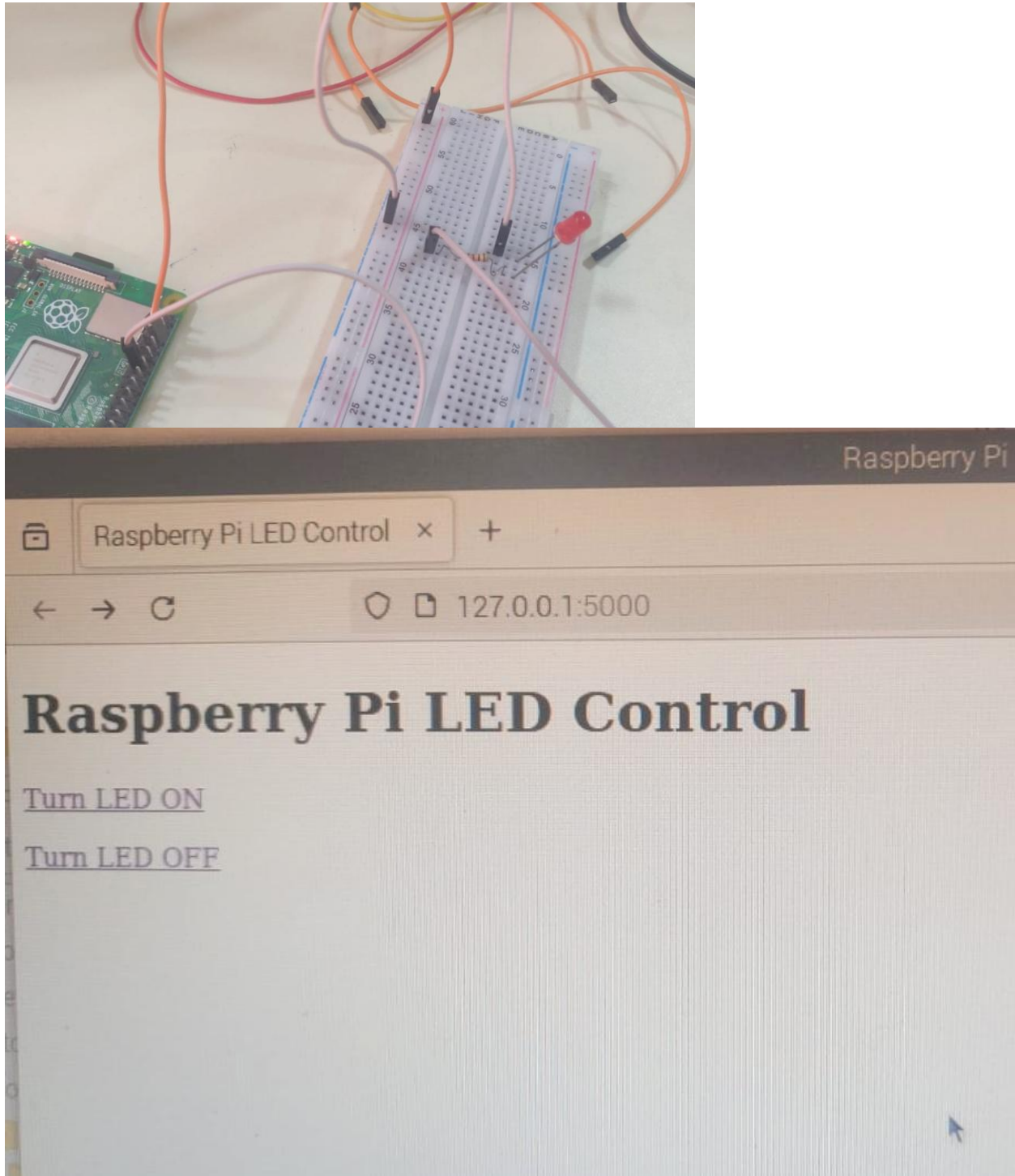


write a program to demonstrate light an LED through web

simplified guide to set up and control an LED connected to a Raspberry Pi via a web interface using Python, Flask, and GPIO:



Hardware Setup

1. Components Needed:

- Raspberry Pi (with GPIO pins)
- LED
- 220-ohm resistor
- Breadboard and jumper wires

2. Circuit Diagram:

- **LED Anode (longer leg)** → GPIO pin 17 through the 220-ohm resistor.
- **LED Cathode (shorter leg)** → GND (ground) on the Raspberry Pi.

Software Setup

1. Update Raspberry Pi:

```
sudo apt update
sudo apt upgrade
```

2. Install Python and Flask:

```
sudo apt install python3 python3-pip
pip3 install flask
```

3. Create Flask App:

- Create a file named `app.py` and add the following code:

```
from flask import Flask, render_template
import RPi.GPIO as GPIO

app = Flask(__name__)

# GPIO setup
GPIO.setmode(GPIO.BCM)
led_pin = 17
GPIO.setup(led_pin, GPIO.OUT)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/led/on')
def led_on():
    GPIO.output(led_pin, GPIO.HIGH)
    return 'LED is ON'

@app.route('/led/off')
def led_off():
    GPIO.output(led_pin, GPIO.LOW)
    return 'LED is OFF'
```

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

4. Create HTML Template:

- Create a folder named `templates` in the same directory as `app.py`.
- Inside `templates`, create a file named `index.html` with the following content:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Raspberry Pi LED Control</title>  
</head>  
<body>  
    <h1>Raspberry Pi LED Control</h1>  
    <p><a href="/led/on">Turn LED ON</a></p>  
    <p><a href="/led/off">Turn LED OFF</a></p>  
</body>  
</html>
```

Running the Flask Application

1. Start the Flask App:

```
python3 app.py
```

- Flask will start a web server on port 5000.

2. Access the Web Interface:

- Open a web browser on any device connected to the same network as your Raspberry Pi.
- Enter the IP address of your Raspberry Pi followed by :5000 (e.g., `http://<raspberry_pi_ip>:5000/`).

3. Control the LED:

- Use the links on the web page to turn the LED on and off.

Explanation of the Code

1. Flask and GPIO Imports:

- `Flask` is used to create the web server.
- `RPi.GPIO` controls the GPIO pins on the Raspberry Pi.

2. GPIO Setup:

- Set the GPIO mode to BCM.
- Configure GPIO pin 17 as an output for the LED.

3. Flask Routes:

- `/` renders the `index.html` template.
- `/led/on` and `/led/off` control the LED by setting the GPIO pin HIGH or LOW.

4. Run Flask App:

- `app.run()` starts the Flask development server and makes the app accessible on all network interfaces.

This setup allows you to control an LED connected to the Raspberry Pi through a simple web interface.

Code explanation

1. Import Libraries

```
from flask import Flask, render_template
import RPi.GPIO as GPIO
```

- **Flask:** This is the main class from the Flask web framework. It is used to create the web application.
- **render_template:** A function from Flask that renders HTML templates.
- **RPi.GPIO:** A library for controlling the GPIO pins on the Raspberry Pi.

2. Initialize Flask Application

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

- **Flask(__name__):** Creates an instance of the Flask class. `__name__` is a special variable in Python that holds the name of the current module. Flask uses this to know where to look for resources such as templates.

3. GPIO Setup

```
GPIO.setmode(GPIO.BCM)
led_pin = 17
GPIO.setup(led_pin, GPIO.OUT)
```

- **GPIO.setmode(GPIO.BCM):** Sets the GPIO pin numbering mode to Broadcom (BCM). This means you refer to the pins by their Broadcom SOC channel numbers.
- **led_pin = 17:** Defines which GPIO pin (pin 17) will be used to control the LED.
- **GPIO.setup(led_pin, GPIO.OUT):** Configures the specified GPIO pin (17) as an output pin, so it can be used to control the LED.

4. Define Routes and Views

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

- **@app.route('/')**: A decorator that tells Flask to execute the `index` function when the root URL (/) is accessed.

- **index()**: This function is called when someone visits the root URL. It returns the rendered HTML template `index.html`, which is the main page of your web interface.

```
@app.route('/led/on')
def led_on():
    GPIO.output(led_pin, GPIO.HIGH)
    return 'LED is ON'
```

- **@app.route('/led/on')**: A decorator that maps the URL `/led/on` to the `led_on` function.
- **led_on()**: This function sets the GPIO pin (17) to a high state (`GPIO.HIGH`), turning the LED on. It returns the text `'LED is ON'` to the web page.

```
@app.route('/led/off')
def led_off():
    GPIO.output(led_pin, GPIO.LOW)
    return 'LED is OFF'
```

- **@app.route('/led/off')**: A decorator that maps the URL `/led/off` to the `led_off` function.
- **led_off()**: This function sets the GPIO pin (17) to a low state (`GPIO.LOW`), turning the LED off. It returns the text `'LED is OFF'` to the web page.

5. Run the Flask Application

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

- **if __name__ == '__main__':**: This ensures that the Flask app runs only if the script is executed directly, not if it is imported as a module in another script.
- **app.run(debug=True, host='0.0.0.0')**: Starts the Flask development server.
 - **debug=True**: Enables debug mode, which provides detailed error messages and automatically reloads the server on code changes.
 - **host='0.0.0.0'**: Makes the Flask app accessible from any IP address on the local network, not just from localhost. This is useful for accessing the app from devices other than the Raspberry Pi.

Summary

- **Flask** is used to create a web server that listens for HTTP requests.
- **RPi.GPIO** is used to control the Raspberry Pi's GPIO pins to interact with hardware.
- **Routes** (e.g., `/`, `/led/on`, `/led/off`) define how the web server responds to different URL requests.
- **Templates** (e.g., `index.html`) are used to generate HTML content served to the client.
- **GPIO Control** is handled via functions that set the state of the LED based on URL requests.

This setup allows you to control an LED connected to the Raspberry Pi from any device on the same network by visiting the web interface you created.

HTML Code Breakdown

HTML file, index.html, is a basic web page used to control an LED connected to a Raspberry Pi via a Flask web server.

HTML Code Breakdown

1. **<!DOCTYPE html>**
 - This declaration defines the document type and version of HTML being used. **<!DOCTYPE html>** specifies HTML5, which is the latest version.
2. **<html>**
 - The root element of the HTML document. It contains all other HTML elements.
3. **<head>**
 - Contains meta-information about the HTML document, such as its title.
4. **<title>Raspberry Pi LED Control</title>**
 - Sets the title of the web page, which appears in the browser tab.
5. **<body>**
 - Contains the content of the web page that is visible to users.
6. **<h1>Raspberry Pi LED Control</h1>**
 - Defines a top-level heading (**<h1>**) for the web page. It's used to display a prominent title on the page.
7. **<p>Turn LED ON</p>**
 - **<p>**: Defines a paragraph element.
 - ****: Creates a hyperlink (anchor tag) that, when clicked, sends a request to the Flask server to access the /led/on route.
 - **Turn LED ON**: The text of the link that users click to turn the LED on.
8. **<p>Turn LED OFF</p>**
 - **<p>**: Defines another paragraph element.

- ****: Creates a hyperlink that sends a request to the Flask server to access the /led/off route.
- **Turn LED OFF**: The text of the link that users click to turn the LED off.

How It Works

- **Web Interface**: When you run the Flask application, it serves this index.html file to users who access the root URL of your web server.
- **Control Links**: Clicking on "Turn LED ON" or "Turn LED OFF" will send a request to the Flask server:
 - **/led/on**: This URL triggers the led_on function in the Flask application, which turns on the LED.
 - **/led/off**: This URL triggers the led_off function in the Flask application, which turns off the LED.

Summary

The index.html file provides a simple user interface with links to control the LED connected to your Raspberry Pi. It uses standard HTML elements to display a title and two links that interact with your Flask web server, allowing users to turn the LED on and off through a web browser.