



# ROBOT CONTROL USING PYTHON

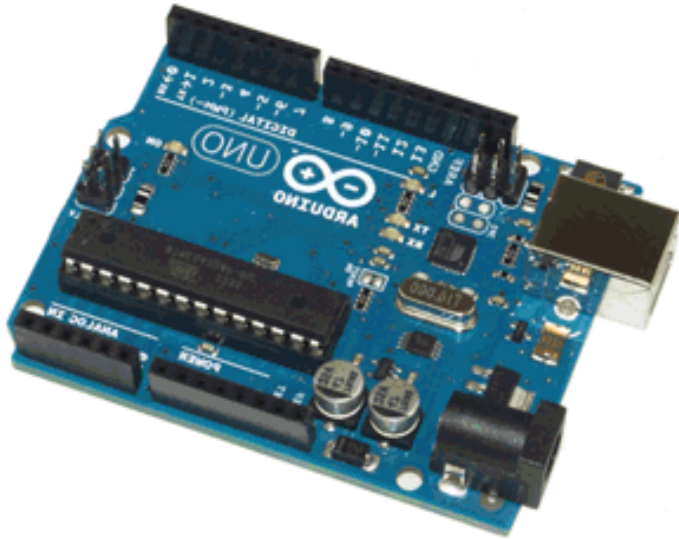
# Contents

## ❑ Arduino control with Python

## ❑ Robot Control using Python

- Motor Control
- Distance measurement with Ultrasonic sensor
- RGB LED Control
- Robot Control through Bluetooth

# Arduino control with Python



## Controlling Arduino with Raspberry Pi

**Arduino**  
**+**  
**Python**



# Nanpy

## ❑ Nanpy is a python library

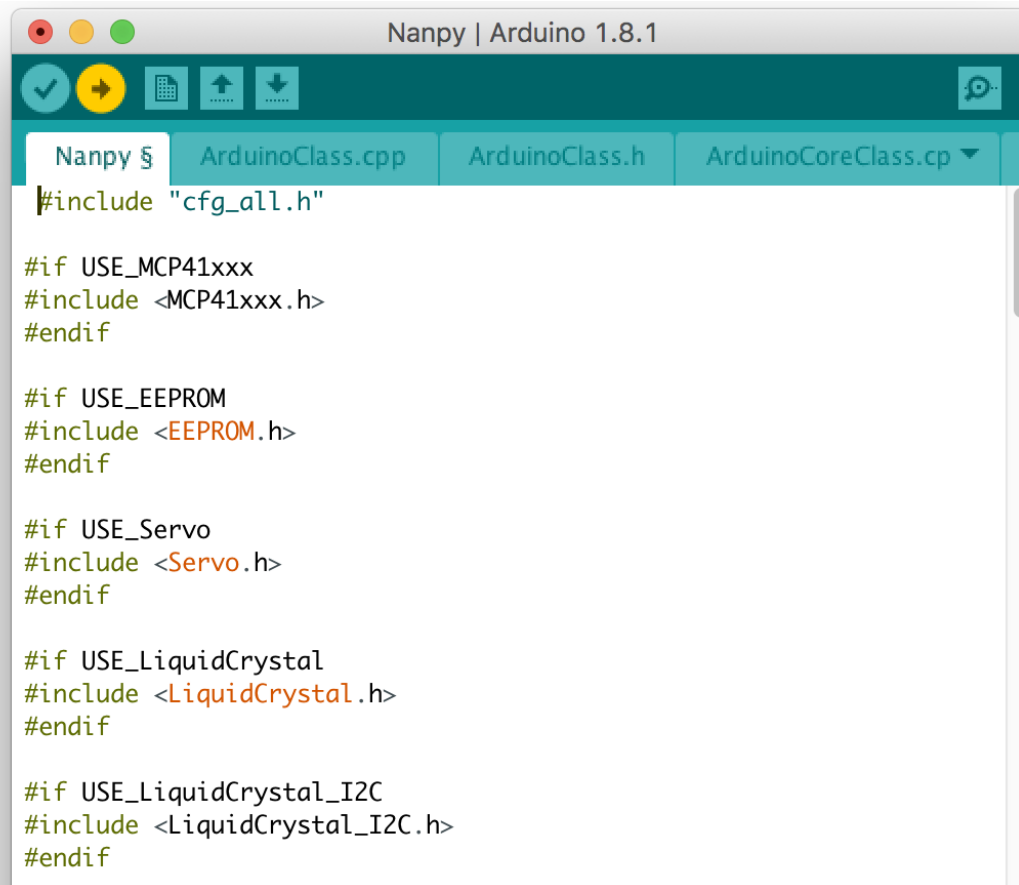
- use your Arduino as a slave, controlled by a master device where you run your scripts, such as a PC, a Raspberry Pi etc.
- control Arduino through serial connection over a USB cable.
- allow to program with python in Arduino



# Using Nanpy

## ❑ Upload Nanpy program to Arduino board

: Open Nanpy file in the Arduino IDE and upload it to Arduino board

A screenshot of the Arduino IDE interface. The title bar reads "Nanpy | Arduino 1.8.1". The top toolbar contains icons for a checkmark, a yellow arrow, a document, an upload button, and a download button. The file explorer shows four tabs: "Nanpy §", "ArduinoClass.cpp", "ArduinoClass.h", and "ArduinoCoreClass.cp". The "Nanpy §" tab is active, displaying C++ code with preprocessor directives for various hardware components.

```
#include "cfg_all.h"

#if USE_MCP41xxx
#include <MCP41xxx.h>
#endif

#if USE_EEPROM
#include <EEPROM.h>
#endif

#if USE_Servo
#include <Servo.h>
#endif

#if USE_LiquidCrystal
#include <LiquidCrystal.h>
#endif

#if USE_LiquidCrystal_I2C
#include <LiquidCrystal_I2C.h>
#endif
```



# Using Nanpy

## ❑ Write python code using Nanpy library

1. import the necessary libraries
2. create a connection to the arduino using SerialManager
3. initiate a arduino object using ArduinoApi
4. write arduino code with arduino object

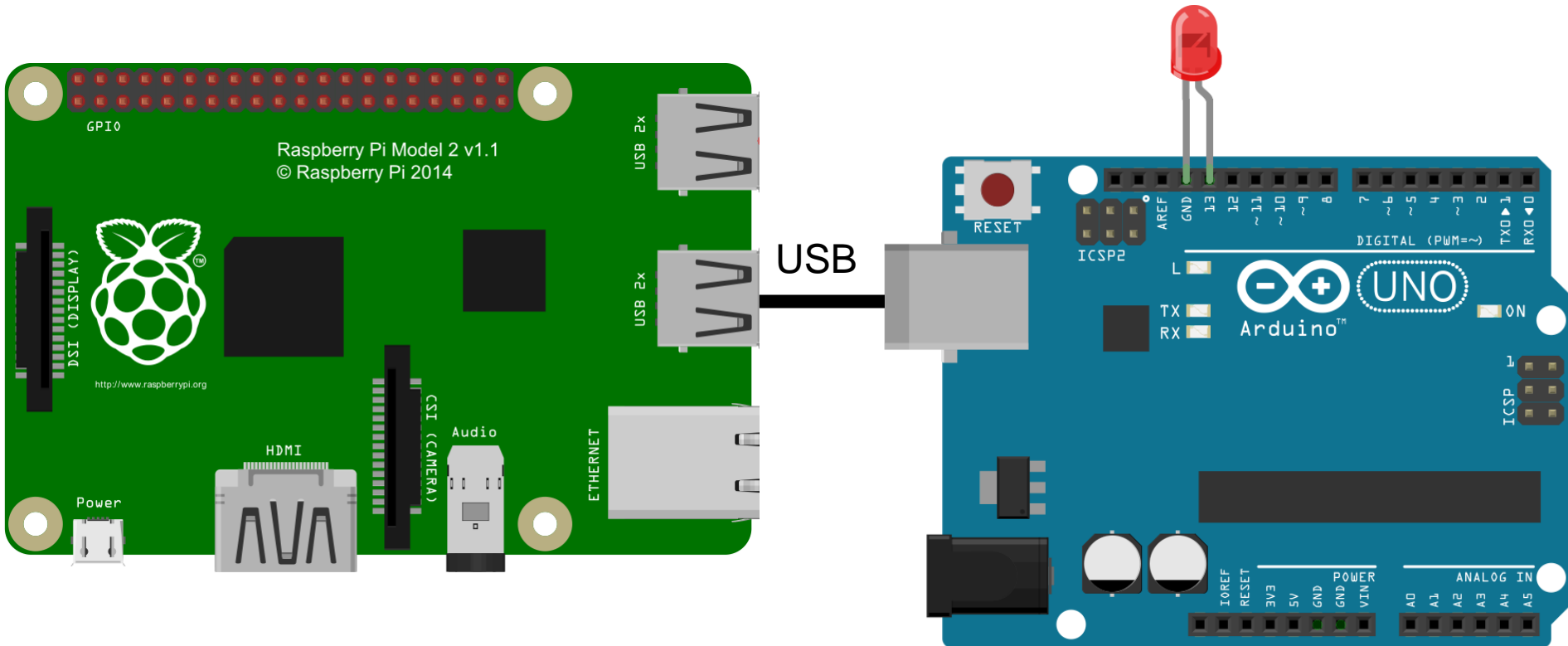
```
from nanpy import ArduinoApi
from nanpy import SerialManager

connection = SerialManager()

a = ArduinoApi(connection=connection)

a.pinMode(13, a.OUTPUT)
a.digitalWrite(13, a.HIGH)
...
...
```

# Arduino Wiring: LED



# Sample Code

```
from nanpy import ArduinoApi, SerialManager  
from time import sleep
```

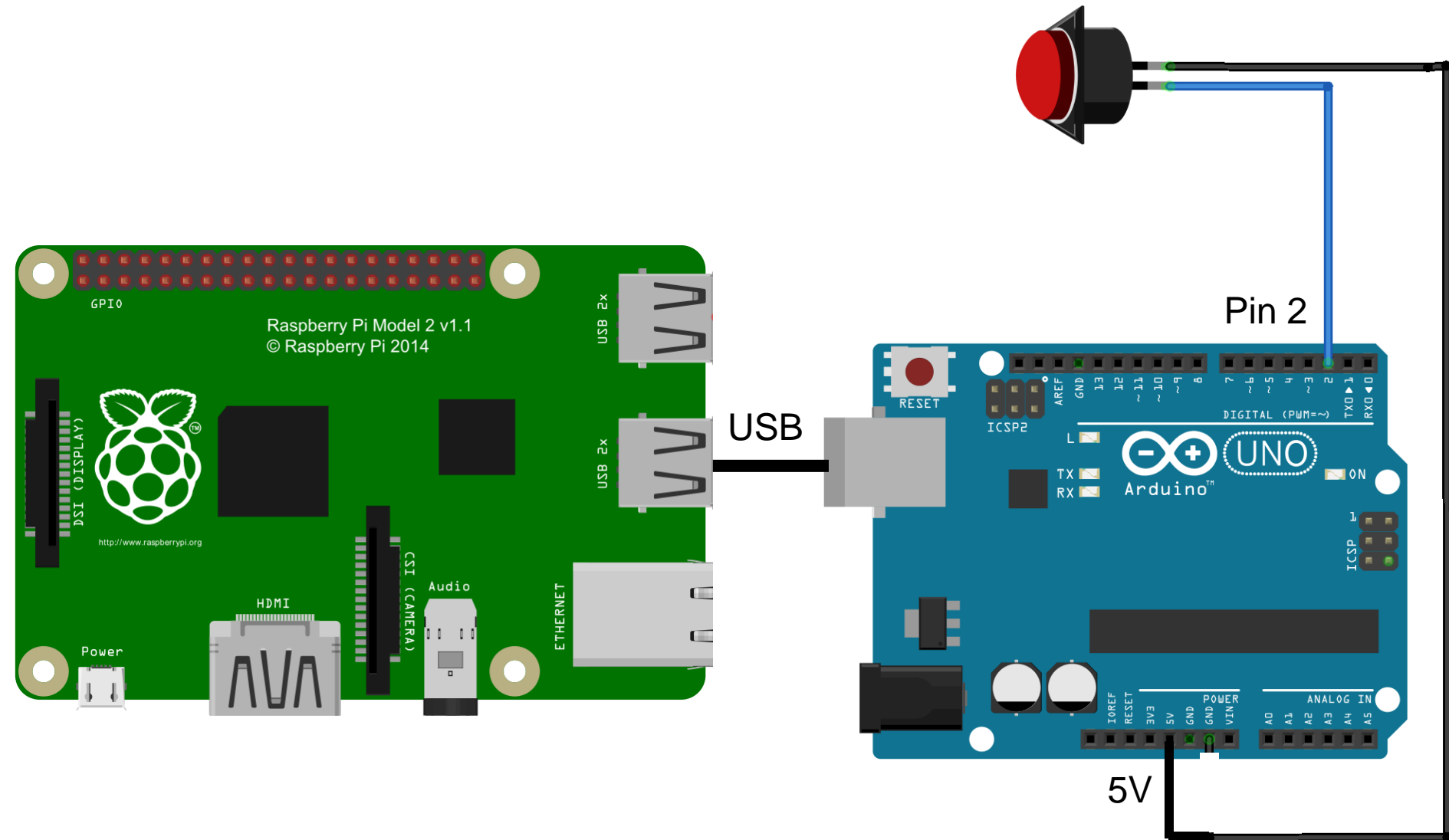
```
connection = SerialManager()  
a = ArduinoApi(connection=connection)
```

```
LED = 13  
a.pinMode(LED, a.OUTPUT)
```

```
while True:  
    a.digitalWrite(LED, a.HIGH)  
    sleep(0.5)  
    a.digitalWrite(LED, a.LOW)  
    sleep(0.5)
```



# Arduino Wiring: Push button



# Sample Code

```
from nanpy import ArduinoApi, SerialManager  
from time import sleep
```

```
connection = SerialManager()  
a = ArduinoApi(connection=connection)
```

```
BUTTON = 2  
a.pinMode(BUTTON , a.INPUT)
```

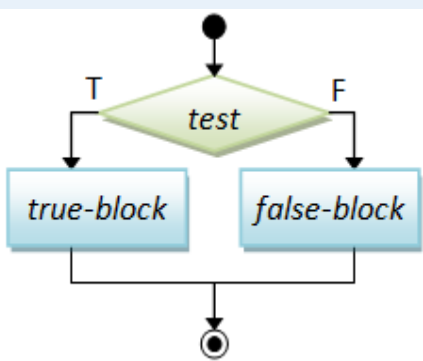
```
while True:  
    state = a.digitalRead(BUTTON)  
    print("Button state is ", state)  
    sleep(0.5)
```

# #1 задача: Button & LED

## □ Task(Задание)

1. Read digital value from a pin connected with push button
2. Turn LED on when push button is pressed
3. Turn LED off when push button is released

## □ Use Tip(Использование Совет)

Flowchart	Syntax(синтаксис)	Example(пример)
	<pre># if...else if expression :     statement_1     statement_2 else :     statement_3     statement_4</pre>	<pre>str1 = 'Start' if 's' in str1 :     print("Hello World!") else :     print("Hi, There")</pre>

# Contents

❑ Arduino control with Python

❑ Robot Control using Python

- Motor Control
- Distance measurement with Ultrasonic sensor
- RGB LED Control
- Robot Control through Bluetooth

# Robot Control using Python

- ❑ Control DC motor, Ultrasonic sensor and RGB LED with Alphabot2





# DC Motor Control with TB6612



N20 motor + rubber wheel

## Arduino ↔ TB6612

5V supply → VCC

GND → GND

### Motor Direction Control

Digital output pin → AIN1

Digital output pin → AIN2

Analog pin → PWMA

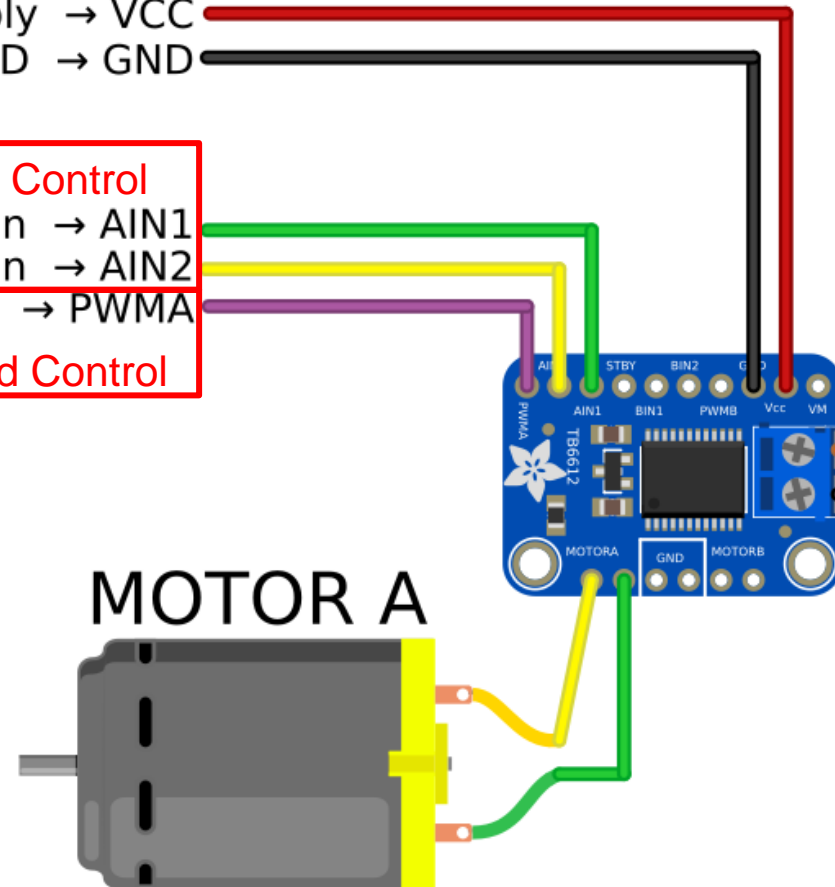
### Motor Speed Control

## ❑ Motor Speed Control

PWMA	Motor Output
0~255	0 is off, 255 is full speed

## ❑ Motor Direction Control

AIN1	AIN2	Motor Output
LOW	HIGH	Turn in Clockwise direction
HIGH	LOW	Turn in Anti-clockwise direction
LOW	LOW	Stop
HIGH	HIGH	Stop





# Sample Code

```
from nanpy import ArduinoApi, SerialManager
```

```
connection = SerialManager()
```

```
a = ArduinoApi(connection=connection)
```

```
PWMA = 6                                # Left Motor Speed pin
```

```
AIN1 = 15                               # Motor-L backward
```

```
AIN2 = 14                               # Motor-L forward
```

```
PWMB = 5                                # Right Motor Speed pin
```

```
BIN1 = 16                               # Motor-R forward
```

```
BIN2 = 17                               # Motor-R backward
```

```
SPEED = 100                             # Motor speed value
```

```
a.analogWrite(PWMA,SPEED)               # Set the speed on MotorA
```

```
a.digitalWrite(AIN1,a.LOW)              # Move MotorA forward
```

```
a.digitalWrite(AIN2,a.HIGH)             # Move MotorA forward
```

```
a.analogWrite(PWMB,SPEED)               # Set the speed on MotorB
```

```
a.digitalWrite(BIN1,a.LOW)              # Move MotorB forward
```

```
a.digitalWrite(BIN2,a.HIGH)             # Move MotorB forward
```

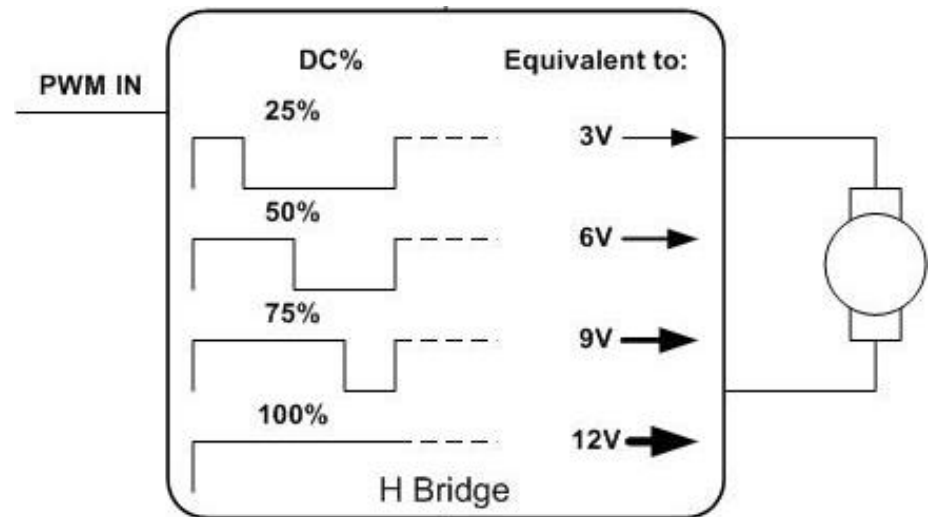
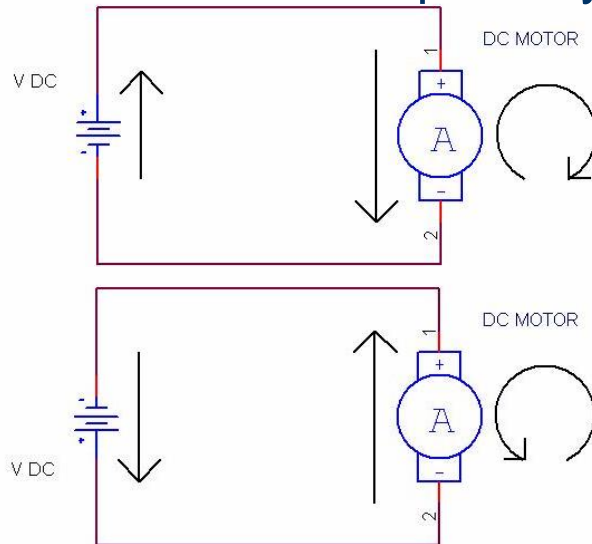
# #2 задача: Motor Control

## □ Task(Задание)

1. Write a code to move the robot backward
2. Write a code that slowly rotates the robot left or right

## □ Use Tip(Использование Совет)

- Apply DC voltage correctly to the two digital pins
- Set the motor speed by applying appropriate voltage to analog pin



# Ultrasonic Ranging



Ultrasonic

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  float duration, distance;

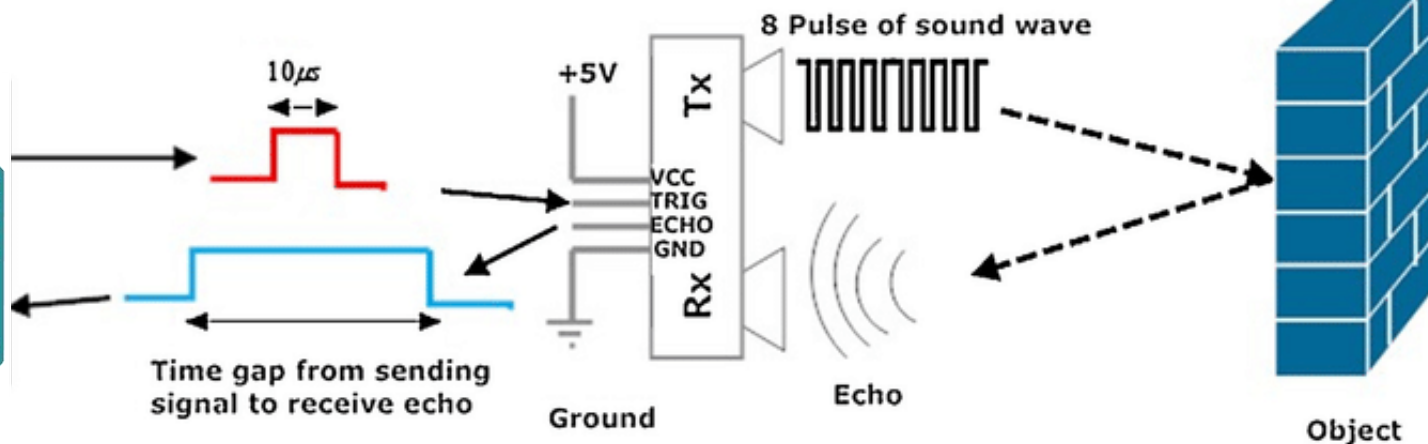
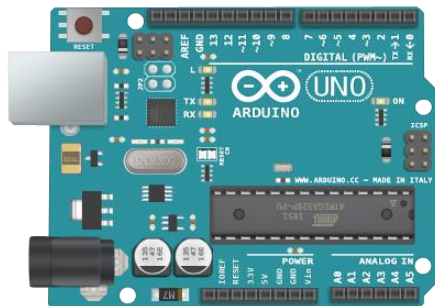
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration) / 58;
}
```

// sets the trigPin as output  
// sets the echoPin as input

// the trigPin is at 5 volts  
// 10us delay  
// the trigPin is at ground

// read the width of Echo pin  
// calculate the distance in cm



# Sample Code: Ultrasonic Ranging

```
from nanpy import ArduinoApi, SerialManager, Ultrasonic
from time import sleep

connection = SerialManager()
a = ArduinoApi(connection=connection)

ECHO = 2 # Echo pin number
TRIG = 3 # Trigger pin number
u = Ultrasonic(ECHO, TRIG, False) # an object of Ultrasonic class

while True:
    distance = u.get_distance() # a function that calculates distance
    print("The distance is ", distance)
    sleep(0.5)
```

# #3 задача: Ultrasonic Ranging

## □ Task(Задание)

: According to the range, control Motor and show corresponding messages as below table.

Distance Range	Motor	Print message
distance $\leq 5$	Stop	Obstacle in front of the robot
$5 < \text{distance} \leq 20$	Forward slowly	Attention to front obstacle
distance $\geq 20$	Forward fast	No obstacle

## □ Use Tip(Использование Совет)

```
if distance <= 5:
    ...
elif 5 < distance <= 20:
    ...
else:
    ...
```

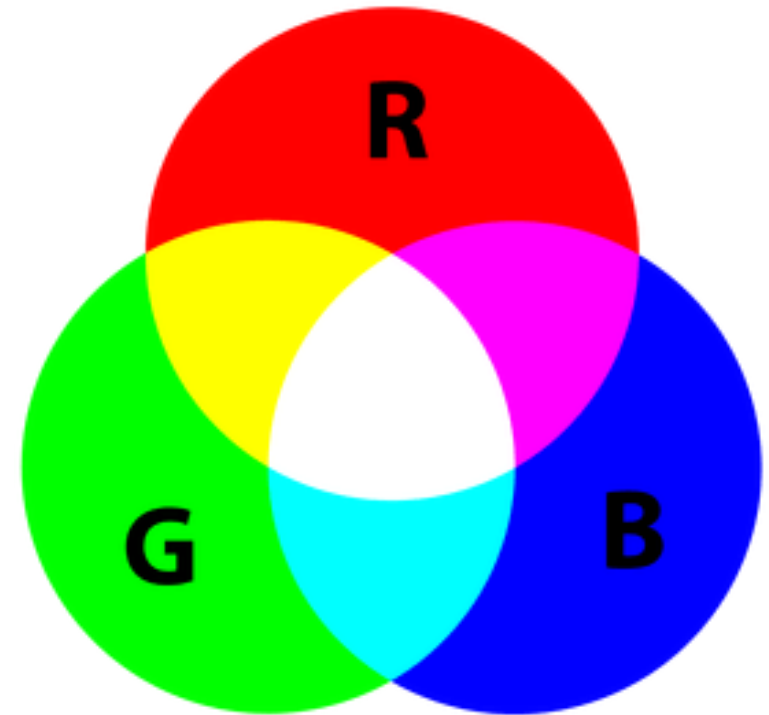
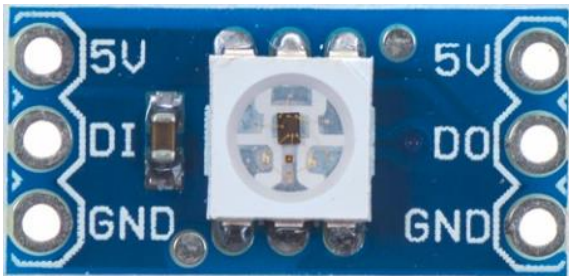


# RGB LED Control with W2812



RGB LED

- ❑ RGB or Red-Green-Blue LEDs are a fusion of three LEDs in a single package.



Display any color of the rainbow by fusing together combinations of three colors



# Sample Code: RGB LED Control

```
from nanpy import ArduinoApi, SerialManager, W2812
```

```
connection = SerialManager()  
a = ArduinoApi(connection=connection)
```

```
NUM_RGB = 4
```

```
RGB = 7
```

```
w = W2812(NUM_RGB, RGB)
```

```
# Numbers of RGB
```

```
# RGB LED pin
```

```
# an object of W2812 class
```

```
w.setColorRGB(0, 255, 0, 0)
```

```
w.setColorRGB(1, 0, 255, 0)
```

```
w.setColorRGB(2, 0, 0, 255)
```

```
w.setColorRGB(3, 255, 255, 0)
```

```
# Red LED on 1st LED
```

```
# Green LED on 2nd LED
```

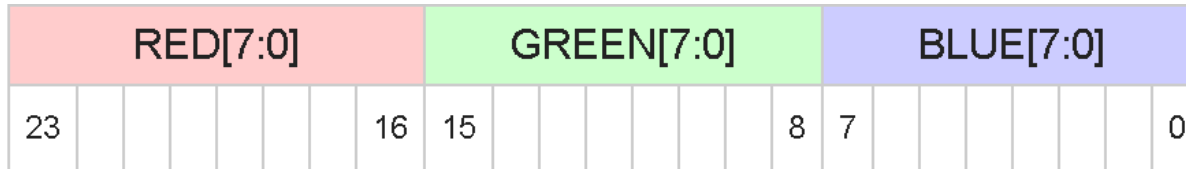
```
# Blue LED on 3rd LED
```

```
# Yellow LED on 4th LED
```

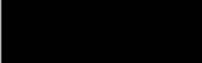

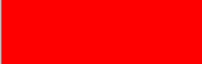





# RGB Color Codes

## □ RGB color format

: RGB code has 24 bits format (bits 0..23)



## □ RGB color table

Color	HTML / CSS Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
	Black	#000000	(0,0,0)
	White	#FFFFFF	(255,255,255)
	Red	#FF0000	(255,0,0)
	Lime	#00FF00	(0,255,0)
	Blue	#0000FF	(0,0,255)
	Yellow	#FFFF00	(255,255,0)
	Cyan / Aqua	#00FFFF	(0,255,255)
	Magenta / Fuchsia	#FF00FF	(255,0,255)

# #4 задача: RGB LED Control

## ❑ Task(Задание)

: Turn all 4 LEDs on in accordance with user input value

User input	LED
Red	RED LED on
Green	Green LED on
Blue	Blue LED on
...	
...	

## ❑ Use Tip(Использование Совет)

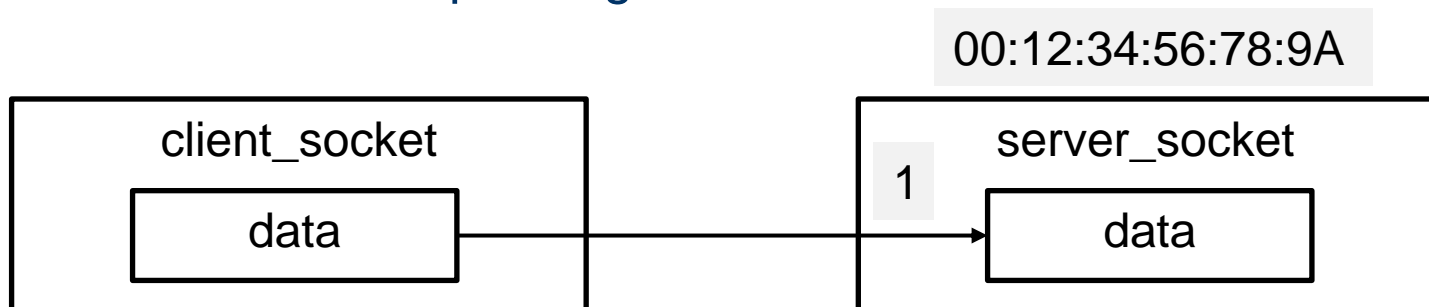
# Bluetooth programming with Python

## ❑ PyBluez

- is Python Bluetooth library written in C for the Windows and GNU/Linux operating systems
- allows Python code to access the host machine's Bluetooth resources

## ❑ Bluetooth programming in Python follows the socket programming model called RFCOMM

- RFCOMM socket is a connection-oriented protocol, similar to TCP
- one process acts as a server accepting connections, another process acts as the client requesting the connection



# Sample Code: Bluetooth (Client)

```
import bluetooth                                # import python bluetooth library

ADDR = "xx:xx:xx:xx:xx:xx"                    # address of bluetooth device
PORT = 1                                       # port number to connect

# Create a socket for RFCOMM service
socket = bluetooth.BluetoothSocket(bluetooth.RFCOMM)

socket.connect((ADDR, PORT))                  # a connection with xx:xx:xx:xx:xx:xx on port 1

socket.send("Hello World")                    # Send data through the socket

socket.close()                                # Close the connection on socket
```

[ Python Code in Raspberry Pi ]

# Example Code: Bluetooth (Server)

```
char data = 0;                                // for incoming serial data

void setup() {
    Serial.begin(115200);                      // opens serial port, sets data rate to 115200 bps
}

void loop() {
    if (Serial.available() > 0) {             // reply only when you receive data:
        data = Serial.read();                 // read the incoming byte:
        Serial.print(data);                   // say what you got:
    }
}
```

[ C++ Code in Arduino ]



# #5 задача: Bluetooth Control

## □ Task(Задание)

: According to Bluetooth data, control components of robot

Bluetooth data	Motor	Bluetooth data	LED
F	Move Forward	r	Red
B	Move Backwards	g	Green
L	Turn Left	b	Blue
R	Turn Right	y	Yellow
S	Stop	w	White

## □ Use Tip(Использование Совет)

- Write Arduino code to react with Bluetooth data

```
#include <Adafruit_NeoPixel.h>
Adafruit_NeoPixel RGB = Adafruit_NeoPixel(4, 7, NEO_GRB + NEO_KHZ800)
RGB.setPixelColor(0, 0x00FF00) // # Green LED on 1st LED
```

- Write Python code to send Bluetooth data continuously