

## Web Based LED Control

### Introduction

This micro-app demonstrates how to hook up an LED to the Raspberry Pi and use a web app to turn the LED on and off. The app demonstrates the use of Javascript, PHP, and HTML to control a very simple device. Referring to figure 1, the user interacts with a web browser running on computer or mobile unit that has a network connection to the Raspberry Pi. When the user requests the LED web page from the Raspberry Pi, a PHP script embedded in the web page executes on the server side. Each time the user selects an “ON” or “OFF” radio button, the PHP script executes. The PHP script, in turn, executes system GPIO commands that turn the LED on or off.

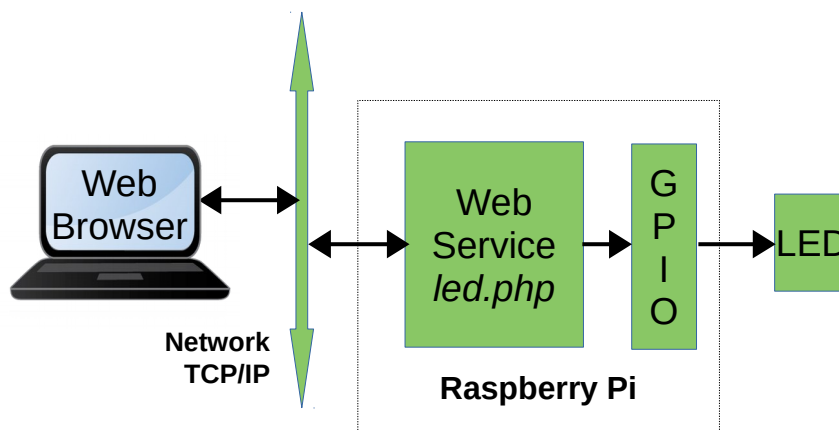


Figure 1. Block diagram showing logical relationship between web client, Raspberry Pi, and the LED.

### Hardware Hookup

Referring to figure 2, hookup the LED between one of the GPIO I/O pins and a GPIO ground pin. Use a resistor in series with the LED to limit current flow through the LED. In figure 2, the LED is connected to GPIO pin 6. Note that for Raspberry Pi 3 and above, GPIO pin numbers do not correspond with the pin numbering on the header. Therefore the LED is connected to pin 31 on the header, which corresponds to GPIO pin 6. When connecting the LED to a different GPIO pin, be sure to edit the line in the software that defines the GPIO pin number. The 220 Ohm resistor in series with the LED limits current flow through the GPIO I/O interface.

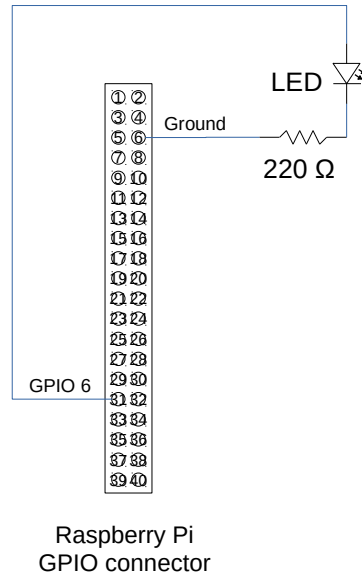


Figure 2. Schematic circuit of LED connected to the Raspberry Pi GPIO.

## Software Description

The LED is controlled via a web page where either the on or off state may be selected. From the browser, the user opens the document *led.php*, which includes a PHP script. This PHP script sends an HTML document to the client browser, and configures the GPIO pin connected to the LED. The script sets GPIO pin 6 to output mode by executing the shell command

```
gpio -g mode 6 out
```

Next, the script gets the current state of GPIO pin 6 by executing the shell command

```
gpio -g read 6
```

The script then checks to see if a form submit event caused the script to execute. The submit event occurs whenever the user selects a previously un-selected radio button. For example, if the ON radio button is currently selected and the user clicks the OFF radio button, the web page will automatically submit an HTML form to the server. The script detects that a radio button has been selected by calling the PHP function `isset` to determine if a radio button state change has occurred. If a state change has occurred the script will load the appropriate HTML code showing the new state of the LED.

The new state gets written to GPIO pin 6 by the shell command

```
gpio -g write 6 1 # Turn ON the LED
```

or

```
gpio -g write 6 0 # Turn OFF the LED
```

The above procedure may be summarized by the following steps

1. Set up in output mode the GPIO pin connected to the LED.
2. Read the current state of the GPIO pin.
3. If the ON radio button has been selected, display the ON radio button in the “checked” state. Set the PHP variable `$ledState` to 1.

4. Else if the OFF radio button has been selected, display the OFF radio button in the “checked” state. Set `$ledState` to 0.
5. Else if no radio button has been selected (when *led.php* first gets executed), display the appropriate radio button as checked based on the state of the LED (on or off) determined in step 2. That is, set `$ledState` to 0 or 1 based on step 2.
6. Write the value of `$ledState` to the GPIO pin connected to the LED.

## Software Installation

### Important notes:

1. The LED application should be installed on a Raspberry Pi which meets the following requirements:
  - The software should be installed on a recent Linux distribution such as Debian or Raspberry Pi OS.
  - Apache 2 should be installed and configured to allow serving HTML documents from the user's `public_html` folder.
  - PHP should be installed and configured to allow running PHP scripts from the user's `public_html` folder.
2. See *Setting Up the Raspberry Pi for Internet of Things* hookup guide for detailed notes on how to set up a Raspberry Pi server, and how to install and configure the components in note 1 above.

## Software Inventory

The following software items from the repository need to be installed on the Raspberry Pi

### Led html folder:

- `index.html`
- `led.php`

## Installing on a Raspberry Pi

Note that the following installation procedure assumes that the document root for HTML documents will be the user's `public_html` folder. Typically the full path name to this folder will be something like `/home/{user}/public_html`, where `{user}` is the name of the user account hosting the app. The following steps will assume the LED app is running under the `pi` user account. The `pi` account is the default account when Raspberry Pi OS is first installed on a Raspberry Pi.

1. Follow the instructions in *Setting Up the Raspberry Pi for Internet of Things* to install and configure web services on your Raspberry Pi.
2. Navigate to the **raspiot** repository archive and open the **led** folder. The following steps will involve copying files from this folder.
3. If it doesn't already exist, use `mkdir` to create a folder **public\_html** in the `pi` user's home folder. The full path to the folder should look like `/home/pi/public_html`.
4. In the **public\_html** folder, use `mkdir` to create a folder **led** to contain the LED files. The full path to the folder should look like `/home/pi/public_html/led/`.
5. Copy the contents of the repository **led/html** folder to the **led** folder created in step 4.

## LED Control – Hookup Guide

6. Open the Raspberry Pi home page in a browser running on a computer connected to the same network as the Pi. Click on the “Led” link to open the LED web page. Test the installation by turning the LED on and off a few times.

This completes installation of the led software.