How to use the Python demo program

For Pi-UpTimeUPS and Pi-Zero-UpTime (PiZ-UpTime) boards

(The >> line shows the commands to execute). You need to open a command line window to execute these scripts.

Step 1

Make sure you have the latest version of the operating system.

```
>> sudo apt-get update
>> sudo apt-get dist-upgrade (Skip this step if you DO NOT want to upgrade to Raspbian STRETCH. Doing this will automatically upgrade you to STRETCH. Say Yes to installing updates/upgrades.)
>> sudo apt-get upgrade
>> sudo apt-get autoclean
```

If the dist-upgrade went through with updates, please reboot the Pi to ensure you have the latest OS and updates.

Step 2

In the home directory (or some other location), create a separate folder.

```
>> mkdir some-folder-name (for example, mkdir uptime)
>> cd some-folder-name (for example, cd uptime)
```

Use wget and download the python script from the alchemy-power web site.

```
>> wget http://alchemy-power.com/wp-content/uploads/2017/06/GPIO-shutdown-sample.zip
```

This creates a file GPIO-shutdown-sample.zip file.

Step 3

Unzip the GPIO-shutdown-sample.zip file using the unzip command

```
>> unzip GPIO-shutdown-sample.zip
```

This creates a GPIO-shutdown-sample.py file.

Step 4

Run the program locally from the command line, from the directory where the file GPIO-shutdown-sample.py is located.

```
>> sudo python GPIO-shutdown-sample.py
```

sudo privileges are needed so that when the battery runs low, the software can execute a proper shutdown. Without the sudo privileges, the Pi will not execute the shutdown command and the power will be provided to the Pi till the batteries run down to approximately 2.5 to 2.6 Volts. At that voltage, the electronics shut off the Pi.

You should see the time stamp every 10 seconds. You can change how frequently the time stamp is generated by changing the variable “schlaf” (German for sleep) from 10 to whatever else you want it to be in the sample code.

If you do not want the time stamp printed, you can uncomment the line

print("Time is %s " % (time.ctime())) (line number 78) to be

```
# print("Time is %s " % (time.ctime()))
```

Step 5 (Recommended)

This step is to ensure that the GPIO s tracked and the unit is shutdown if the battery runs low. To do that, we set up a crontab entry to run a script every two minutes. First extract the file as listed in the steps above. Then copy the crontab file to /usr/local/bin directory. If the directory /usr/local/bin does not exist, please create one. Execute the two commands below if the directory does not exit.

```
>> sudo mkdir /usr/local/bin
>> chmod 755 /usr/local/bin
```

```
>> sudo cp GPIO-crontab.py /usr/local/bin/uptime-gpio.py
>> sudo chmod 755 /usr/local/bin/uptime-gpio.py
```

Now add an entry in crontab (as root) to execute the command every two minutes.

```
>> sudo crontab -e
```

It will most likely ask you what your favorite editor is. Please add the line below after you have selected the editor.

```
*/2 * * * * python /usr/local/bin/uptime-gpio.py
```

Save the cron file. Recommend you do a reboot to ensure all the changes take effect.

For Asus Tinker boards, please follow recommendations below.

https://tinkerboarding.co.uk/forum/archive/index.php/thread-381.html

If you have questions, please send email to support@alchemypower.com – enjoy!