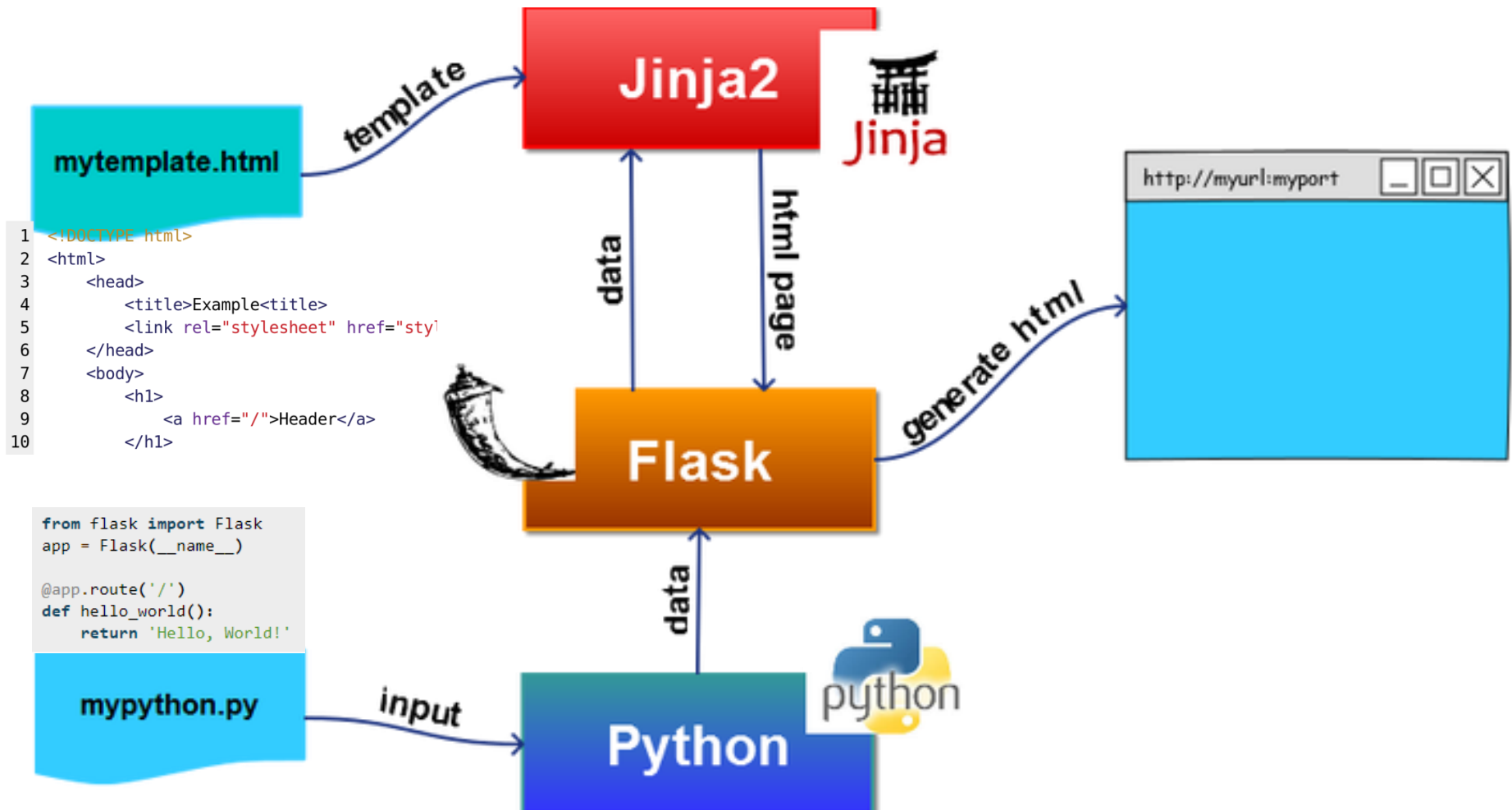




ВЕБ-УПРАВЛЕНИЕ РОБОТОМ

Веб-шаблон и приложение

❑ Как создать веб-страницу в Flask



Обработка данных в Flask

❑ Обработка данных между HTML-шаблоном и приложением Python

- `{{ }}`: Отправка данных из приложения Flask в шаблон HTML : заменяет динамическую разметку (`{{ }}`) данными
- ``: отправка данных из HTML-шаблона в приложение Flask : создайте ссылку для отправки данных и создайте новый маршрут

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>{{No}}</title>
5     <link rel="stylesheet" href="style.css">
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <nav>
12      <a href="one/">One</a>
13      <a href="two/">Two</a>
14      <a href="three/">Three</a>
15    </nav>
```

`{{ }}`

``

```
from flask import Flask
app = Flask(__name__)

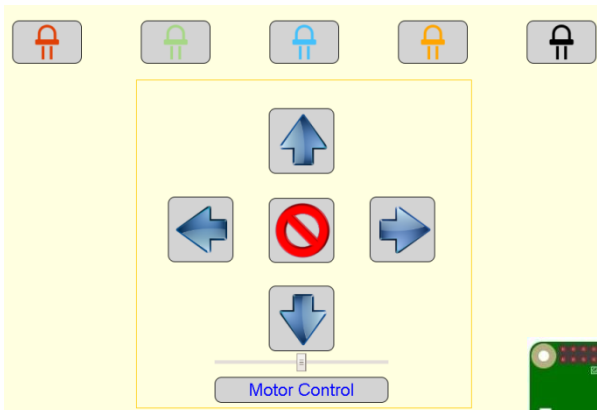
@app.route('/')
def hello_world():
    return 'Hello, World!'

@app.route('/<control>')
def led(<control>):
    if control == 'one':
        ....
    return render_template(
        'index.html', No = 'LED')
```

Веб-управление с помощью Nanpy

❑ Управления роботом с помощью Nanpy

- Отправите команду управления с веб-страницы в Arduino
- Управляйте роботом с помощью соответствующей команды

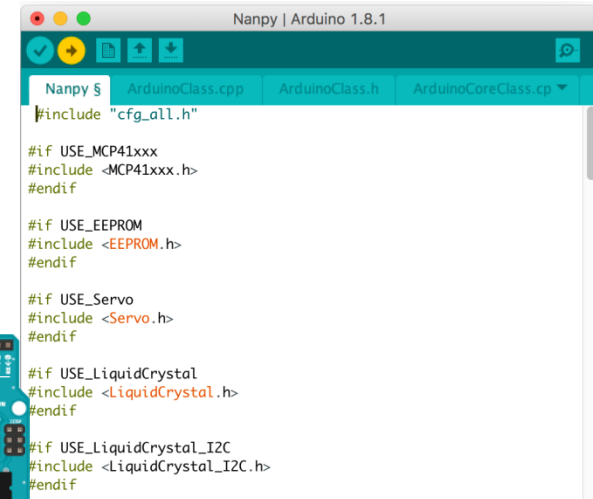
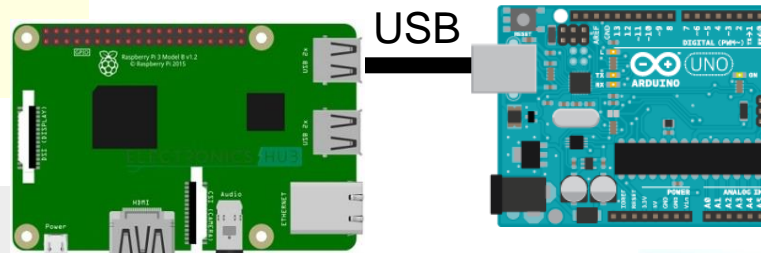


Веб-страница (HTML)

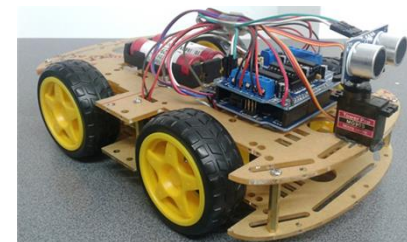
```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

Веб-сервер + Код управления роботом(Python)

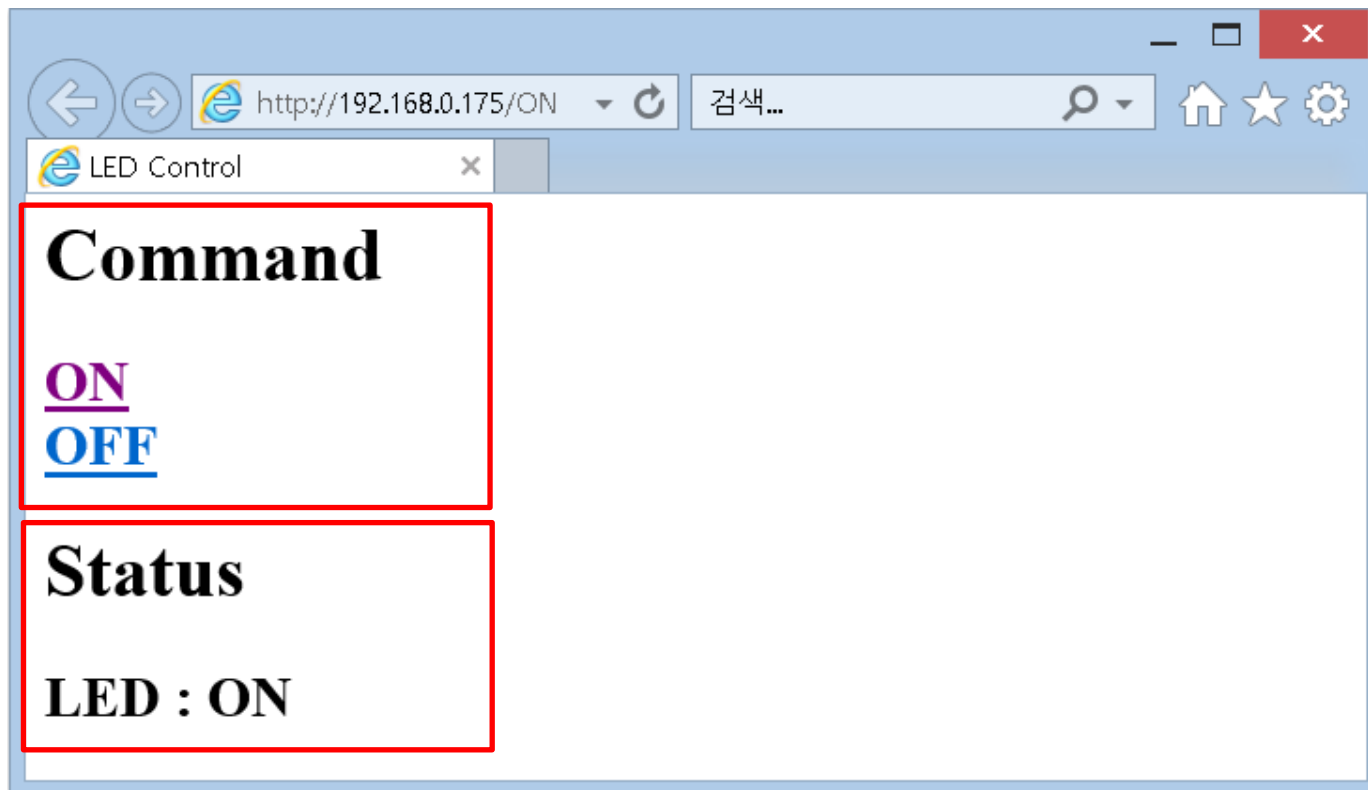


Код Nanpy (C++)



Веб-управление светодиодом

- ❑ Управление светодиодом через Интернет
 - Отправить команду ВКЛ / ВЫКЛ через веб-страницу
 - Включение / выключение светодиода и отображение статуса светодиодов



Примерный код

```
<html>
<head>
  <title>LED Control </title>
</head>
<body>
<h1> Command </h1>
<h2>
<a href = "ON"> ON </a>
<br>
<a href = "OFF"> OFF </a>
</h2>
<h1> Status </h1>
<h2> LED : {{ LED }} </h2>
</body>
</html>
```

файл html

[index.html\(шаблон\)](#)

```
from nanpy import ArduinoApi, SerialManager
from flask import Flask, render_template
```

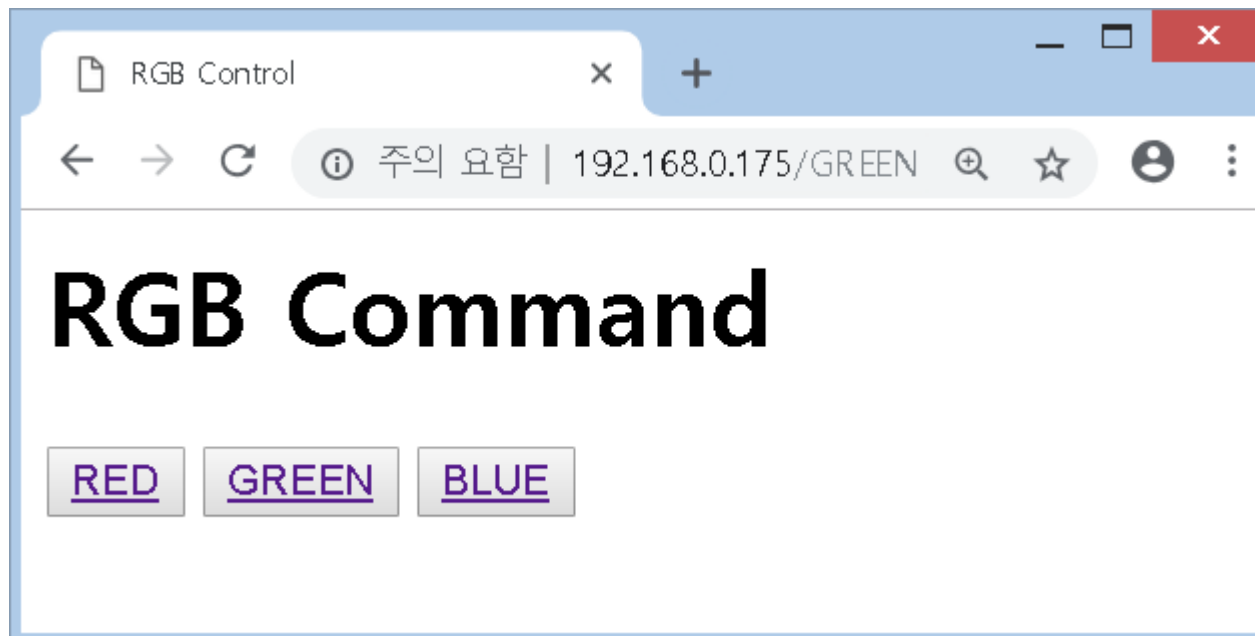
```
connection = SerialManager()
a = ArduinoApi(connection=connection)
app = Flask(__name__)
LED = 13
a.pinMode(LED, a.OUTPUT)
```

```
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/<control>')
def led(<control>):
    ledSts = 'OFF'
    if <control> == 'ON':
        a.digitalWrite(LED, a.HIGH)
        ledSts = 'ON'
    elif <control> == 'OFF':
        a.digitalWrite(LED, a.LOW)
        ledSts = 'OFF'
    return render_template('index.html', LED = ledSts)
```

файл python

Веб-управление RGB светодиодом

- ❑ Управление RGB светодиодом через Интернет
 - Отправить команду RED / GREEN / BLUE через веб-страницу
 - Кнопка команды
 - Включить соответствующий светодиод RGB



Примерный код

```
<html>
<head>
  <title> {{ target }} Control </title>
</head>
<body>
<h1> {{ target }} Command </h1>
<button><a href = "RED"> RED
</a></button>

<button><a href = "GREEN"> GREEN
</a></button>

<button><a href = "BLUE"> BLUE
</a></button>
</body>
</html>
```

файл html

index.html(шаблон)

```
from nanpy import ArduinoApi, SerialManager, W2812
from flask import Flask, render_template
```

```
NUM_RGB = 1
RGB = 7
connection = SerialManager()
a = ArduinoApi(connection=connection)
w = W2812(NUM_RGB,RGB)
app = Flask(__name__)
```

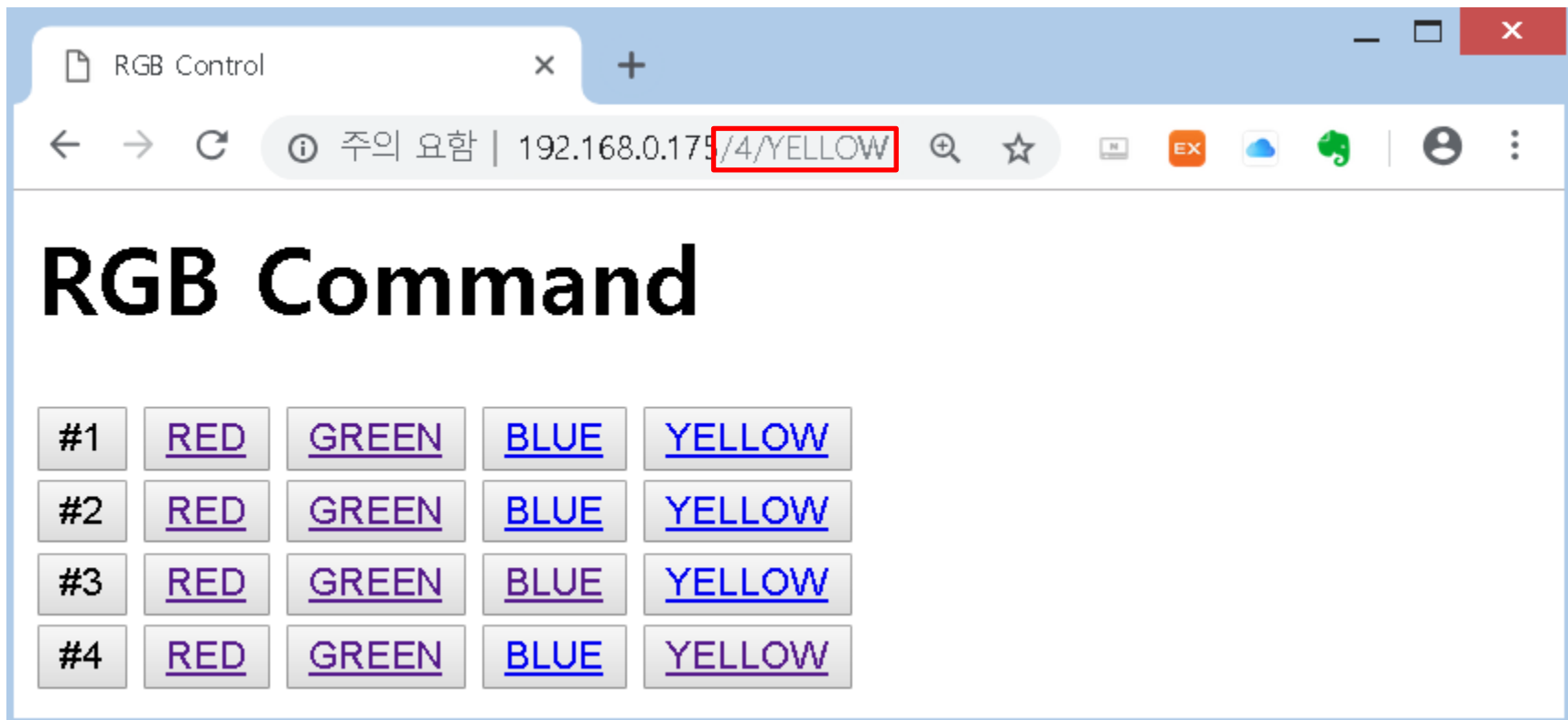
```
@app.route('/')
def index():
    return render_template('index.html', target = 'RGB')
@app.route('/<color>')
def rgb_Ctrl(color):
    if color == 'RED':
        w.setColorRGB(0, 255, 0, 0)
    elif color == 'GREEN':
        w.setColorRGB(0, 0, 255, 0)
    elif color == 'BLUE':
        w.setColorRGB(0, 0, 0, 255)
    return render_template('index.html', target = 'RGB')
```

файл python

#1 задача: Матрица RGB светодиода

□ Task(Задание)

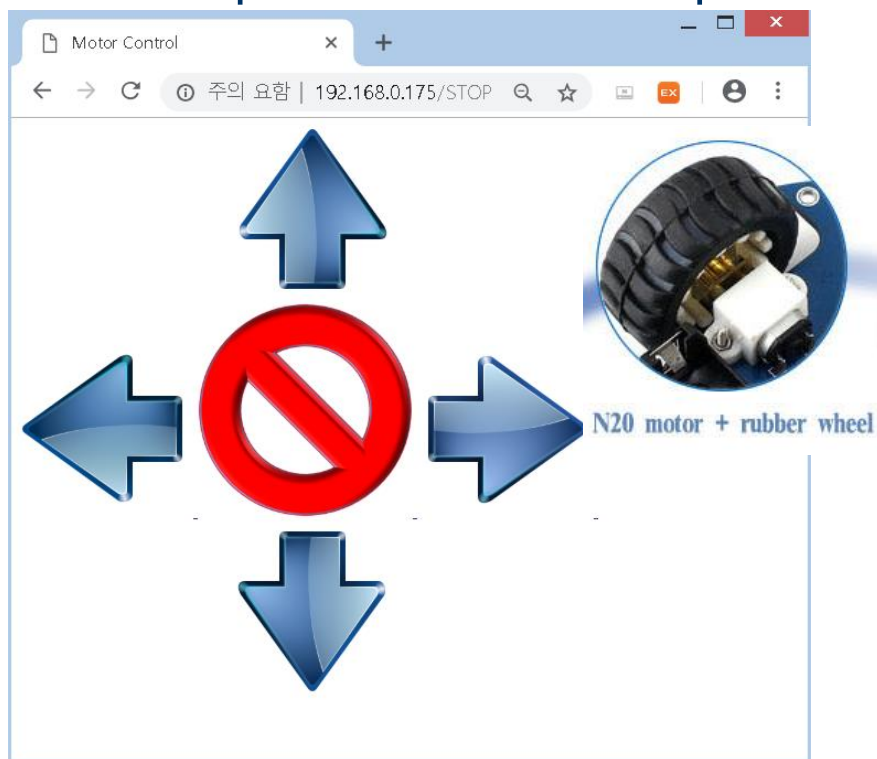
: Создайте HTML-шаблон и код приложения Python для управления RGB LED через RGB LED Matrix ниже



Управление мотором через Интернет

❑ Управление мотором через сеть

- Управление мотором через сеть Отправить команду вперед / назад / влево / вправо через веб-страницу
 - команда типа изображения
- Переместите колесо робота по соответствующей командой



backward.png



forward.png



left.png



right.png



stop.png

файлы изображений

Примерный код

```
<html>
<head>
  <title> {{ target }} Control </title>
</head>
<body>
  <a href="FORWARD">
    
  </a>
  ...
  ...
  ...
</body>
</html>
```

файл html

index.html(шаблон)

```
from nanpy import ArduinoApi, SerialManager
from flask import Flask, render_template
```

```
PWMA = 6; PWMB = 5
```

```
AIN1 = 15; AIN2 = 14
```

```
BIN1 = 16; BIN2 = 17
```

```
connection = SerialManager()
```

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

```
app = Flask(__name__)
```

```
@app.route('/')
def index():
```

```
    return render_template('index.html', target = 'Motor')
```

```
@app.route('/<control>')
```

```
def motor_Ctrl(control):
```

```
    if control == 'FORWARD':
```

```
        a.analogWrite(PWMA,SPEED)
```

```
        a.analogWrite(PWMB,SPEED)
```

```
        a.digitalWrite(AIN1,a.LOW)
```

```
        a.digitalWrite(AIN2,a.HIGH)
```

```
        a.digitalWrite(BIN1,a.LOW)
```

```
        a.digitalWrite(BIN2,a.HIGH)
```

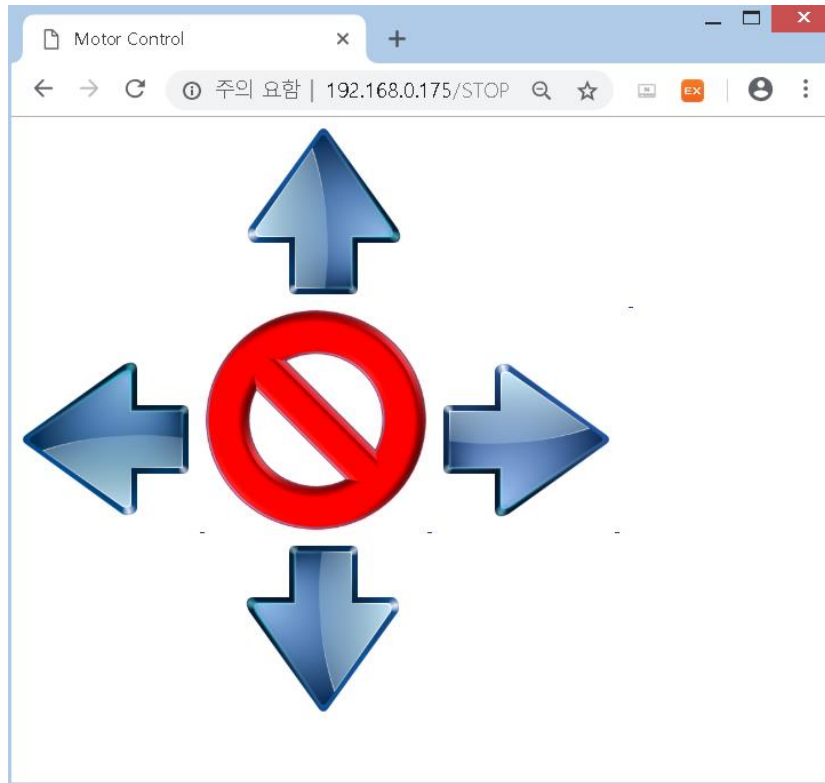
```
    return render_template('index.html', target = 'Motor')
```

файл python

#2 задача: Панель управления мотором

□ Task(Задание)

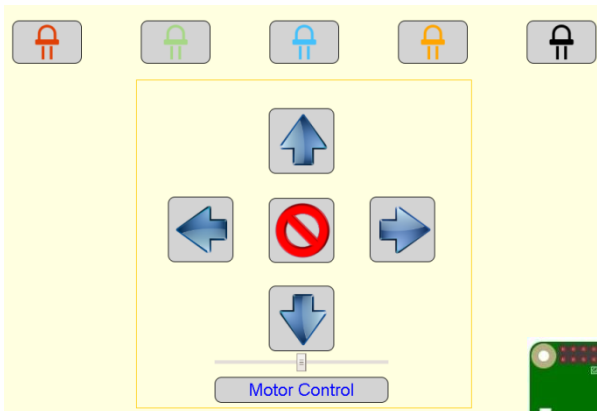
: Заполните шаблон HTML и код приложения Python для управления двигателем через панель управления мотором ниже.



Веб-управление через Bluetooth

❑ Управления роботом через Bluetooth

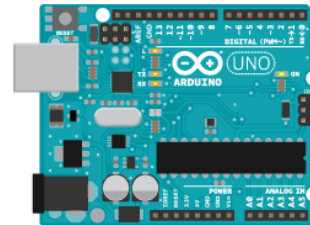
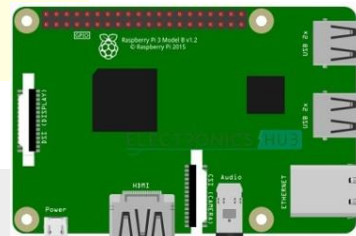
- Отправите команду управления Bluetooth с веб-страницы в робот
- Управляйте роботом с помощью соответствующей команды Bluetooth



Веб-страница (HTML)

```
from flask import Flask
app = Flask(__name__)

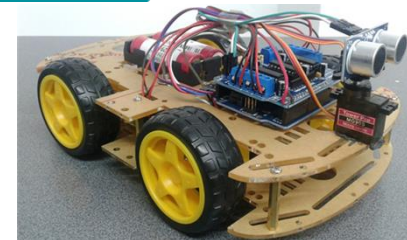
@app.route('/')
def hello_world():
    return 'Hello, World!'
```



```
void setup() {
    Serial.begin(115200);
}

void loop() {
    char data = 0;
    if (Serial.available() > 0) {
        data = Serial.read();
        Serial.print(data);
    }
}
```

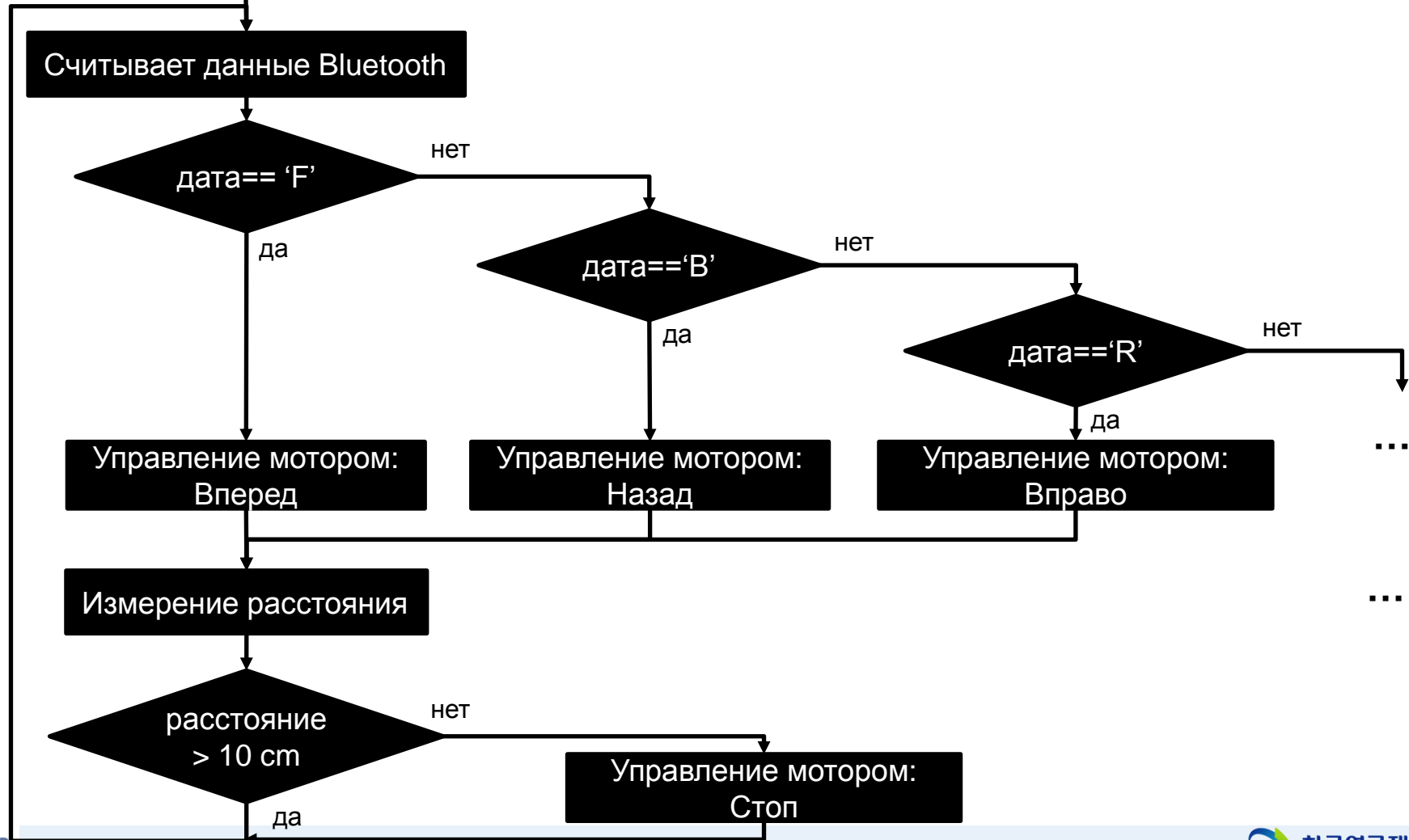
Управление роботом (C++)



Веб-сервер + команды Bluetooth (Python)

Код C++ (Arduino)

- Заданный pin
- Устанавливает скорость передачи данных



코드 Python (Задней конец)

```
import bluetooth
from flask import Flask, render_template

socket = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
socket.connect(("xx:xx:xx:xx:xx:xx", 1))

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html', target = 'Robot')

@app.route('/<control>')
def motor_Ctrl(control):
    if control == 'FORWARD':
        socket.send("F")
    elif control == 'BACKWARD':
        socket.send("B")
    ...
    return render_template('index.html', target = 'Robot')
```

Код HTML (Передний конец)

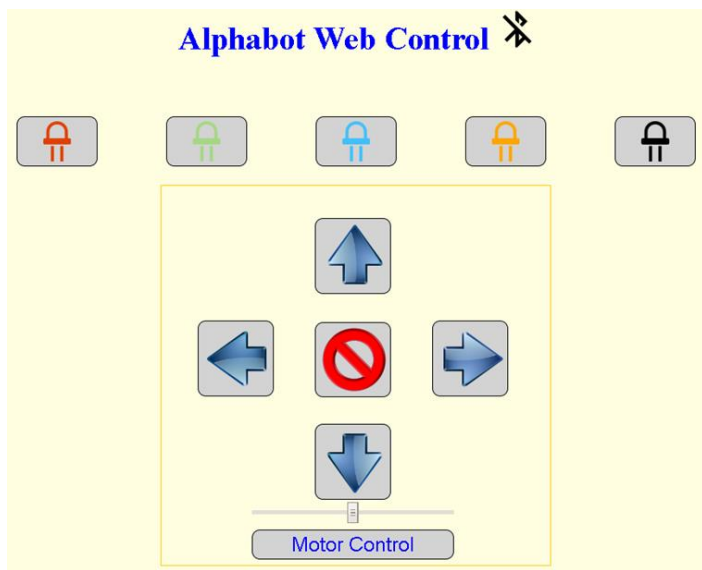
```
<html>
<head>
  <title> {{ target }} Control </title>
</head>
<body>
  <button>
    <a href = "RED"> RED </a>
  </button>
  ...
  ...
  <a href="FORWARD">
    
  </a>
  ...
  ...
</body>
</html>
```

[index.html\(шаблон\)](#)

Финальный тест

❑ Создание веб-приложения для управления роботом

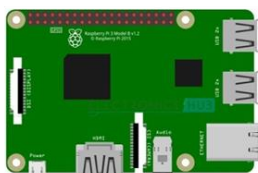
- Создайте HTML-шаблон в качестве панели управления роботом
- Напишите файл Python как веб-приложение использующий Flask
- Напишите код Arduino для управления роботом через Bluetooth



Веб-разработка



Управление роботом



Raspberry Pi and Python



Bluetooth®



Финальный тест

□ Задание

- Создание HTML-шаблона в качестве панели управления роботом
 - Сделайте любой тип панели управления, используя HTML, CSS и JavaScript
 - : любой макет, текст, цвет, кнопка, изображение,...
- Написание файла Python в качестве веб-приложения
 - Используйте Bluetooth (библиотека Python), чтобы общаться с Arduino
 - Используйте Flask (веб-фреймворк Python) для веб-приложения
- Управление роботом через веб-страницу
 - Компоненты управления роботом через панель управления роботом
 - : Светодиоды RGB, моторы, ультразвуковой датчик,...

□ Оценка

- Покажите и объясните свой код
- Проведите демонстрацию с помощью Alphabot2