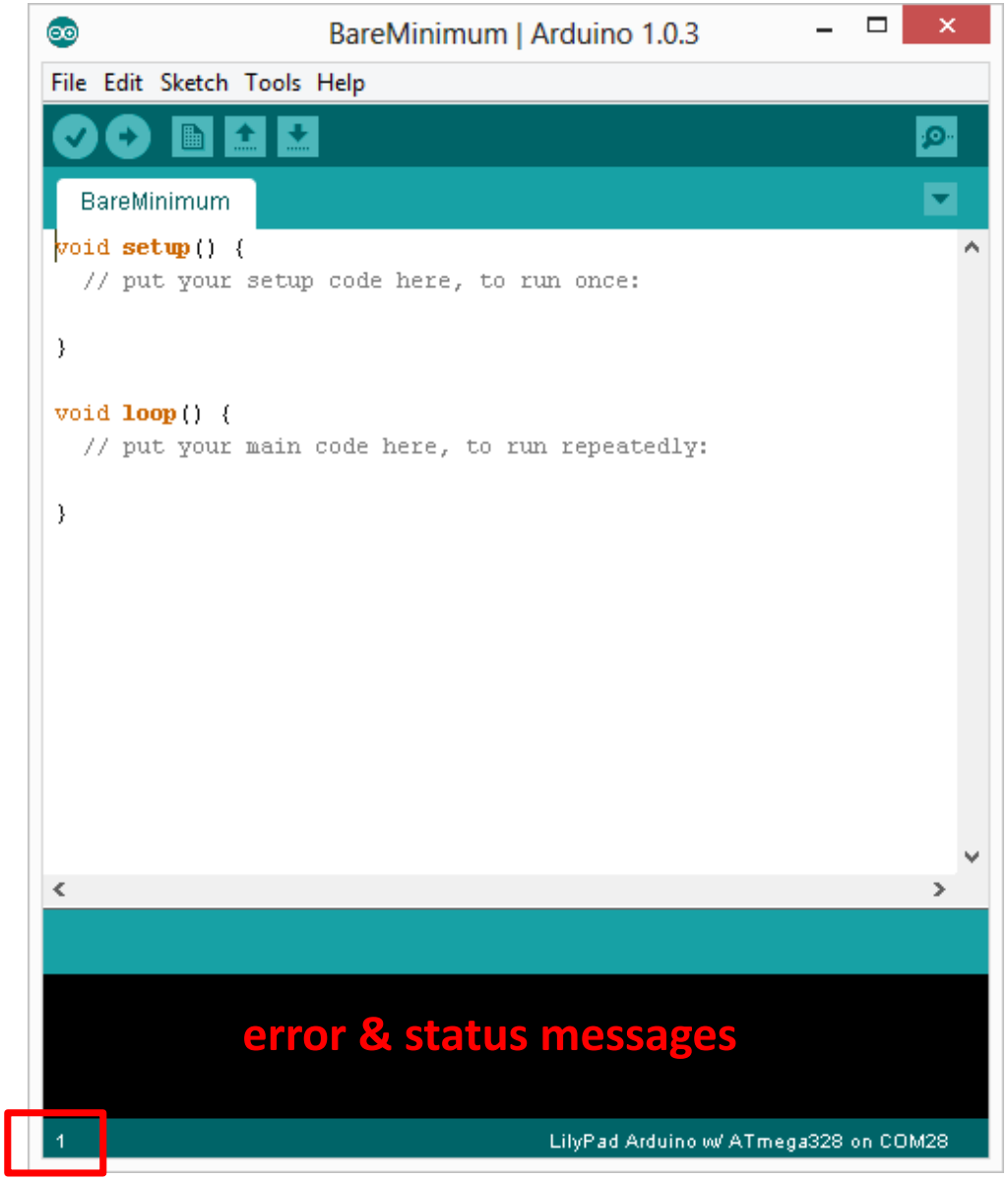


SIK Components

Name	Image	Type	Function	Notes
Push Button		Digital Input	Switch - Closes or opens circuit	Polarized, needs resistor
Trim potentiometer		Analog Input	Variable resistor	Also called a Trimpot.
Photoresistor		Analog Input	Light Dependent Resistor (LDR)	Resistance varies with light.
Relay		Digital Output	Switch driven by a small signal	Used to control larger voltages
Temp Sensor		Analog Input	Temp Dependent Resistor	
Flex Sensor		Analog Input	Variable resistor	
Soft Trimpot		Analog Input	Variable resistor	Careful of shorts
RGB LED		Dig & Analog Output	16,777,216 different colors	Ooh... So pretty.

Arduino

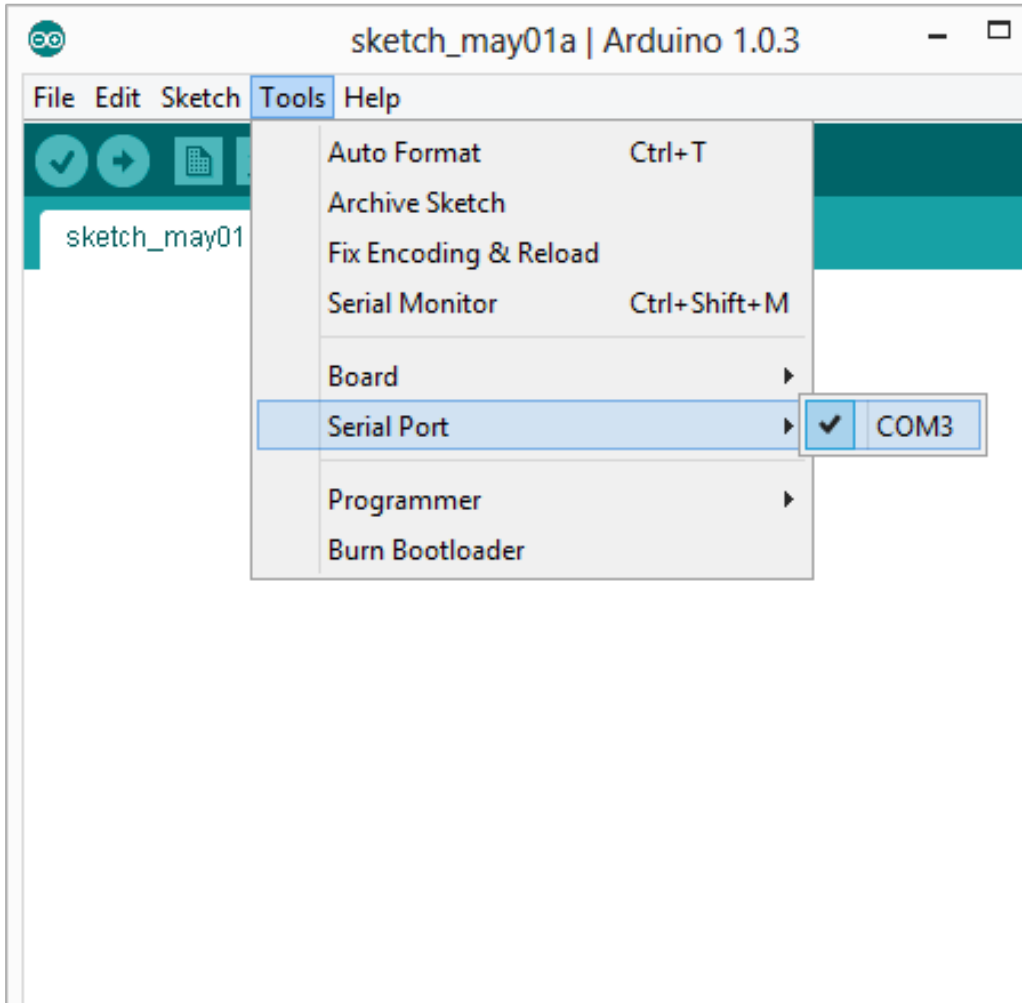
Integrated Development Environment (IDE)



Two required functions / methods / routines:

```
void setup()
{
      // runs once
}

void loop()
{
      // repeats
}
```

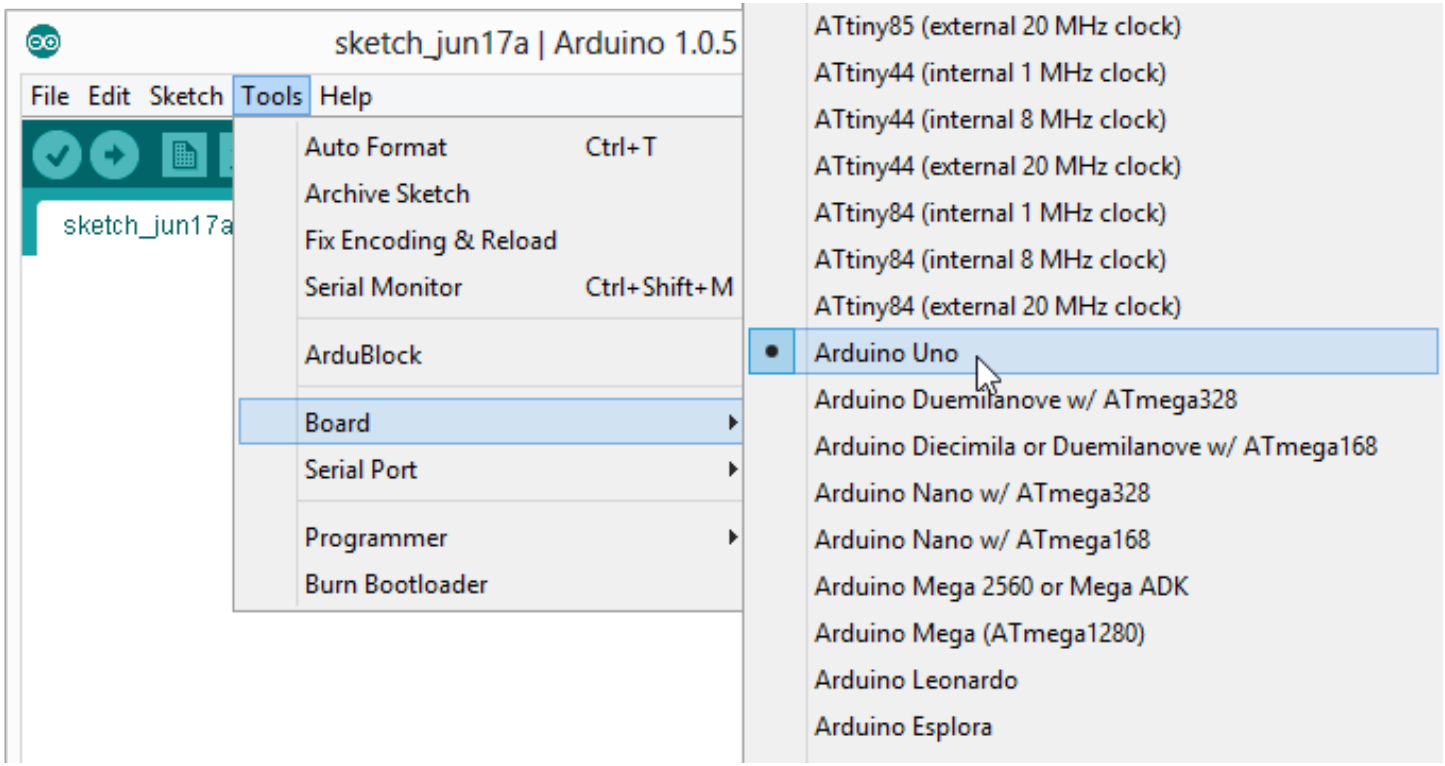
Settings: Tools → Serial Port



Your computer communicates to the Arduino microcontroller via a serial port → through a USB-Serial adapter.

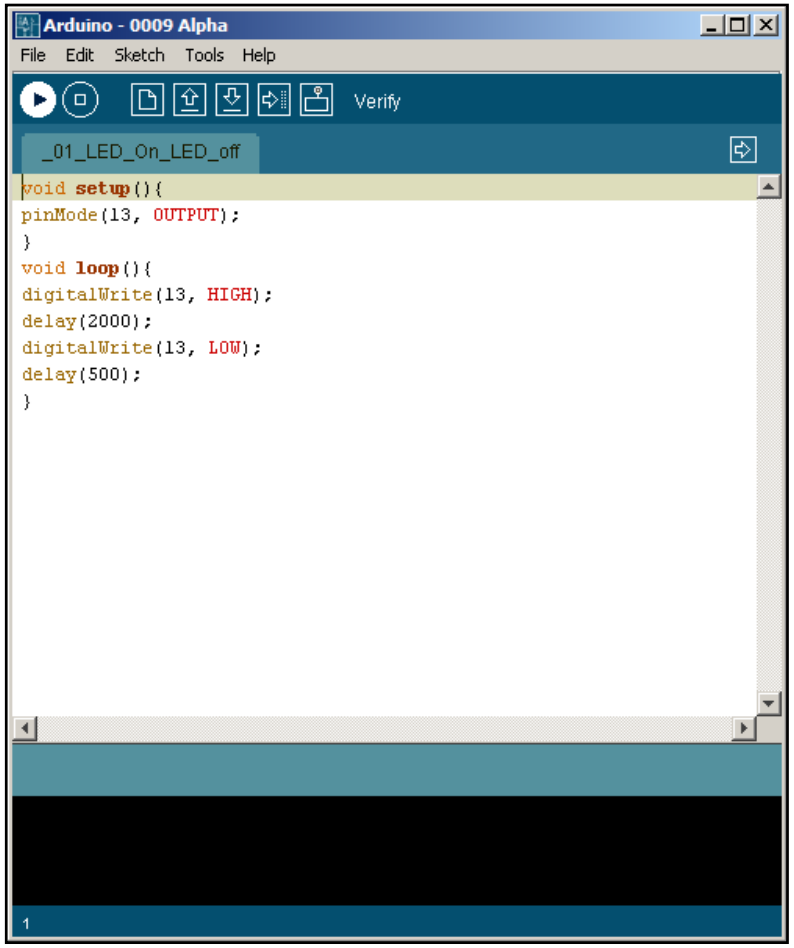
Check to make sure that the drivers are properly installed.

Settings: Tools → Board



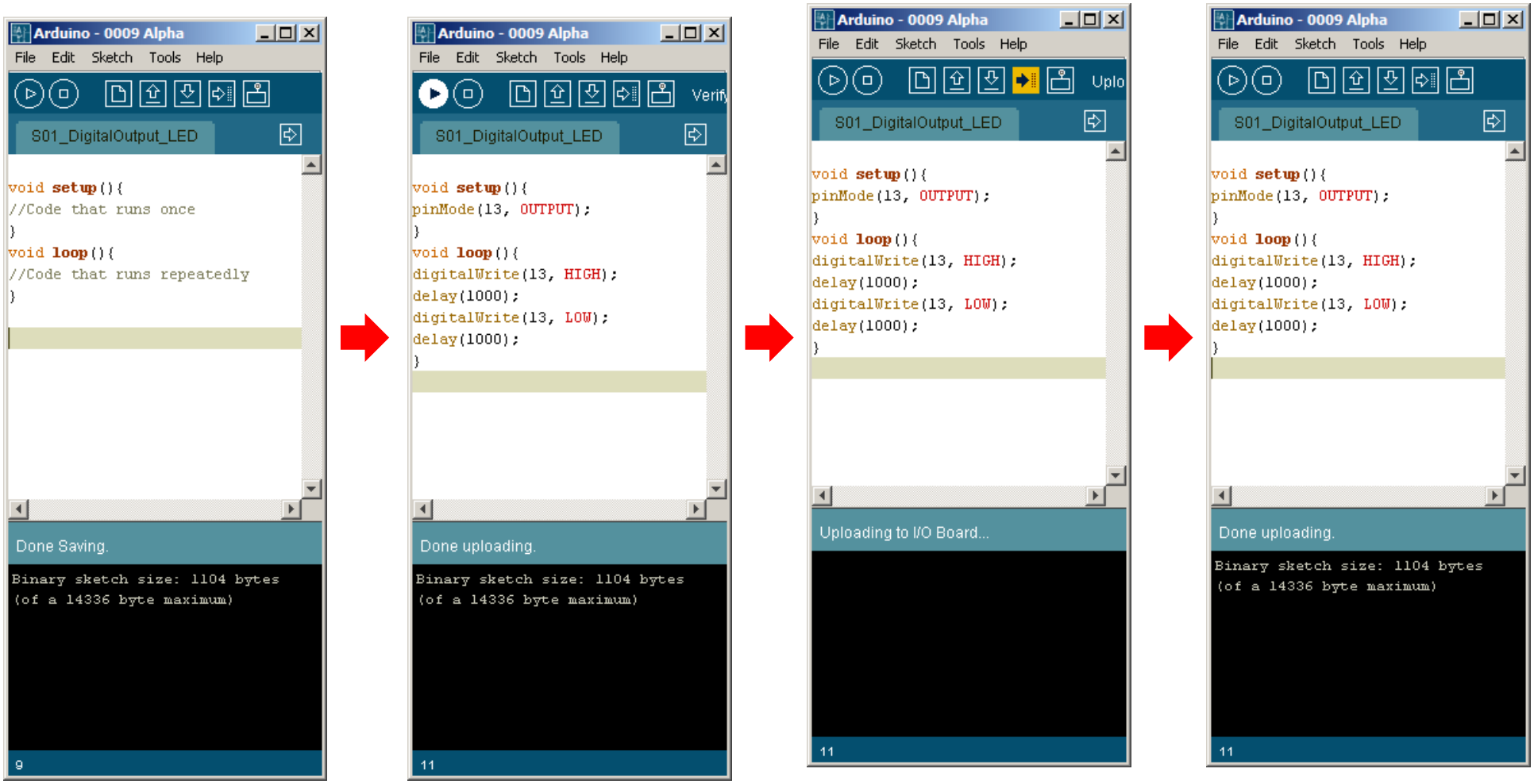
Next, double-check that the proper board is selected under the Tools→Board menu.

Arduino-Digital Output-LED

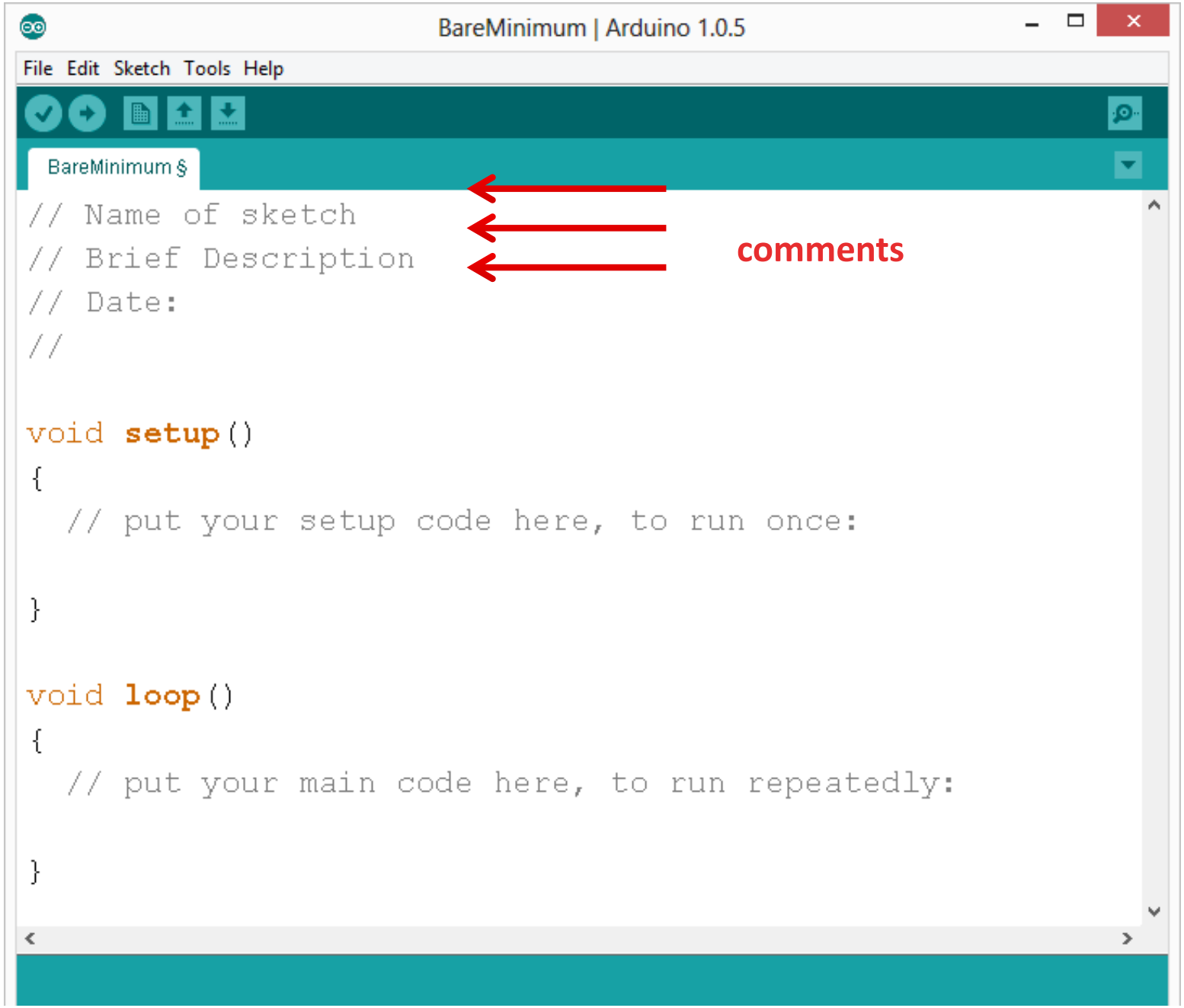



```
void setup(){
pinMode(13, OUTPUT);
}
void loop(){
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

Arduino-Compiling and Uploading Code



1. Write the code
2. Compile the code
3. Check Arduino Port Connection
4. Upload the Code
5. The Arduino and Connected Circuits start to show behavior based on the uploaded code


BareMinimum | Arduino 1.0.5
File Edit Sketch Tools Help
BareMinimum §
// Name of sketch
// Brief Description
// Date:
//
void setup()
{
// put your setup code here, to run once:
}
void loop()
{
// put your main code here, to run repeatedly:
}
comments

Three commands to know…

```
pinMode(pin, INPUT/OUTPUT);
  ex: pinMode(13, OUTPUT);

digitalWrite(pin, HIGH/LOW);
  ex: digitalWrite(13, HIGH);

delay(time_ms);
  ex: delay(2500); // delay of 2.5 sec.

// NOTE: -> commands are CASE-sensitive
```

Project #1: Wiring Diagram

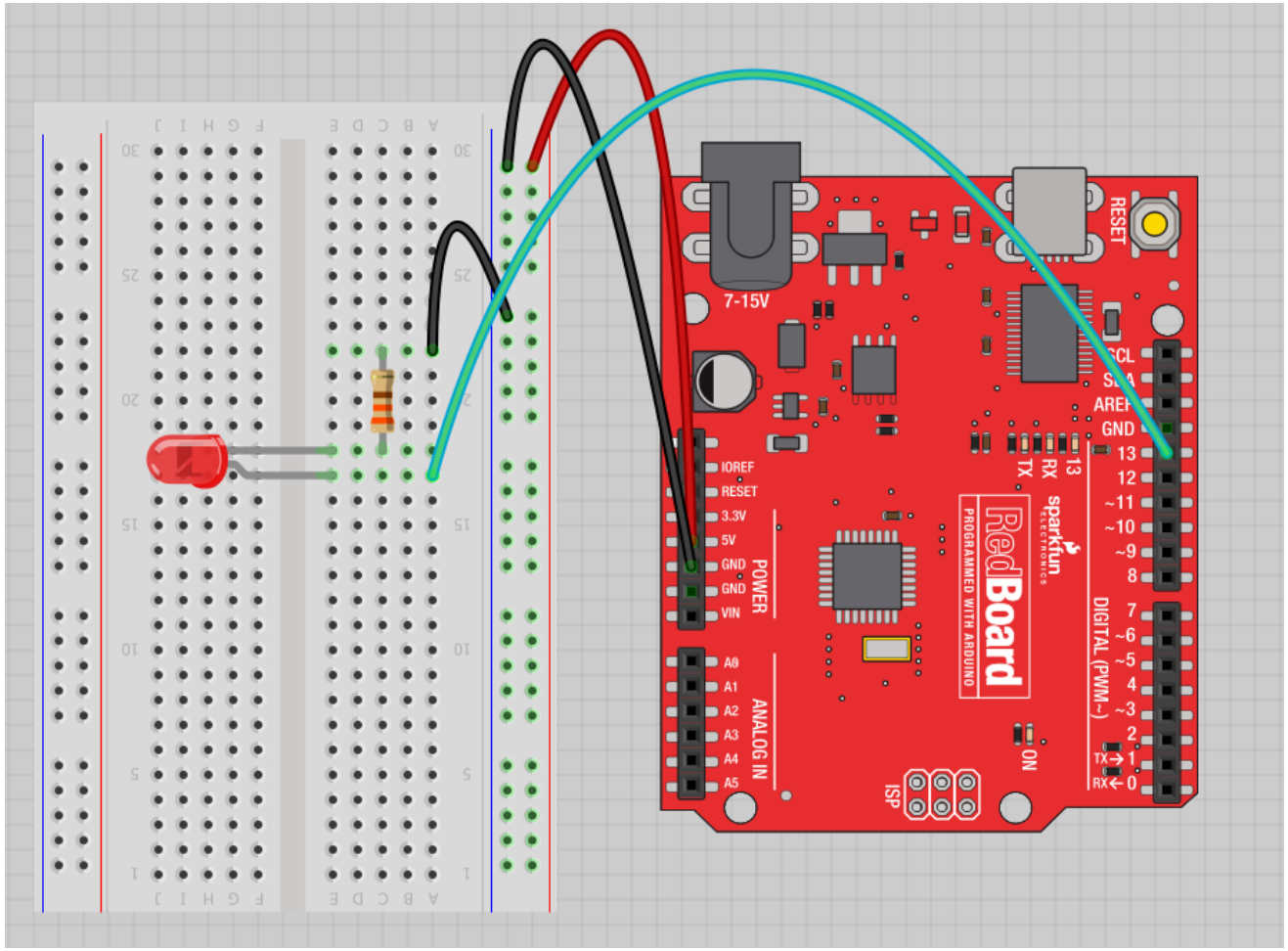
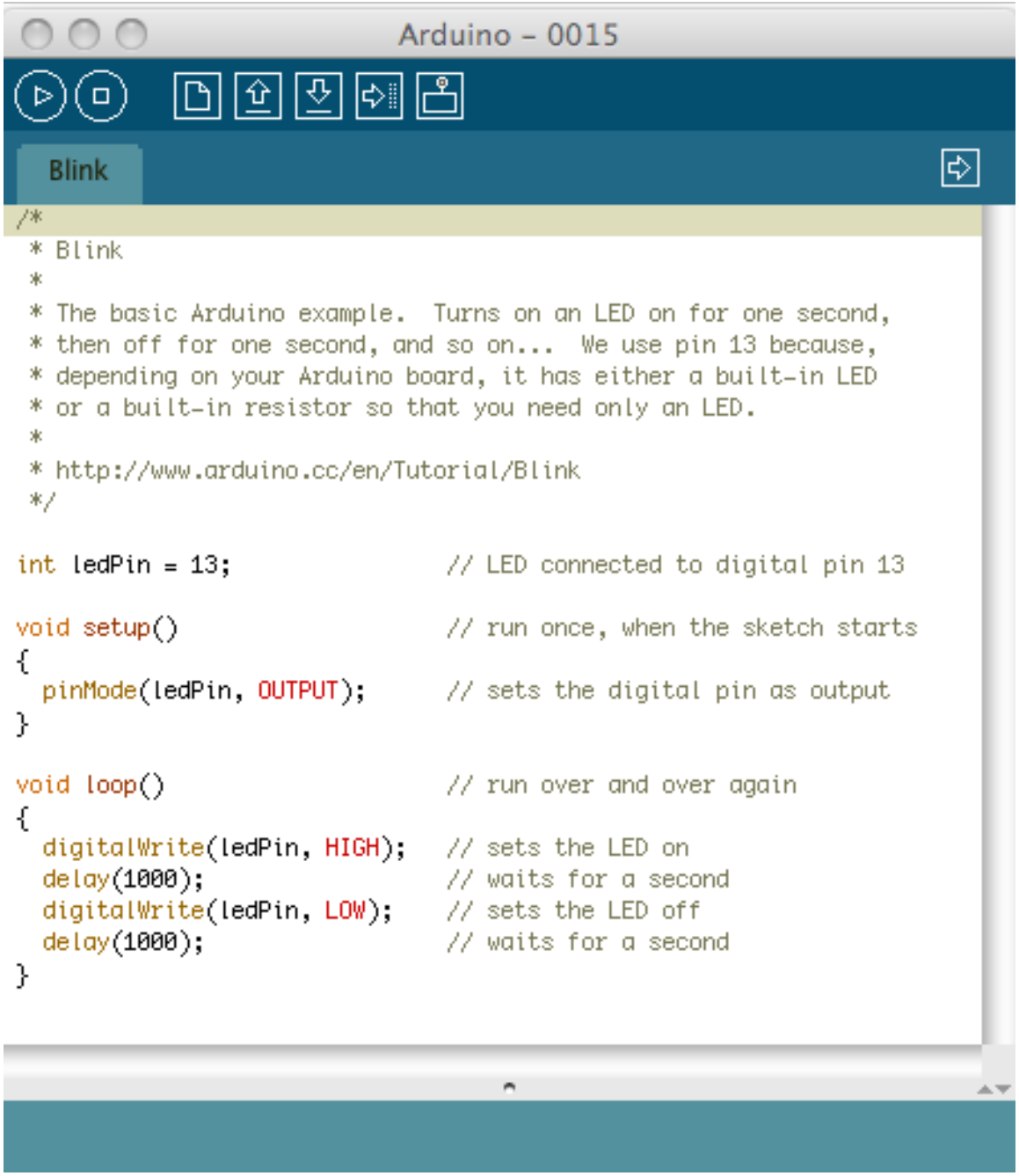


Image created in Fritzing

Move the green wire from the power bus to pin 13 (or any other Digital I/O pin on the Arduino board.


Arduino - 0015
Blink
/*
* Blink
*
* The basic Arduino example. Turns on an LED on for one second,
* then off for one second, and so on... We use pin 13 because,
* depending on your Arduino board, it has either a built-in LED
* or a built-in resistor so that you need only an LED.
*
* http://www.arduino.cc/en/Tutorial/Blink
*/
int ledPin = 13; // LED connected to digital pin 13
void setup() // run once, when the sketch starts
{
pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop() // run over and over again
{
digitalWrite(ledPin, HIGH); // sets the LED on
delay(1000); // waits for a second
digitalWrite(ledPin, LOW); // sets the LED off
delay(1000); // waits for a second
}

A few simple challenges
Let's make LED#13 blink!

Challenge 1a – blink with a 200 ms second interval.

Challenge 1b – blink to mimic a heartbeat

Challenge 1c – find the fastest blink that the human eye can still detect…

1 ms delay? 2 ms delay? 3 ms delay???

Try adding other LEDs

Can you blink two, three, or four LEDs?

(Hint: Each LED will need it's own 330Ω resistor.)

Generate your own morse code flashing

How about → Knight Rider? Disco? Police Light?

More about Arduino

Programming Concepts: Variables

ProtosnapProMiniExample2 §

```
// Comments go here
// Written by:  Joesephine Jones
// Date:  April 12, 2013

int sensorValue;
int ledPin;

void setup()
{
  // put your setup code here, to run once:
  int setupVariable;

}

void loop()
{
  // put your main code here, to run repeatedly:
  int loopScopeVariable
}
```

Variable Scope

- *Global*
- ---
- *Function-level*

Fading in and Fading Out (Analog or Digital?)

A few pins on the Arduino allow for us to modify the output to mimic an analog signal.

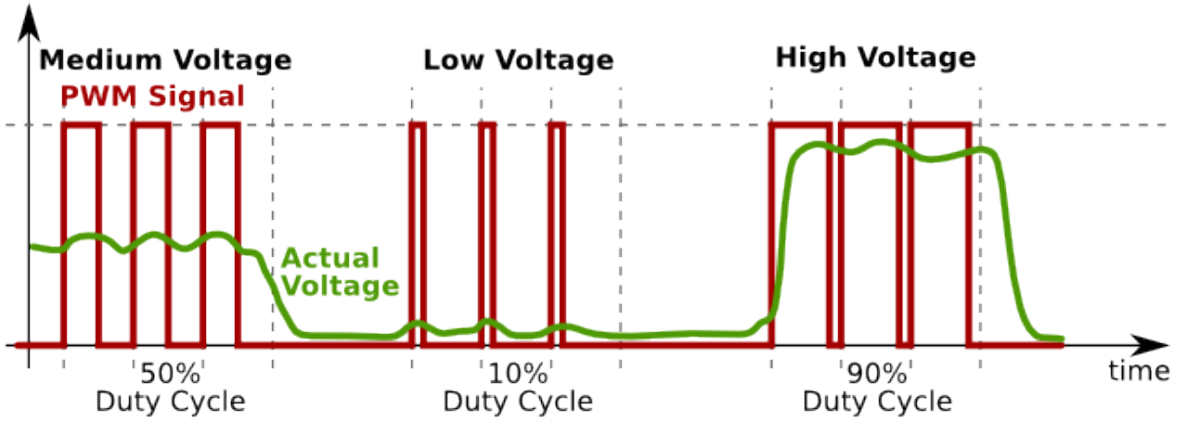
This is done by a technique called:

Pulse Width Modulation (PWM)

Concepts: Analog vs. Digital

•To create an analog signal, the microcontroller uses a technique called PWM. By varying the duty cycle, we can mimic an "average" analog voltage.

•Pulse Width Modulation (PWM)



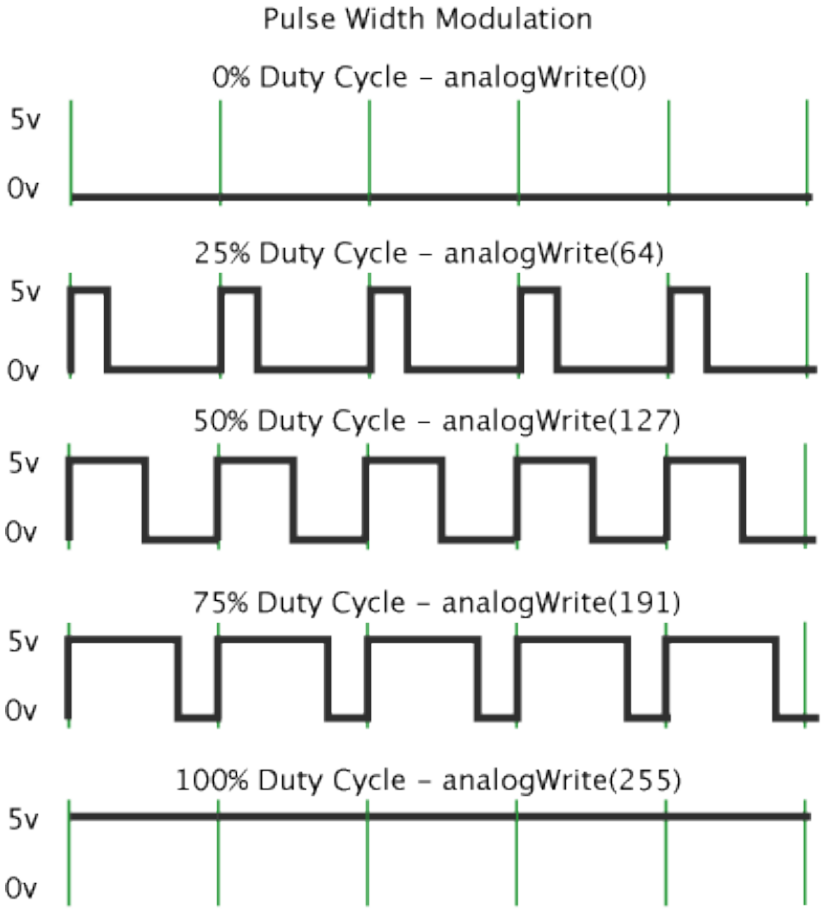
Project #2 – Fading
Introducing a new command…

```
analogWrite(pin, val);
```

pin – refers to the OUTPUT pin (limited to pins 3, 5, 6, 9, 10, 11.) – denoted by a ~ symbol

val – 8 bit value (0 – 255).

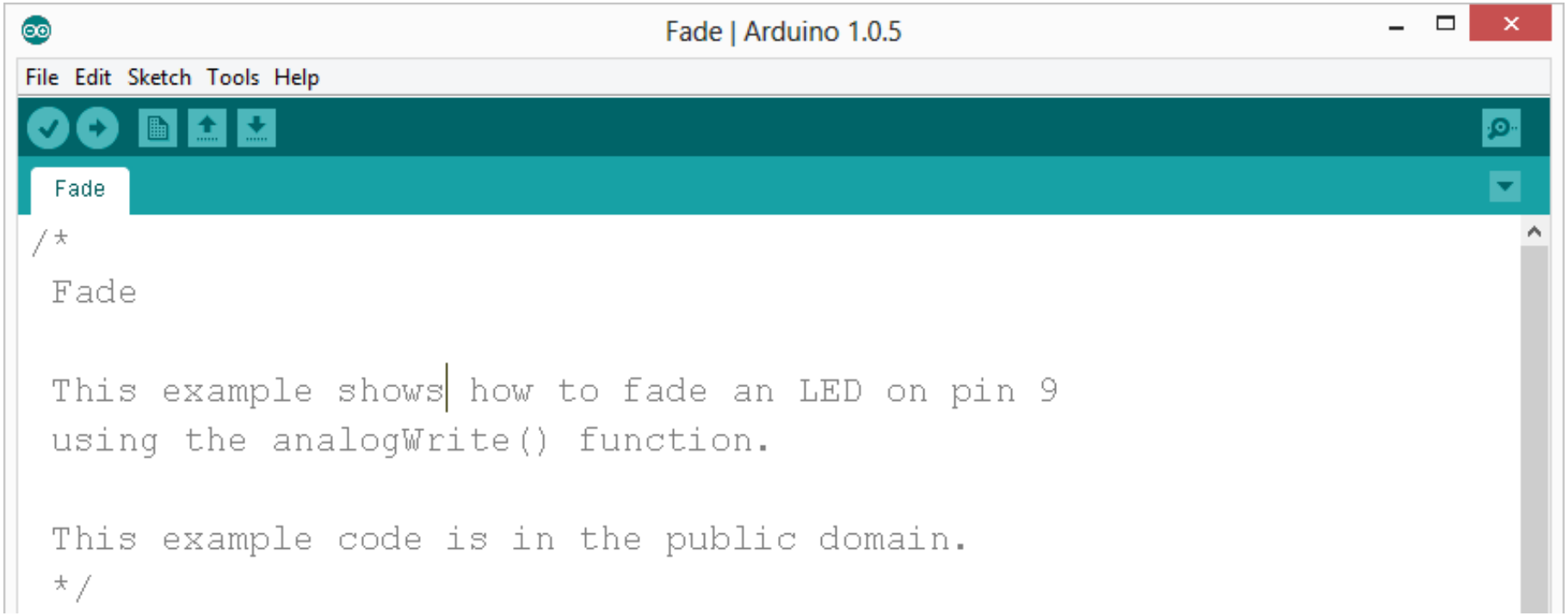
0 => 0V | 255 => 5V



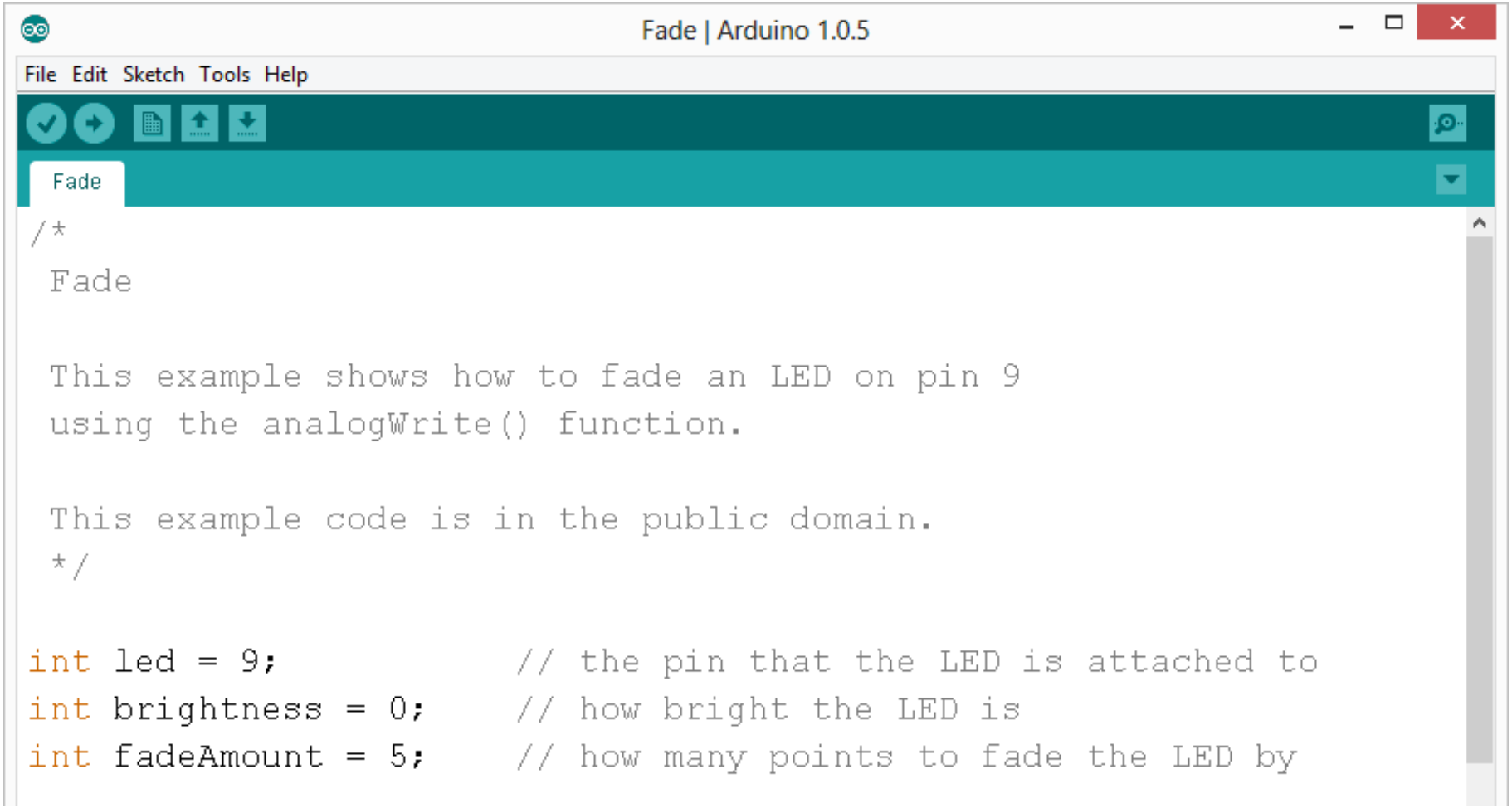
Move one of your LED pins over to Pin 9

In Arduino, open up:

File → Examples → 01.Basics → Fade



Fade - Code Review



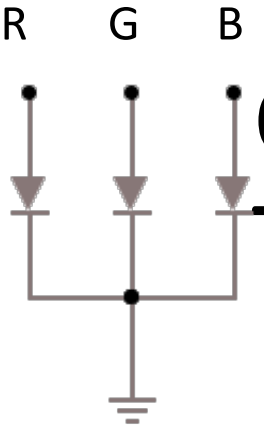
Fade - Code Review

```
void setup()  {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop()  {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

Project# 2 -- Fading

Challenge 2a – Change the rate of the fading in and out. There are at least two different ways to do this – can you figure them out?

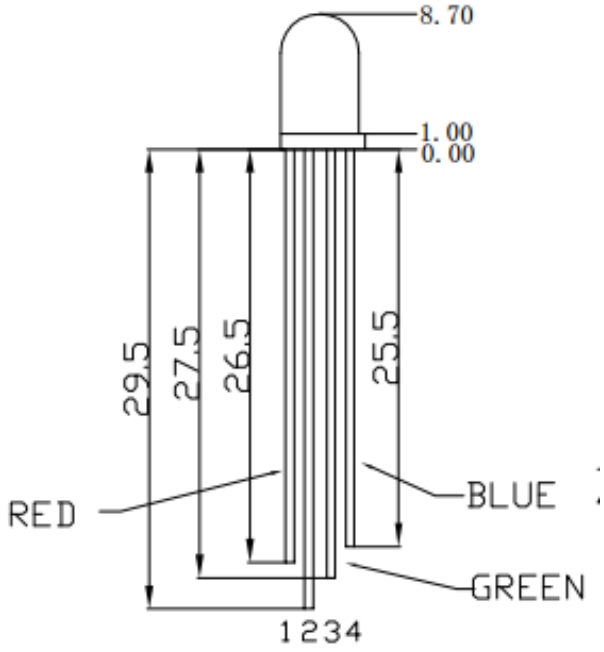
Challenge 2b – Use 2 (or more) LEDs – so that one fades in as the other one fades out.



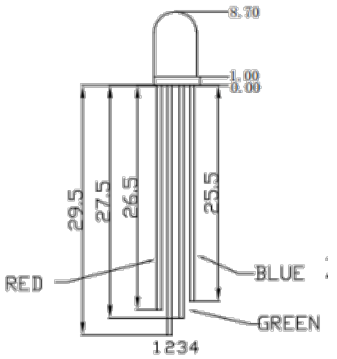
Color Mixing
Tri-color LED

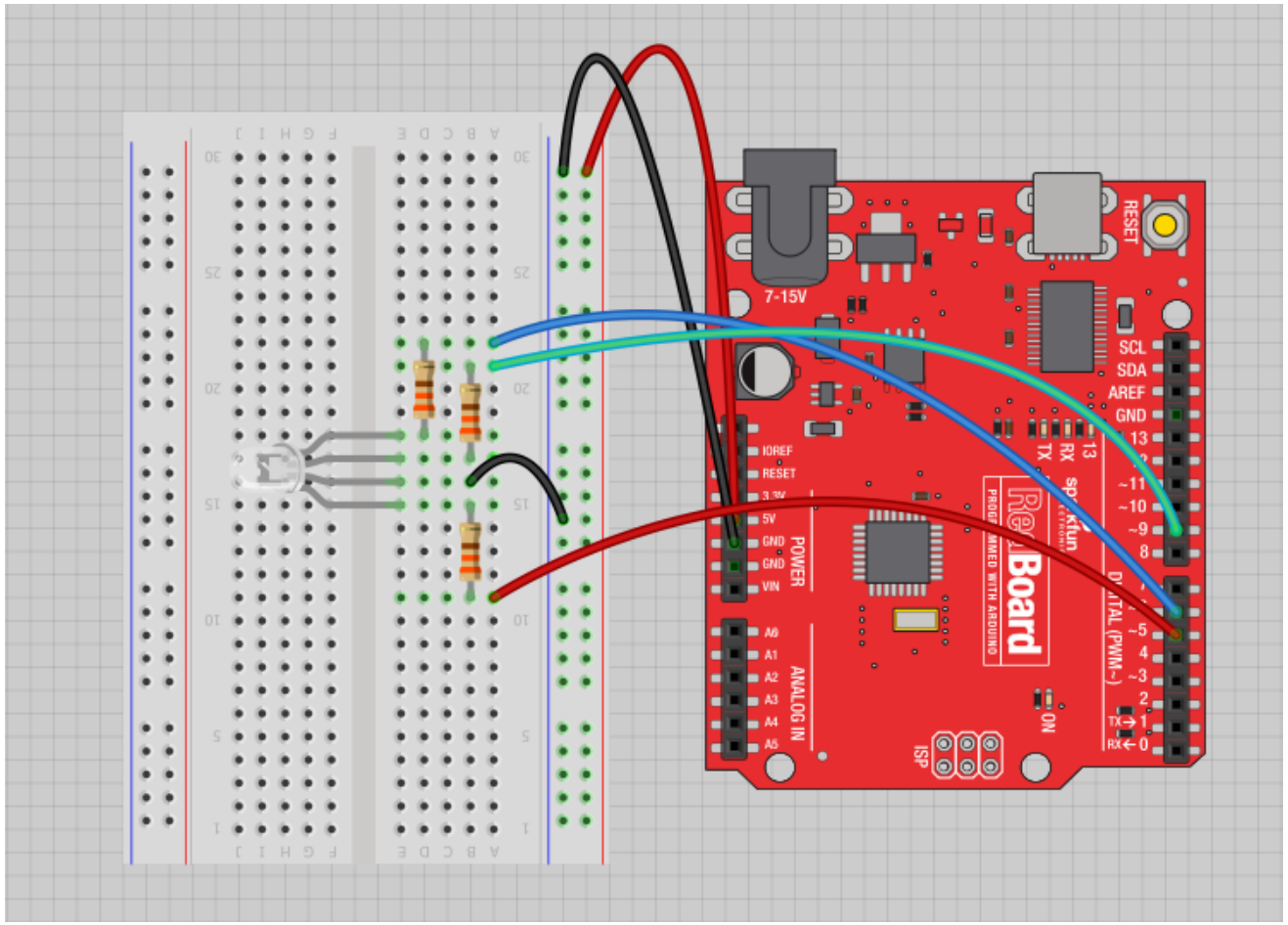
In the SIK, this is a standard – Common Cathode LED

This means the negative side of the LED is all tied to Ground.



Project 3 – RGB LED



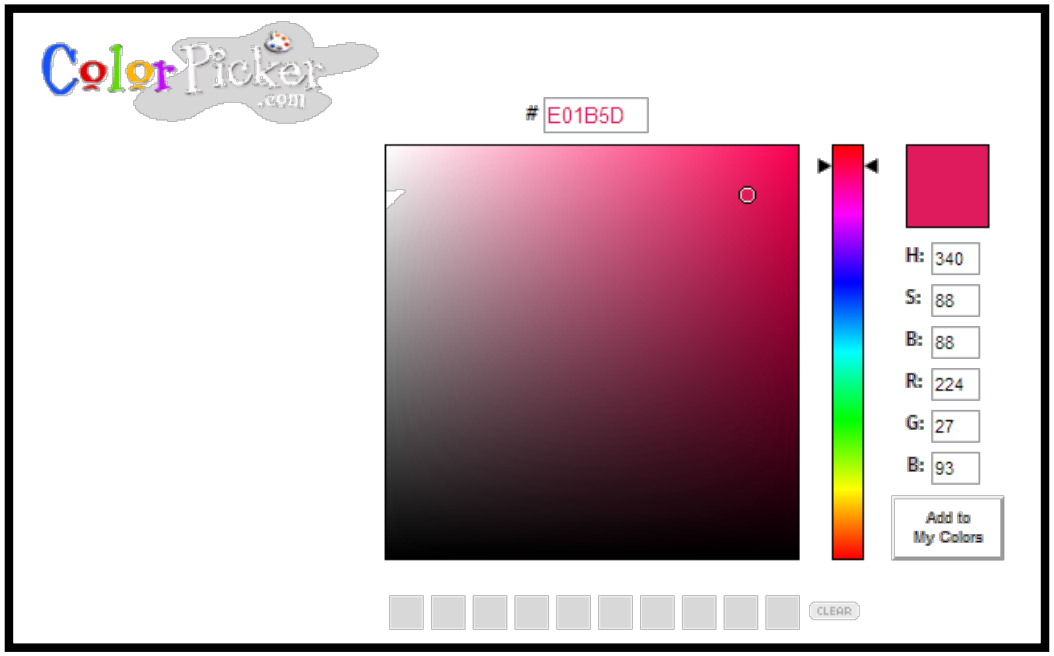


Note: The longest leg of the RGB LED is the Common Cathode. This goes to GND.

Use pins 5, 6, & 9

How many unique colors can you create?

$$\#\ of\ unique\ colors = 256 \cdot 256 \cdot 256$$
$$= 16{,}777{,}216\ colors!$$



Use Colorpicker.com or experiment on your own.

Pick out a few colors that you want to try re-creating for a lamp or lighting display...

Play around with this with the `analogWrite()` command.

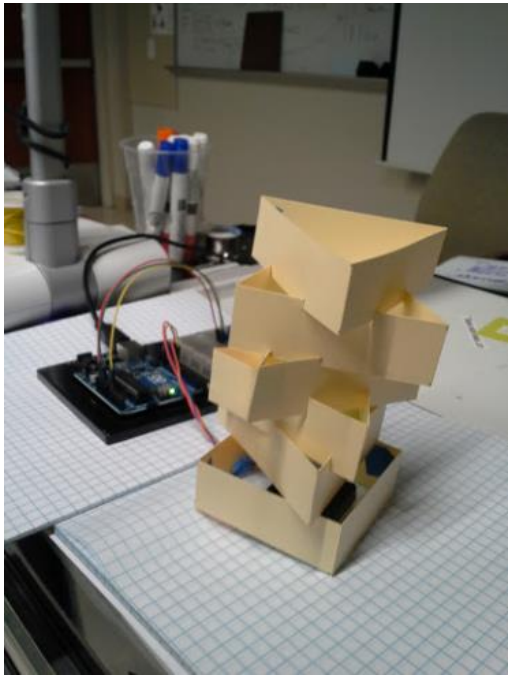
RGB LED Color Mixing

```
int redPin = 5;
int greenPin = 6;
int bluePin = 9;


void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
```

RGB LED Color Mixing

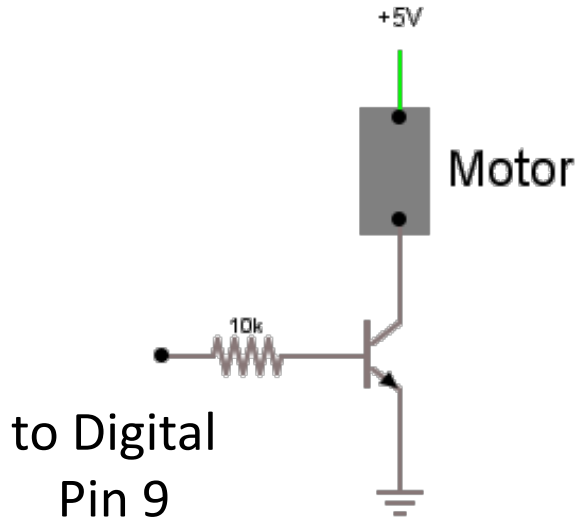
```
void loop()
{
  analogWrite(redPin, 255);
  analogWrite (greenPin, 255);
  analogWrite (bluePin, 255);
}
```

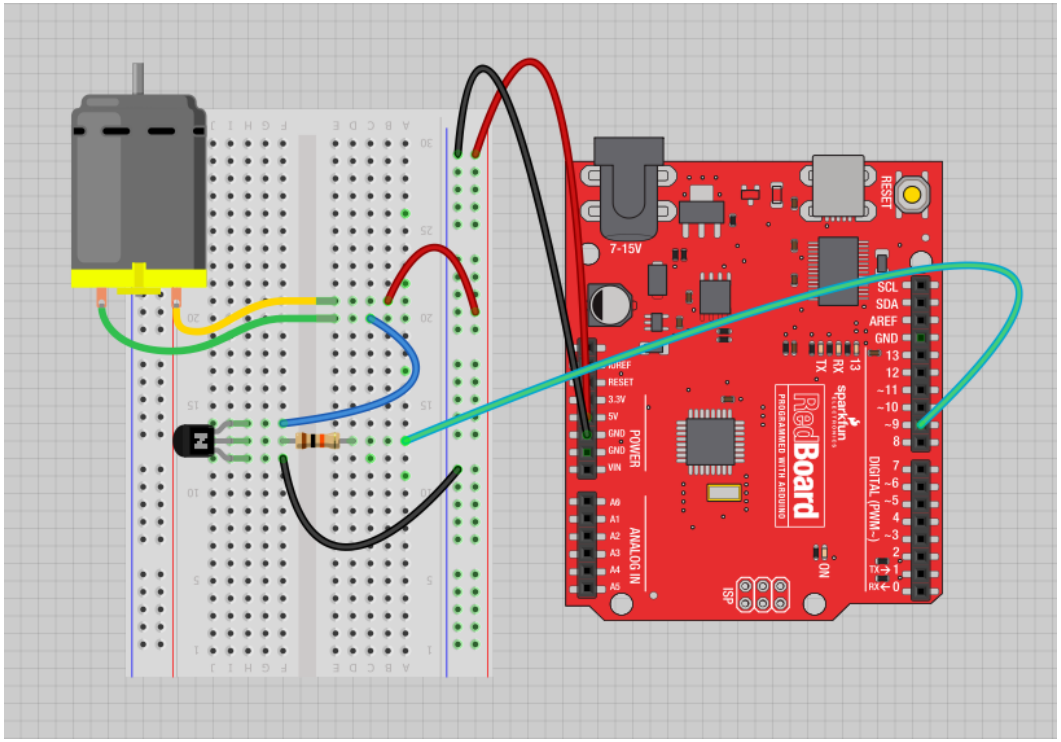
Project: Mood Lamp / Light Sculpture



Driving Motors or other High Current Loads

NPN Transistor (Common Emitter "Amplifier" Circuit)





Input

•Input is any signal entering an electrical system .

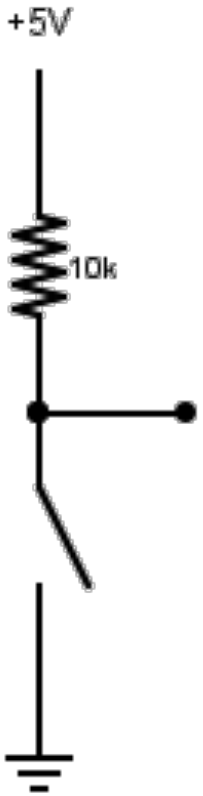
- Both digital and analog sensors are forms of input
- Input can also take many other forms: Keyboards, a mouse, infrared sensors, biometric sensors, or just plain voltage from a circuit

Project #4 – Digital Input

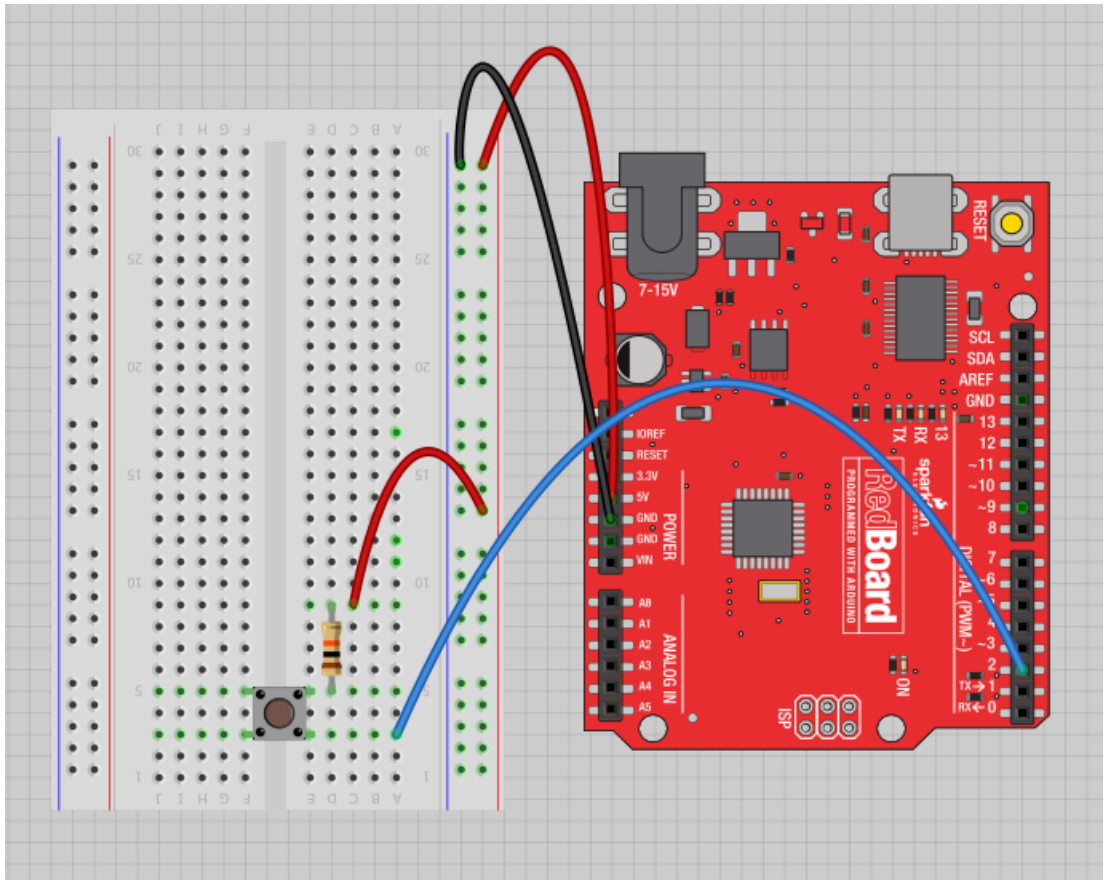
In Arduino, open up:

File → Examples → 02.Digital → Button

Digital Sensors (a.k.a. Switches)
Pull-up Resistor (circuit)

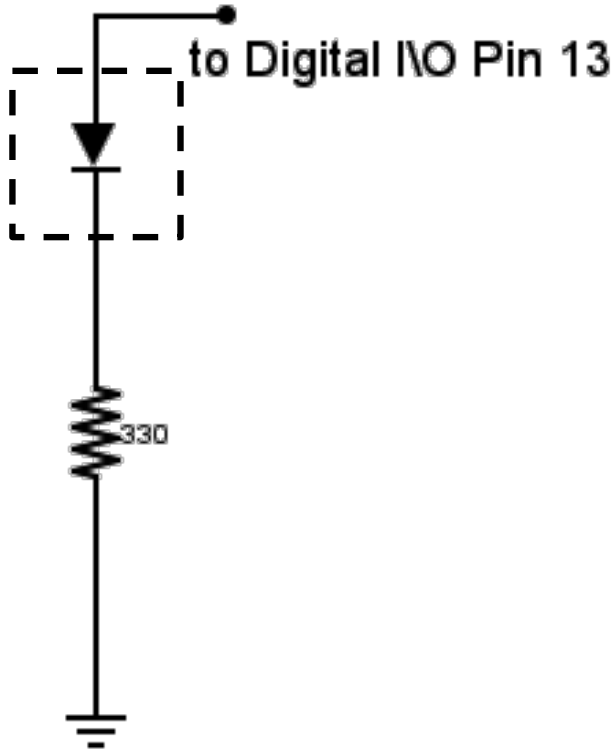


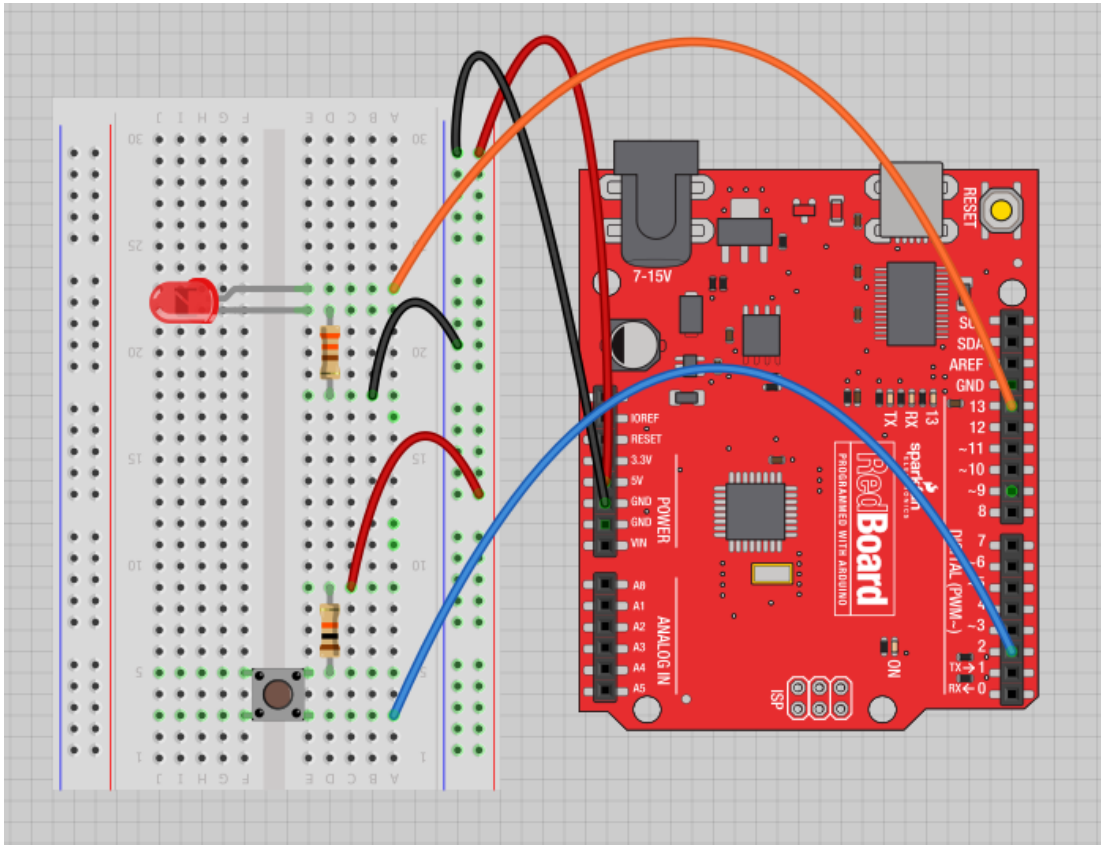
to Digital Pin 2



Digital Sensors (a.k.a. Switches)
Add an indicator LED to Pin 13

This is just like our 1st circuit!





Digital Input

- Connect digital input to your Arduino using Pins # 0 – 13 (Although pins # 0 & 1 are also used for programming)

- Digital Input needs a `pinMode` command:

```
pinMode (pinNumber, INPUT);
```

*Make sure to use ALL CAPS for **INPUT***

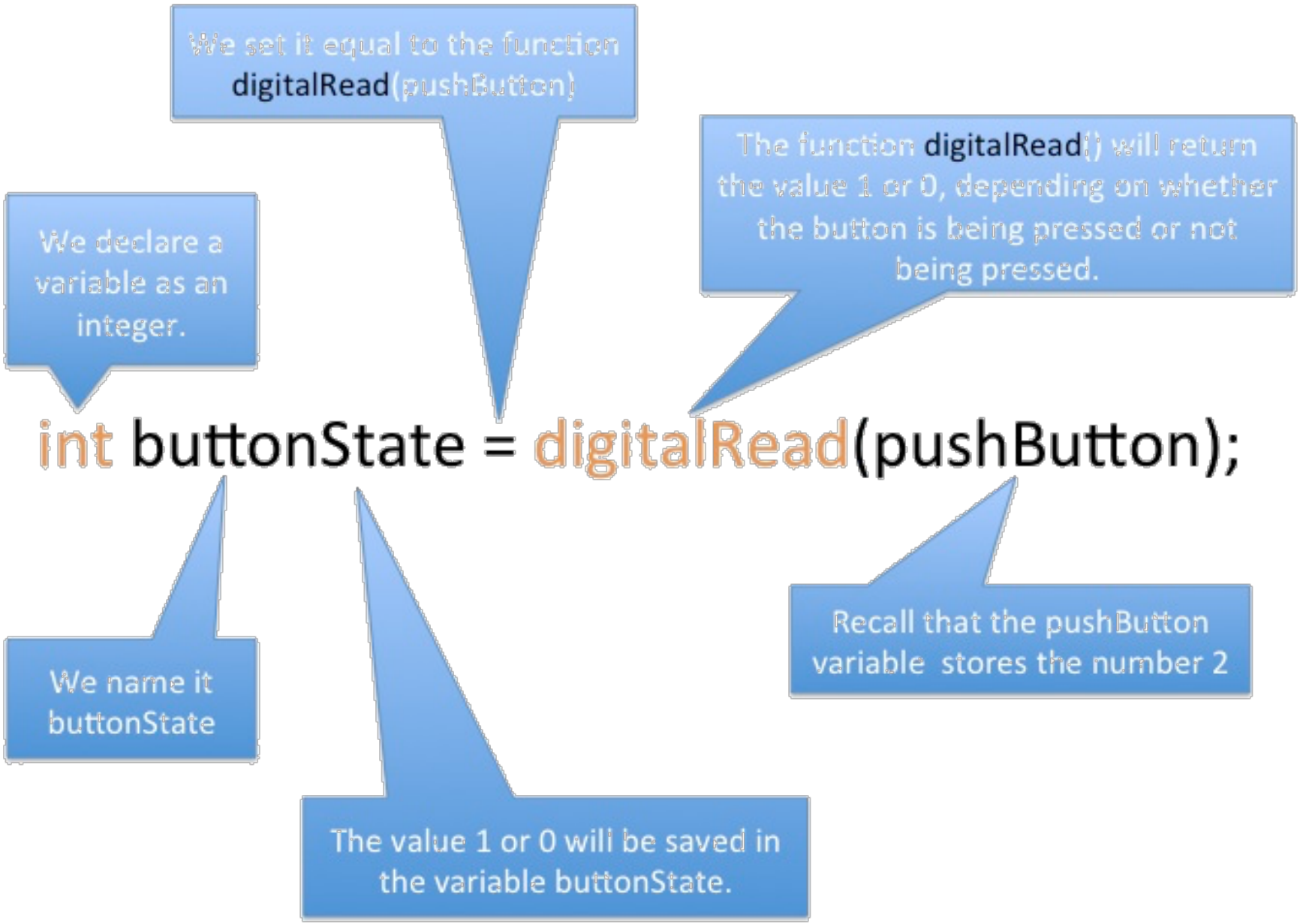
- To get a digital reading:

```
int buttonState = digitalRead (pinNumber);
```

- Digital Input values are only **HIGH** (On) or **LOW** (Off)

Digital Sensors

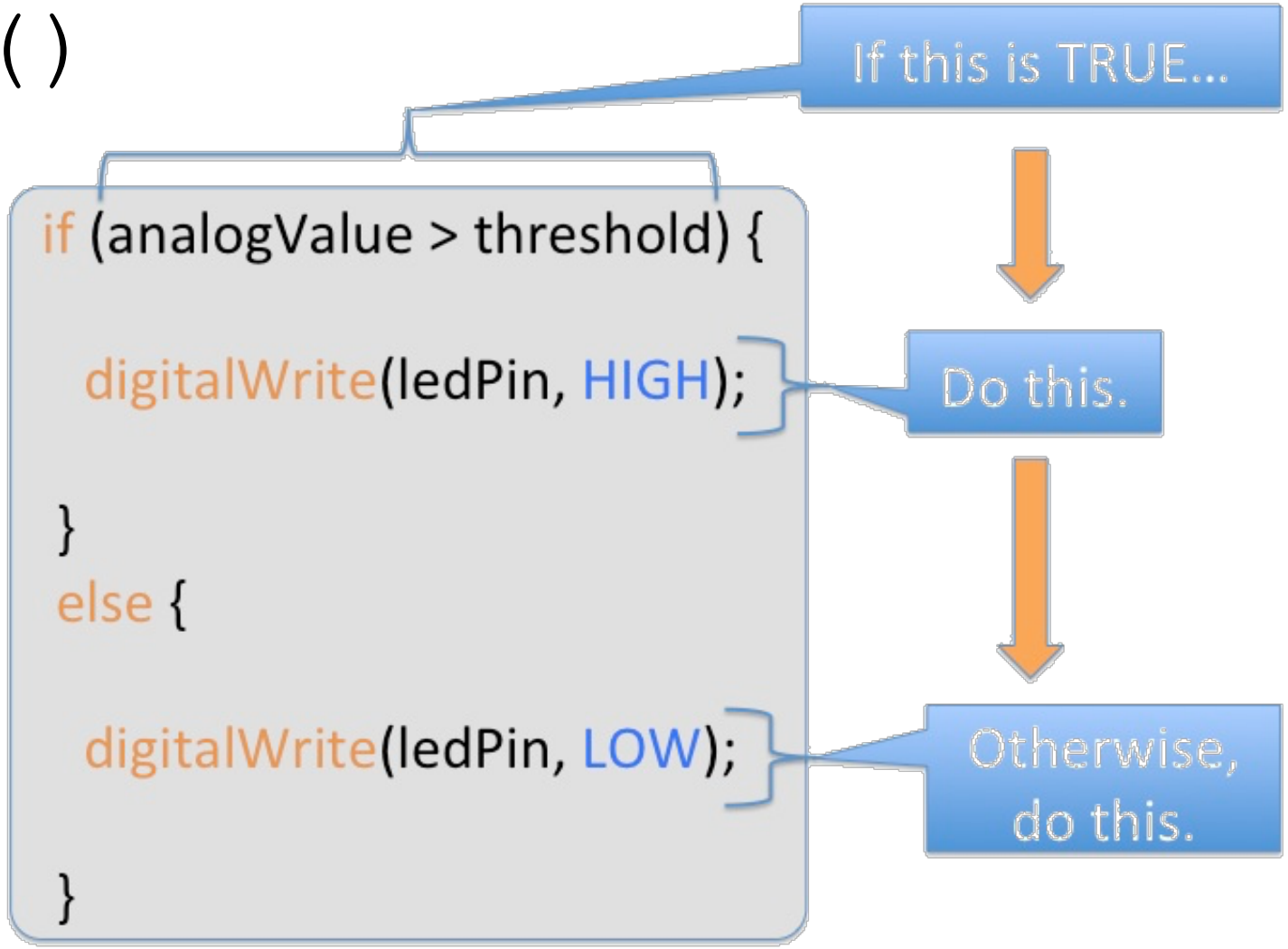
- Digital sensors are more straight forward than Analog
- No matter what the sensor there are only two settings: On and Off
- Signal is always either HIGH (On) or LOW (Off)
- Voltage signal for HIGH will be a little less than 5V on your Uno
- Voltage signal for LOW will be 0V on most systems



http://opensourcehardwarejunkies.com/tutorial-03-digitalread-and-serial-port-communication/

Programming: Conditional Statements

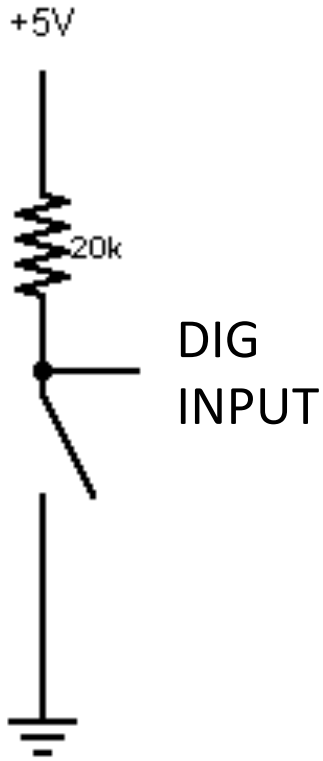
`if()`



Programming: Conditional Statements

`if()`

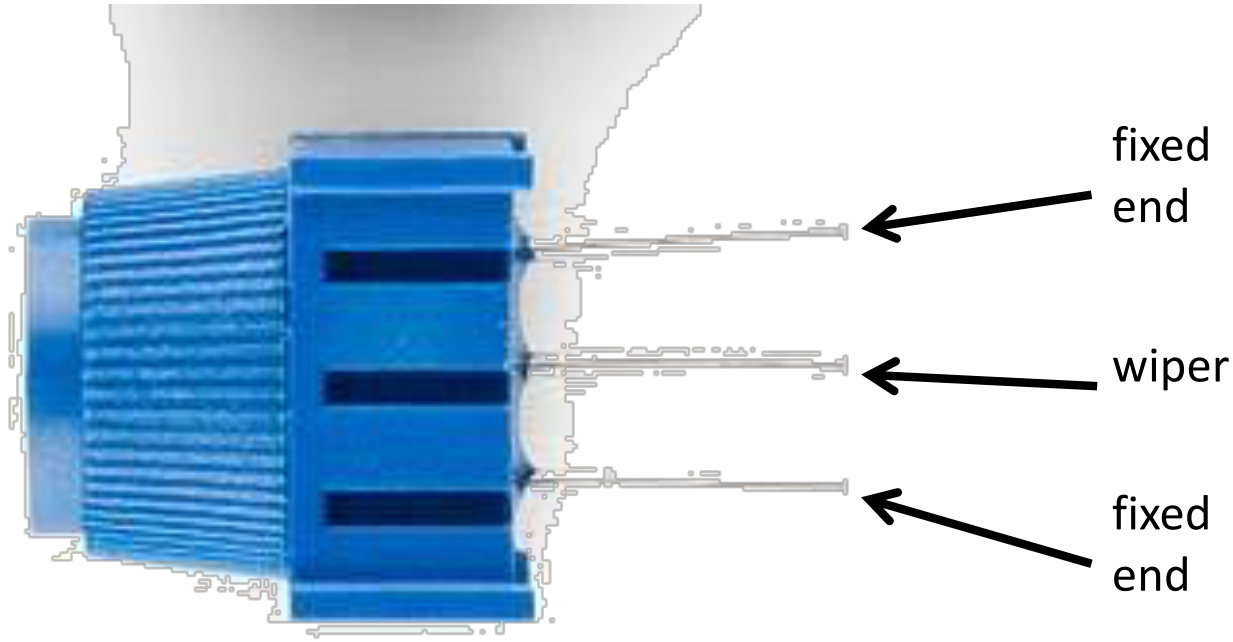
```
void loop()
{
  int buttonState = digitalRead(5);
  if(buttonState == LOW)
  {   // do something
  }
  else
  {  // do something else
  }
}
```

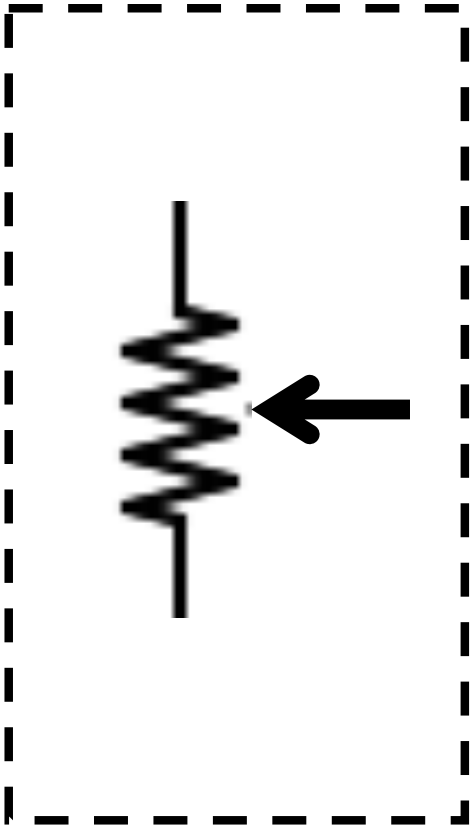


Boolean Operators

<Boolean>	Description
( ) == ( )	is equal?
( ) != ( )	is not equal?
( ) > ( )	greater than
( ) >= ( )	greater than or equal
( ) < ( )	less than
( ) <= ( )	less than or equal

Trimpot (Potentiometer) Variable Resistor

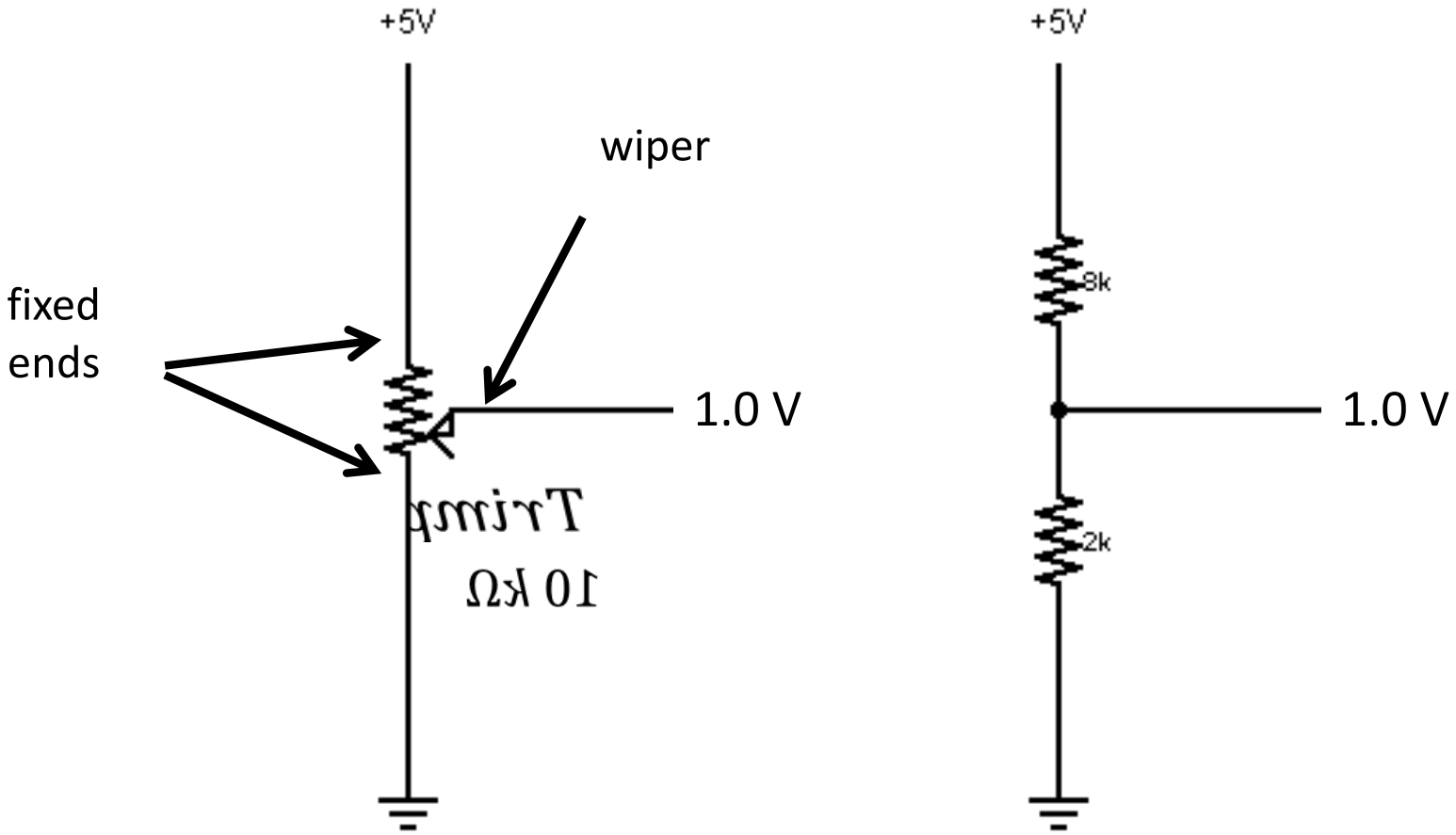




Analog Sensors

3 Pin Potentiometer = var. resistor (circuit)

a.k.a. Voltage Divider Circuit



Ohms Law… (just the basics)
Actually, this is the “voltage divider”

$$V_{R1} = V_{CC} \cdot \left(\frac{R_1}{\mathrm{R}_{Total}} \right)$$

$$V_{R2} = V_{CC} \cdot \left(\frac{R_2}{\mathrm{R}_{Total}} \right)$$

$$R_{Total} = R_1 + R_2$$

analogRead()

Arduino uses a 10-bit A/D Converter:

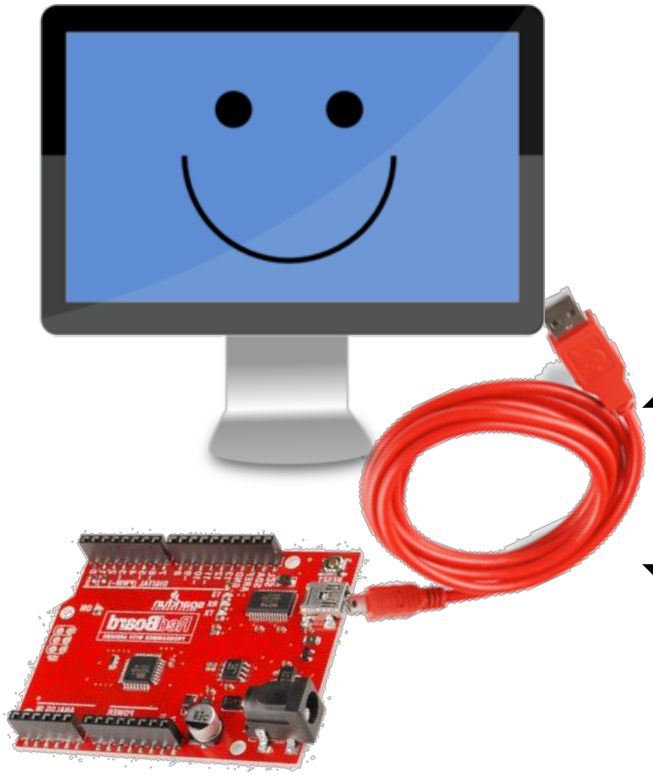
- this means that you get input values from 0 to 1023
  - 0 V → 0
  - 5 V → 1023

Ex:

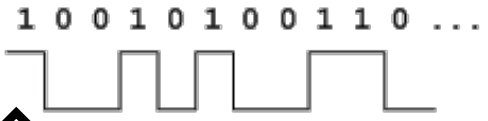
```
int sensorValue = analogRead(A0);
```

Using Serial Communication

Method used to transfer data between two devices.



Data passes between the computer and Arduino through the USB cable. Data is transmitted as zeros (‘0’) and ones (‘1’) sequentially.



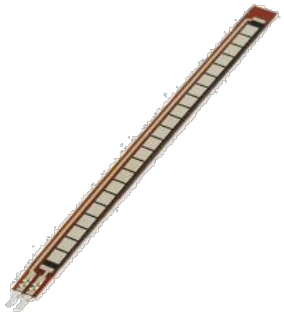
Arduino dedicates Digital I/O pin # 0 to receiving and Digital I/O pin #1 to transmit.

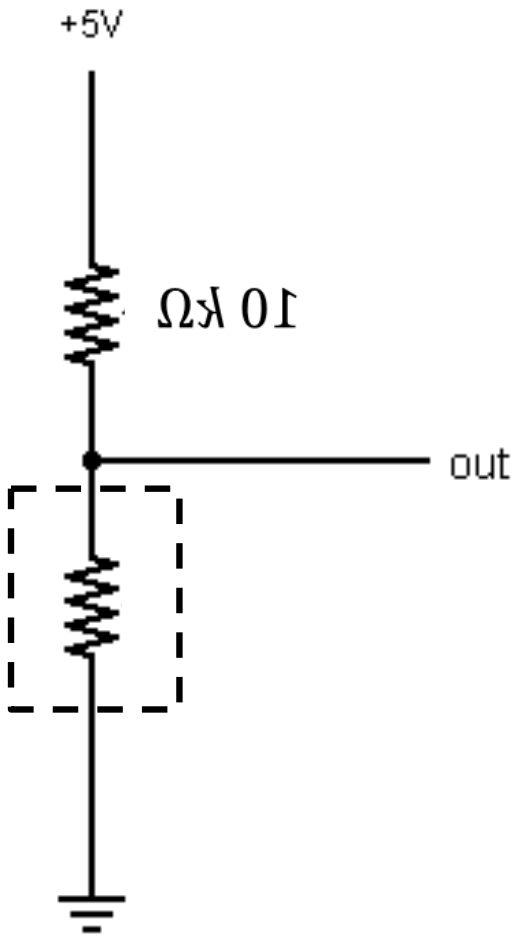
Serial Monitor & analogRead()

Serial Monitor & analogRead()

Opens up a Serial Terminal Window

Analog Sensors
2 Pin Analog Sensors = var. resistor





- Take two sensors -- Use the Serial Monitor and find the range of input values you get for each sensor.

- MaxAnalogRead = _____

- MinAnalogRead = _____

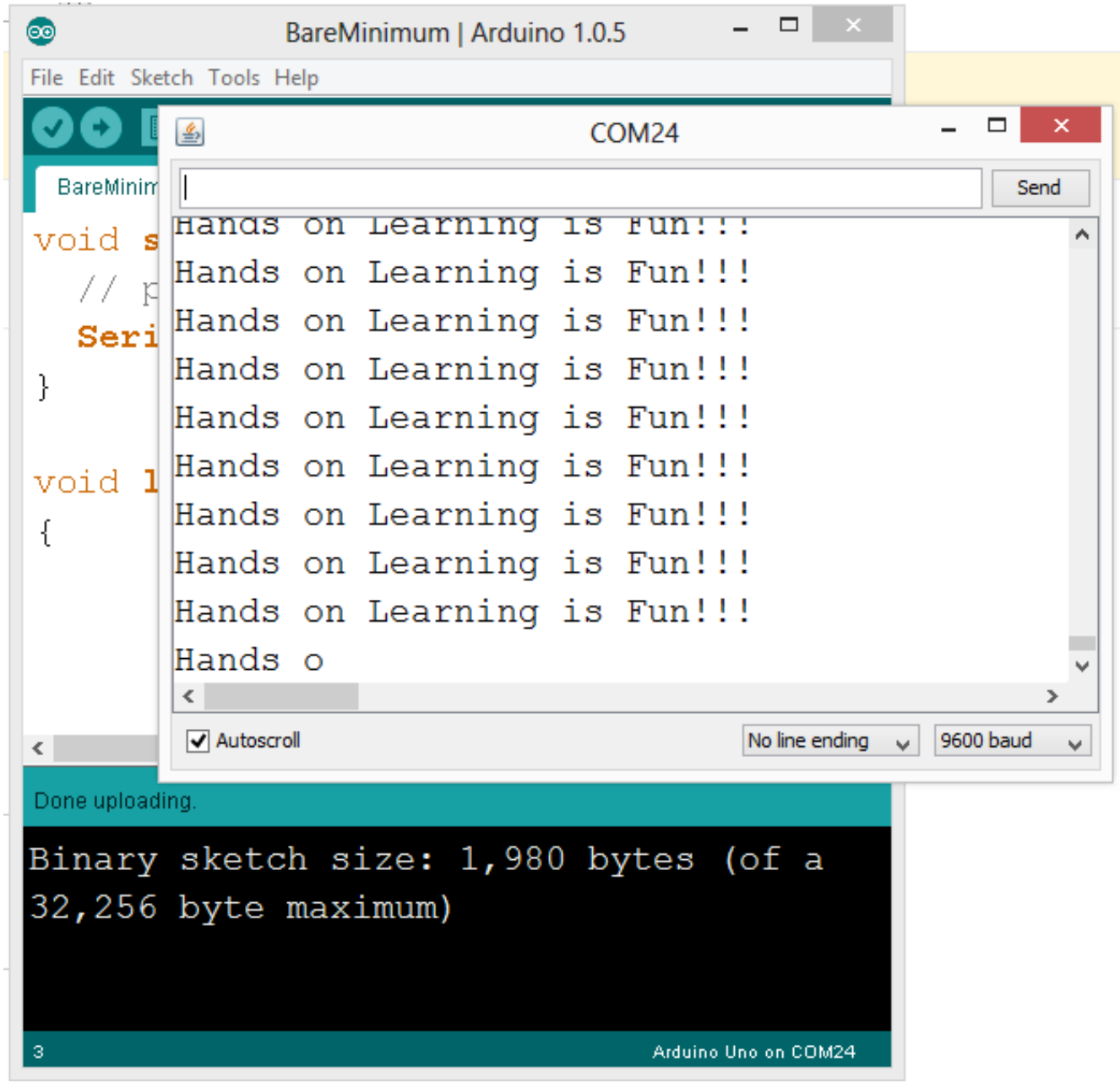
Analog Sensors

- Examples:

Sensors	Variables
Mic	soundVolume
Photoresistor	lightLevel
Potentiometer	dialPosition
Temp Sensor	temperature
Flex Sensor	bend
Accelerometer	tilt/acceleration

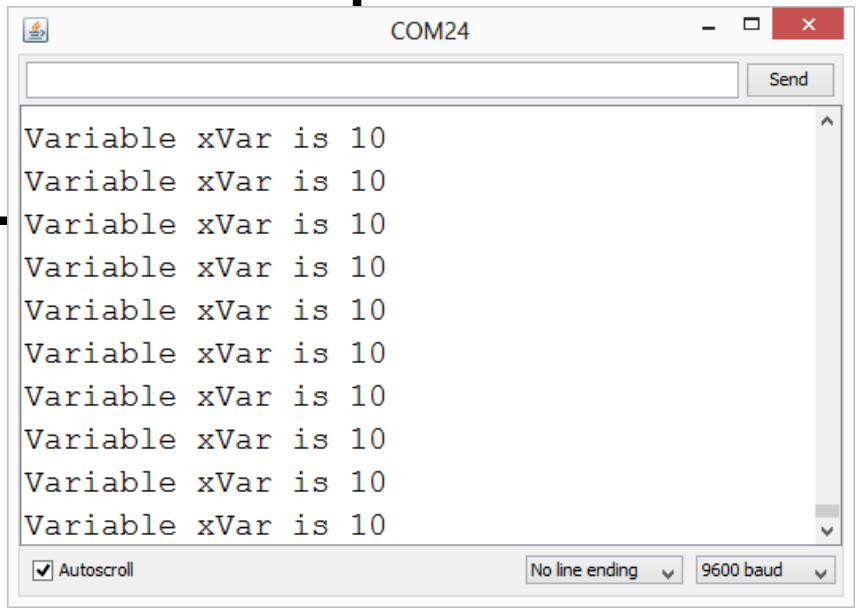
Additional Serial Communication Sending a Message

```
void loop ( )
{
 Serial.print(“Hands on “) ;
 Serial.print(“Learning ”) ;
 Serial.println(“is Fun!!!”) ;

}
```


BareMinimum | Arduino 1.0.5
File Edit Sketch Tools Help
BareMinim
void s
// p
Seri
}
void l
{
COM24
Send
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands o
Autoscroll
No line ending
9600 baud
Done uploading.
Binary sketch size: 1,980 bytes (of a 32,256 byte maximum)
3
Arduino Uno on COM24

Serial Communication: Serial Debugging

```
void loop()
{
  int xVar = 10;
  Serial.print ( "Variable xVar is " ) ;
  Serial.println ( xVar ) ;
}
```



Serial Communication: Serial Troubleshooting

```
void loop ( )
{
  Serial.print ("Digital pin 9: ");
  Serial.println (digitalRead(9));
}
```