Raspberry Pi UPS Hat Guide

Add battery power to your Raspberry Pi project to protect against power failures or to go mobile! Switch between mains power or battery on the fly without having to power down your Pi project!

Overview

The Raspberry Pi's small size and broad range of applications means it can be ideal for use in situations where power can be unreliable or not accessible. While most microcontrollers are capable of recovering in the event of a sudden power loss the Raspberry Pi is more like a desktop computer and so can suffer from data corruption if not properly shut down. It can also sometimes not be practical to power the Raspberry Pi from a standard power source and so batteries must be used instead.

The Raspberry Pi UPS Hat by Buyapi.ca is ideal for these circumstances.

In the event of a power failure the UPS Hat can notify the Raspberry Pi that power has been lost and using shell scripting the Pi can be instructed to shut down gracefully. The UPS HAT can also be used to provide a continuous 5 volt 3 amps to the Raspberry Pi from a battery source.

The battery included with the UPS hat is capable of powering the Raspberry Pi for about 30 minutes, more than enough to ensure the Pi shuts down gracefully and no data is lost.

Theory of Operation

The Raspberry Pi UPS Hat provides uninterrupted power to a Raspberry Pi when connected via the standard 40 pin GPIO header. Primary power is provided to the Raspberry Pi via its Micro-USB jack, which is then supplied to the UPS Hat via the GPIO header. A power supply capable of providing at least 2 amps is recommended in order to supply enough power to charge the battery and support the Pi.

Status communication between the Raspberry Pi and the UPS Hat is provided by way of three GPIO connections. The UPS Hat uses a stacking header so it can be used with other Pi hats.Mounted to the printed circuit board of the UPS Hat is a Molex connector for the battery connection, a green surface mount LED, and a momentary button.

A real time clock module (RTC) is included that can be used to keep track of time continuously and read through i2c. The UPS Hat PCB has five pins that are unpopulated, but can be used to reprogram the onboard ARM processor. These pins are not needed for normal operation and most users will not need to concern themselves with them.

Included with the UPS Hat is 4 sets of stacking standoffs for securely mounting the hat to the Raspberry Pi, and a 3.75v / 450mah battery capable of providing approximately 30 mins of emergency power. The battery can be substituted for a higher capacity one and is charged through the UPS Hat.

GPIO Connections

The UPS Hat can address 3 different pins on the Raspberry Pi GPIO header to share state information. The following GPIO connections refer to the Broadcom numbering system.

- GPIO 27 - Used by the Raspberry Pi to determine if the UPS Hat is online.
    - UPS toggles this pin every 0.5s when active.
- GPIO 17 - Used by the Raspberry Pi to determine if a power failure has occurred.
    - 0 / Low : normal (or battery power switched on manually).
    - 1 / High : power fault, switched to battery.
- GPIO 18 - Used by the UPS Hat to determine if the Raspberry Pi has shut down.

LED Indicator

The LED Indicator on the UPS Hat visually displays the status of the UPS Hat.

- Solid - UPS Hat online.
- Flashing Slowly - Battery charging.
- Flashing Quickly - Battery discharging.

Momentary Button

The power button turns the UPS Hat on and off. If power is being supplied to the Raspberry Pi by the UPS Hat toggling this button will immediately cut power to the Raspberry Pi.

How To

The following instructions are meant to guide a user through simple setup and operation. It assumes the user has assembled all the necessary equipment and that they are starting with a brand new installation.

Necessary Equipment

- UPS Hat: including battery and standoffs
- Raspberry Pi: Any model with a 40 pin GPIO header.
- Micro-USB AC/DC Adapter: 5v and >2A recommended.
- Monitor
- Keyboard

Assembly

> *NOTE!: If you are using an 18650 battery instead of the included 450mah battery you may want to attach it to the UPS Hat BEFORE you begin the rest of this assembly. This can be easily achieved with some small zip ties.*

1. Begin by installing the stacking standoffs. Place a screw through the bottom of the Raspberry Pi and thread the female-to-female coupling standoff to it on the other side.
   Repeat this step for the other three mounting holes on the Raspberry Pi. (Note: If you are using a Raspberry Pi Zero or other non standard sized board there may be only two mounting holes.
2. Orient the UPS Hat so that the black header block is facing towards the Raspberry Pi's GPIO header. Align the pins of the Raspberry Pi's GPIO header so that they fit into each hole in the black header block, ensure no pins are bent or out of position.
3. Push down gently but firmly on either side of the GPIO header until the UPS Hat is firmly attached.
4. Thread the male ends of the male-to-female coupling standoffs through the UPS Hat to secure it to the Raspberry Pi.
   Repeat this step with the three other mounting holes on the UPS Hat.
5. Attach the battery cable to the Molex connector on the UPS Hat.

The UPS Hat is now fully connected and you are ready to begin setup.

Setup

Boot into Raspbian

Insert a Micro SD card with the Raspbian Operating System installed (Buyapi.ca sells a premade SD card ready for use, or you can make your own following the official Raspberry Pi directions for installing Raspbian on an SD card).

> *NOTE!: This guide was written to be used with Raspbian Lite but Raspbian Desktop will work too.*

Insert the SD card into the Raspberry Pi's SD card slot. Connect the monitor and keyboard, plug the micro-USB power cable into the Raspberry Pi. Allow Raspbian to fully start up, login and then follow the next set of instructions.

Install the shell script

Buyapi.ca has written a shell script to provide shutdown functionality. Once Raspbian has booted to the terminal and you are connected to the internet enter the following commands:

1. Install git onto your pi
   ```
   sudo apt-get install git -y
   ```
2. Clone the shell script from the Buyapi.ca repository

```
git clone https://github.com/buyapi/ups.git
```

3. Navigate into the UPS script folder

```
cd ups/scripts
```

4. Copy the script to the /usr/bin/ directory

```
sudo cp ups.sh /usr/bin/
```

5. Make the script executable

```
sudo chmod +x /usr/bin/ups.sh
```

6. Open the rc.local file

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

7. When presented with a file like this:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

exit 0
```

8. Before the “exit 0” line add the following line:

```
bash /usr/bin/ups.sh &
```

Press Ctrl-X to exit, and then press “y” to save the changes to the rc.local file.

9. Restart the Raspberry Pi to start the script at startup.

```
sudo reboot now
```

The Raspberry Pi should now be able to gracefully shut down in the event of a power failure after five seconds. This value can be changed by editing the shell script.

Configure Shutdown Time

To configure the time when Raspberry Pi will gracefully shutdown once the main power is lost you need to change the $power_timer in the UPS.sh script. At the bottom of the UPS.sh script change the $power_timer to any value you need. In the example below, 600 is 60 seconds since the timer works in 0.1 Second increments (600*0.1= 60 Seconds):

```
#If power was not restored in 60 seconds
if (( "$power_timer" == 600 )); then
#echo $power_timer;
```

Enable the Real Time Clock (RTC)

1. Enable I2C on your Raspberry Pi:
   a. In a terminal open raspi-config

   ```
   sudo raspi-config
   ```

   b. Select the Interfacing Options menu item and press Enter
   c. Select the I2C menu item and press Enter
   d. When asked if you want to enable the ARM I2C interface select “” and press Enter
   e. Press Escape to exit raspi-config.

2. Install i2c-tools

```
sudo apt-get update
sudo apt-get install i2c-tools
```

3. Ensure the RTC is present on the i2c bus:

```
i2cdetect -y 1
```

You should see it present at address 0x68:

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

4. Open the config file and enable RTC support:

   ```
   sudo nano /boot/config.txt
   ```

5. To the end of the config file add the following line:

   ```
   dtoverlay=i2c-rtc,ds3231
   ```

6. Edit the hwclock-set file using nano or your text editor of choice.

   ```
   sudo nano /lib/udev/hwclock-set
   ```

7. Find the section that looks like:

   ```
   if [ -e /run/systemd/system ] ; then
       exit 0
   fi
   ```

   Comment out (prepend a # symbol) so that it becomes:

   ```
   #if [ -e /run/systemd/system ] ; then
   #   exit 0
   #fi
   ```

   Save the file.

8. Restart the Pi:

   ```
   sudo reboot
   ```

9. Test the clocks operation:

   ```
   sudo hwclock -r
   ```

   You should get a reasonably coherent time readout like this:

   ```
   2019-11-17 00:59:15.201057-0500
   ```

Accessories and Upgrades

2200mah 18650 Battery: Dramatically increase the endurance of your project with this 2200mAh battery.

844-774-4338, 302-500-4020
Email: support@pishop.us

Other Stores By Our Team

- PiShop.ca
- PiShop.mx