NYU TANDON SCHOOL OF ENGINEERING

Promoting robotic design and entrepreneurship experiences among students and teachers

Lesson 5: Introduction to Arduino

Innovative Technology Experiences for Students and Teachers (ITEST), Professional Development Program, July 2017 - 19
Mechatronics, Controls, and Robotics Laboratory, Department of Mechanical and Aerospace Engineering, NYU Tandon School of Engineering

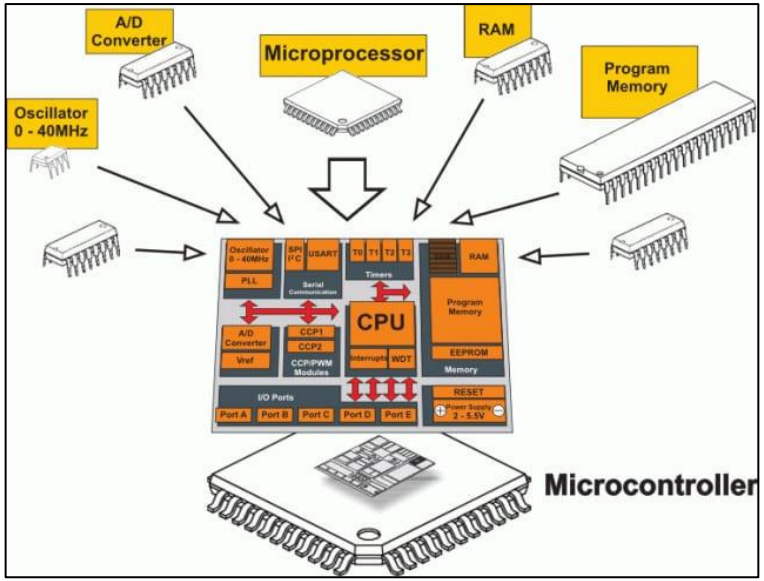
CONTENTS

- Microcontrollers and microprocessors
- Introduction to Arduino
- Types of Arduino boards
- Programming basics: Structure of a code

- **TASK/ACTIVITY**:
  - Blink on-board LED
  - Blink LED(s)
  - Change brightness of LEDs

WHAT IS A MICROCONTROLLER?

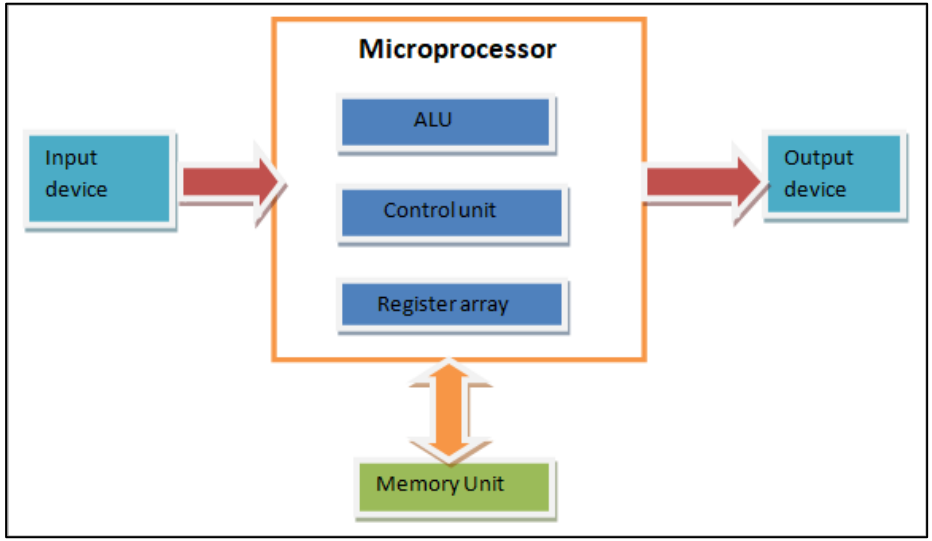
- A **compact integrated circuit** on a **single chip** containing a **processor**, **memory**, and **input/output** as its main components
- Typically, it is "**embedded**" inside a device that is controlL E DS
- A microcontroller is often small and of low cost



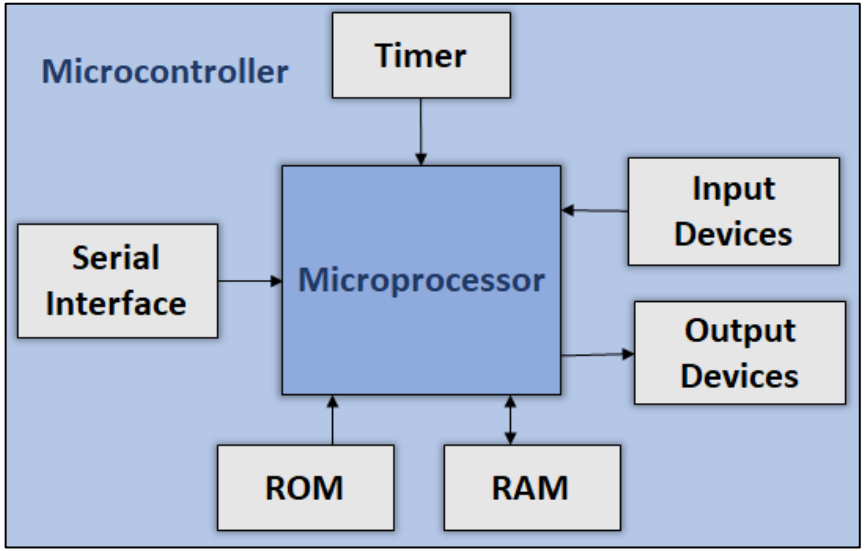
Source

MICROPROCESSOR VS MICROCONTROLLER

- **The key difference: Microprocessor** consists of only a **Central Processing Unit**, whereas the **microcontroller** contains a **CPU, memory, I/O** all integrated into one chip



Source



Source

MORE ABOUT THE DEVICES

What is the difference between a **computer**, a **microprocessor** and a **microcontroller**?

- **Microprocessor** is a full computation engine fabricated on a single chip; It acts as the central processing unit **(CPU) of a microcomputer**

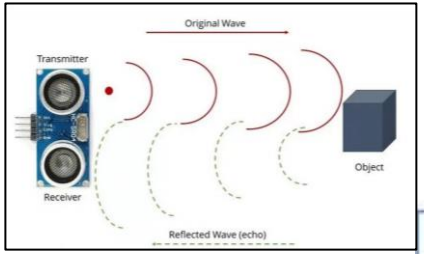
- A **Computer** is a **microprocessor** packaged on a single circuit board with **many interfaces** and **memory chips**; General purpose computers, i.e., PCs, are designed explicitly to **interface with humans**

- **Microcontrollers** are designed to **interface, interact, and communicate with**

➢ Electrical/electronic devices
➢ Sensor/actuators
➢ High-tech gadgets, etc.

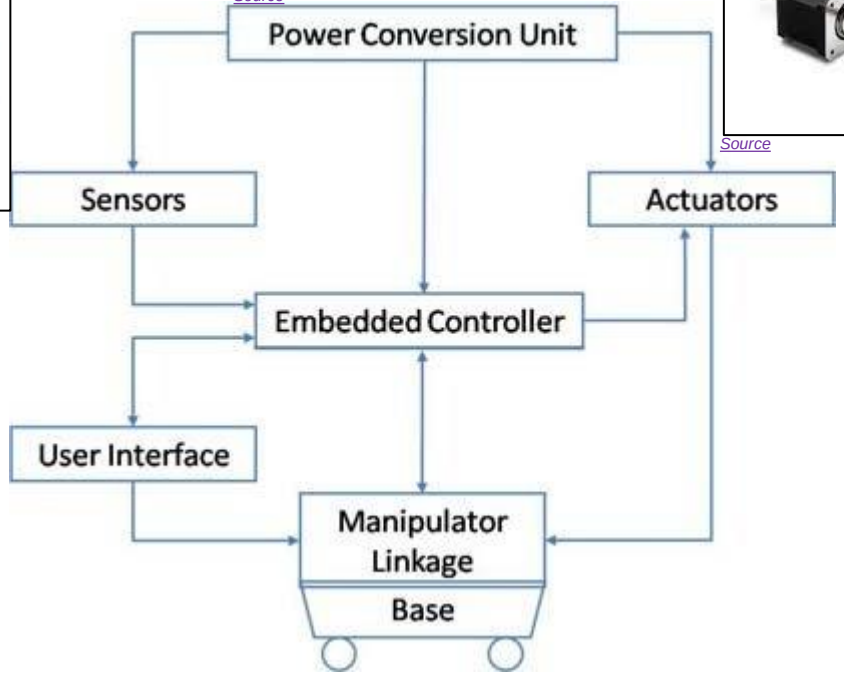
COMPONENTS OF A ROBOT

- As discussed earlier:

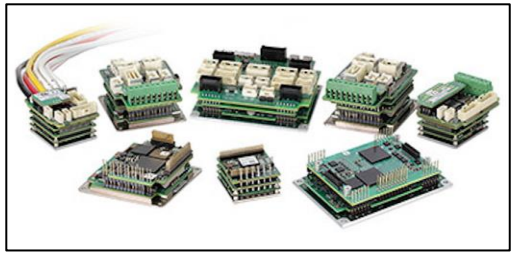


Source

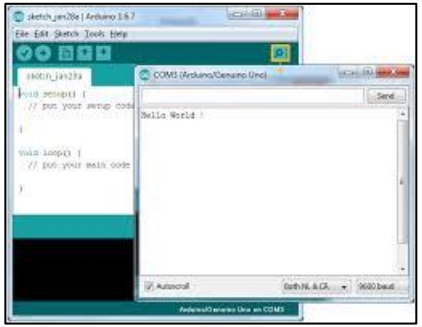
Source



Source



Source

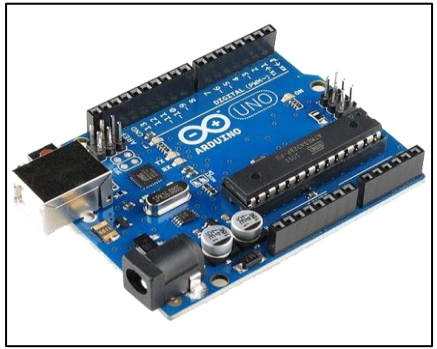


Source

Source

WHAT IS ARDUINO???

- Arduino is an **open-source** electronic platform for easy use of hardware and software
- It can **sense the environment** by receiving **input** from variety of sensors and **make decision** and then give the **output**
- It can be used to **develop stand-alone interactive objects** or can be connected to software on your computer

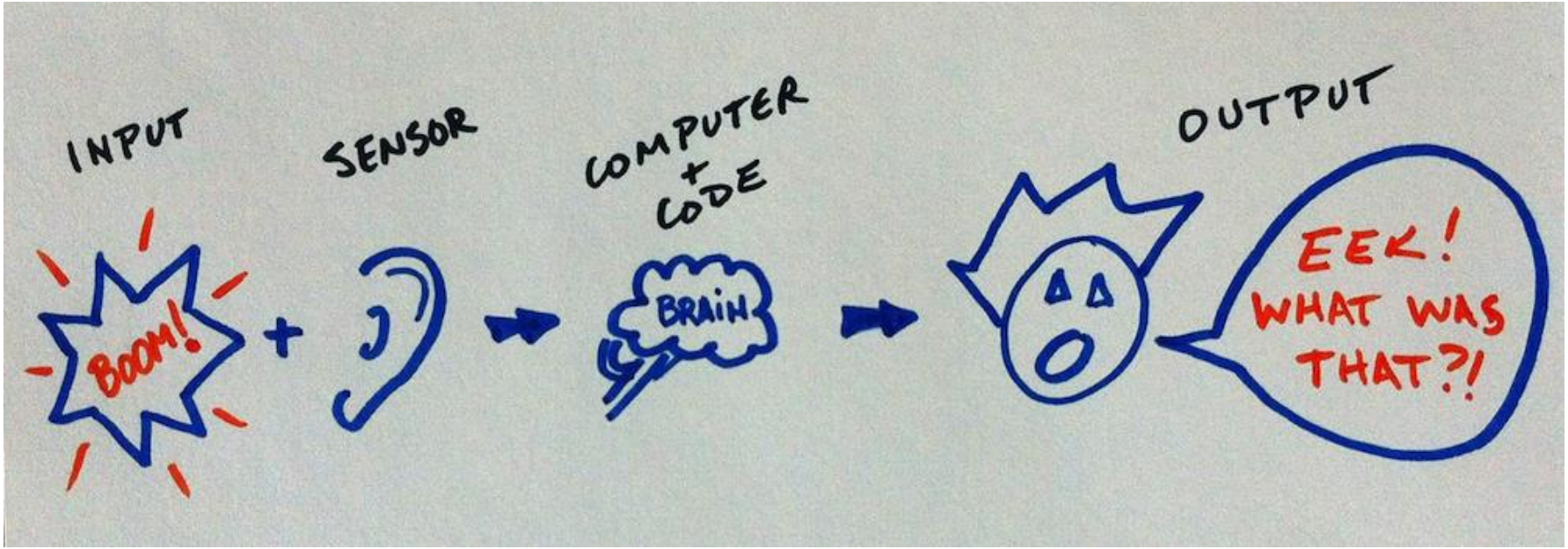


Source

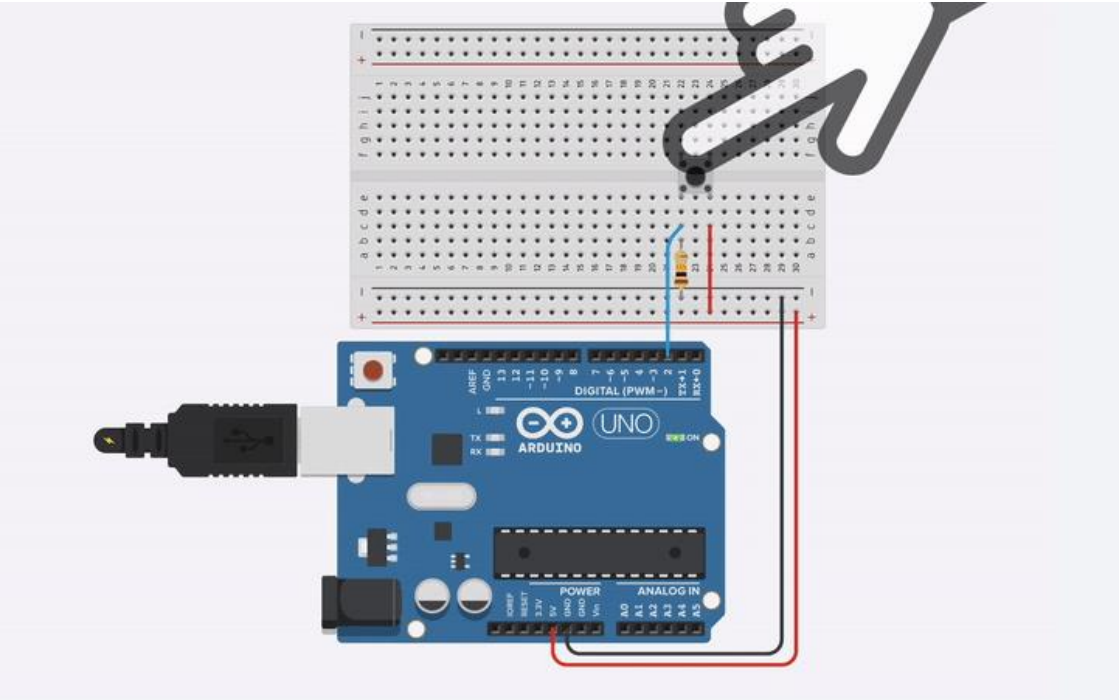
Arduino

WHAT DOES IT DO???

- **Working:** The microcontroller (**computer**) is programmed (**code**) to receive information (**input**) from the **sensors** and the **output** is given through **computer (IDE)** or other **peripherals**



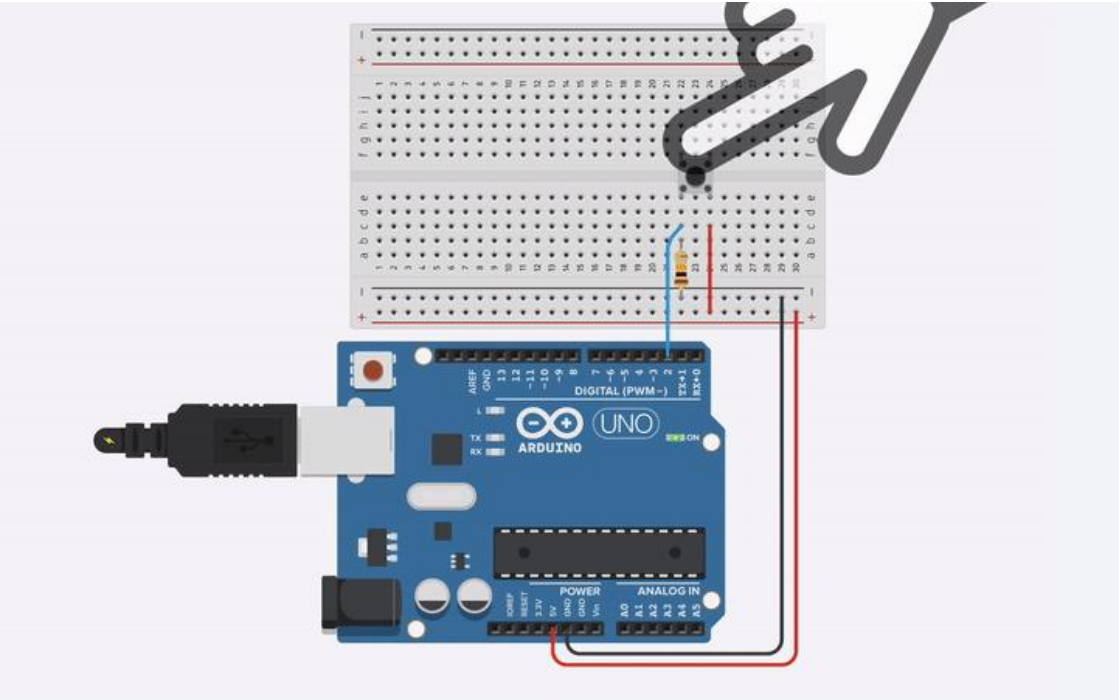
ACTIVITY 1



Source

- What is the circuit doing?
- What elements do you see on the circuit?
- Is there any input?
- If yes, what is the input?
- Is there any output?
- If yes, what is the output?

ACTIVITY 1 - SOLUTION



Source

- What is the circuit doing?
  Light up the on-board LED on button press

- What elements do you see on the circuit?
  Button, resistor, wires

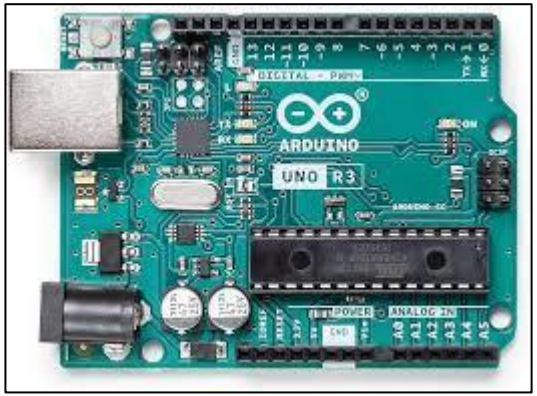
- Is there any input? **Yes**
- If yes, what is the input?
  Button

- Is there any output? **Yes**
- If yes, what is the output? **LED**

ARDUINO

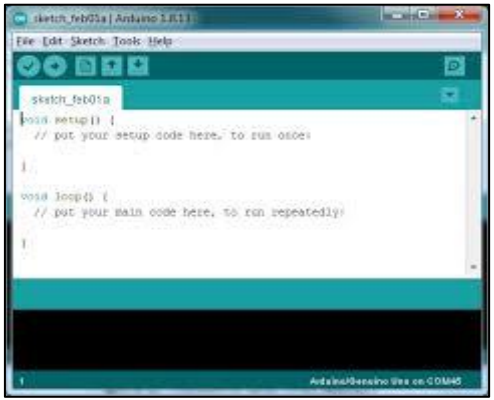
- The word Arduino can mean 3 things:

Hardware



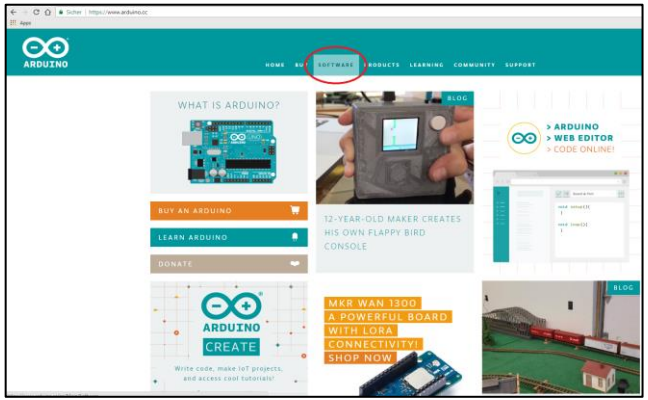
Source

An interface



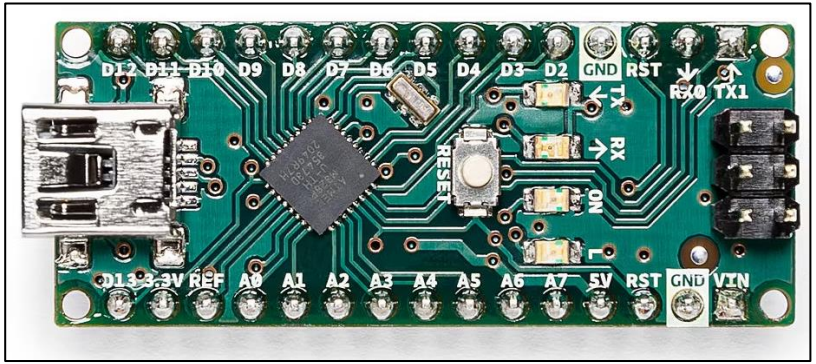
Source

A community

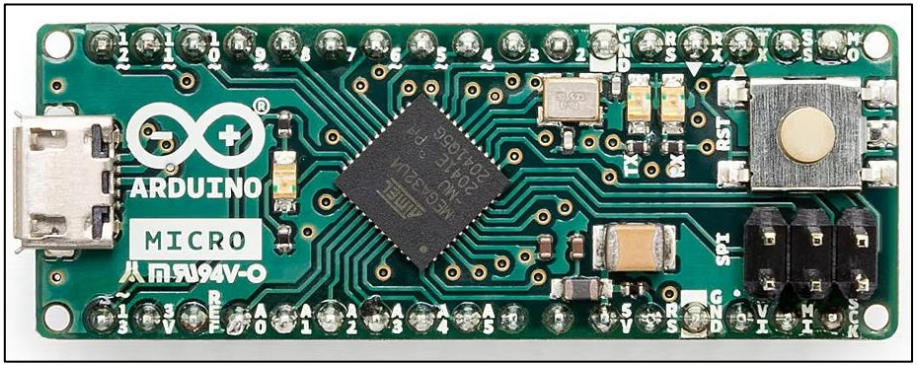


Source

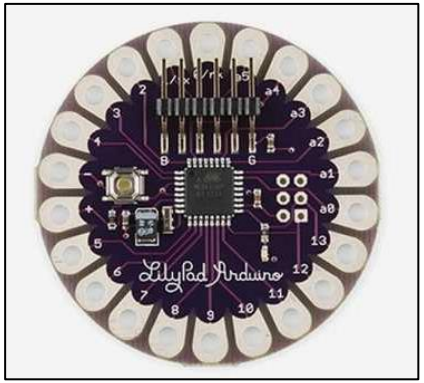
DIFFERENT TYPES OF ARDUINO



Arduino NANO

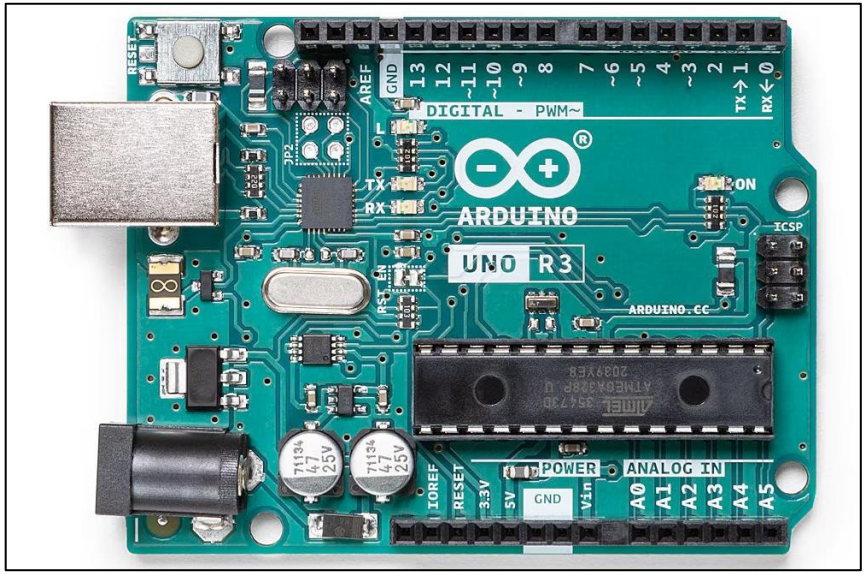


Arduino Micro

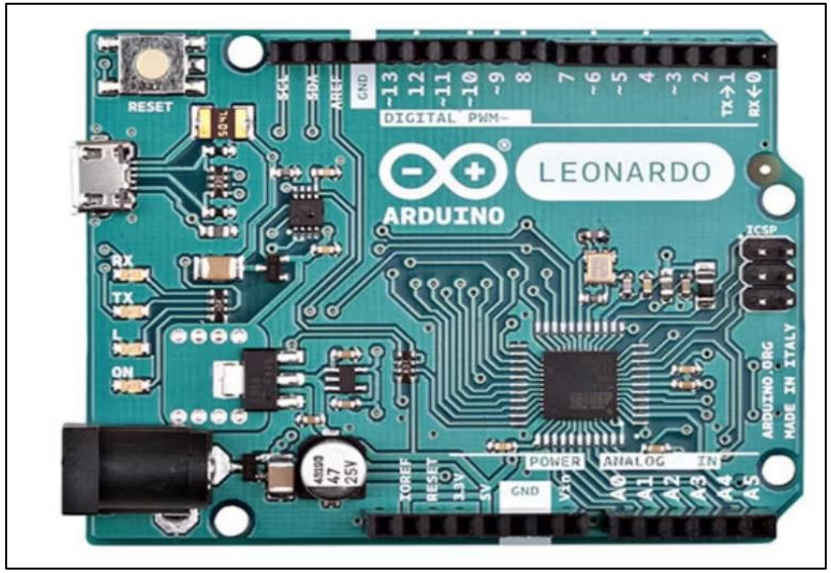


Arduino LilyPad

DIFFERENT TYPES OF ARDUINO

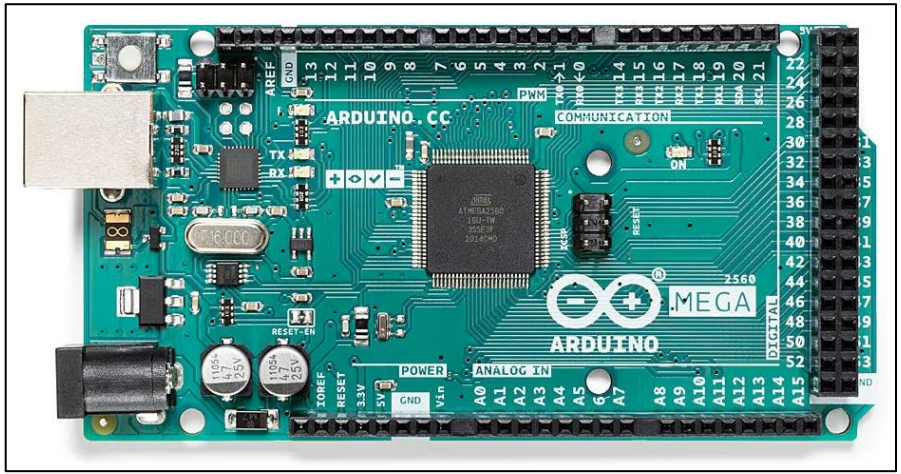


Arduino UNO R3

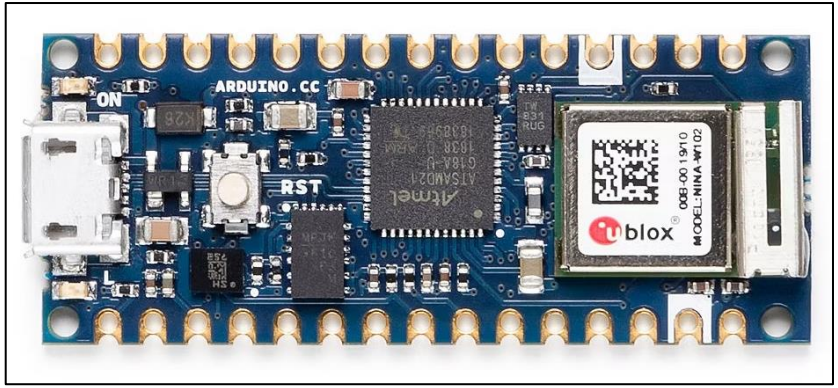


Arduino Leonardo

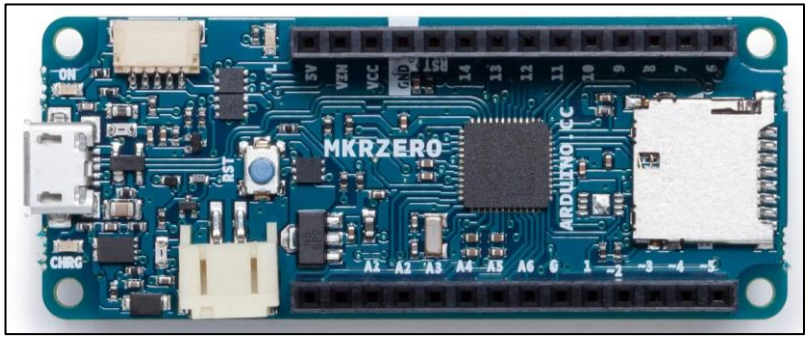
DIFFERENT TYPES OF ARDUINO



Arduino MEGA 2560 R3

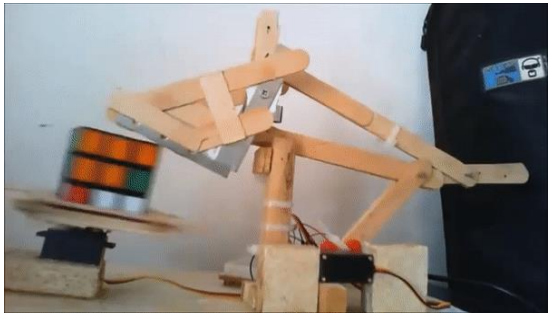


Arduino IOT board

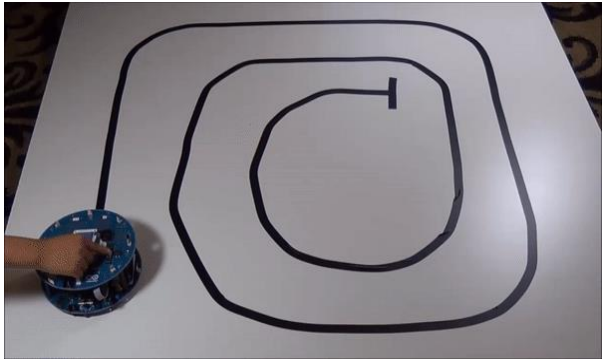


Arduino MKR ZERO

PROJECTS BASED ON ARDUINO

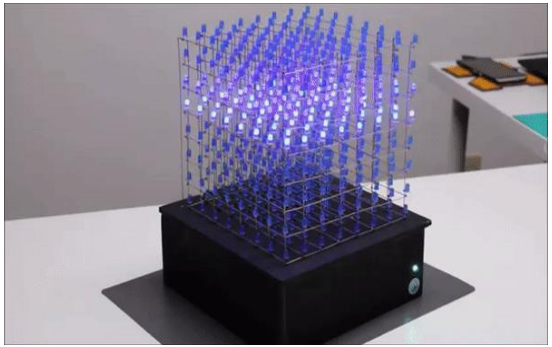


Rubik's cube solver

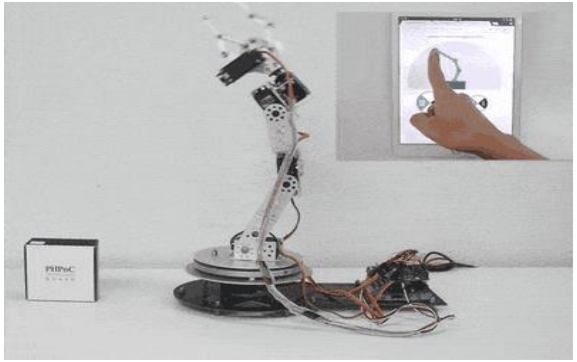


Line following Turtlebot

Spider bot with Arduino



LED cube powered by Arduino



App controlled trainable arm with Arduino

PROJECTS BASED ON ARDUINO

- An example of a human-following cooler called, “**Follow me**”

Follow me cooler with Arduino

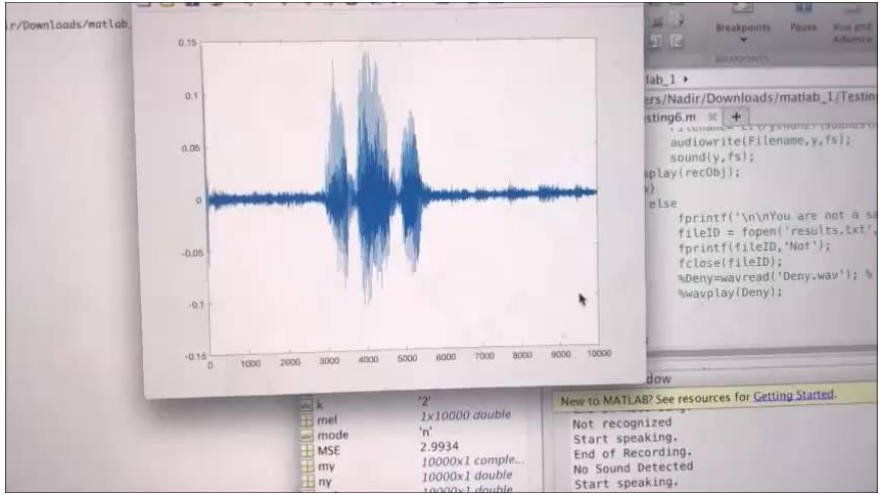
PROJECTS BASED ON ARDUINO - NYU

- Human-following luggage with Arduino:

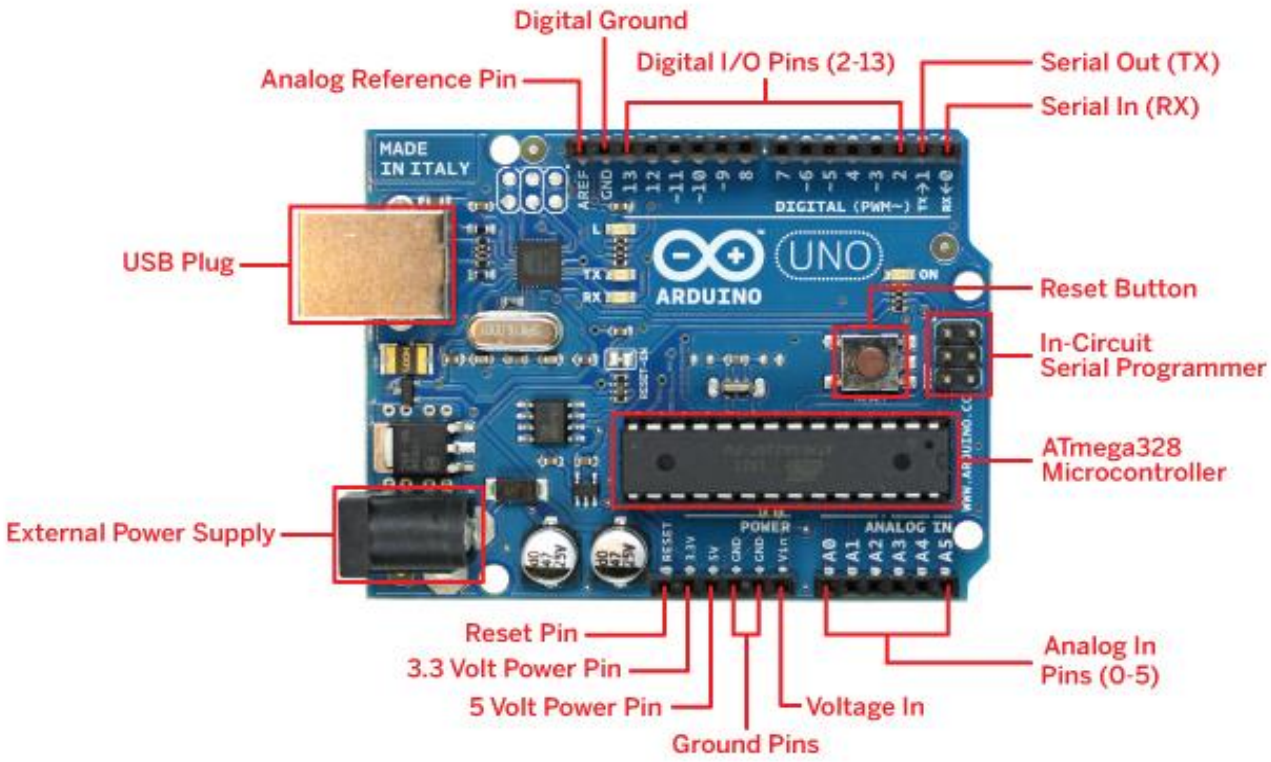
PROJECTS BASED ON ARDUINO - NYU

- Jarvis – Voice assistant with Arduino:

Source



ARDUINO UNO DEVELOPMENT BOARD



Source

ARDUINO HARDWARE SUMMARY

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
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

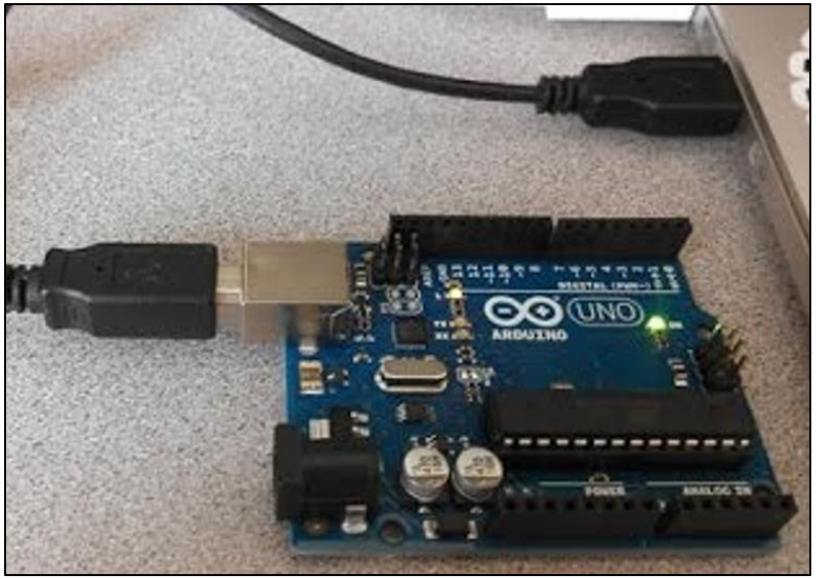
ACTIVITY 2

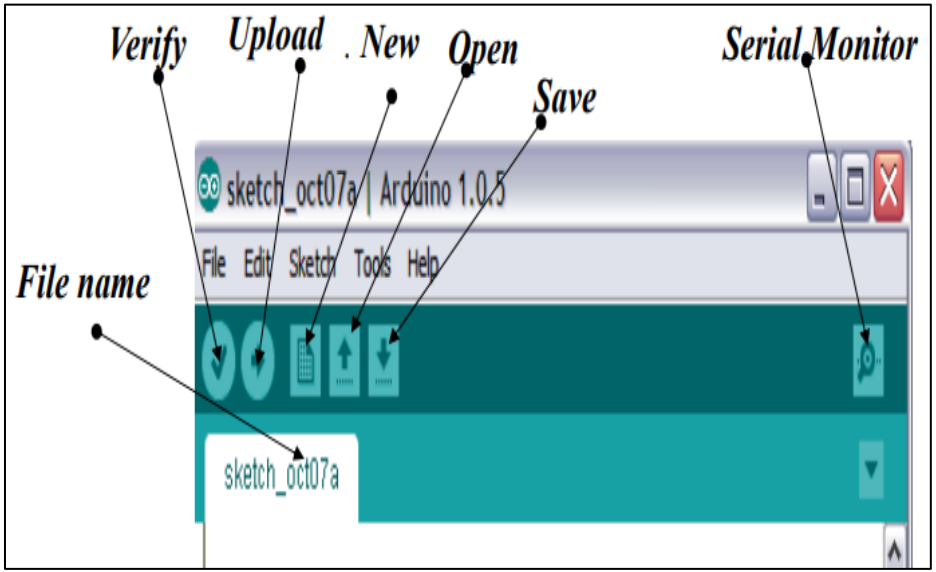
Check out: http://arduino.cc/en/Guide/HomePage

1. Download and install the Arduino environment (IDE) according to your system (Mac or Windows)
2. Connect the board to your computer via the USB cable
3. If needed, install the drivers
4. Launch the Arduino IDE

POWERING UP THE ARDUINO WITH USB

- Connecting via **USB cable** to load the **code** and provide **power** at the same time



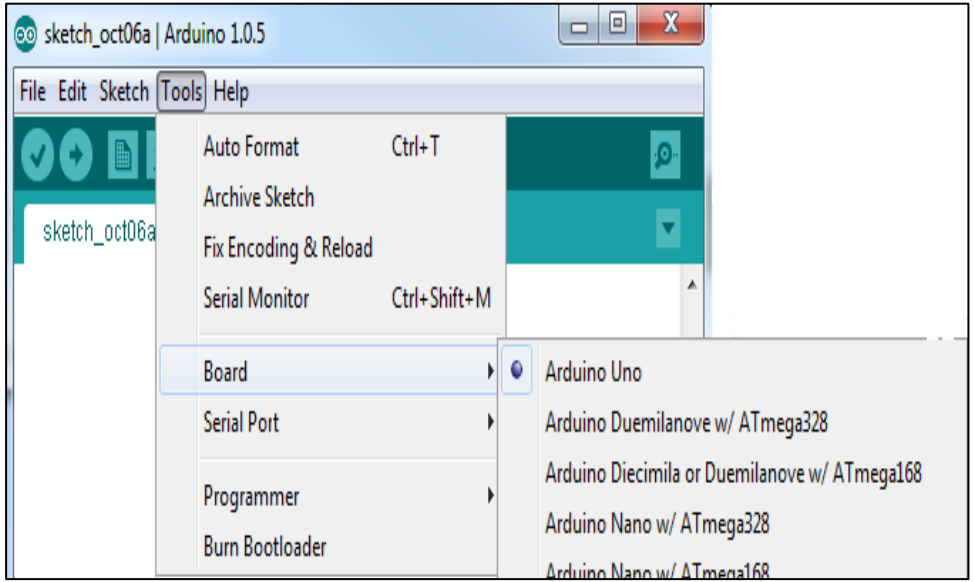


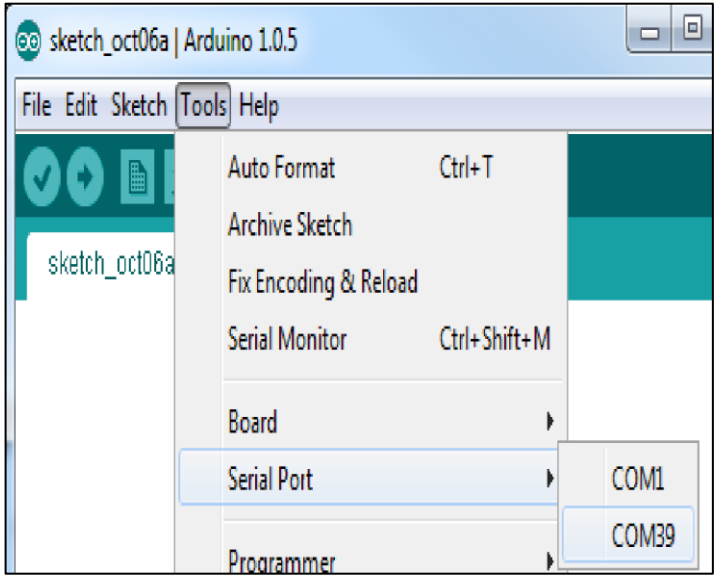
Source

tadbot.com/blog/bionicarduino

SELECT SERIAL PORT AND BOARD

- Select the board **Arduino UNO** and the port showing in the **Serial Port** section





ACTIVITY 3

Source

Program to **blink** the **on-board LED**

1. Select your board
2. Select your serial port
3. Open the blink example
4. Upload the program

ACTIVITY 3 - SOLUTION

- Go to **File** > **Examples** > **Digital** > **Blink**

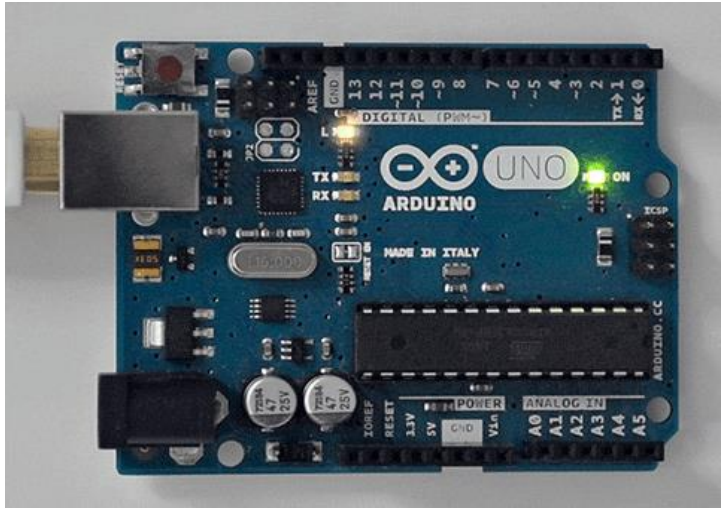
Blink §

```
/*
  Blink

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```



Source

A LITTLE BIT ABOUT THE CODE

Sketch: Arduino code is referred to as sketch and consists of:

void setup() {

- Instructions between the two curly brackets will be **run only once** when the Arduino program first runs and used for the **purpose of setup**
- **Initialize** I/O pins (directions), initialize serial communication, etc.

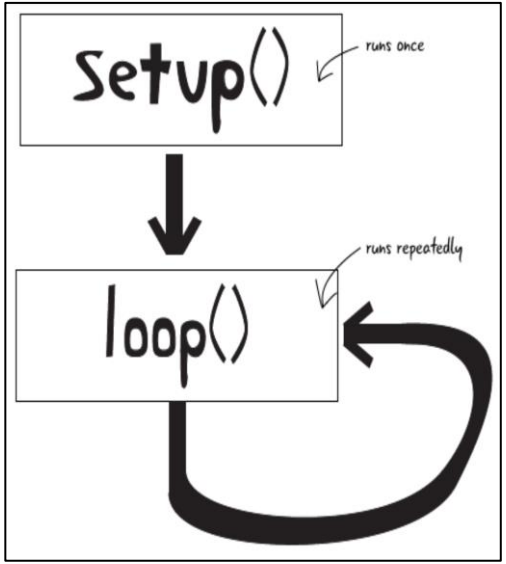
}

void loop() {

- This function is run **after setup** has finished
- It runs in a **continuous loop** until power is removed

}

Sketch



Source

BASIC SYNTAX FOR ARDUINO PROGRAMS

Basic **rules** of programming **Arduino**:

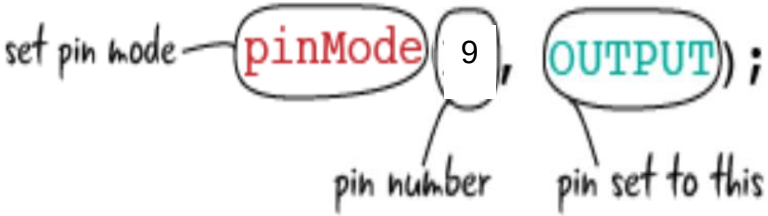
- ***//*** Single line comment
- ***/****
  Multi line comment, or block-comment
- ****/***
- **{ }** Curly braces to indicate starting and ending of a block of code unbalanced curly braces will cause compile-time errors
- **;** Each line of code must end with a semicolon
  NOTE: Semicolon should not be used after #include, #define and for “if, else statements, for, while loops” if code is inside the curly braces

PROGRAMMING: I/O

pinMode(pin, mode)

- **pin** refers to digital **I/O pin number** and **mode** argument refers to **INPUT** or **OUTPUT**
- Can use digital I/O pin 2 to 13 (**0** and **1** used for **RX** and **TX**)
- Digital I/O pins have **default mode** as **INPUT**, no need to explicitly declare as inputs

➢ **Connection:** Connect an OUTPUT pin to an external device in series with a 470Ω or 1KΩ resistor



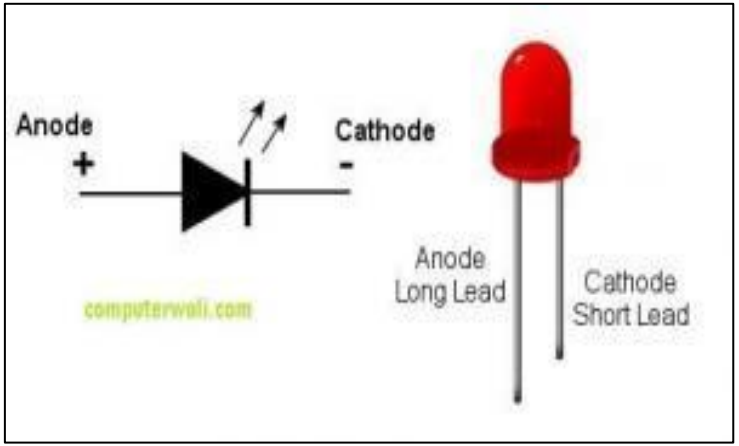
UPLOADING YOUR CODE

- Select your board **Arduino UNO**
- Select your **port** from **tools**
- **Upload** the code to the Arduino as shown
- Check for **errors**
- Check if the code is **compiled** successfully

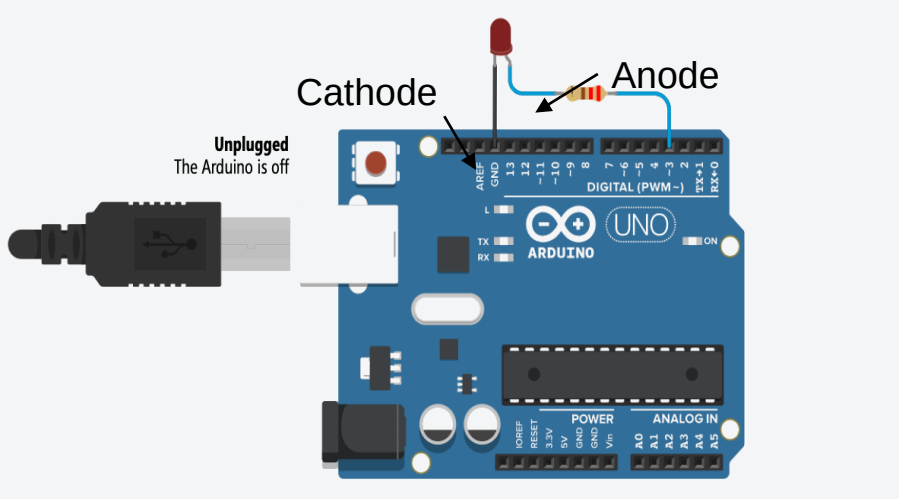
BLINK AN EXTERNAL LED

Things to remember:

1. Longer leg **(Anode)** of the **LED** goes to one end of the **resistor in series** and the other end of resistor goes to **pin (3)** here
2. Shorter leg **(Cathode)** goes to **ground (GND)**



Source



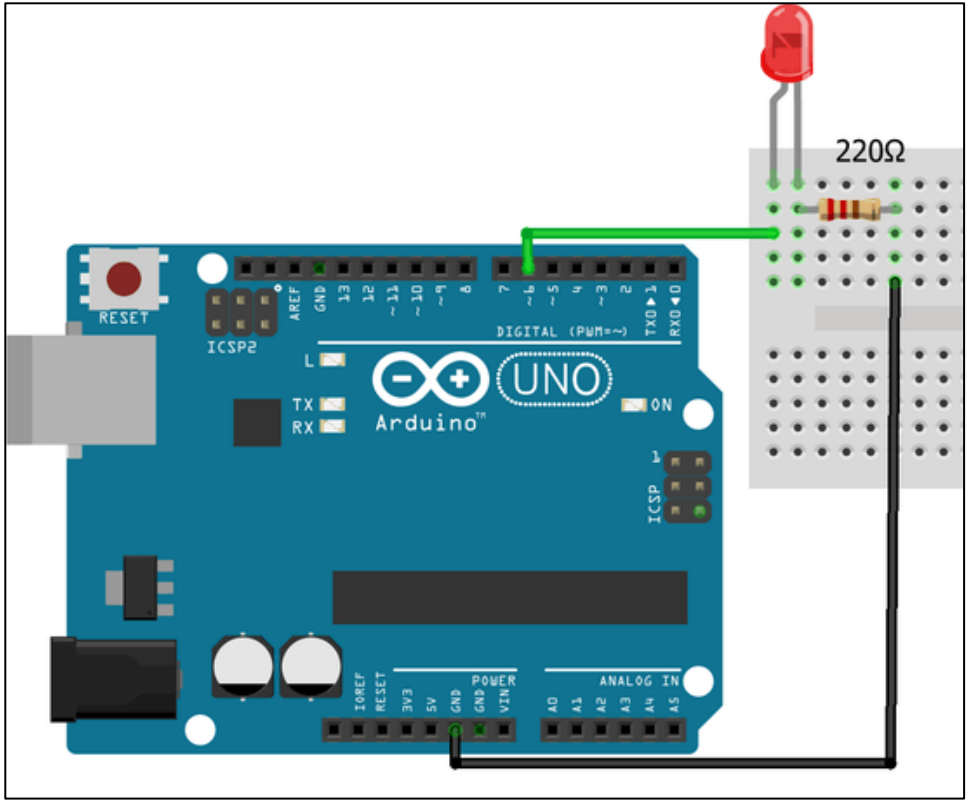
Blink LED

ACTIVITY 4

- Write a program to blink one external LED using Arduino

ACTIVITY 4 - SOLUTION

- Connections:
  Pins: 6, GND



Source

ACTIVITY 4 - SOLUTION

- Code:

Blink_oneExternalLED | Arduino 1.8.19

File Edit Sketch Tools Help

Blink_oneExternalLED §

```
void setup() {
pinMode(6, OUTPUT);  // initialize digital pin 6 as an output
}
void loop() {
digitalWrite(6, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(1000);    // wait for a second
digitalWrite(6, LOW);   // turn the LED off by making the voltage LOW
delay(1000);            // wait for a second
}
```

PROGRAMMING ESSENTIALS: I/O

- **Digital I/O pins:**

PROGRAMMING ESSENTIALS: I/O

digitalRead(pin)

- **Reads** the value from a specified digital pin with the result being **1 (HIGH)** or **0 (LOW)**
- Pin can be specified as a variable or constant (0-13)

➢ **Syntax:** value = digitalRead(pin); //sets 'value' equal to the state of the input pin

digitalWrite(pin, value)

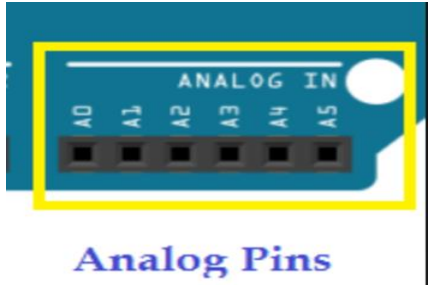
- Pin refers to a digital pin, can be a variable or a constant (0-13)
- Value is either logic level **HIGH/LOW, TRUE/FALSE, 1/0**
- **Outputs** either logic level **HIGH** or **LOW** at a specified digital pin

➢ **Syntax:** digitalWrite(pin, value); //sets 'value' as the state of the input pin

PROGRAMMING ESSENTIALS: I/O

analogRead(pin)

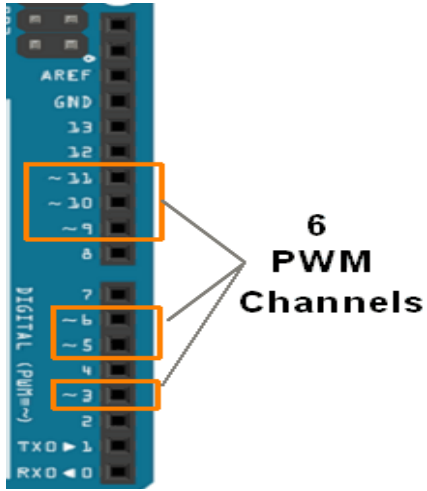
- Reads the value from a specified analog pin with a 10-bit resolution
- Pin argument can be specified as a variable or constant (0-5)
- Resulting integer values range from 0 to 1023 and must be scaled to appropriate units



➢ **Syntax:** value = analogRead(pin); //sets 'value' equal to analog reading on 'pin'

analogWrite(pin, value)

- **Outputs** a **PWM signal** to the specified output **pin**
- This function works on pins 3, 5, 6, 9, 10, and 11
- **Value** can be specified as a variable or constant with range 0-255



➢ **Syntax:** analogWrite(pin, value); //writes 'value' to 'pin'

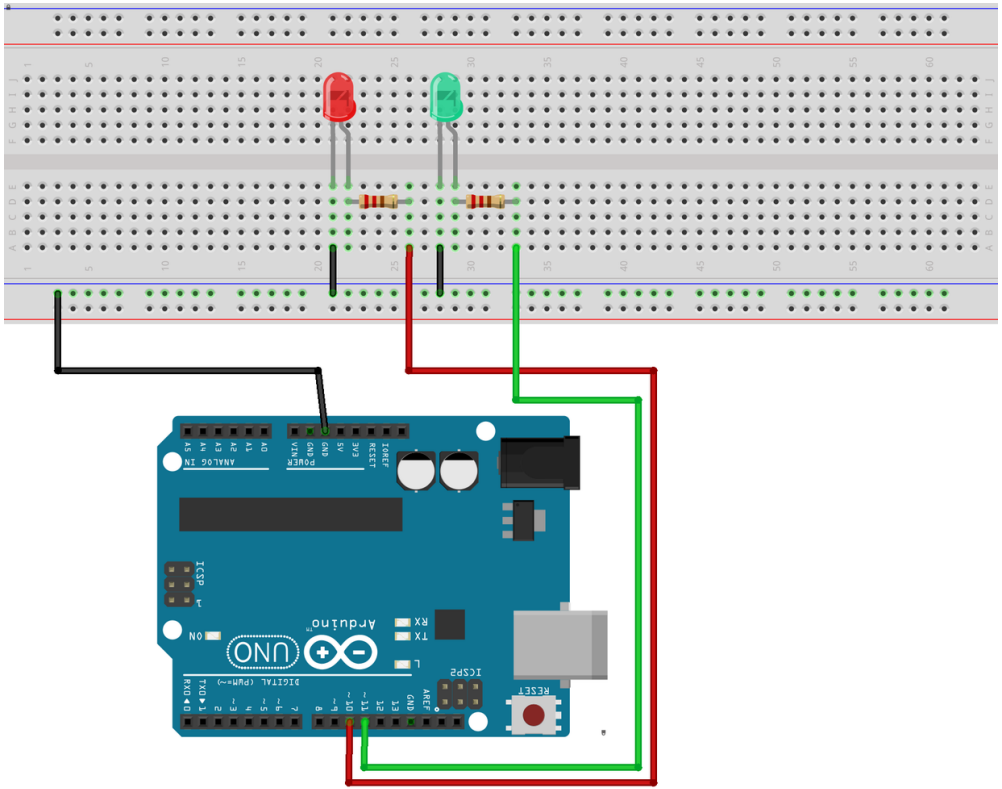
ACTIVITY 5

- Attach 2 LEDs to your Arduino, each to a separate pin, blink one LED only once and the other repeatedly

Source

ACTIVITY 5 - SOLUTION

- Connections:
  Pins: 10, 11, GND



Source

Source

ACTIVITY 5 - SOLUTION

- Code:

Blink_twoExternalLEDs | Arduino 1.8.19

File Edit Sketch Tools Help

Blink_twoExternalLEDs §

```
void setup()
{
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
delay(100);

// Blink LED on pin 11 once
digitalWrite(11, HIGH);
delay(1000);
digitalWrite(11, LOW);
delay(1000);
}
```

```
void loop()
{
// Blink LED on pin 10 forever
digitalWrite(10, HIGH);
delay(1000);
digitalWrite(10, LOW);
delay(1000);
}
```

PROGRAMMING ESSENTIALS

Serial.begin(rate)

- Opens serial port and sets the baud rate for serial data transmission
- Typical baud rate for communicating with PC is **9600** although **<u>other speeds are supported</u>**
- When using serial communication, digital pins **0 (RX)** and **1 (TX)** cannot be used at the same time

delay(ms)

- **Pauses** the program for **time** as specified in **milliseconds**, with 1000 being equal to 1 second

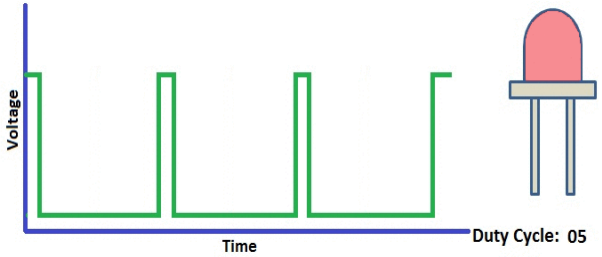
PWM BASIC CONCEPTS

Pulse width modulation (PWM): Converts a digital value to analog output

Pulse width modulation (PWM) allows Arduino to generate **a series of pulses**

When a pin is **output high**, the **apparent** voltage at that pin will be **close to 5 V**

When the pin is made **output low** it is close to **0 V**



Source

Source

PWM BASIC CONCEPTS

- **Duty cycle: Higher** the **duty cycle, higher** is the **voltage**
- **Duty** specifies the analog output level as a **fraction of 256ths of 5 V** ranging from 0 to 4.98 V

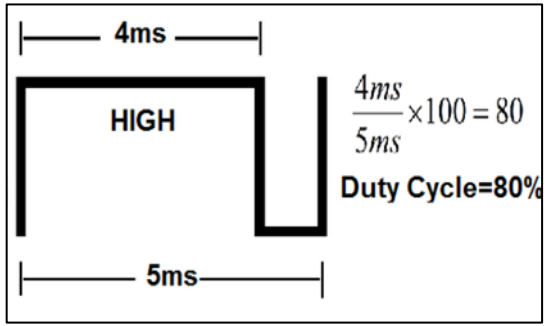
$$\text{duty cycle} = \frac{t_{ON}}{t_{ON} + t_{OFF}}$$

t_{ON} = ON time

t_{OFF} = OFF time

$t_{ON} + t_{OFF}$ = Time period

Source



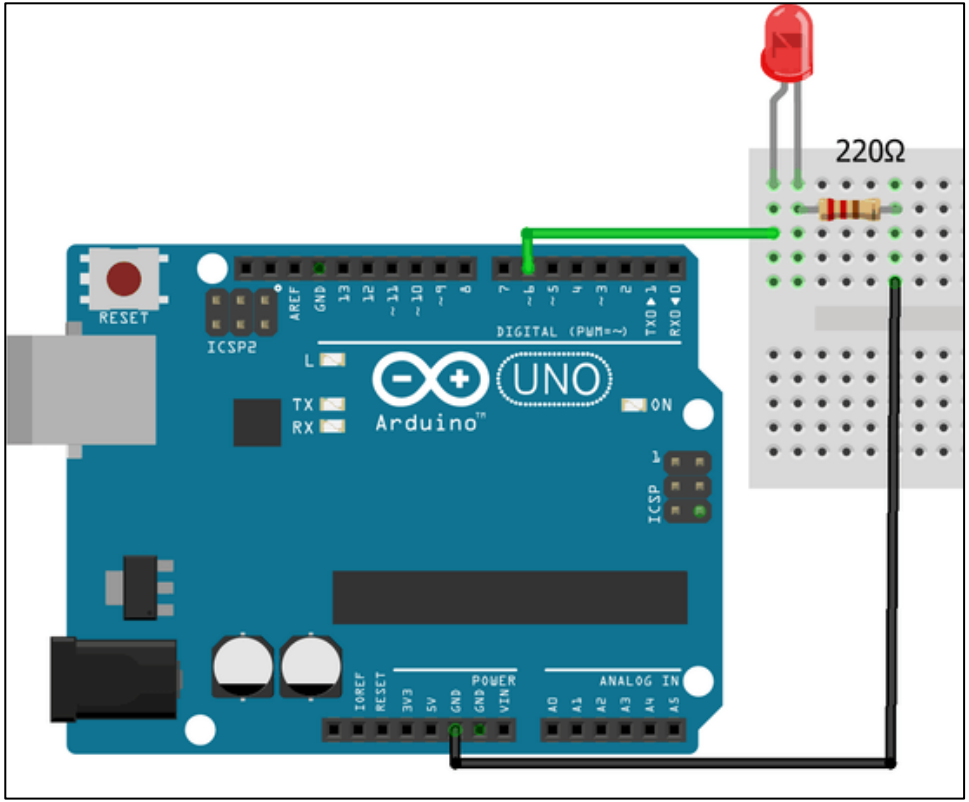
Source

ACTIVITY 6

- Attach one LED to your Arduino and write a program to change its brightness

ACTIVITY 6 - SOLUTION

- Connections:
  Pins: 6, GND



Source

ACTIVITY 6 - SOLUTION

- Code:

Change_brightnessLED | Arduino 1.8.19

File Edit Sketch Tools Help

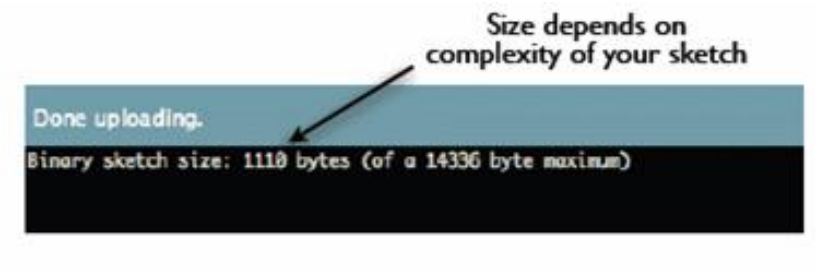
Change_brightnessLED §

```
void setup() {
pinMode(6,OUTPUT);   // initialize digital pin 6 as an output
}

void loop() {
analogWrite(6,255);  // turn the LED on to highest brightness value
delay(200);     // wait for 200  milliseconds
analogWrite(6,100);  // turn the LED on to a medium brightness value
delay(200);       // wait for 200  milliseconds
analogWrite(6,0);    // turn the LED on to lowest brightness value/turn OFF
delay(200);       // wait for 200  milliseconds
}
```

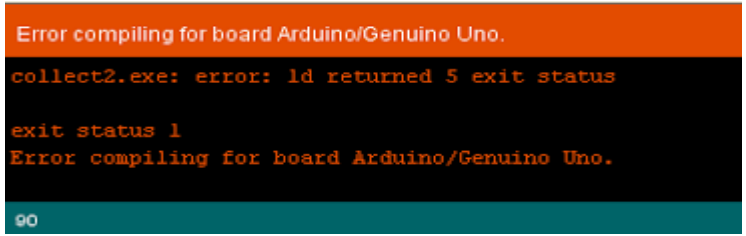
STATUS MESSAGES

Successful upload:

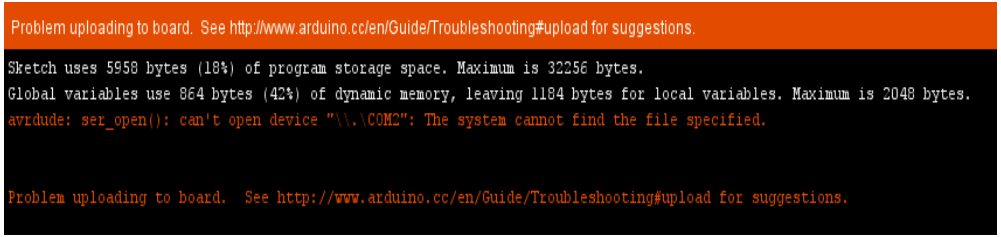


Wrong port selected:

Error in compiling:



Wrong board selected:

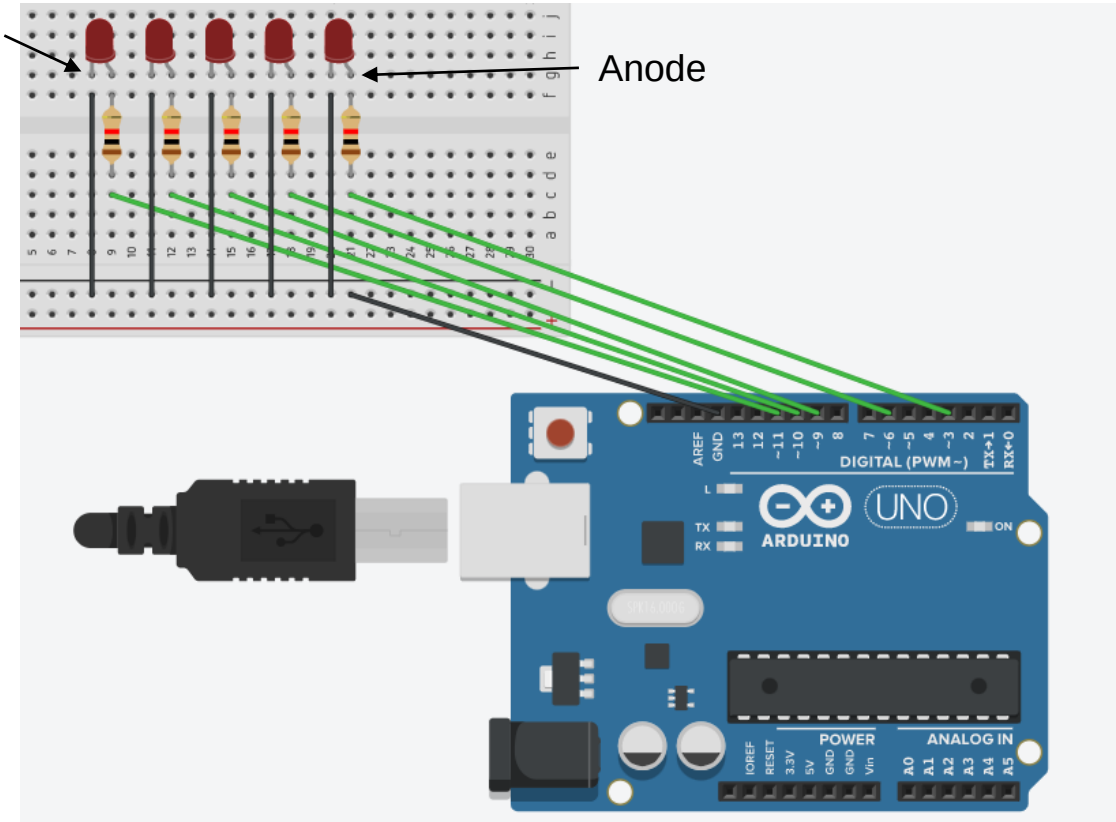


ACTIVITY 7

- Attach at least 5 LEDs to your Arduino, each to a separate pin. Blink them together

ACTIVITY 7 - SOLUTION

- Connections:
  Pins: 3, 6, 9, 10, 11, GND



ACTIVITY 7 - SOLUTION

- Code:

Blink_5LEDs | Arduino 1.8.19

File Edit Sketch Tools Help

Blink_5LEDs

```
void setup()
{
pinMode(11,OUTPUT);   // initialize digital pin 11 as an output
pinMode(10,OUTPUT);  // initialize digital pin 10 as an output
pinMode(9,OUTPUT);   // initialize digital pin 9 as an output
pinMode(6,OUTPUT);   // initialize digital pin 6 as an output
pinMode(3,OUTPUT);   // initialize digital pin 3 as an output
}
```

ACTIVITY 7 - SOLUTION

```
void loop()
{
// make pins 3, 6, 9, 10, 11 HIGH

digitalWrite(11, HIGH);
digitalWrite(10, HIGH);
digitalWrite(9, HIGH);
digitalWrite(6, HIGH);
digitalWrite(3, HIGH);

// Delay for a second
delay(1000);
```

```
// make pins 3, 6, 9, 10, 11 LOW

digitalWrite(11, LOW);
digitalWrite(10, LOW);
digitalWrite(9, LOW);
digitalWrite(6, LOW);
digitalWrite(3, LOW);

// Delay for a second
delay(1000);
}
```

VARIABLES

- A **variable** is a place to store a piece of data
- It has a name, a value, a type and size accordingly

- **Syntax:** variable_type variable_name = value;

Type	Bytes(size)	Example
boolean	1	boolean led_on=True;
char	1	char char_1 = ‘a’;
int	2	int temp = 48;
float	4	float height = 2.5;
long	4	long time = 5;

VARIABLES

- Save the results in different variable types:

int a = 50;
int b = 30;
int c = 0;
c = a / b;
 	
Output: 1
	int a = 50;
	int b = 30;
	float c = 0;
	c = a / b;
	
	Output: 1.00
		float a = 50;
		float b = 30;
		float c = 0;
		c = a / b;
		
		Output: 1.66
			float a = 50;
			float b = 30;
			int c = 0;
			c = a / b;
			
			Output: 1

ACTIVITY 8

- Divide 5 by 2 and store the value as int and as float

ACTIVITY 8 - SOLUTION

int a = 5;
int b = 2;
int c = 0;
c = a / b;
 	
Output: 2
	int a = 5;
	int b = 2;
	float c = 0;
	c = a / b;
	
	Output: 2.00

NYU TANDON SCHOOL OF ENGINEERING

Thank You!

Questions and Feedback?