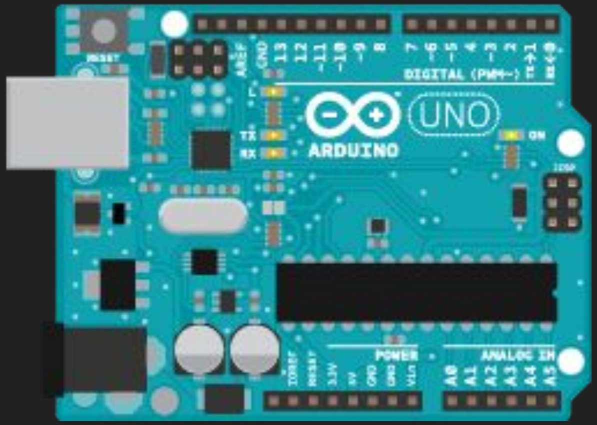
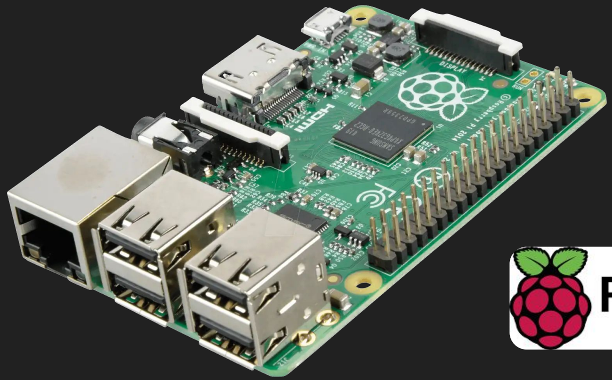


ARDUINO



Raspberry Pi®

Introduction to IoT

Basic Arduino & RPi

Arduino vs Raspberry Pi?

Arduino Uno

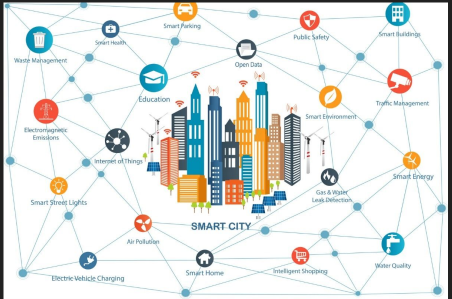
- Microcontroller board
- No OS, just only firmware
- Single Program (in a loop)
- Good at interfacing with sensors, motors, etc.
- No built-in networking

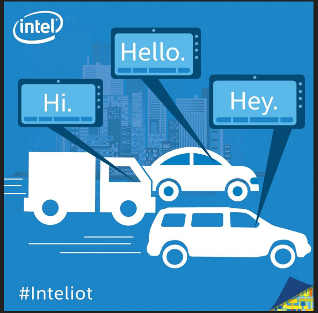
Raspberry Pi Model 3 B+

- Microcontroller board
- Runs an OS
- Can run multiple programs
- Good at doing more complex tasks,
- 802.11 Wifi, Bluetooth 4.2

What is IoT?

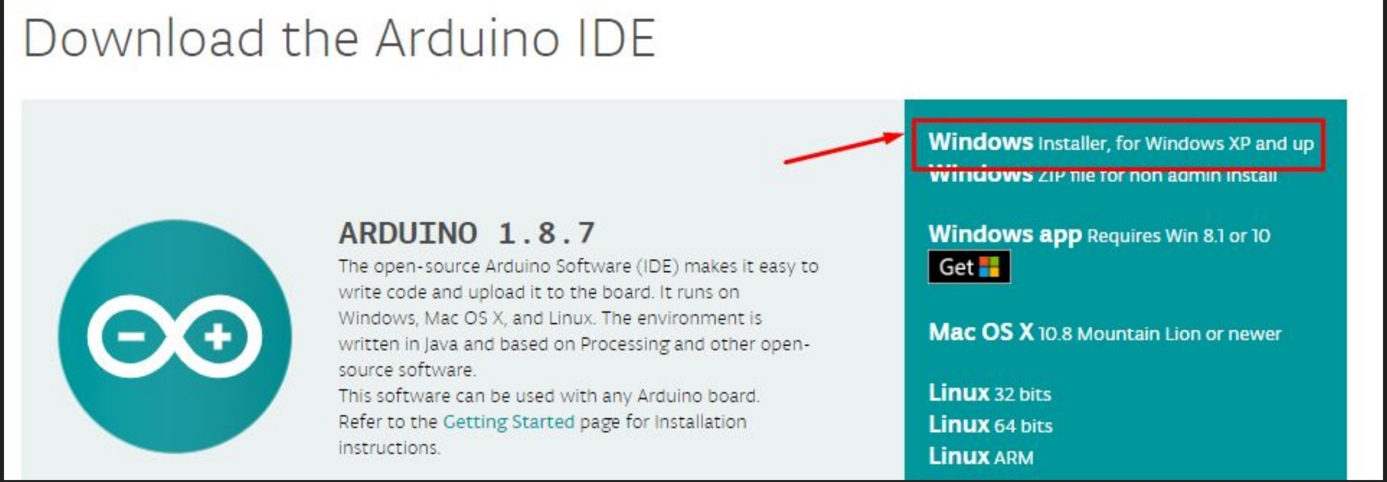
Image Courtesy of Samsung





Download Arduino IDE

https://www.arduino.cc/en/Main/Software



Download the Arduino IDE

ARDUINO 1.8.7

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, for Windows XP and up

Windows ZIP file for non admin install

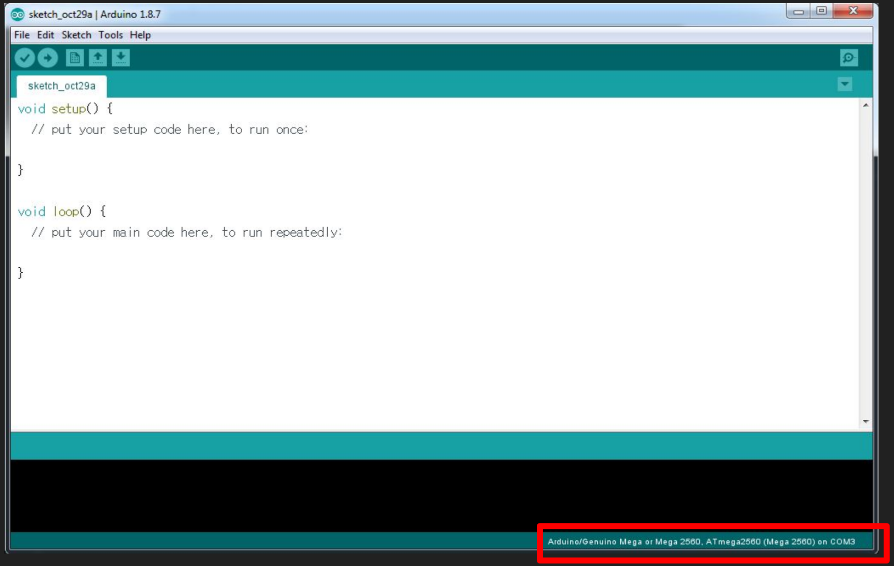
Windows app Requires Win 8.1 or 10

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits

Linux 64 bits

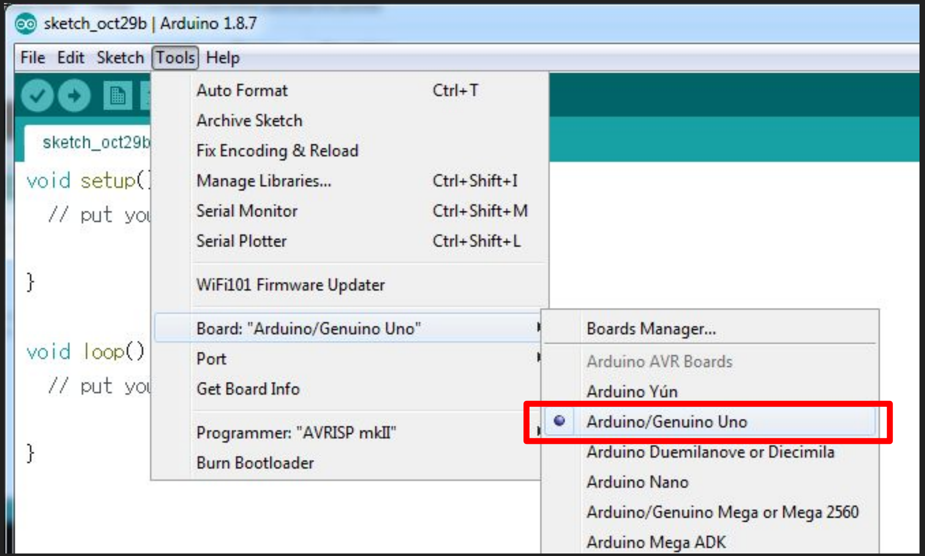
Linux ARM



sketch_oct29a | Arduino 1.8.7

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3



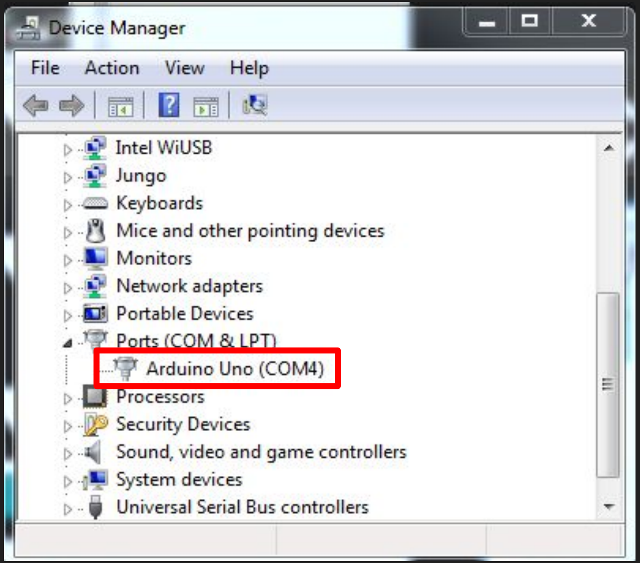
sketch_oct29b | Arduino 1.8.7

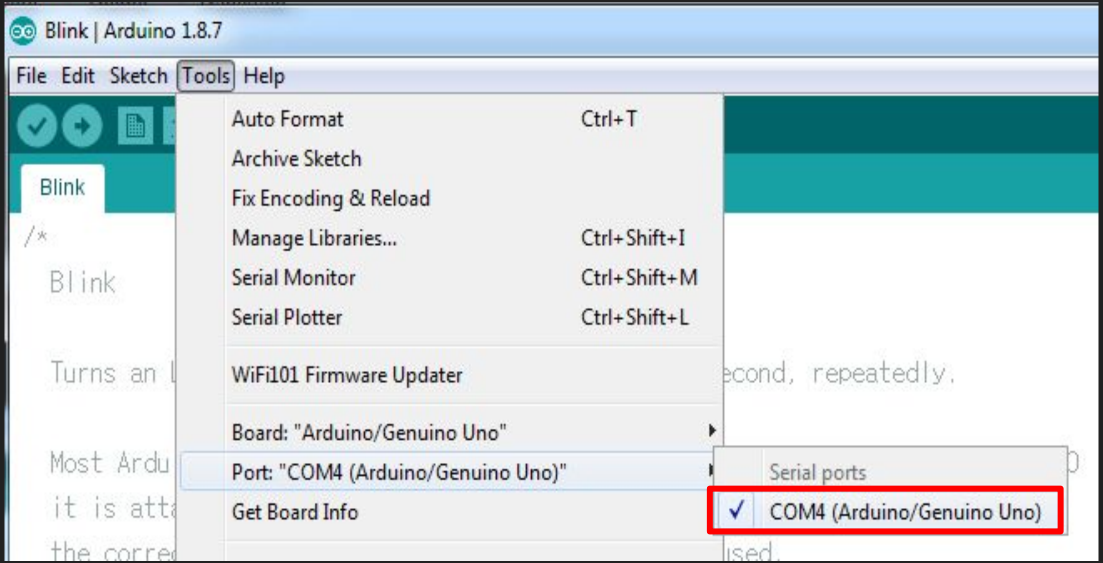
```
void setup(
  // put yo

}

void loop()
  // put yo

}
```

Board: "Arduino/Genuino Uno"

● Arduino/Genuino Uno

Programmer: "AVRISP mkII"





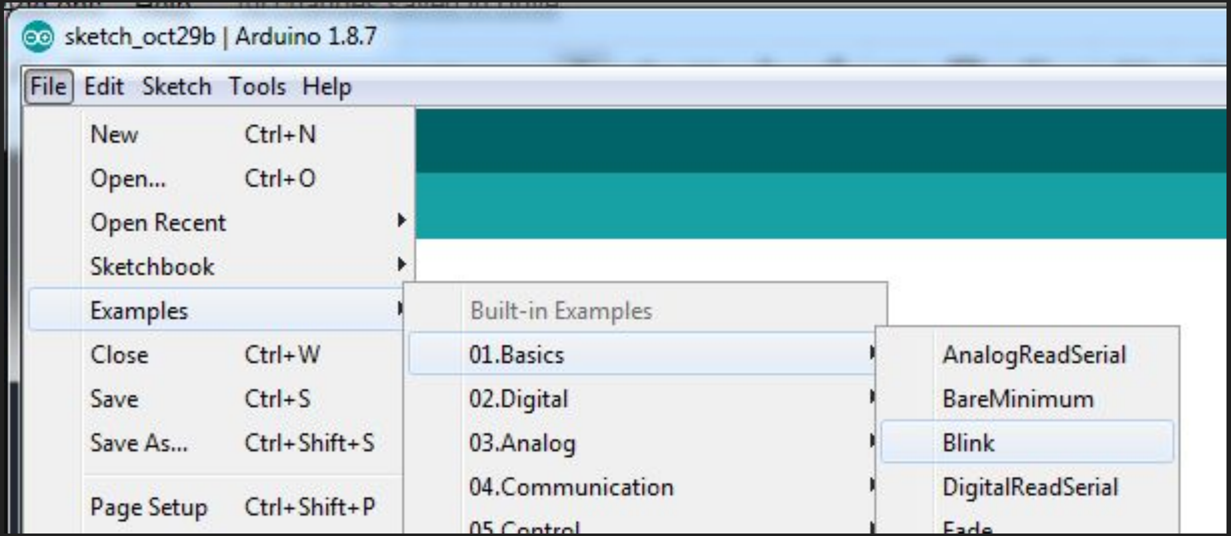
Board: "Arduino/Genuino Uno"

Port: "COM4 (Arduino/Genuino Uno)"

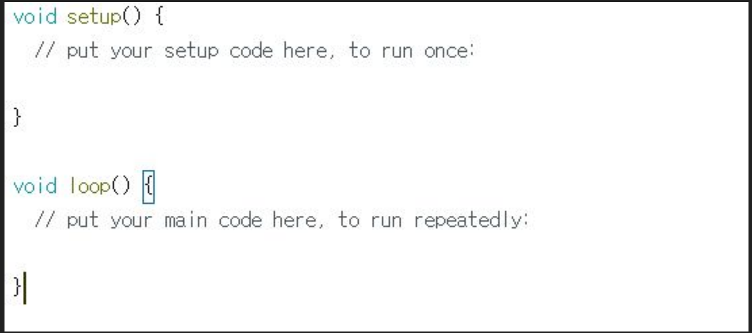
✓ COM4 (Arduino/Genuino Uno)

```
/*
 Blink

 Turns an l                                    econd, repeatedly.

 Most Ardu
 it is att
 the corre                                     used.
```

Built in Programs! - BLINK



How Arduino Code Works:



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Before **void setup()**:

- Initialize classes
- Initialize variables

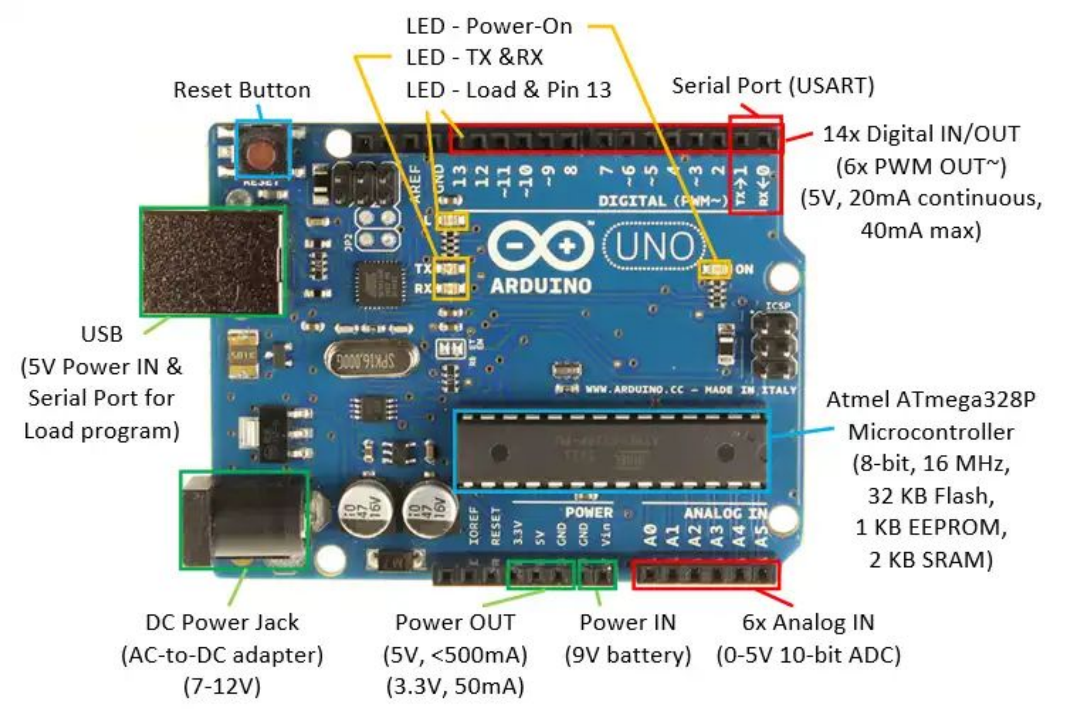
void setup() is used to set the following:

- Pin functionality using the **pinMode** function
- Initial state of pins
- Code logic

void loop() is where the rest of your code goes:

- Code runs repeatedly

Arduino Pinouts



LED - Power-On

LED - TX &RX

LED - Load & Pin 13

Reset Button

Serial Port (USART)

14x Digital IN/OUT
(6x PWM OUT~)
(5V, 20mA continuous,
40mA max)

USB
(5V Power IN &
Serial Port for
Load program)

Atmel ATmega328P
Microcontroller
(8-bit, 16 MHz,
32 KB Flash,
1 KB EEPROM,
2 KB SRAM)

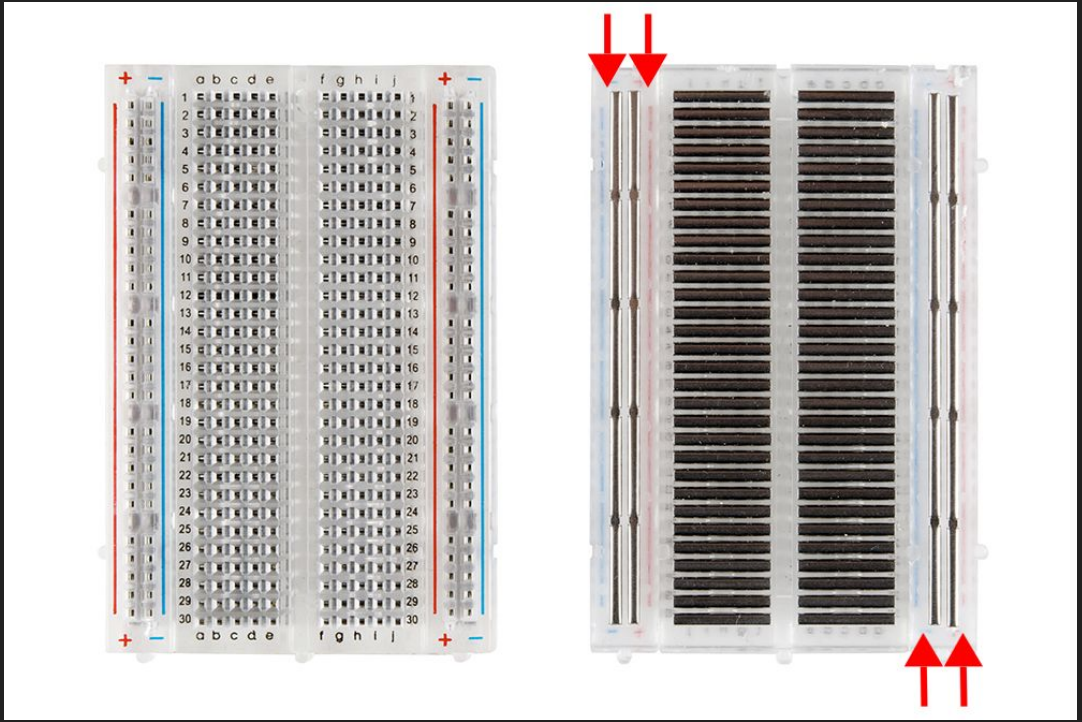
DC Power Jack
(AC-to-DC adapter)
(7-12V)

Power OUT
(5V, <500mA)
(3.3V, 50mA)

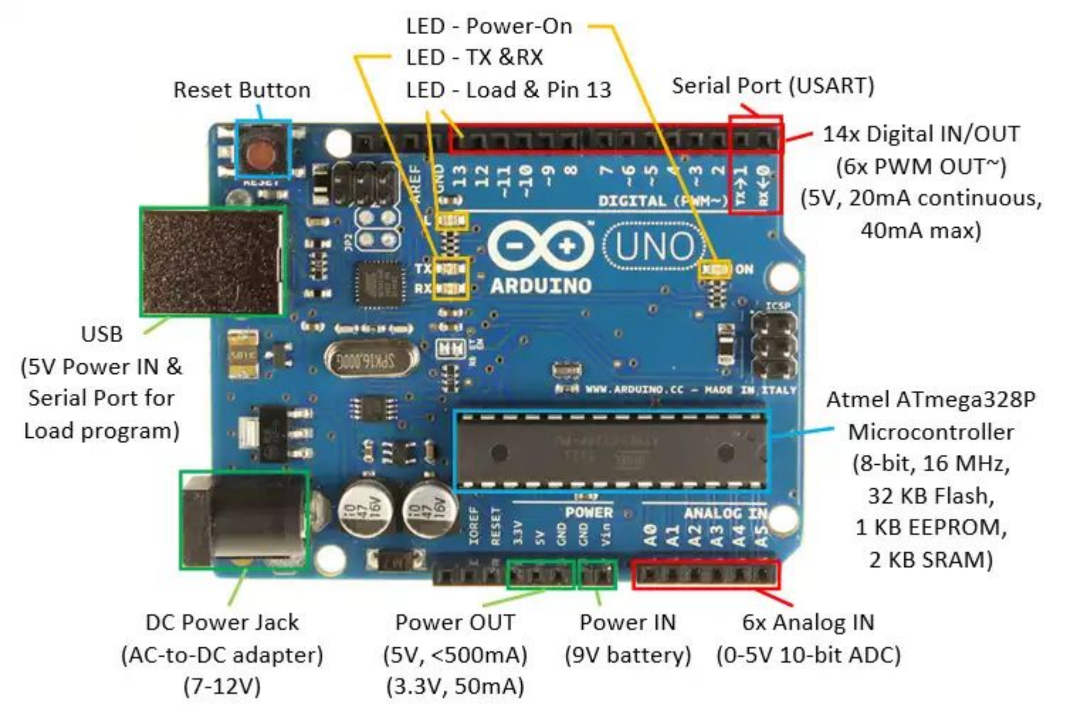
Power IN
(9V battery)

6x Analog IN
(0-5V 10-bit ADC)

Breadboards



Arduino Pinouts



LED - Power-On

LED - TX &RX

LED - Load & Pin 13

Reset Button

Serial Port (USART)

14x Digital IN/OUT
(6x PWM OUT~)
(5V, 20mA continuous,
40mA max)

USB
(5V Power IN &
Serial Port for
Load program)

Atmel ATmega328P
Microcontroller
(8-bit, 16 MHz,
32 KB Flash,
1 KB EEPROM,
2 KB SRAM)

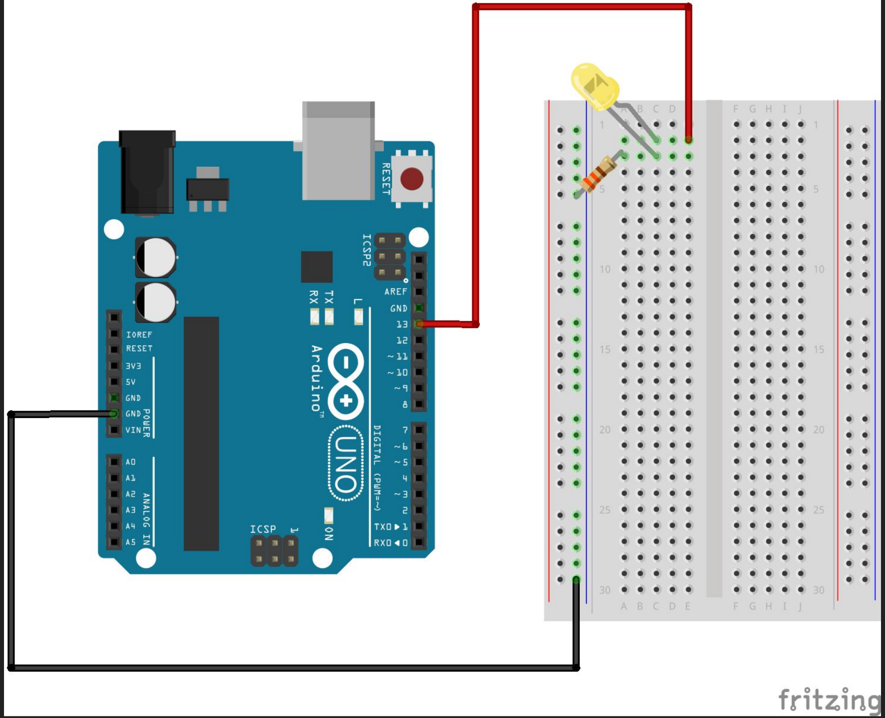
DC Power Jack
(AC-to-DC adapter)
(7-12V)

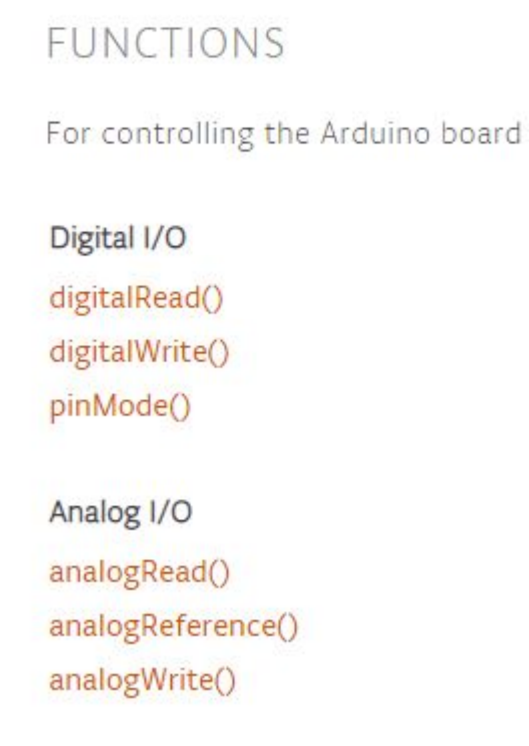
Power OUT
(5V, <500mA)
(3.3V, 50mA)

Power IN
(9V battery)

6x Analog IN
(0-5V 10-bit ADC)

Goal 1: Make external LED Blink





FUNCTIONS

For controlling the Arduino board

Digital I/O

digitalRead()

digitalWrite()

pinMode()

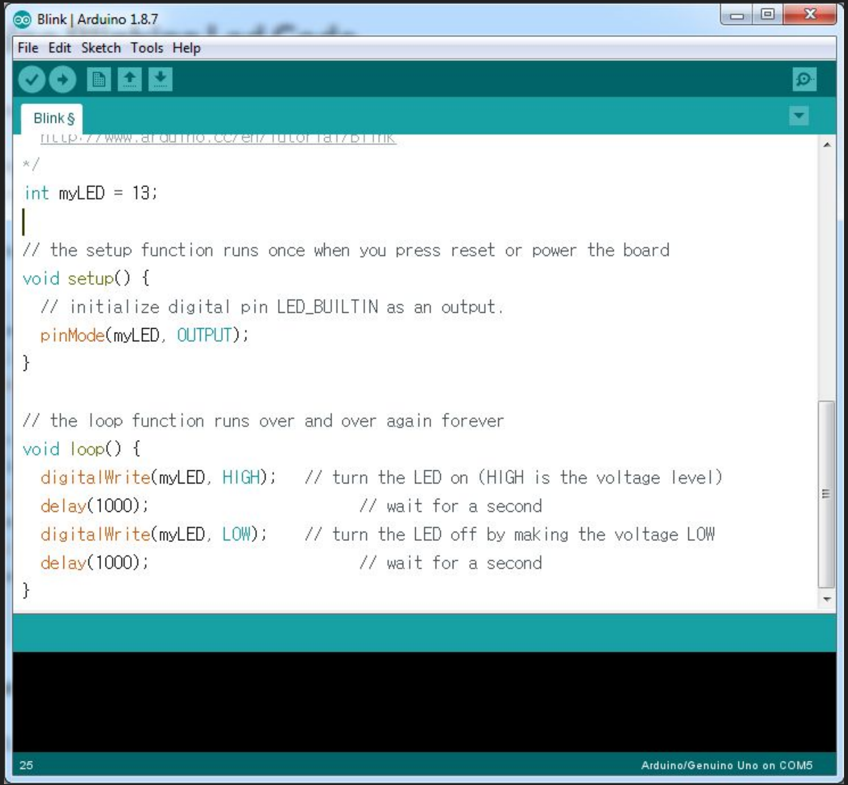
Analog I/O

analogRead()

analogReference()

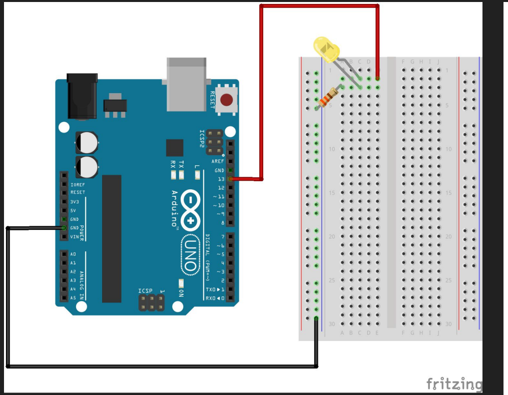
analogWrite()

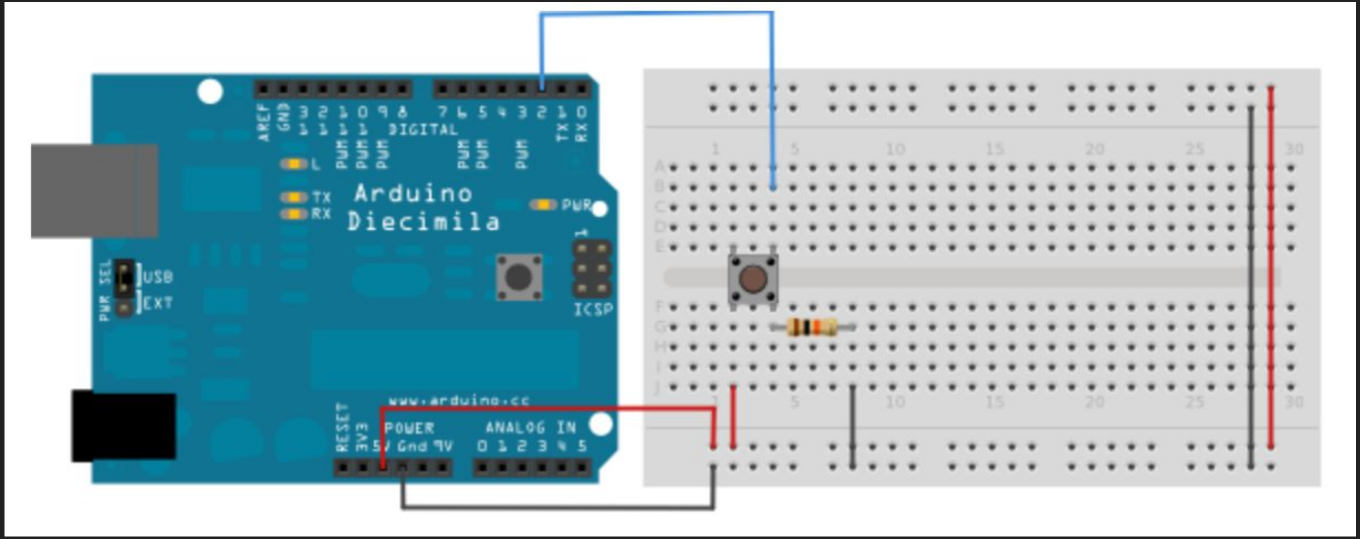
Goal 1: Make external LED Blink



```
  http://www.arduino.cc/en/Tutorial/Blink
*/
int myLED = 13;

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


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

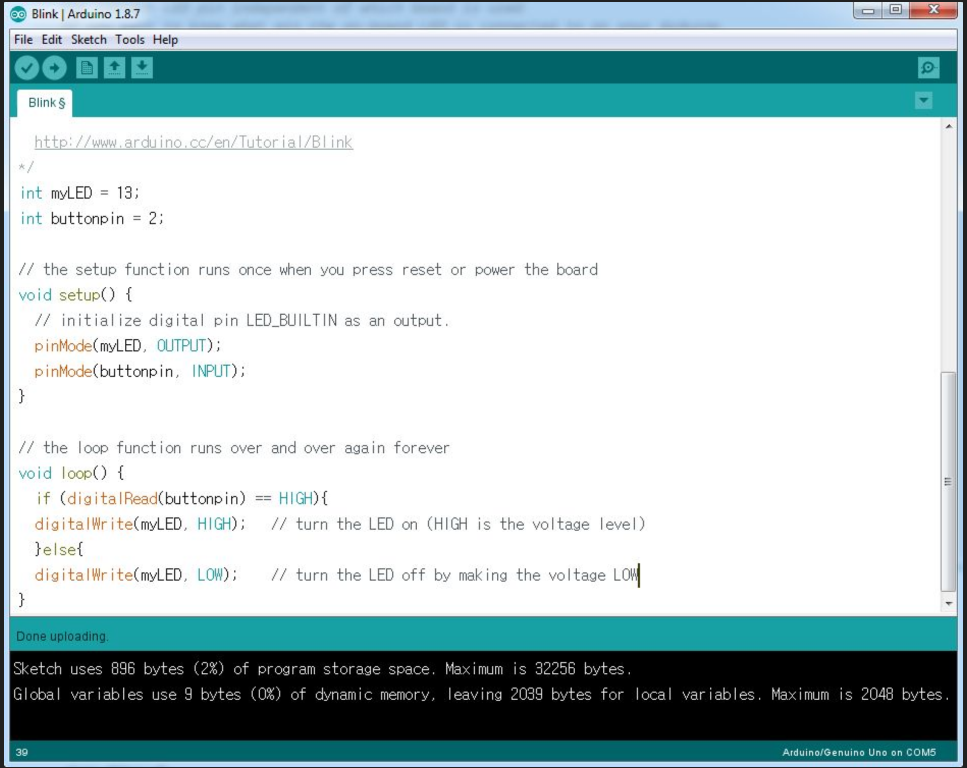
Goal 2: Add a Button to LED Blink





~10K Ohm Resistor

Goal 2: Add a Button to LED Blink



```
  http://www.arduino.cc/en/Tutorial/Blink
*/
int myLED = 13;
int buttonpin = 2;

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

// the loop function runs over and over again forever
void loop() {
  if (digitalRead(buttonpin) == HIGH){
  digitalWrite(myLED, HIGH);   // turn the LED on (HIGH is the voltage level)
  }else{
  digitalWrite(myLED, LOW);    // turn the LED off by making the voltage LOW
}
```

Done uploading.

```
Sketch uses 896 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

What is a Stepper Motor?

- DC motors that move in discrete steps
- Coils are organized in “phases”

What are the good for?

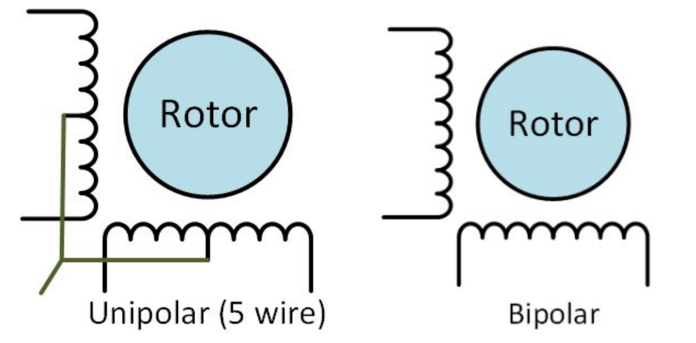
- Precision Positioning
- Speed Control
- Low Speed Torques

Cons?

- Inefficient energy use
- Limited top speed
- No Feedback (usually no need)

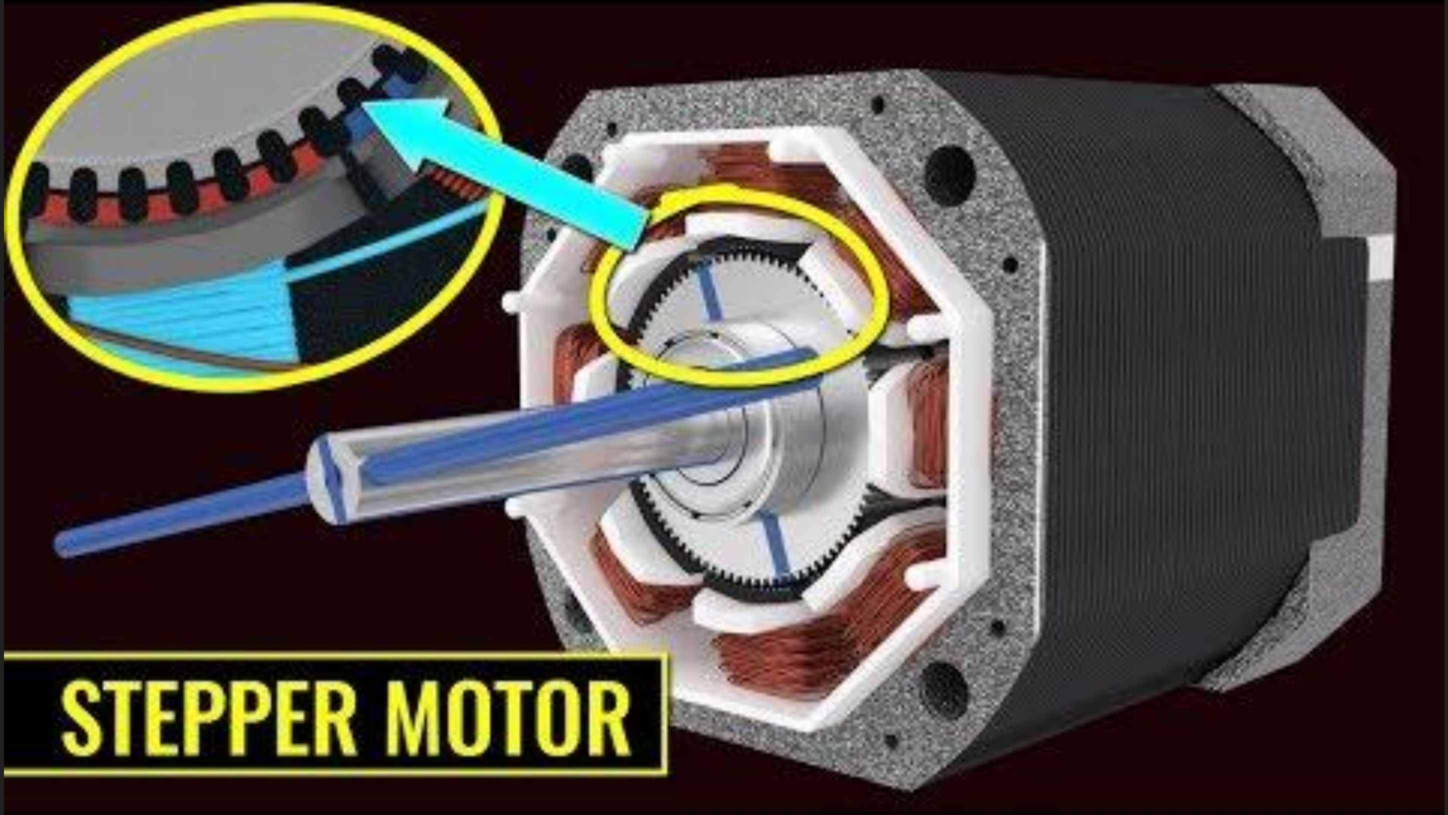
Examples:

- 3D printers, CNC, Camera platforms and X,Y Plotters



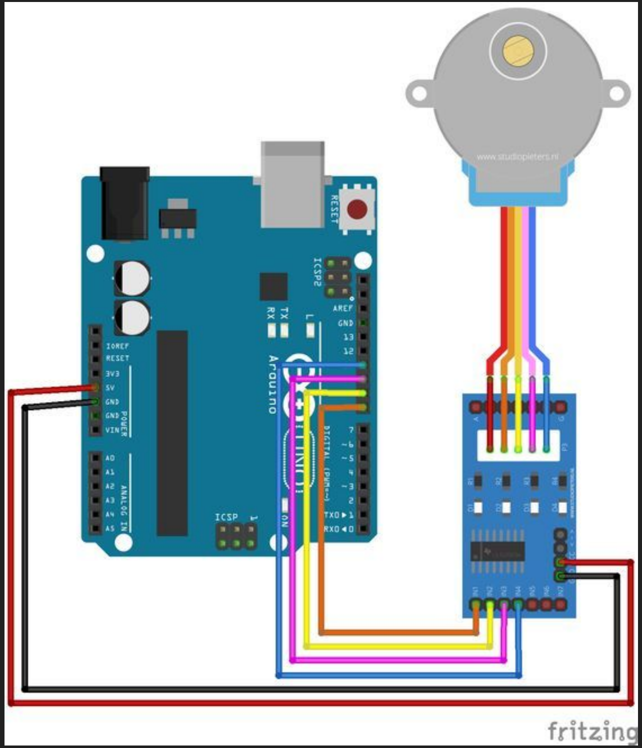
Examples:

- 3D printers, CNC, Camera platforms and X,Y Plotters

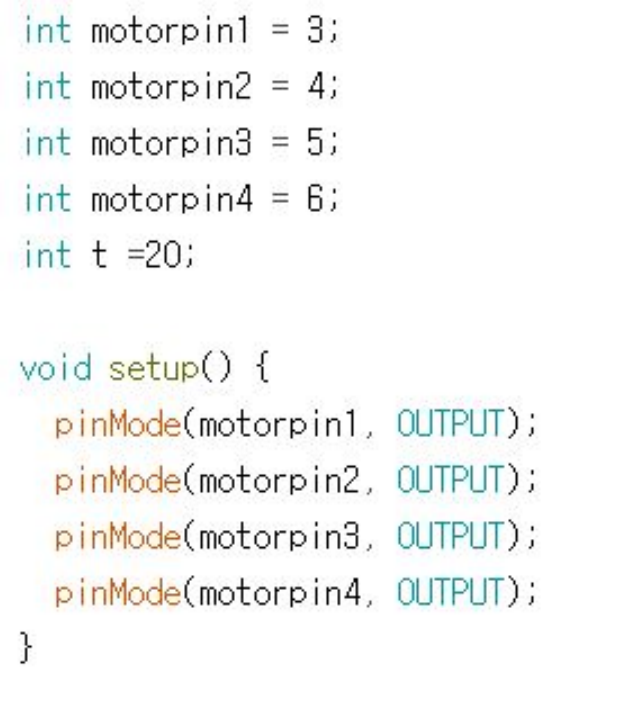


STEPPER MOTOR

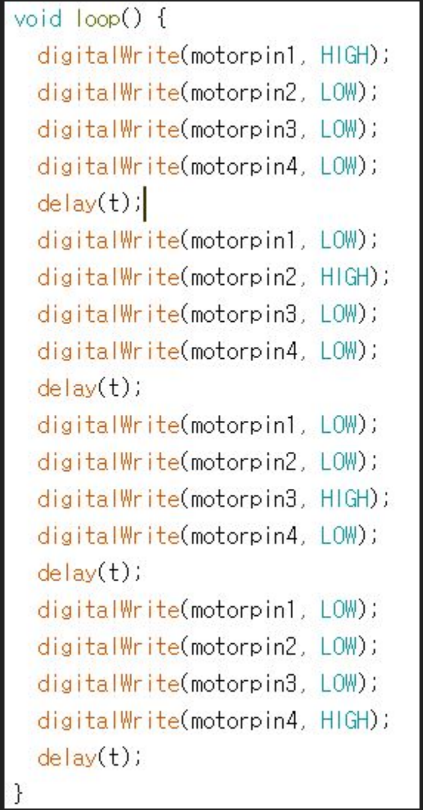
Goal 3: Stepper Motor



Goal 3: Stepper Motor

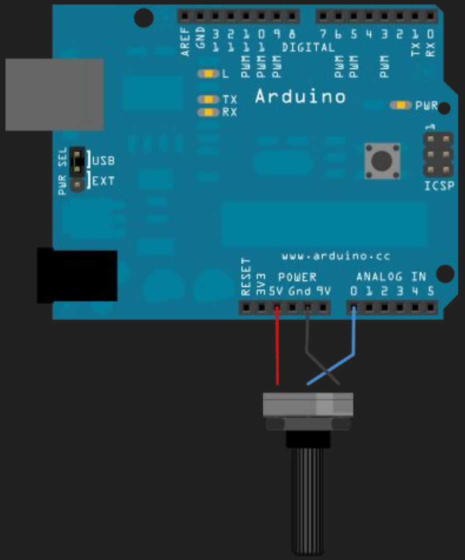


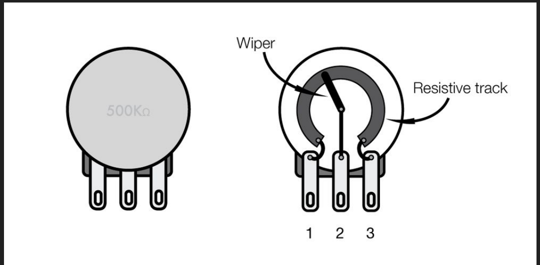
```
int motorpin1 = 3;
int motorpin2 = 4;
int motorpin3 = 5;
int motorpin4 = 6;
int t =20;

void setup() {
  pinMode(motorpin1, OUTPUT);
  pinMode(motorpin2, OUTPUT);
  pinMode(motorpin3, OUTPUT);
  pinMode(motorpin4, OUTPUT);
}
```



```
void loop() {
  digitalWrite(motorpin1, HIGH);
  digitalWrite(motorpin2, LOW);
  digitalWrite(motorpin3, LOW);
  digitalWrite(motorpin4, LOW);
  delay(t);
  digitalWrite(motorpin1, LOW);
  digitalWrite(motorpin2, HIGH);
  digitalWrite(motorpin3, LOW);
  digitalWrite(motorpin4, LOW);
  delay(t);
  digitalWrite(motorpin1, LOW);
  digitalWrite(motorpin2, LOW);
  digitalWrite(motorpin3, HIGH);
  digitalWrite(motorpin4, LOW);
  delay(t);
  digitalWrite(motorpin1, LOW);
  digitalWrite(motorpin2, LOW);
  digitalWrite(motorpin3, LOW);
  digitalWrite(motorpin4, HIGH);
  delay(t);
}
```

Potentiometer





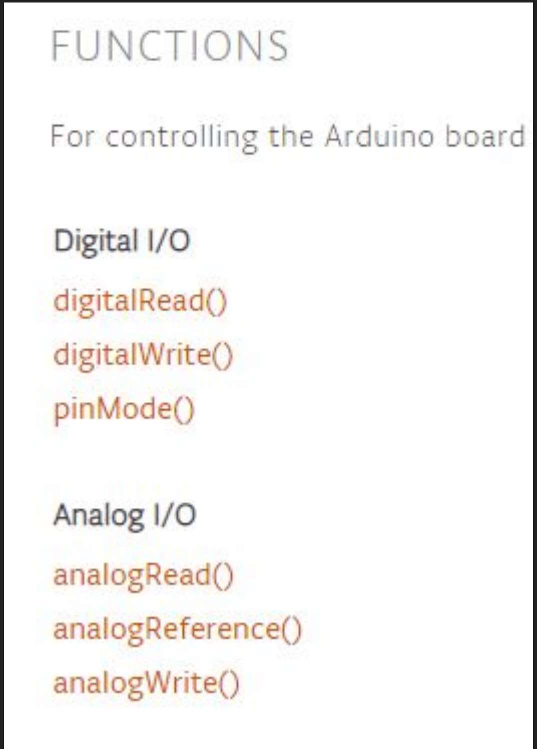
- Variable Resistor
- Apply Voltage & Ground to Pin 1 & 3
- Output is Pin 2

Final Goal: Stepper Motor w/ Button & potentiometer

Wire & Write code so that:

- When Button is pressed, Stepper motor starts turning
- Potentiometer takes variable input
- Depending on value of potentiometer, motor turns faster/slower

https://www.arduino.cc/reference/en/#functions



FUNCTIONS

For controlling the Arduino board

Digital I/O

digitalRead()

digitalWrite()

pinMode()

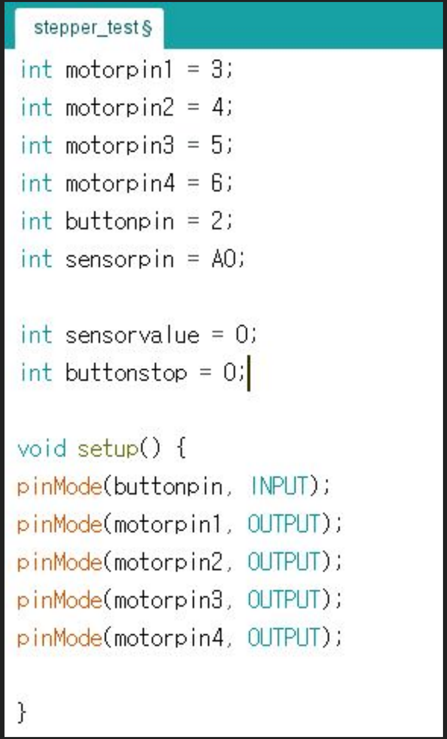
Analog I/O

analogRead()

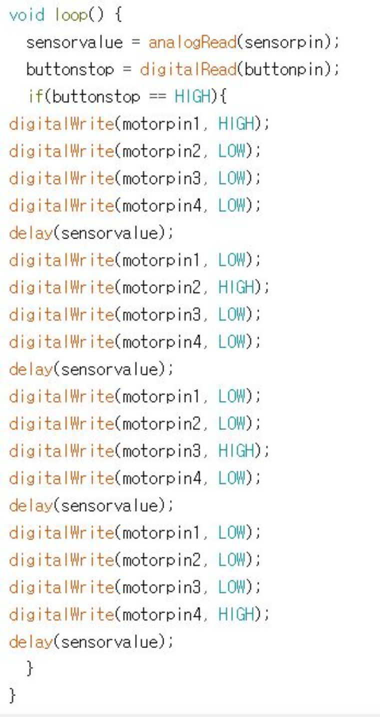
analogReference()

analogWrite()

Final Goal: Stepper Motor w/ Button & potentiometer



```
int motorpin1 = 3;
int motorpin2 = 4;
int motorpin3 = 5;
int motorpin4 = 6;
int buttonpin = 2;
int sensorpin = A0;

int sensorvalue = 0;
int buttonstop = 0;

void setup() {
pinMode(buttonpin, INPUT);
pinMode(motorpin1, OUTPUT);
pinMode(motorpin2, OUTPUT);
pinMode(motorpin3, OUTPUT);
pinMode(motorpin4, OUTPUT);

}
```



```
void loop() {
  sensorvalue = analogRead(sensorpin);
  buttonstop = digitalRead(buttonpin);
  if(buttonstop == HIGH){
digitalWrite(motorpin1, HIGH);
digitalWrite(motorpin2, LOW);
digitalWrite(motorpin3, LOW);
digitalWrite(motorpin4, LOW);
delay(sensorvalue);
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin2, HIGH);
digitalWrite(motorpin3, LOW);
digitalWrite(motorpin4, LOW);
delay(sensorvalue);
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin2, LOW);
digitalWrite(motorpin3, HIGH);
digitalWrite(motorpin4, LOW);
delay(sensorvalue);
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin2, LOW);
digitalWrite(motorpin3, LOW);
digitalWrite(motorpin4, HIGH);
delay(sensorvalue);
  }
}
```