



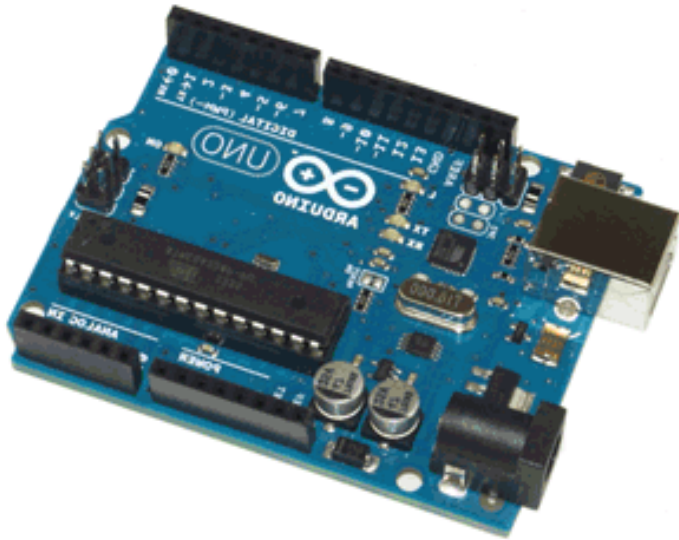
УПРАВЛЕНИЕ РОБОТОМ С ПОМОЩЬЮ PYTHON

Содержание

- ❑ Arduino управления с Python

- ❑ Управление роботом с помощью Python
 - Управление Мотором
 - Измерение расстояния с помощью ультразвукового датчика
 - Управление RGB LED-светодиодом
 - Управление роботом через Bluetooth

Arduino управления с питоном



Controlling Arduino with Raspberry Pi

Arduino
+
Python



Nanpy

□ Nanpy - это библиотека python

- используйте Arduino в качестве подчиненного устройства, управляемое ведущим устройством, где вы запускаете свои скрипты, такие как ПК, Raspberry Pi и т. д.
- управляйте Arduino через последовательное соединение через USB-кабель.
- разрешить программирование с помощью python в Arduino



+



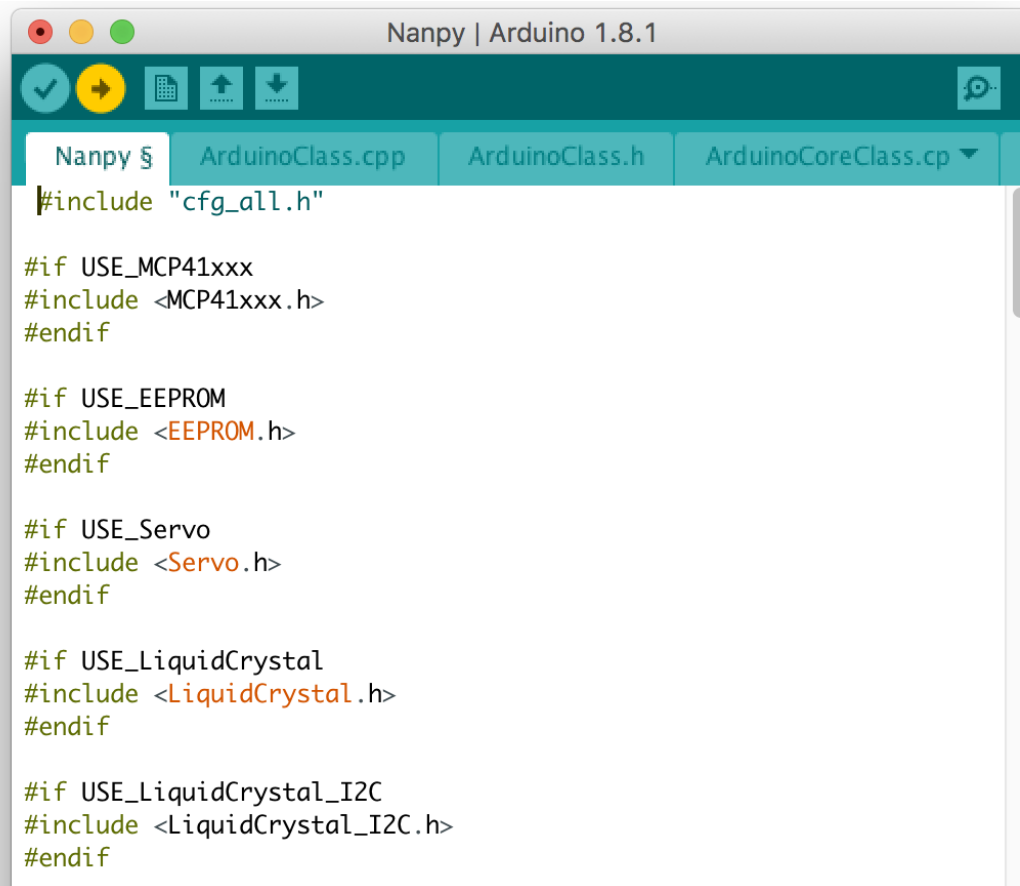
=



Использование Nanpy

❑ Загрузите программу Nanpy на плату Arduino

: Откройте файл Nanpy в среде Arduino и загрузите его на плату Arduino



```
Nanpy | Arduino 1.8.1
Nanpy § ArduinoClass.cpp ArduinoClass.h ArduinoCoreClass.cp
#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
```


Использование Nanpy

❑ Записать код Python с помощью библиотеки Nanpy

1. импортировать необходимые библиотеки
2. создать соединение с arduino с помощью SerialManager
3. инициировать объект arduino, используя ArduinoApi
4. написать код arduino с объектом arduino

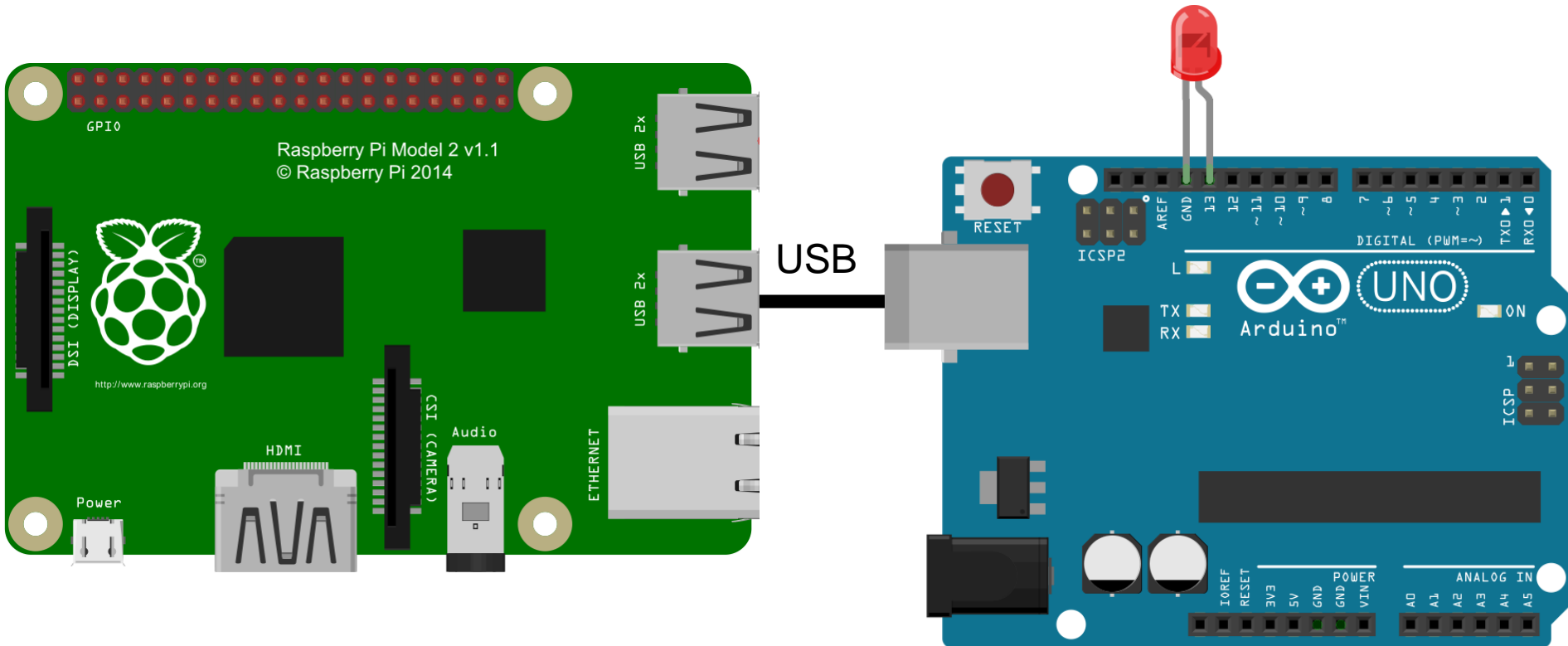
```
from nanpy import ArduinoApi
from nanpy import SerialManager

connection = SerialManager()

a = ArduinoApi(connection=connection)

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

Arduino схема подключения: светодиода



Примерный код

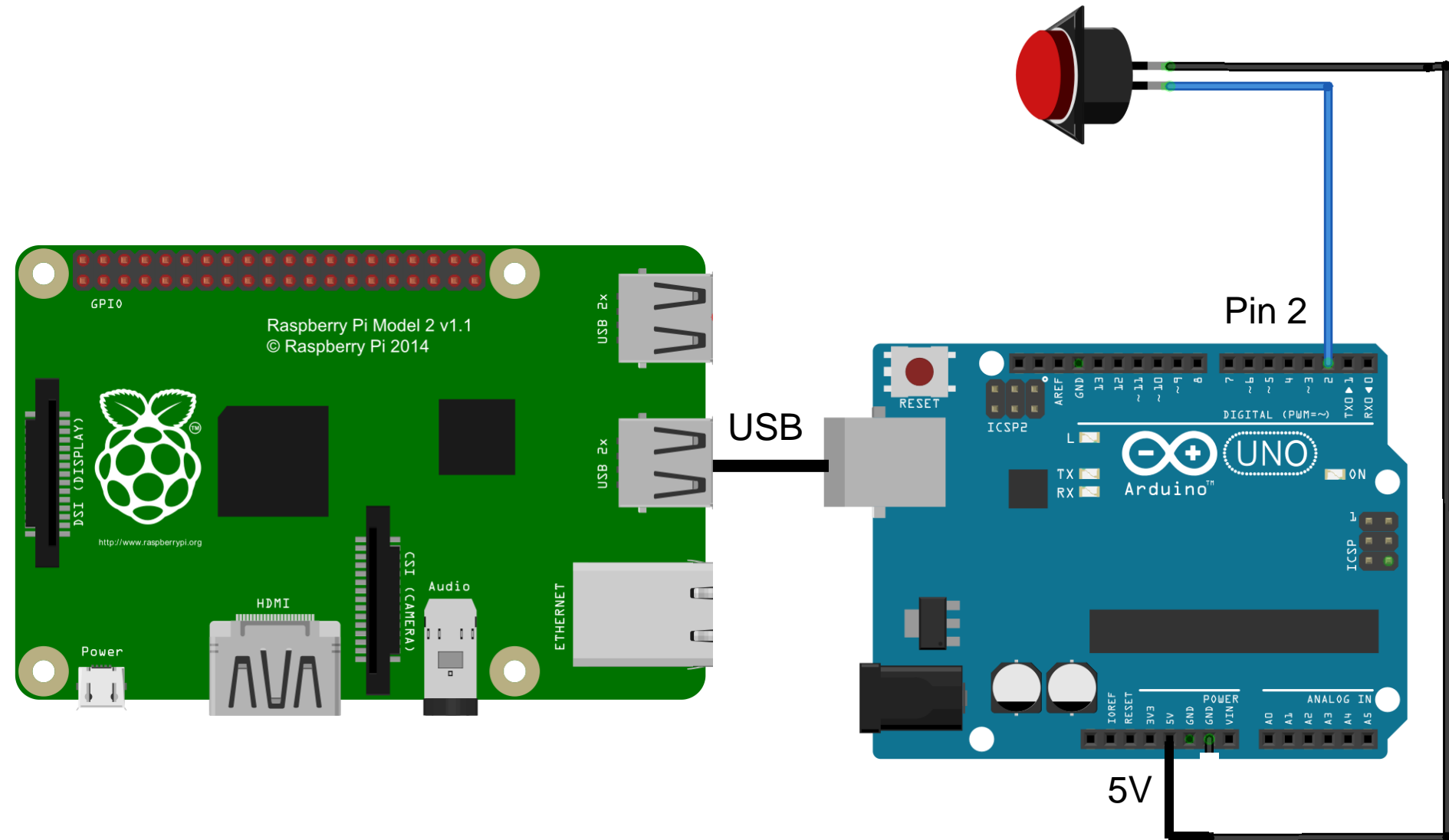
```
from nanpy import ArduinoApi, SerialManager  
from time import sleep
```

```
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 схема подключения : Нажатие кнопки



Примерный код

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

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

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

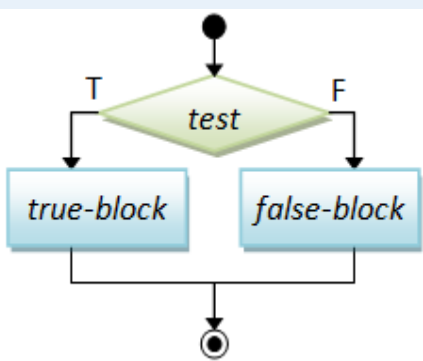
```
while True:  
    state = a.digitalRead(BUTTON)  
    print("Button state is ", state)  
    sleep(0.5)
```

#1 задача: кнопка и светодиод

□ Task(Задание)

1. Прочтите значение с pin 2 подключенного с помощью кнопки
2. Включите светодиод(pin 13) при нажатии кнопки
3. Выключите светодиод(pin 13) при отпускании кнопки

□ 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>

Содержание

❑ Arduino управления с Python

❑ Управление роботом с помощью Python

- Управление Мотором
- Измерение расстояния с помощью ультразвукового датчика
- Управление RGB LED-светодиодом
- Управление роботом через Bluetooth

Управление роботом с помощью Python

- ❑ Управление DC motor, ультразвуковой датчик и светодиод RGB с Alphabot2



Управление двигателем с TB6612



N20 motor + rubber wheel

- ☐ Управление скоростью двигателя

PWM	Выход двигателя
0~255	0 выключен, 255 на полной скорости

- ☐ Управление направлением мотора

Input1	Input2	Выход двигателя
LOW	HIGH	Повернуть в направлении по часовой стрелки
HIGH	LOW	Повернуть в направлении против часовой стрелке
все HIGH / LOW		Стоп

Arduino ↔ TB6612

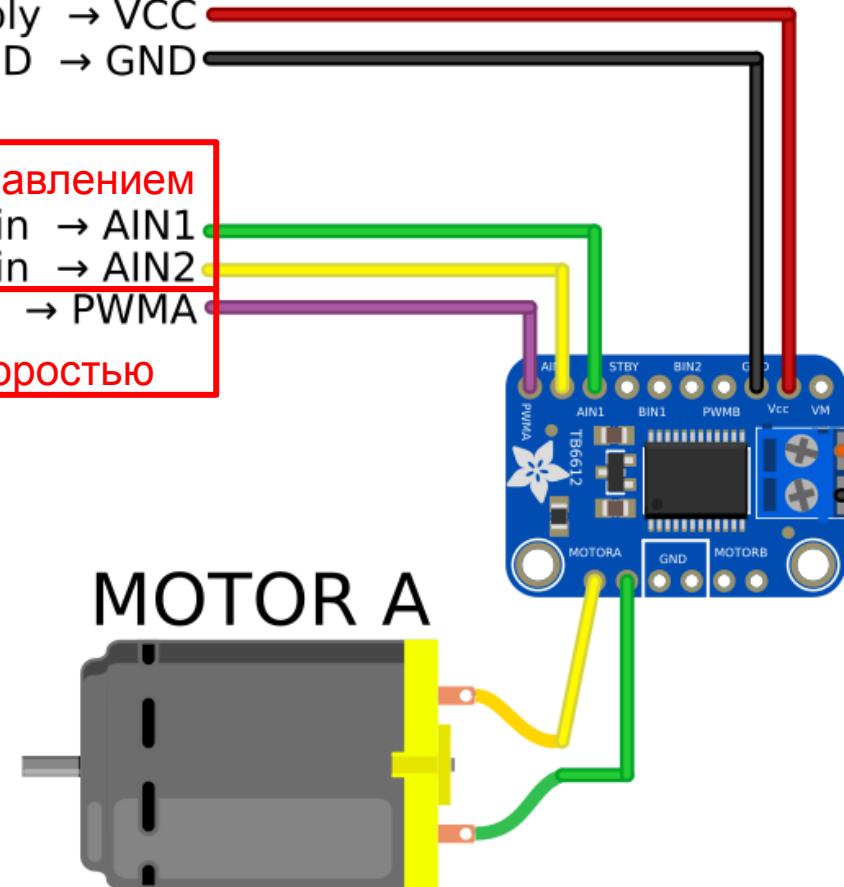
5V supply → VCC
GND → GND

Управление направлением

Digital output pin → AIN1
Digital output pin → AIN2

Analog pin → PWMA

Управление скоростью



Примерный код

```
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
```

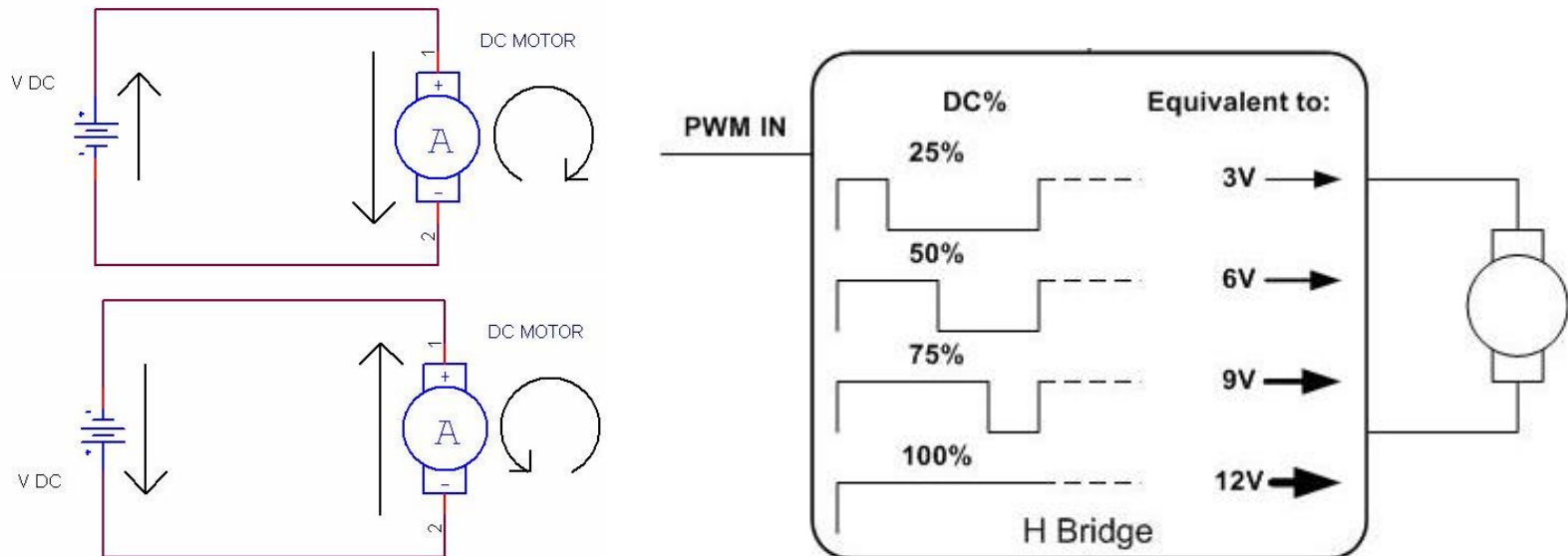
#2 задача: управление мотором

❑ Task(Задание)

1. Напишите код для перемещения робота назад
2. Напишите код, который медленно вращает робота влево или вправ

❑ Use Tip(Подсказка)

- Правильно подайте напряжение постоянного тока на два цифровых контакта
- Установите скорость двигателя, применяя соответствующее напряжение для аналогового вывода



Измерение дальности с Ultrasonic



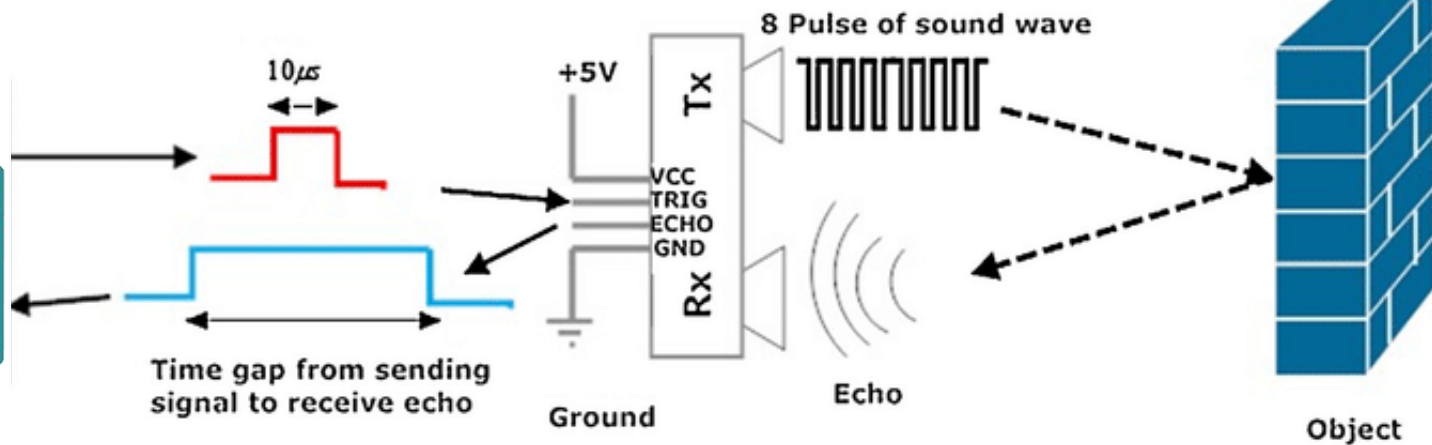
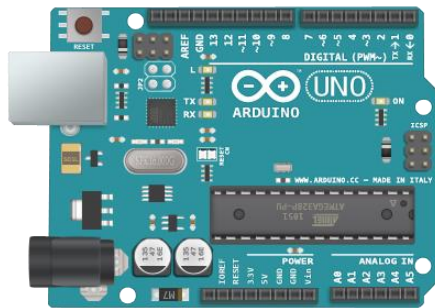
Ultrasonic

```
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



Примерный код: Ultrasonic измерение дальности

```
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 измерение дальности

□ Task(Задание)

: В соответствии с диапазоном, управляйте мотором и покажите соответствующие сообщения в таблице ниже.

Расстояние	Мотор	Сообщения
$\text{distance} \leq 5$	Стоп	Препятствие перед роботом
$5 < \text{distance} \leq 20$	Медленно вперед	Внимание к переднему препятствию
$\text{distance} \geq 20$	Быстро вперед	Нет препятствий

□ Use Tip(Использование Совет)

```
if distance <= 5:
```

```
...
```

```
elif 5 < distance <= 20:
```

```
...
```

```
else:
```

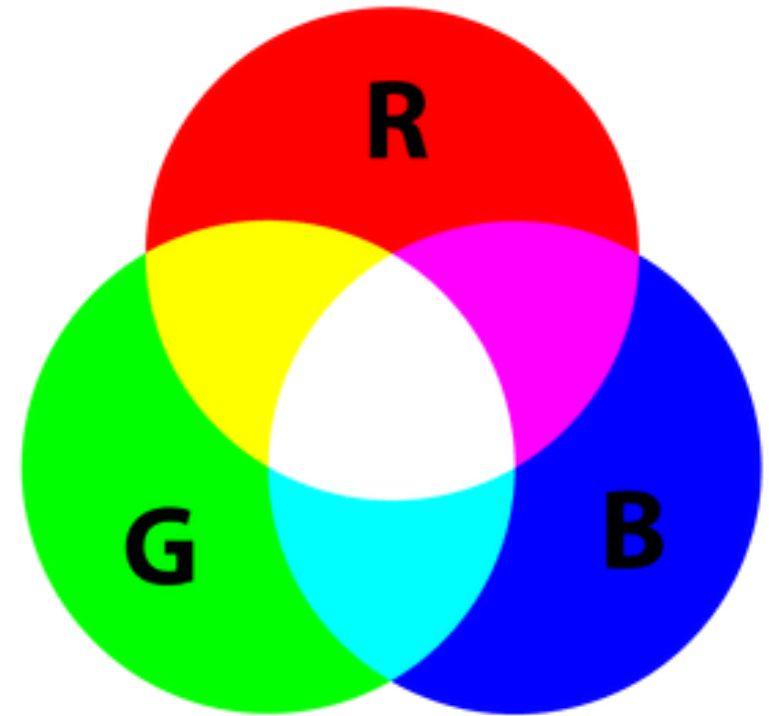
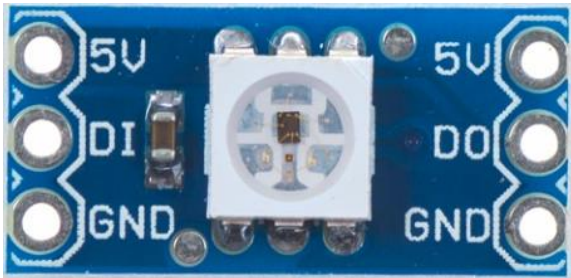
```
...
```


Управление RGB LED-светодиодом с помощью W2812



RGB LED

- ❑ RGB или Red-Green-Blue светодиоды являются слиянием трех светодиодов в одном корпусе.



Покажите любой палитр цвета , делая комбинации трех цветов

Примерный код: управление светодиодом LED RGB

```
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
```

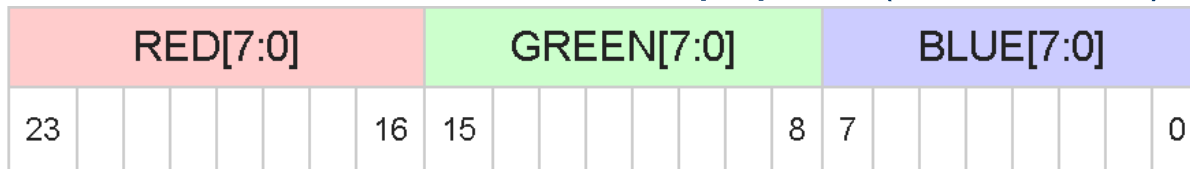
```
# Blue LED on 3rd LED
```

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

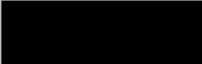







Цветовые коды RGB

❑ Цветовой формат RGB

: Код RGB имеет 24-битный формат(биты 0..23)



❑ Таблица цветов RGB

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: Управление RGB светодиодом

□ Task(Задание)

: Включите все 4 светодиода в соответствии с входным значением пользователя

Пользовательский ВВОД	Светодиод
Red	Включите Красный светодиод
Green	Включите Зеленый светодиод
Blue	Включите Синий светодиод
...	
...	

□ Use Tip(Использование Совет)

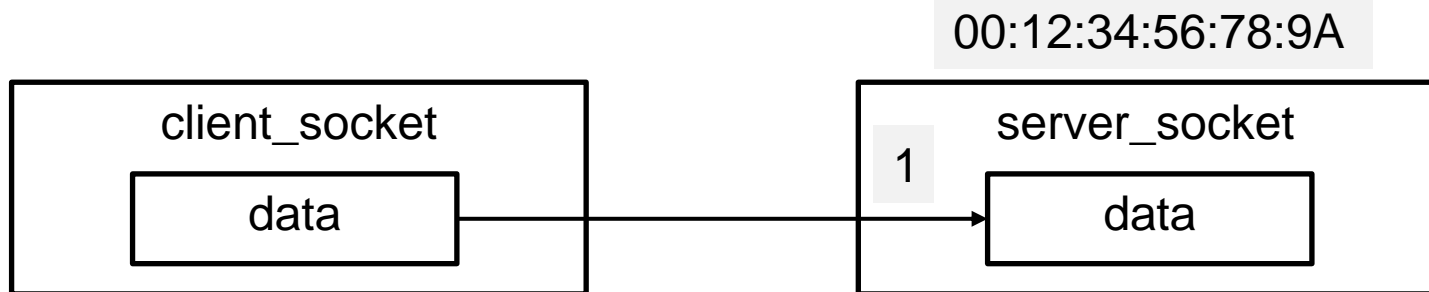
Программирование Bluetooth с Python

❑ PyBluez

- такая библиотека Python Bluetooth, написанная на C для операционных систем Windows и GNU / Linux
- позволяет коду Python получить доступ к ресурсам Bluetooth хост-машины

❑ Программирование Bluetooth в Python следует модели программирования сокетов под названием RFCOMM

- RFCOMM сокет - это протокол, ориентированный на соединение, похожий на TCP
- один процесс действует как сервер, принимающий соединения, другой процесс действует как клиент, запрашивающий соединение



Примерный код: Bluetooth(клиент)

```
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
```

[Код Python]

Примерный код: Bluetooth(Сервер)

```
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:
    }
}
```

[Код Arduino]

#5 задача: Управление Bluetooth

□ Task(Задание)

: в зависимости от Bluetooth данных, управляйте компонентами робота

Bluetooth данные	Мотор	Bluetooth данные	LED
F	Движение вперед	r	Красный
B	Движение назад	g	зеленый
L	Поворот налево	b	Синий
R	Поворот направо	y	Желтый
S	стоп	w	Белый

□ Use Tip(подсказка)

- Напишите код Arduino для реагирования на данные Bluetooth

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

- Напишите код Python для непрерывной отправки данных Bluetooth