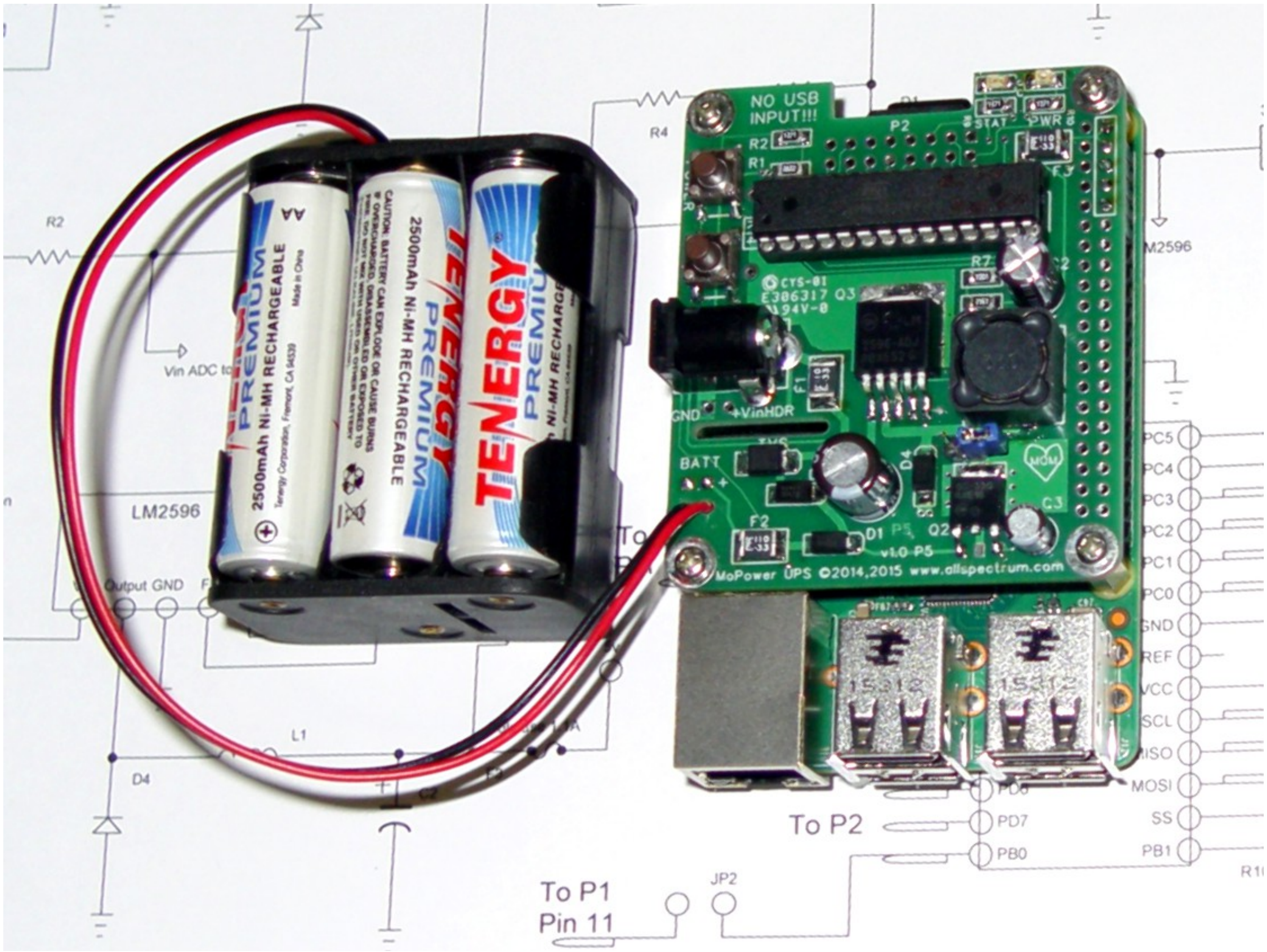
MoPower UPS Kit
Designed for the Raspberry Pi



Assembly and User Manual

Bootloader v0.91 - Firmware v0.92 – PCB v1.0 P5

Manual revision date December 13, 2015

WARNING!!! DO NOT USE USB POWER INPUT WHILE USING THIS UPS!!!

Please see the warnings and precautions page for further details

This product is provided in kit form for assembly, evaluation, testing, and use by an experienced user.

This product has not been tested against and may or may not meet various regulatory, emissions and safety regulations in your country. It is provided in kit form for assembly and use by those experienced in the components and general design of the product. Safe use of this item is ultimately the users responsibility.

LICENSE

THE SOFTWARE PROVIDED WITH THIS KIT IS LICENSED FOR USE WITH THE HARDWARE PROVIDED ONLY. All content, including text, pictures, firmware, PC Boards, design layouts and any other materials created by All Spectrum are reserved. Duplication is strictly prohibited. Copying, duplication, dis-assembly, or reverse engineering of any software or firmware contained in this product is strictly prohibited.

DISCLAIMER

THIS SOFTWARE AND HARDWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ALL SPECTRUM, OR ANY OF THEIR AGENTS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, HARDWARE OR DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Without limiting the foregoing, We make no warranty that:

the software will meet your requirements. the software will be uninterrupted, timely, secure or error-free.
the results that may be obtained from the use of the software will be effective, accurate or reliable.
the quality of the software or hardware will meet your expectations.
any errors in the software or hardware will be corrected.

Software and its documentation made available:

could include technical or other mistakes, inaccuracies or typographical errors.
may be out of date and we make no commitment to update such materials.

We assume no responsibility for errors or omissions in the software, hardware, or documentation.

WARRANTY

While no general warranty is provided, we will replace defective components for a period of 90 days from the date of purchase. Please be aware that modern semiconductors are extremely reliable and a majority of failures are caused by improper use or assembly. If you suspect a defect or malfunction in parts of the kit you can contact us for technical assistance at the following URL http://www.allspectrum.com/mopower Please note this is a defective parts warranty ONLY, we will not troubleshoot entire kits unless by prior arrangement and payment of labor fees. Replacement parts may be available for purchase on items damaged by improper use. Please contact us in case you need assistance.

Table of Contents

Introduction

The MoPower UPS is primarily designed as an Uninterrupted Power Supply for the Raspberry Pi board computer. It includes many of the features normally found in a commercial UPS as well as a few geared specifically for the Raspberry Pi, including:

- Powered by a common household DC 'wall-wart' or other source of DC power, negating the need to buy/find and test sometimes inferior quality USB chargers and cables.
- Wide input voltage range, 12-20 VDC recommended, 7-30 VDC possible
- Battery backup time of 6 hours using NIMH AA batteries or 30 minutes using NIMH 9v
- Build in trickle charge circuit to charge NIMH based rechargeable batteries.
- Other battery backup options can be used, including Super-cap or high capacity D batteries
- ATX/computer style push-button power up, restart and shutdown feature
- Watchdog for the Raspberry Pi, can automatically power cycle your Pi if it crashes!
- Real time clock, keep the time correctly on you Raspberry Pi without a network connection
- Voltage based power up and power down capability, sensitive enough to detect vehicle engine on/off status and power your Raspberry Pi accordingly, great for car computers or solar powered applications!
- Time/Interval based power up and power down, great for low power remote sensing

Circuit Description

The hardware design of the UPS is fairly simple and robust, most of the functionality comes from the programmable micro-controller at its heart. A brief circuit description follows.

Main input power passes through a 1.1A polyfuse and is protected by an optional TVS. Main input power is routed to the micro-controller for voltage measurement.

An LM317 regulator is used as a constant current source and takes main input power to trickle charge a battery pack. The battery pack is protected by a 1.1A polyfuse. Battery pack voltage is routed to the micro-controller for voltage measurement.

The main input power and battery power is routed through two diodes to a 3.3v regulator that powers the micro-controller. It also powers the main switch mode regulator IC (LM2596) that generates the 5v needed for the Raspberry Pi. The 5v that powers the Pi is protected by a 1.1A polyfuse.

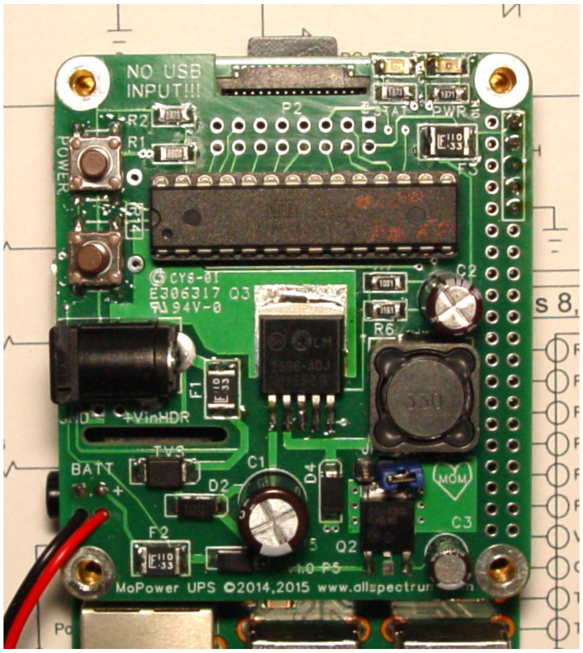
The micro-controller is the heart of the UPS and is responsible for controlling power to the Raspberry Pi. It can be programmed to control the UPS depending on various input and timer variables in addition to common UPS functions measuring input and battery voltage. Two switches are provided, one is a main power switch which controls the main power of the Raspberry Pi. A second switch is provided for custom or future use. The micro-controller has a 32khz watch crystal for providing a time reference and can be used as a Real Time Clock in the absence of a network source for time. The micro-controller uses a serial UART to communicate with the Raspberry Pi. Two LEDs, PWR and STAT, are provided for power and status display. A 16 pin connector is provided for power, input, and output for various user functions or future functionality of the UPS.

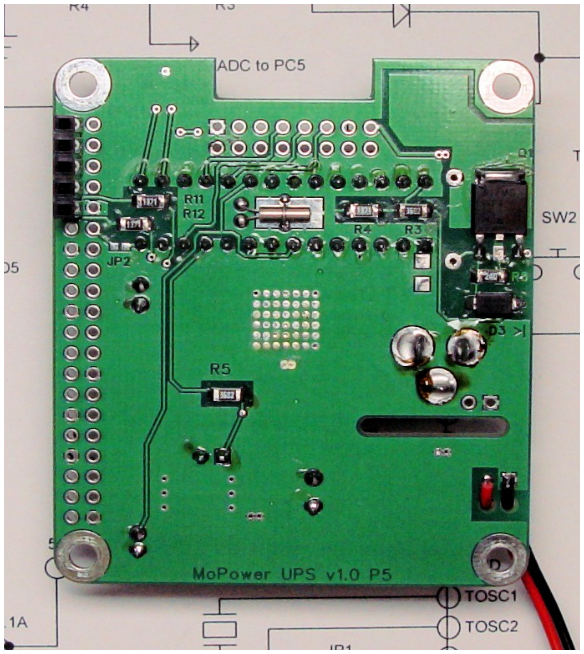
Warnings and Precautions

It is important that you take the time to read this section thoroughly before assembly and use of this kit. We cannot cover all use cases for this UPS, and the ultimate safe use of this UPS is the users responsibility. Nevertheless here are some important points you should be aware of when using this kit.

- You should **NEVER** use the USB power supply input on the Raspberry Pi when using this kit!!! Possible problems include back-feeding the power from the UPS into the wall adapter of the USB supply, this may cause your USB adapter to overheat and may be a fire hazard. While some adapters MAY have protection for back-feeding, this is generally not advertised and cannot be easily confirmed. Our recommendation is to NEVER use USB power input while using the UPS. The user can assure no USB power is used by removing the input poly-fuse and/or physically blocking or removing the USB input jack. Back-feeding of power may also be an issue with USB hubs. Ensure you know the sources and paths of power flow through your setup!
- Batteries are dense sources of energy and must be handled properly. Users should be familiar with proper battery handing/use and general electronics safety. Improper use can pose safety and fire hazards. Among many precautions include; designs that prevent short circuits, ensuring proper condition of wires, connectors, batteries and battery holders. Observe proper temperature ranges for your batteries. As with almost all battery types, enclosures should be properly vented.
- The UPS contains static sensitive components, you should observe proper static precautions when assembling, handling and using the UPS.
- This product is provided in kit form for assembly and use by an experienced user.
- This product has not been tested against and may or may not meet various regulatory, emissions and safety regulations in your country. It is provided in kit form for assembly and use by those experienced in the components and general design of the product. Safe use of this item is ultimately the users responsibility.
- We make an effort to provide components which are less hazardous to the user and environment. Most all of our components are ROHS compliant and lead free, and we will attempt to provide the least hazardous components commonly available. Some components may nevertheless contain lead or not be ROHS compliant (or conform to other rules or regulations). Items may also contain other harmful chemicals or materials, including lead, normally found in electronics components. Please handle/use this kit appropriately and recycle your batteries, as they are almost always harmful if thrown away.

Assembly





Recommended tools and supplies:

- A temperature controlled soldering iron with fine tip, medium power 30-50 watts
- Good quality non-corrosive electronics grade solder, rosin core 0.03" diameter recommended
- Pair of good tweezers for placement of SMD components
- A digital multimeter
- A lab power supply with current reading or multimeter capable of current measurement
- Small diagonal cutters, for cutting component leads
- Magnifying lens if needed.
- Screwdriver and small pliers for attachment of spacers and hardware

Before you begin:

- Semiconductors are used in this kit and they are sensitive to electrostatic discharge. Please use proper tools and handing techniques when assembling and using this product.
- You should be familiar with soldering techniques used to hand assemble SMD components
- Do not apply excessive heat when soldering components. A few seconds of heat application should be all that is needed for smaller components, slightly longer for the larger regulators, etc. Excessive or high heat will damage components or reduce their lifespan. Excessive heat may also damage or destroy PC board pads and traces.
- Components should be properly aligned to prevent shorts to adjacent components or traces.

Assembly

- Install the polyfuses, F1, F2 and F3, they are marked with "LF", "110" and "33"
- The TVS diode is optional but recommended, especially for automotive environments. Please see the TVS diode and Input voltage notes in the "Rating Notes and Recommended Limits" section for more information. Optionally Install the TVS diode, its polarity is not important. Be careful not to confuse this part with the other diodes in the kit, the TVS has unique markings and there is only one included in the kit. The most prominent markings on the TVS are a large "L" and a slightly smaller "F", the other diodes in the kit are marked differently.
- Install diodes, D1, D2, D3, D4, they are all of the same type and marked "B340A". WATCH POLARITY! The cathode of the diode is marked with a stripe and the cathode on the PCB has a notch, these markings should be matched. D3 is on the bottom of the board and its cathode (stripe) should be towards the edge of the PC board. BE CAREFUL not to confuse the TVS diode with one of the regular diodes.
- Install resistors:
  - R1, R3 (bottom) , R5 (bottom), 36K resistors marked 3602
  - R6 a 3.16K resistor marked 3161
  - R7 a 1K resistor marked 1001
  - R8 (bottom), depending on current desired for battery charger circuit
    - 24 ohm for 52ma marked 240, recommended for AA NIMH batteries
    - 200 ohm for 6.25ma marked 201, recommended for 9V NIMH batteries
    - NOTE: R14 on the PCB is provided in case the user wises to use their own through-hole resistor to set a different charge current.
  - Top of PCB: R2, R9, R10. On the bottom: R4, R11, R12, 1370 ohm resistors marked 1371
- Install Q2 and C3. Q2 is the 3.3v regulator marked 50-33G, do not confuse it with the LM317 regulator Q1. C3 is the smallest of the capacitors at 10uf, 35v. Save some lead clippings from the capacitors for use later to mount the crystal.
- Install the DC jack Vinj
- Now we can test the 3.3v regulator. Apply power to the UPS and measure pin #7 on U1 (the 28pin dip) and the 3rd (right most pin) of Q2. You should have very close to 3.3v at these pins. Remove power once this is confirmed.
- Install Q1 (bottom), an LM317 regulator, marked 317MG.
- Install the 9v snap connectors wiring (or a suitable connector) to the BATT terminal on the lower left hand corner of the board. WATCH FOR CORRECT POLARITY!!! For wire connections you can loop the wire through the holes found just below the BATT terminal for strain relief.
- Now we can test the LM317 charger circuit. It is highly recommended for this test you use a lab power supply or a current meter in line to test power draw, without proper current measurements it will not be possible to detect proper operation of the charger circuit. First connect your power supply to the UPS, you should measure very little power draw, on the order of a few milliamps, this is the idle current draw of the LM317 and 3.3v regulator. Next connect your battery to the

UPS (either the 9v snap, or other connector you have installed). MAKE SURE TO OBSERVE CORRECT POLARITY, as there is no immediate indication of a reverse connection, you must insure the batteries are properly connected. Once the batteries are connected you will see an increase in current of approx the same rate as your charging circuit (set by R8 of the LM317). Remove the battery and the current should drop to idle level. Remove the power and the battery from the board.

- Install the connector on P1 (40 pin double row going to the Raspberry Pi). Note that only 5 pins are needed to interface with the Pi, hence a small 5 pin connector is provided. You can also use optional full pins, or pass-through type pins, if needed. If using the provided 5 pin connector, install it on the upper right side of P1. These correspond to pins 2,4,6,8 and 10 on the Pi. These 5 pins also have an outline on the silkscreen layer on the UPS PCB. We solder this connector now because if you use a pass-through connector with long tails, placement of L1 will make soldering of this connector difficult. It is recommended to temporarily fit the connector and UPS onto the Raspberry Pi, and fasten the hex spacers to both the UPS and the Pi before soldering the connector. There is sufficient play in the P1 connector and mounting hardware that this step is recommended for proper fit and alignment. Remove the UPS from your Pi when finished.
- Install Q3 (the large LM2596 regulator chip), L1, C1 (150uf 35v) and C2 (150uf 16v).

  WATCH THE POLARITY OF C1 and C2!
- Now we can test the 5v output. Power up the UPS and measure pin 2 and 4 on the P1 connector, you should measure approx 5.1 to 5.2 volts. Power usage should be very minimal, approx 10ma Remove power once you have completed the test.
- Install the 32khz watch crystal to the bottom of the board. Handle the crystal carefully, do not drop it. Do not make sharp bends of the crystals leads close to the base of the crystal. Make sure the crystal is seated in the tinned copper area under U1 and is secured and soldered by using lead trimmings saved previously. Solder the leads only, and not the body of the crystal, as heat damage may occur. Because of the micro-controller, make sure to trim leads to minimal length on the top of the board. Make sure the crystal leads do not short to the ground plane.
- Install a socket onto U1 and install the microcontroller, or install the microcontroller directly. Watch the polarity of the microntroller when soldering or installing onto the socket.
- Install the LEDs. There are two LEDs, the status led “STAT” and power led marked “PWR” They are above resistors R9 and R10. The red power led is taller then the amber status LED. The cathodes of both LEDs are marked with a green dot or stripe. The PCB is marked with a small C to denote the cathodes of both LEDs. Both LED cathodes face each other to make assembly orientation easier.
- Install switches SW1 and SW2
- Optionally install P2 (not normally provided)
- Solder the 2 pin header/connector on JP1 and install the provider jumper on this connector. Install the connector on one pin only unless needed, as shorting JP1 will prevent the UPS from controlling the power to the Raspberry Pi.

Using the UPS

Attach the UPS to your Raspberry Pi, you can use the included hex spacers and hardware. Ensure that the bottom of the UPS board will not short to the Pi. Connect power to the UPS board, your Pi should power up. Connect the battery to the UPS observing proper polarity. Make sure the batteries are secured to prevent a short.

NOTE: Some rechargeable AA batteries are a very snug fit in the battery holder.

The negative spring may not have sufficient force to push the battery into contacting the positive lead. Some initial pressure or movement may be needed to ensure that they make proper contact with the terminals.

Install the appropriate software to communicate with the UPS, see the software section.

Usage of jumper JP1. This jumper is provided to override power control of the UPS. The UPS shuts off power to the Pi for various reasons; watchdog timer reset, timed power off, low battery when power has failed, etc.. when this jumper is installed it will prevent the UPS from powering off the Pi. This can be used when working on your Pi if you do not want the UPS to shut off power inadvertently, such as during a firmware update. This jumper must be removed for normal operation otherwise the UPS will not be able to control power to the Pi.

Use of the power switch SW1. This switch can be used to reboot or shutdown your Raspberry Pi. Pressing it briefly (0.06 to 1.5 seconds) will instruct your Pi to reboot. Pressing it for 2 to 5 seconds will instruct your Pi to shutdown. Pressing it for 10 seconds or longer will forcibly cut power to your Pi. Note that for reboot and shutdown to work properly, your Raspberry Pi must have the appropriate software installed to communicate with the UPS and initiate a reboot or shutdown. The forced power cut actually shuts down the power regulator feeding the Pi and no software is required. When the Pi is off, a short press of this button will apply power to the Pi.

Please see the “Serial Communications Protocol” section for detailed information on configuring your UPS.

Raspberry Pi Software

Software is provided for the Raspbian Operating system to control basic UPS functions. Other scripts and software may be provided in the future, please check the link below for current information

Software can be downloaded from http://www.allspectrum.com/mopower/

Parts List

PCB Marking	Name	Notes
P1	Connector	5 pin provided
L1	Inductor 33uh SMD	Marked "330"
U1	Micro Controller	ATMEGA328P-PU
Q1	LM317 regulator	marked "317MG"
Q2	3.3v regulator	marked "50-33G"
Q3	LM2596	5v regulator
R1, R3, R5	SMD resistor	36K resistors marked 3602
R6	SMD resistor	3.16K resistor marked 3161
R7	SMD resistor	1K resistor marked 1001
R8	SMD resistor	24 ohm marked 240, for 52ma
R8-secondary	SMD resistor	200 ohm marked 201, for 6.25ma
R2, R4, R11, R12, R9, R10	SMD resistor	1370 ohm resistors marked 1371
PWR	POWER LED	Red LED, taller of the two provided
STAT	STATUS LED	Amber led
--	9V snap	
F1, F2, F3	Polyfuses	marked with “LF”, “110” and “33”
D1-D4	Diodes B340A	marked “B340A”
C1	MAIN IN CAP	150uf 35v
C2	SMPS OUT CAP	150uf 16v
C3	10uf	10uf 35v
---	32 khz crystal	* FRAGILE *
---	Mounting Hardware	
TVS	600W SMB	
SW1, SW2	Switches	
Vinj	DC Vin Jack	2.1mm Center Positive
---	Jumper	
JP1	Jumper male PCB pins	

Serial Communications Protocol

The MoPower UPS uses a serial UART at 19200 baud to communicate with the Raspberry Pi direct through its onboard UART. The official Raspbian OS uses /dev/ttyAMA0 for this port. You must disable the serial terminal that Raspbian normally has active on this port. See the UPS software install instructions for how to disable the serial terminal. You can also install the serial program minicom for configuring the UPS.

The UPS acts similar to a shell prompt and settings are printed and controlled like variables.

For example the variable name for the Status LED is "LED_STAT"... so to display the status of this LED we type.

```
LED_STAT
```

In the terminal program. We get a response of "LED_STAT=0" which indicates that the status LED is currently off (0). We can modify the LED by using the assignment operator "=". To turn on the LED..

```
LED_STAT=1
```

To turn it off, set the value to 0. You can do this with most of the options provided by the UPS. Variables/Settings which are READ ONLY are marked with a preceding * before the variable name.

***IMPORTANT NOTE* We have tried to keep the UPS firmware as flexible as possible, allowing many different operating modes via the numerous settings. If you are changing numerous settings be mindful of how they will effect the operation of the UPS, some may not behave in the way you initially intended, or various settings may conflict with each other.**

Table of common commands/variables/settings

COMMAND	DESCRIPTION
show all	Prints all common settings
show time	Prints all common time settings
show ups	Prints all common UPS settings
show data	Prints selected data variables w/CRC check
EEPROM RESET	Resets all common EEPROM variables
EEPROM SAVE	Saves all common settings to EEPROM
UPS-REBOOT	Reboots UPS

“show data” command and CRC check

The “show data” command is primarily for use by automated software to quickly retrieve the most commonly used data variables. The command returns the variables in a space delimited string followed by a CRC value. It is recommended that any automated software that reacts to output from the UPS to use this command and check the CRC value before acting on the output. The CRC value is in common XMODEM format and has a Polynomial of 0x1021

An example of returned data is below..

“0.91 11 23 2015 16 30 24 2 4811 1 1 0 0 1 12.61 8.53 4.08 267 247 209 228 300 7e34”

In this case the CRC value is “7e34” and is computed on the entire string EXCEPT for the SPACE before the CRC value and the CRC value itself.

Watchdog Timer

The UPS has a watchdog feature which can be used to reset your Raspberry Pi in case it is unresponsive.

When enabled the watchdog timer will keep counting down and is reset by two events, one is any command received by the serial UART from the Pi and second is an actual RESET event caused by the watchdog timer itself. The watchdog considers any command, valid, invalid or blank (a RETURN), coming from the UART as a signal that the Pi is active and will reset its timer.

OPTION	**DESCRIPTION**	**DEFAULT**
SETTINGS_WATCHDOGTIMER_ENABLE	0=DISABLE, 1=ENABLE TIMER	0=OFF
SETTINGS_WATCHDOGTIMER	This is the main watchdog “timer” in SECONDS. If it doesn't receive a valid signal from the Pi in this time, a hard reset is done. Range: 0-65,535	300
SETTINGS_WATCHDOGTIMER_OFFTIME	Once a watchdog reset is triggered, this is the time to stay in the OFF state before reapplying power to the Pi Range: 0-65,535	10
*WATCHDOGTIMER	*Read Only, This is the watchdog countdown	NA
*WATCHDOGTIMER_OFFTIME	*Read Only, This is the watchdog countdown for power off time	NA

UPS Input Control Settings

The UPS Input Control Settings allow a way to control the Pi (or the UPS I/O) depending on various inputs available to the UPS.

The following two controls are a good example of using the input control for a 12v automotive application. By selecting appropriate power up and shutdown voltages, we can make our Pi start and shut down depending on if our engine is running. This saves us from having to run an ignition wire to our Pi and also gives us more flexibility, for example we can set the TIME AT LEVEL option to allow the Pi to stay on for a predetermined time after engine shutdown.

Continuing our automotive example, here are two settings that will accomplish this.

Example, Turns on Pi when input voltage is greater then 13.50 volts for more then 3 seconds

```
INPUT_CONTROL[1]=2,1,20,13.50,3,1,0,2,1
```

Example, Turns off the Pi when input voltage is less then 12.80 volts for more then 10 seconds

```
INPUT_CONTROL[2]=3,3,20,12.80,10,0,0,1,1
```

See the UPS Input Control Settings table for more information. The voltage levels chosen should work for most 12v systems and can easily be adjusted. You may have noticed the last two options, Option 8 and 9 are set. Option 9 “USE TRIGGER” when set only triggers our control setting ONCE, it then needs to be reset by an external source to be triggered again. Our external source in this case ends up being the other command. The trigger options allow our automotive car computer to be turned on with the engine off, or turned off with the engine on. If these triggers were NOT set, once you turned off a Pi (via software or the power button) when the engine was running, the power on input control would immediately turn it back on again. In the opposite case when the engine was off, you would not be able to turn ON the Pi, since the power-off command would persist. This is generally not what is expected to happen in this example automotive application, hence the need for the trigger options.

In our example above, the power-off command resets the trigger of the power-on command and vice-versa. You can setup another control setting to act as a trigger reset as well, by setting COMMAND option to be 0 and the appropriate trigger option, you would only be resetting the trigger of the other command and not executing any power functions.

Table of UPS Input Control Settings

OPTION	DESCRIPTION
INPUT_CONTROL[1-10]=	Number in BRACKETS specifies the Input control index, 1-10
Option 1 – PRIORITY	The Priority of this control, 1-10. A higher priority level will override other control settings acting on the same command. Timer control will have priority over input control functions of the same priority level.
Option 2 – COMMAND	The Command to execute, 0=None 1=ON 2=REBOOT 3=SHUTDOWN
Option 3 – PORT	1=VIN, 2=VBATT.. see PORT OPTIONS table for further settings
Option 4 – LEVEL	Test level, float value
Option 5 – TIME AT LEVEL	Consecutive Seconds where test level has to match
Option 6 – TEST	0=Less then, 1=Greater then, 2=Equal to
Option 7 – Trigger	Holds trigger variable, 0=not triggered, 1=triggered
Option 8 – RESET TRIGGER	Index of other command to reset trigger of
Option 9 – USE TRIGGER	Use (1) or don't use (0) our own trigger

Table of PORT OPTIONS for Input Control

PORT OPTION	DESCRIPTION
1	Volts Input, float value
2	Volts Battery, float value
10	ADC input P2 PIN10, Range: 0-1023
12	ADC input P2 PIN12, Range: 0-1023
14	ADC input P2 PIN14, Range: 0-1023
16	ADC input P2 PIN16, Range: 0-1023
50	Switch 2, 0=switch pressed, non-zero otherwise
53	Input level P2 PIN3
54	Input level P2 PIN4
55	Input level P2 PIN5
57	Input level P2 PIN7
58	Input level P2 PIN8
59	Input level P2 PIN9
60	Input level P2 PIN10
61	Input level P2 PIN11
62	Input level P2 PIN12
64	Input level P2 PIN14
66	Input level P2 PIN16

UPS Timer Control Settings

The UPS provides a way to wake up a powered off Raspberry Pi at intervals using the timer feature.

It is accessed using the “TIMER_CONTROL” variable. There are 10 individual timers available for use, they are accessed like a variable array. For example to set the first timer control option one can use..

```
TIMER_CONTROL[1]=5,1,30,24
```

This sets timer # 1

- with a priority of 5
- a command of 1 (which is the command to turn on the Pi),
- The minute to run at, set to 30 minutes
- Set to run at all HOURs (24 is a special variable that matches all HOURs)

With this setting the Pi will power up every hour at half passed the hour.

The options are comma separated and there should be no spaces, options should not be skipped.

You can use the other 9 indexes as well, 1-10 is available for use. Remember to issue a EEPROM SAVE command otherwise you will loose any set options if the UPS reboots.

Table of UPS Timer Control Settings

OPTION	**DESCRIPTION**
TIMER_CONTROL[1-10]=	Number in BRACKETS specifies the Timer control index, 1-10
Option 1 – PRIORITY	The Priority of this timer control
Option 2 – COMMAND	The Command to execute
Option 3 – MINUTE	The minute to run the command, 0-59, 60 is wildcard 101-130 = step values, similar to Unix 'cron' For example 105 would equal to */5 in 'cron' and would run every 5 minutes, 130 would be equal to */30, etc..
Option 4 – HOUR	The hour to run the command, 0-23, 24 is wildcard 101-112 = step values, similar to Unix 'cron' For example 102 would equal to */2 in 'cron' and would run every 2 hours, 108 would be equal to */8, etc..

Time and Date – Daylight Savings Time

The UPS is capable of making adjustments for Daylight Savings Time to its own internal clock.

While we believe most use cases would involve the Pi correcting both its own and UPS time, this feature is provided in case the Pi will be off for periods of time that may transition though a Daylight Savings Time adjustment event.

OPTION	DESCRIPTION	DEFAULT
SETTINGS_DLS_ON	Daylight Savings On or Off 1 = OFF, 1 = ON	0 = OFF
SETTINGS_DLS_STATUS	Used to indicate if Daylight Saving Time is currently in effect (Summer time), or not in effect (Winter time). It must be set to the correct current value for Daylight Saving Time correction to operate properly. 0= DST 1 = NOT DST	0 = NOT DAYLIGHT SAVINGS TIME
SETTINGS_DLS_SPRING_HOUR	Specifies the hour (in 24 hour mode) that Daylight Saving Time takes effect during the Spring and the clock is moved forward 1 hour Range: 0-23	2 = 2 AM
SETTINGS_DLS_SPRING_DAY	The day that Daylight Savings Time takes effect. If SETTINGS_DLS_SPRING_WEEK is set to 0, this option refers to the day of the month in which the adjustment takes place. If SETTINGS_DLS_SPRING_WEEK is set to any value other then 0, this option refers to the Day of the Week in which the correction takes place (ie, 1=Sunday,2= Monday, 3=Tuesday...). If this option refers to the day of the week, you must make sure the current day of the week (DAYOFWEEK) is properly set. Range: 1-31	1 = SUNDAY
SETTINGS_DLS_SPRING_WEEK	Specifies the week in which the adjustment is made. 0 and 5 are special cases. 0 is used where we do not want to refer to a particular week, but to a particular day of the month in which we want the adjustment made (refer to SETTINGS_DLS_SPRING_DAY). A value of 5 also has special meaning in that it refers to the last week of the month. Range: 0-5	2 = SECOND WEEK
SETTINGS_DLS_SPRING_MONTH	Specifies the Month in which the Spring time adjustment is made. Range: 1-12	3=March
SETTINGS_DLS_FALL_HOUR	See above	2 = 2AM
SETTINGS_DLS_FALL_DAY	See above	1 = SUNDAY
SETTINGS_DLS_FALL_WEEK	See above	1 = FIRST WEEK
SETTINGS_DLS_FALL_MONTH	See above	11=NOVEMBER
DAYOFWEEK	Current day of the week. Required for proper operation of DST if the day option is used to specify a weekday. Range 1-7, 1 = SUNDAY	5=Thursday

Time Correction Settings – Watch Crystal Adjustment

This setting allows one to 'software trim' the clock crystal responsible for keeping time on the UPS. Trimming of the watch crystal is generally not required if you have an accurate network time source available and you will be updating the UPS time accordingly. If you require accurate time keeping without a network connection or other time source you can trim this value if needed. Take accurate long term measurements of the UPS clock before changing the default value.

OPTION	DESCRIPTION	DEFAULT
SETTINGS_TC_SEC	Error correction to be applied to the clock crystal. Value must be a whole number and is calculated by using 604800 / (seconds per week correction). Unsigned 32-bit value	235331, which corresponds to a value of (-) 2.57 seconds per week of correction. *DEFAULT SUBJECT TO CHANGE WITHOUT NOTICE AND MAY NOT RESULT IN MOST ACCURATE CLOCK*
SETTINGS_TC_SEC_POS	Indicates if time correction value is positive or negative. 0=NEGATIVE 1=POSITIVE	0 = NEGATIVE

Debug and Special Use Options

These options are rarely used and should not be normally adjusted. They are NOT reset by the normal EEPROM reset commands of the main firmware. It is recommended that you NOT program these options into any automated scripts and execute them sparingly.

Boot power on and brownout reset flags

On power up the micro-controller on the UPS will normally leave the 5v power to the Pi untouched EXCEPT when an initial power-on or brownout condition occurs. Remember that the UPS micro-controller will run with lower voltage then the 5v converter.. and hence will run at voltages that would not power up the 5v line to the Pi. When such a condition is detected the UPS ***bootloader*** will TURN OFF the 5v power and allow the main UPS program to decide if power should be turned on or off. This prevents the errant power up and then power down that would normally occur if we did not detect these two events. You may still see a very momentary blip of power on initial power up, this is a result of the very small bootup time of the micro-controller and has not been seen to cause any problems with the Pi.

Again this only occurs when initial power is applied to the UPS or when a brownout event occurs. It should also be noted that the micro-controller in use almost always detects both a power on and brownout AT THE SAME TIME, so setting one or the other in general will give you the same effect, see the micro-controller datasheet for further information.

These options are provided in case you need to turn off this functionality. Both options are ENABLED by default. Again, you will normally never need to adjust these values.

– To disable power on reset flag check:	BOOTPORFCHECK DISABLE
– To enable power on reset flag check:	BOOTPORFCHECK ENABLE
– To check power on reset flag state:	BOOTPORFCHECK STATUS
– 255 is ENABLED, 0 IS DISABLED	
– To disable brownout reset flag check:	BOOTBORFCHECK DISABLE
– To enable brownout on reset flag check:	BOOTBORFCHECK ENABLE
– To check brownout on reset flag state:	BOOTBORFCHECK STATUS
– 255 is ENABLED, 0 IS DISABLED	

Micro-controller Oscillator Calibration for UART speed adjustment

The micro-controller and Raspberry Pi both must keep an accurate clock source to properly communicate with each other via the serial UART. The speed of the UART that we use (19200) is slow enough where we can tolerate a fair bit of error. While there is always the option to automatically adjust the micro-controller oscillator, we have elected not to provide this auto-adjust feature. We have found in our testing that the factory calibration of the internal oscillator of the micro-controller to be very good, due to this experience we believe providing an auto adjustment feature would needlessly complicate things and add yet another troubleshooting step when problems do arise. There is also the UART clock source on the Raspberry Pi to consider, it is not clear as to the initial accuracy of this source, having an auto-adjust on our micro-controller might just make things worse.

With this in mind, we realize there may be cases where you would need to adjust the oscillator frequency. For example a very large temperature swing may cause clock problems, or we may have a combination of Raspberry Pi and UPS micro-controllers that are just too far out of alignment to initially communicate. Thus, we have provided a method to test and modify the main oscillator frequency of the micro-controller on the UPS.

This is another one of those options that are rarely used and should not be normally adjusted. They are NOT reset by the normal EEPROM reset commands of the main firmware.

Note your UPS may have a different calibration other then default

- To display current oscillator calibration value: EEPROM_OSCCAL
- To set oscillator calibration value: EEPROM_OSCCAL=VALUE
  - Value is 0 to 255 255 is our default setting meaning no adjustment is done!
- Special command to print all oscillator calibration values. You can issue command “RUNALLOSCCAL” to run an oscillator test routine. This will loop the UPS through most of the oscillator calibration values (0-254, 255 IS DEFAULT NO ADJUST!) and echo that value on the screen. This is a good test to run to see what oscillator values will work, since if the value echos on our screen it was transferred successfully to the Pi via the UART. You will see garbage on your screen for values that are out of spec. We generally pick a value in the middle of oscillator values that print successfully on the screen. Note that there are TWO ranges in the adjustment range of 0-254, a low and high range. Please refer to the micro-controller data sheet for further information on the OSCCAL value.

NOTE that you can also adjust the clock speed of the UART on the Raspberry Pi side. If you cannot communicate with the UPS, you will have to use the UART adjustment on the Pi in order to communicate, initially set it up, or to adjust the oscillator calibration on the UPS. This may be needed in cases where either the Raspberry Pi clock and/or the UPS clock is too far out of adjustment.

You can do this by setting “bcm2708.uart_clock=3000000” in /boot/cmdline.txt and setting “init_uart_clock set=3000000” in /boot/config.txt

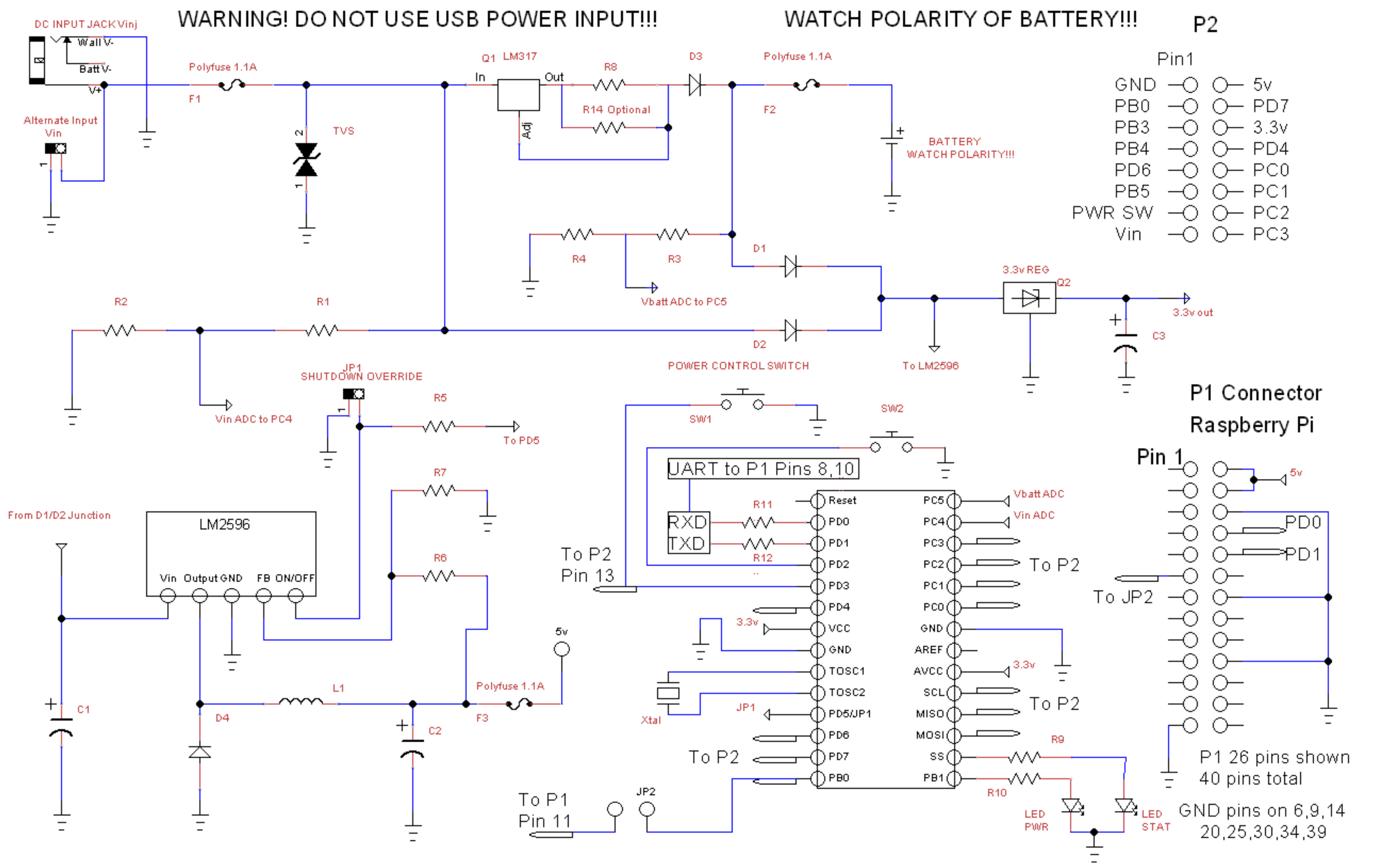
Adjusting init_uart_clock set in increments of 50000 should get you talking to the UPS. There are reports that some versions of the Raspberry Pi have trouble setting the uart clock or have different syntax to do so, the commands above have been tested to work with Raspbian Wheezy dated 2015-05-05. Check syntax on your version if you are running a different operating system.

Rating Notes and Recommended Limits

Input Voltage Recommended: 12-20 VDC, a regulated or non-regulated 12v adapter rated at 500ma to 1A is recommended for most applications. Input voltage up to 30v is possible with removal of the TVS Diode.

Recommended operating temperature: 0c – 45 c / 32 – 113 f

- **TVS diode**: Please note that the TVS diode is intended to suppress very brief TRANSIENT voltages only. Constant over voltage will damage the TVS and other components and may present a hazard. If the TVS diode is installed the maximum input voltage should not exceed 20v
- **Input voltage low range**: The low range of the input voltage is primarily dictated by the voltage required to charge the batteries fully. With 6 AA NIMH batteries this is approximately 9.0 volts, or 1.5 volts per cell times the number of cells. Some 9v NIMH batteries have 7 cells, so the approximate voltage would be 10.5. To this voltage you will need to add approximately 2 volts to account for the voltage drop in the LM317 current regulator. For example 6AA would need 9 volts + 2 volts or approx 11 volts. A 7 cell NIMH block battery would need 10.5 + 2 = 12.5 volts.
- **Input voltage high range**: The high range limit has several considerations.
  - If the TVS diode is installed the maximum input voltage should not exceed 20v. The TVS diode is intended for operation on 12v/14v electrical systems, it will start conducting above approx 20v.
  - The next lowest voltage tolerant semiconductor device is the 3.3v regulator. It has a absolute maximum voltage rating of 30v. We have ran the device over short periods (several days) at 28v (REMEMBER TO REMOVE THE TVS DIODE AT THIS VOLTAGE!) without noticeable ill effects, and should be reliable up to the manufacturers rated 30v.
  - Another consideration is the power dissipation of the LM317 current regulator. It is setup as a constant current device and is a linear regulator, it must dissipate all power above the voltage level of the battery. For example a fully charged 6xAA battery pack will be held at 9.0 volts with a charge of 50ma. If your input is 14v, 5 volts at 50ma will have to be dissipated by the regulator, this is 5 x 0.050, or approx 0.25 watts. If it were ran at 28 volts, it would dissipate 0.95 watts, and may overheat. You can use a 9V block battery at high voltage levels to reduce power dissipation in the LM317, the recommended current level for a 9V block battery is set at 6.25ma and thus the regulator needs to dissipate a lot less power.
  - The DC power jack also has a voltage/current rating, the one supplied with the kit is rated at 24v/5a. While DC jacks of this type/design can generally accept a higher voltage level it is recommended to stay within the manufacturers rating. Wires can be attached to power input pads/holes for higher voltage levels.
- **Recommended temperatures**: Batteries behave differently at low and high temperature, capacity will diminish at temperature extremes for most batteries. For NIMH batteries most manufacturers recommend less then a 0.1C charge level below 0 degrees C. Although our UPS at recommended charge levels is well below this limit, we recommend a minimum operating temperature of 0 degrees C. Most semiconductors also have recommend minimum temperature of 0 degrees C for proper operation.
- Manufacturers of NIMH batteries generally recommend a maximum charge temperature of 40-45 degree C. NIMH batteries will also have diminished charge acceptance at high temperatures. The power dissipation in the LM317 may also be a consideration at higher temperatures. It is recommended that you operate the UPS at not more then 45 degrees C.



- WARNING!!! DO NOT USE USB POWER INPUT, RECOMMENDED TO REMOVE USB INPUT POLYFUSE FROM THE RASPBERRY PI!!!

- JP1 can be installed to override UPS control of 5v power (always on when installed)

- JP2 for possible future use

Copyright © 2014-2015 All Spectrum Electronics

Title MoPower UPS, designed for the Raspberry Pi		
Author Copyright © 2014, 2015 All Spectrum Electronics		
File Mopower-UPS-schematic-v1.0-P5-12-13-2015.dsn		Document
Revision v1.00 P5	Date 12/13/2015	Sheets 1 of 1