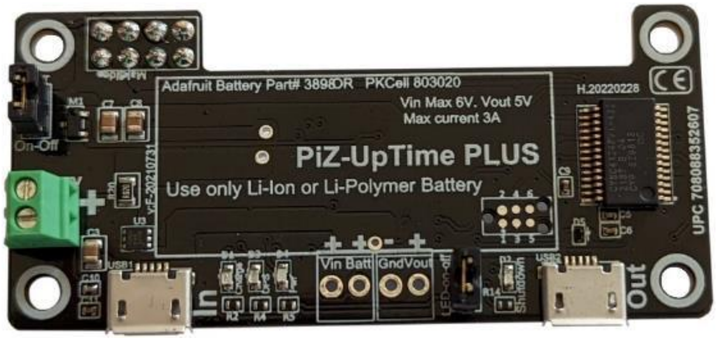
PiZ-UpTime PLUS

UPS guarantees reboot after shutdown

Key Features:

- Guarantees reboot after shutdown and when power returns.
- Several operating modes – operates as UPS; shuts down on power failure; reboot immediately when power returns or wait after power returns – all controlled by a Python script.
- Very configurable - all parameters are controlled using the Python script. Sample script provided.
- Several operating parameters are monitored: Input V, Output V, Battery V, Input current, Battery current and board temperature.
- Uses Li-Ion or Li-Polymer batteries. *Batteries not included.*
- Provides UPS functionality for all Pi models with a 40 pin header, including Pi-4.
- Provision for on-off switch, external power in / out / external battery connections. LED's are turned also on/off via a jumper.



To reboot a Raspberry Pi after a software shutdown, the input power needs to be toggled. This is done by pulling the USB power source out or installing an on/off switch. This becomes an issue for remotely installed Raspberry Pi's where the cost of a "truck roll" to reboot a Pi is expensive. Pi's are also used in situations where the power is on/off many times as part of normal use. Such use-cases are, for example, using a Pi with solar panels or other green energy sources; using a Pi in an automobile; using a Pi to monitor temperature in a freezer or a truck etc. Using a GPIO, a simplistic approach is to issue a shutdown command when the power fails (assuming a UPS is connected to the Pi). In that case, when the power returns, the Pi will have to be manually rebooted. Another approach is to keep the Pi running till the battery drains out (hard failure of the UPS). Any hard failure will damage the SSD card sooner or later.

PiZ-UpTime PLUS supplies intelligent UPS capability in a Raspberry Pi-Zero-form-factor. It is designed to be used with regular sized Raspberry Pi's e.g., Pi-3 and Pi-4. To operate a Pi-Z-UpTime PLUS, connect a battery and start the provided Python script. The connector (2x4 pin connector) on PiZ-UpTime PLUS follows the 40 pin Raspberry Pi header format. Besides powering a Raspberry Pi, PiZ-UpTime PLUS also powers other SBC's following the Raspberry Pi header format. Alternately, by using the USB Power Out port other devices take advantage of the PiZ-UpTime PLUS capabilities.

All information subject to change without notice

All operating parameters are customized using the sample Python script. These are communicated from the Pi to the PiZ-UpTime PLUS via I^{2}C, usually during the reboot. Different operating modes supported are:

1. Run PiZ-UpTime PLUS as a UPS i.e., supply power till the battery runs low. After the battery runs low, the shutdown command is run. The battery level, which triggers the shutdown command, is configured from the Python script.
2. Start the shutdown process immediately on power failure. PiZ-UpTime PLUS run as a UPS till the Pi shuts down. The time PiZ-UpTime PLUS runs as a UPS is configurable from the Python script. After the UPS time is complete, power to the Pi is cut off.

After the shutdown is complete, a decision needs to be made as to how to reboot the Pi. The different modes supported are:

3. Reboot the Pi as soon as power returns.
4. Wait for a specified duration after power returns before supplying power to the Pi. This duration is configured via the Python script. This delay allows for power to stabilize when it returns from a power failure. Such use cases are recommended when there are inductive loads, for example, starting an automobile, starting motors (freezers, refrigerators, water pumps etc.).

These modes are mixed and matched to suit your individual needs.

An on-board micro controller on PiZ-UpTime PLUS monitors the input power (on the USB port) and makes the decision as to when or if the power should be toggled. Besides watching the power status, the microcontroller reports back the values of input Voltage (V_{in}), Output Voltage (V_{out}), Battery Voltage (V_{batt}), Battery Current ($I_{battery}$), Input current (I_{in}) and on-board temperature (Temp). The battery current shows charging current or discharging current when operating as a UPS. All measurements are done by a 12-bit ADC built into the micro controller, and the values are reported back via I^{2}C.

A separate Battery Charge Management (BCM) circuit manages the battery charge as well as boosting the power from the battery to 5V, supplying the necessary power (current) for running the Pi. Since the battery is charged at the same time the Pi is running, please make sure the USB power source has enough capacity (amps) for the Pi and the battery charging. A USB charger rated for a minimum of 230 Watts is recommended with PiZ-UpTime PLUS with a Raspberry Pi-4.

Battery charging is usually safe between 0^{o}C and 50^{o}C[1]. An onboard thermistor monitors the temperature on the underside of the PiZ-UpTime PLUS and reports that to the BCM as well as via the Python Script. Ideally, the battery is placed close to the board so that the battery is safely recharged. If the battery is placed further away, the Python script is used to compensate for that distance. The Battery Management System (BMS) has many safety features. These includes temperature, over current protection, over charging the battery, keeping the battery charged, slow charging a deeply discharged battery and others. Maximum charging current provided is 2A. Please ensure that the Lithium-Ion battery is capable of handling the charging current.

No GPIO connection are used by PiZ-UpTime PLUS, except the I^{2}C connections defined by the Raspberry Pi headers, which are also used as GPIO's.

The Python Script is well documented with supporting comments showing use of the variable. For example, low battery threshold value is changed from a default of 3.1V to any other value in the Python

[1] The temperature will vary from one manufacturer to another. Please check with the manufacture specifications.

script. Should a use-case require the Pi to be rebooted when the battery is down to 3.8V, that is done via the sample Python Script by changing the low battery threshold V level!

As to how long the UPS keep the power supply on is also controlled from the Python script. This timer value is critical to ensure shutdown completes properly. The default timer value is set to 30 seconds, which is more than sufficient (in most cases) to shut down the Pi.

A supporting document describing these variables and how they are used is also provided.

A Shutdown LED shows the operating status of PiZ-UpTime PLUS.

1. If Green power LED is on or the Blue UPS LED is on and the Shutdown LED is off indicates that PiZ-UpTime PLUS is operating normally.
2. If the Green power LED is off, UPS LED is off and the Shutdown LED is off indicates that the battery has been drained (usually below 2.8V).
3. A blue Shutdown LED blinking on for one second and off for one second, and blinking ½ second on / off for the last 5 seconds for the duration set shows a shutdown in process. After the blinking, the Blue UPS LED may remain on, indicating the UPS is still operating. The UPS power out port continues to supply the 5V to the USB port.
4. The Blue Shutdown LED blinking for 2 seconds on, 2 seconds off shows the PiZ-UpTime PLUS is waiting for power to stabilize. While the PiZ-UpTime PLUS waits for the power to stabilize, the battery charging starts while the power to the Pi is held back for the specified duration.

PiZ-UpTime PLUS Output V is 5V when operating as a UPS. When power from the electrical outlet via the USB port is on, the power is “passed through” to the output and the battery is charged in parallel. Max battery charging current is 2A. Max system current is 3A. The battery charging current is varied by the BCM, prioritizing power to the Pi (pins 2,4). When power fails and the UPS kicks in, the input V may drop momentarily. Raspbian OS may report a low V message when the UPS takes over the Power supply function.

A PTC resettable fuse protects the circuitry against ESD and other power spikes. For example, using the PiZ-UpTime PLUS in an automobile, there could be current spikes when the engine is tuned on. Power spikes usually occur when a Pi is used in conjunction with motors and other electrical equipment e.g., in automobiles, refrigeration units, water pumps, heaters etc.

Key Features

- Different operating modes as described earlier.
- Supplies UPS power using one or more rechargeable Lithium-Ion or Lithium Polymer battery connected via terminal blocks. Batteries are not provided. Recommend using Li-Ion batteries with PiZ-UpTime PLUS.
- Choice of Li-Ion or Li-Polymer batteries. Please ensure Li-Polymer batteries will support 2A charging.
- Stable 5V output, cuts pesky “bolt” on Raspbian with UPS power. Max 3A out, enough for most uses. Note – if a bolt appears with USB power, it means the USB power supply is not capable of providing power to both the Pi and PiZ-UpTime PLUS for charging the battery simultaneously.
- Connection for an on/off switch. Use a 2.54mm header cable (e.g., Amazon Product ID B07FXXDN2M) to connect a switch inline. Please ensure the on-off switch supports 10V/3A or more.
- Includes Battery management system (BMS). Charges the battery when power is available. The features of the BMS are:
    - Ensures battery stays charged over time.

  - Ensures fully discharged batteries are managed properly. The fully discharged battery is trickle charged initially at low amps (appx 200mA) and when the battery reaches a suitable V, the CCCV charging starts.
  - Monitors CCCV battery charging and regulates the necessary Voltages and currents.
  - Disables battery charging when temperature is below 0^{o}C and above 50^{o}C. The temperature is measured using a NTC resistor on the board. Please make sure the battery is in proximity of the board to ensure proper charging.
  - Ensures safety by monitoring output current limit (3A), and battery charging current limit (2A). Should current exceed that, it will protect the circuit by shutting down operations and resuming operations when the offending cause has been removed and temperature drops to normal operating conditions.
- On board microcontroller allows System monitoring via I^{2}C. Sample Python code is supplied and is downloadable for use. The sample code is modified by the user to adapt to any of the operating variables.
- System shuts down when batteries fall below 3.1V or Voltage level set in the Python script.
- Several LEDs indicators:
  - **Yellow or Charge** – LED on shows battery is charging and power is On. LED Off when battery is fully charged. A fast-blinking yellow LED shows charging error. Reset input power (usually plug and unplug the power) to continue charging. LED turns off after charging is complete.
  - **Green or power** – shows input power is on.
  - **Blue or UPS** – shows UPS is operational.
  - **Blue or Shutdown** – blinks 5 times when power is connected for the first time. Also blinks on for 1 second, off for one second for the duration of the timeout for the shutdown, indicating power is going to be turned off after the duration specified in the Python Script. Blinks on for 500 milli seconds and off for 500 milliseconds 5 times at the end indicating power will be turned off in 5 seconds. Shutdown LED also blinks on/off for 2 seconds if PiZ-UpTime PLUS is waiting for power on to stabilize.
- All LED's are turned off or on using the jumpers labelled "LED on/off" (except the Shutdown LED which is normally off). A SPST – normally on switch is recommended to control the LED's or unplug the shunt on the jumper "LED-on-off" to turn LED's off.
- Maximum battery charge current is limited to 2A[2]. Battery charge time will vary based on battery capacity and battery charge level. The BCM supplies priority to the output and will lower the battery charge current appropriately. This may prolong the battery charge time.
- Power to the Pi is provided via the 2x4 pin header, pins 2,4. Ground is via Pin 6.
- Sample Python code and documentation on how to use the Python code is provided.
- Output Voltage – UPS generates regulated 5V (±1% across full load range). Please make sure the battery is capable of sustaining at least 3.5 Amps or more as its discharge current.
- M2.5 x 11mm spacers are used with PiZ-UpTime PLUS to mount it securely on a Raspberry Pi. Four spacers are used with a Pi-Zero. Two spacers are used with the larger form factor Raspberry Pi (e.g., Pi-2, Pi-3, Pi-4). Spacers are not included.

Please visit www.alchemypower.com/products for more information and sample python code.

[2] Some Lithium Polymer batteries maybe damaged by this current. Please ensure that the Lithium Polymer battery is capable of withstanding 2A charge current.

Recommended installation steps

1. Remove Jumper J3 to turn off all power to the Pi and external power sources (USB-out etc.).
    a. Insert an on/off switch if desired at Jumper J3. Ensure the switch is in the off position.

2. Connect the battery to the terminal block (J2) labelled Battery. Make sure you follow the proper polarity. Improper battery insertion will damage the electronics. **When you connect the battery, the Shutdown (blue) LED blinks 5 times, indicating the processor is ready to monitor power status.** This also indicates the battery is connected properly. If the LED-on-off jumper is connected (default), the UPS (blue) LED will be on.

3. Mount the PiZ-UpTime PLUS board on the Pi. Recommend using 11mm spacers to mechanically secure the unit.

4. Plug in the USB power to USB-In port on Pi-UpTime PLUS. The yellow/amber LED may come on, indicating the power is on and the battery is charging.

5. Finally, when ready, insert the shunt on Jumper J3. If an on-off switch is installed, turn the on/off switch to on. The Pi should be booting up once power is applied.

6. Run the sample Python code to ensure the shutdown happens properly when the power fails, and battery runs low. Some key variables to pay attention to are:
    a. Minimum battery V at which the shutdown is triggered. The Python script monitors battery V, and when the battery V falls below this level, shutdown is initiated. The Python variable in the sample script is "V_batt_min"

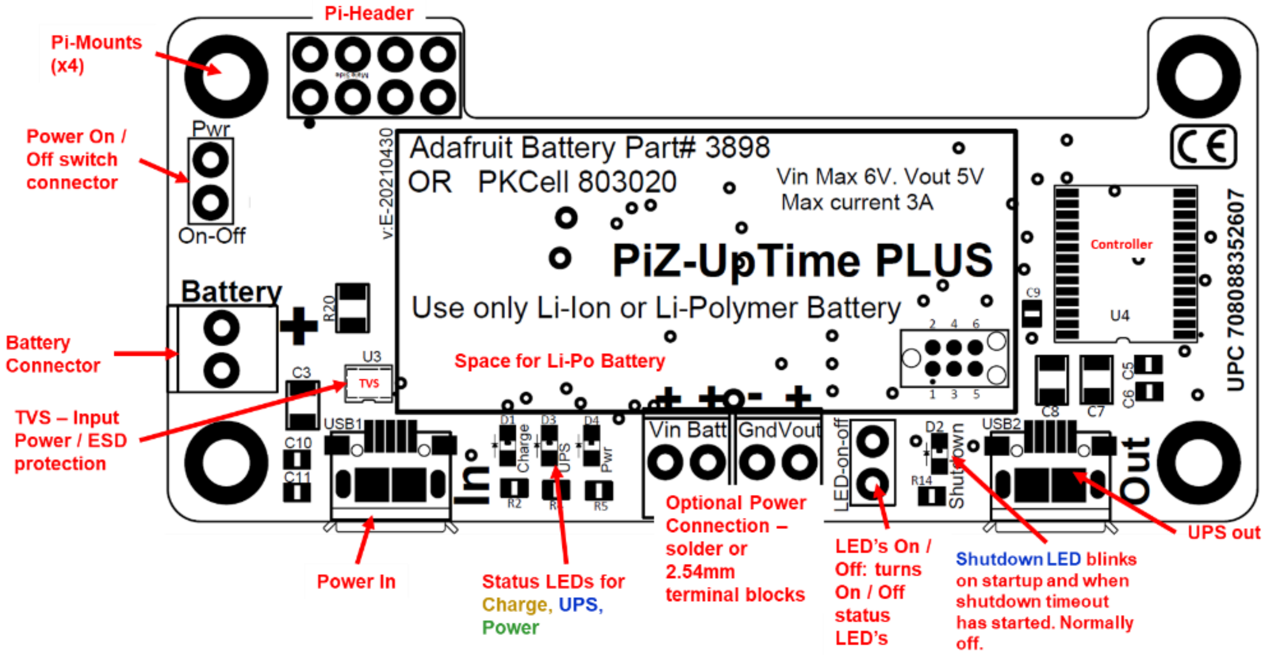
    b. The amount of time the UPS should be on AFTER shutdown has been initiated to ensure shutdown completes properly. The Python variable in the sample script is "timer". Note once shutdown has been started, the Shutdown (blue) LED with blink. If the shutdown has NOT been started, but the battery is below the "V_batt_min" level, the Shutdown (blue) LED will also blink indicating the power will be off after "timer" seconds.

    c. Operation mode ie. whether the PiZ-UpTime PLUS board runs as a UPS or a shutdown is started immediately after the shutdown is detected. The power is kept on for "timer" seconds. This mode is useful for Pi's installed in automobiles or devices where power is turned on/off often.

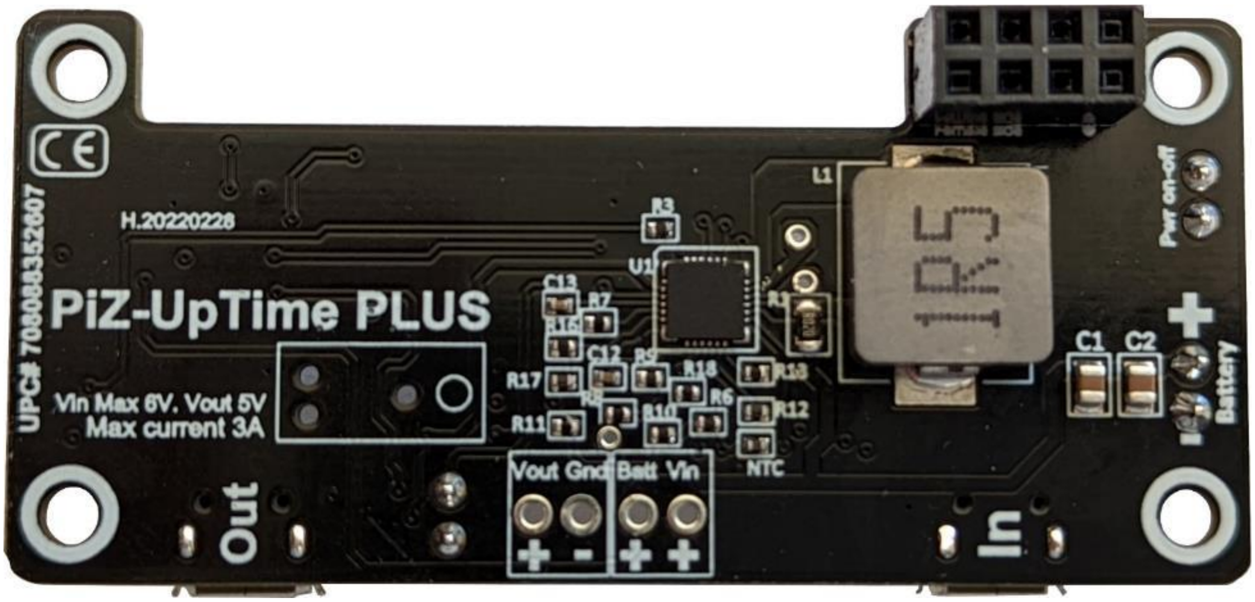
    d. Should the system wait after power returns. The wait time is configured by the user.

It is recommended to run the script from /etc/rc.local to ensure operating parameters are set when the Pi reboots.

Picture Gallery

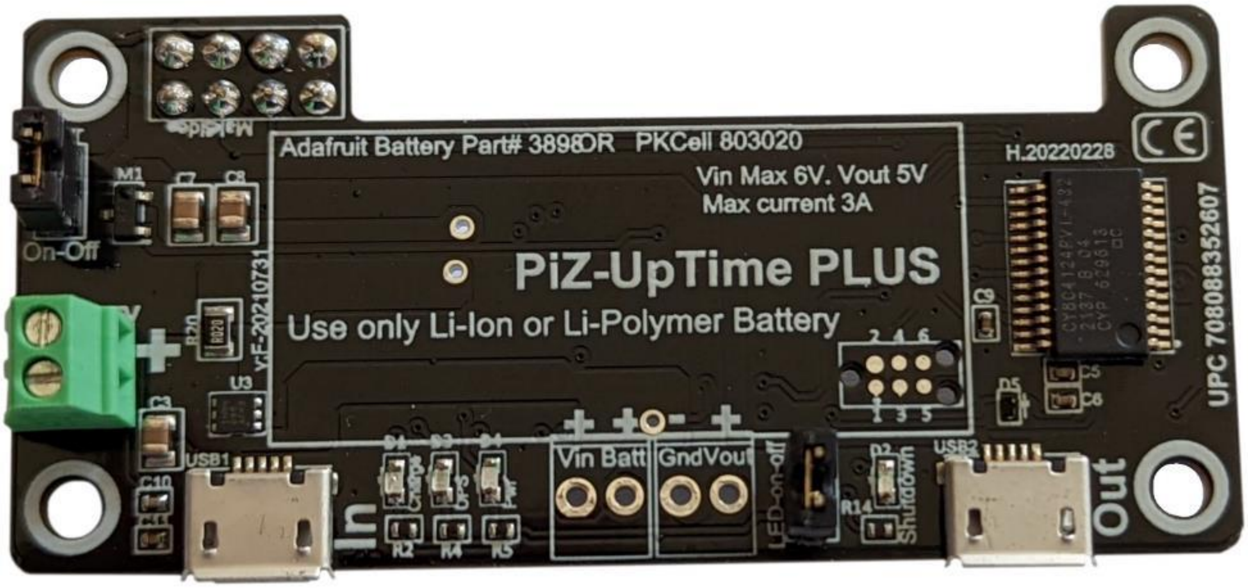

Pi-Header
Pi-Mounts (x4)
Power On / Off switch connector
Pwr
On-Off
Battery
Battery Connector
TVS – Input Power / ESD protection
TVS
U3
C3
R20
C10
C11
USB1
In
D1
Charge
D3
UPS
D4
Pwr
R2
R5
v:E-20210430
Adafruit Battery Part# 3898
OR PKCell 803020
Vin Max 6V. Vout 5V
Max current 3A
PiZ-UpTime PLUS
Use only Li-Ion or Li-Polymer Battery
Space for Li-Po Battery
Vin Batt
GndVout
LED-on-off
D2
R14
Shutdown
USB2
Out
C8
C7
C5
C6
C9
Controller
U4
UPC 708088352607
Power In
Status LEDs for Charge, UPS, Power
Optional Power Connection – solder or 2.54mm terminal blocks
LED's On / Off: turns On / Off status LED's
Shutdown LED blinks on startup and when shutdown timeout has started. Normally off.
UPS out

Capabilities detailed.

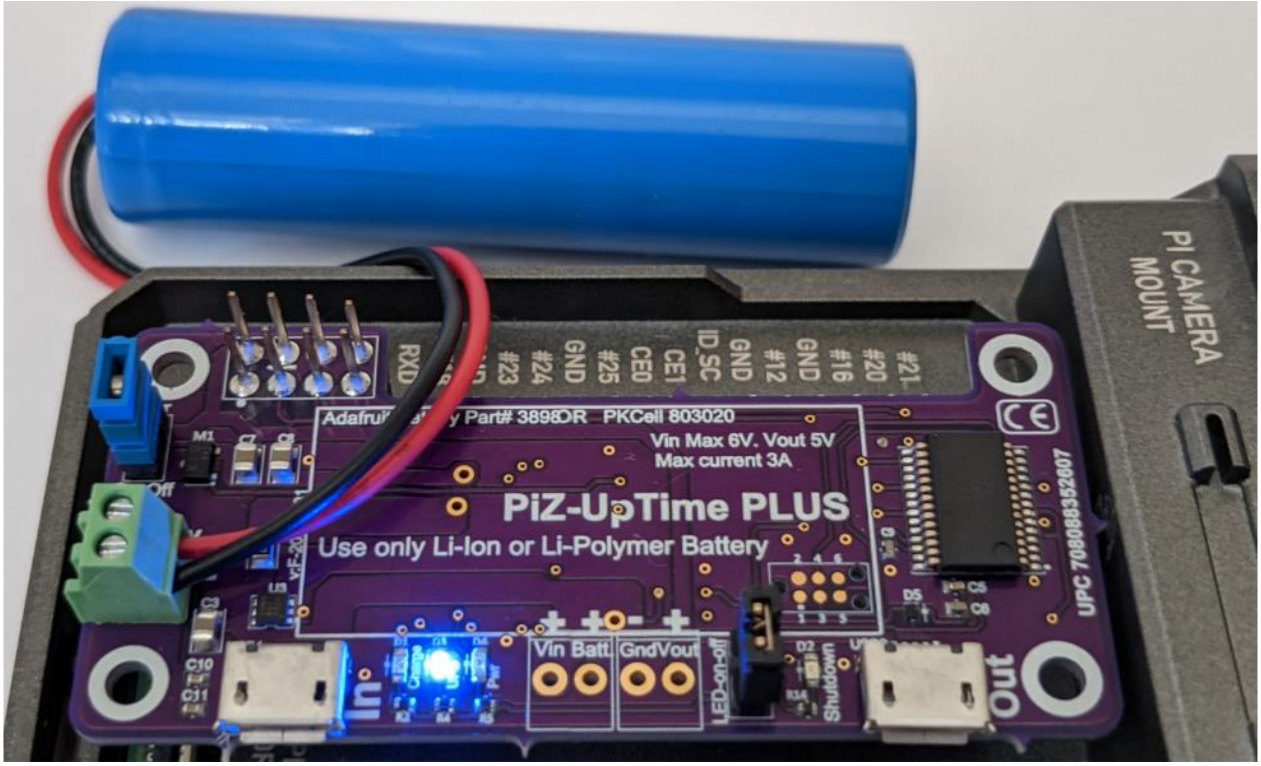


Bottom Layer.

All information subject to change without notice



Top Layer. Shown without connectors or parts soldered.



Mounted on Pi-4 enclosed in an Argon case, with 18650 batteries. Pi-4 is powered by PiZ-UpTime PLUS – providing the power from the 18650 battery. The blue UPS LED indicates UPS is on.

All information subject to change without notice

Table 1: PiZ-UpTime PLUS vs PiZ-UpTime 2.0

This table compares the older version of PiZ-UpTime with the current version of PiZ-UpTime 2.0

	PiZ-UpTime PLUS	**PiZ-UpTime 2.0**
Charge Current	Max 2.0 A.	Max 0.5A
Battery Chemistry (battery not supplied)	Li-Ion preferred – attached via terminal block.	Li-Ion or Li-Polymer – attached via terminal block
Battery Holder	None, attach via Terminal Block(s)	None, attach via Terminal Block(s)
Access to V_{in}, V_{out}, Battery	Yes, via terminal blocks, solder points and USB	Yes, via terminal blocks, solder points and USB
Input V	Max 5.5V, via micro-USB (in) **OR** via solder points or terminal blocks (not included). Input power passed through to output.	Max 5.5V, via micro-USB (in) **OR** via solder points or terminal blocks (not included). Input power passed through to output.
Output V	Regulated 5V out (±1%), irrespective of input V. No irritating "bolts" on Raspbian	Regulated 5V out (±2%), irrespective of input V. No irritating "bolts" on Raspbian
Max Current	3A	2A
On/Off	Attach an external on/off switch for manual on/off. Power reset done by micro-controller, assuring reboot.	Attach an external on/off switch for manual on/off. Power reset done manually by pulling shunt out/in
UPS USB Out	Yes, via micro-USB **OR** via terminal blocks	Yes, via micro-USB **OR** via terminal blocks
GPIO used	None. All communications by I^2C.	None – internal ADC is used
Operating parameters	V_{in}, V_{out}, $V_{battery}$ temperature (Temp ±2°C), I_{in} $I_{battery}$ - information available from microcontroller. Information via 12-bit ADC, correct to 3mV (Temp ±2°C)	V_{in}, V_{out}, $V_{battery}$ and temperature information available via on-board 4 channel 12-bit ADC, correct to 3mV (Temp ±2°C)
Python code for shutdown	Yes, sample code provided.	Yes, sample code provided.
External Battery	External batteries are attached. Space on board for Li-Polymer batteries	External batteries are attached. Space on board for Li-Polymer batteries
Battery size	Any capacity is used. Higher capacity takes longer to charge & provides longer run time on UPS	Any capacity is used. Higher capacity takes longer to charge & provides longer run time on UPS

	PiZ-UpTime PLUS	**PiZ-UpTime 2.0**
LED’s	Charge, Power, UPS and Shutdown / waiting for Power on to stabilize (normally off).	One – when battery is being charged. LED is off when battery is fully charged
Power in	Via micro-USB or solder points or terminal blocks	Via micro-USB or solder points or terminal blocks
UPS power out	Via micro-USB or solder points or terminal blocks	Via micro-USB or solder points or terminal blocks
Other	Ideal diode and max 3A power boost.	Uses Ideal diode OR – very little heat or power loss – efficient operation and seamless transition to battery backup for UPS
Shutdown Trigger point	Modified by user in Python script. Battery V level when shutdown is triggered is changeable	Modified by user in Python script. Battery V level when shutdown is triggered is changeable
Battery management / charging	Built on board, uses CCCV charging	Built on board, uses CCCV charging
Board Color	Black with white text	Black with white text
UPC code	708088352607	708088352577
Planned use / future	Recommended for use with Pi’s which needs to be rebooted in case of a shutdown.	Recommended for UPS needs at home and hobby applications.
Models supported	Any Pi with 40 pin headers. Any SBC with conformant power and ground pins (pins 2,4 and 6). Supports Pi-4’s. Please ensure the battery supports 3A or more discharge current.	Any Pi with 40 pin headers. Any SBC with conformant power and ground pins (pins 2,4 and 6). Pi-4’s may stress the unit. Please ensure the battery support s3A or more discharge current.
Software support	Via Python sample code. Uses I^{2}C to interact with processor on board. All data gathering and power monitoring done by onboard processor.	Via Python sample code. Uses I^{2}C to interact with 4 channel ADC to collect operating information.

All information subject to change without notice

Specifications

General Information

Model Number: PiZ-UpTime PLUS

Raspberry Pi Models supported

Any Raspberry Pi or other SBC with a 40-pin header compliant to Raspberry Pi Foundation header definition. Pins 2,4 are used for power. I^2C is used to control operating parameters. External devices are powered using USB out port.

Power

Input power: Power-in via micro-USB connector. 25-30W capable power adapter recommended. Batteries are charged only when power is provided to Power In connector on PiZ-UpTime PLUS board.

Power Adapter: 5V, 3.5A USB power recommended. Power adapter not included.

Maximum Charge Current: Maximum of 2 Amps.

Maximum UPS current: 3.0A

Ripple: Less than 25mV p-p.

Battery Low shutdown: 3.1V. Changed by the user in the Python code provided.

Input Operating range: 4.8 V to 6.0V. Above 6.0V the circuit will be damaged.

Output

UPS power: Regulated 5V, ± 1% via OTG micro-USB and 2x4 pin header. Power provided to the Raspberry Pi via the 8-pin header, pins 2,4.

Batteries

Batteries: One Lithium Ion non-protected battery connected via terminal blocks. Recommend only one battery. Using multiple batteries may cause current loops and could damage the circuitry. Li-ion battery should be capable of charging with 2A charge current. Battery with less than 2A charge capability may be damaged.

Recommended Batteries: Adafruit Part number 3898 or other similar batteries, PKCell 803020 or Li-Ion battery with wires for connection. Run time and charge time varies on capacity of the battery.

Battery capacity: Battery capacity will depend on your use. The higher the capacity, the longer the UPS run time, the longer the charge time.

LEDs: Yellow Charging LED is off when battery is charged, on when battery is charging. Green power LED shows input power is on. Blue UPS LED shows UPS is on. Blue Shutdown LED shows the operating parameters of the board (power on, waiting for power, shutting down).

Shutdown LED: Flashes 5 times when power is first connected. When shutdown is initiated, Shutdown LED flashes the number of times as set by timer variable in the sample Python script. After that, flashes for five seconds fast to show power is being turned off. Shutdown LED blinks for 2 seconds on and two seconds off when waiting for power to stabilize. The time and functions are set in the sample Python code provided.

Battery polarity: Marked on terminal blocks / board. **Inserting batteries incorrectly will damage the electronics and will cause the board to get extremely hot.**

Terminal Blocks recommended/used: Two Position Wire-to-Board Terminal Block, Horizontal with Board, 2.54mm (0.1inch) pitch, green color (color may vary), Thermoplastic, -30°C ~ 105°C, 6A, 125V. 20-30AWG wire is be used. Steel Zinc Screws, max torque 0.15 Nm (1.3 lb-in). Brass contact and clam material. RoHS 2 compliant. Recommend Onshore Technology Inc OSTVN02A150 Terminal blocks. Other manufacturers also provide equivalent terminal blocks.

Safety: Onboard circuitry prevent over charging and negative flow (from battery to power supply). If the battery is depleted, the charging is done slowly initially. After safe level, the CCCV charging method is initiated.

Battery Charging: Uses CCCV charging method. Safe operating temperature range recommended is from 0°C to 50°C.

Depleted Batteries: Senses depleted batteries and uses low current charging to bring charge level up and then CCCV charging. Initial charging for depleted batteries is a max of 100mA.

Monitoring: Monitor Input V, Output V, Battery V and board temperature using an on-board ADC. Sample Python Code provides capabilities to monitor ADC and shutdown the Pi when the battery V is low.

ADC capabilities: A 12-bit, SAR analog-to-digital converter (ADC). I^{2}C addresses is changed via sample Python script. Use with extreme caution and testing if I^{2}C code is changed. Minimum measurement sensitivity is 3mV.

Battery Full Charge Voltage: Li-Ion, Li-Polymer 4.2 V. Nominal 3.7V.

Spacers

M2.5x11m, not included. Spacers are recommended for mechanical stability and provide ambient cooling air space.

Connection Points

Connection points are spaced 2.54mm (0.1 inches) apart. Jumper studs, terminal-blocks or JST connectors are used for connections. Connections are marked on the board top layer as well as on the bottom layer.

Power on/off Switch

Reset: Turn power off by pulling out power On/Off jumper shunt. Turn power back on by inserting jumper shunt. Power shunt remove-reinsert sequence causes a power reset and a reboot.

External Switch: Instead of a shunt, an external on/off switch is connected to the jumpers via a cable. The shunts are 2.54mm. Please make sure the switch sustains a minimum of 10V, 5A or more.

On/Off Switch cable: Available at several web sites. An example is via Amazon, product-id B07FXXDN2M.

Dimensions

Board dimensions: 65mm x 30mm x 12mm (2.6”x 1.2” x 0.5”).

Weight: About 20g (0.7 oz.) without batteries. Note each battery adds approximately 30 grams of weight.

Header: Female pins on bottom. Pin thickness appx. 0.6mm. Female pin height appx 8.5 mm. No male pins on the top.

Warranty

90-day limited warranty. Warranty voided by improper use.

Other Information

Temperature:

Battery Charging Temperature: 0°C to +50°C, outside this range, battery charging is not recommended. Since the battery is separate from the board, battery temperature is not monitored. Only the board temperature is monitored.

Board Operating Temperature: -40°C to +80°C

Operating Humidity: 10% to 80% non-condensing.

RoHS etc.

RoHS Compliance: Electronic components, board etc. are all RoHS compliant.

Other: CE Compliance certificate available on request.

Recommended peripherals:

Please visit www.alchemy-power.com/products for recommended products.

Code download: www.alchemy-power.com/downloads

Product Video: none at this time.

7 08088 35260 7

UPC Code: 708088352607

Raspberry Pi and other Trademarks as shown in the document and belong to the respective trademark holders. Please refer to the respective organizations for Trademark, right of use and other information.

GP Consulting Inc.
(Owner of Alchemy Power Inc. IP and all rights)
Phone: 650.823.2316
Email: sales@alchemypower.com
www.alchemypower.com