

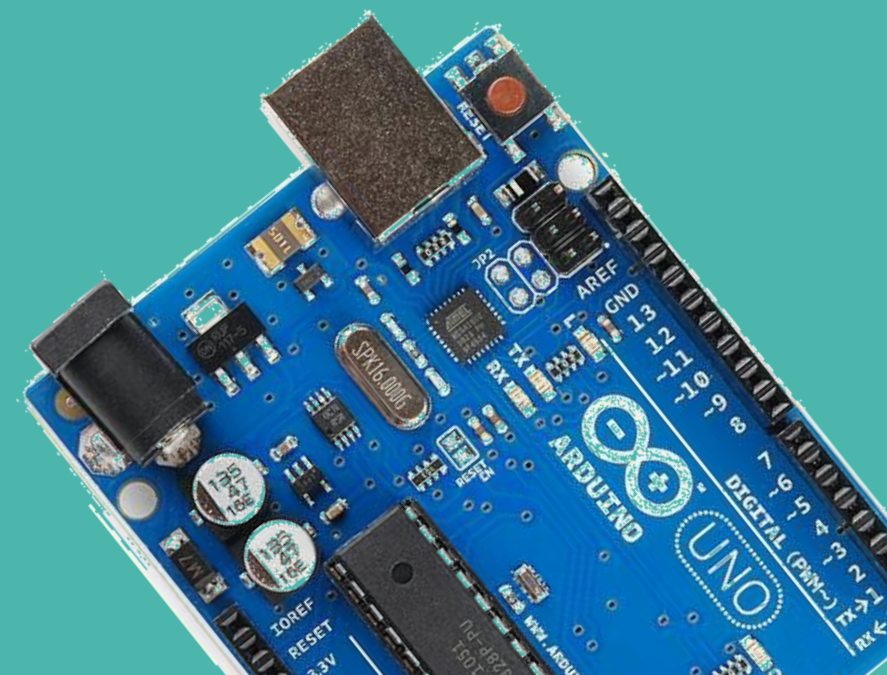
# AgTech Club

Exploring the Intersection between Agriculture and Modern Technologies



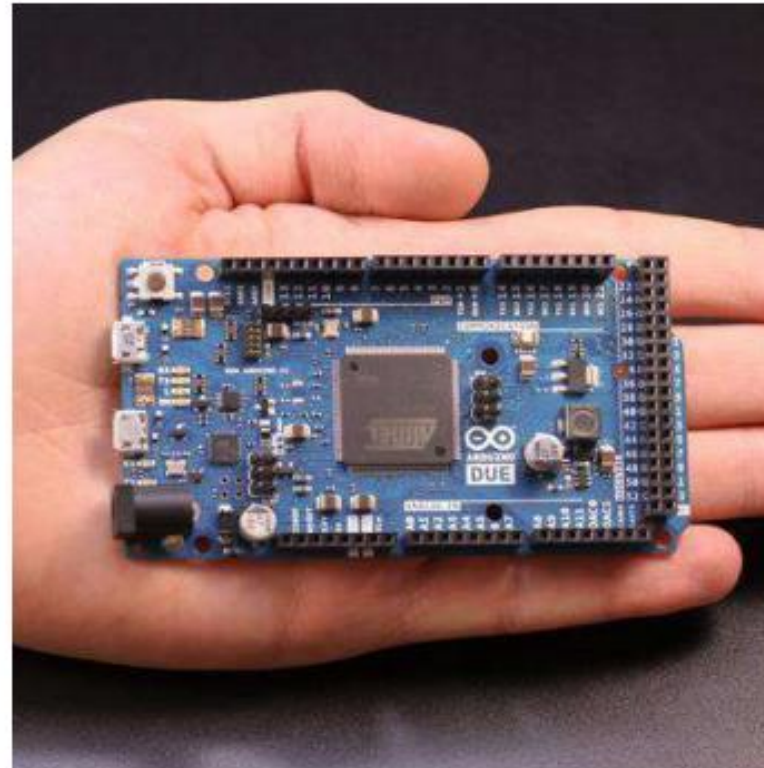
# Agtech club Arduino workshop

<https://github.com/mucyo-coder/ArduinoWorkshop> AgTech Club RICA/



# What is Arduino

- Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software.
- The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller, though a new model has been designed around a 32-bit Atmel ARM





# Types of Arduino



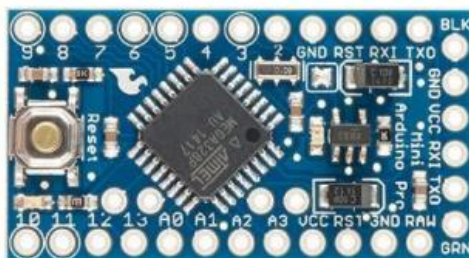
UNO



PRO



MICRO



PRO MINI

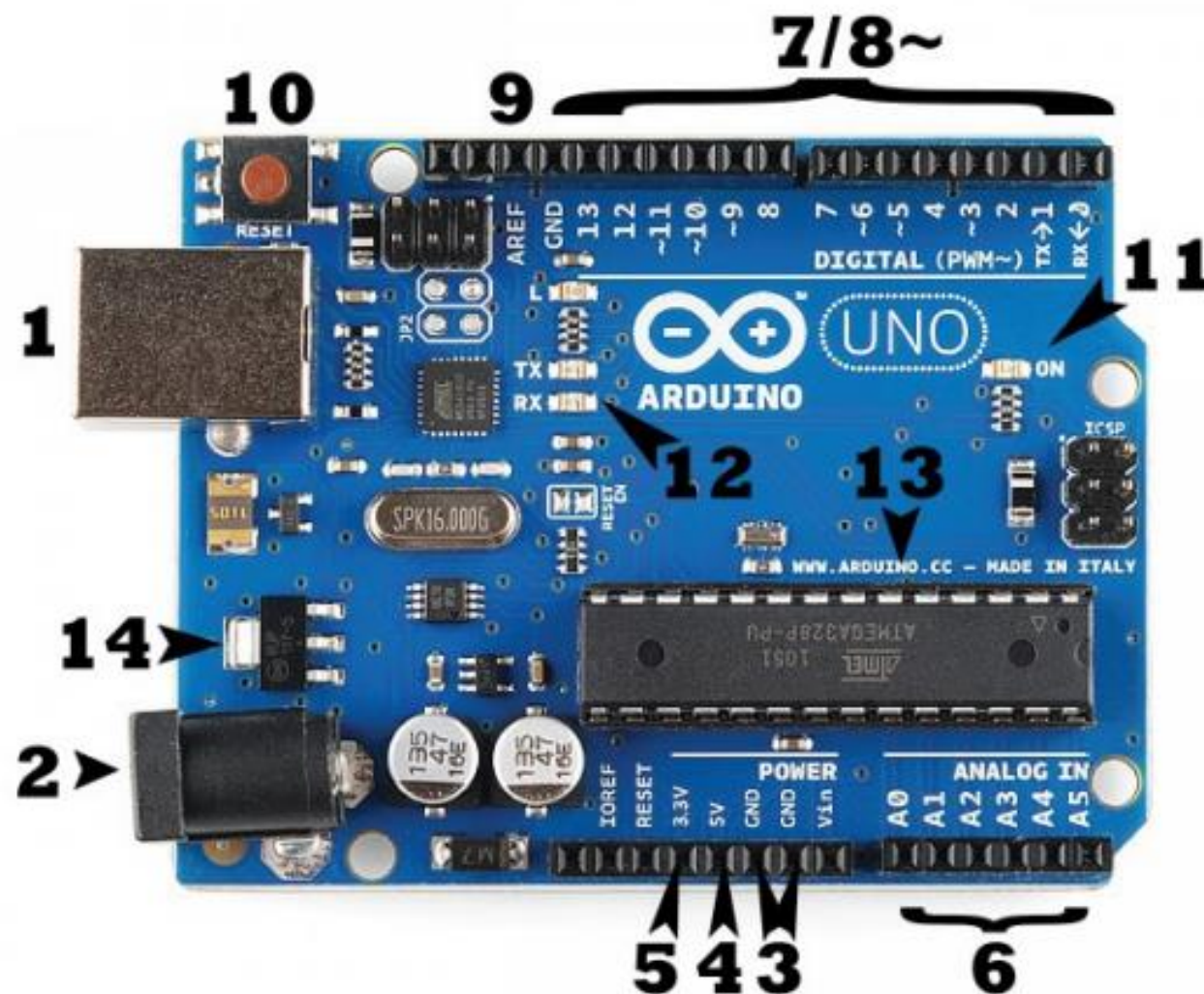


NANO

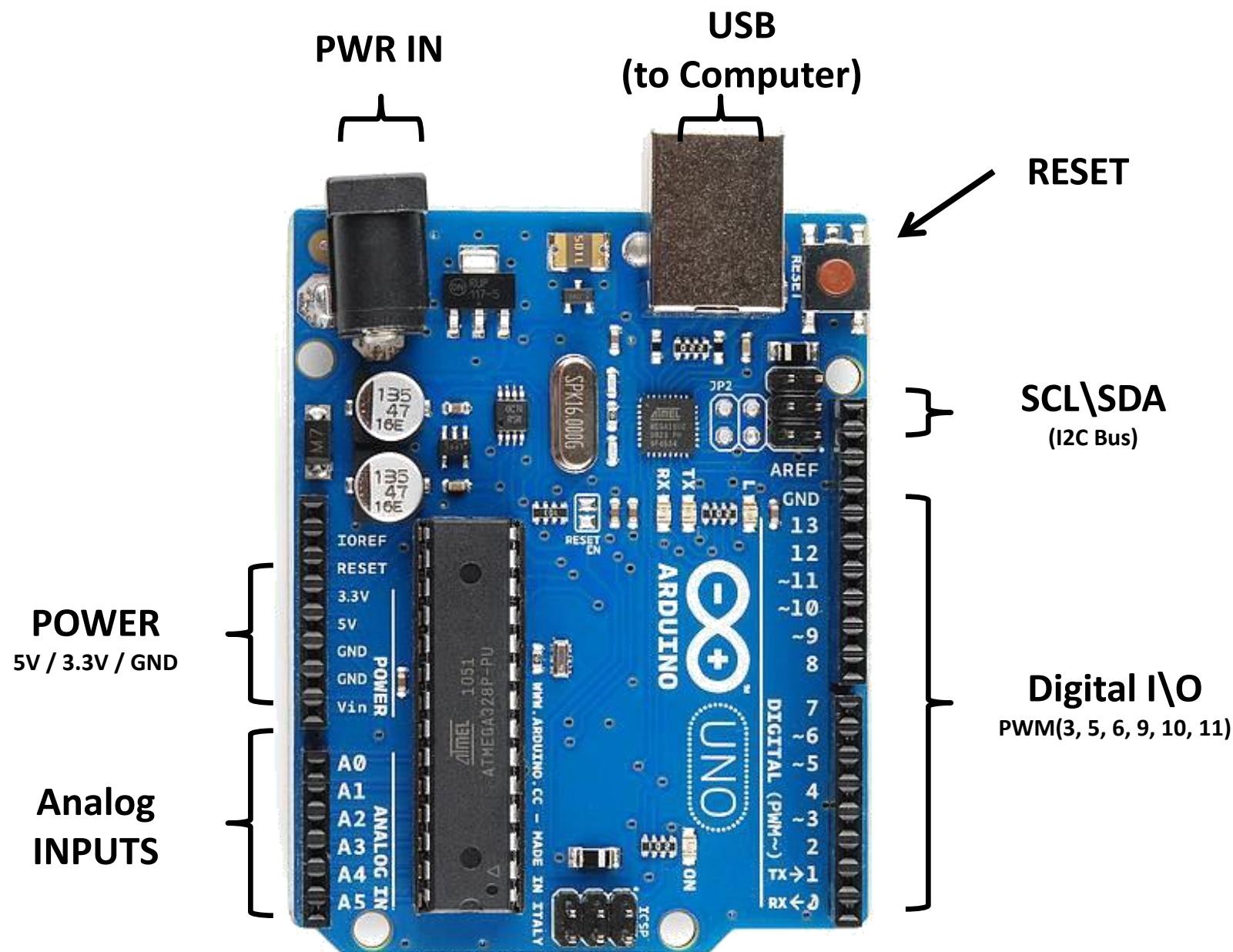


MEGA

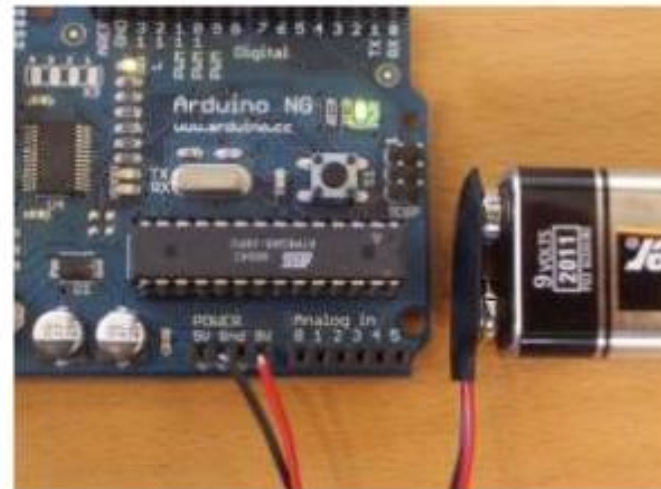
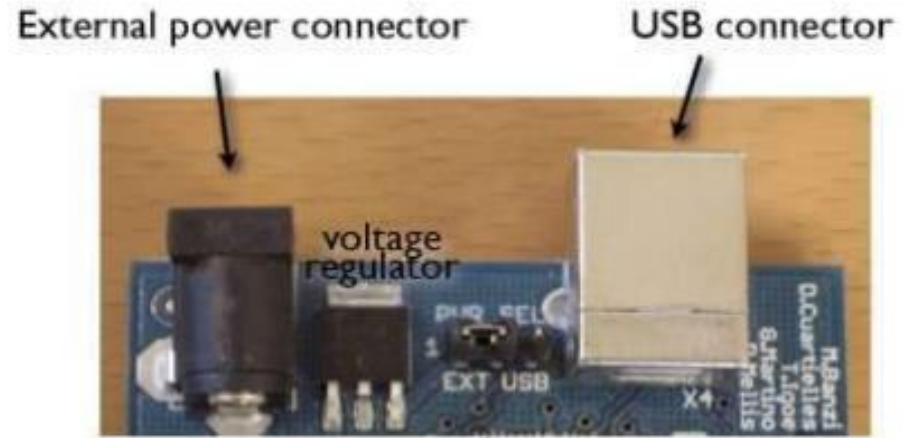
# Arduino UNO







# Powering it up



Be careful about polarity! And shorts!




also solves polarity concerns

# How to code in arduino

- You need to download Arduino IDE (Integrated Development Environment).
- Arduino IDE is available for all Mac, Windows.and Linux.

## Download the Arduino Software



### ARDUINO 1.6.4

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer  
**Windows** ZIP file for non admin install

**Mac OS X** 10.7 Lion or newer

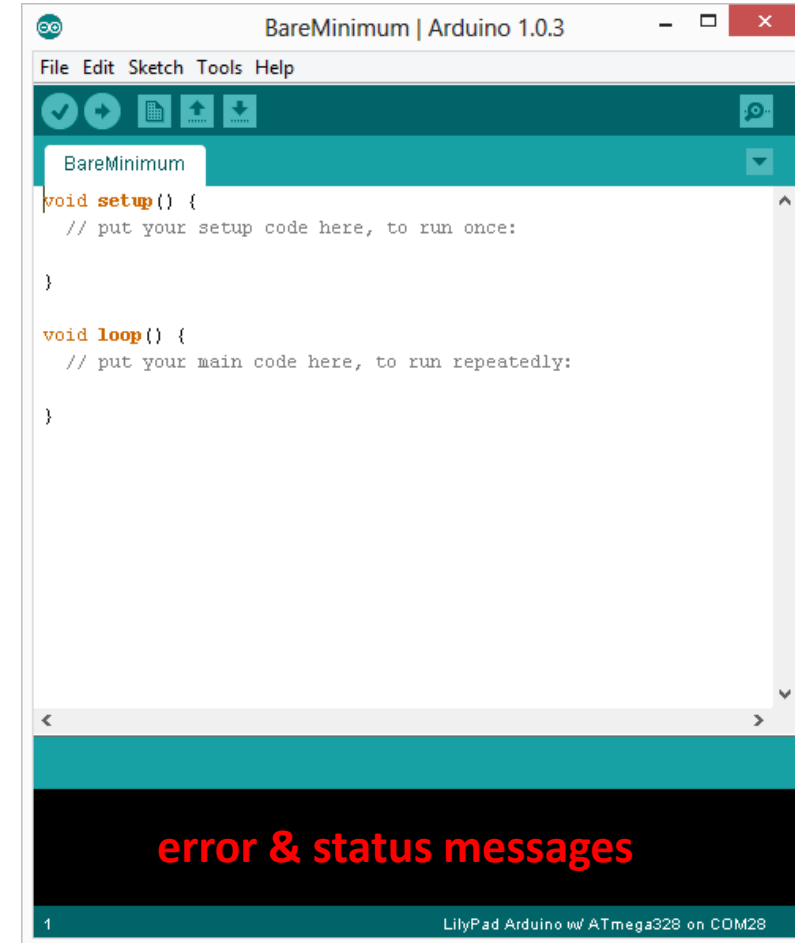
**Linux** 32 bits  
**Linux** 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums](#)



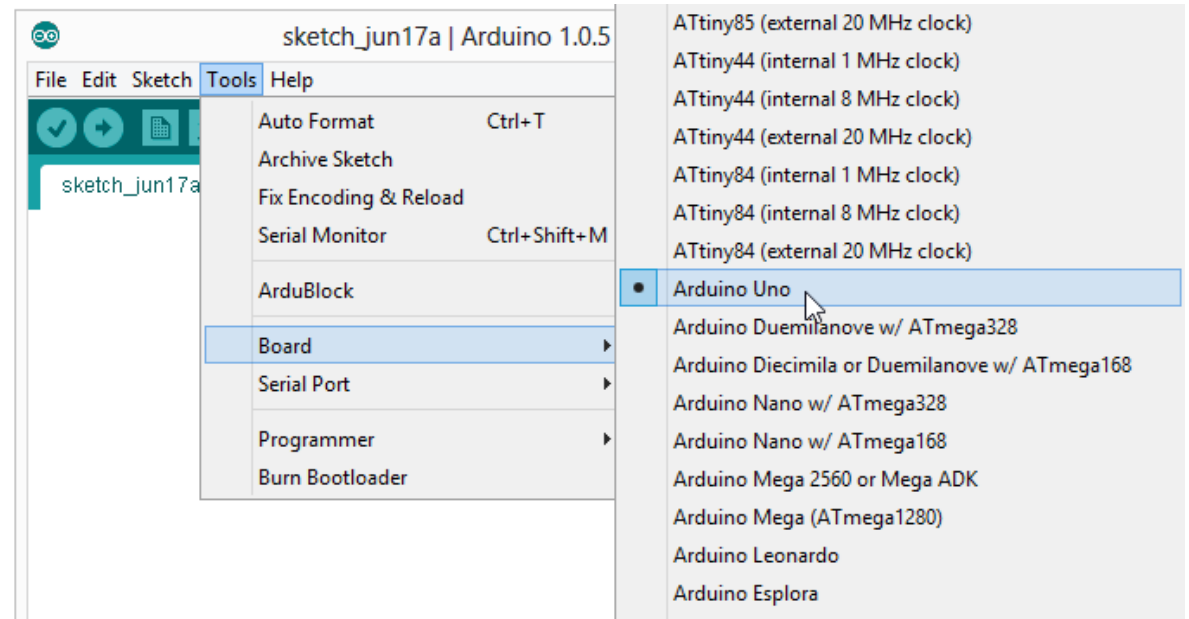
# How to code in arduino

- Once you have downloaded and installed/extracted the folder, you can directly run Arduino.exe, which will take you to its IDE.
- The IDE will look like the shown screenshot.



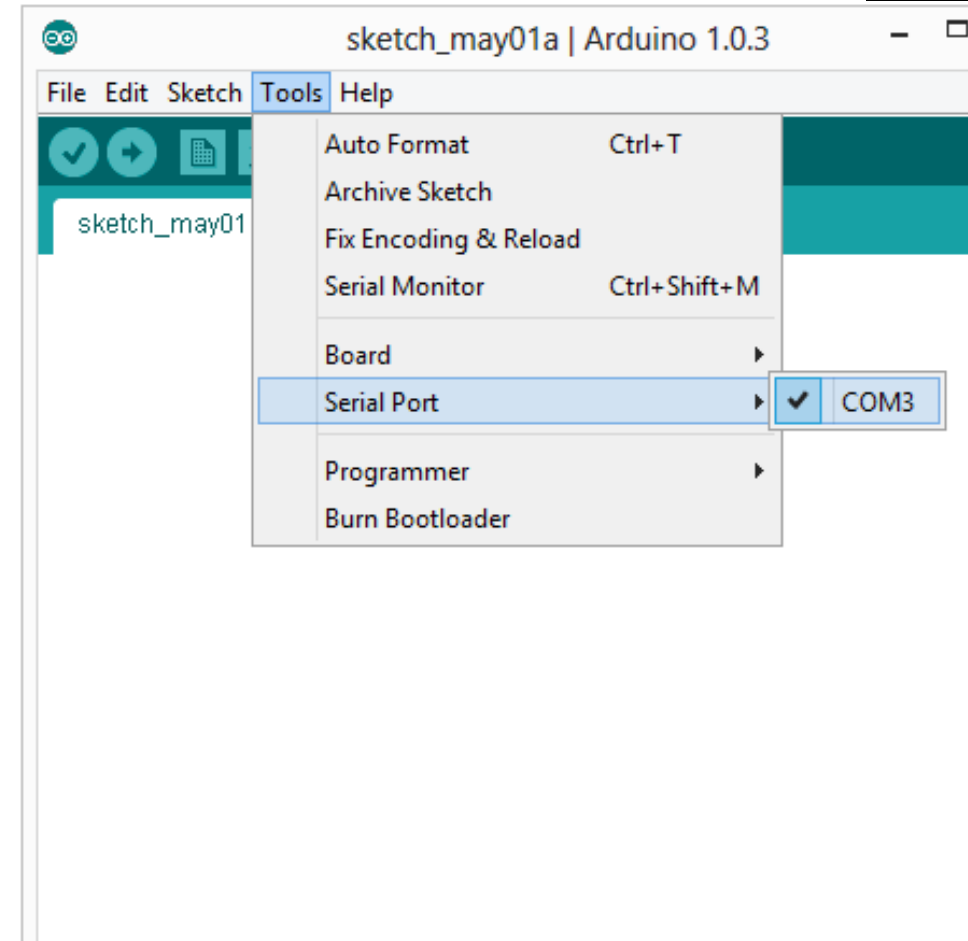
# Program your arduino

- Before you start programming, double check that correct board is selected under Tools → Board.
- Now, you can start playing with Arduino.



# Program your arduino

- The Arduino Uno can be programmed with the Arduino software. Select "Arduino Uno" from the Tools > Board menu (according to the microcontroller on your board).
- All the peripheral connected with Computers are using Serial Port.
- You can check port for Arduino Uno in Device Manager.

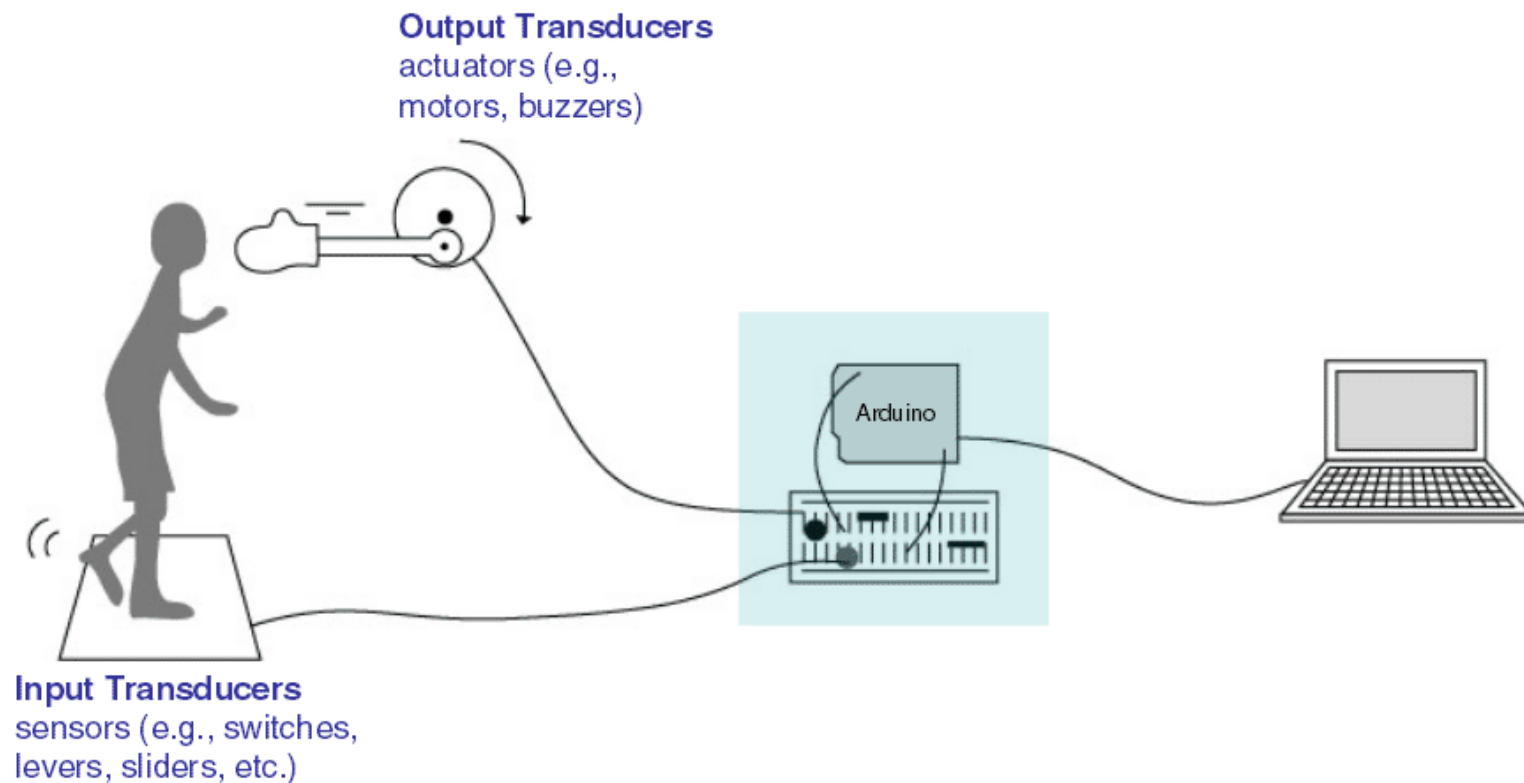




# 7 Major concepts

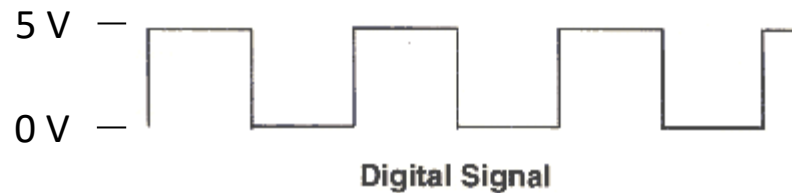
- pinMode()
- digitalWrite()
- analogWrite()
- digitalRead()
- If (statements) / Boolean
- analogRead
- Serial Communication

# Input vs output



# 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.





# Analog vs digital

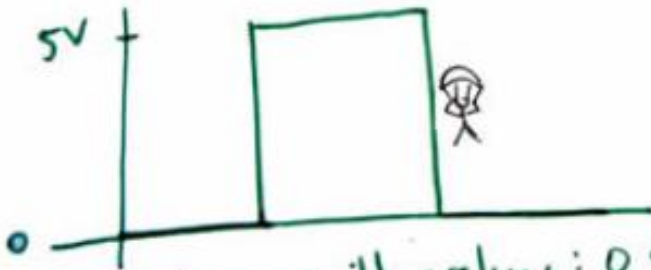
## Analog vs Digital

Analog:



Any value between 0 and 5V is possible

Digital:



Only two possible values: 0 & 5V  
Also known as 0,1 or Low, HIGH

### Why digital?

- Digital is simple, but lets us use complex math.
- Digital is much less sensitive to noise
- Digital opens up the possibility of software

### Why analog?

- our world is analog!
- Digital is just a special case of analog

# Analog vs digital

- Analog Sensors

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

- Digital Sensors

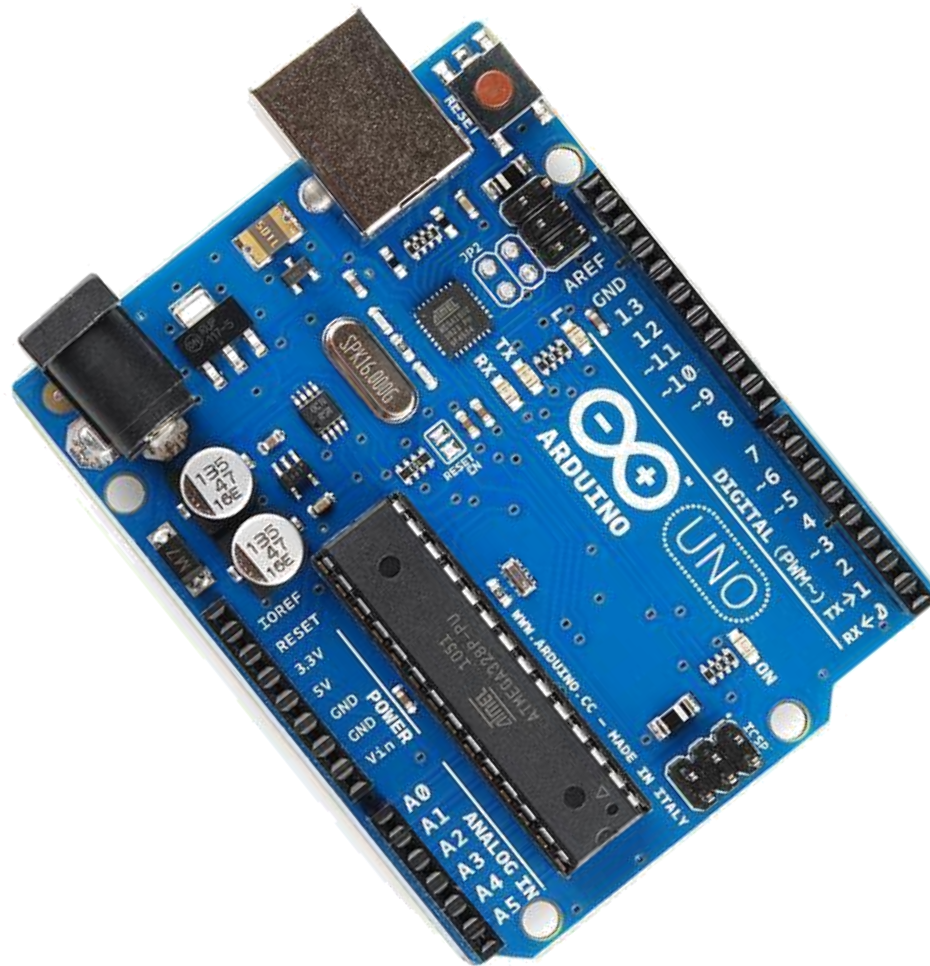
- Digital sensors are more straight forward than Analog.
- No matter what the sensor there are only two settings: On and Off
- Example, Push button, Switch

# Serial communication

- “Serial” because data is broken into bits, each sent one after another in a single wire.
- Compiling turns your program into binary data (ones and zeros)
- Uploading sends the bits through USB cable to the Arduino
- The two LEDs near the USB connector blink when data is transmitted.
  - RX blinks when the Arduino is receiving data.
  - TX blinks when the Arduino is transmitting data



# Let play with it



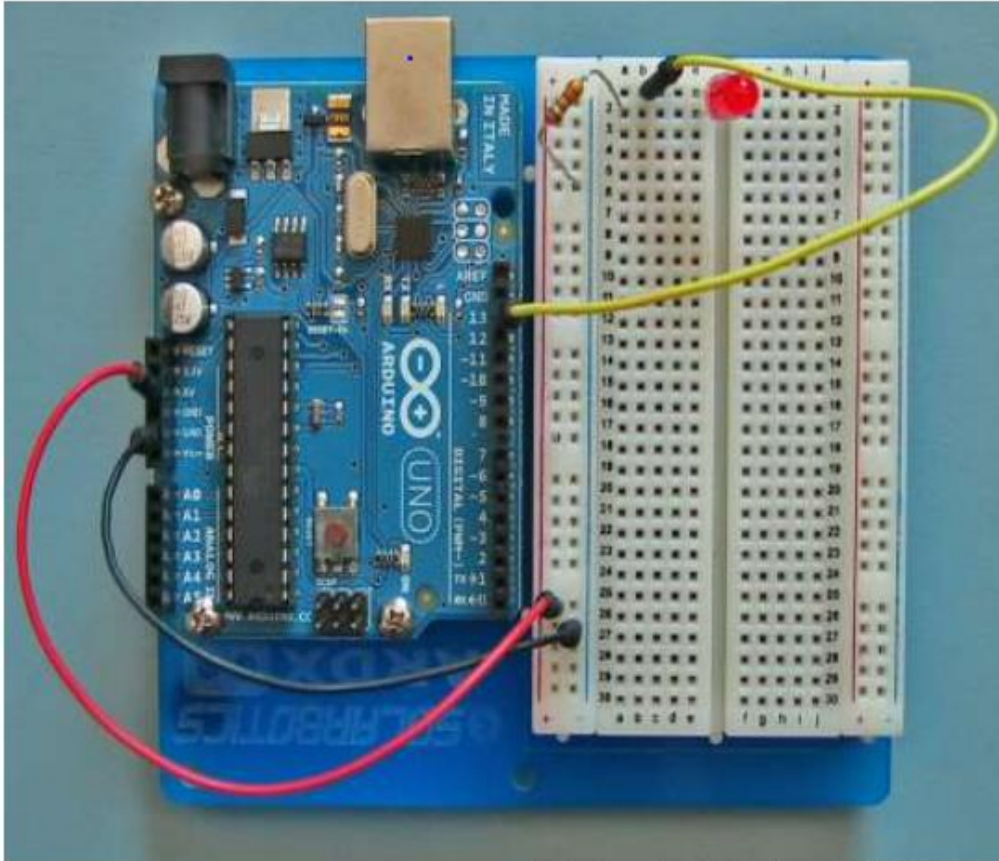
# #1 Hello World for Arduino



```
• void setup() {  
•   pinMode(13, OUTPUT);           //setup pin 13 as an output  
• }  
•  
• void loop() {  
•   digitalWrite(13, HIGH);        // turn the LED on (HIGH  
  outputs 5V)  
•   delay(500);                    // wait for 500 milliseconds  
•   digitalWrite(13, LOW);         // turn the LED off (LOW  
  outputs 0V)  
•   delay(500);                    // wait for 500 milliseconds  
• }
```

# #2 Blink

- Connect Arduino Vcc and GND to breadboard power rails
- Put in a LED on breadboard
- D13> 220 Ohm resister -> LED -> GND

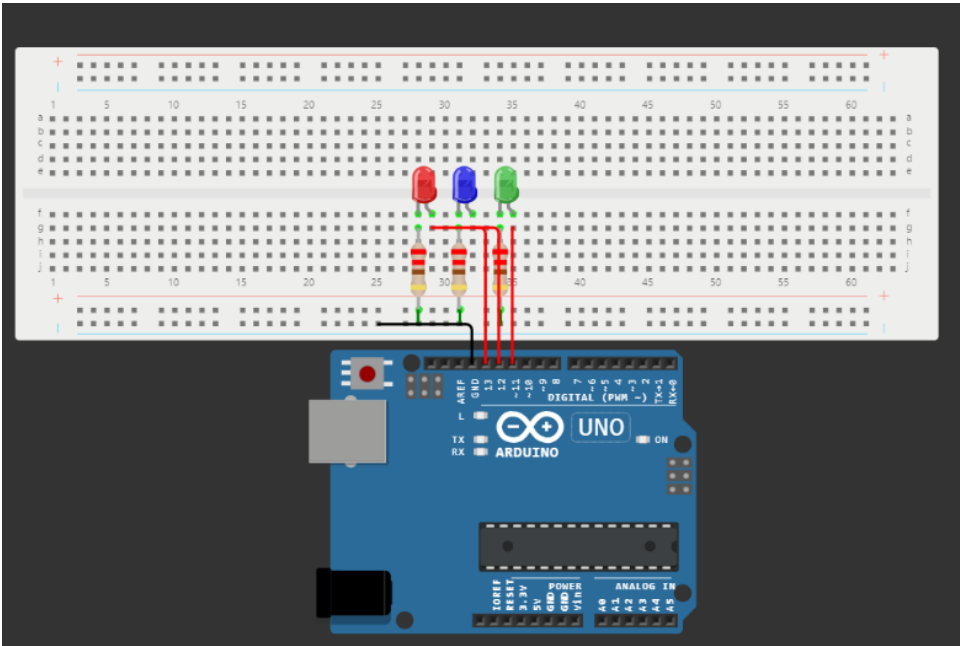


## • Software

- `/*`
- `Blink`
- `Turns on an LED on for one second, then off for one second, repeatedly.`
- `*/`
- `This example code is in the public domain.`
- `*/`
- `// Pin 13 has an LED connected on most Arduino boards.`
- `// give it a name:`
- `int led = 13;`
- `// the setup routine runs once when you press reset:`
- `void setup() {`
- `// initialize the digital pin as an output.`
- `pinMode(led, OUTPUT);`
- `}`
- `// the loop routine runs over and over again forever:`
- `void loop() {`
- `digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)`
- `delay(1000); // wait for a second`
- `digitalWrite(led, LOW); // turn the LED off by making the voltage LOW`
- `delay(1000); // wait for a second`
- `}`



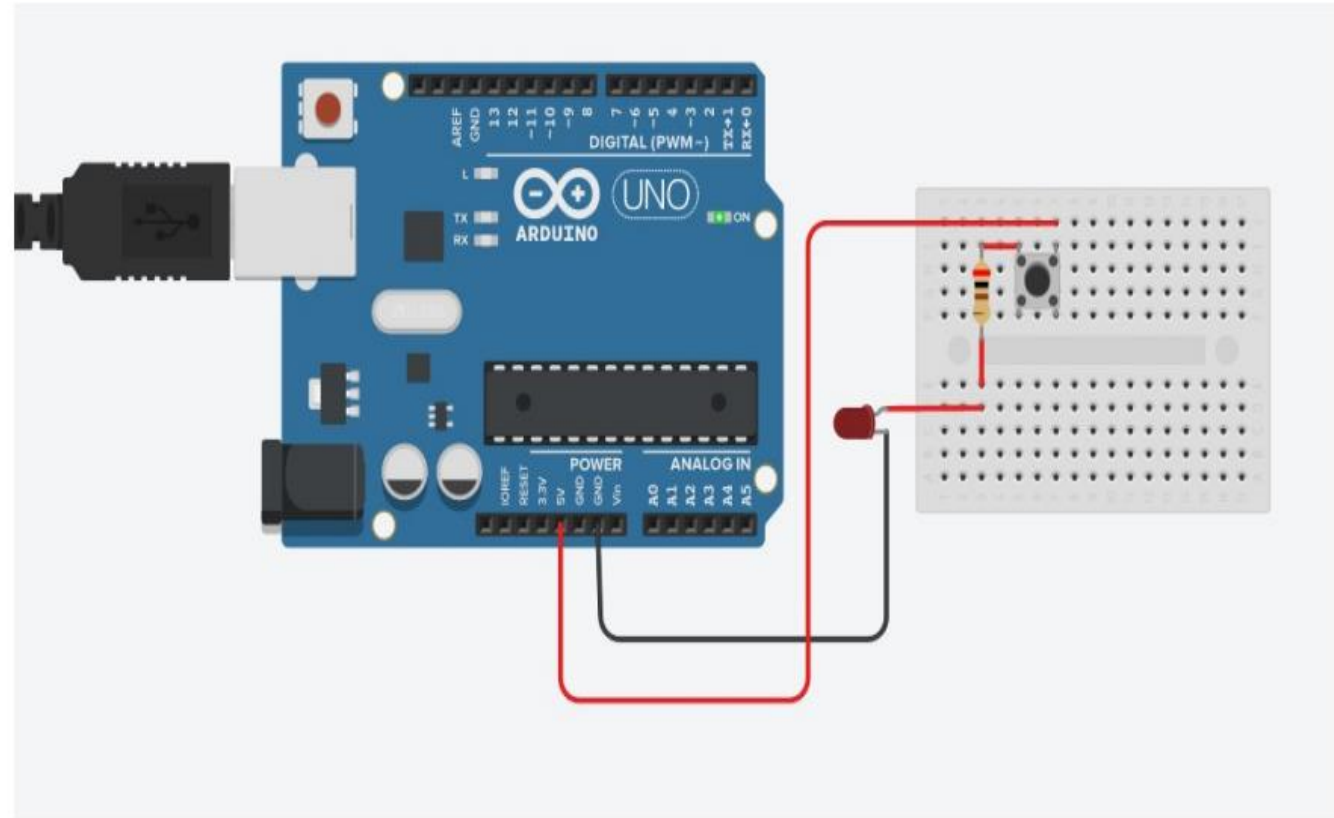
# #3 LEDs blink in series and repeat.



```
1  /*
2   Blink
3   Cycle through Array list of LEDs. For each LED in Array turns it on for 1 second.
4   Modified by Matt Royer May 5, 2014
5   */
6   const int numberOfLED = 3; // Number of LED in Array
7   const int LEDToBlink[numberOfLED] = { // Array to store LED Pins
8     13, 12, 11
9   };
10  // the setup routine runs once when you press reset:
11  void setup() {
12    // initialize the digital pin as an output
13    // For each LED in Array, initialize
14    for (int initializeLED = 0; initializeLED < numberOfLED; initializeLED++){
15      pinMode(LEDToBlink[initializeLED], OUTPUT);
16    }
17  }
18  // the loop routine runs over and over again forever:
19  void loop() {
20    for (int lightLED = 0; lightLED < numberOfLED; lightLED++){ // For each LED in Array, Blink
21      digitalWrite(LEDToBlink[lightLED], HIGH); // turn the LED on (HIGH is the voltage level)
22      delay(1000);
23      digitalWrite(LEDToBlink[lightLED], LOW); // turn the LED on (HIGH is the voltage level)
24    }
25  }
```

# #4 Push Button

A Push Button is a type of switch work on a simple mechanism called “**Push-to-make**”. When pressed it allows current to pass through it or else it remains in off state(or open state) It has four legs out of which two legs are internally connected.



# #4 Push Button

## CODE

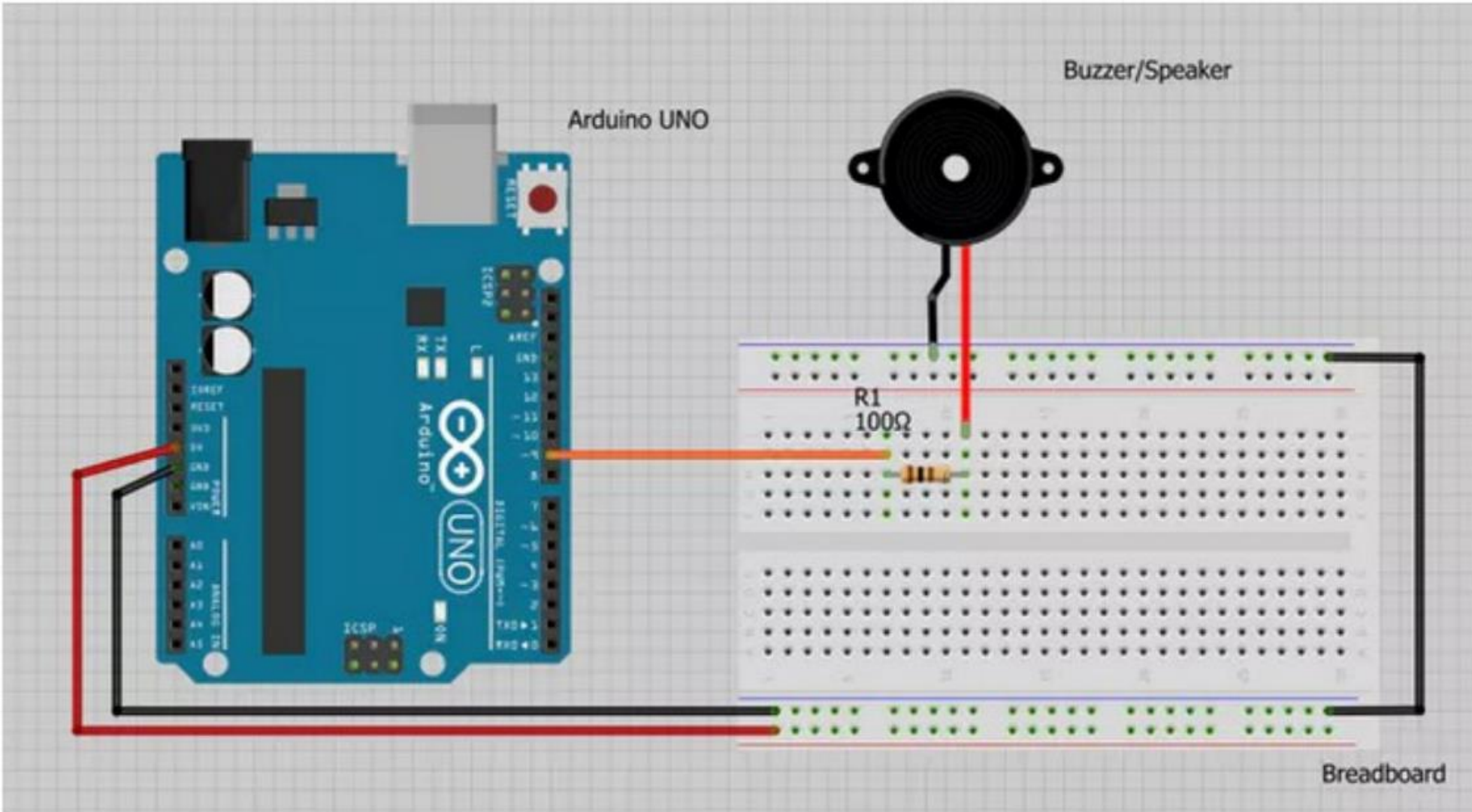
```
int ledPin = 13; // choose the pin for the LED
int inPin = 7;  // choose the input pin (for a pushbutton)
int val = 0;    // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inPin, INPUT);   // declare pushbutton as input
}

void loop(){
  val = digitalRead(inPin); // read input value
  if (val == HIGH) {        // check if the input is HIGH (button released)
    digitalWrite(ledPin, LOW); // turn LED OFF
  }
  else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```



## #5 Buzzer



# #5 Buzzer

## Code

```
const int buzzer = 9; //buzzer to arduino pin 9

void setup(){

  pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output
  id loop()
  {

  }

void loop(){

  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);      // ...for 1 sec
  noTone(buzzer);   // Stop sound...
  delay(1000);      // ...for 1sec

}
```



# #6 Servo Motor

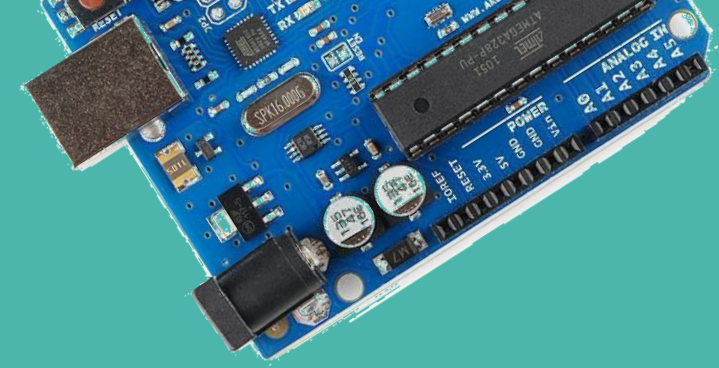
A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism



## Code

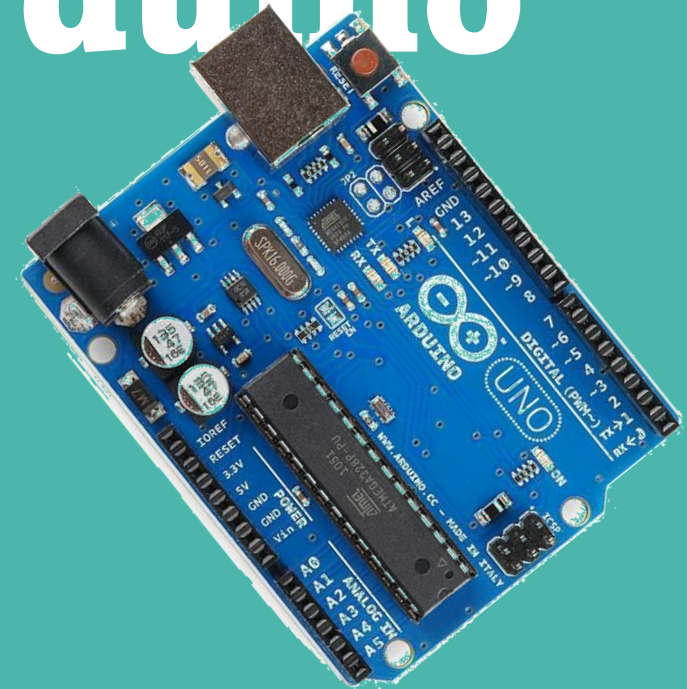
```
#include <Servo.h>
Servo servo;
void setup() {
  // put your setup code here, to run once:
  servo.attach(8);
  servo.write(0);
  delay(2000);
}

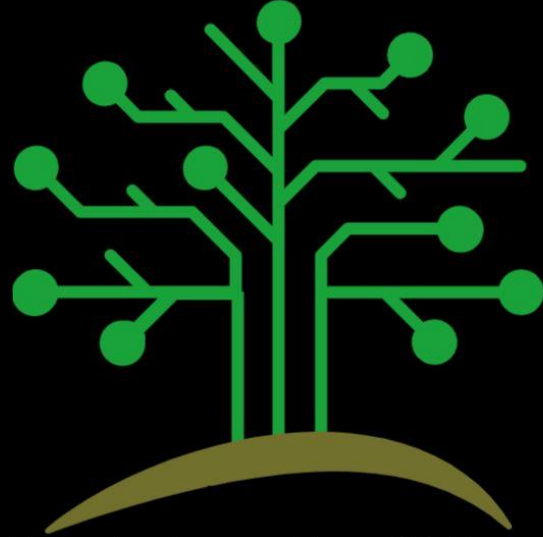
void loop() {
  // put your main code here, to run repeatedly:
  servo.write(90);
  delay(1000);
  servo.write(0);
  delay(1000);
}
```



# Thanks

## Next: Advanced Arduino





# AgTech Club

Exploring the Intersection between Agriculture and Modern Technologies