Web Based Push Button Counter

Introduction

This app demonstrates how to hookup a push button to the Raspberry Pi and use a web app to monitor the number of times the button gets pushed. The app demonstrates the use of Python, Javascript, PHP, and HTML to display the number of times the button has been pushed. Referring to figure 1, the user interacts with the Raspberry Pi web service via a browser running on computer or mobile unit . When the client browser first loads the web page, Javascript submits a form to the server, and the form executes a PHP script on the server. The PHP script determines whether the push button background process is running and launches the process if necessary.

The background process, a Python script, registers an interrupt handler that gets triggered each time the button is pushed. When triggered the interrupt handler writes to a file the number of times the button has been pushed along with the time stamp of the most recent push. Periodically Javascript running on the browser reads this file and displays the number of times the button has been pushed, as well as the time and date of the most recent push.

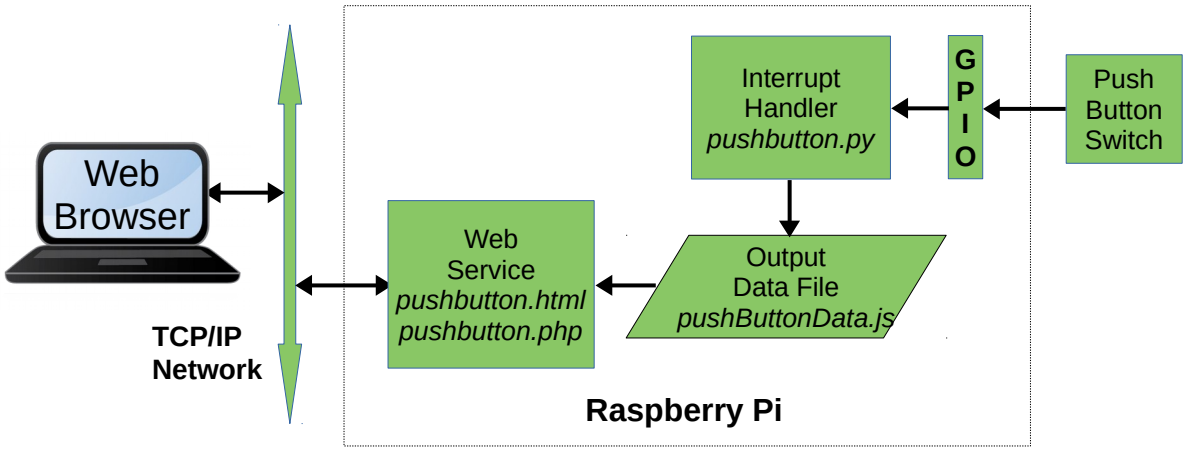

Figure 1. Block diagram showing logical relationship between web client, Raspberry Pi, and the push button switch.

Hardware Hookup

As shown in figure 2, the push button is connected between power and ground through a 10K Ohm resistor. Pin 1 of the GPIO header provides 3.3 Volts and pin 6 a connection to ground. The circuit node, connecting the resistor and switch together, is also connected to GPIO pin 5, which is pin 29 on the header, as shown in the schematic. Normally GPIO pin 5 is pulled down to ground by the 10K Ohm resistor. When the user pushes the button, GPIO pin 5 gets a 3.3 Volt pulse, which lasts as long as the user depresses the button. The rising edge of this pulse triggers the interrupt handler, updating the output data file with the number of times the button has been pushed.

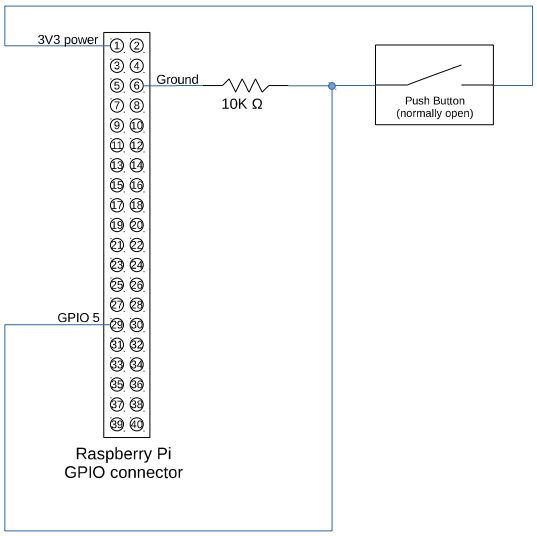

Figure 2. Schematic diagram of the push button circuit connected to the Raspberry Pi GPIO.

Software Description

The *pushbutton.py*, a Python script running as a background process, comprises the heart of the push button app. The Python script sets GPIO pin 5 to input mode and registers an event listener which calls a handler routine each time a rising edge event occurs on pin 5. The handler routine increments a counter and calls a routine that writes to the output data file the current button push count, as well as the time and date of the most recent push.

When the client browser loads the *pushbutton.html* web page, Javascript submits a form that requests, as an action, that the server execute the PHP script *pushbutton.php*. This PHP script determines if the *pushbutton.py* script is running, and, if not, starts up the script. The PHP script determines if *pushbutton.py* is running by executing the shell command

```
ps ax | awk -v a=[p]ushbutton.py '$7 ~ a {print $1}'
```

If *pushbutton.py* is already running in the background, this command will output the process ID number. If the script is not running, then the command will output nothing. If the command outputs nothing, the PHP script starts up *pushbutton.py* by executing the shell command

```
nohup sudo -u pi -S /home/pi/bin/pushbutton.py >/dev/null 2>&1 & echo $!
```

Periodically the web page Javascript sends an XML request to the server for the output data file, updated by *pushbutton.py*. The Javascript then updates the appropriate HTML document elements with the data from the response.

The above procedure may be summarized in the following steps

1. If not already running, start up the push button event handler .
2. Every time a button push event occurs, write the push button count, along with the time and data of the most recent button push, to the output data file.
3. Periodically get the time, date, and button push count from the output data file.
4. Display the date, time, and button count on the *pushbutton.html* web page.

Software Installation

Important notes:

1. The push button application should be installed on a Linux host which meets the following requirements:

    - The server 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 user public_html folders.
    - PHP 7 should be installed and configured to allow running PHP scripts from user public_html folders.
    - The Python I2C smbus library should be installed.
    - Python 3 usually comes per-installed in virtually all Linux distributions. Type “python3” at a command line prompt to verify Python has been installed.

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

Pushbutton html folder:
- index.html
- pushbutton.html
- pushbutton.php

Pushbutton bin folder:
- pushbutton.py
- btnstop

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 node power. The following steps will assume the push button 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 **pushbutton** 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 working folder should look like **/home/pi/public_html**.

4. In the p**ublic_html** folder, use **mkdir** to create a folder called **pushbutton** to contain the pushbutton HTML and PHP files. The full path should look like **/home/pi/public_html/pushbutton/**.

5. Copy all the contents of the repository **pushbutton/html** folder to the **pushbutton** folder created in step 4.

6. Copy all the contents of the repository **pushbutton/bin** folder to the **bin** folder in the user pi home folder. (The full path to this folder should look like **/home/pi/bin/**.)

7. Create a folder in the temporary file system for dynamic content, and modify the folder's ownership and permissions to allow the Apache www-data user to have access.

   mkdir /tmp/pushbutton
   sudo chown :www-data /tmp/pushbutton
   chmod g+w /tmp/pushbutton

8. [Optional] The commands in step 7 may be placed in a startup shell script and run at boot up time by launching a startup script with the **su** command from **/etc/rc.local**. For example, place the above commands in a script **/home/pi/bin/startup.sh** and place the following line in the **/etc/rc.local** file.

   (su - pi -c "~/bin/startup.sh")&

   Whenever the host boots up, the **startup.sh** script will run the commands in step 7. Be sure that you grant **startup.sh** execute permissions by running

   chmod u+x ~/bin/startup.sh

9. In the **pushbutton** folder created in step 4, create a symbolic link to **/tmp/pushbutton** in the temporary file system by running

   ln -s /tmp/pushbutton dynamic

10. Acting as superuser open **/etc/sudoers** in a text editor. Add the following line to the end of the file

    ```
    www-data ALL=(ALL) NOPASSWD: /home/pi/bin/pushbutton.py
    ```

    Adding this line gives the Apache www-data user permission to execute the *pushbutton.py* script in the background from the *pushbutton.php* script.

11. Open the pidev web page in a browser running on a computer connected to the same network as the Raspberry Pi. Click on the "Pushbutton" link to open the push button web page. Test the installation by pushing the button a few times. You should see the push count increment on the web page.

This completes installation of the push button software.