Web Based Servo Controller

Introduction

This app demonstrates how to hookup a servo to the Raspberry Pi and use a web app to control the servo. The app demonstrates the use of Python, Javascript, PHP, and HTML to control the motion and angle of the servo arm. Referring to figure 1, the user interacts with the Raspberry Pi web service via a browser running on computer or mobile unit . Whenever the user changes the servo angle or selects continuous motion on the web page, a PHP script executes a Python script as a shell command.

The Python script sets up the appropriate GPIO pin to output a pulse width modulated (PWM) signal. The servo angle selected by the user determines the pulse width of the PWM signal. The GPIO pin outputs the PWM signal to move the servo arm to the selected angle. If the user chooses continuous motion, the Python script continuously increases and decreases the pulse width causing the servo arm to move back and forth continuously.

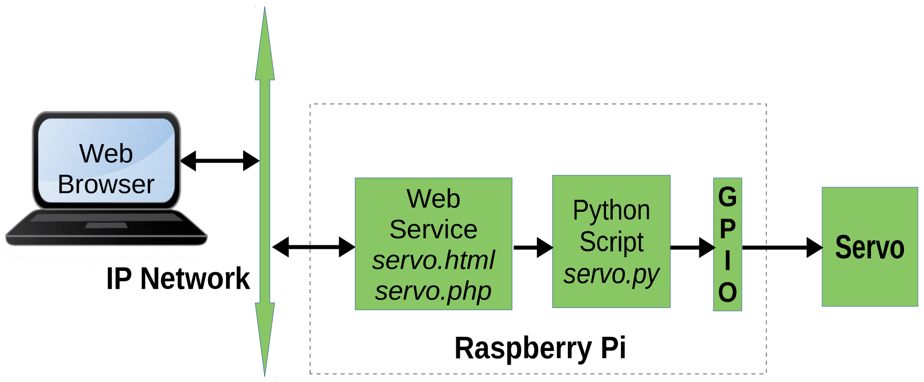

Figure 1. Block diagram showing logical relationship between web client, Raspberry Pi, and the servo.

Hardware Hookup

Referring to figure 2, hookup the servo to power, ground, and a GPIO output capable of pulse width modulation. Pin 2 of the GPIO header provides 5.0 Volts and pin 6 a connection to ground. In this example we hookup the control signal through a 1K Ohm current limiting resistor to GPIO pin 19, which is pin 25 on the GPIO header. GPIO pin 19 will be set up as an output in PWM mode. The Python script *servo.py* controls the pulse width modulated signal output from GPIO pin 19. This signal, in turn, controls the servo angle.

Important note: Be sure the Raspberry Pi can handle the transient current requirements of the servo. If the current requirements of the servo exceed what the Raspberry Pi can supply, you will need to hookup an external power supply to the servo.

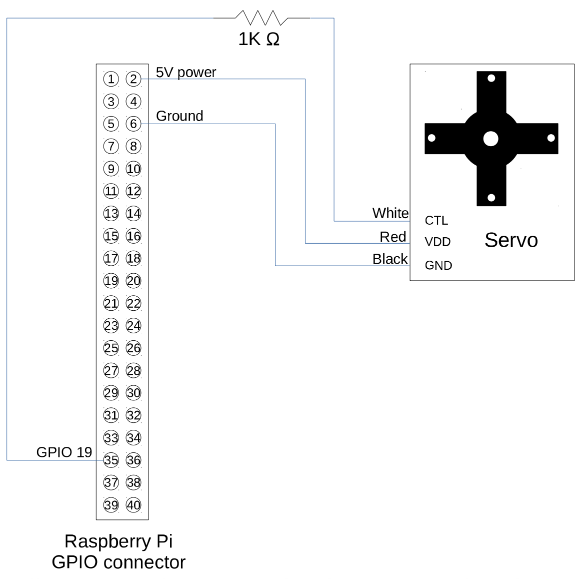

Figure 2. Schematic diagram of the servo circuit connected to the Raspberry Pi GPIO.

Software Description

The Python script *servo.py* is at the heart of the servo app. The script can operate the servo in two different modes: single motion and continuous motion. Single motion sets the servo arm to a specific position and leaves it there. In continuous motion, the servo moves back and forth continuously through its entire range of movement. The servo can also be operated from the command line in a terminal session by entering the script name followed by the appropriate arguments. For example, to move the servo arm to a specific angle, enter the following command

```
servo.py -a ANGLE
```

where ANGLE is a number between 0 and 180 degrees. To run the servo continuously back and forth, enter the command

```
servo.py -c
```

When started up, the script first kills any previously launched instances of itself. Note that if the script gets executed in the background with the continuous motion option, then the script continues to run in the background as a process until killed or terminated. As long as the script continues to run, the servo moves. Executing the script with the -k option kills this previously running instance and then exits the current instance. That is, to stop continuous servo motion when *servo.py* is running in the background, execute the following on the command line

```
servo.py -k
```

When the user opens the HTML document *servo.html* in the browser, Javascript submits a form to be acted upon by the PHP script *servo.php*. The HTML form submits two values to the PHP script whenever the user changes the angle slider or selects continuous motion: run state and angle. The PHP script executes the Python script *servo.py* in a shell with the appropriate arguments depending on the run state submitted by *servo.html*. The run state specifies whether the Python script *servo.py* gets executed with the -a, -c, or -k arguments.

The above procedure may be summarized in the following steps

1. Open the servo control web page *servo.html* in the browser. Javascript submits the default angle of 90 degrees to the PHP script *servo.php*.
2. The PHP script executes the Python script *servo.py* as a shell command with the -a option and the default angle of 90 degrees.
3. If the user changes the angle slider, Javascript will submit this new angle to the PHP script. Else if the user selects continuous motion, Javascript will submit to the PHP script that continuous motion has been selected.
4. The PHP script executes the Python script as a shell command with the appropriate options depending on whether an angle or continuous motion has been selected.
5. Return to step 3.

Software Installation

Important notes:

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

    - The server software should be installed on a recent 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 7 should be installed and configured to allow running PHP scripts from the user pi public_html folder.
    - 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

Servo html folder:
- index.html
- servo.html
- servo.php

Servo bin folder:
- servo.py

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 servo 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. If it doesn't already exist, use **mkdir** to create a folder **public_html** in the **pi** user's home folder. The full path should look like **/home/pi/public_html**.

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

4. Copy all the contents of the repository **servo/html** folder to the **servo** folder created in step 3.

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

6. 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/servo.py
   ```

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

7. Servo operating conditions change from servo to servo depending make and model. Open *servo.py* in a text editor. The following items should be modified depending on the operational parameters of your servo

   - PWM_FREQUENCY = 50 # Hertz
   - SERVO_MIN_PULSE_WIDTH = 0.75 # milliseconds
   - SERVO_MAX_PULSE_WIDTH = 2.25 # milliseconds

   For most servos the pulse width modulation frequency will be 50 Hz. Check your servo's documentation to be sure. The documentation should also give the minimum and maximum width of the control pulse.

8. Open the pidev web page in a browser running on a computer connected to the same network as the Raspberry Pi. Click on the "Servo" link to open the servo web page. Test the installation by moving the servo angle slider back and forth a few times. You should see the servo arm move back and forth.

This completes installation of the servo software.