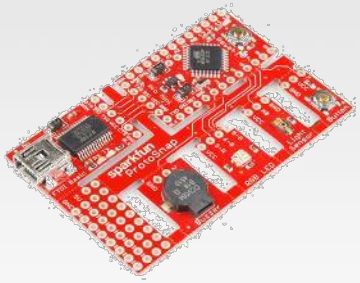
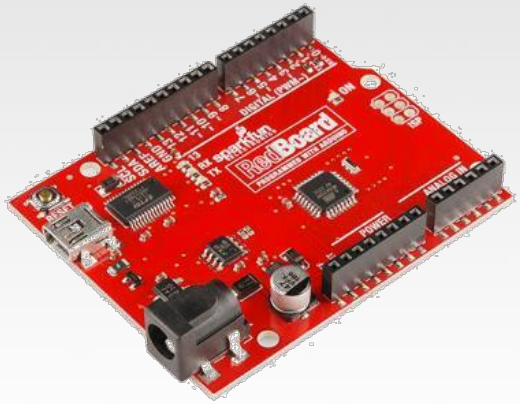
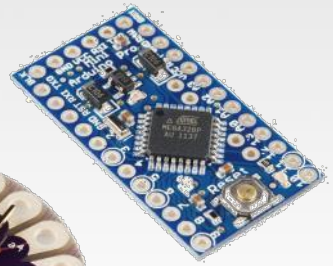
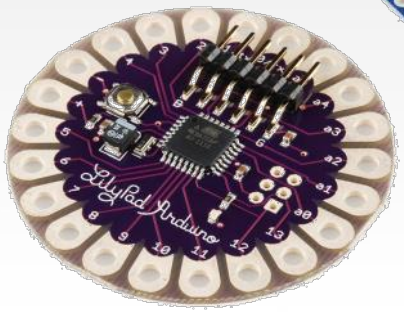
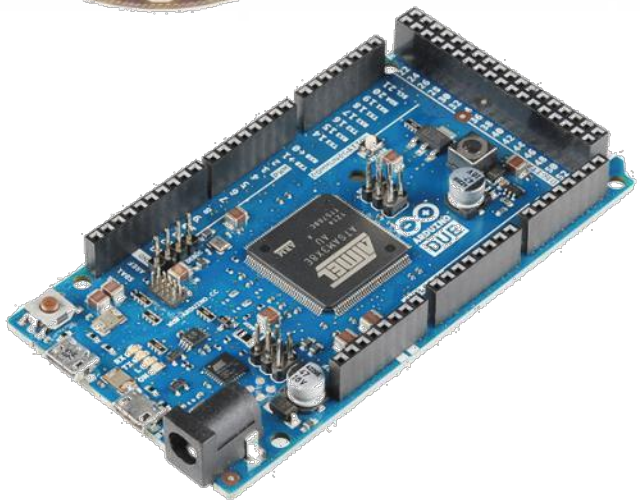
Arduino & Arduino Compatible Boards











Ohm’s Law

Ohm's Law describes the direct relationship between the Voltage (V), Current (I), and Resistance (R) of a circuit.

The three different forms of Ohm's Law are as follows:

$$V = I \cdot R \quad I = \frac{V}{R} \qquad R = \frac{V}{I}$$

V = I R

Electrical Properties

Voltage V	Current I	Resistance R
• Defined as the amount of potential energy in a circuit. • Units: Volts (V)	• The rate of charge flow in a circuit. • Units: Amperes (A)	• Opposition to charge flow. • Units: Ohms (Ω)

$$[V = I \cdot R]$$

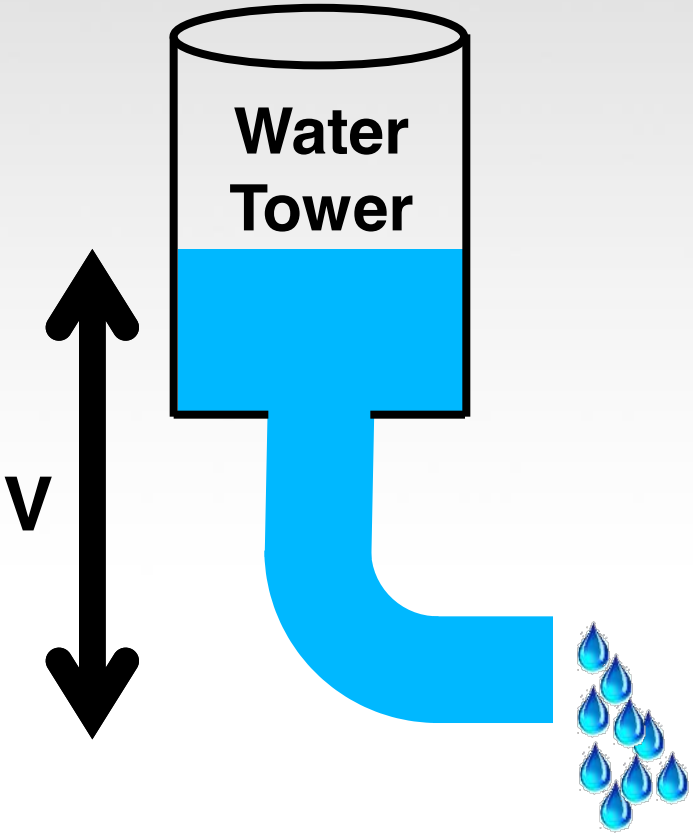
$V = I R$

Current Flow Analogy

High Current

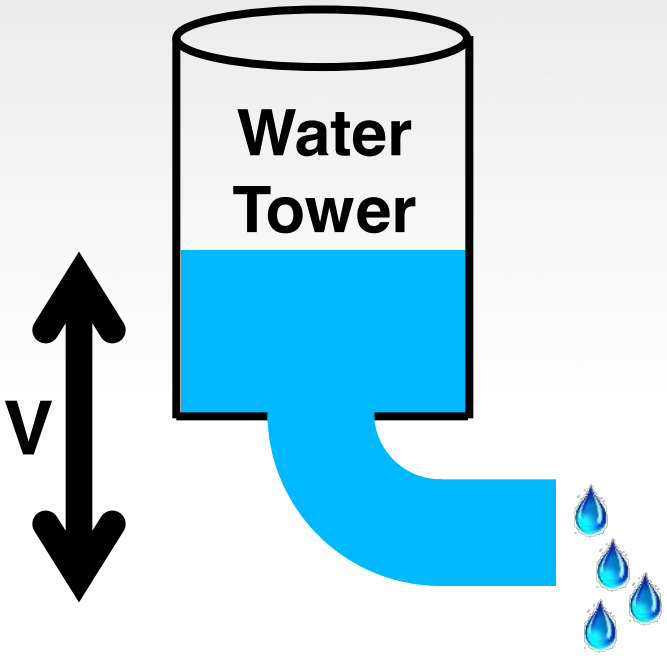
Low Current

Voltage Analogy



More Energy == Higher Voltage

$V = I R$

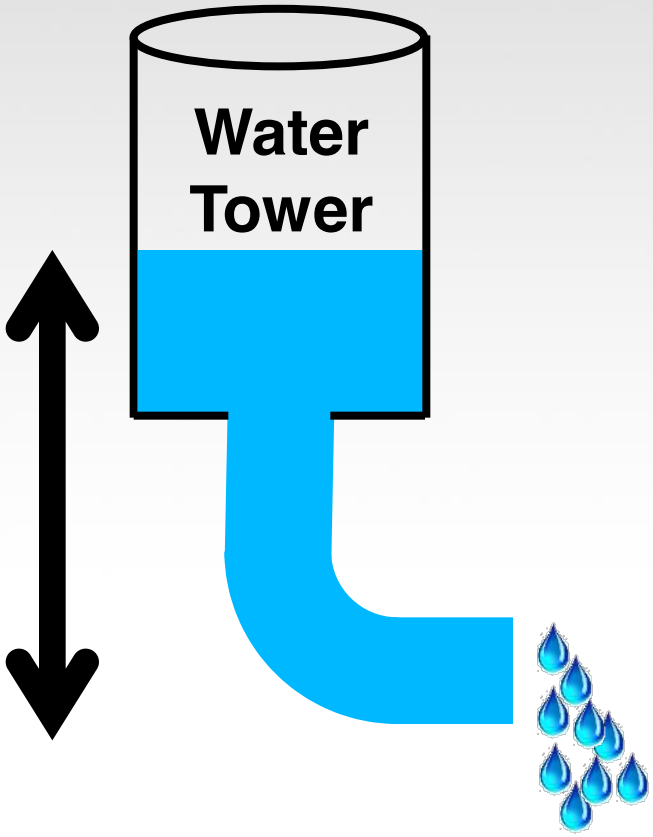


Less Energy == Lower Voltage

$V = I R$

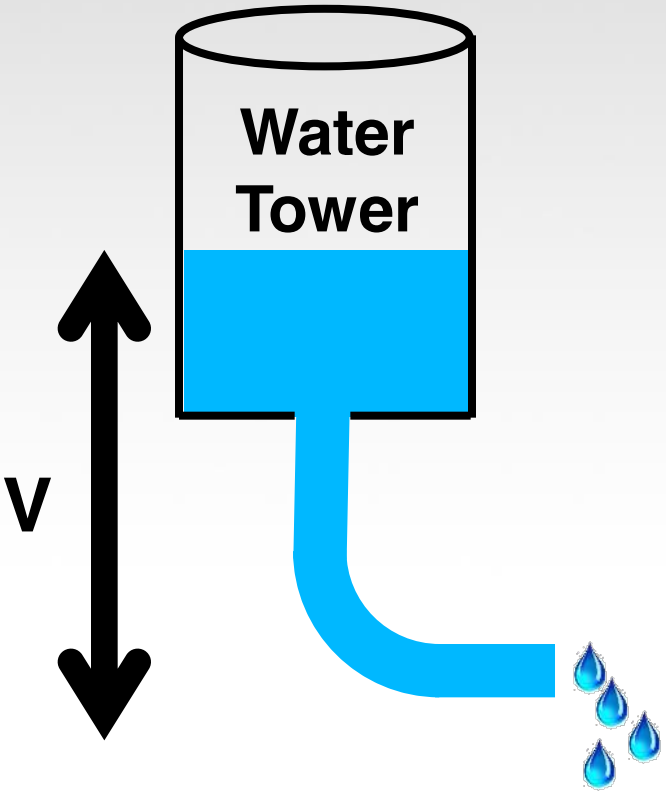
$V = I R$

Resistance Analogy



Big Pipe == Lower Resistance

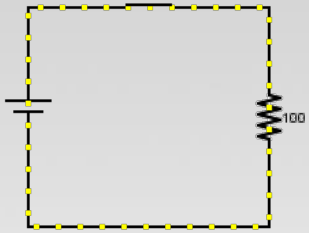
$V = I R$



Small Pipe == Higher Resistance

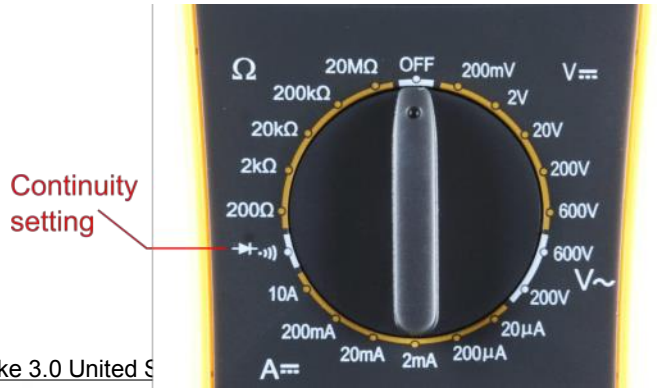
$V = I R$

Continuity – Is it a Circuit?

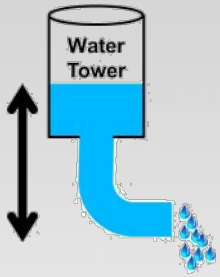


The word “circuit” is derived from the circle. An Electrical Circuit must have a continuous LOOP from Power (V_{cc}) to Ground (GND).

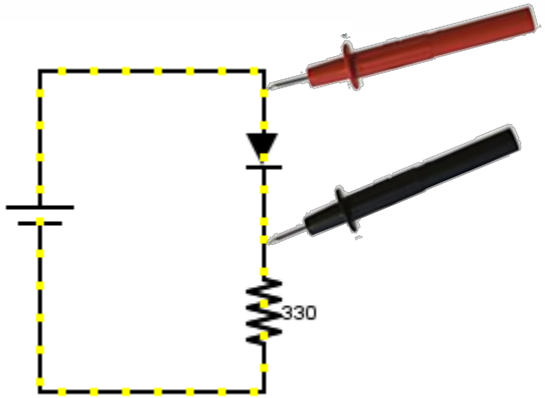
Continuity is important to make portions of circuits are connect. Continuity is the simplest and possibly the most important setting on your multi-meter. Sometimes we call this “ringing out” a circuit.

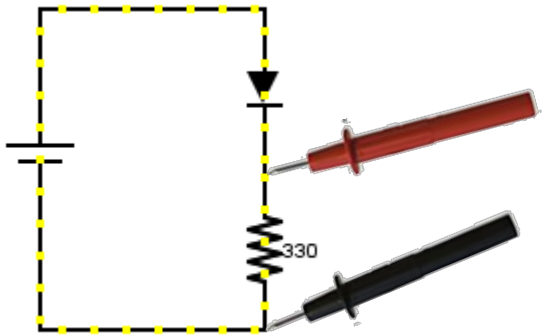


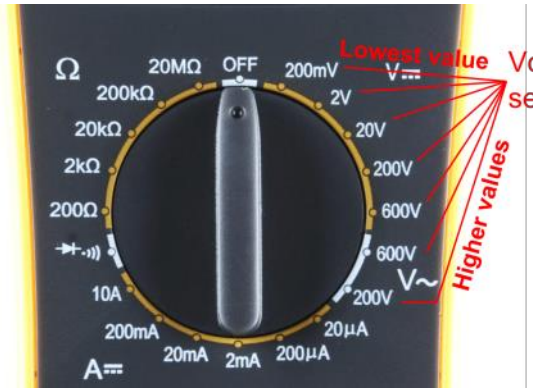
Measuring Electricity – Voltage



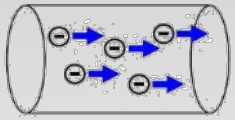
Voltage is a measure of potential electrical energy. A voltage is also called a potential difference – it is measured between two points in a circuit – across a device.





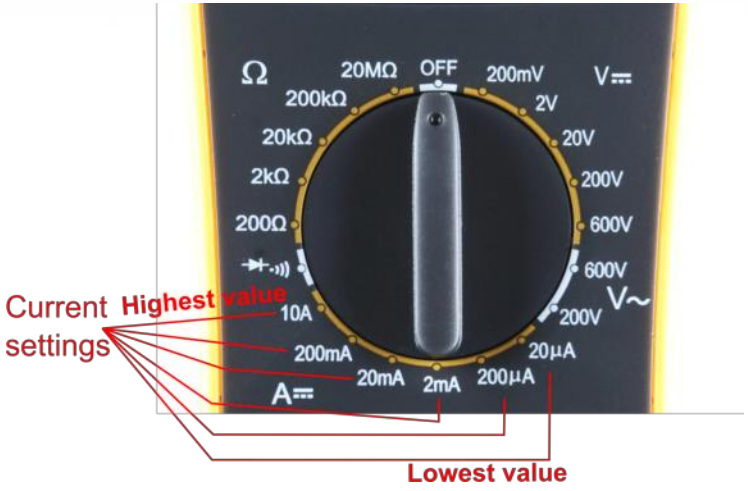


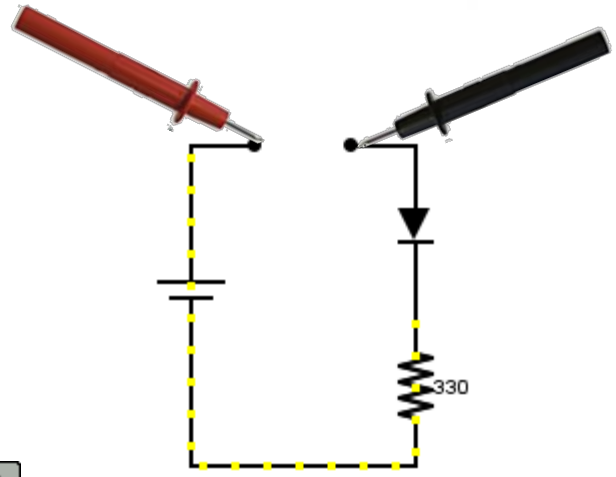
Measuring Electricity -- Current



Current is the measure of the rate of charge flow. For Electrical Engineers – we consider this to be the movement of electrons.

In order to measure this – you must break the circuit or insert the meter in-line (series).



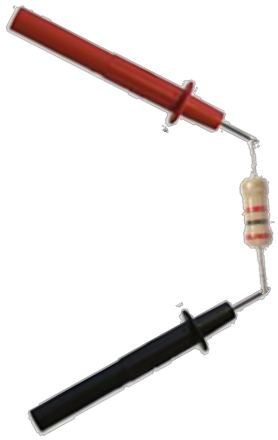


Measuring Electricity -- Resistance

Resistance is the measure of how much opposition to current flow is in a circuit.

Components should be removed entirely from the circuit to measure resistance. Note the settings on the multi-meter. Make sure that you are set for the appropriate range.

Resistance settings

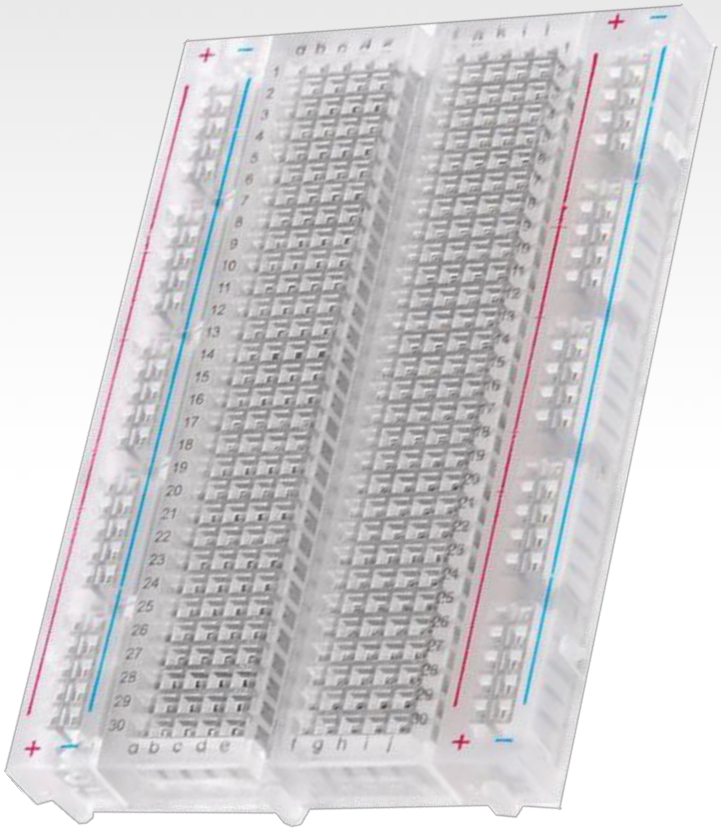


Prototyping Circuits
Solderless Breadboard

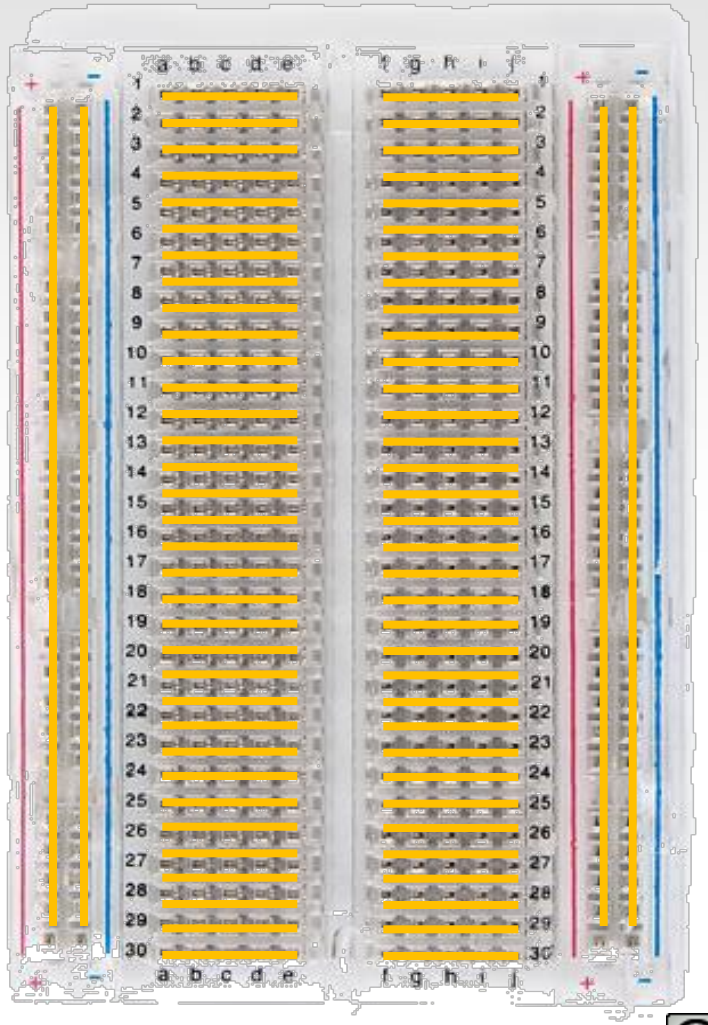
One of the most useful tools in an engineer or Maker's toolkit. The three most important things:

- A breadboard is easier than soldering
- A lot of those little holes are connected, which ones?
- Sometimes breadboards break

What's a Breadboard?



Solderless Breadboard



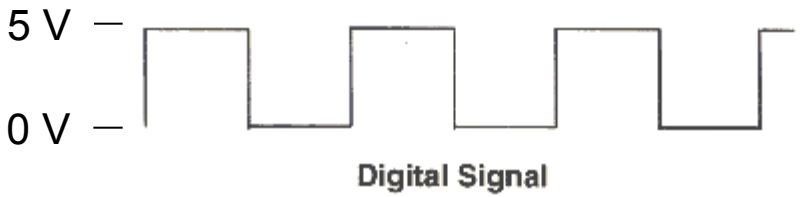
Each row (horiz.) of 5 holes are connected.

Vertical columns – called power bus are connected vertically

Concepts: Analog vs. Digital

Microcontrollers are **digital** devices – ON or OFF. Also called – discrete.

analog signals are anything that can be a full range of values. What are some examples? More on this later…



Digital Signal

Analog Signal

Arduino

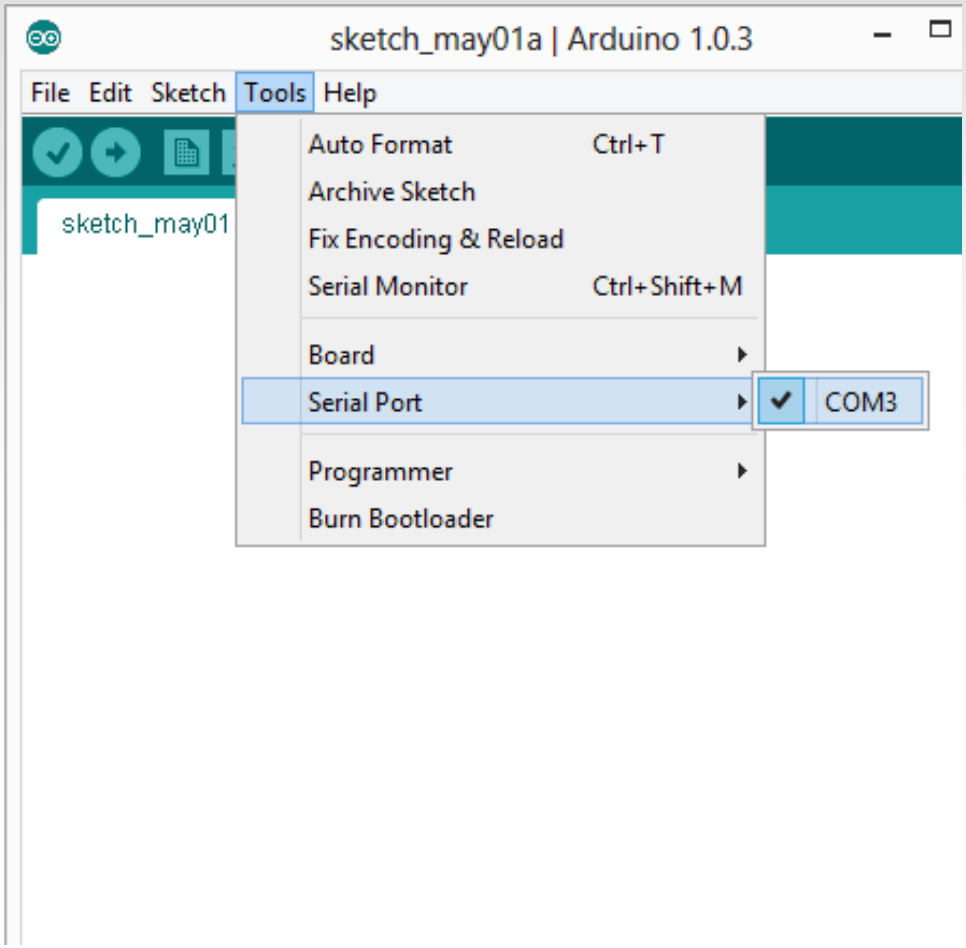
Integrated Development Environment (IDE)

Two required functions / methods / routines:

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

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

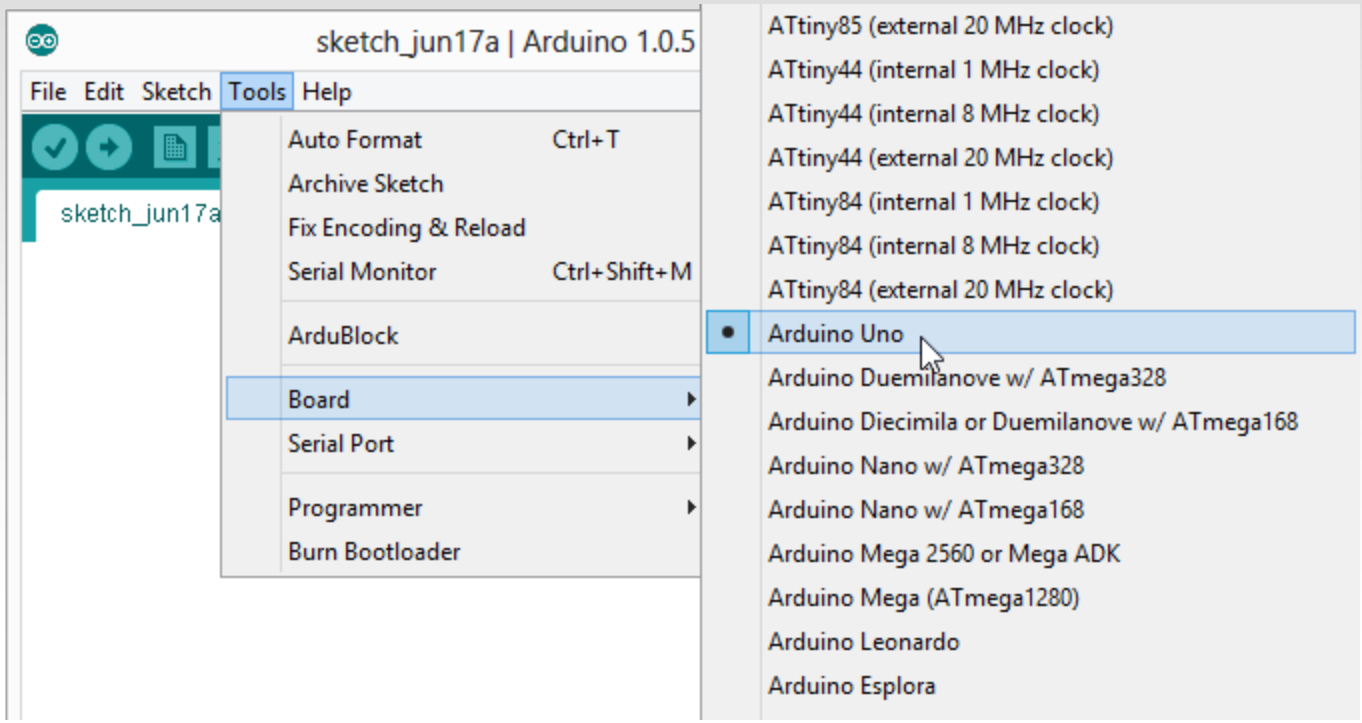
Settings: Tools → Serial Port



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

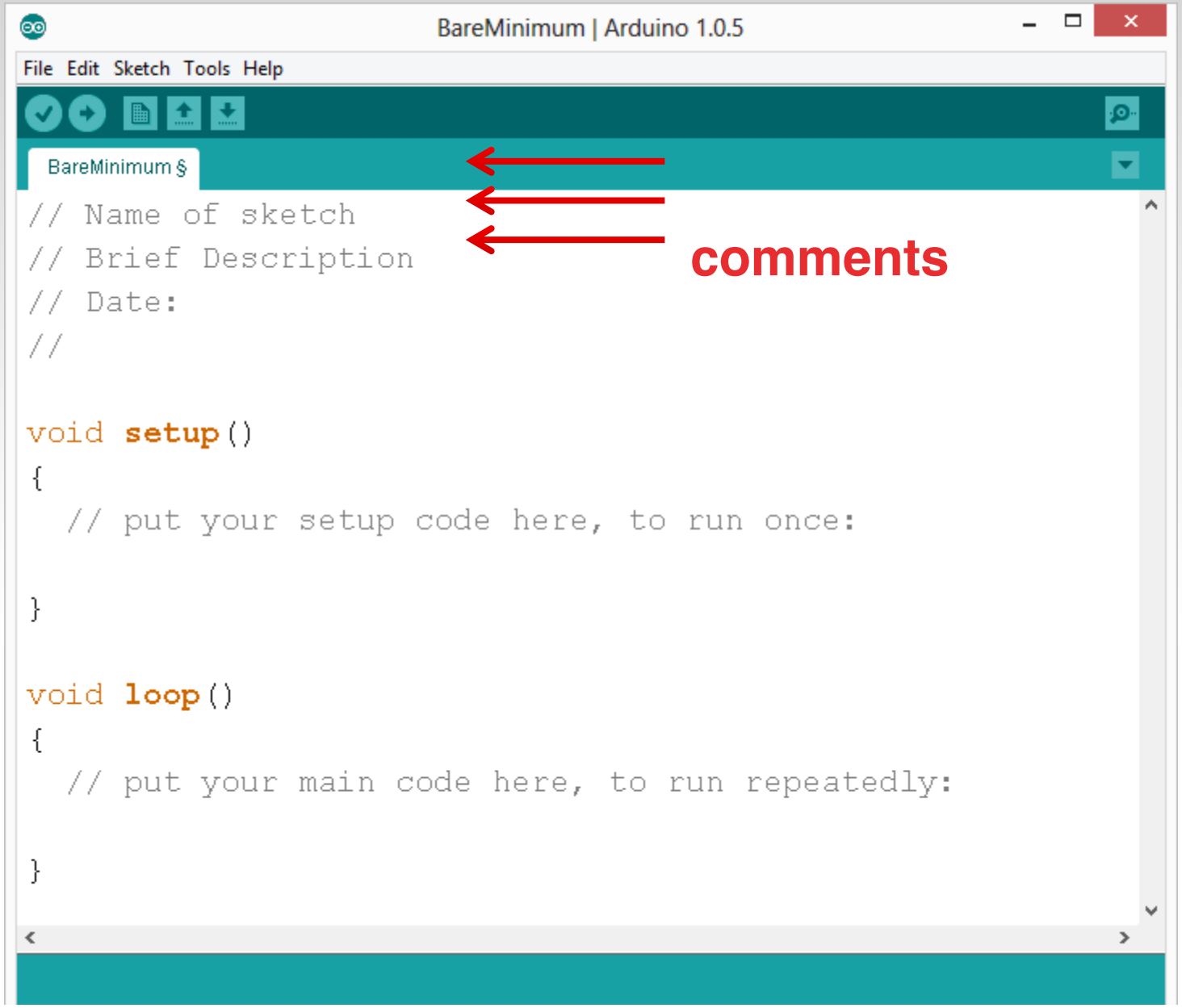


Next, double-check that the proper board is selected under the Tools→Board menu.

Comments, Comments, Comments

Comments are for you – the programmer and your friends…or anyone else human that might read your code.

```
// this is for single line comments
// it's good to put a description at the
   top and before anything 'tricky'
/* this is for multi-line comments
    Like this…
    And this….
*/
```



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
```

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

We set it equal to the function digitalRead(pushButton)

We declare a variable as an integer.

The function digitalRead() will return the value 1 or 0, depending on whether the button is being pressed or not being pressed.

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

We name it buttonState

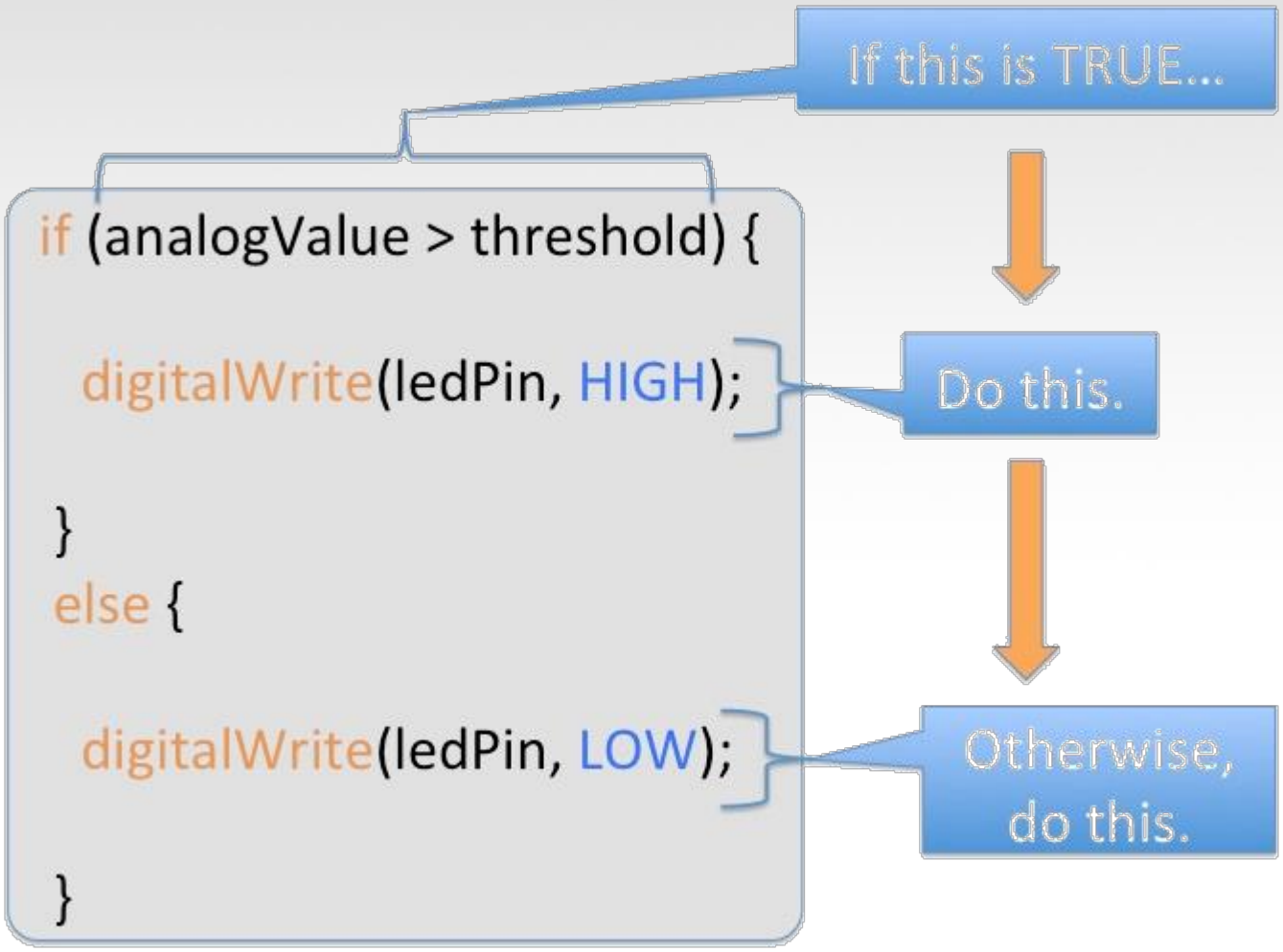
The value 1 or 0 will be saved in the variable buttonState.

Recall that the pushButton variable stores the number 2

http://opensourcehardwarejunkies.com/tutorial-03-digitalread-and-

Programming: Conditional Statements

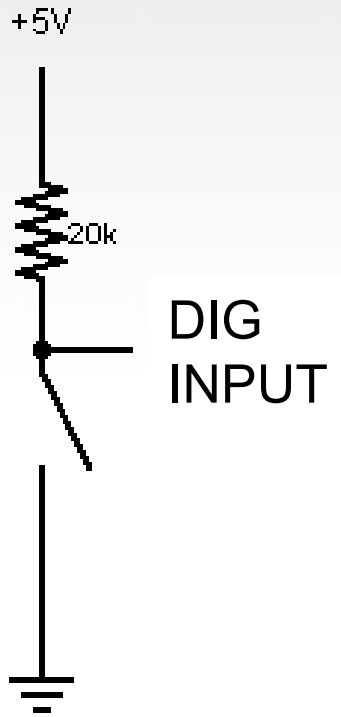
`if()`



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

Programming Concepts: Variable Types

Variable Types:

8 bits	16 bits	32 bits
byte char	int unsigned int	long unsigned long float

Fading in and Fading Out (Analog or Digital?)

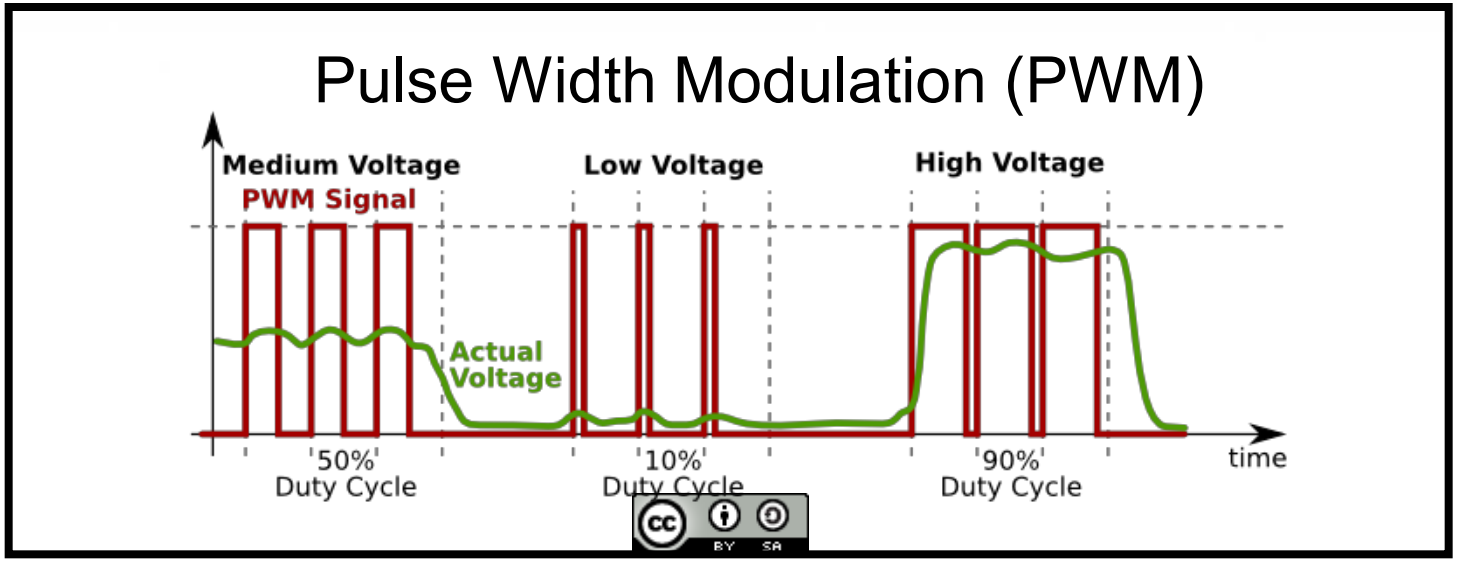
A few pins on the Arduino allow for us to modify the output to mimic an analog signal.

This is done by a technique called:

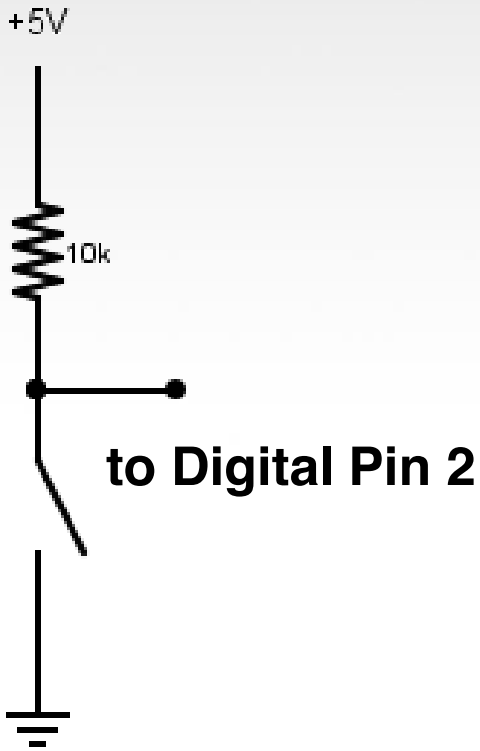
Pulse Width Modulation (PWM)

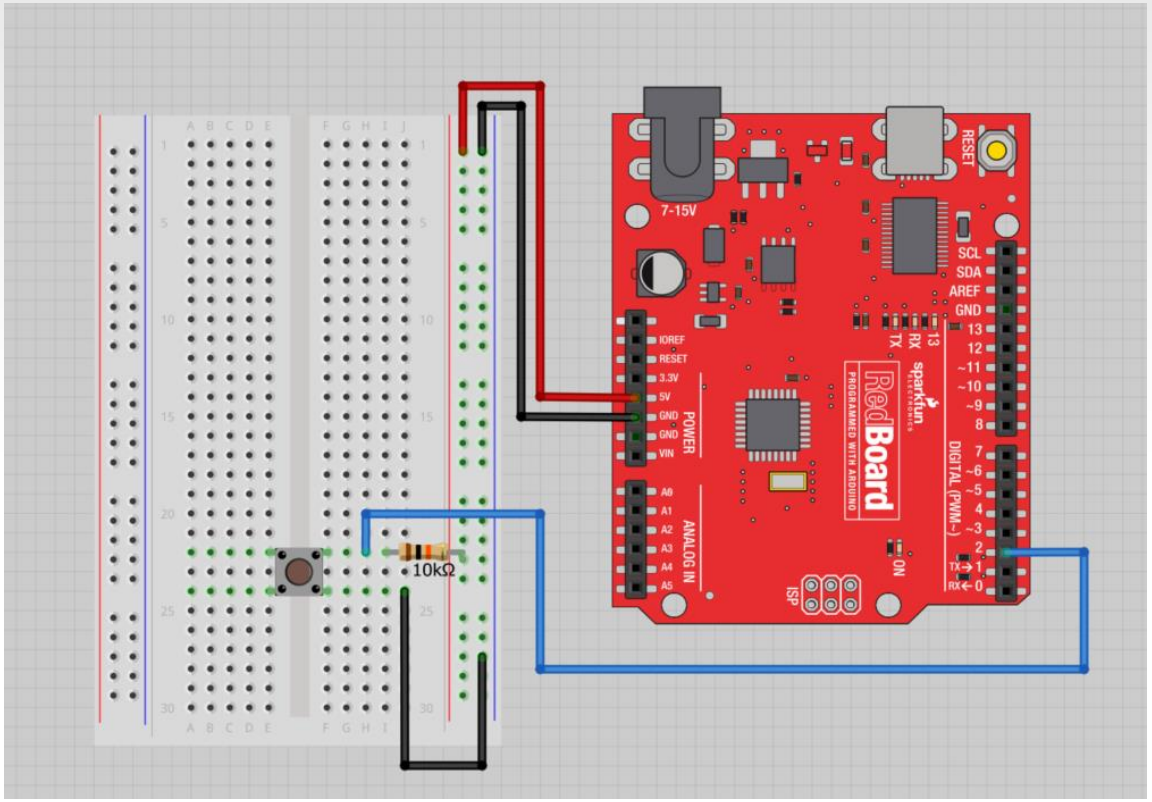
Concepts: Analog vs. Digital

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



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

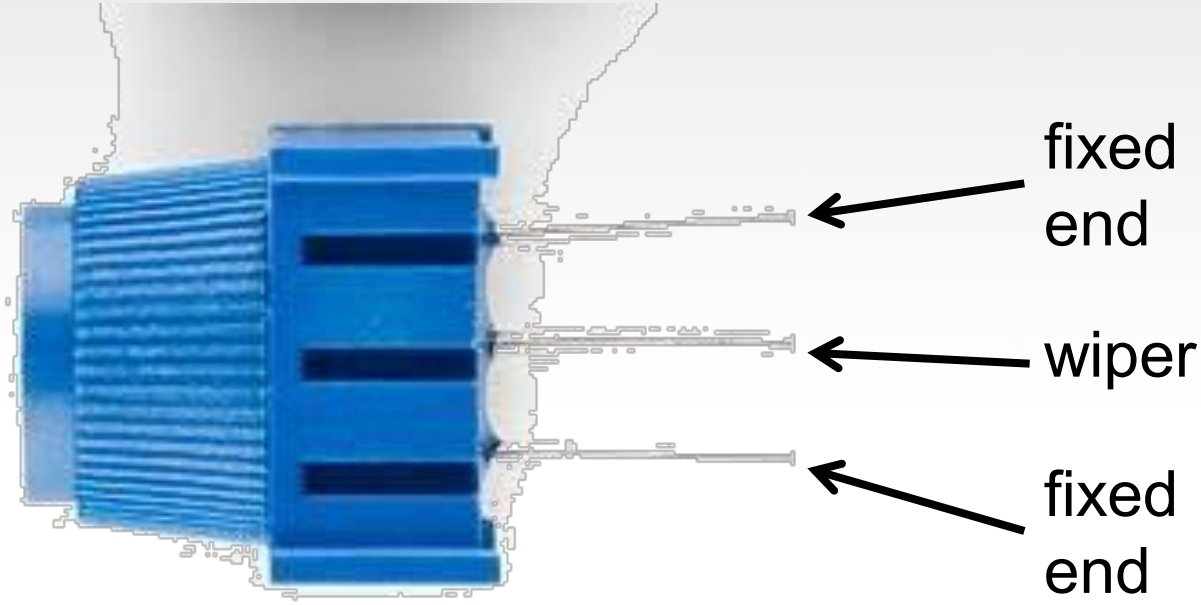
```
int buttonState = digitalRead (pinNumber);
```

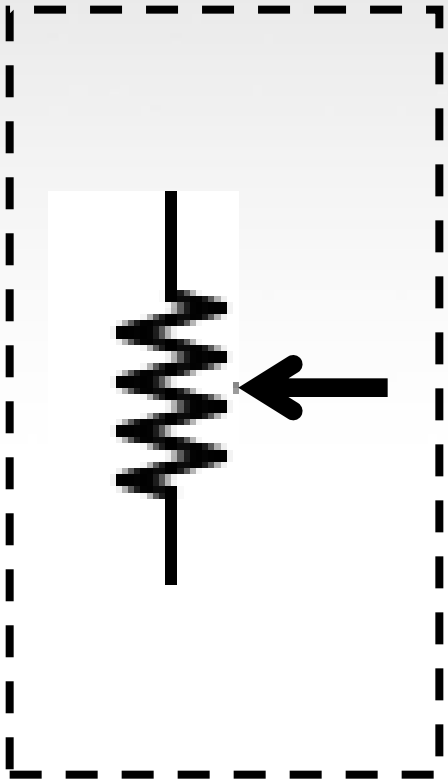
- Digital Input values are only **HIGH** (On) or **LOW** (Off)

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

Trimpot (Potentiometer)
Variable Resistor

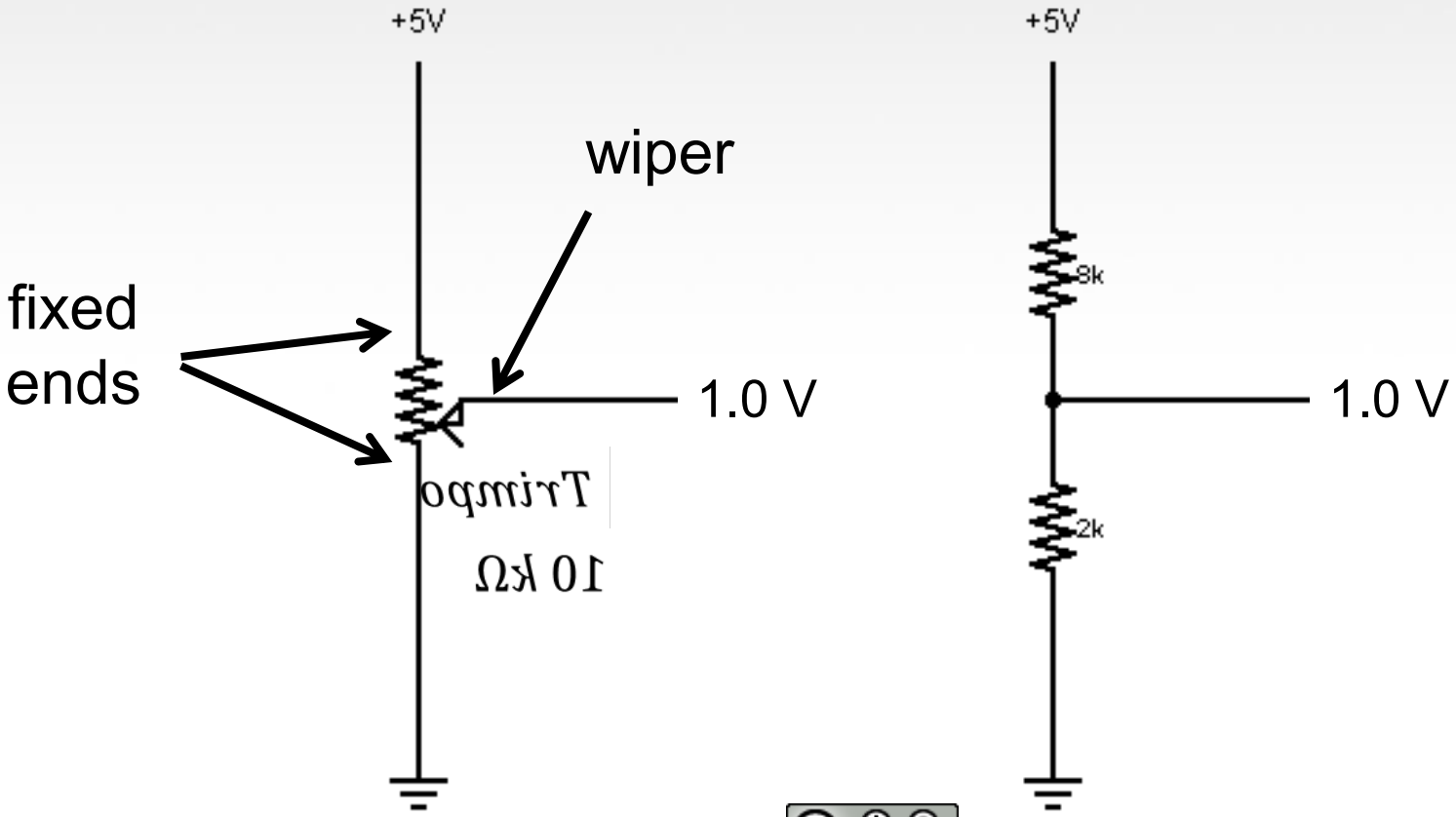




Analog Sensors

3 Pin Potentiometer = var. resistor (circuit)

a.k.a. Voltage Divider Circuit



analogRead()

Arduino uses a 10-bit A/D Converter:

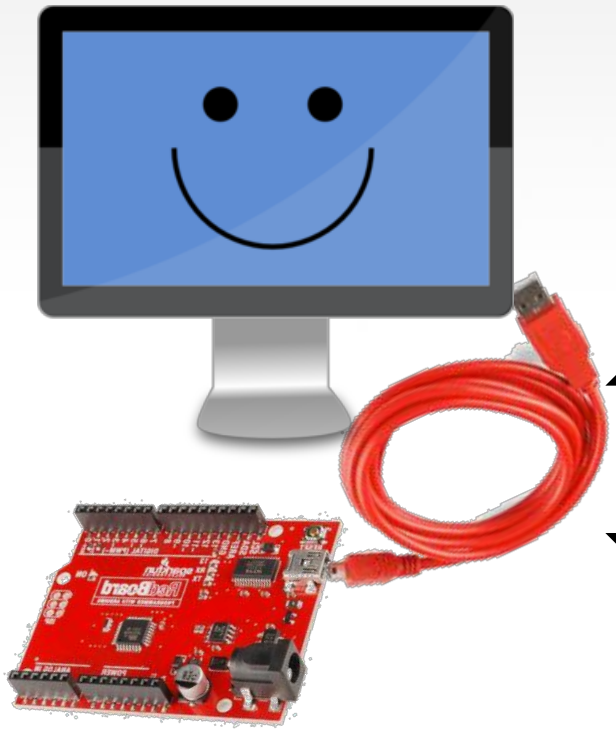
- this means that you get input values from 0 to 1023
  - 0 V → 0
  - 5 V → 1023

Ex:

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

Using Serial Communication

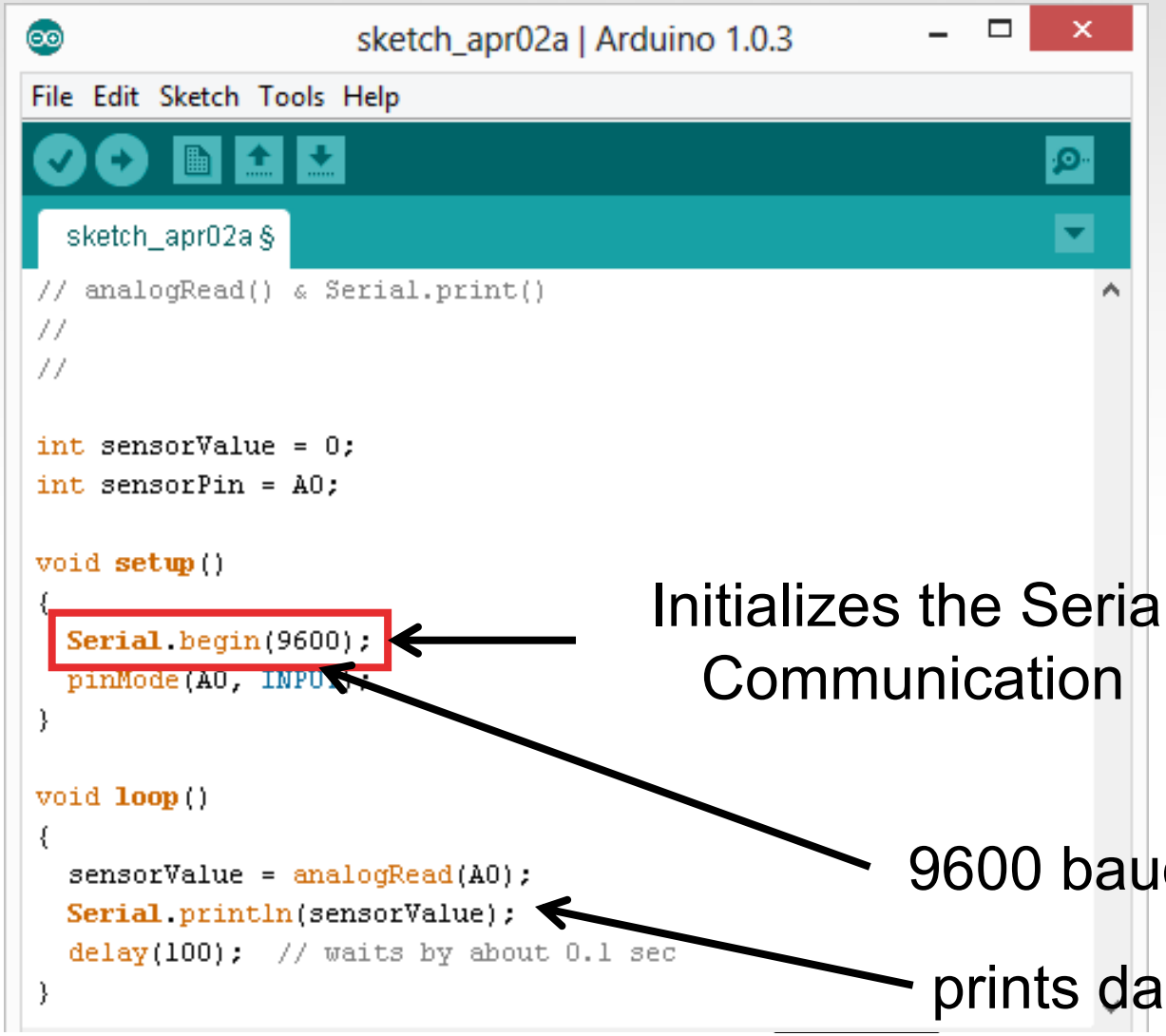
Method used to transfer data between two devices.



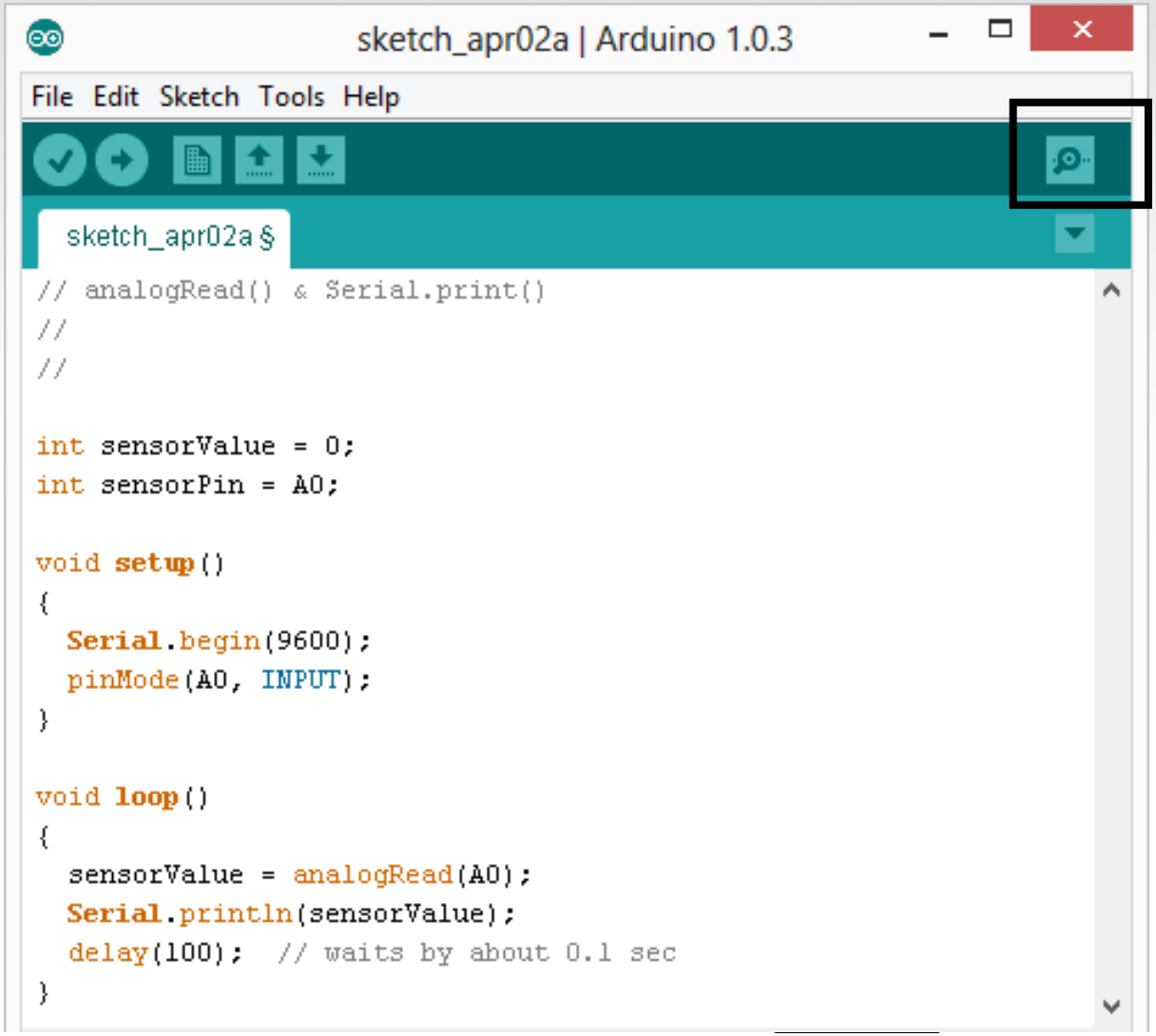
Data passes between the computer and Arduino through the USB cable. Data is transmitted as zeros (‘0’) and ones (‘1’) sequentially.

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