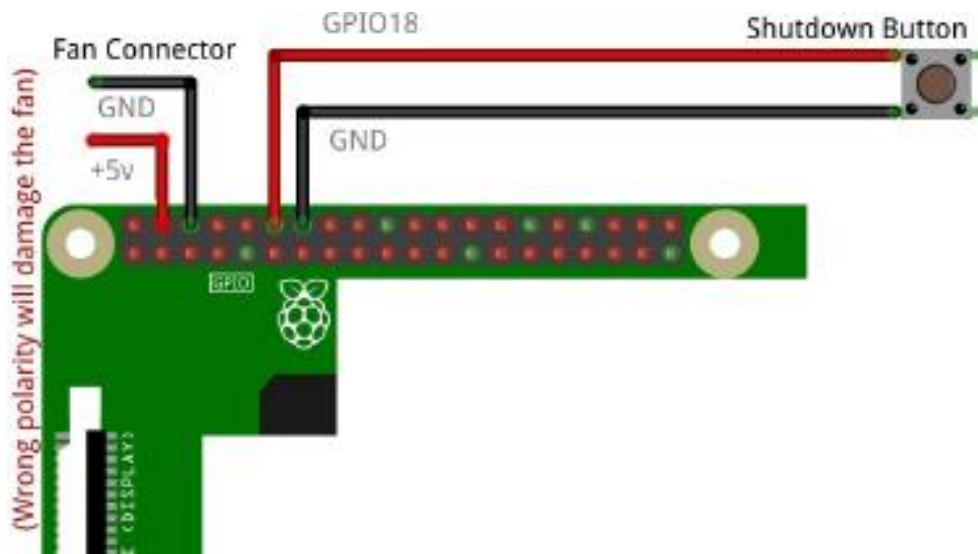


Hardware Setup:



Shutdown Button Setup:

In this tutorial we show you how to configure the proper shutdown button script at startup.

```
sudo wget https://github.com/Unio-IoT/smart-case/archive/master.zip
```

This will download the zip file named master.zip and you need to unzip the same by using the following command.

```
sudo unzip master.zip
```

After unzip you will get a directory named smart-case-master, please open the directory by using the following command.

```
cd smart-case-master
```

Make the script executable.

```
sudo chmod +x UNQ4B.pyc
```

To execute and register the script, please use the registration code provided on the sticker.

```
sudo python UNQ4B.pyc
```

```
pi@unio:~/smart-case-master $ sudo python UNQ4B.pyc
Please enter registration code : ZIuUHMQd912681UNQ4B
```

The best method to run a program on your Raspberry Pi at startup is to use the file **rc.local**. In order to have a command or program run when the Pi boots, you can add commands to the **rc.local** file. This is especially

useful if you want to power up your Pi in headless mode (that is without a connected monitor), and have it run a program without configuration or a manual start.

Editing rc.local

On your Pi, edit the file `/etc/rc.local` using the editor of your choice. You must edit it with root permissions:

```
sudo nano /etc/rc.local
```

Add commands to execute the python program, preferably using absolute referencing of the file location (complete file path are preferred). Be sure to leave the line **exit 0** at the end, then save the file and exit. In nano, to exit, type Ctrl-x, and then Y.

```
sudo python home/pi/smart-case-master/UNQ4B.pyc &
```

```
sudo python home/pi/smart-case-master/UNQ4B.pyc &
exit 0
```

If your program runs continuously (runs an infinite loop) or is likely not to exit, you must be sure to fork the process by adding an ampersand (“&”) to the end of the command, like:

The Pi will run this program at bootup, and before other services are started. If you don’t include the ampersand and if your program runs continuously, the Pi will not complete its boot process. The ampersand allows the command to run in a separate process and continue booting with the main process running.

Now reboot the Pi to test it:

```
sudo reboot
```

Hints

Also, be sure to reference absolute file names rather than relative to your home folder. For example use ``/home/pi/myscript.py`` instead of ``myscript.py``.

If you add a script into `/etc/rc.local`, it is added to the boot sequence. If your code gets stuck then the boot sequence cannot proceed. So be careful as to which code you are trying to run at boot and test the code a couple of times.