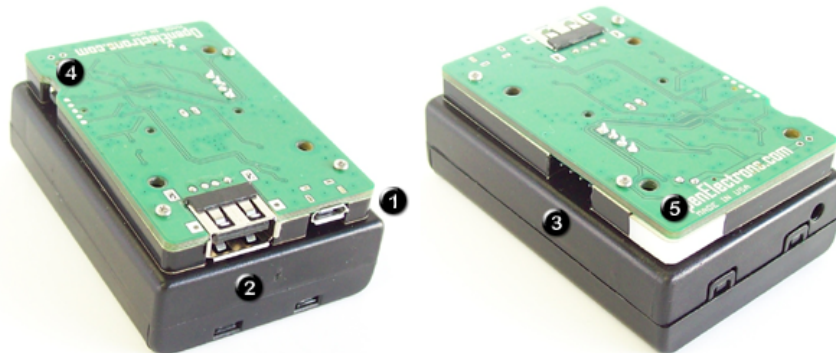


## What is SmartUPS

SmartUPS is Uninterruptible Power Supply & Portable Power for Raspberry Pi. SmartUPS runs on 3 AA NiMH rechargeable batteries.



## Connections

### 1) Power connection (micro-USB):

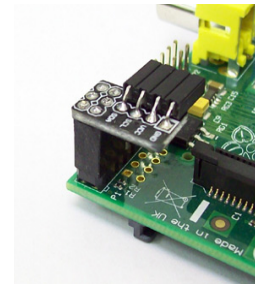
For input power connect a MicroUSB cable from mains power to connector 1 in picture above.

### 2) USB host connection:

Connect a MicroUSB cable from your Pi to USB host connector on SmartUPS. (at connector labeled 2 in picture above).

### 3) Data connection:

Connect I2C Access Point Board on GPIO pins of your Pi, (as shown in adjacent picture) and connect the 4-pin colored wire from the i2c connector of that board to i2c connector of SmartUPS (at connector labeled 3 in picture above).



## Button and LED Indicator

### 4) Press Button:

This button can be used to power SmartUPS on and off. The button clicks can also be read by your Pi over the I2C interface.

### 5) LED indicator

LED will blink to show status of the SmartUPS. See LED indications below.

## How to Use

Connect the cables as mentioned above and insert 3 AA NiMH rechargeable batteries\*\*.

### Power ON:

Press the button (labeled 4 in picture above) to power on the SmartUPS. Button must be pressed to send output power and I2C communication to Raspberry Pi anytime the LED flashes red, green, and blue upon startup.

### Power OFF:

Hold and press the button for about 4 seconds. The LED will start blinking at about 1 second interval indicating power will be turned off soon. (It will blink for about 2 minutes before powering off the Pi).

**Automatic startup and shutdown options can be configured. See Startup and Shutdown options below.**

\*\* Currently supported chemistry: NiMH

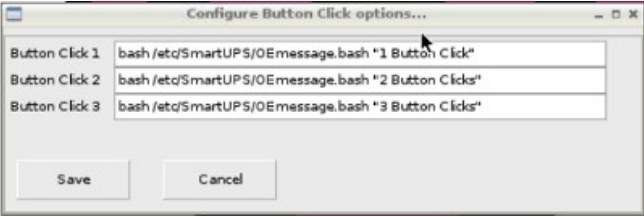
## LED indications

Flashing	Meaning
Red, Green, and Blue	SmartUPS is starting up and button must be pressed once to initialize I2C communication and send output power.
White occasional blink	Normal operation, battery is full.
Blue occasional blink	Normal operation, battery is being charged.
Yellow occasional blink	Normal Operation, battery is being drained.
Red fast blink	SmartUPS is in shutdown mode.
White blink on Power ON	When SmartUPS is powered off, it will blink once to acknowledge change in power state.
Green blink on Click	When SmartUPS button is clicked (single click or Double click), it will blink once to acknowledge.
Red occasional blink	Battery internal resistance is high (battery is going bad).
Red fast blink	Battery is very bad, and won't last long for UPS.
Red steady light	FAULT (Battery is dead or missing, or thermistor

missing, etc). The SmartUPS will still deliver power to Pi, if connected to mains, but won't operate as UPS.

### Button Clicks

You can configure Raspberry Pi to execute a



command on specific number of button clicks. For example, if you want to take a picture with your PiCamera you can configure appropriate command in upspanel for a number of button clicks (up to three). Any time you simultaneoulsy click for that set number, you will take a picture.

1 Click	Start the output power if turned off/ User defined command from upspanel.
2 Clicks	User defined command from upspanel.
3 Clicks	User defined command from upspanel.
Hold (for about 2 seconds)	LED starts blinking fast Green,indicating shutdown approaching in 50 seconds. (at this point SmartUPS issues a shutdown command to Pi).
Hold longer (about 6 seconds)	LED turns solid Red, and as soon as you let go the button it shuts down power to Pi.

### Startup and Shutdown Options

The startup and shutdown options can be configured by the upspanel GUI or by changing the SmartUPS.cfg file found in /etc/SmartUPS on your Raspberry Pi.



**\*\*Configured options will only work if the I2C cable is connected.**

**\*\*Start up options are designed for automated operation. When SmartUPS is manually shutdown (by pressing and holding the button until it power's off), upon power restoration to SmartUPS, you will have to press the button to power up your Pi.**

<b>Startup Options</b>	
Power up Immediately	Raspberry Pi will power up automatically as soon as the it receives power from SmartUPS.
Power up on Battery Stable	Raspberry Pi will power up automatically when the battery status becomes stable.
Power up on Battery Full	Raspberry Pi will power up automatically when the battery is fully charged.
No Auto Power up	Raspberry Pi will not power up automatically
<b>Shutdown Options</b>	
Shutdown Time	Raspberry Pi will shutdown automatically after a user defined time once the main power is disconnected from the SmartUPS. A value of 0 (zero) will not shutdown the Raspberry Pi until the battery status reaches 'Critical'.
Verbose	The SmartUPS will display a message through command prompt every minute of the shutdown process
Quiet	The SmartUPS will not display messages during the shutdown process

## Programming for SmartUPS

### Python Programming Interface

Python class for user programs, with interface to driver.

Download SmartUPS package from <http://pypi.python.org>

A sample script to get you started:

[http://www.openelectrons.com/index.php?module=documents&JAS\\_DocumentManager\\_op=viewDocument&JAS\\_Document\\_id=22](http://www.openelectrons.com/index.php?module=documents&JAS_DocumentManager_op=viewDocument&JAS_Document_id=22)

Modify this script to suit your needs.

## Power Characteristics

Standby time: about 500 hours with standard batteries.

Input voltage: 4.5V-5.3V (charger voltage)

Input current: 0-1.2A (charger current)

Output voltage: 5.00V

Output current: 0-1.5A

## I2C pins used

I2CAPB board uses i2c clock and signal pins as follows:

On Rev 1 board - Pin 0 and Pin 1

On Rev 2 board - Pin 2 and Pin 3

### I2C pins for other devices:

If you need to use i2c clock and signal pins for any other device, you can use them by using a fork or T joint for these pins.

## I2C Pin Layout :

The pins on the boards are as follows:

	Pin with white square paint mark			
I2CAPB	Ground	5V	SCL	SDA
SmartU PS	Ground	-	SCL	SDA

## I2C Bus address

**Factory Default Address: 0x24**

**Changing the I2C Bus Address:**

Address change instructions can be found at

<http://openelectrons.com/pages/52>

## APPENDIX A - Advanced Information

### I2C Registers:

The SmartUPS appears as a set of registers as follows:

Register	Read	Write
0x00-0x07	Firmware version - <i>Vxxxx</i>	-
0x08-0x0f	Vendor Id - <i>Openelec</i>	-
0x10-0x17	Device ID - SmartUPS	-
0x41		Command
0x42		Restart option
0x43	Button click status	
0x44	Restart time	
0x46	Battery state (See battery states for more info.)	
0x48	Battery current (signed int)	
0x4A	Battery voltage (unsigned int)	
0x4C	Battery capacity (unsigned int)	
0x4E	Estimated battery time (unsigned int)	
0x50	Battery temperature (unsigned byte)	
0x51	Battery health (unsigned byte)	
0x52	Output voltage (unsigned int)	-
0x54	Output current (unsigned int)	-
0x56	Battery maximum capacity (unsigned int)	
0x58	Seconds (unsigned long)	

### Supported I2C Commands:

CMD	Hex	Description
S	0x53	Shutdown the SmartUPS in 50 seconds.

These commands are issued on command register (0x41).

## UPSPanel Headings:

The upspanel gives multiple outputs:

Heading	Description
Heartbeat	Tick count lets user know the I2C connection is continuously reading.
Driver Version	Version of the driver.
Device Version	Firmware version of the device.
Battery Voltage	Voltage of the battery.
Battery Current	Current of the battery. If reading is negative, battery is discharging; if reading is positive, battery is charging.
Capacity	Battery capacity. SmartUPS is learning this information and data will become accurate once battery cycle is complete (full charge to full discharge)
Time Remaining	Time remaining until battery is fully discharged. SmartUPS is learning this information and data will become accurate once battery cycle is complete (full charge to full discharge).
Health	Life Health of the battery. Not charge percentage.
Charging	Charging state of the battery. See 'Battery States' section below for more information.
Battery Temperature	Battery temperature in celcius

## Battery States:

Idle - SmartUPS is figuring out the battery status

Precharg - SmartUPS is conditioning batteries for charging

Charging - battery is charging

Topup - battery is begin topped off (think fluids in a car)

Charged - battery is fully charged

Discharging - battery is discharging

Critical - battery is very low and will go into shutdown mode if not charged

Discharged - battery is fully discharged

Fault - battery is dead or missing or thermistor is missing

## Daisy chaining SmartUPS:

You can daisy chain multiple SmartUPS via the USB ports. I2C connections are only needed for the one closest to your Pi.

## Upgrading SmartUPS firmware:

SmartUPS firmware is upgradeable through the Raspberry Pi.

Firmware upgrading instructions can be found at  
[openelectons.com/pages/38](http://openelectons.com/pages/38)