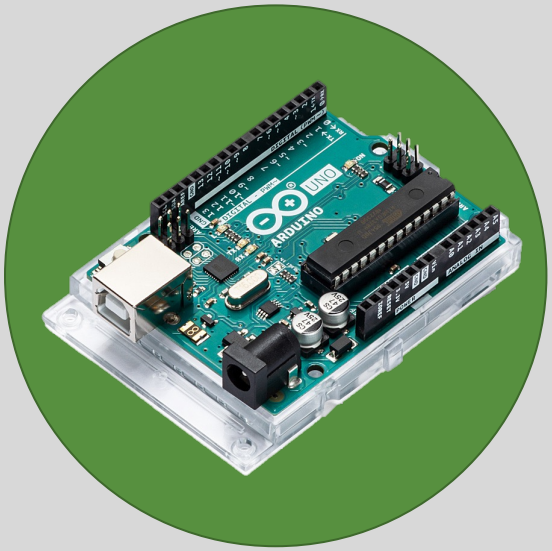
ARDUINO

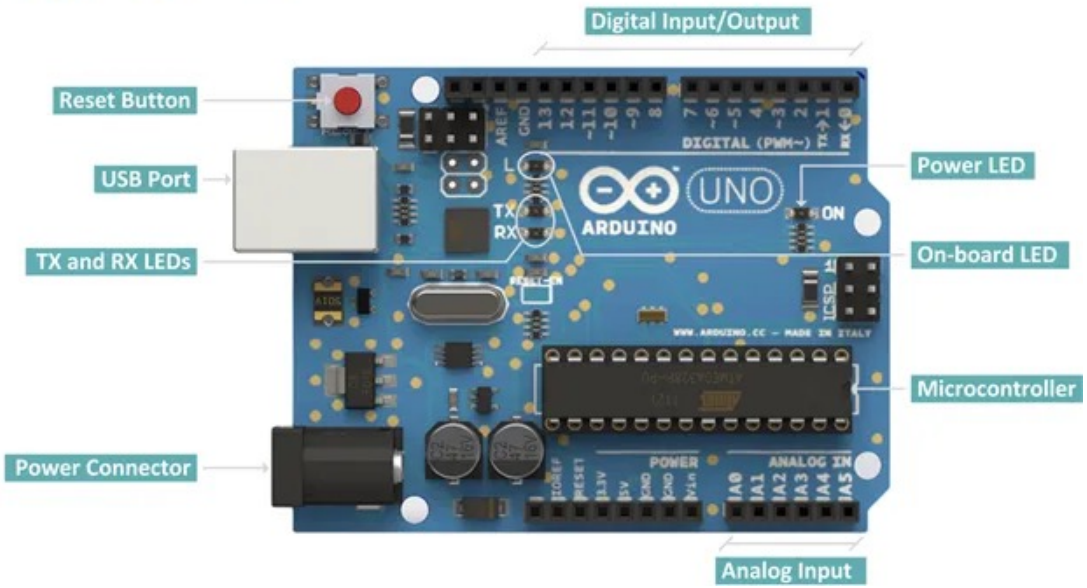
BME 66

Arduino is a combination of a hardware, development environment, and a community

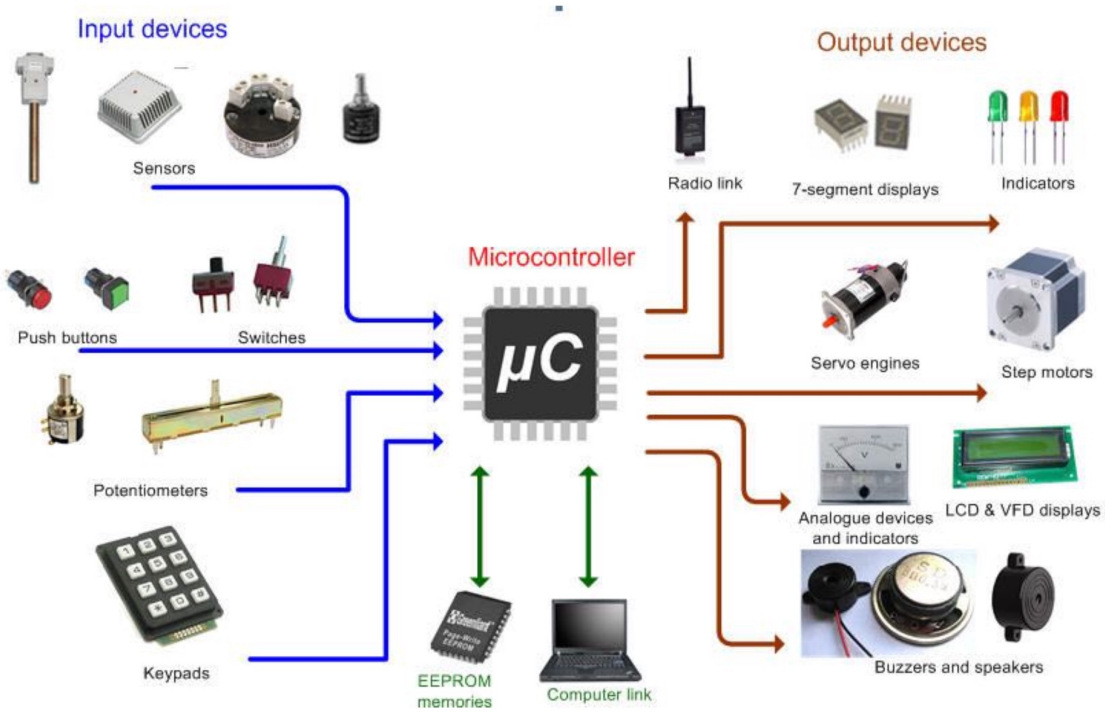


ARDUINO
OPEN-SOURCE
COMMUNITY

https://www.arduino.cc/en/software



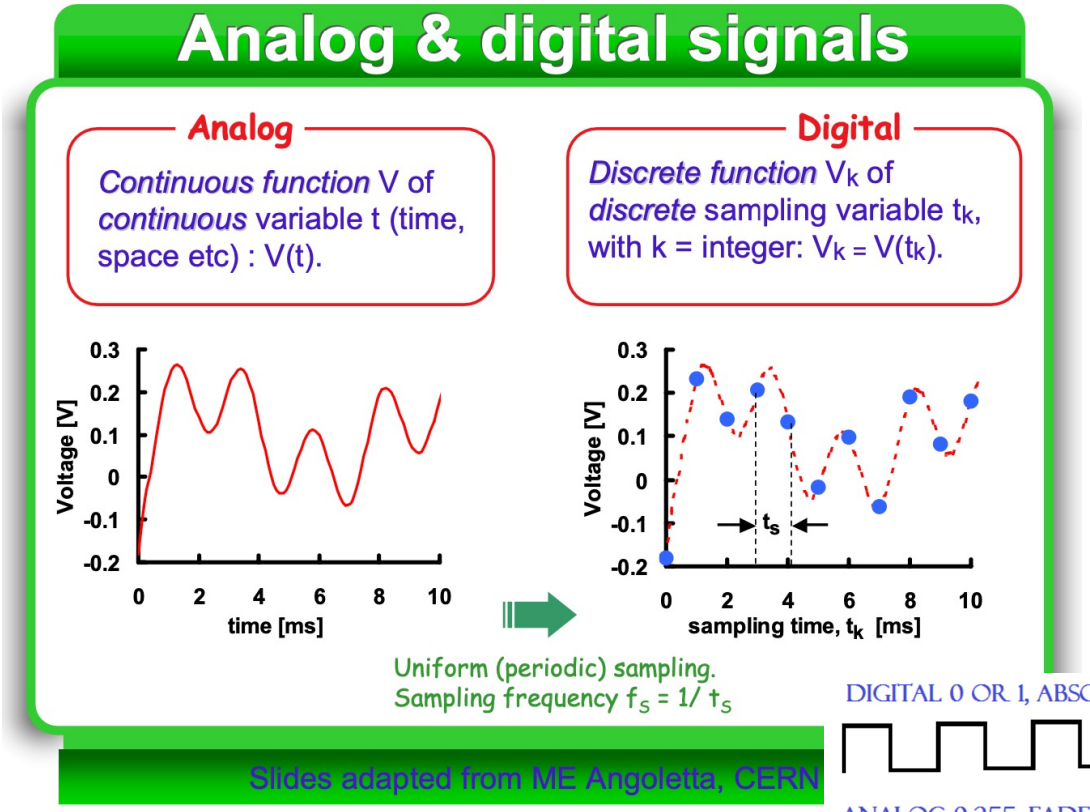
Arduino is a microcontroller

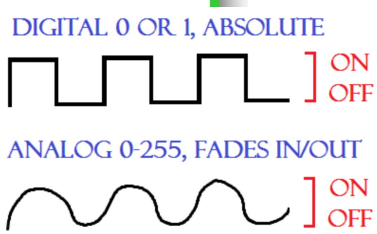


Arduino can be connected to different input and output devices

Sensors or buttons can be used to generate an output effect:

Ex: A push button can be pressed to turn on a light or a servo motor to spin a small fan.



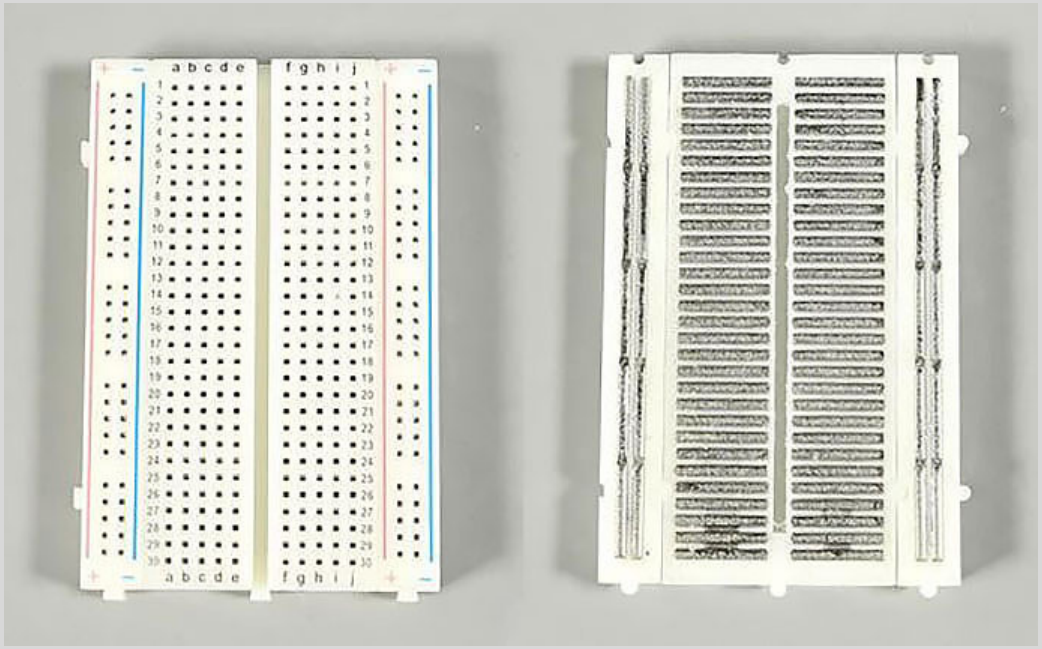


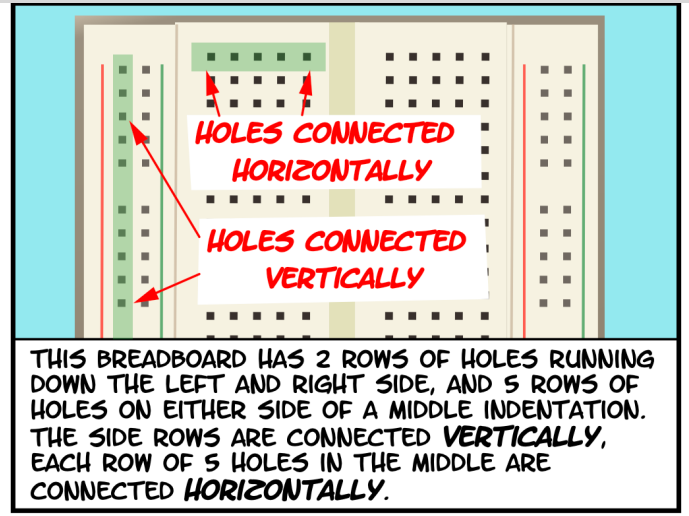
Arduino Uno's can convert analog to digital signals

- Digital signals have only two values, HIGH and LOW.
- Analog signals can take on any number of values.
- Think of Analog like a dimmer switch while Digital is a light switch.

There are many different types of Arduino microcontrollers that each have their own use.

Solderless breadboards are easy ways to create test projects without having to solder together wires and parts





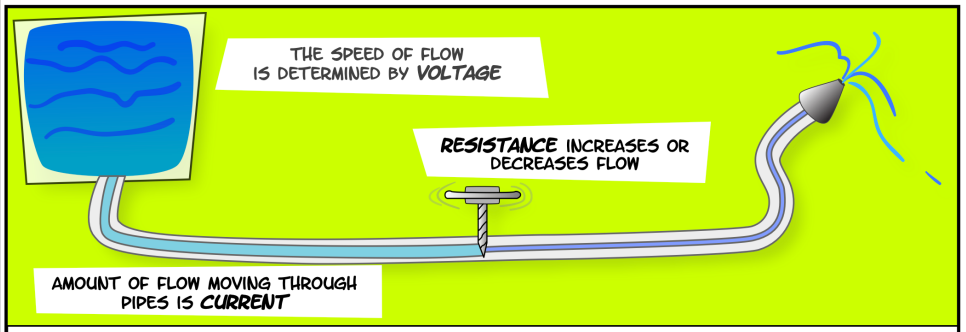
Electricity Review

VOLTAGE (V) IS A MEASURE OF ELECTRICAL POTENTIAL. IT IS MEASURED IN **VOLTS**.

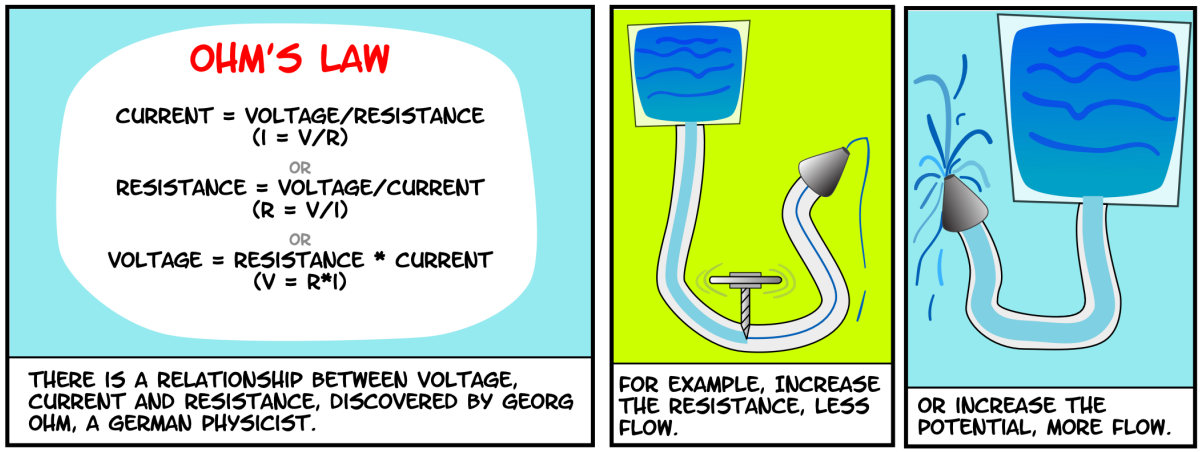
CURRENT (I) IS THE AMOUNT OF FLOW THROUGH A CONDUCTIVE MATERIAL. IT IS MEASURED IN **AMPERES** OR **AMPS**.

RESISTANCE (R) IS A MATERIAL'S OPPOSITION TO THE FLOW OF ELECTRIC CURRENT. IT IS MEASURED IN **OHMS**.

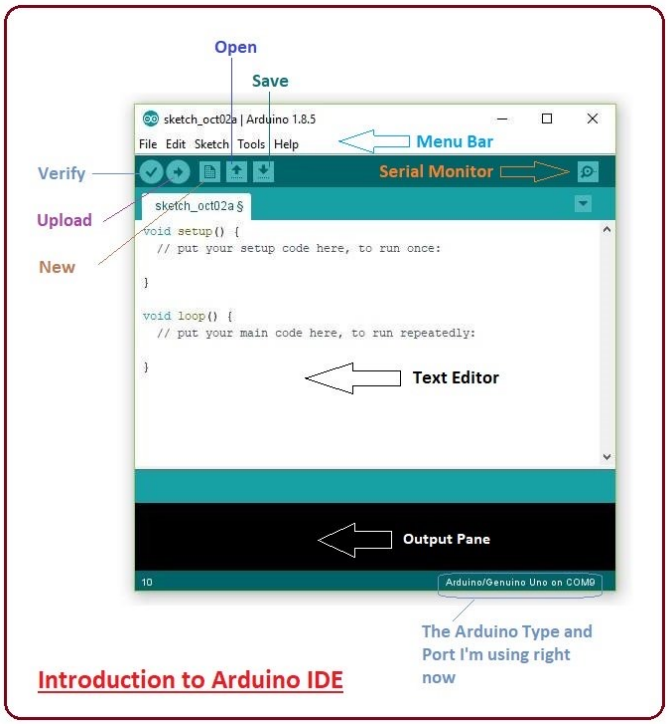
ELECTRICITY IS THE FLOW OF ENERGY THROUGH A CONDUCTIVE MATERIAL.



THE WATER ANALOGY IS COMMONLY USED TO EXPLAIN THESE TERMS. HERE'S ONE MODEL.


OHM'S LAW
CURRENT = VOLTAGE/RESISTANCE
(I = V/R)
OR
RESISTANCE = VOLTAGE/CURRENT
(R = V/I)
OR
VOLTAGE = RESISTANCE * CURRENT
(V = R*I)
THERE IS A RELATIONSHIP BETWEEN VOLTAGE, CURRENT AND RESISTANCE, DISCOVERED BY GEORG OHM, A GERMAN PHYSICIST.
FOR EXAMPLE, INCREASE THE RESISTANCE, LESS FLOW.
OR INCREASE THE POTENTIAL, MORE FLOW.

Coding in Arduino is done in Arduino "sketches" in the Arduino IDE*

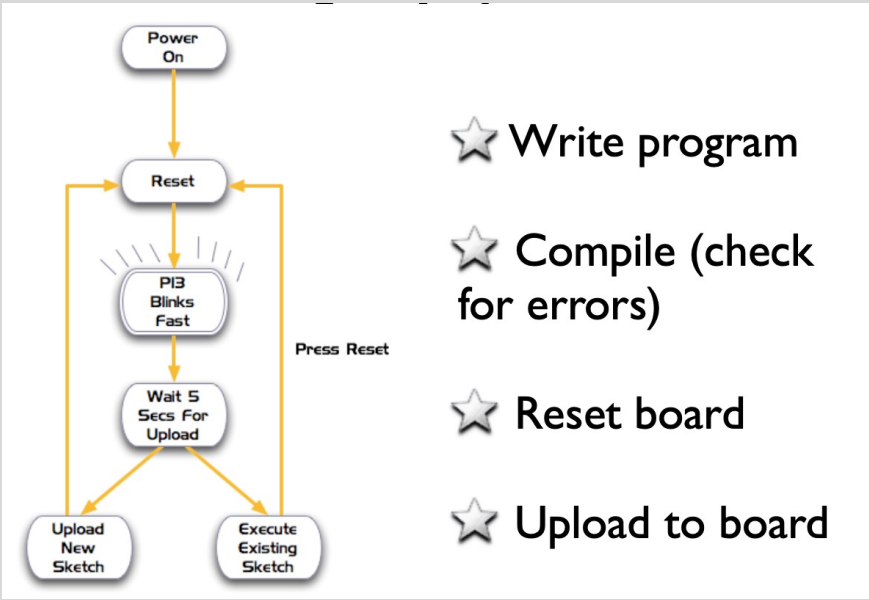


Introduction to Arduino IDE

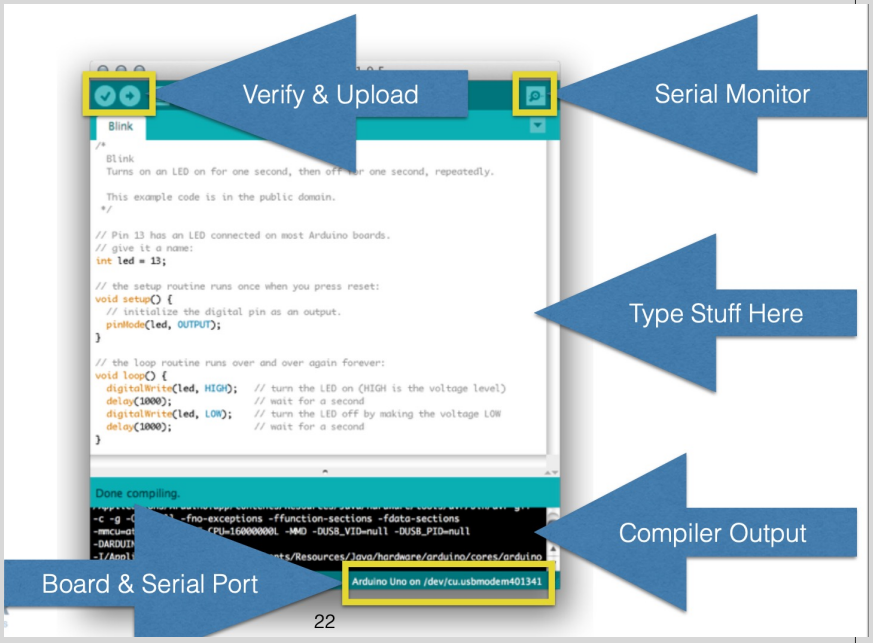
- A Sketch is composed of 3 components
  - Global Variable which are declared at the top
    - Here you define variables and libraries you want to use
  - Setup() is the initialization code you wish you run at the start of the code, this is also where you defines Pins and notify the Arduino where things are connected
  - The last part is the Loop(). This is where you list your functions that you want to run continuously.

*IDE: Integrated Development Environment

Programming an Arduino



- ☆ Write program
- ☆ Compile (check for errors)
- ☆ Reset board
- ☆ Upload to board



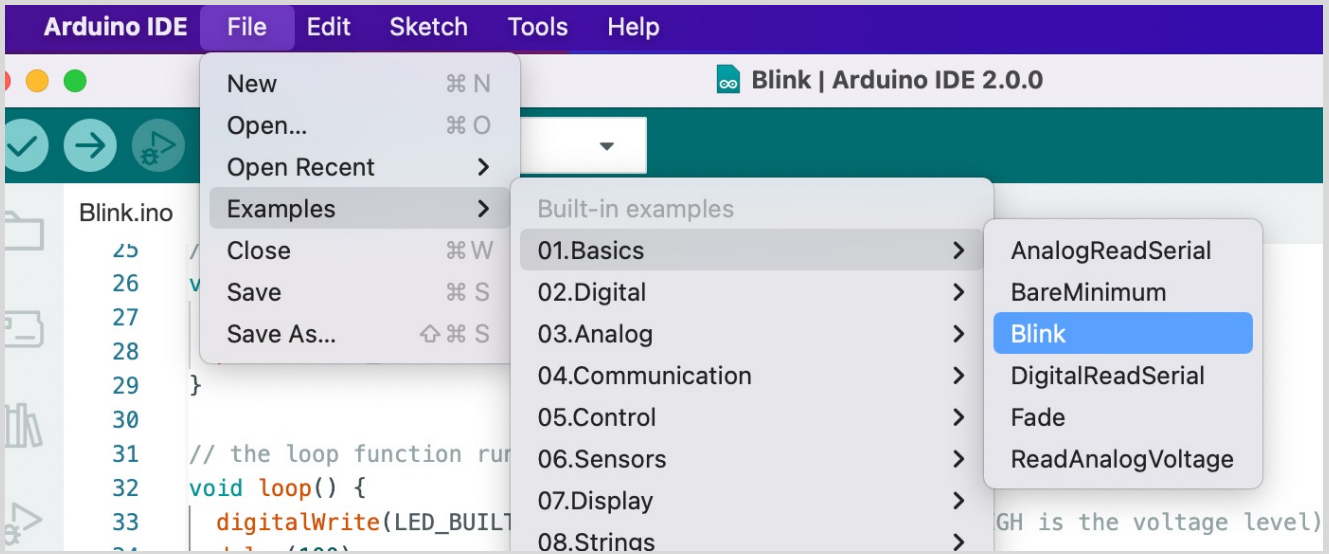
Download Software

https://www.arduino.cc ➔software

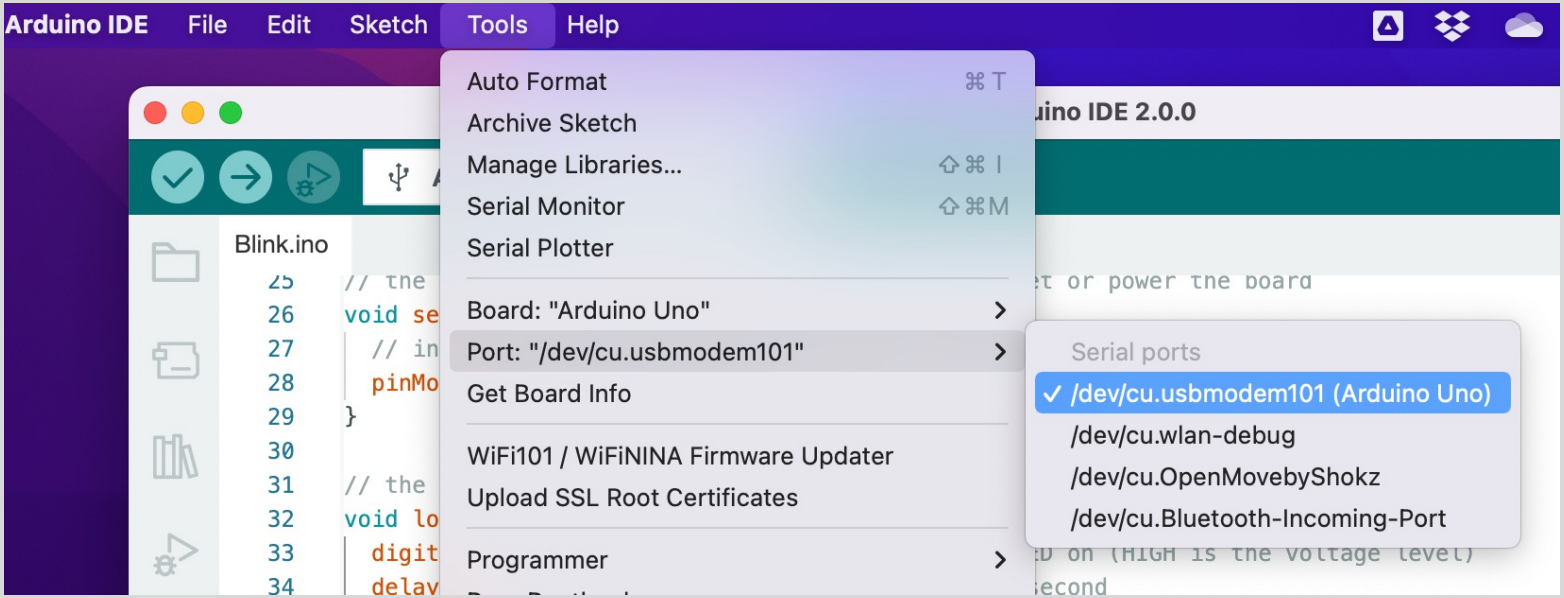
https://www.arduino.cc/en/software

Simple Exercise: BLINK

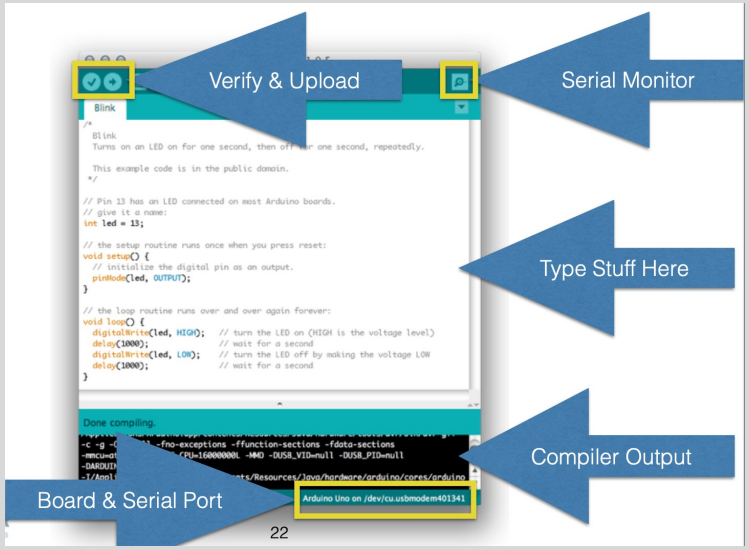
Step 0: Load Blink Example



Blink Step 1: Check the correct board and serial port are selected in the tools menu



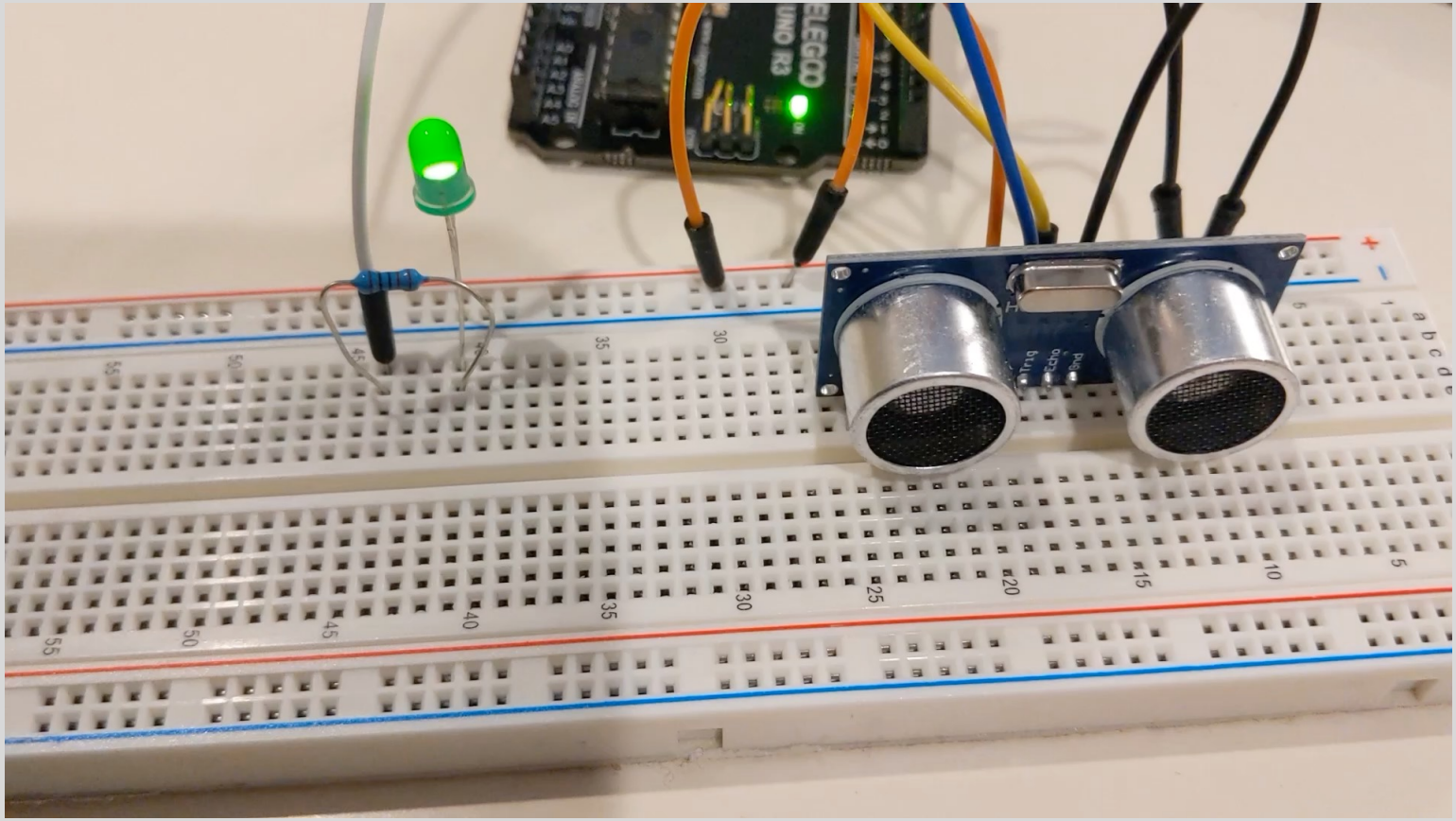
Blink Step 2: Verify & Upload the program



- Change the values of delay() to see how it affects the behavior

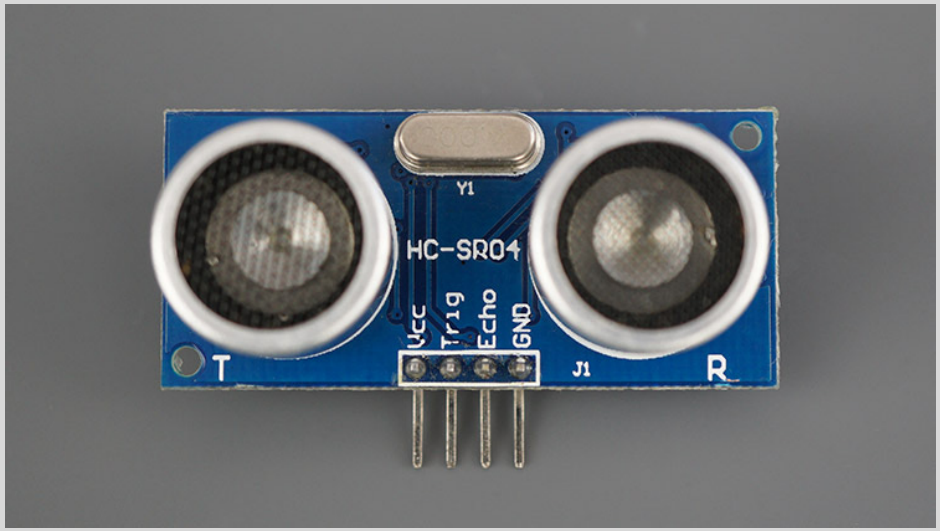
CLASS EXERCISE: ULTRASONIC SENSOR

Determine proximity, Add LED light

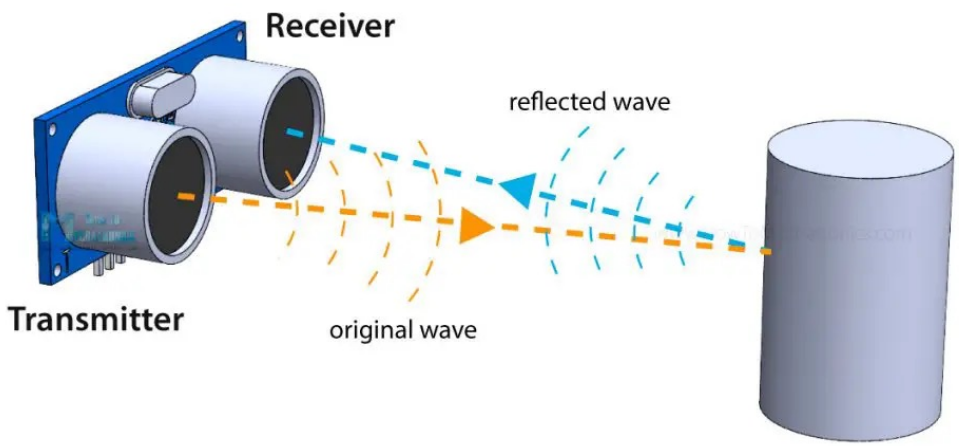

ELEGOO
UNO R3
ON
Pad
Pad
Pad
Gnd
Trig
Echo
+
-
a b c d e
1
5
10
15
20
25
30
35
40
45
50
55

Ultrasonic Sensor HC-SR04

Operating Voltage	5V DC
Operating Current	15mA
Operating Frequency	40KHz
Min Range	2cm / 1 inch
Max Range	400cm / 13 feet
Accuracy	3mm
Measuring Angle	<15°
Dimension	45 x 20 x 15mm



The sensor has 4 pins. ***VCC*** and ***GND*** go to ***5V*** and ***GND*** pins on the Arduino, and the ***Trig*** and ***Echo*** go to any digital Arduino pin. Using the ***Trig*** pin we send the ultrasound wave from the transmitter, and with the ***Echo*** pin we listen for the reflected signal.



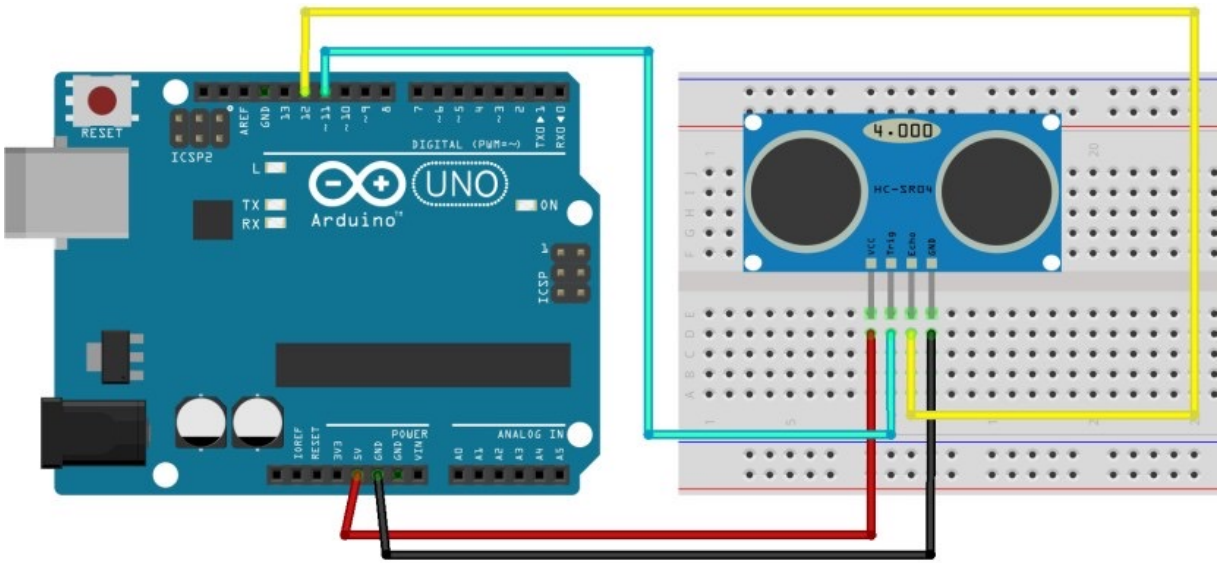
The time between the transmission and reception of the signal allows the calculation of the distance to an object.

We know the sound's velocity in the air:

distance to an object =

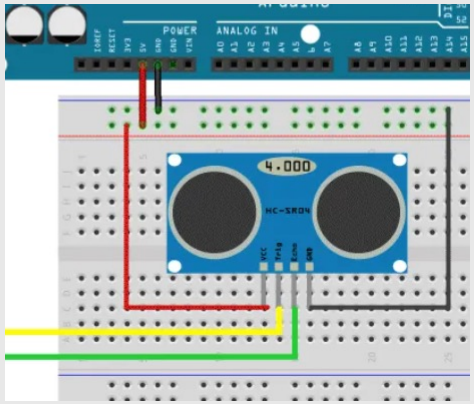
((speed of sound in the air)*time)/2

speed of sound in the air ~**340m/s**



Ultrasonic Sensor HC-SR04	Arduino
VCC	5V
Trig	Pin 11
Echo	Pin 12
GND	GND

Best practice



Ground "-" on breadboard

Voltage "+"

Then wire to Arduino

CODE

```
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
float duration, cm, inches; //for number
storage, 6-7 decimal precision
```

- create variables for the trigger and echo pin called `trigPin connected to Pin 11` and `echoPin to Pin 12`
- three variables of type float: `duration`, `cm` and `inches`. The `duration` variable saves the time between the emission and reception of the signal. The `cm` variable will save the distance in centimeters, and the `inches` variable will save the distance in inches
- Float has 6-7 digit decimal precision

CODE

```
void setup() {
Serial.begin (9600); // initialize serial
port
pinMode(trigPin, OUTPUT); //set arduino
pin to output mode
pinMode(echoPin, INPUT); //set arduino pin
to input mode
}
```

- initialize the serial port at a baud rate of 9600, and set the trigger pin as an `OUTPUT` and the echo pin as an `INPUT`

CODE

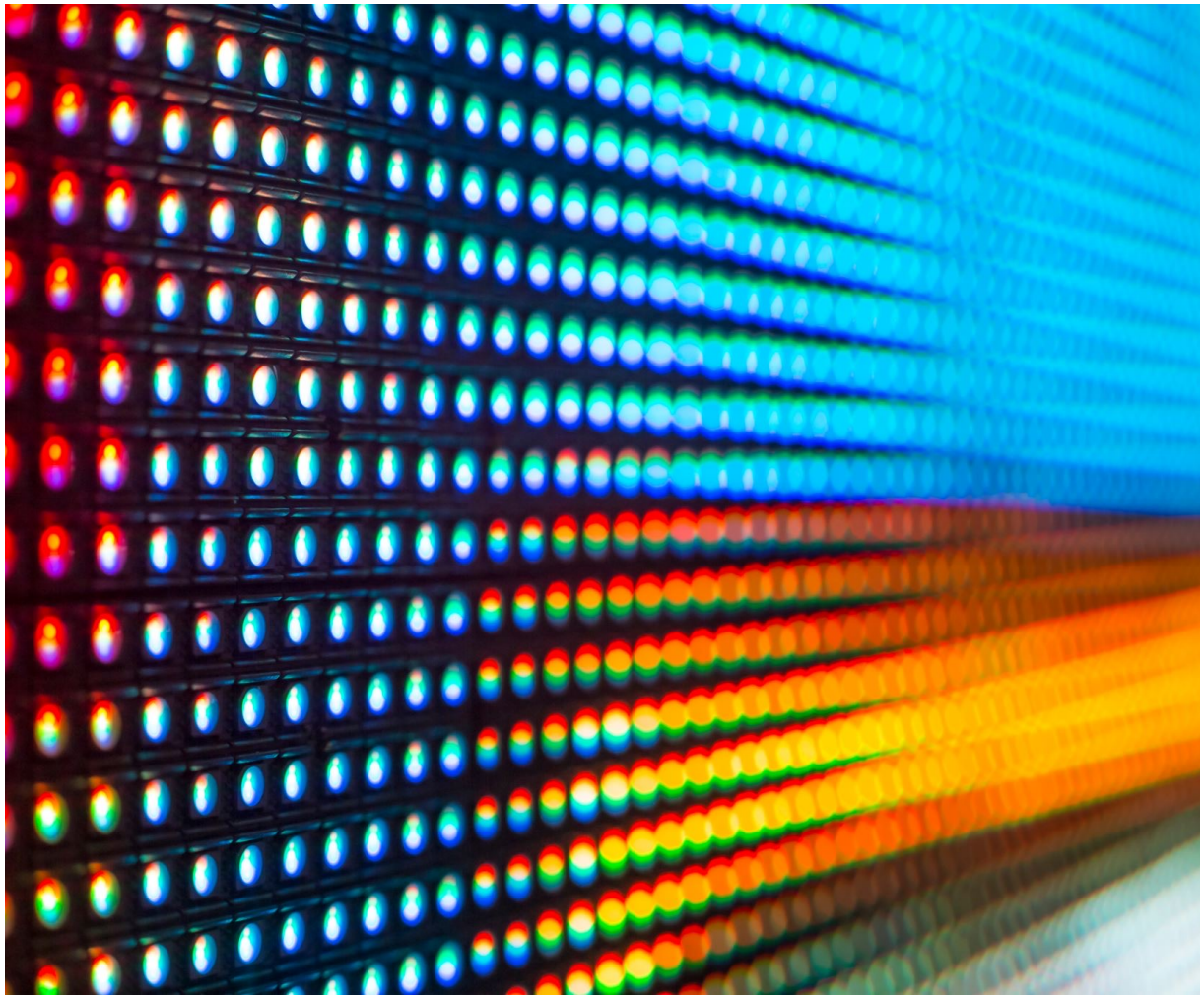
```
void loop() {
// generate 10-microsecond pulse to trigPin
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// measure the duration of pulse from echoPin
duration = pulseIn(echoPin, HIGH);
// Convert the time into a distance
cm = (duration/2) / 29.1; // Divide by 29.1 or multiply by 0.0343
inches = (duration/2) / 74; // Divide by 74 or multiply by 0.0135
```

- Trigger the sensor by sending a `HIGH` pulse of 10 microseconds.
- Use the `pulseIn()` function to get the sound wave travel time
- The `pulseIn()` function reads a HIGH or a LOW pulse on a pin. It accepts as arguments the pin and the state of the pulse (either HIGH or LOW). It returns the length of the pulse in microseconds. The pulse length corresponds to the time it took to travel to the object plus the time traveled on the way back.
- Calculate the distance to an object, taking into account the sound speed.
  - **distance = (traveltime/2) x speed of sound** The speed of sound is: 343m/s = 0.0343 cm/uS = 1/29.1 cm/uS Or in inches: 13503.9in/s = 0.0135in/uS = 1/74in/uS

CODE

- Print the results in the Serial Monitor

```
// print value to Serial Monitor

Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(250);
}
```



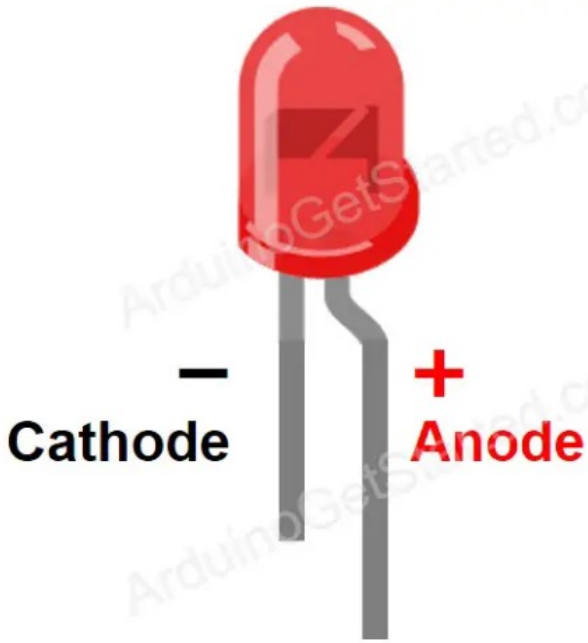
Now Add LED

LED to turn on when object moves close

LED to turn off when object moves away

LED includes two pins:

- Cathode(-) pin: needs to be connected to **GND**
- Anode(+) pin: is used to control LED's state



LED

Long leg on LED is anode (+) and connects to power.

Short leg on LED is cathode (-) and connects to ground

Resistor Information

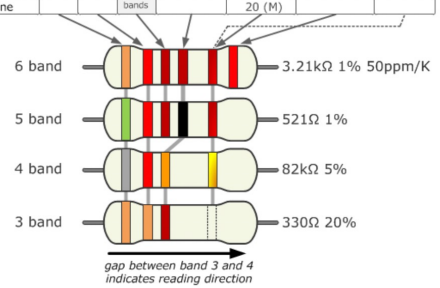
Resistor Color Code Chart

The chart below shows how to determine the resistance and tolerance for resistors. The table can also be used to specify the color of the bands when the values are known. An automatic resistor calculator can be used to quickly find the resistor values.

www.resistorguide.com

	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1 (K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		

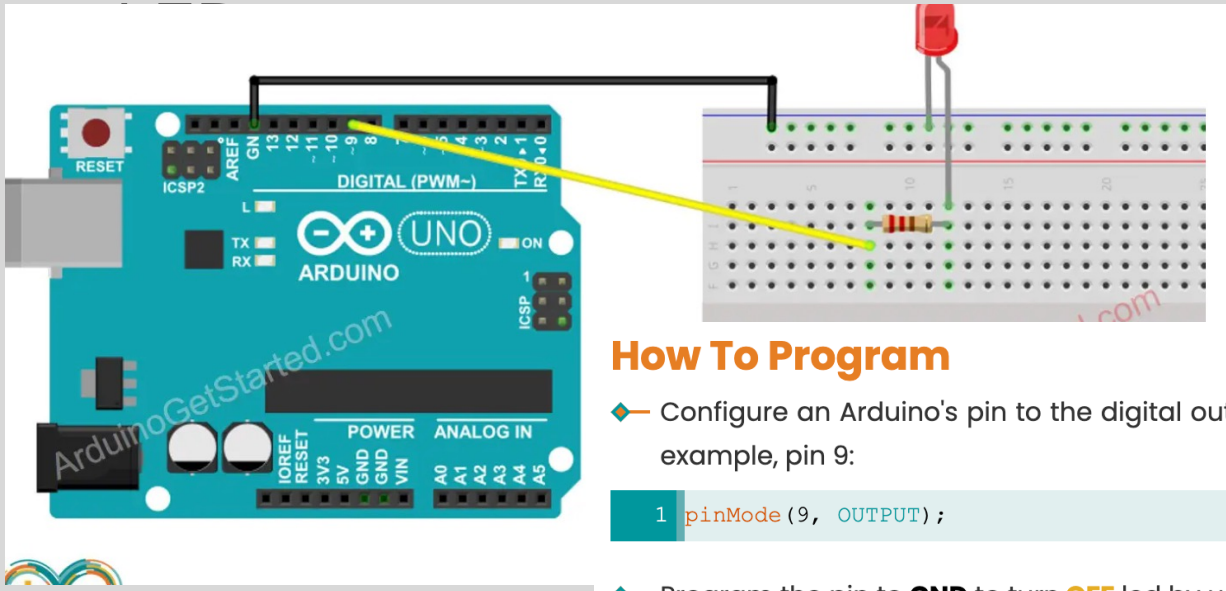
Resistor Color Codes Chart



- The reading direction might not always be clear. Sometimes the increased space between bands 3 and 4 provide an indication of the reading direction. Also, the first band is usually the closest to a lead. A gold or silver band (the tolerance) is always the last band.
- It is a good practice to check the manufacturer's documentation to be sure about the color coding system used.
- When in doubt, measure the resistance with a ohmmeter. In some cases this might even be the only way to figure out the resistance; for example when the color bands are burnt off.

https://eepower.com/resistor-guide/resistor-standards-and-codes/resistor-color-code/#

Add a LED and 220 Ohm resistor



You can use any pin – just make sure you code the correct pin

How To Program

- Configure an Arduino's pin to the digital output mode by using pinMode() function. For example, pin 9:

```
pinMode(9, OUTPUT);
```

- Program the pin to **GND** to turn **OFF** led by using digitalWrite() function:

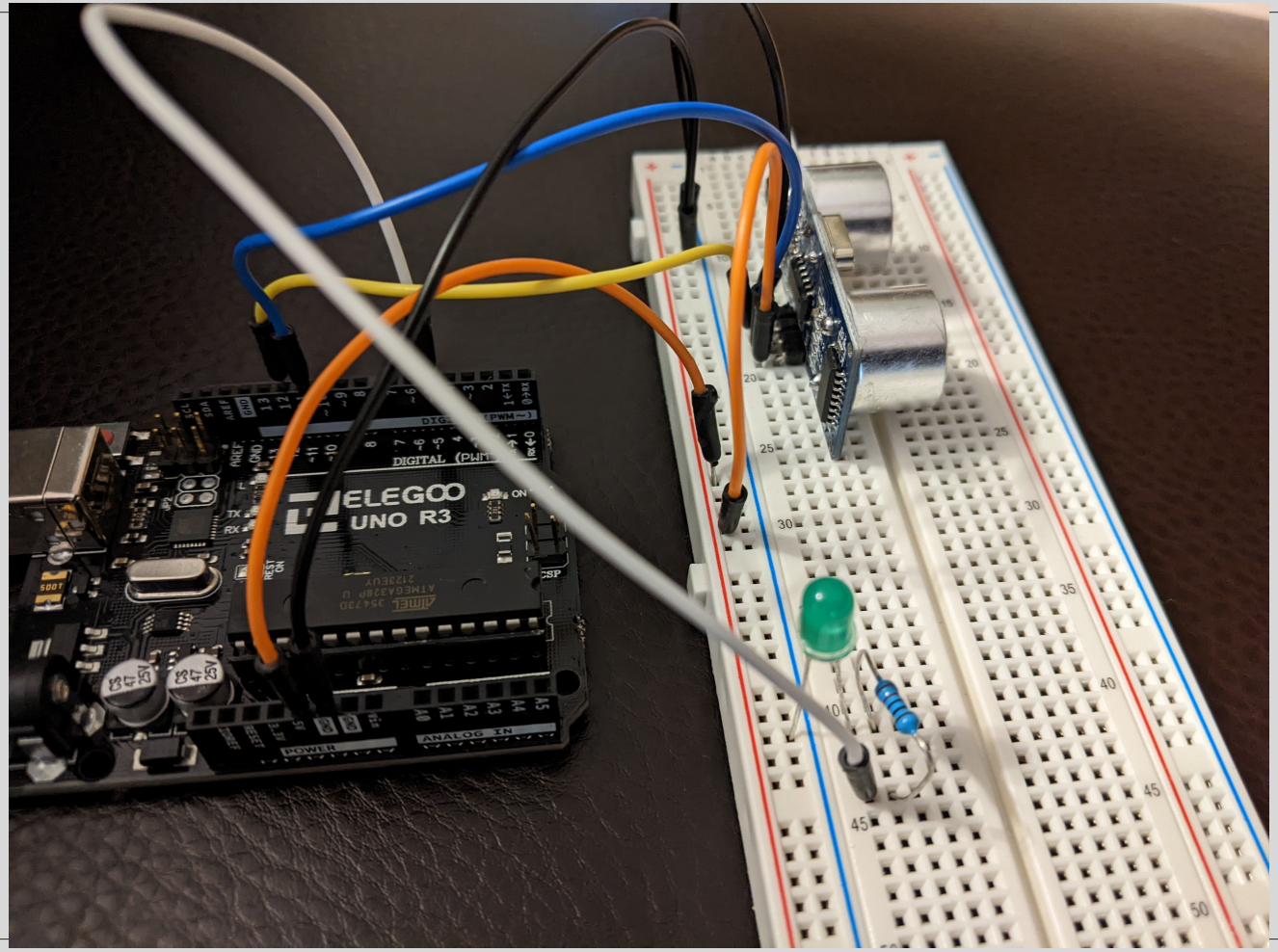
```
digitalWrite(9, LOW);
```

- Program the pin to **VCC** to turn **ON** led by using digitalWrite() function:

```
digitalWrite(9, HIGH);
```

What do you need to add to your code?

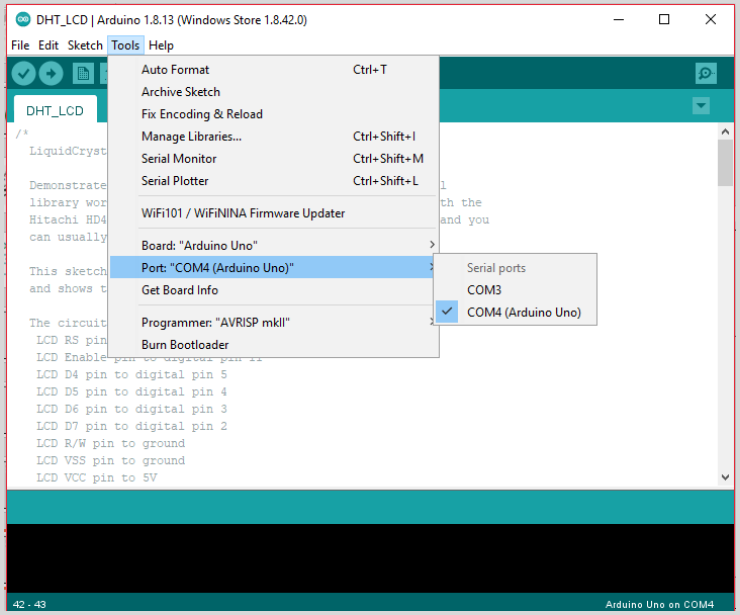
```
int distance_threshold = 15; // Centimeters


pinMode (5, OUTPUT); //set LED pin to output mode


if (cm < distance_threshold)
digitalWrite (LEDPin, HIGH); //Turn on LED
else
digitalWrite (LEDPin, LOW); //Turn off LED
```


ELEGOO
UNO R3
DIGITAL (PWM~)
POWER
ANALOG IN

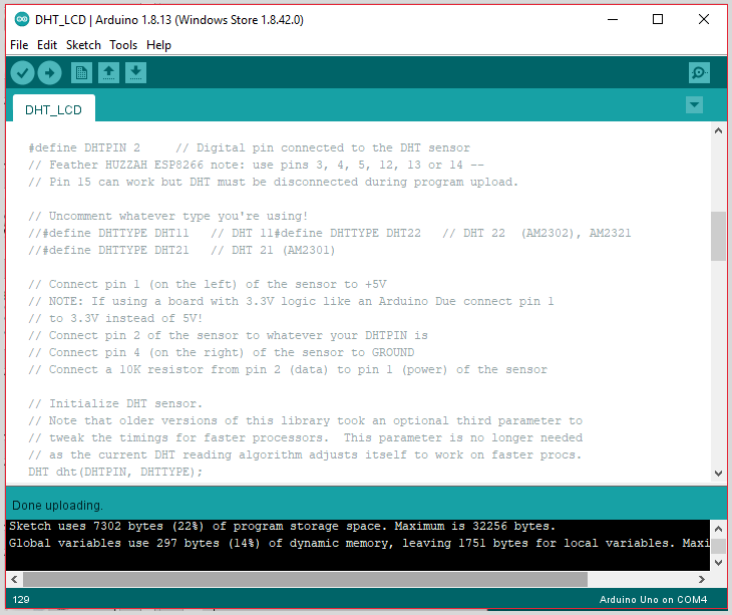
CLASS EXERCISE: TEMPERATURE & HUMIDITY SENSOR, LEDS

Determine temperature and humidity, LED lights depending on temperature

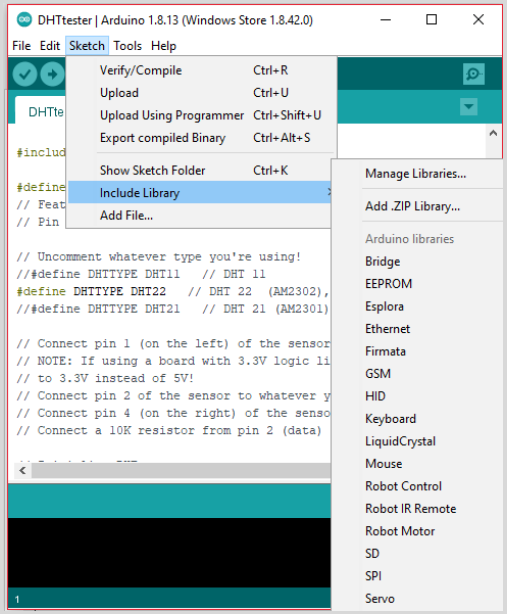
Step 1: Make sure your board is detected by going to Tools-> Port-> Arduino Uno



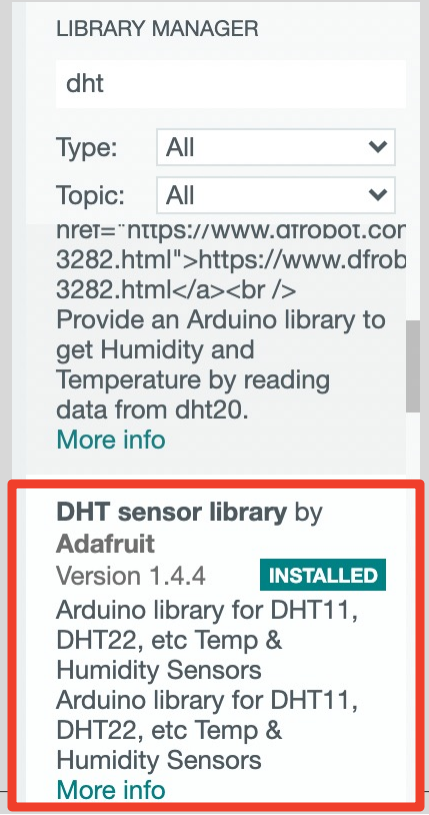
Step 2: Open the DHT_uncommented.ino sketch from canvas



Step 2b: Download the DHT sensor library(Sketch-> Include Library-> Manage Libraries)



Step 2c: Search for DHT and install the latest version



DHT11 Humidity And Temperature Sensor

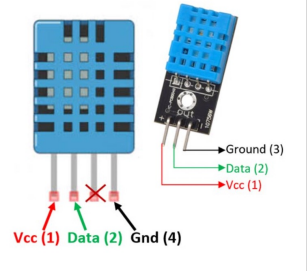
- Measures water vapor by measuring the electrical resistance between two electrodes.
  - Change in resistance between the electrodes is proportional to RH
  - Higher (lower) RH decreases (decreases) the resistance between the electrodes
- Measures temperature with surface mounted NTC temperature sensor (thermistor)

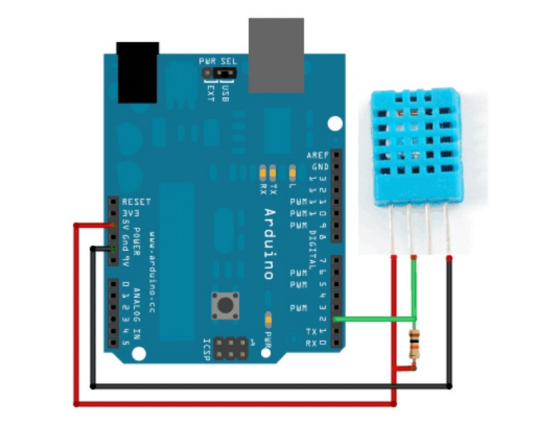
$$RH = (\frac{\rho_w}{\rho_s})\ x\ 100\%$$

$RH:\ Relative\ Humidity$
$\rho_w:\ Density\ of\ water\ vapor$
$\rho_s:\ Density\ of\ water\ vapor\ at\ saturation$

DHT11 Humidity And Temperature Sensor

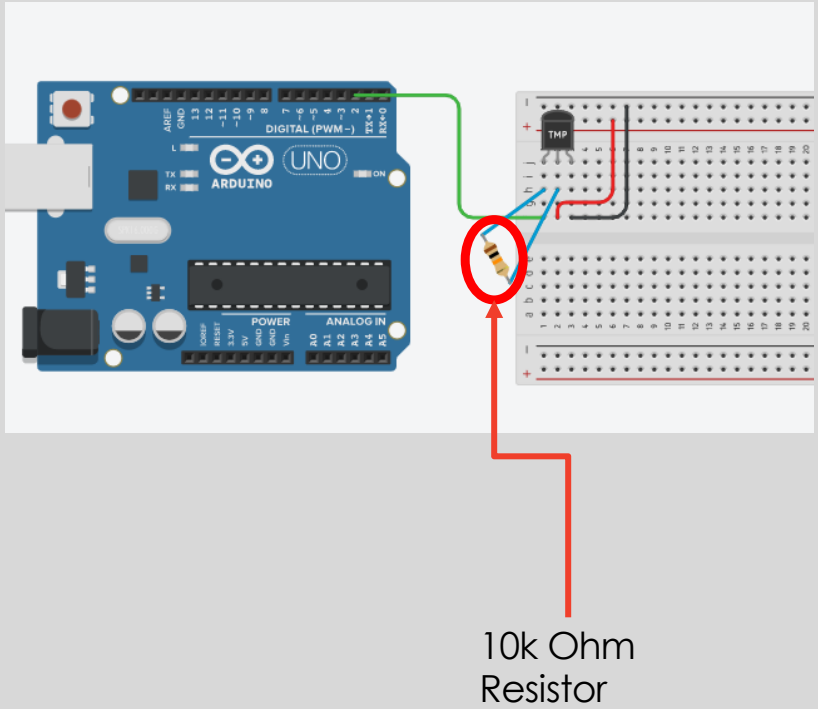
- Uses 1 signal wire to transmit data to the Arduino
- Power from separate 5v and Ground
- 10K Ohm resistor needed between signal and 5V(Vcc) to make sure signal level stays high





*sensor datasheets will tell you this information

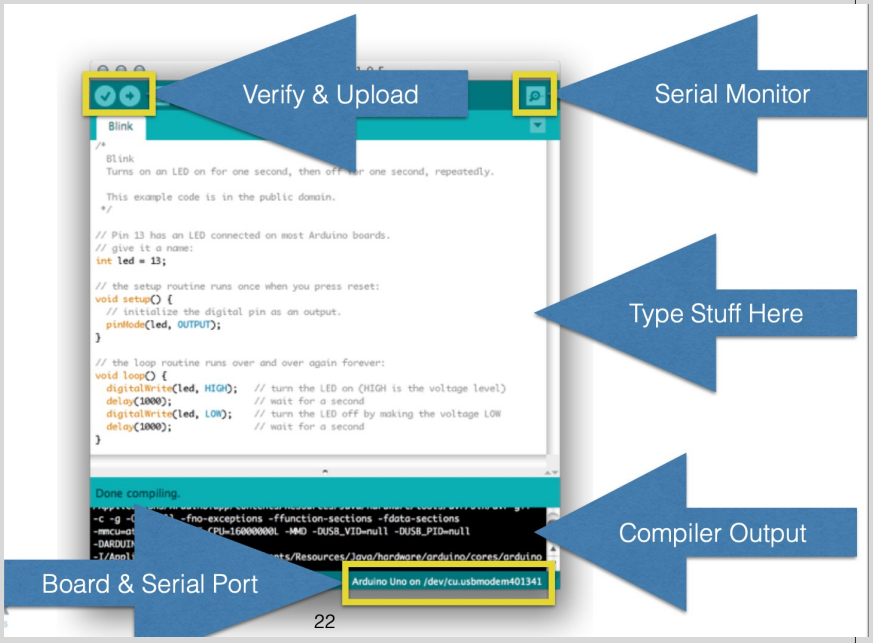
Step 3: Wiring Our Temperature and Humidity Sensor



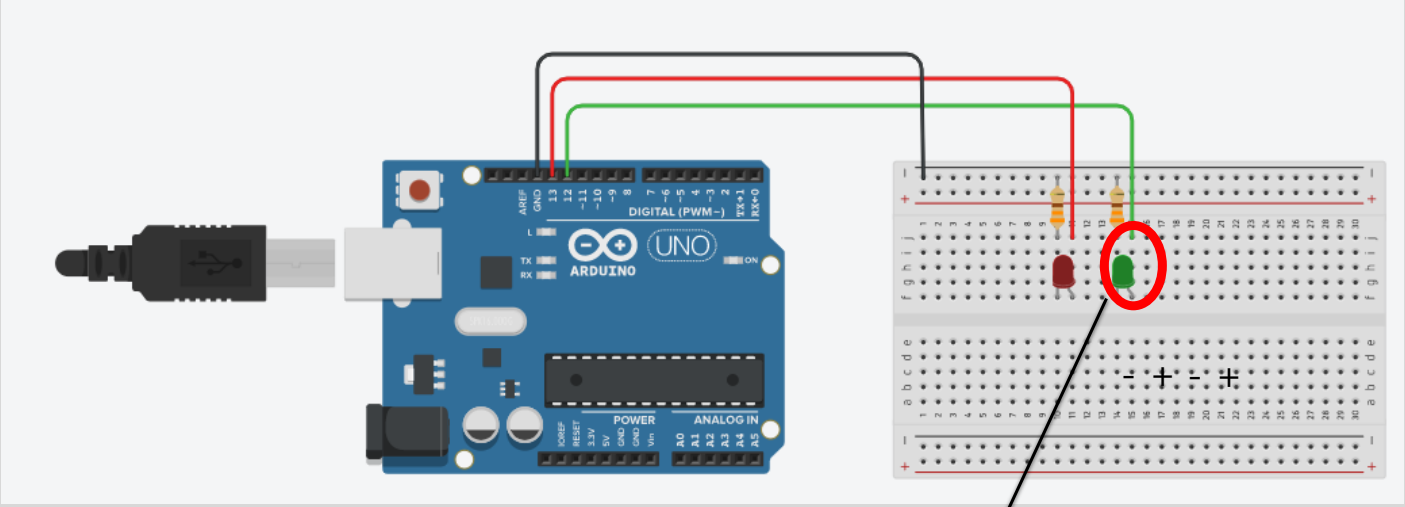
- Connect the left most pin to Digital pin 2
- Connect a 10k resistor from the left most pin to center pin
- Connect the center pin to the 5V track
- Connect the right most pin to the ground track.

Step 4: Confirm Everything is Working

- Make sure you Port is selected
- Verify
- Upload
- Open Serial Monitor
- Read Temperature and Humidity
- What happens if you blow air onto the sensor?



Step 5: Wiring LED to Arduino, repeat for Green LED



Long leg on LED is anode (+) and connects to power.
Short leg on LED is cathode (-) and connects to ground

330 Ohm Resistor
What happens if you use a 220 Ohm Resistor?

Resistor Information

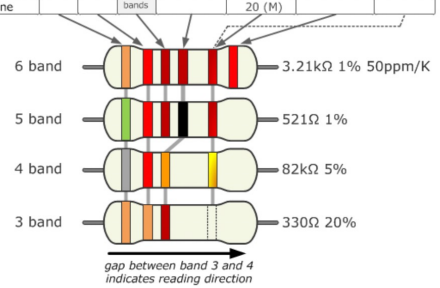
Resistor Color Code Chart

The chart below shows how to determine the resistance and tolerance for resistors. The table can also be used to specify the color of the bands when the values are known. An automatic resistor calculator can be used to quickly find the resistor values.

www.resistorguide.com

	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1 (K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		

Resistor Color Codes Chart



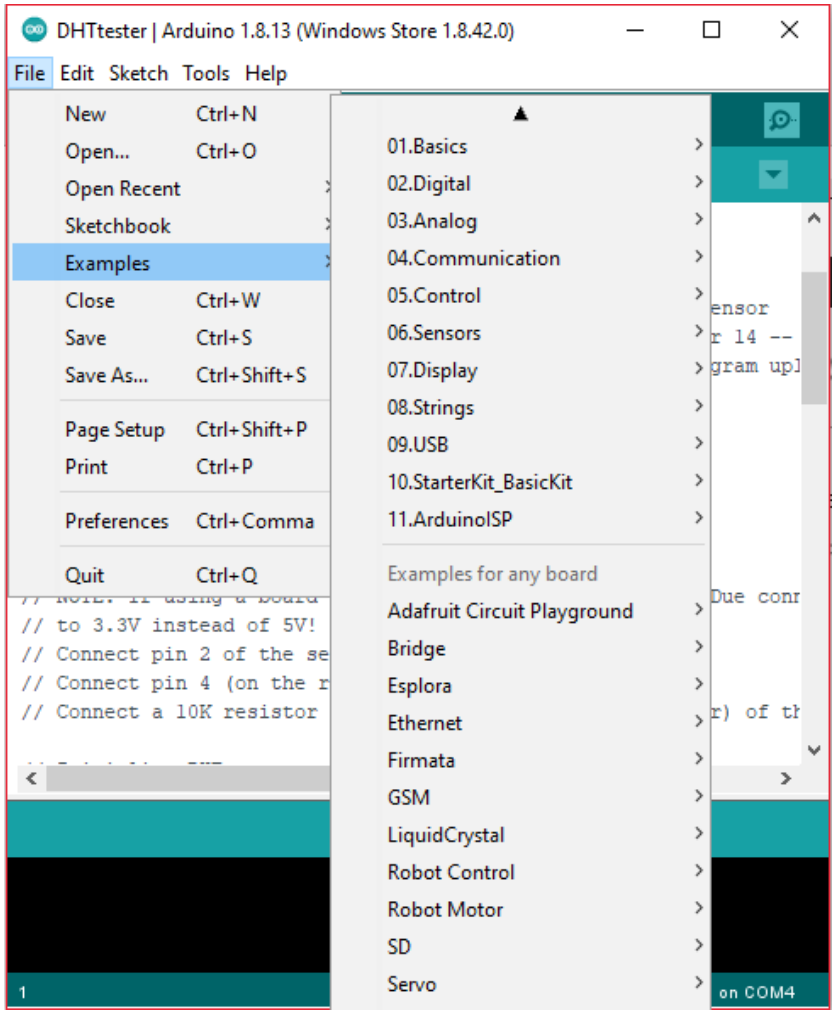
- The reading direction might not always be clear. Sometimes the increased space between bands 3 and 4 provide an indication of the reading direction. Also, the first band is usually the closest to a lead. A gold or silver band (the tolerance) is always the last band.
- It is a good practice to check the manufacturer's documentation to be sure about the color coding system used.
- When in doubt, measure the resistance with a ohmmeter. In some cases this might even be the only way to figure out the resistance; for example when the color bands are burnt off.

https://eepower.com/resistor-guide/resistor-standards-and-codes/resistor-color-code/#

Step 6: Lets comment our code and upload it

```
#include "DHT.h"

#define DHTPIN 2     // Set the Temperature sensor pin

#define DHTTYPE DHT11   // Define sensor type

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600); // Start a serial monitor section
  Serial.println("Testing Temp"); // Initialization of serial monitor
  pinMode(12,OUTPUT); // Define Red LED output pin
  pinMode(13,OUTPUT); // Define Green LED output pin
  dht.begin(); // Start the DHT sensor
}
```

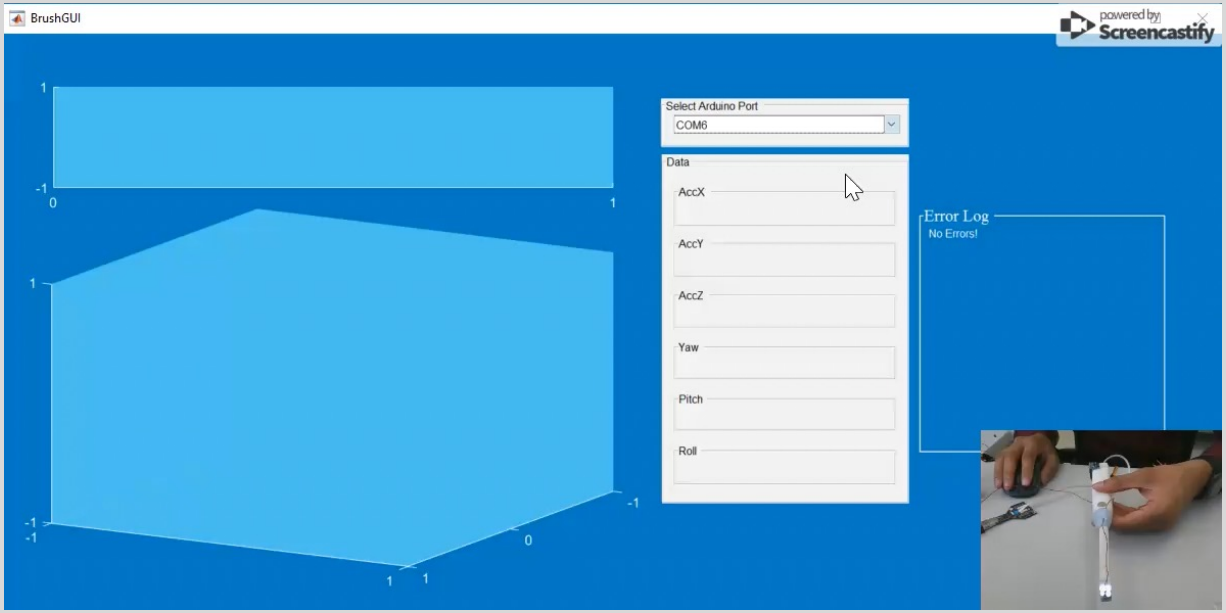
```
void loop() {
  delay(500);
  float h = dht.readHumidity(); // collect humidity measurements after 500ms

  float t = dht.readTemperature(); //collect degrees celsius

  float f = dht.readTemperature(true); // collect degrees fahrenheit
  Serial.print(("Temperature is "));
  Serial.print(f);
  Serial.println(("°F ")); // Print temperature on serial monitor
  if (isnan(h) || isnan(t) || isnan(f)) { //Catches errors during DHT collection
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  if (f > 74) { // If degree fahrenheit is above 74, turn of green light, otherwise, turn on red light
    digitalWrite(12,HIGH);
    digitalWrite(13,LOW);
  } else {
    digitalWrite(13,HIGH);
    digitalWrite(12,LOW);

  delay(1500); //wait 1.5 seconds
  }
}
```



Where to get started on your own assignment

- Look at examples that come with every library
- Look at the Arduino community page to find new ideas
- Make something interesting by combining the sensors you have

Example of the power of Arduino



Resources

- Circuitio.io
- https://www.arduino.cc/en/Tutorial/HomePage
- https://ladyada.net/learn/arduino/
- https://www.circuitbasics.com/arduino/
- https://create.arduino.cc/projecthub/
- https://www.arduino.cc/en/Guide
- https://arduinogetstarted.com/

Electronics

- http://hyperphysics.phy-astr.gsu.edu/hbase/electric/ecircon.html

  Covers the basics of electric circuits (Ohm's Law, power formulas, basic circuit symbols & design)

- https://app.knovel.com/web/view/khtml/show.v/rcid:kpBSAE0006/cid:kt007RK1G3/viewerType:khtml//root_slug:6-electronics/url_slug:electronics?kpromoter=marc&b-toc-cid=kpBSAE0006&b-toc-root-slug=&b-toc-url-slug=electronics&b-toc-title=Building%20Scientific%20Apparatus%20(4th%20Edition)&page=25&view=collapsed&zoom=1

  Electronics chapter from "Building Scientific Apparatus"- generally a great resource for lab research that's available electronically at Tisch. It goes well beyond the scope of BME66 but covers a lot of practical aspects as well (e.g., how to read the stripes on resistors, or how to connect the wire leads of an LED lamp).