
THODE MAKERSPACE
INTRO TO ARDUINO

WORKSHOP
LEARNING OUTCOMES:

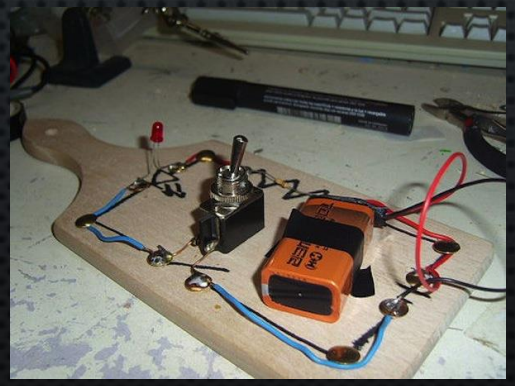
In this presentation, you will:

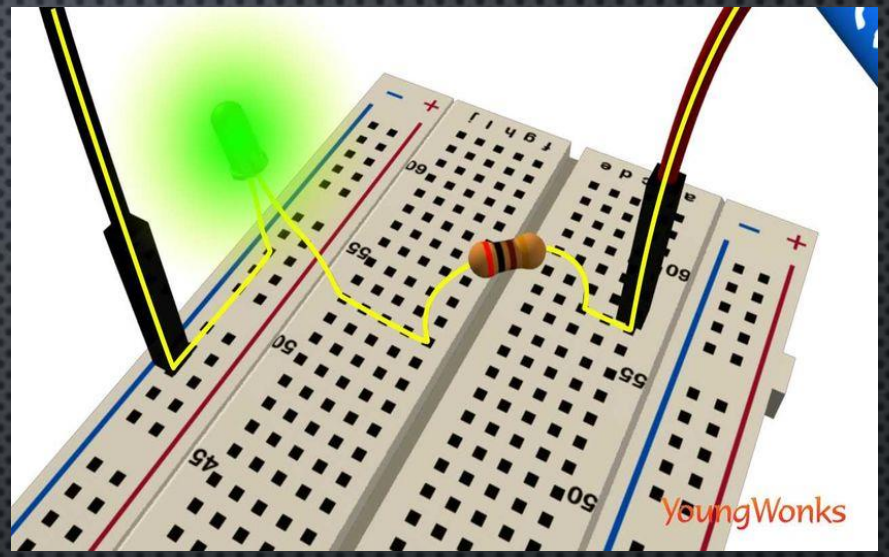
1. Learn about Arduino
2. Cover basic Arduino programing concepts
3. Practice using Arduino IDE with different exercises
4. Learn about how to improve and further expand on this knowledge

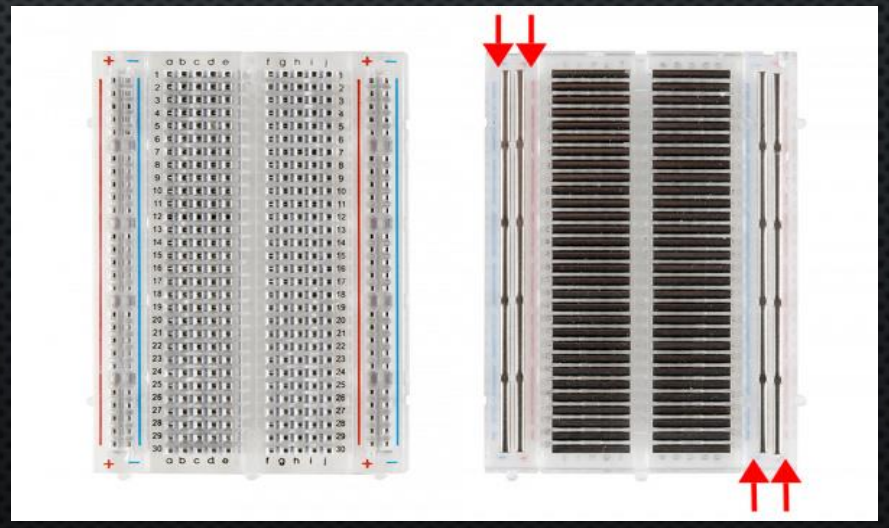
- Code: https://github.com/SyedwRehman/ArduinoWS

CIRCUITRY BASICS

- How Does A Breadboard work?:
- A Breadboard allows you to prototype your electronic circuits by using its different
- Projects ranging from changing colors for RGB LEDS to making an automatic fan when it gets too hot.
- Fun Fact: it's called a breadboard because back in the day they used to make the circuits on actual bread boards with wire connections

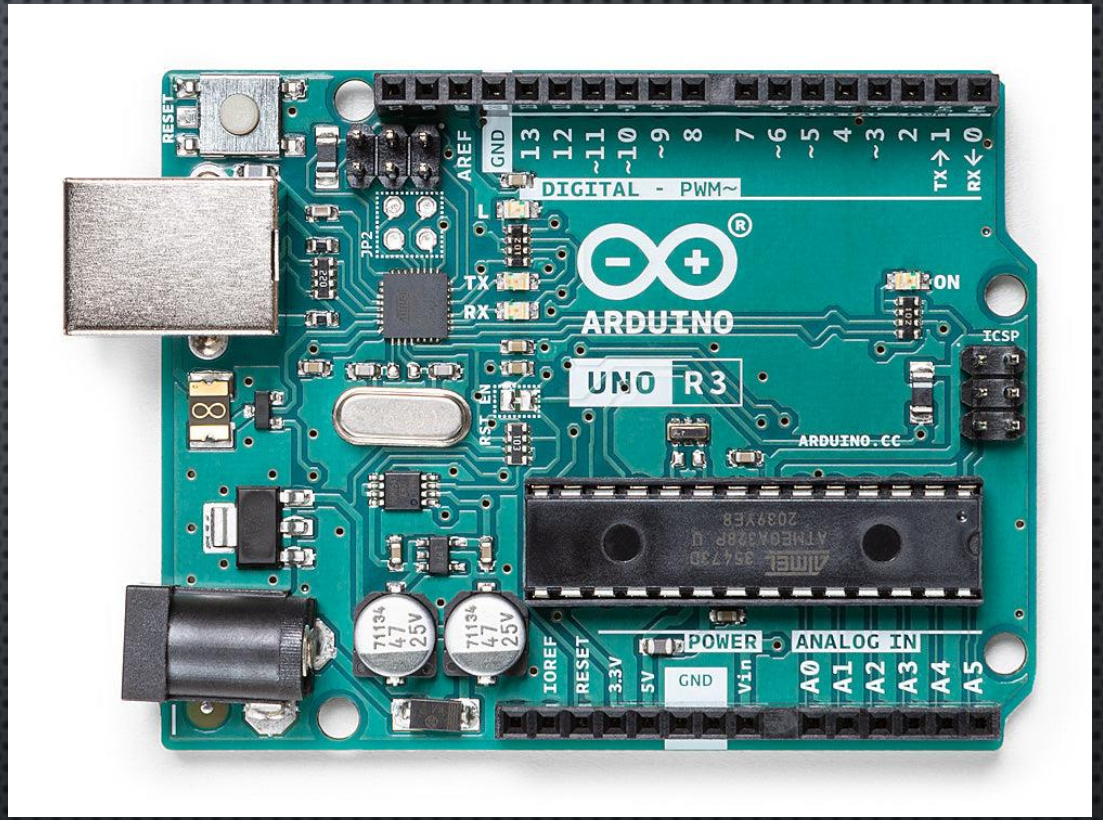






ARDUINO INTRO

- **What is an Arduino?:**
- An Arduino can be thought of as a minicomputer controlling electronic components.
- **What can you use it for?:**
- Projects ranging from changing colors for RGB LEDS to making an automatic fan when it gets too hot.
- **How can you use it?:**
- Arduino can be used to give power to different electronic components or even coding the Arduino to get those components to do a certain task like spinning a motor

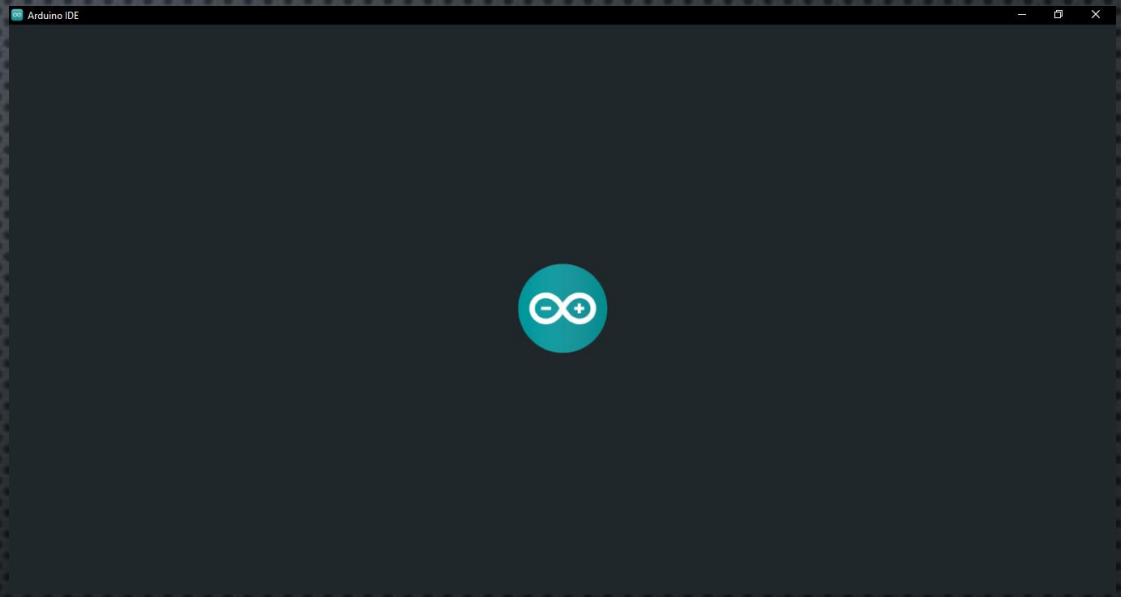


Arduino Equipment in Thode MakerSpace

- At MakerSpace we have an Arduino kit you can sign out
- The Arduino kit consists of the following:
  - A bag consisting of a few 5mm Red LEDs and a couple of jumper wires
  - A USB to USB 2.0 Type-B converter
  - An Arduino UNO R3
  - And a small Breadboard
- NOTE: you must sign out the kits from MakerSpace using the library system

ARDUINO SOFTWARE

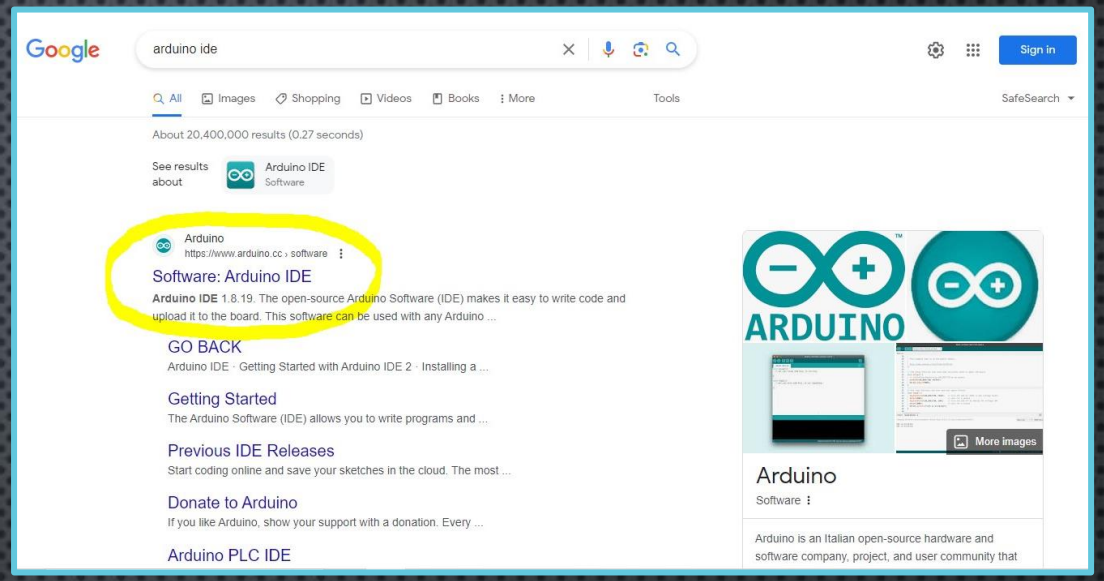
- Arduino Has its own language which is a derivation from the C & C++ language
  - Low level language
  - Better performance for the hardware
- We will cover basic concepts and some of their exceptions to help you get started
- To write the code for the Arduino we will need to install the Arduino IDE

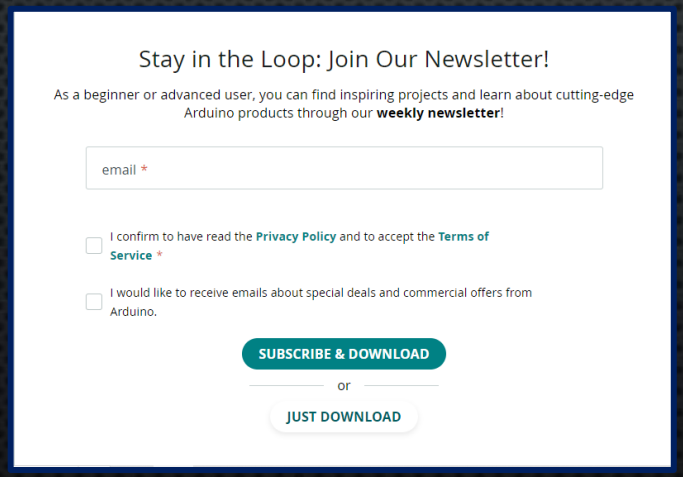


QUICK GUIDE:
HOW TO INSTALL ARDUINO SOFTWARE

ARDUINO IDE DOWNLOAD

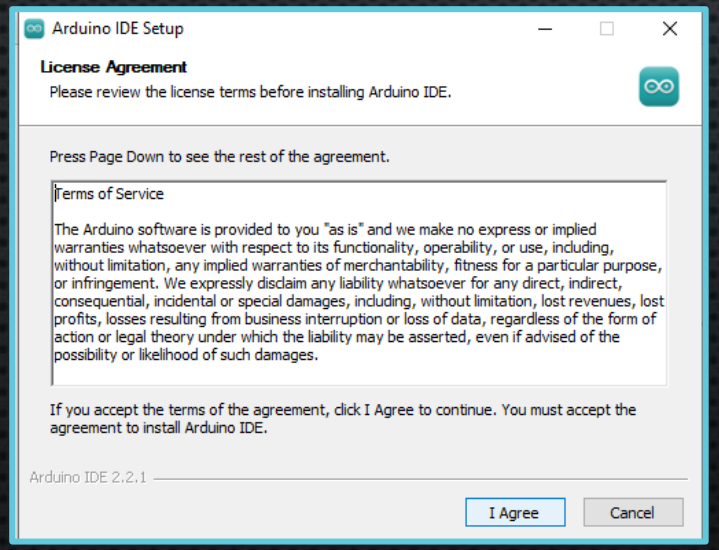
1. SEARCH ARDUINO IDE & FIND DOWNLOAD LINK
   LINK: HTTPS://WWW.ARDUINO.CC/EN/SOFTWARE
2. DOWNLOAD THE NEWEST VERSION
3. CLICK "JUST DOWNLOAD" TO GET THE SOFTWARE

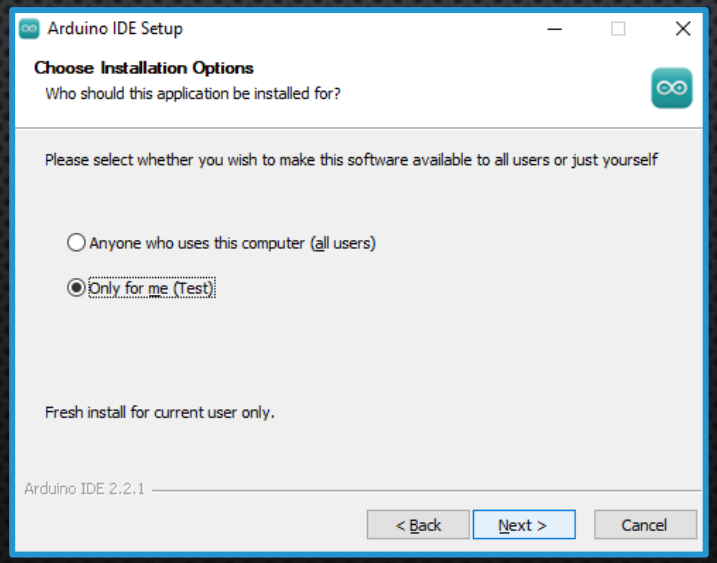


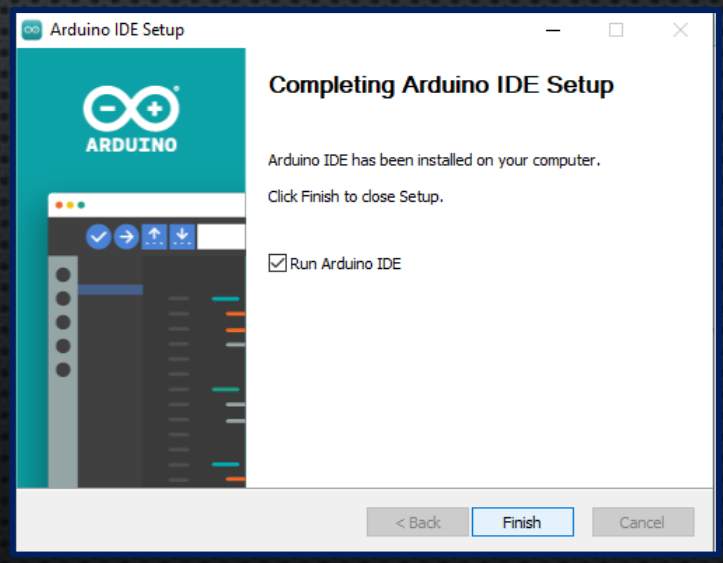


ARDUINO IDE DOWNLOAD

4. START THE FILE
5. GO THROUGH THE SETUP FOR INSTALLING
6. START THE IDE

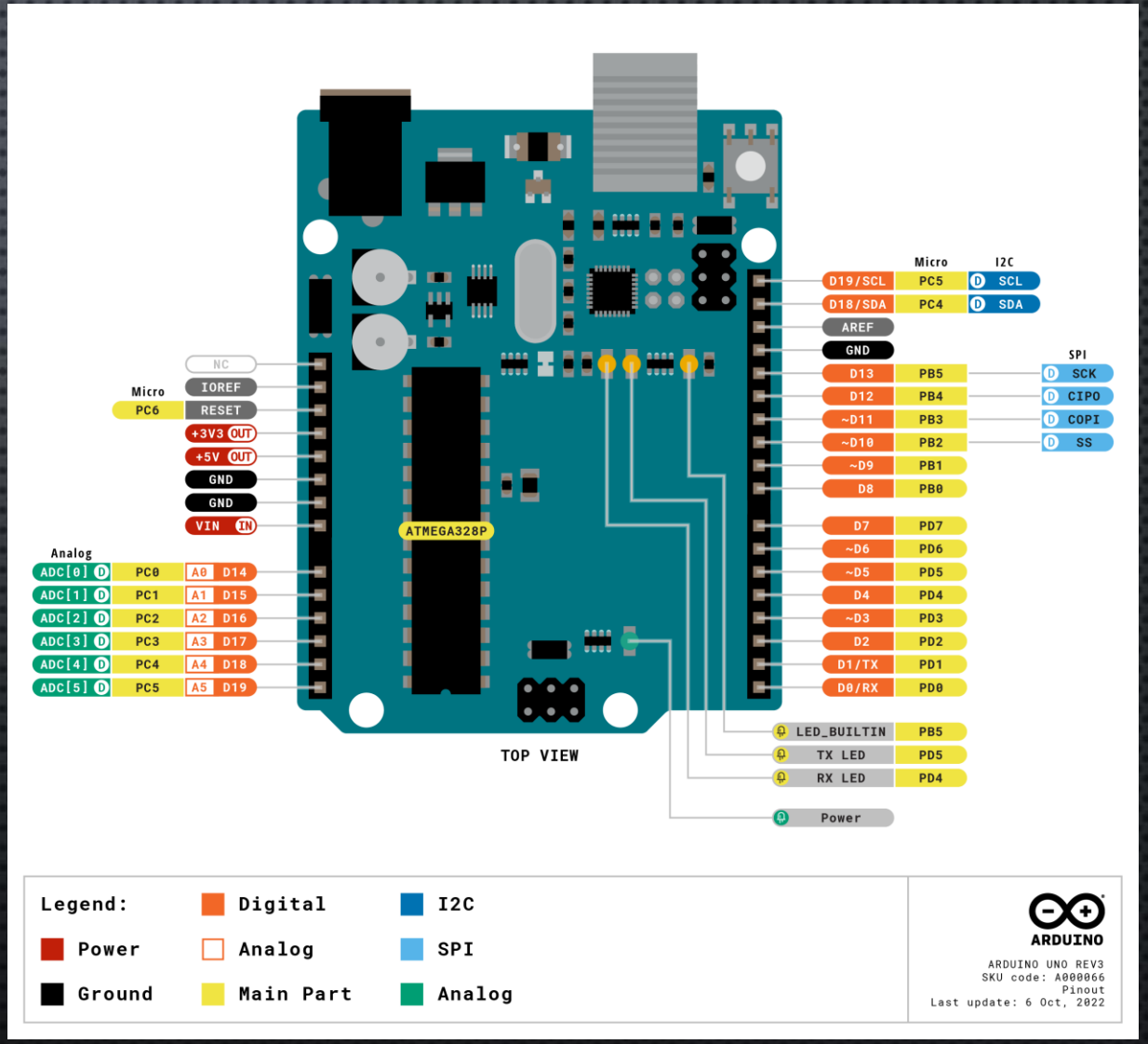






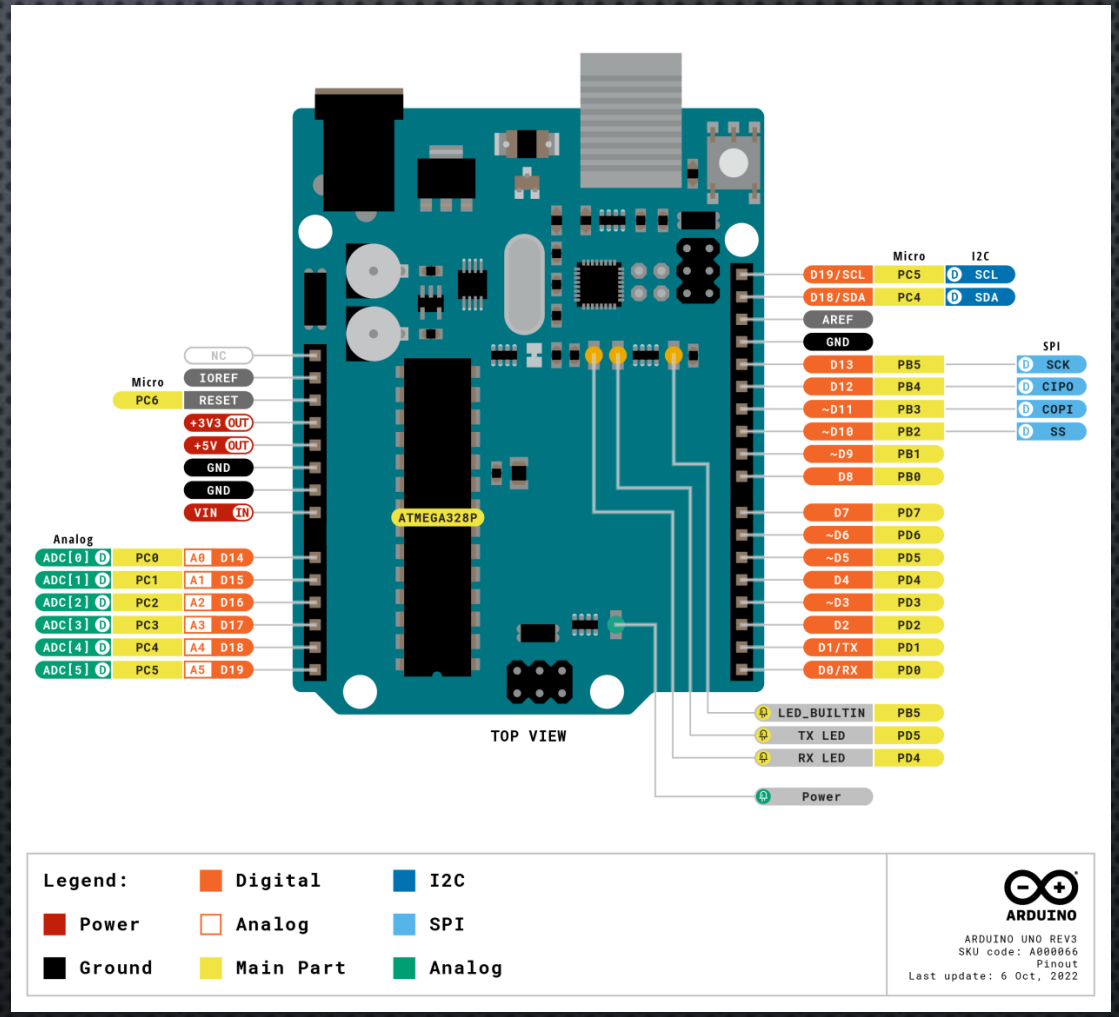
ARDUINO PIN BREAKDOWN

- The Arduino has power and ground pins to be used as a power supply
- The Arduino has digital pins to be used to turn different components on or off at different instances
- It has analog pins which can be used to read or send information to other components
- It has a built in LED which can be used by changing the code
- It has other pins which can be useful for certain specific cases, but we will only be covering the first few.



ARDUINO PIN BREAK DOWN
WHAT YOU NEED TO KNOW

- USING THE BUILT IN LED FOR THE FIRST EXERCISE
- ONLY USING THE DIGITAL PINS (1-13) AS INPUT AND OUTPUT
- USING GROUND FROM THE ARDUINO BOARD
- CAN USE THE POWER PORTS TO SUPPLY POWER TO SMALLER COMPONENTS
- CAN USE ANALOG PINS TO SEND AND RECEIVE DATA FROM COMPONENTS



KEY CODING CONCEPTS

- VARIABLE DATA TYPES : INT, FLOAT, STR,
- INT NUMBER HOLDER = 5; //*NO DECIMALS*
- FLOAT DECIMALS = 5.5431;
- CHAR CHARACTERS = "A"
- STR STRING OF CHARACTERS = "HELLO";
- VOID : NO OUTPUT
- END LINES OF CODE WITH SEMICOLON ";"

Data Type	Size (Bytes)	Range of Values
void	0	null
bool/boolean	1	True/False
char	1	-128 to +127
unsigned char	1	0 to 255
byte	1	0 to 255
int	2	-32,768 to 32,767
unsigned int	2	0 to 65,535
word	2	0 to 65,535
long	4	-2,147,483,648 to 2,147,483,647
unsigned long	4	0 to 4,294,967,295
float	4	-3.4028235E+38 to 3.4028235E+38
double	4	-3.4028235E+38 to 3.4028235E+38
string	-	character array

mbedgeek.blogspot.com

```
char Str1[15];
char Str2[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'};
char Str3[8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\0'};
char Str4[] = "arduino";
char Str5[8] = "arduino";
char Str6[15] = "arduino";
```

KEY CODING CONCEPTS

- Arduino language is a build up of the C/C++ coding language
- Any prior coding concepts are beneficial for you to know
- ** Required for every Arduino project
- * Good to know for other projects
- Variable Types: int, float, str,
- End lines of code with semicolon ";"

Functions to Note:

- void setup() **
- void Loop() **
- pinMode()*
- digitalWrite()*
- digitalRead()*
- analogWrite()*
- analogRead()*
- delay()*
- millis()*

KEY CODING CONCEPTS

setup()

Last revision · 05/16/2024

Description

The `setup()` function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The `setup()` function will only run once, after each powerup or reset of the Arduino board.

loop()

Last revision · 05/16/2024

Description

After creating a `setup()` function, which initializes and sets the initial values, the `loop()` function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

digitalWrite()

Last revision · 05/15/2024

Description

Write a `HIGH` or a `LOW` value to a digital pin.

If the pin has been configured as an `OUTPUT` with `pinMode()`, its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for `HIGH`, 0V (ground) for `LOW`.

pinMode()

Last revision · 05/15/2024

Description

Configures the specified pin to behave either as an input or an output.

See the **Digital Pins** page for details on the functionality of the pins.

It is possible to enable the internal pullup resistors with the mode `INPUT_PULLUP`. Additionally, the `INPUT` mode explicitly disables the internal pullups.

Syntax

```
pinMode(pin, mode)
```

delay()

Last revision · 05/15/2024

Description

Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)

Syntax

```
delay(ms)
```

Functions to Note:

- void setup() **
- void Loop() **
- pinMode()*
- digitalWrite()*
- digitalRead()*
- analogWrite()*
- analogRead()*
- delay()*
- millis()*

Home / Programming / **Language Reference**

Language Reference

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

Functions | Variables | Structure

For controlling the Arduino board and performing computations.

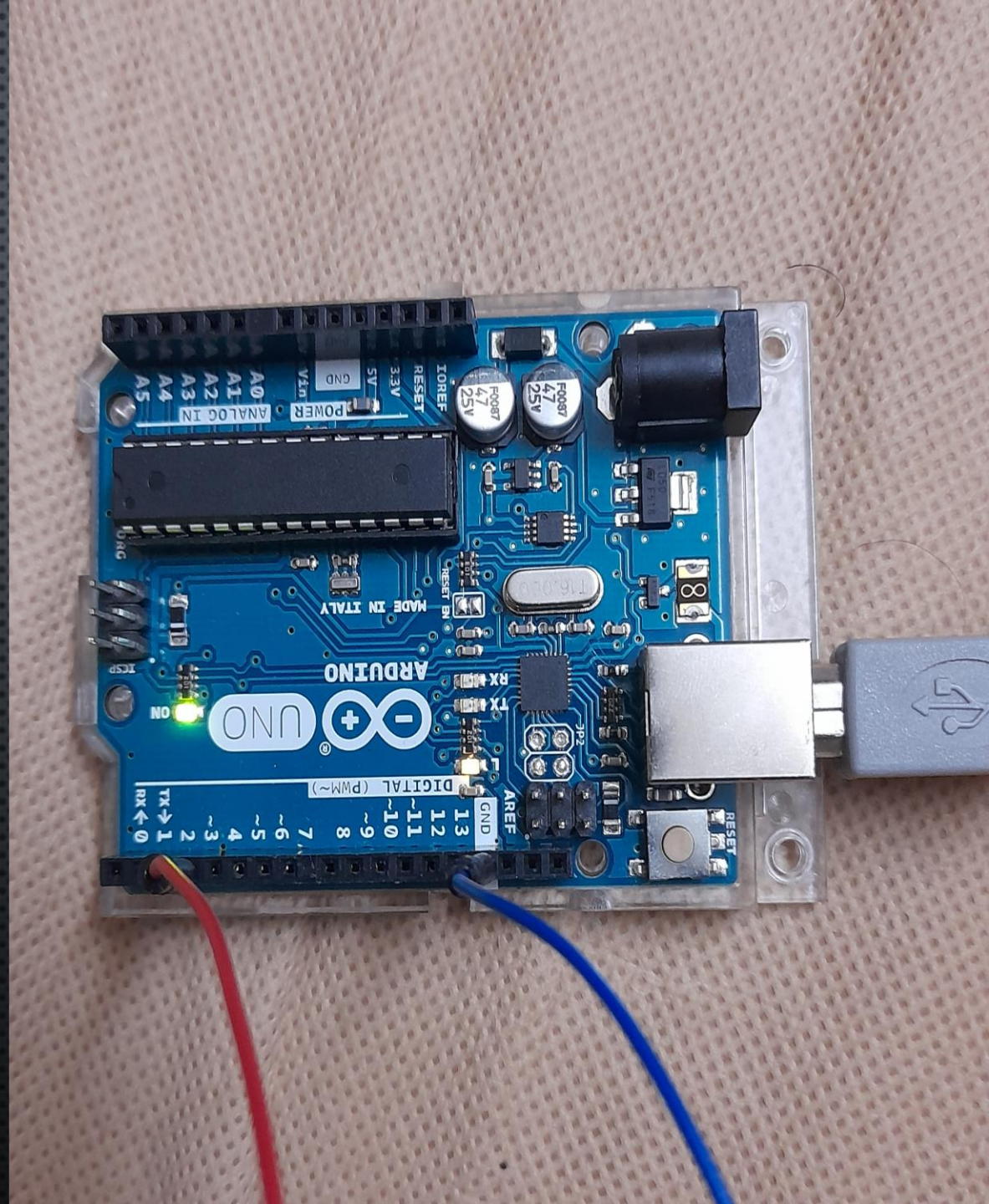
Digital I/O	Math	Bits and Bytes
digitalRead()	abs()	bit()
digitalWrite()	constrain()	bitClear()
pinMode()	map()	bitRead()
	max()	bitSet()
	min()	bitWrite()
	pow()	highByte()
	sq()	lowByte()

EXERCISE BREAKDOWN

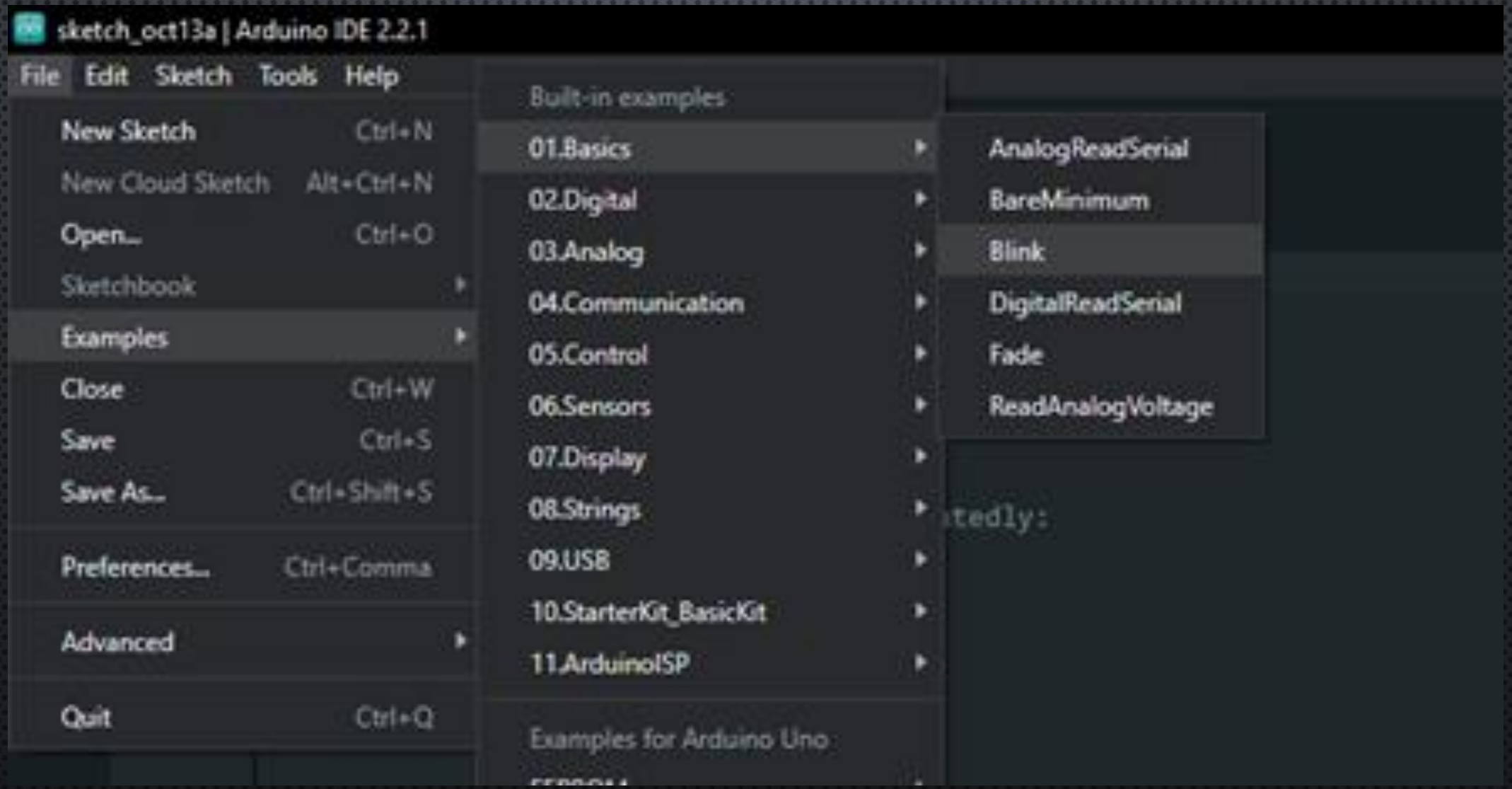
- EXERCISE 1: BLINK LED
- EXERCISE 2: BUTTON INPUTS
- EXERCISE 3: RGB LED
- EXERCISE 4: ACTIVE &
  EXERCISE 4: PASSIVE BUZZERS

EXERCISE 1:
BLINK LED

- This is an intro Exercise to make sure the Arduino is properly connected to your system
- It is also apart of the Built-In Examples given by the Arduino IDE.
- The Expected output of running this code on the Arduino is having the Built-in LED turn on and off every second.

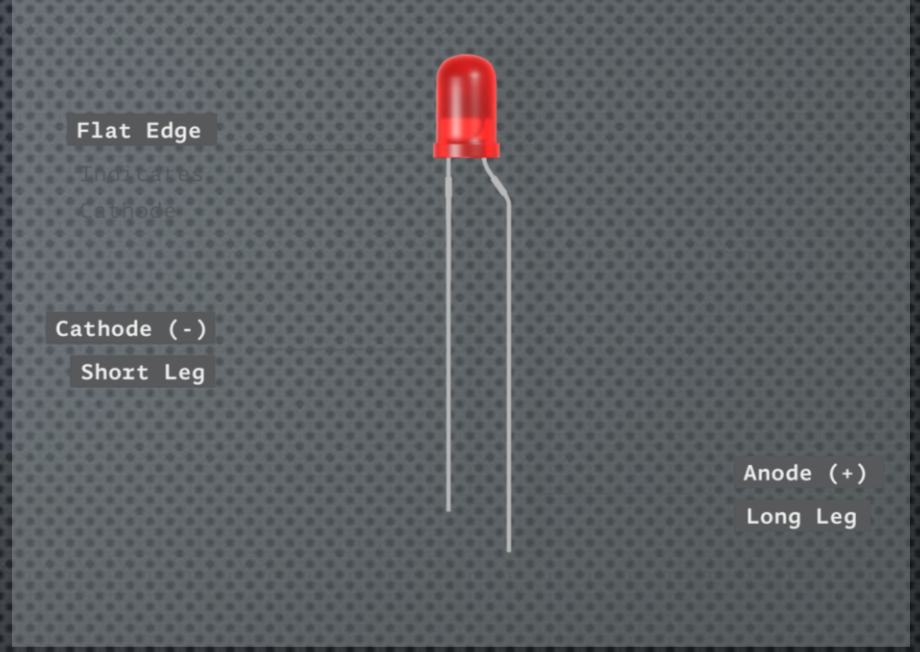


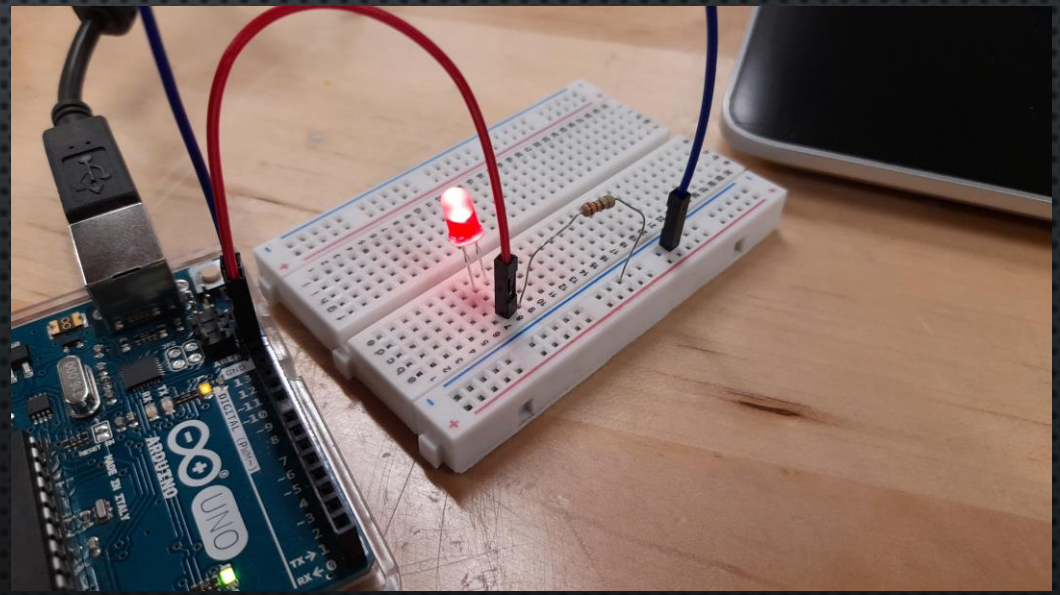
EXERCISE 1: BLINK LED



EXERCISE 1:
BLINK LED

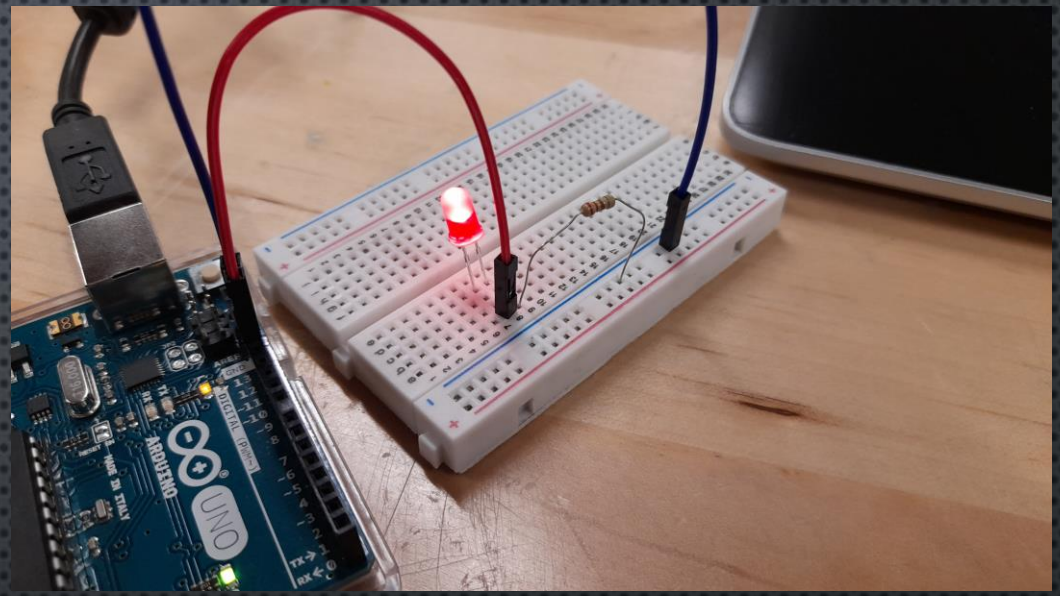
- Now we can connect an LED to pin 13
- Notes & Tips:
  - Setup the circuit before connecting it to power
  - LEDs are polarized and require a resistor to limit the current
  - Make sure to connect to ground to complete the circuit
  - Should expect The LED to turn on and off at the same time as the built-in one

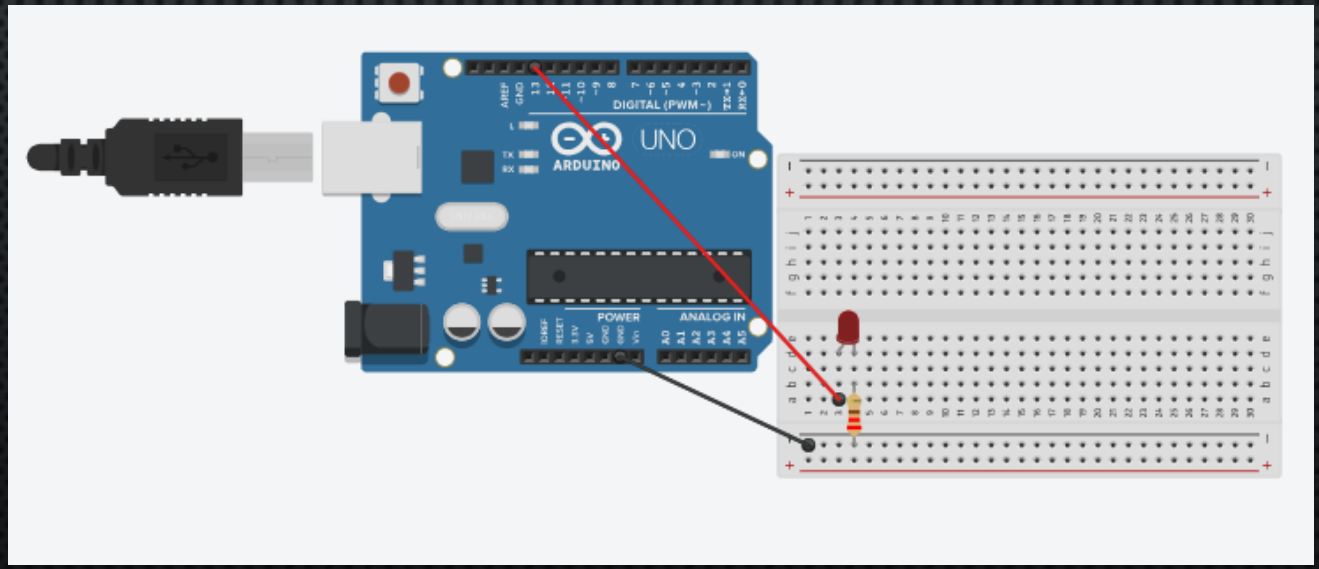




EXERCISE 1:
BLINK LED

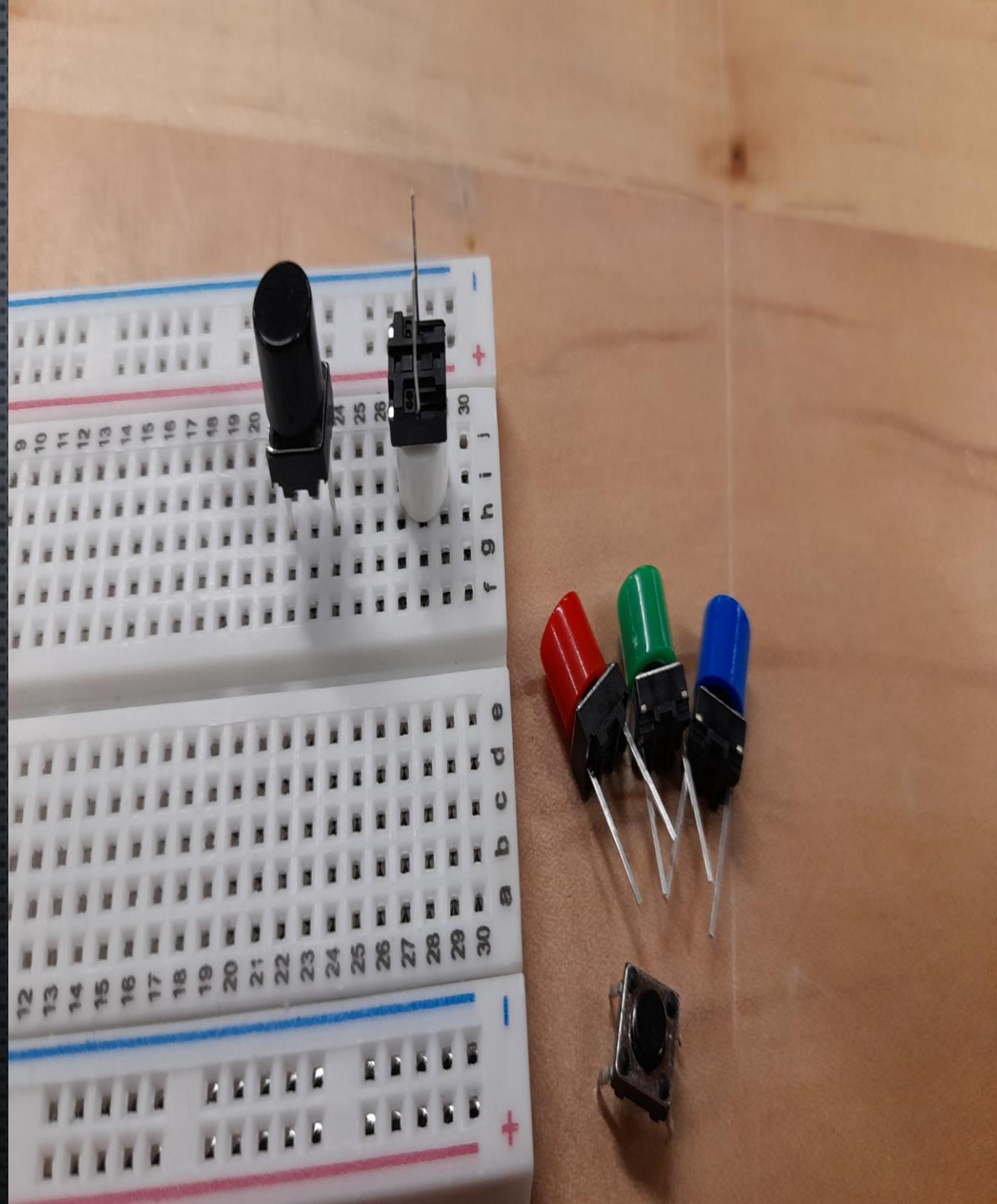
- Circuit Notes:
  - Resistor placement isn't strict
  - Polarization is strict ground cathode (shorter leg)
  - Wiring color doesn't matter

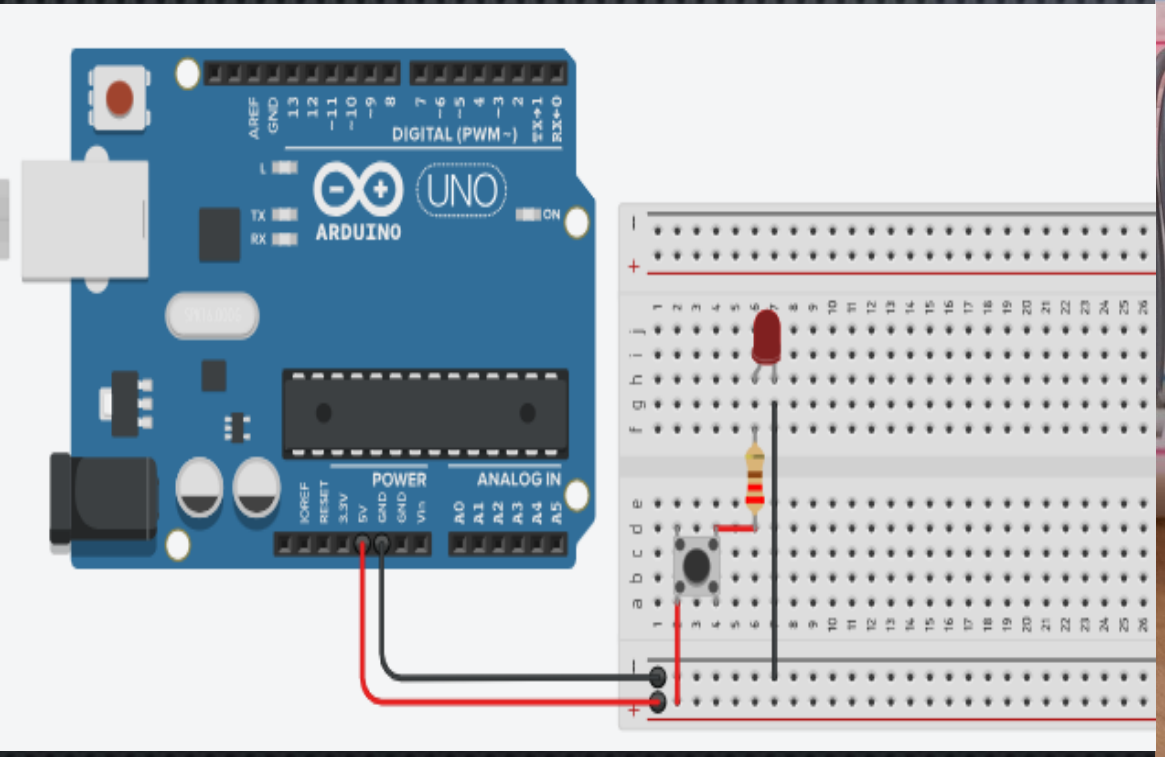


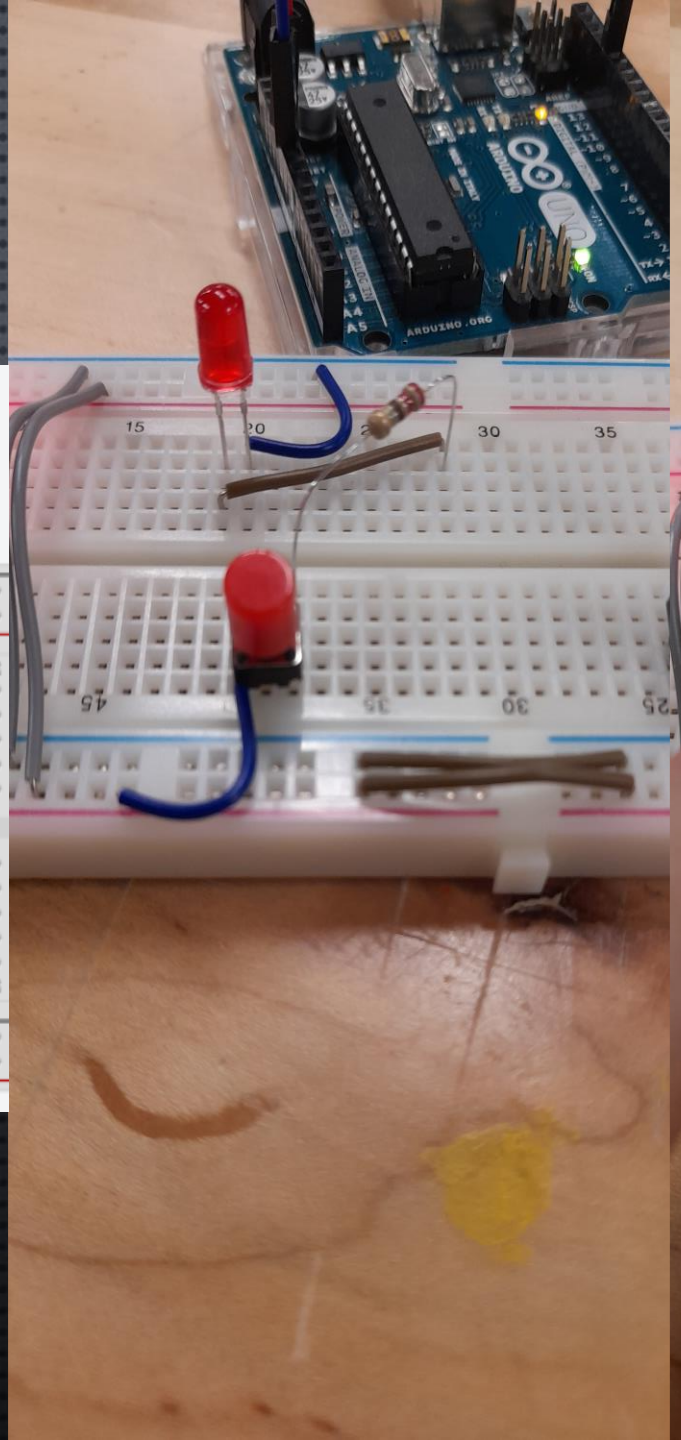


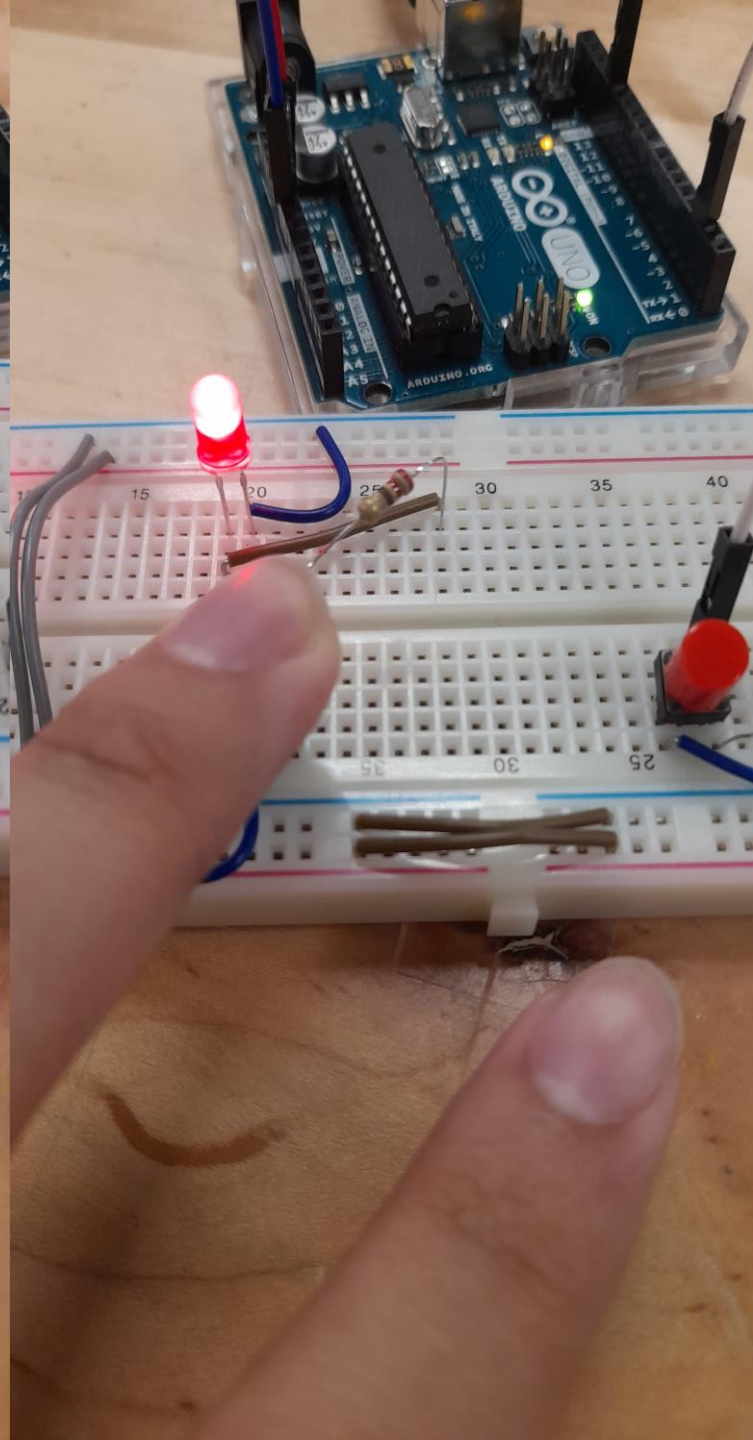
EXERCISE 2: PUSHBUTTONS

- Push buttons are used in 2 main ways:
  - Toggling the current state
  - Changing the current state (counter)
- When using Button with supply It will toggle when you push
  - It acts like a switch to complete the circuit



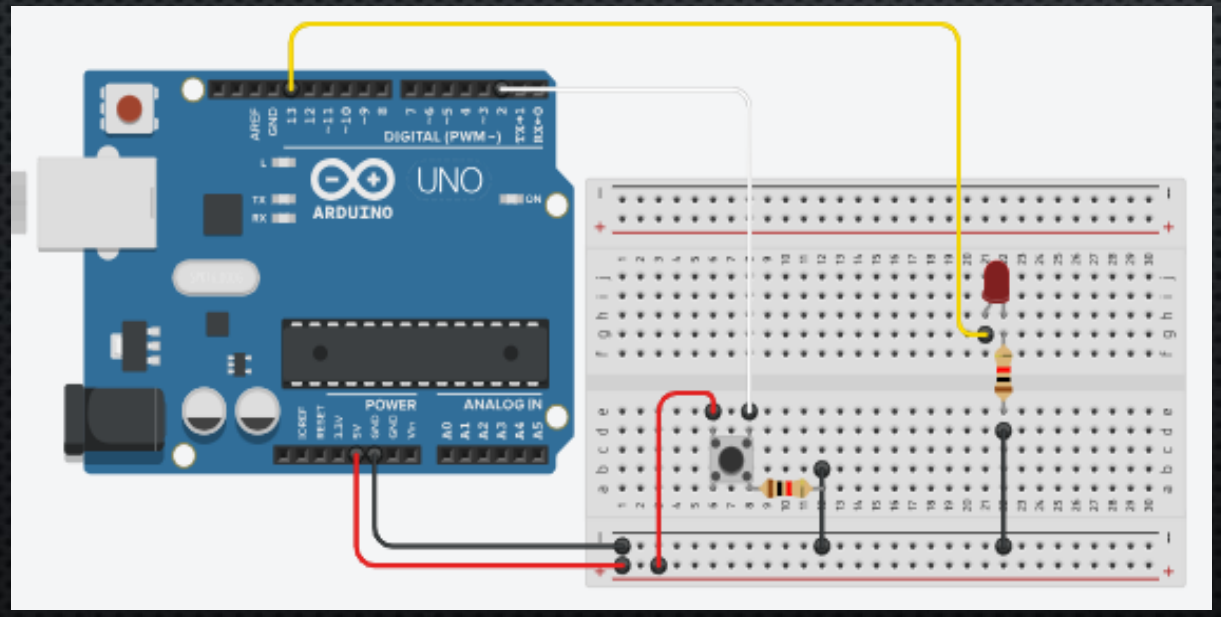

DIGITAL (PWM~)
ARDUINO
UNO
POWER
ANALOG IN

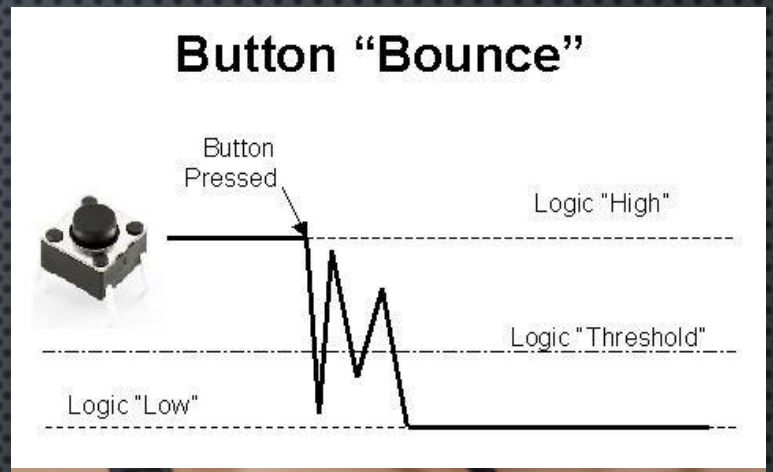


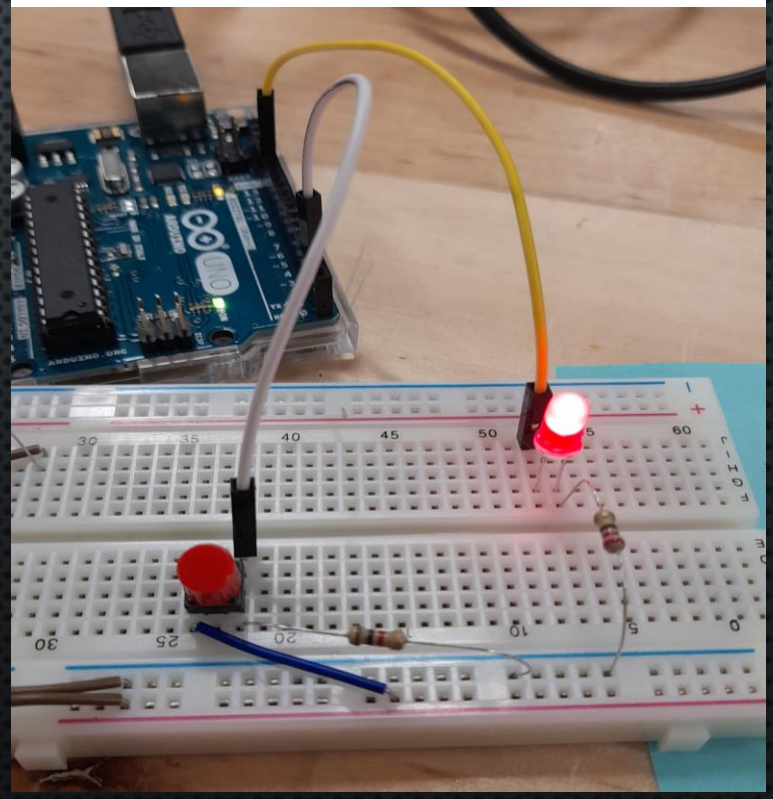


EXERCISE 2: PUSHBUTTONS

- ONE THING YOU MAY HAVE NOTICED IS THE TOGGLE BEING INCONSISTENT
- THE BUTTON IS BOUNCING ITS STATES USUALLY AT THE BEGINNING
- THIS CAN BE SOLVED BY ADDING A DELAY AND TAKING IN THE INPUT AFTER IT HAS A MORE CONSISTENT INPUT FROM YOU

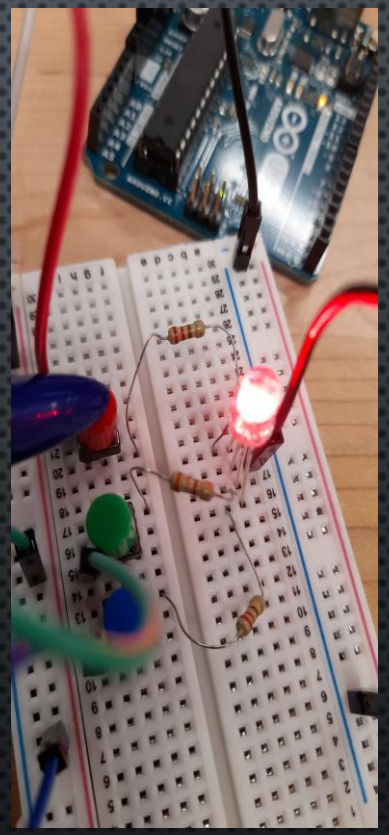


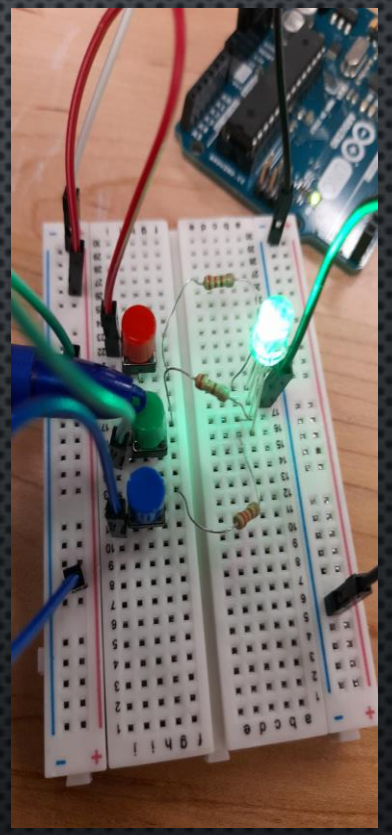


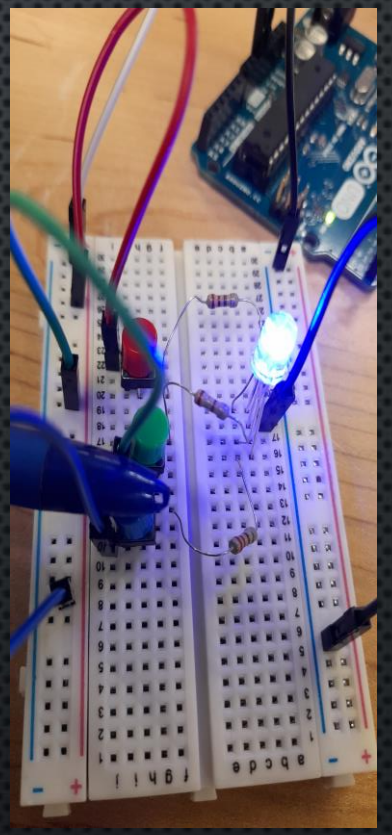


EXERCISE 3:
RGB LED

- An RGB LED can emit the colors Red, Green, & Blue
  - Can also emit all or some of the colors at once
- RGB LED Pins are Red, **GROUND**, Green, & Blue

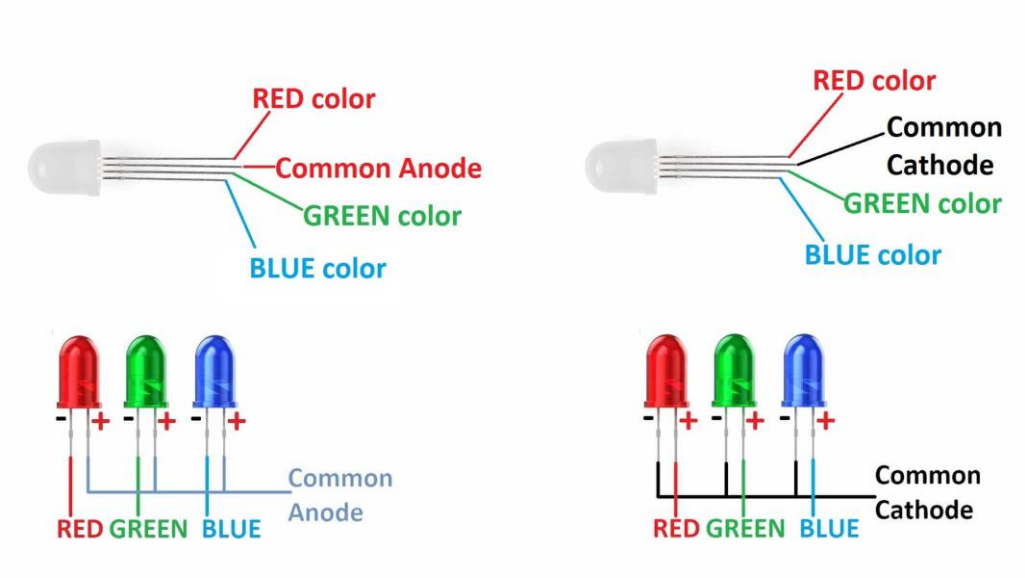


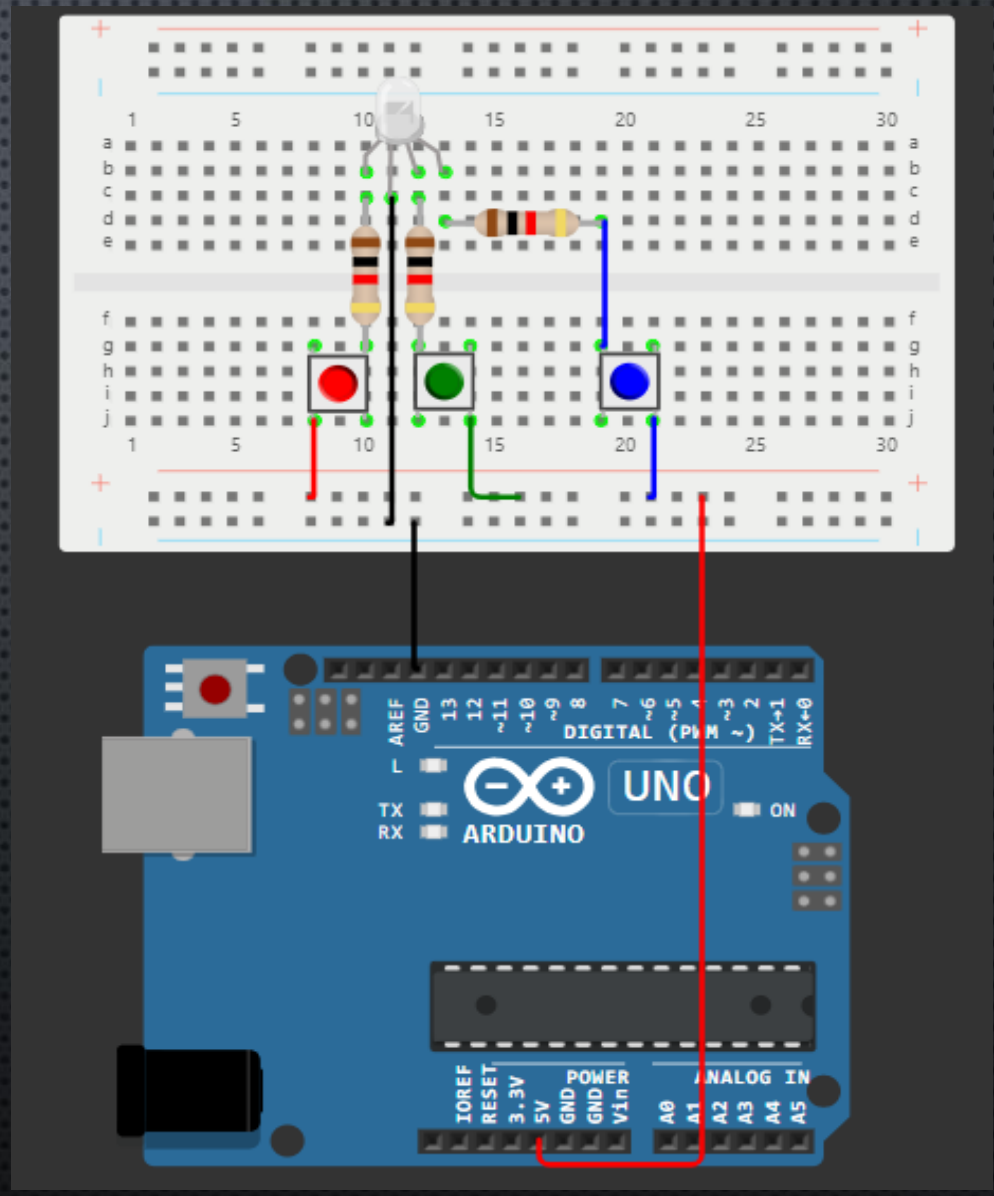


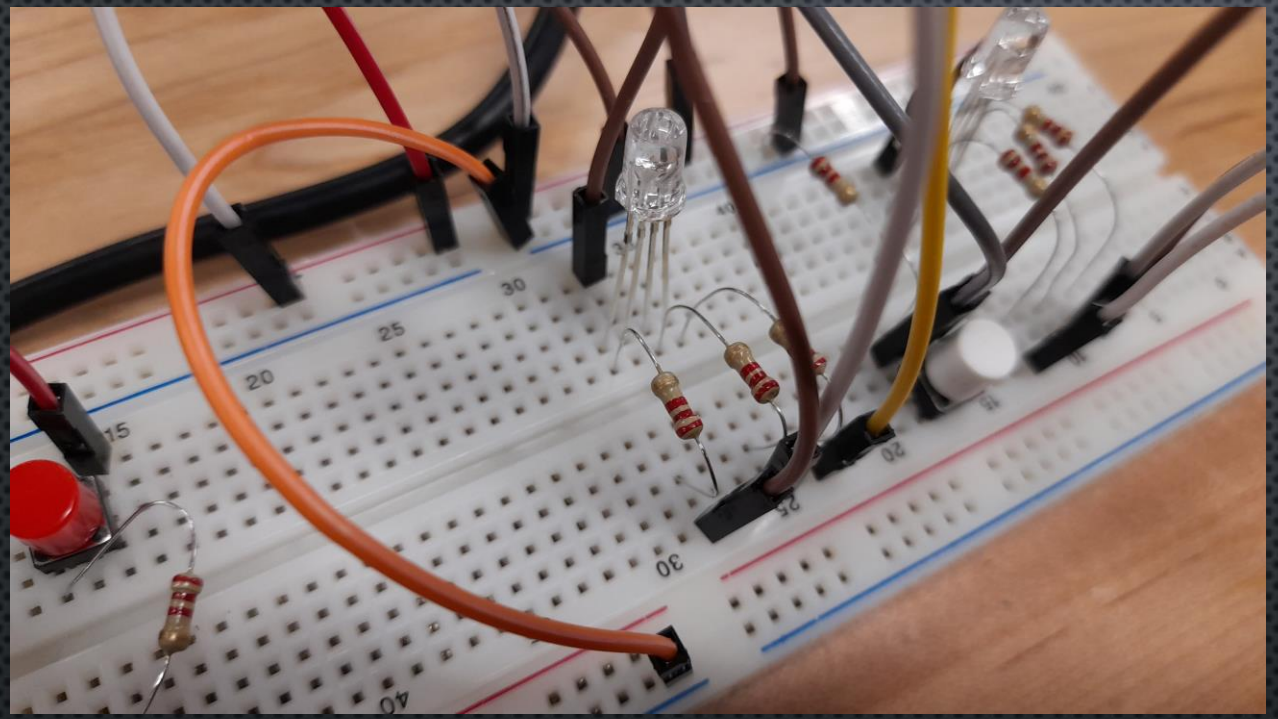


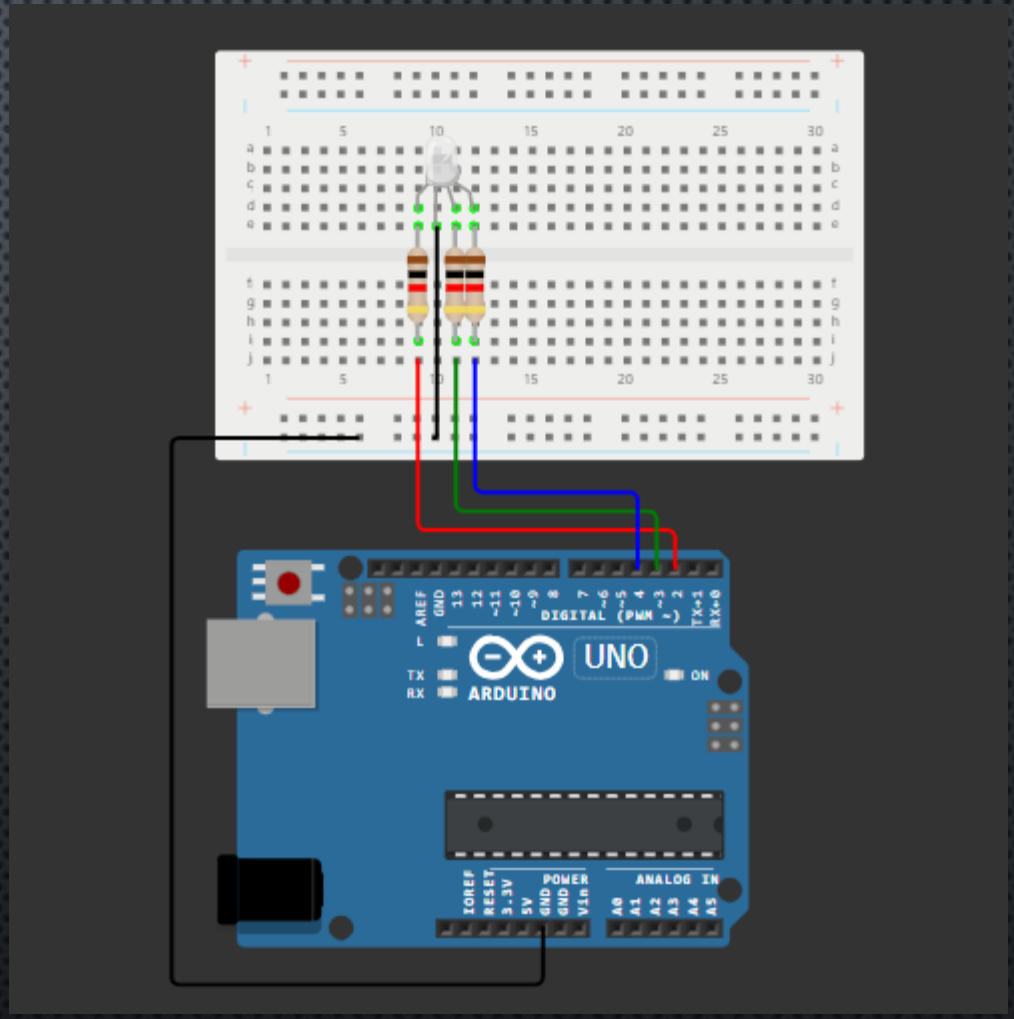
EXERCISE 3:
RGB LED

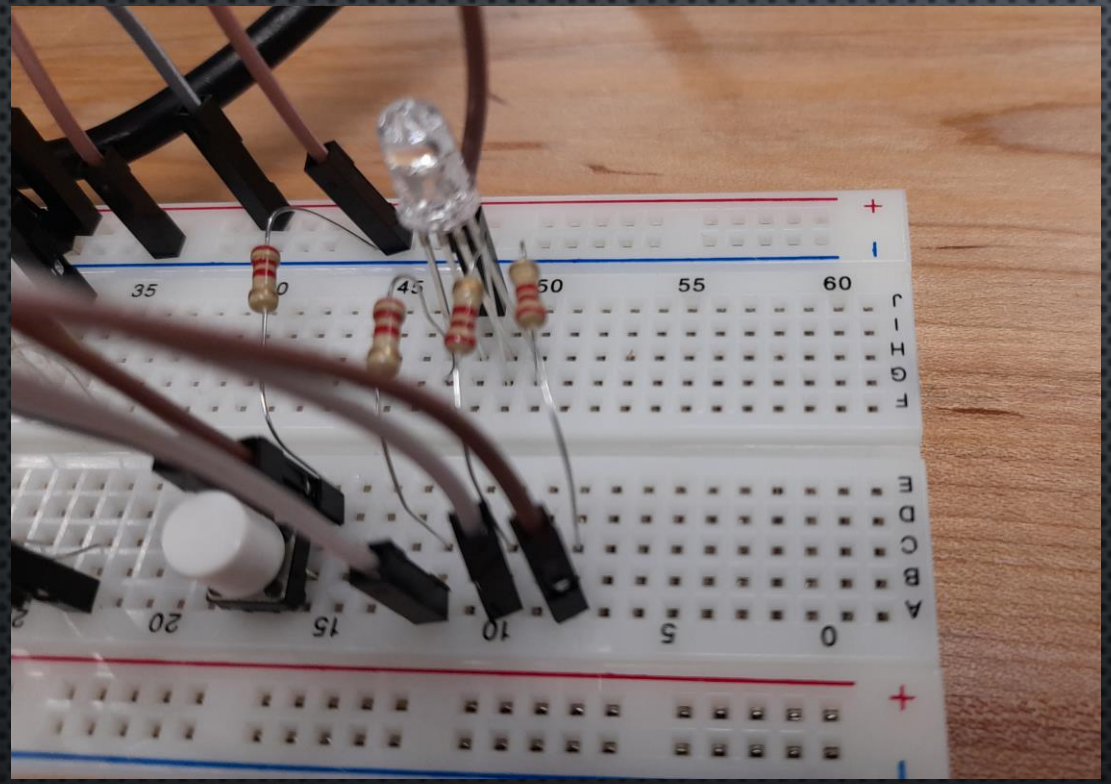
- RGB LED PINS ARE RED, **GROUND**, GREEN, & BLUE
- THEY SHARE A COMMON GROUND OR COMMON CATHODE LEG

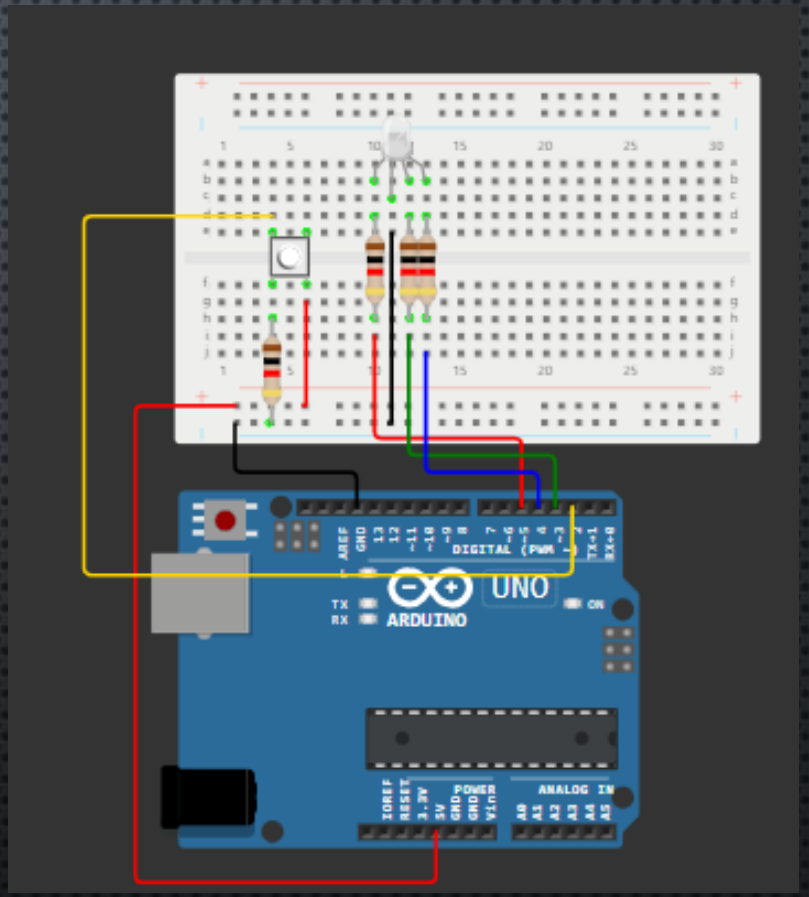






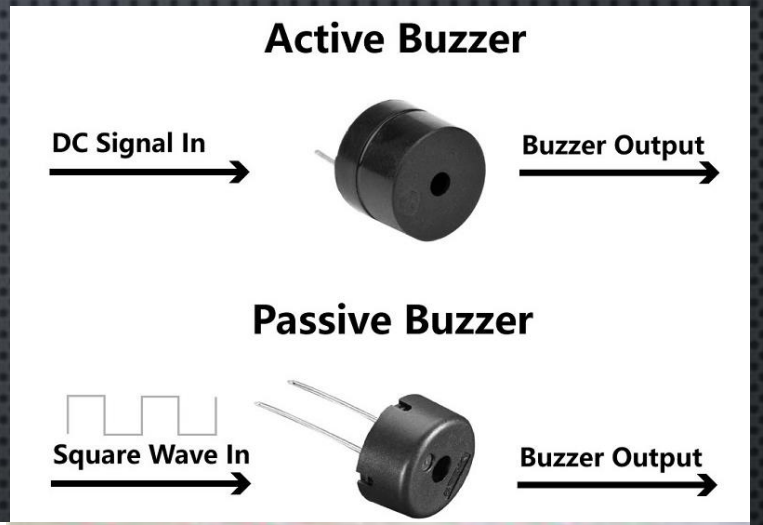

UNO
ARDUINO
DIGITAL (PWM ~)
POWER
ANALOG IN

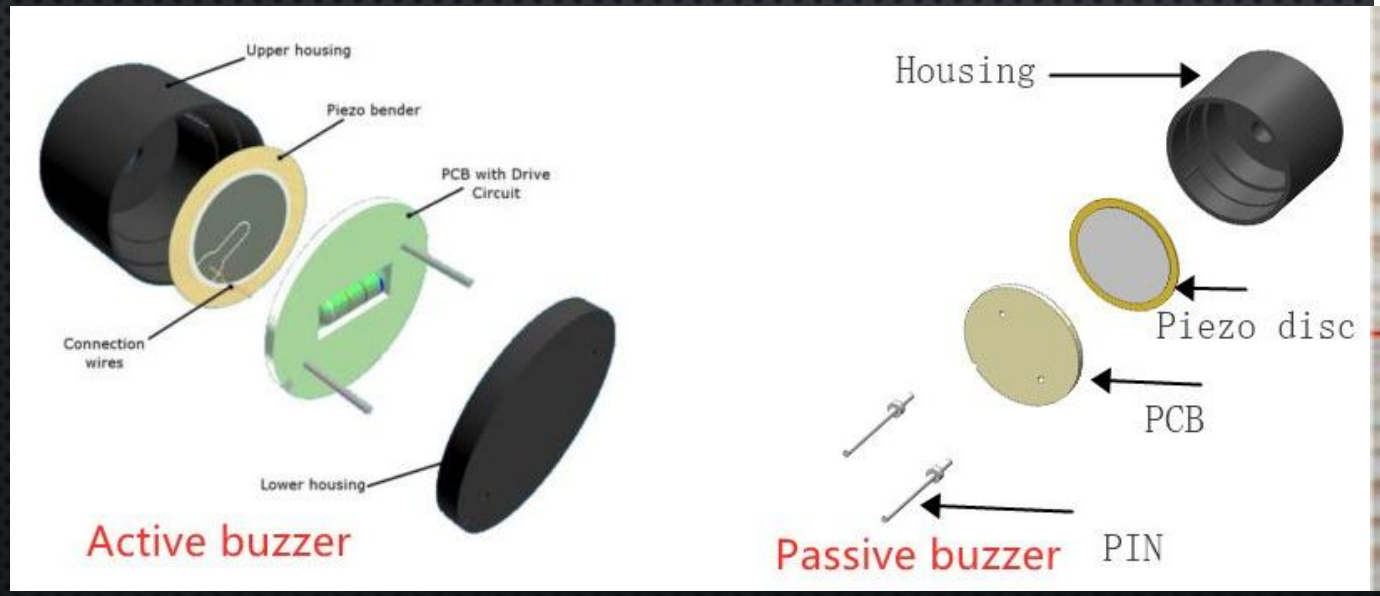



UNO
ARDUINO
TX
RX
ON
DIGITAL (PWM~)
POWER
ANALOG IN

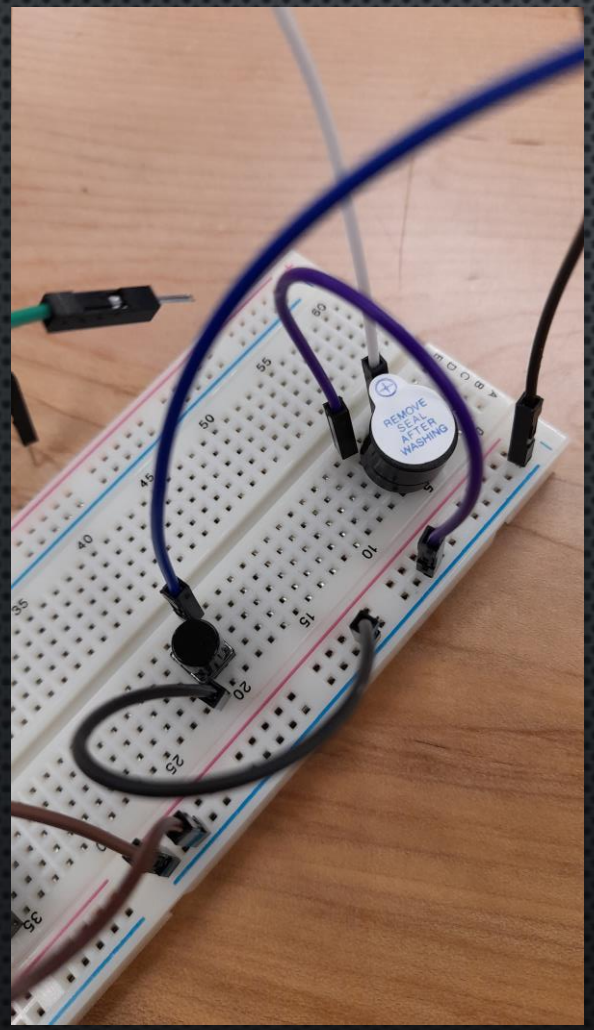
EXERCISE 4: ACTIVE & PASSIVE BUZZERS

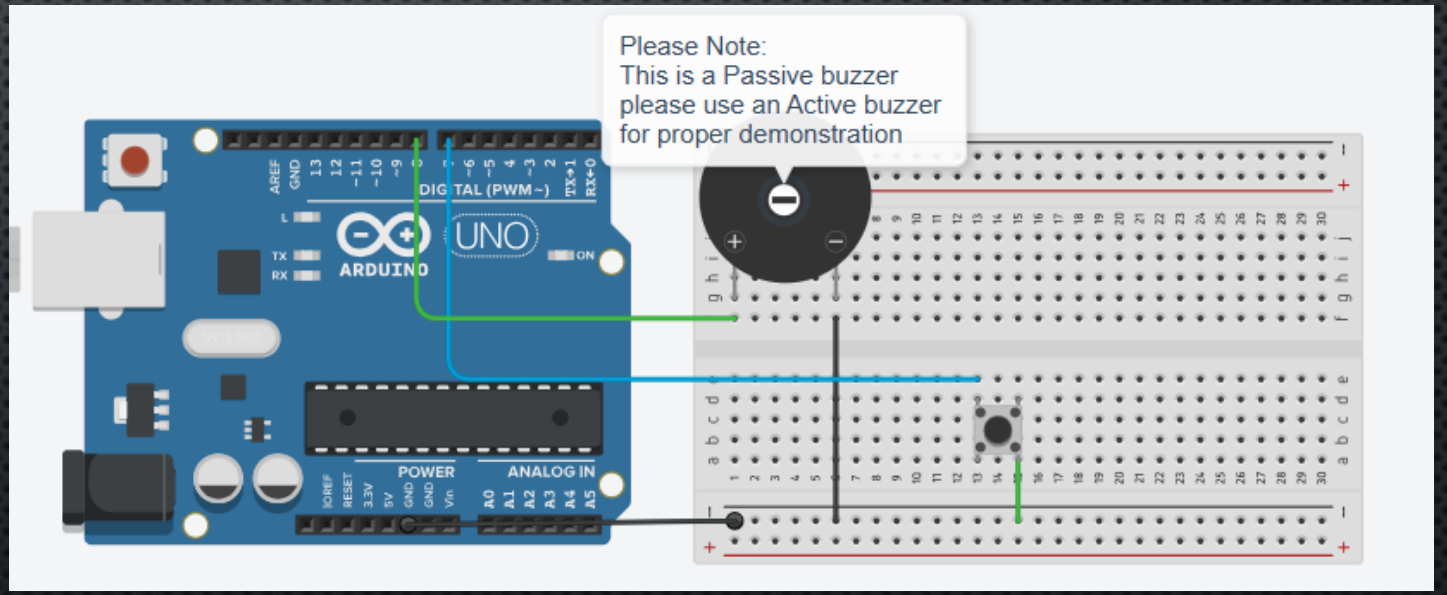
- ACTIVE BUZZERS PRODUCE SOUND AT A PRE-SET FREQUENCY BY SIMPLY PROVIDING POWER TO THE BUZZER
- PASSIVE BUZZERS CAN CHANGE THEIR FREQUENCY BASED ON THE INPUT SIGNAL



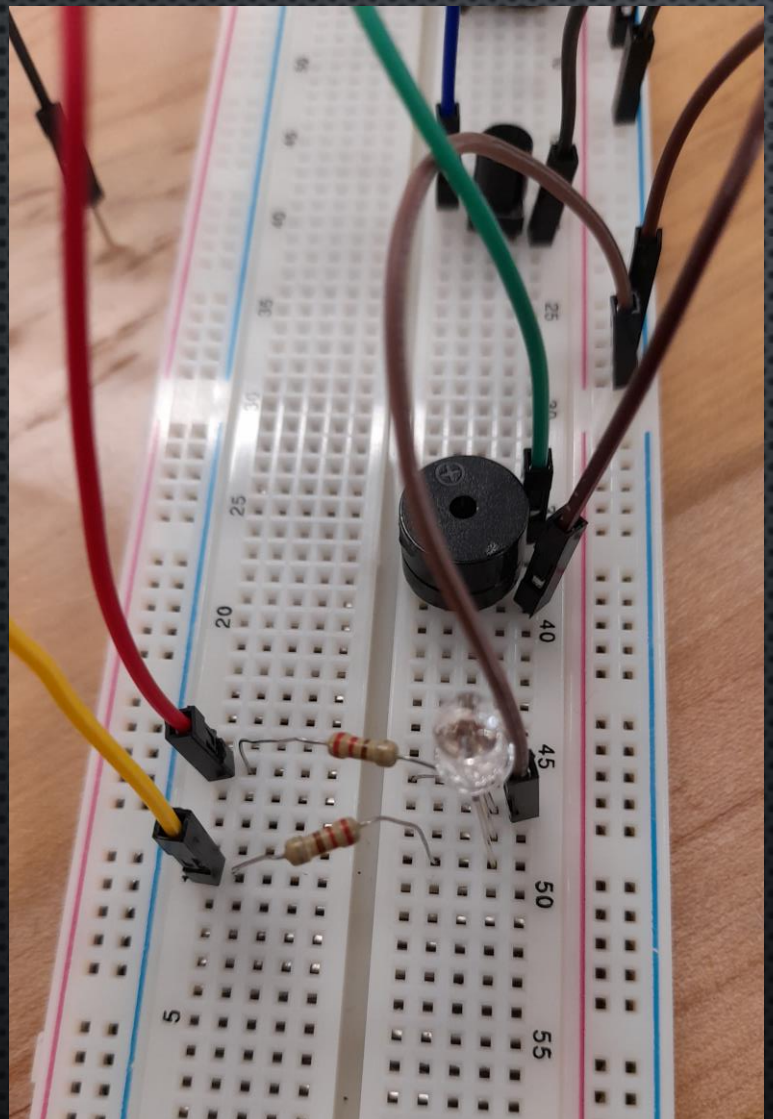


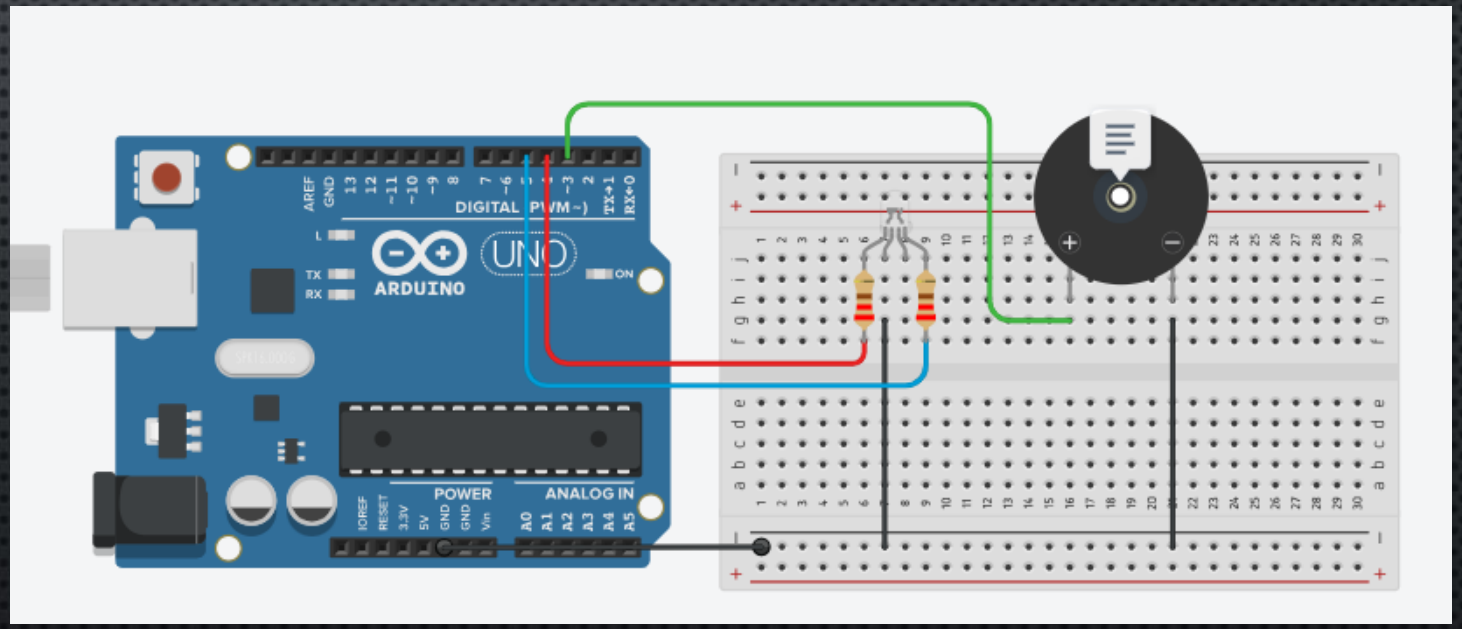
EXERCISE 4: ACTIVE BUZZERS





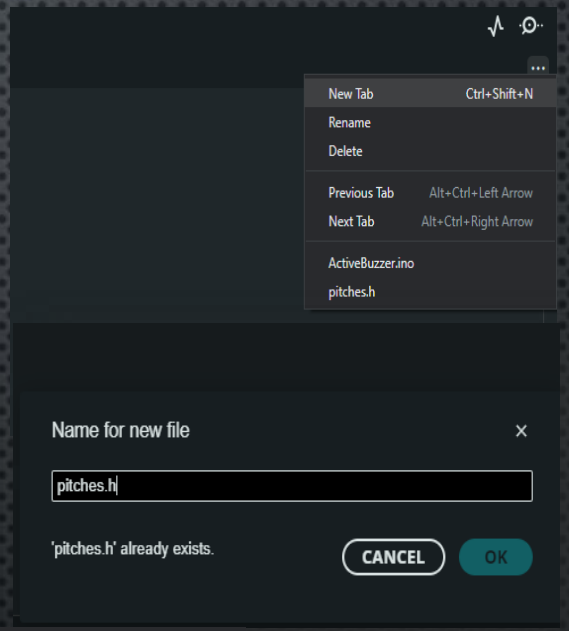
EXERCISE 4: ACTIVE BUZZERS

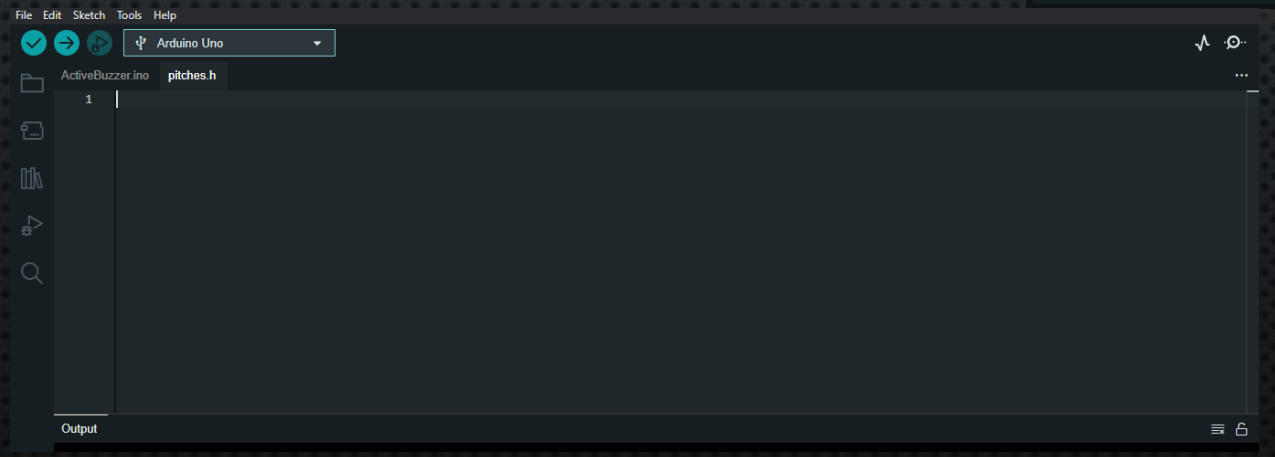




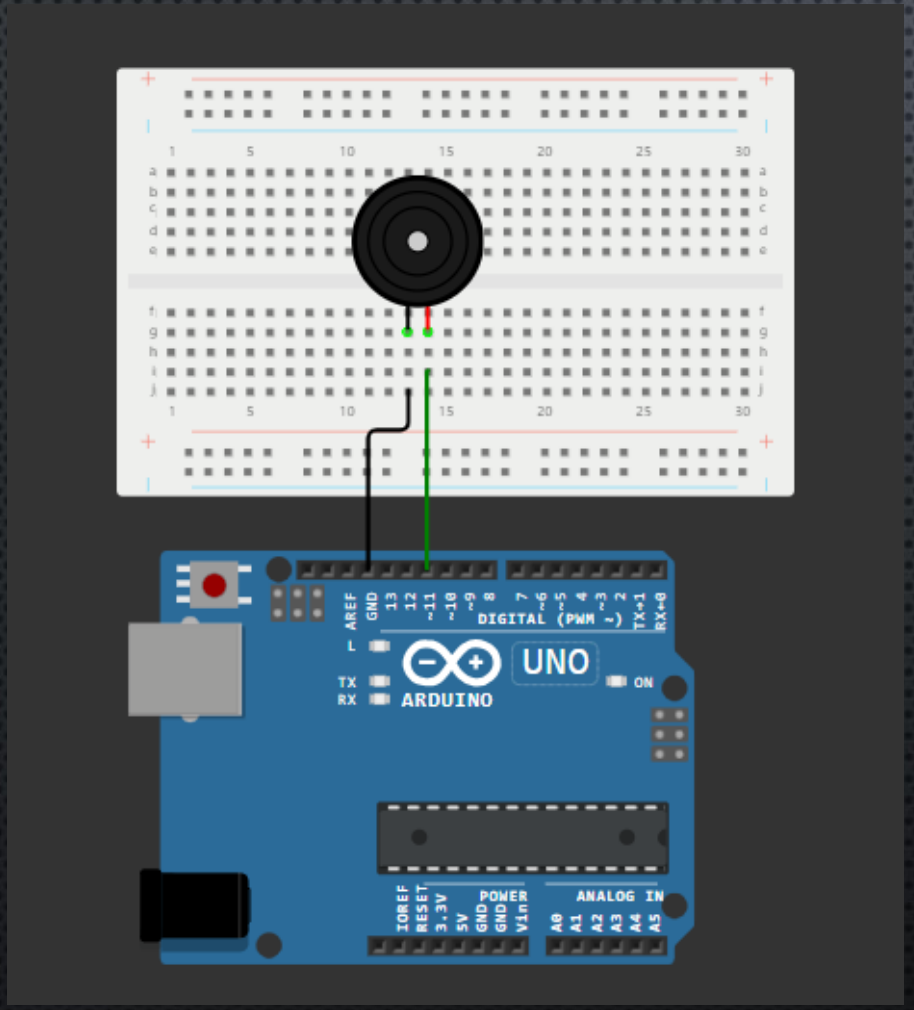
HOW TO MAKE NEW FILES

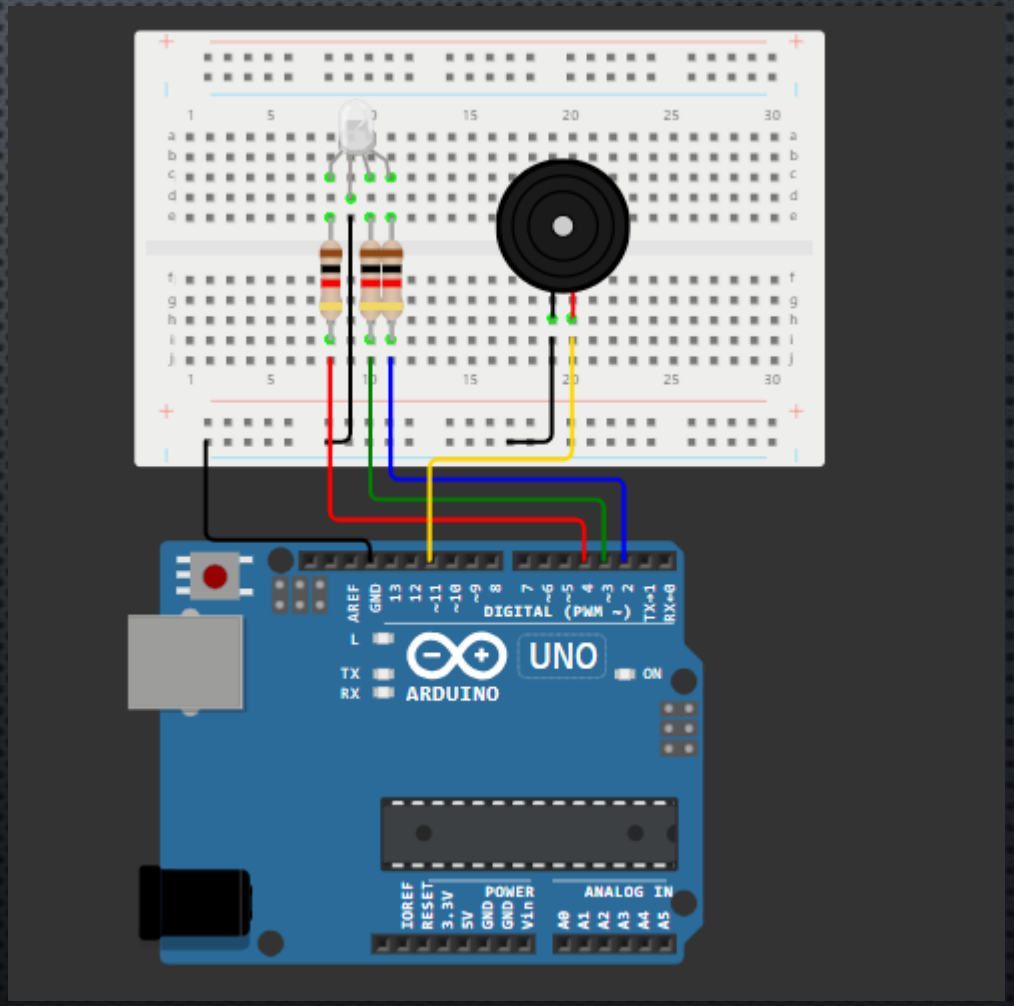
- GO TO THE 3 DOTS ON THE RIGHT
- CLICK NEW TAB
- ENTER FILE NAME AND TYPE OF FILE (.H)
  - NOTE: IF YOU DON'T DO THE FILE TYPE IT DEFAULTS TO A .INO FILE
- THE FILE SHOULD NOW BE IN THE SAME PROJECT FOLDER



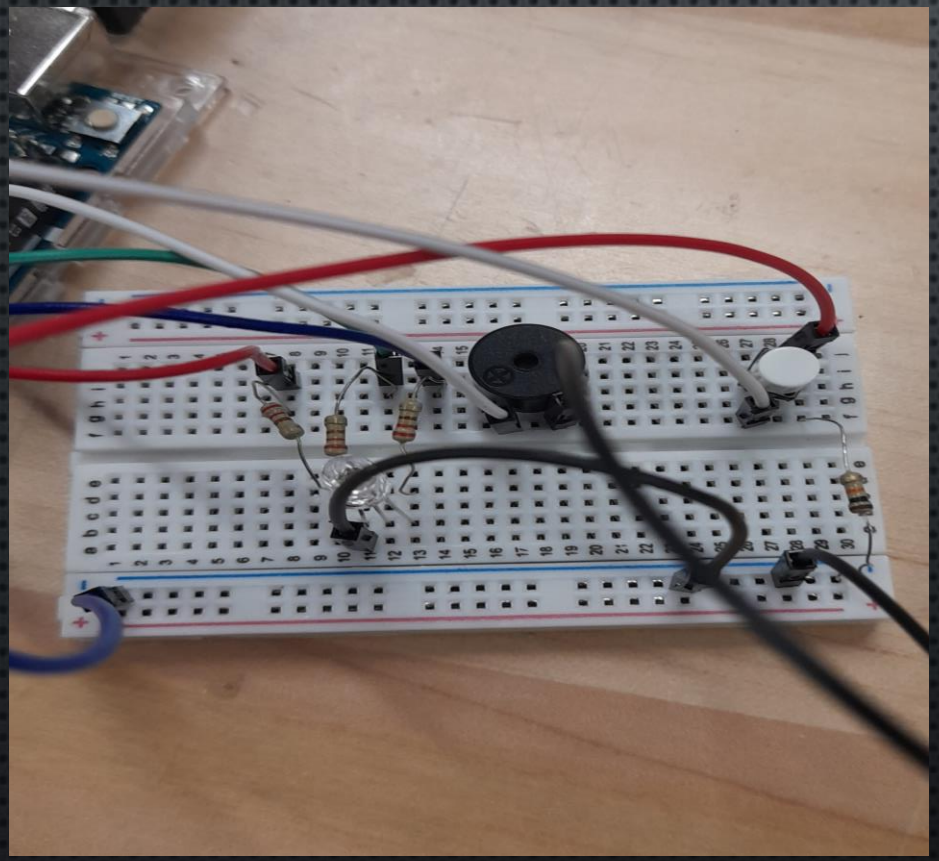


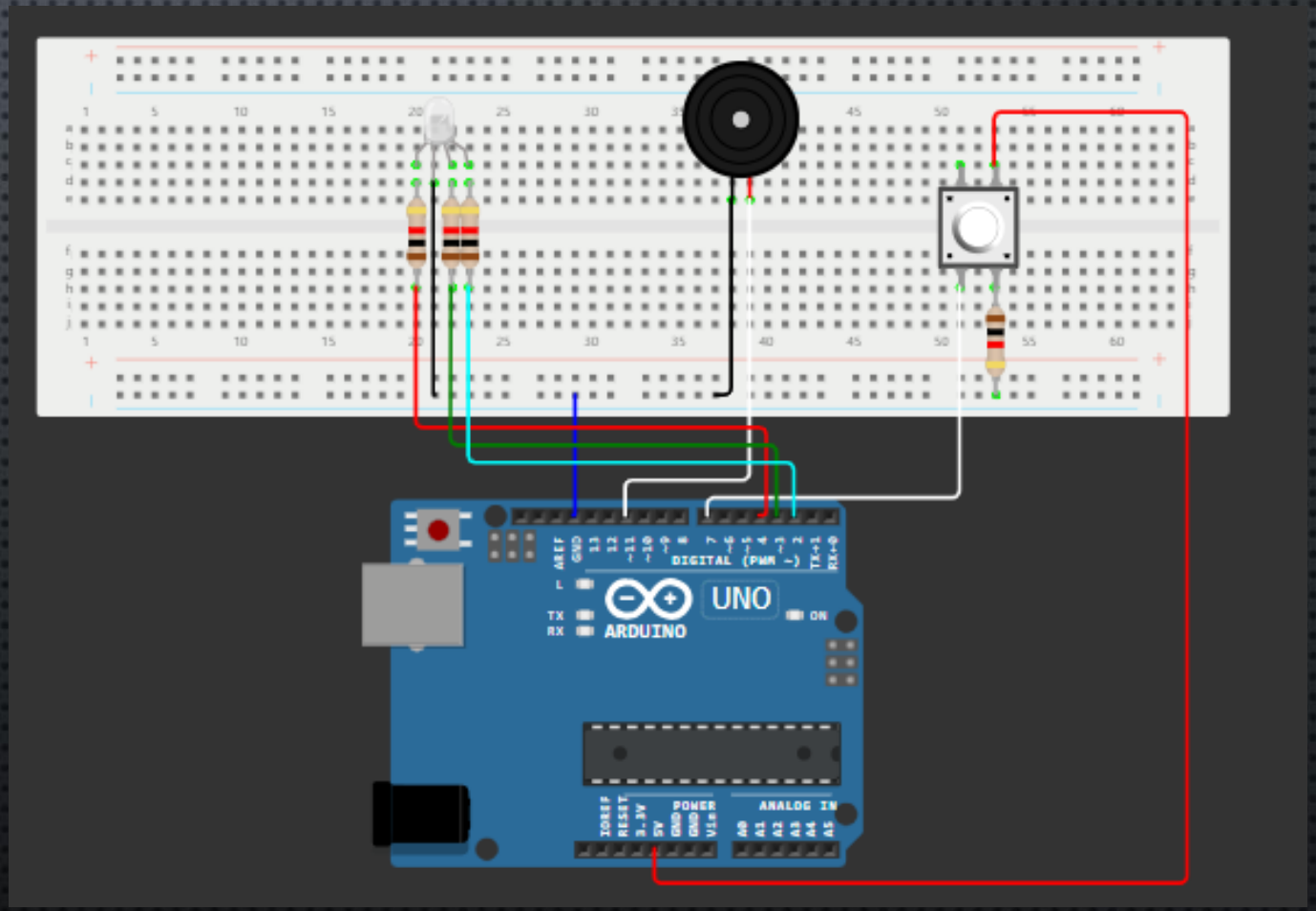
EXERCISE 4: PASSIVE BUZZERS





EXERCISE 4: PASSIVE BUZZERS





Arduino Next Steps

You should try to look at the other built-in examples from the IDE

Some of them have very good explanations on the Arduino website as well

- https://docs.arduino.cc/built-in-examples/

Check out the Arduino Language:

- https://www.arduino.cc/reference/en/

Check out some Beginner Projects on YouTube

Look at the C/C++ Language:

- https://www.w3schools.com/c/
  - I would Recommend this first then moving on to C++
- https://www.w3schools.com/cpp/

QUESTIONS?


THANK
YOU
MKRSPACE@MCMASTER.CA