

**MIT AI2 204**

# **IoT with MIT App Inventor**

**Fundamental**

X. Tang

# Raspberry Pi Wi-Fi based projects

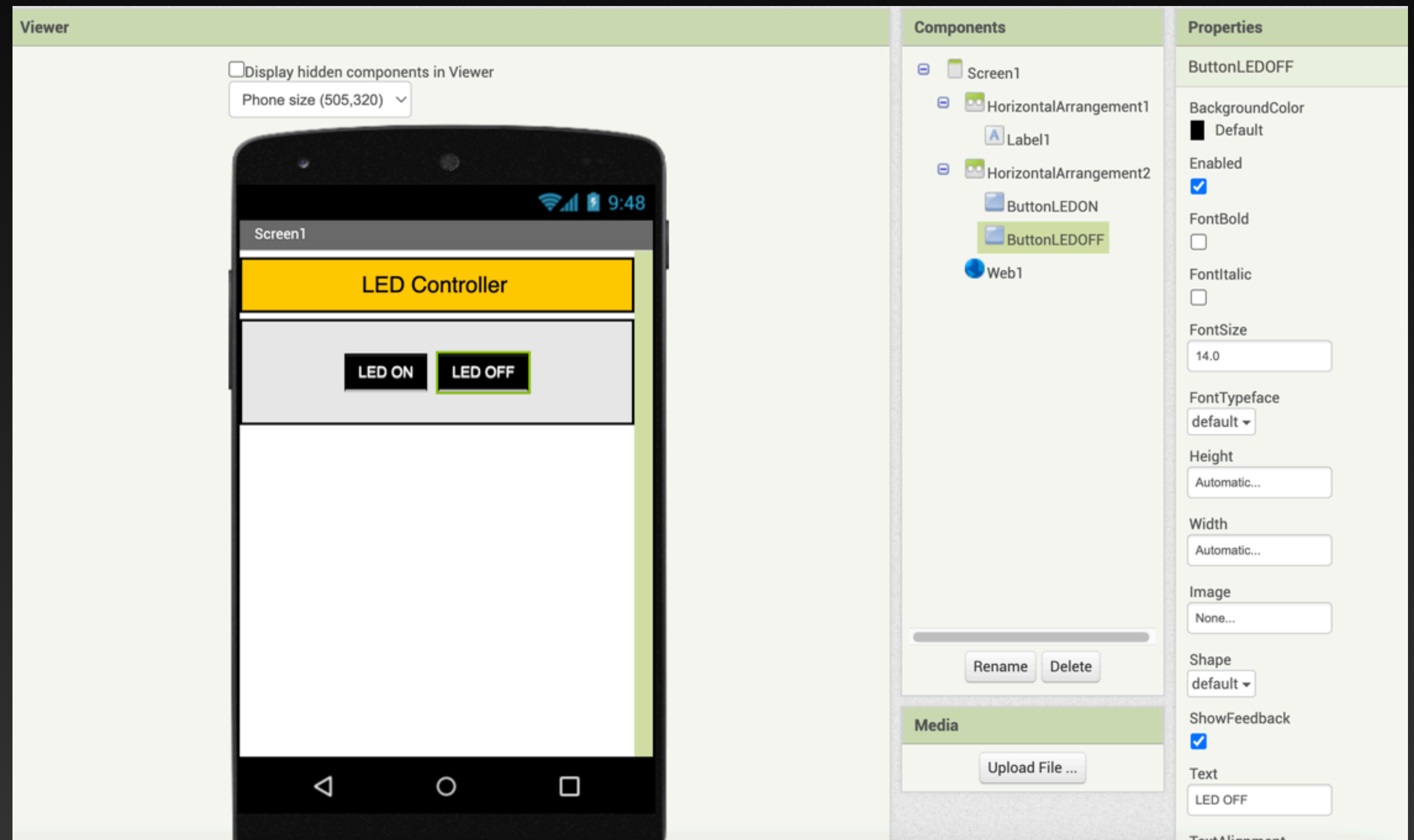
## Project 2 - Web server to control LED

Description: In this project, an LED is connected to the Raspberry Pi and is controlled from an Android mobile phone using a web server application.

Use the same circuit program from previous project to setup LED on raspberry Pi.

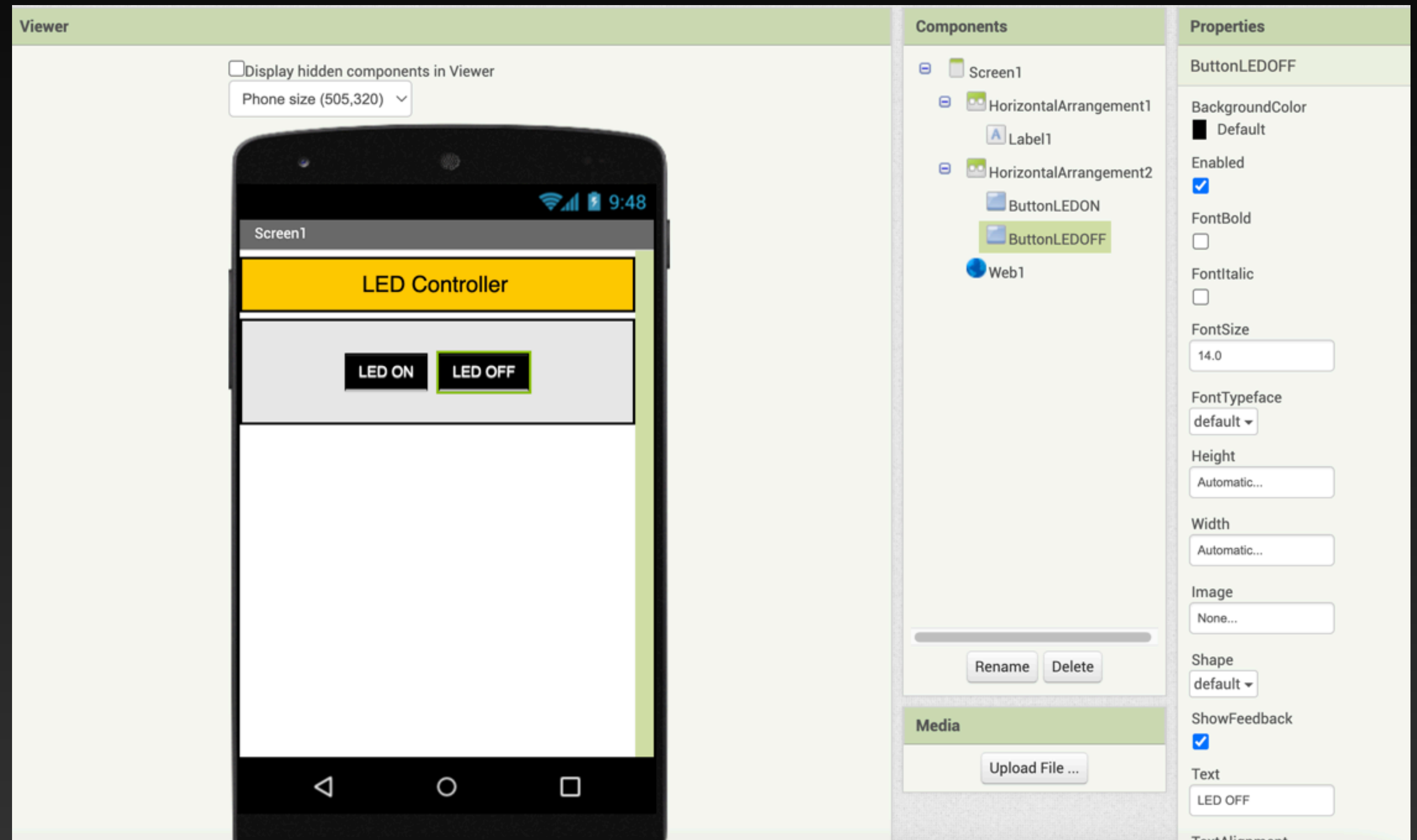
# Web Server to Control LED

- Create a new project and name it as WEB\_LED
- Insert a HorizontalArrangement and insert a Label on it with its Text set to LED CONTROLLER
- Insert another HorizontalArrangement



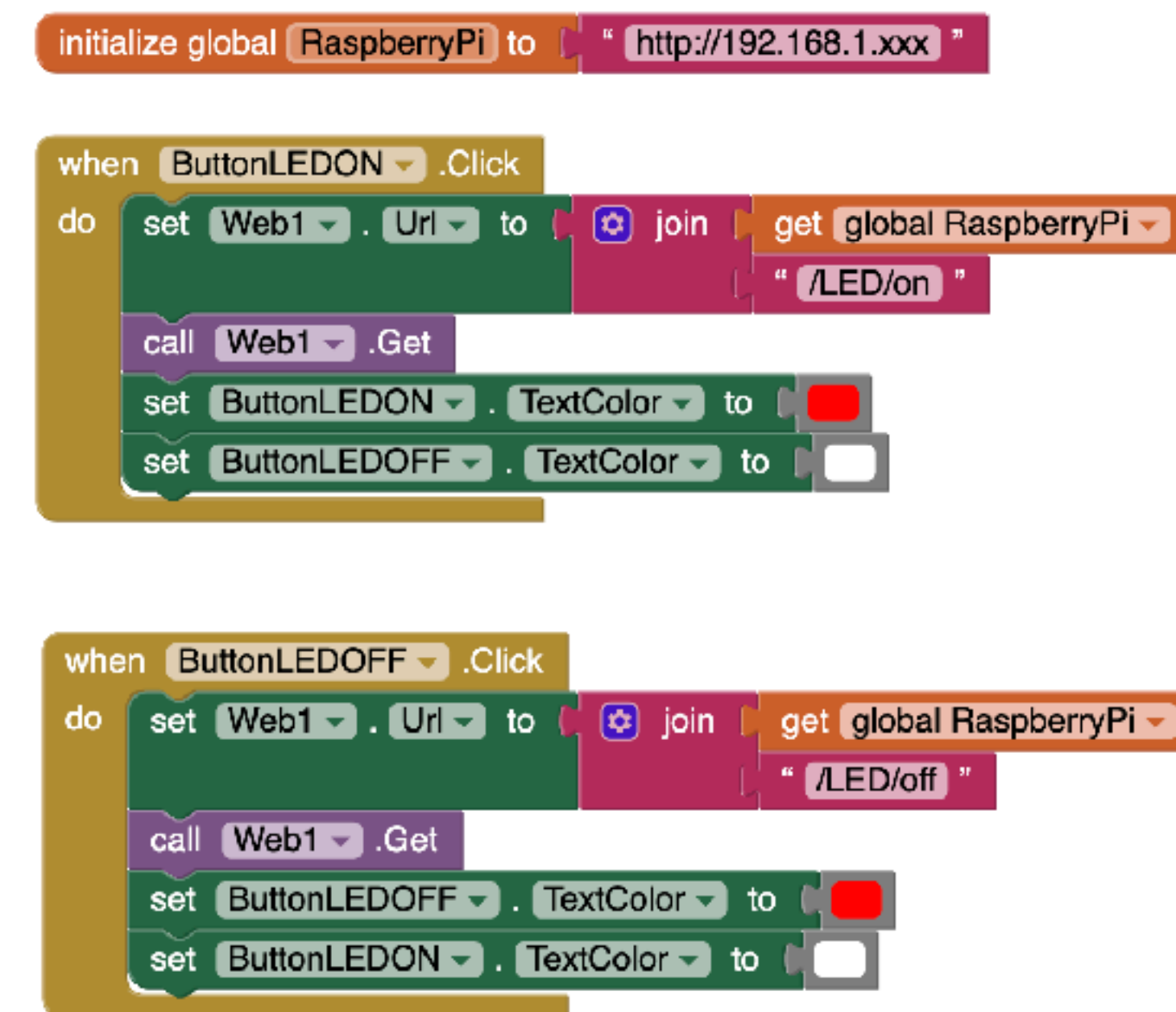
# Web Server to Control LED

- Insert two buttons on the HorizontalArrangement with the names ButtonON and ButtonOFF, with their texts set to LED ON and LED OFF respectively
- Click the Connectivity tab and insert a Web component on the Viewer. This is a hidden component.
- Initialize a variable called RaspberryPi and set it to the IP address of your RaspberryPi



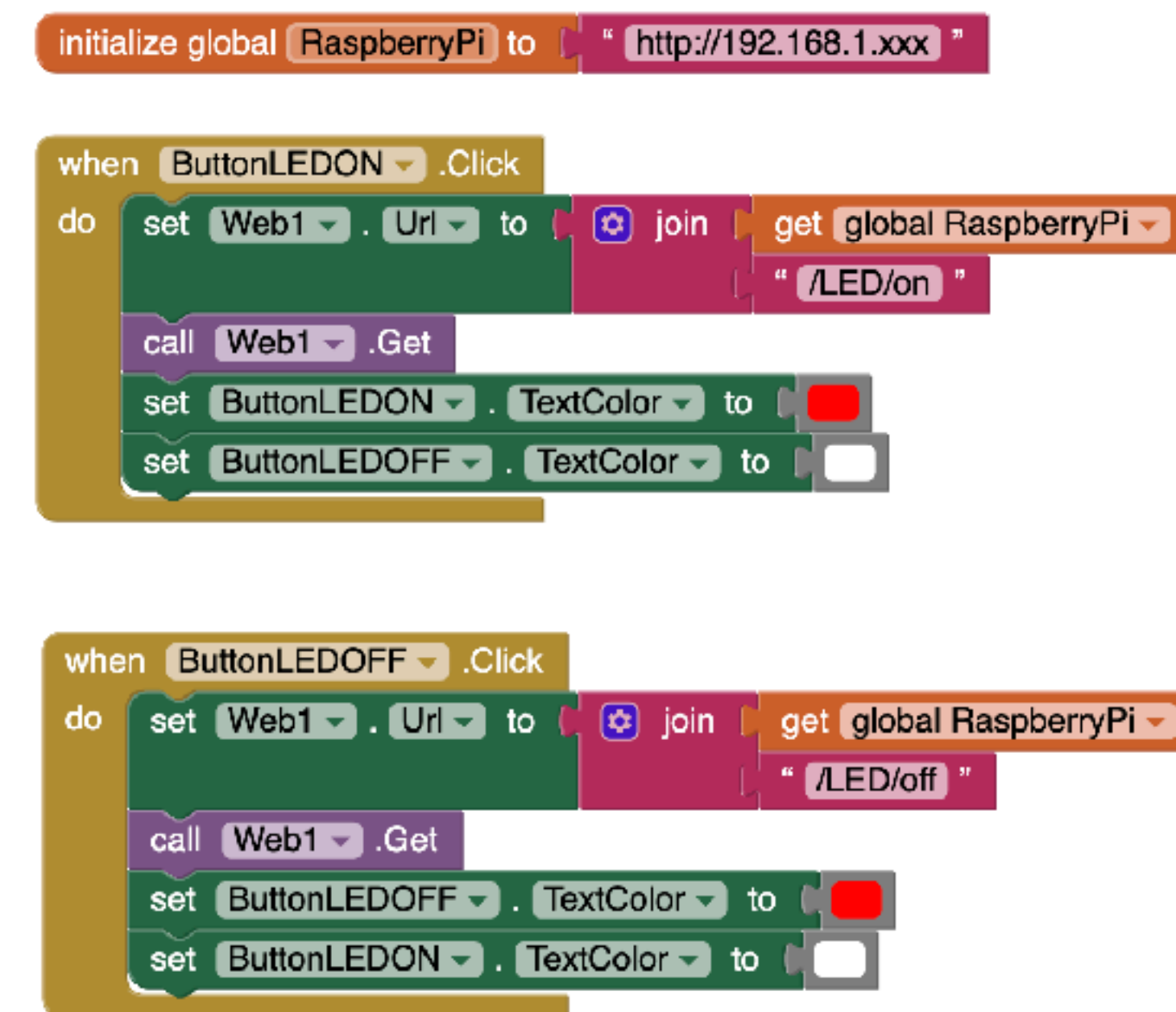
# Web Server to Control LED

- Initialize a variable called RaspberryPi and set it to the IP address of your RaspberryPi
- Click ButtonON and select when ButtonON.Click do. This block will be executed when button LED ON is clicked.
- Click on Web1 and select set Web1.Url to.
- Insert a Join block and set the URL to the IP address of your URL and add string /LED/on to this block.



# Web Server to Control LED

- Click on Web1 and select call Web1.Get. In this project, the URL is set to <http://192.168.1.xxx/LED/on>
- Set the color of ButtonON to red , and the color of ButtonOFF to white
- Repeat for the ButtonOFF as shown in the second group of blocks. But this time set the URL to <http://192.168.1.xxx/LED/off>



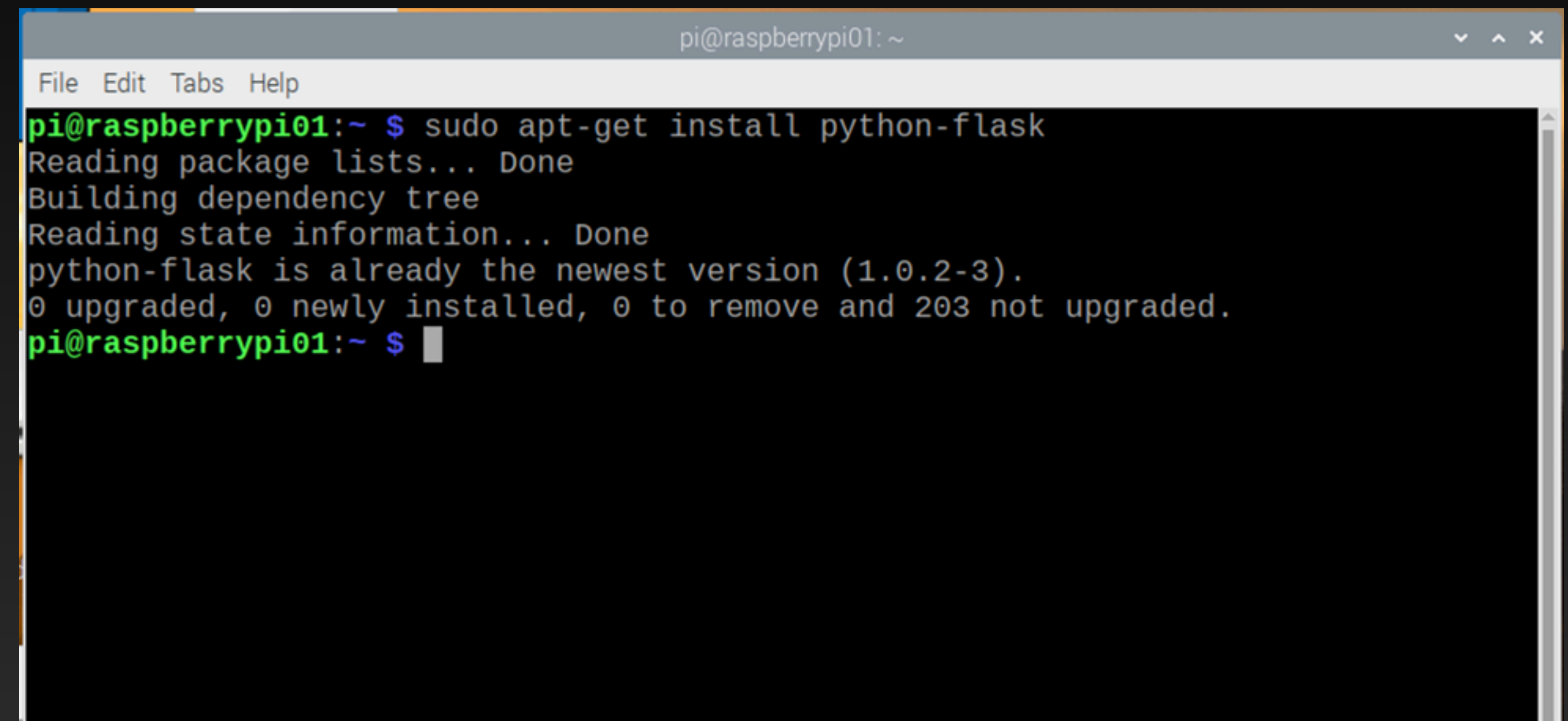


# Web Server to Control LED

Python program: Python program is based on Flask which is a simple micro-framework written in Python. It is free of charge and can be used to create a web server on Raspberry Pi to control GPIO ports over the Internet.

Flask should already be available in Python on your Raspberry Pi, but if it isn't, it can be installed using command below:

```
sudo apt-get install python3-flask
```

A terminal window titled 'pi@raspberrypi01: ~' with a menu bar containing 'File', 'Edit', 'Tabs', and 'Help'. The terminal shows the command 'sudo apt-get install python-flask' being executed. The output indicates that the package is already the newest version (1.0.2-3) and that no packages were upgraded, installed, or removed, with 203 packages not upgraded. The prompt returns to 'pi@raspberrypi01:~ \$' with a cursor.

```
pi@raspberrypi01:~ $ sudo apt-get install python-flask
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-flask is already the newest version (1.0.2-3).
0 upgraded, 0 newly installed, 0 to remove and 203 not upgraded.
pi@raspberrypi01:~ $
```

# Web Server to Control LED

Python program:

Import flask and GPIO library

```
from flask import Flask,render_template
```

```
import RPi.GPIO as GPIO
```

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

```
GPIO.setwarnings(False)
```

```
GPIO.setmode(GPIO.BCM)
```

```
LEDAppControl_WiFi.py ✕
1  #-----
2  #
3  #     LED Control from android app
4  #
5  #
6  # In this project, one LED is connected to ports GPIO 2
7  # of Raspberry Pi through a current limiting resistor.
8  # The LED is turned ON and OFF from android app
9  #
10 #
11 # Program: LEDAppControl_WiFi.py
12 # Date   : 02/27/2022
13 # Author : Xincheng Tang
14 #-----
15
16 from flask import Flask,render_template # import Flask library
17 import RPi.GPIO as GPIO
18
19 app=Flask(__name__)
20
21 GPIO.setwarnings(False) # disable warnings
22 GPIO.setmode(GPIO.BCM)  # set BCM pin numbering
23
```



# Web Server to Control LED

Python program:

Set LED to 2 which is the GPIO 2 port it is connected to. This port is configured as an output and the LED is turned OFF at the beginning of the program.

```
LED = 2
```

```
GPIO.setup(LED, GPIO.OUT)
```

```
GPIO.output(LED, 0)
```

```
LEDAppControl_WiFi.py ✕
11 # Program: LEDAppControl_WiFi.py
12 # Date   : 02/27/2022
13 # Author : Xincheng Tang
14 #-----
15
16 from flask import Flask,render_template # import Flask library
17 import RPi.GPIO as GPIO
18
19 app=Flask(__name__)
20
21 GPIO.setwarnings(False) # disable warnings
22 GPIO.setmode(GPIO.BCM) # set BCM pin numbering
23
24 LED = 2 # LED on GPIO 2
25
26 GPIO.setup(LED, GPIO.OUT) # conf LED1 as output
27
28 GPIO.output(LED, 0) # turn off LED1 to start with
29
30 #
```

# Web Server to Control LED

Python program:

An `app.route` is created with parameters `device` and `action`. For every actuator, we must have an action. If the action is on, the LED is turned ON, otherwise, if the action is off, the LED is turned OFF.

```
@app.route("/<device>/<action>")
```

```
def action(device, action):
```

```
    actuator = LED
```

```
    if action == "on":
```

```
        GPIO.output(actuator, GPIO.HIGH)
```

```
#LED1 on
```

```
    if action == "off":
```

```
        GPIO.output(actuator, GPIO.LOW)
```

```
#LED1 off
```

```
    return ""
```

```
LEDAppControl_WiFi.py ✕
25
26 GPIO.setup(LED, GPIO.OUT) # conf LED1 as output
27
28 GPIO.output(LED, 0)      # turn off LED1 to start with
29
30 #
31 # start of the main program loop, read commands from
32 # the android mobile phone, decode them and control LED
33 #
34
35 @app.route("/<device>/<action>")
36 def action(device, action):
37     actuator = LED
38     if action == "on":
39         GPIO.output(actuator, GPIO.HIGH)    #LED1 on
40     if action == "off":
41         GPIO.output(actuator, GPIO.LOW)     #LED1 off
42     return ""
43
44 if __name__ == '__main__':
45     app.run(debug=True, port=80, host='0.0.0.0', use_reloader=False)
46
47
48
49
```

# Web Server to Control LED

Python program:

Notice that the LED can be turned ON by entering the URL: <http://192.168.1.xxx/LED/on> to our web browser. Similarly, the LED can be turned OFF by the URL: <http://192.168.1.xxx/LED/off>

```
if __name__ == '__main__':  
    app.run(debug=True, port=80,  
            host='0.0.0.0', use_reloader=False)
```

```
LEDAppControl_WiFi.py ✕  
25  
26 GPIO.setup(LED, GPIO.OUT) # conf LED1 as output  
27  
28 GPIO.output(LED, 0)      # turn off LED1 to start with  
29  
30 #  
31 # start of the main program loop, read commands from  
32 # the android mobile phone, decode them and control LED  
33 #  
34  
35 @app.route("/<device>/<action>")  
36 def action(device, action):  
37     actuator = LED  
38     if action == "on":  
39         GPIO.output(actuator, GPIO.HIGH)    #LED1 on  
40     if action == "off":  
41         GPIO.output(actuator, GPIO.LOW)     #LED1 off  
42     return ""  
43  
44 if __name__ == '__main__':  
45     app.run(debug=True, port=80, host='0.0.0.0', use_reloader=False)  
46  
47  
48  
49
```



# Web Server to Control LED

Python program:

To test your program, go to the folder where your project file is stored, start python program on your Raspberry Pi from command line as:

`sudo python3 LEDAppControl_WiFi.py`

```
38 pi@raspberrypi01:~ $ ls
39 Bookshelf      Downloads      mu_code       PythonCode     test01
40 BrachioGraph   env           Music         python_games   thinclient_drives
41 Desktop       hello.py      Pictures      Templates      Videos
42 Documents     MitAI        Public        test
43
44 pi@raspberrypi01:~ $ cd MitAI/
45 pi@raspberrypi01:~/MitAI $ ls
46 decodePi.py      flashingTrafficLED.py      LEDAppControl_WiFi.py
47 DHT11App_BT.py   LEDAppControl_BT.py       MotorAppControl_BT.py
48 flashingLED_Board.py LEDAppControl_traffic_BT.py
49 flashingLED.py    LEDAppControl_traffic_WiFi.py
50
51 pi@raspberrypi01:~/MitAI $ sudo python3 LEDAppControl_WiFi.py
52 * Serving Flask app "LEDAppControl_WiFi" (lazy loading)
53 * Environment: production
54   WARNING: Do not use the development server in a production environment.
55   Use a production WSGI server instead.
hell * Debug mode: on
yth * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
>>
```

# Python Program - LED control through bluetooth

Home project1 - Develop app to control LED through speech commands

Home project2 - Develop app to control 3 LEDs through WIFI