Hardware installation Safe-power (all versions of Raspberry Pi)
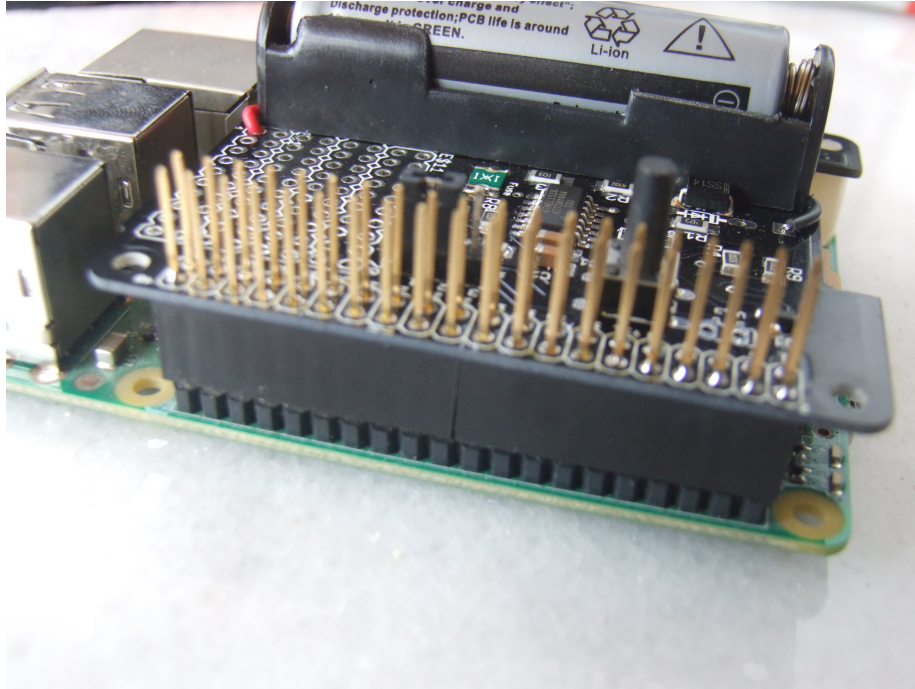
1. Install the shutdown script as per `http://safe-power.appspot.com/setup`

2. Turn off the Raspberry and disconnect USB power

3. insert the AA size LiPo battery, negative pole goes side of the spring

4. connect safe-power, Pin1 of Safe-power connects to Pin1 of the Raspberry

5. connect USB power to the Raspberry

Pin 1 Pin1 Raspberry (all models)

Operation of Safe-power

Safe-power monitors constantly the power provided by USB. In case power fails, there is no delay, and your Raspberry is immediately powered by the LiPo battery.

On Raspberry version 1 the USB ports and network will cease to work when on battery power.

If the power cut on USB takes longer than 10 seconds, Safe-power will send a shutdown signal to the Operating System. This operation will finish after 30 seconds, the red LED will be on. Battery power will be cut, and only the

microcontroller of Safe-power continues to operate. This state is indicated by a flash of the red LED every 2 seconds.

When USB power is restored the Raspberry switches back on.

The LiPo battery has it's own charging circuit with 2 LEDs (the small blue PCB underneath).

Red – the battery is charging, blue- the battery is full.

Warning: DO NOT replace the provided battery with any other rechargeable battery, except Lithium Polymer which has a charging voltage of 4.2V.

LED blink codes

Button for manual shutdown

LED red

LED green

Steady green – power has been applied, Raspberry boots

Blinking green 2 seconds – normal operation power ok

Blinking red fast – power failure detected

Steady red – shutdown initiated (manual or after power failure)

Blinking red 2 seconds – power failure, Raspberry is shutdown

Blinking red and green 2 seconds – system in shutdown after manual shutdown by button

Red and green alternating 5 times – Safe-power Microcontroller boots

Manual shutdown

Manual shutdown can be initiated by pressing the button once. The red LED will turn on for 35 seconds and your Raspberry will shut down the operating System.

After completed shutdown, red and green will blink. You can now restart the Operating system by pressing the button.

shutdownscript

```python
#!/usr/bin/env python
#script to shutdown the raspberry by safe-power raspberry UPS
#add this script to the end of root's crontab in
#the following way
#  @reboot /path-to/safe-power.py &
# important!! dont forget the "&" in the end
#in this way the script will be started in the background
# at reboot and safe power will be operational
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
import os
import time
# GPIO 11 = pin23 set up as input.
#It is pulled up to stop false signals
GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# now the program will do nothing until the shutdown
# signal on pin 23
# starts to fall towards zero. This is why we used the pullup
#during this waiting time, your raspberry is not
```

```python
20  #wasting resources by polling
21
22  try:
23      GPIO.wait_for_edge(11, GPIO.FALLING)
24
25  #    time.sleep(5)
26  #    os.system('logger -s " restart after to powerfailure"')
27  #with the following two lines you could warn all logged users
28  #of the shutdown event,
29  # just adjust the path to your shutdown message
30      os.system("wall /etc/shutdownmessage")
31  #    time.sleep(5)
32  #   now the system will shut down
33      os.system("sudo poweroff")
34  #except if this script will be cancelled by the user explicitely
35  except KeyboardInterrupt:
36      GPIO.cleanup()         # clean up GPIO on CTRL+C exit
37  GPIO.cleanup()             # clean up GPIO on normal exit
```