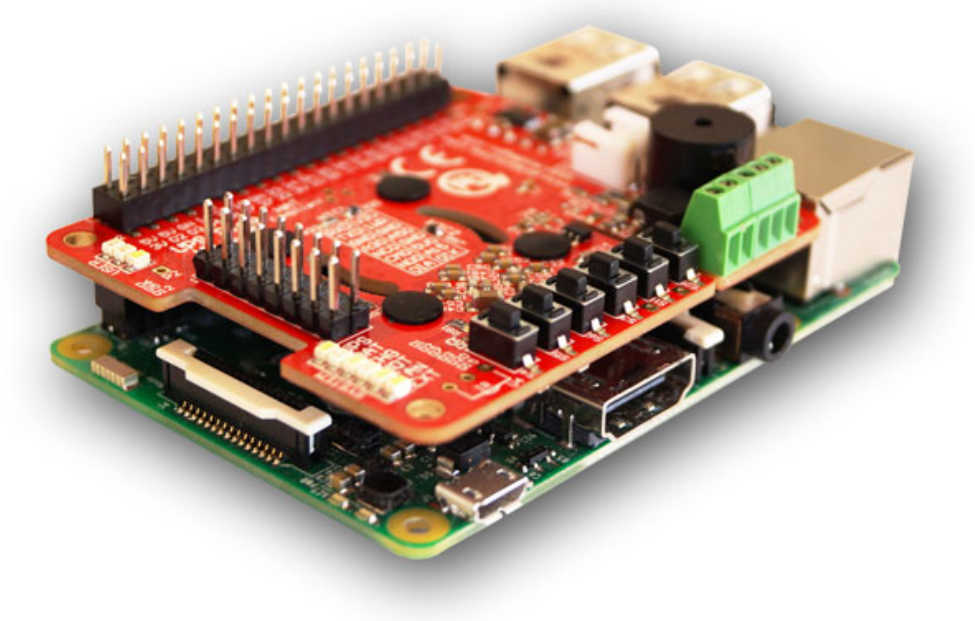
UPS PIco HV3.0A

Versions Stack/TopEnd/Plus/PPoE

Uninterruptible Power Supply with Peripherals and I²C control Interface



User Guide

Designed for the Raspberry Pi® 3

Compatible with

Raspberry Pi® 2, Pi Zero, A+, B+,

HAT Compliant

Designed and Manufactured by PiModules and ModMyPi

www.pimodules.com www.modmypi.com

Document Revisions

Version	Date	Modified Sections	Comments
na	06/11/2016	na	First Preliminary Public Document Release

Unlocked Hardware Features

Version	Date	Active Features
		TBC

Unlocked Firmware features

Version	Date	Active Features
Initial	Initial	Interrupts driven interaction with Raspberry Pi® based on Daemons
Initial	Initial	Simple status email broadcasting system based on Daemons
Initial	Initial	Intelligent Uninterruptible Power Supply (UPS)
Initial	Initial	3 User defined keys handler
Initial	Initial	3 User defined LEDs handler
Initial	Initial	LiPO and LiFePO4 chemistry battery support
Initial	Initial	Automatic Battery Charger
Initial	Initial	Battery Charger Power Tracking (Only Version Plus)
Initial	Initial	Integrated Hardware RTC with Battery Back-up
Initial	Initial	Automatic Files Safe Shutdown and Wake-up
Initial	Initial	Single button System ON/OFF for Battery and Cable (Only Version Plus) Powered Applications
Initial	Initial	PWM FAN Control
Initial	Initial	Zero Power Bi Stable Relay Control
Initial	Initial	Programmable Integrated Sounder
Initial	Initial	System Monitoring: Temperature, Voltages

Initial	Initial	Programmers I^2C PICo Interface
Initial	Initial	Programmers RS232 Interface (Basic Version)
Initial	Initial	Boot loader for on site firmware update
Initial	Initial	3 x A/D converters (Basic Version, without high voltage interface)

Firmware and Manual Fixes and Add-ons

Version	Code	Date	Firmware and Manual Fixes and Add-ons
0x22	22.001	15.01.2017	Enable/Disable and set data rate of the PIco RS232 is now working. Fixed default to disabled RS232. Fixed data rate programmable by user written to PIco EEPROM. Usage is described in the manual in a proper section.
0x22	22.002	15.01.2017	Activated Register **charger** at **0x69** and address **0x20** informing user if battery charger is ON or OFF. Usage is described in the manual in a proper section.
0x22	22.003	15.01.2017	Corrected bug with low battery threshold shutdown for **LiPO** and **LiFePO4** batteries, to 2.95V and 3.5 V
0x22	22.004	15.01.2017	Added full **LiFePO4** battery charger cutoff @3.75V
0x22	22.005	15.01.2017	Added for **LiPO** and **LiFePO4** battery charger trickle charging start up threshold to 4.1V and 3.6V respectively. This threshold is in additional controlled by the charger IC itself.
0x22	22.006	15.01.2017	User Manual re-structuralist
0x22	22.007	15.01.2017	Activated Register **key** at **0x69** and address **0x1A** informing user about the pressed **User Key (A,B or C)**. Usage is described in the manual in a proper section.
0x22	22.008	15.01.2017	Changed Battery Powering Backup activation threshold from **4.75V** on 5V GPIOs to **4.65V** **Remark:** Now the UPS PIco HV3.0 is checking the following conditions for Battery Power Backup activation on the following conditions: 1. Continuously Fast Falling Powering Edge Monitoring (proprietary algorithm) 2. GPIO 5V powering is unconditionaly lower than 4.65V This will help when used lower quality PSUs that are giving higher voltage drop than should be when system is loaded. In any case we **strongly recommend** to use PSU with micro USB cable on the same body with the PSU, and minimum **2.5A@5.1V,** recommend **3.0A@5.1V.**
0x22	22.009	15.01.2017	Corrected (activated) option with Battery Voltage measure **batlevel** at **0x69** and address **0x08** informing user about It. This option was temporary (in former version of firmware - 0x18) disabled in order to rewrite firmware part procedures from floating point arithmetic to fixed one, in order to decrease memory footprint and system speed-up. Usage is described in the manual in a proper section.
0x22	22.010	15.01.2017	Corrected (activated) option with Raspberry Pi Voltage measure **rpilevel** at **0x69** and address **0x0a** informing user about It. This option was temporary disabled in order to change firmware from floating point arithmetic to fixed one, in order to decrease memory footprint and system speed-up. Usage is described in the manual in a proper section.
0x22	22.011	15.01.2017	Corrected (activated) option with Raspberry Pi Voltage measure **eprlevel** at **0x69** and address **0x0c** informing user about It. This option was temporary disabled in order to change firmware from floating point arithmetic to fixed one, in order to decrease memory footprint and system speed-up. Usage is described in the manual in a proper section.
0x22	22.012	15.01.2017	Corrected (activated) option for the first A/D converter pre scaled to 5.2V at **0x69** and

			address **0x14**. This option was temporary disabled in order to change firmware from floating point arithmetic to fixed one, to decrease memory footprint and system speed up. Higher Voltage **could not be** supplied to this A/D converter. Readings are in 10^{th} of mV in BCD format. Usage is described in the manual in a proper section.
0x22	22.013	15.01.2017	Corrected (activated) option for the first A/D converter pre scaled to 5.2V at **0x69** and address **0x16**. This option was temporary disabled in order to change firmware from floating point arithmetic to fixed one, to decrease memory footprint and system speed up. Higher Voltage **could be** supplied to this A/D converter when added an extra resistor and set a proper variable in the PICo interface. Readings are in 10^{th} of mV in BCD format. Usage is described in the manual in a proper section. **The High Voltage interface is not activated yet.**
0x22	22.014	15.01.2017	Corrected (but not activated activated) option for the first A/D converter pre scaled to 5.2V at **0x69** and address **0x18**. This option was temporary disabled in order to change firmware from floating point arithmetic to fixed one, to decrease memory footprint and system speed up. Higher Voltage **could be** supplied to this A/D converter when added an extra resistor and set a proper variable in the PICo interface. Readings are in 10^{th} of mV in BCD format. Usage is described in the manual in a proper section. **The High Voltage interface is not activated yet.**
0x22	22.015	15.01.2017	Corrected the option with PIco Serial Port Data Rate selection and activation/deactivation (RS232) **rs232_rate** at **0x6B** and address **0x02.** Default option is PIco RS232 is disable (available for user applications). Usage is described in the manual in a proper section.
0x22	22.016	15.01.2017	Corrected (activated) Battery Selection independent of the boot loader presetting.
0x22	22.017	15.01.2017	Improved the cable reentry recognition time, checking is every 5 seconds

UPS PIco HV3.0 Models

The **UPS PIco HV3.0** is available in 5 different models all based on the same PCB:

1. **UPS PIco HV3.0 Stack**
2. **UPS PIco HV3.0 Stack Plus**
3. **UPS PIco HV3.0 Top End**
4. **UPS PIco HV3.0 PPoE (with Passive Power Over Ethernet – coming soon)**
5. **UPS PIco HV3.0 Top End Plus (future option – not released yet)**

The differences between each model are listed here below in the table with Technical Specifications however the core features are presented here:

UPS PIco 3.0 Core Features

The list of features of the **UPS PIco HV3.0** is as follows:

- Raspberry Pi B+ **HAT Compliant**
- **Plug and Play –** Ultra Simple semi **Automatic Installation** via GitHub
- **Interrupts driven interaction with Raspberry Pi® based on Daemons**
- Simple **status email broadcasting system based on Daemons**
- **Intelligent Uninterruptible Power Supply** (UPS)
- **Integrated LiPO Battery** (10-15 Minutes of System Power Back-Up)
- **Intelligent Automatic Charger with Power Tracking** (adjusting the battery charging current according to power availability from 100 mA – 1200 mA) - **implemented only in the UPS PIco HV3.0 Plus and PPoE**
- **No Additional External Power Input Required (if Stack/TopEnd is used)**
- **Optional 4000 or 8000 mAh** Battery for 8 – 16 Hours Run-Time (Not Included)
- **Optional 12000 mAh** Battery for extremely long Run-Time (Not Included – on special request)
- Each High Capacity Battery comes with **dedicated plastic screwed mounting base**
- Support for **LiPO and LiFePO4 batteries** on the same PCB with simple command switching
- **5V 2.6A Power Backup (Short Peak Output 5V 3A)**

- Integrated **Hardware Real Time Clock (RTC)** with Battery Back-Up
- **Automatic File Safe Shutdown** and **Wake-up** Functionality
- **Single button System ON/OFF for battery powered applications**
- **Events Triggered RTC Based System Actions Scheduler (Triggered ON/OFF based on RTC or External Events – voltage, RS232 data, A/D, IR etc)**
- **PWM fan control** (Fan Not Included), FAN PWM interface is integrated on each **PIco HV3.0** PCB
- **3 User Defined LEDs** for their own user application
- **3 User Defined Buttons** for their own user application
- **Separate TH Soldering Pads for each user button and FSSD** for external button connection
- **Bi Stable Relay (Zero Power) with dual separated independent contacts (Standard UPS PIco HV3.0 Plus or PPoE Only, optional for the UPS PIco HV3.0)**
- **Additional 5V (independent from Raspberry Pi® powering) power source with battery backup,** available for user applications also when Raspberry Pi is OFF (5V@750mA) **protected with PPTC FUSE** and **reverse current flow diode**
- **Optical Isolated input (opto coupler) – readable also as A/D values (UPS PIco HV3.0 Plus Only)**
- **Programmable Integrated Sounder** for UPS and User Applications (able to play music)
- **Status Monitoring** - Powering Voltage, UPS Battery Voltage, Current and Temperature
- **I²C PICo Interface** for Control and Monitoring
- **RS232 Raspberry Pi®** Interface for Control and Monitoring
- **Double path XTEA Based Cryptography** for User Software Protection
- 2 Level **Watch-dog Functionality** with **FSSD and unconditional Hardware Reset (if system Hangs-up)**
- **Extended Voltage Input 7-28V DC protected with zero voltage drop inverse polarity protection (UPS PIco HV3.0 Plus Only)**
- **Direct Raspberry Pi® Hardware Reset Button via Spring Test Pin (Pogo pin)**
- **16 pins Stack-able Header** for **PIco** Add-On Boards

- **Boot Loader** for Live Firmware Update
- **Intelligent IR Remote Power ON/OFF** (if **IR** received installed) **(UPS PIco HV3.0 Plus Only, or system shutdown – all models)**
- **Infra Red Receiver** Sensor Interface (IR Not Included) directly connected to the GPIO18
- Integrated EDS-Protected **3 Channel A/D 12 Bit Converters 0-5.2V** (optional per-scaled to 10V, 20V and 30V via **Terminals Blocks PCB** add-on or external resistor)
- Optional **A/D converter** with **voltage follower** (ideal for **IoT** applications) - via Terminals Blocks PCB add-on boards to 16 pins header
- **Integrated ESD-Protected 1-Wire Interface**
- **Programmable second RS232 interface (5V tolerant)**
- **12V RS232 interface via Terminals Blocks PCB**
- **Labeled J8 Raspberry Pi® GPIO Pins** for Easy Plug & Play of experimental cables
- **Fits inside to the Official Raspberry Pi® Case with closed lid (version Top End)**
- **Fits Inside Most Existing Cases**

System Overview

Introduction

The **UPS PIco HV3.0A** is an advanced uninterruptible power supply for the **Raspberry Pi® 3** that adds a wealth of innovative power back-up functionality and development features to the innovative micro-computer!

Especially designed for the **Raspberry Pi® 3** but also compatible with all other models; offers now Total **3A** current from the External Power Supply or battery backup, **3** User Keys, **3** User LEDs, **3** different types of high capacity batteries, 2 x **3** pins **bi-stable relay** (Zero Power), as also **3 x A/D 12 bit** converters pre-adjusted to 5V, 15V and 30V conversion (when used with Terminal Blocks PCB). Now, with additional **External Supply Powering Input**; that has implemented **Dynamic Power Tracking**; automatically adjust battery charging current according to power availability from 100mA – 1000 mAh. This feature has been especially designed to support **Solar Panel Powering Raspberry Pi® Systems**, as it is adjusting the charging battery current to available Sunning conditions. The **External Supply Powering Input** is able to accept power from **7 V DC** up to **28 V DC !!** Thus make it ideal for Cars, Trucks, Buses and any industrial applications where voltage is usually higher than 24V DC. The **External Supply Powering Input** is equipped with Over Current protection, Over Voltage as also **with Zero Voltage Drop Inverse Polarity Protection** in order to use all available energy from the Solar Panel in case of use. The New **UPS PIco HV3.0A** is an all-in-one tool that allows implementing easy and fast simple applications as also complicated projects providing a set of pre installed peripherals.

The **UPS PIco HV3.0A** is standard equipped with a 450 mAh 15C LiPO battery specially designed to enable safe shutdown during cabled power cut and automatic system restart when cabled power comes back. The included battery provides enough energy to keep running system for 10-15 minutes. Additionally, this can be easily upgraded to the extended **4000mAh** version, **8000 mAh** as also **12000 mAh** batteries (optional on special request), which enables prolonged use of a Raspberry Pi for **more than 32 hours** without a power supply connected (with biggest battery installed) !

The **UPS PIco HV3.0A** design support now batteries with different chemistry: **LiPO** as also **LiFePO4**. Especially the **LiFePO4** batteries are addressed to applications where temperatures environment is more restricted as can be used for supplying from **-10 degrees up to +60 degrees**. In addition the **LiFePO4** have a unique extremely long life of charging/discharging that can achieve up to **2000 cycles!!**

The implemented trimmed **Hardware Real Time Clock and Calendar**, guarantees time stamp when system is running without access to the Network. The **Hardware RTCC** is backed up and powered from the integrated system battery. The **RTCC** current consumption is **only 1 uA.**

The integrated **Hardware RTCC** enables a new extremely usefully feature – the **Events Triggered RTCC Based System Actions Scheduler**. The **Events Triggered RTCC Based System Actions Scheduler** allows to timely start up, or shutdown the **Raspberry Pi®** on various internal or external events that include, data available on RS232, IR, A/D, RTCC, and temperature, Opto Coupled Input or just on requested Time Stamp.

Professional developers often need to protect their application. In order to support them **UPS PIco HV3.0A** offers the **XTEA** dual path encryption (on read and write path) embedded engine that protect the developed software with the secure code**.**

The **UPS PIco HV3.0A 450 mAh Stack Plus** offers **2.6A** battery power backup for the **Raspberry Pi®** via GPIOs as also **3A** extended power supply, but not only. In addition is offered an independent from the **Raspberry Pi®** powering, battery backed output of **5V@750 mA** available for the user devices connected to the **Raspberry Pi®** that must be running even if the mother **Raspberry Pi®** is shut down and not powered (i.e. USB powered HUBs, WiFi Routers, Motion Detectors, HDDs etc). The total current cannot exceed 3A. The current supply delivered via GPIOs to the **Raspberry Pi®** is **2.6A**

Many applications need to have **secondary RS232** in addition to the primary one offered by the **Raspberry Pi®**. Until today, it has been solved by users with add-on hardware put on the top. Not anymore!! With the **UPS PIco HV3.0A** user have access to integrated secondary serial port 3.3V level but save also to be used with 5V level. This makes the developed application cost effective and more robust

Now with **additional Terminals Blocks** Add-on PCB **UPS PIco HV3.0A** offers a professional I/O connectivity for any industrial application including 12V level converter for both Serial Ports (one of them must be selected).

The **Zero Power Bi Stable Relay** offers two independent sets of terminals, offers up to 1A contacts able to switch ON/OFF various peripherals of the developed system. Due to unique design, no power is required when Bi Stable Relay is in Set/Reset state, making it ideal for battery powered applications. Two independent 3 pins sets offered (NC, NO, COM) are switched at the same time and are able to relay 2 independent applications.

Now, the high voltage signal can be monitored safely with the **Opto Coupled interface**, which can be read as digital as also analogue input.

The **IoT** developers will find useful the **3 ESD protected 12 bits buffered A/D converters** as also number of internal sensors and sensor interfaces that can be used for system monitoring such as Battery Voltage, External Powering Voltage, Raspberry Pi Voltage, Current Consumption**,** System Temperature and 1-wire interface.

Especially with the additional **Terminals Blocks** Add-on PCB user can preset A/D converters to scaled measure of 5V, 15V, as also 30 V with a simple jumper selection. In addition the **Terminals Blocks** Add-on PCB offers one A/D converter followed voltage follower with input impedance of 1M Ohm, allowing very high impedance sensors connectivity.

The **UPS PIco HV3.0A** can also be equipped with an optional **Infra-Red Receiver** which is routed directly to GPIO18 via the PCB. This opens the door for remote operation of the Raspberry Pi® and **UPS Pico** including remotely **ON/OFF.**

The embedded **Electromagnetic Programmable Sounder** can be used as a **simple buzzer** but also as **music player** due to implemented sound generator and dedicated programmer interface.

Finally, the **UPS PIco HV3.0A** features an implemented Automatic Temperature Control **PWM FAN controller**, and can be equipped with a micro fan kit, which enables the use of the

Raspberry Pi® in extreme conditions including very high temperature environments. The FAN speed can be manually/automatically adjusted according to system temperature conditions linear from 0 % (FAN is OFF) up to 100% by increasing and decreasing rotation speed. Thus guarantees the possible lowest level of noise and always cool **Raspberry Pi® 3.**

UPS PIco 3.0 Technical Specifications

Features	Model UPS PIco HV3.0 A		
	UPS PIco HV3.0 A Stack 450	UPS PIco HV3.0 A Stack 450 Plus	UPS PIco HV3.0 A Top End 450
Raspberry Pi®			
Raspberry Pi® System Compatibility			
	Designed for Raspberry Pi® 3 Compatible with Pi2, Pi3, Pi Zero, A+, B+	Designed for Raspberry Pi® 3 Compatible with Pi2, Pi3, Pi Zero, A+, B+	Designed for Raspberry Pi® 3 Compatible with Pi2, Pi3, Pi Zero, A+, B+
Cases Compatibility			
	Most of the cases ModMyPi cases PiModules PIco case	Most of the cases ModMyPi cases PiModules PIco case	Most of the cases **Recommended Raspberry Pi Original Case** adopted to Media Player Applications
Raspberry Pi® GPIO Usage (occupation)			
Permanent use of I²C (User selectable addresses)	GND, 5V, SDA0, SCL0 I²C Addresses: **68 69 6a 6b 6c 6d 6e 6ff**	GND, 5V, SDA0, SCL0 I²C Addresses: **68 69 6a 6b 6c 6d 6e 6ff**	GND, 5V, SDA0, SCL0 I²C Addresses: **68 69 6a 6b 6c 6d 6e 6ff**
Selectable use of Raspberry Pi® RS232	TXD0, RXD0 OFF	TXD0, RXD0 OFF	TXD0, RXD0 OFF
Selectable use of Raspberry Pi® GPIO	GPIO_GEN22 (pulse train generator) GPIO_GEN27 (System Shutdown initiator) GPIO_GEN18 (if IR receiver is used) GPIO_GEN4 (if 1-wire is used)	GPIO_GEN22 (pulse train generator) GPIO_GEN27 (System Shutdown initiator) GPIO_GEN18 (if IR receiver is used) GPIO_GEN4 (if 1-wire is used)	GPIO_GEN22 (pulse train generator) GPIO_GEN27 (System Shutdown initiator) GPIO_GEN18 (if IR receiver is used) GPIO_GEN4 (if 1-wire is used)
Interactions with Raspberry Pi®			
Battery and Charger			
Supported Batteries Types			
LiPO 3.7V with silicone high current cables			
	Standard - LiPO 450 mAh	Standard - LiPO 450 mAh	Standard - LiPO 450 mAh (dedicated to be used with Raspberry Pi Original Case)
	Optional - LiPO 4000 mAh	Optional - LiPO 4000 mAh	
		Optional - LiPO 8000 mAh	
LiFePO4 3.2V with silicone high current cables			
	Optional – LiFePO4 4000	Optional - LiFePO4 4000 mAh	
		Optional - LiFePO4 8000 mAh	
		Optional - LiFePO4 12000 mAh (due to big size of batter only on special order)	
Battery Life Charge/Discharge Cycles			
LiPO	450 cycles	450 cycles	450 cycles
LiFePO4	2000 cycles	2000 cycles	2000 cycles
Battery Charger			
	Standard - Continues fixed current 303 mAh	Automatic Dynamic Power Tracing Charger with charging current 100 mA – 1000 mA, triggered by voltage changes on the 5V GPIO or External Power Source	Standard - Continues fixed current 303 mAh
Charging Modes			
LiPO	Automatic Selected : Full Charging Cycle Trickle Charging	Automatic Selected : Full Charging Cycle Trickle Charging	Automatic Selected : Full Charging Cycle Trickle Charging
LiFePO4	Automatic Selected : Full Charging Cycle Trickle Charging	Automatic Selected : Full Charging Cycle Trickle Charging	Automatic Selected : Full Charging Cycle Trickle Charging

Battery Protection			
450 mAh	On board cut-off protection system when thermal, overcharge or over discharge	On board cut-off protection system when thermal, overcharge or over discharge	On board cut-off protection system when thermal, overcharge or over discharge
High Capacity LiPO and LiFePO4	On board cut-off protection system when thermal, overcharge or over discharge On battery additional PCM protection	On board cut-off protection system when thermal, overcharge or over discharge On battery PCM additional protection	On board cut-off protection system when thermal, overcharge or over discharge On battery PCM additional protection
Battery Electrical Isolation System	Battery is Electrically Isolated (however cable connected) until system start up for the first time and receive 5V from GPIO	Battery is Electrically Isolated (however cable connected) until system start up for the first time and receive 5V from GPIO or 7-28V from EXT	Battery is Electrically Isolated (however cable connected) until system start up for the first time and receive 5V from GPIO
Battery Back-Up			
System Battery Backup	Standard – 5V 2.6A current continuous supply to Raspberry Pi via GPIO Pins	Standard – 5V 2.6A current continuous supply to Raspberry Pi via GPIO Pins	Standard – 5V 2.6A current continuous supply to Raspberry Pi via GPIO Pins
Auxiliary 5V Battery Backed Supply on PIco I/O Pins	Standard – 5V 750 mA current continuous supplies on PIco I/O Pin battery backed, with possibility to continuous supply auxiliary devices with Raspberry Pi disconnected. Total system current should not exceed 3A.	Standard – 5V 750 mA current continuous supplies on PIco I/O Pin battery backed, with possibility to continuous supply auxiliary devices with Raspberry Pi disconnected. Total system current should not exceed 3A.	Standard – 5V 750 mA current continuous supplies on PIco I/O Pin battery backed, with possibility to continuous supply auxiliary devices with Raspberry Pi disconnected. Total system current should not exceed 3A.
Battery Back-up Type			
UPS	UPS Standby Type, with switchover time of 250 uS, during switching time the protected system (Raspberry Pi® with added hardware) is powered by auxiliary online power source for maximum 10mS, therefore no power gap on GPIO during switching time	UPS Standby Type, with switch over time of 250 uS, during switching time the protected system (Raspberry Pi® with added hardware) is powered by auxiliary online power source for maximum 10mS, therefore no power gap on GPIO during switching time	UPS Standby Type, with switch over time of 250 uS, during switching time the protected system (Raspberry Pi® with added hardware) is powered by auxiliary online power source for maximum 10mS, therefore no power gap on GPIO during switching time
Powering Monitoring Point	Raspberry Pi® GPIO 5V	Raspberry Pi® GPIO 5V	Raspberry Pi® GPIO 5V
UPS Activation Powering Triggers	GPIO 5V pins <=4.65V Proprietary Algorithm of Falling Power Peak Analysis	GPIO 5V pins <=4.65V Proprietary Algorithm of Falling Power Peak Analysis	GPIO 5V pins <=4.65V Proprietary Algorithm of Falling Power Peak Analysis
Cable Powering Reactivation	After 3s of continuously cable powering (without spikes)	After 3s of continuously cable powering (without spikes) on any cable power source (GPIO or External)	After 3s of continuously cable powering (without spikes)
Cable Powering Sources			
Cable Powering Sources			
Raspberry Pi ® GPIO 5V Pins	2.6 A	2.6 A	2.6 A
External Power Source 7 - 28 VDC		3A max (adjusted according dynamic power tracking)	
Additional Features - Peripherals			
HAT Compliant			
HAT EEPROM	Exists	Exists	Exists
HAT Dimensions	Compliant	Compliant	Compliant
PIco I/O Interface			
Independent from Raspberry Pi ® 3.3 V supply @200 mA	Yes	Yes	Yes
ESD Protected 1-wire interface	Yes	Yes	Yes
Independent from Raspberry Pi ® 5.0 V supply @750 mA With battery Back-up (Raspberry Pi ® can be OFF when this power Auxiliary 5 V source is available)	Yes	Yes	Yes
12 Bit A/D converters ESD protected, pre-scaled to 5V, 15V and 30V (on TB PCB) with Sampling rate 100K SPS,	Yes	Yes	Yes

buffered			
3V3/5V0 RS232 Port that can be programmed as: **primary Raspberry Pi® Port** **Secondary (independent from the existing on Raspberry Pi®)**	Yes	Yes	Yes
Optical Isolated Interface (readable as digital or analog)	none	Yes	none
Primary 3 Pin Bi-stable (Zero Power) Relay Interface **Rating (resistive)** **Maximum Switching Current/Voltage on Teminal Block** **Current/Voltage on 16 Pin Header**	Yes (Optional) 1A 30 VDC	Yes 1A 30 VDC	Yes (Optional) 1A 30 VDC
PIco Terminals Block Extension PCB (Supplied separately)			
12 V RS232 converter attached to primary or secondary Serial Port	Yes (Optional with TB PCB)	Yes (Optional with TB PCB)	Yes (Optional with TB PCB)
Terminal Block on Each PIco I/O Interface listed above	Valid only for existing Interfaces	Valid only for existing Interfaces	Valid only for existing Interfaces
PIco Plus Terminal Block Standard Interface			
DC in 6 – 28 V with Power Tracking	none	Yes	none
Secondary 3 Pin Bi-stable (Zero Power) Relay Interface	Optional if Relay Installed	Yes	Optional if Relay Installed
Hardware User Interface			
System LEDs Indicators	UPS, BAT, CHG, HOT, FAN	UPS, BAT, CHG, HOT, FAN, EXT	UPS, BAT, CHG, HOT, FAN
User LEDs Indicators	Blue, Green, Red	Blue, Green, Red	Blue, Green, Red
System Keys	RPiR, UPSR, FSSD	RPiR, UPSR, FSSD	RPiR, UPSR, FSSD
User programmable Keys	AKEY, BKEY, CKEY	AKEY, BKEY, CKEY	AKEY, BKEY, CKEY
External Connectivity to PIco Keys	FSSD, AKEY, BKEY, CKEY (soldering TH pads for cables)	FSSD, AKEY, BKEY, CKEY (soldering TH pads for cables)	FSSD, AKEY, BKEY, CKEY (soldering TH pads for cables)
Audio Interface	Electromagnetic Transducer, with programmable sound duration and frequency, able to play music	Electromagnetic Transducer, with programmable sound duration and frequency, able to play music	Electromagnetic Transducer, with programmable sound duration and frequency, able to play music
Other Features			
Battery Backed Hardware Real Time Clock and Calendar	Yes	Yes	Yes
Bi-Stable (Zero Power) Relay	Yes (optional)	Yes	Yes (optional)
Passive Cooling System	Based on multiple copper layers thermal pipes for heating dissipation	Based on multiple copper layers thermal pipes for heating dissipation	Based on multiple copper layers thermal pipes for heating dissipation
Automatic Active Cooling System (FAN)	Yes (optional if FAN installed)	Yes (optional if FAN installed)	Yes (optional if FAN installed)
Automatic File Safe Shutdown Functionality	Yes	Yes	Yes
Raspberry Pi® Reset via POGO Pin	Yes	Yes	Yes
Automatic Restart on Power Return	Yes	Yes	Yes
Events Triggered RTCC Based System Actions Scheduler	Yes Basic	Yes Extended on more Events	Yes Basic
Real Time Raspberry Pi® current measure	Yes (both ways) Incoming to UPS PIco Outgoing from UPS PIco	Yes (both ways) Incoming to UPS PIco Outgoing from UPS PIco Incoming from Extended Voltage Input	Yes (both ways) Incoming to UPS PIco Outgoing from UPS PIco
Real Time Battery Capacity Measure	Yes (based on System current consumption)	Yes (based on System current consumption)	Yes (based on System current consumption)
Secondary Serial Port (based on software driver)	Yes (future firmware option)	Yes (future firmware option)	Yes (future firmware option)
IR interface	Yes	Yes	Yes
Optimized design for a very low noise A/D operation	Yes Split grounds, extended Improved filtering on PSU High Speed Separate Tracing	Yes Split grounds, extended Improved filtering on PSU High Speed Separate Tracing	Yes Split grounds, extended Improved filtering on PSU High Speed Separate Tracing
Optimized Ultra Low Power design for a long time Battery System Operation	Yes	Yes	Yes
XTEA Encryption	Yes	Yes	Yes
Extended Raspberry Pi® Watch-Dog (Still Alive)	Yes	Yes	Yes
System Monitoring	Battery Voltage, Raspberry Pi®	Battery Voltage, Raspberry Pi®	Battery Voltage, Raspberry Pi®

	Voltage, Current Consumption by Raspberry Pi® and PIco, Temperature	Voltage, External Voltage, Current Consumption by Raspberry Pi®, Temperature	Voltage, Current Consumption by Raspberry Pi® and PIco, Temperature
I2C PIco Programmer Interface	Yes	Yes	Yes
RS232 @command Interface on Primary and Secondary Serial Port	Yes	Yes	Yes
Bootloader for Live Firmware Update	Yes	Yes	Yes
PCB Construction			
PCB Manufacturing	4 Layers, 2 OZ Cupper, 8mils/8mils Immersion Gold Plated PB Free alloy assembly	4 Layers, 2 OZ Cupper, 8mils/8mils Immersion Gold Plated PB Free alloy assembly	4 Layers, 2 OZ Cupper, 8mils/8mils Immersion Gold Plated PB Free alloy assembly

UPS PIco HV3.0 Add-On equipment

The **UPS PIco HV3.0** can be combined with additional already available parts. There are:

1. UPS PIco HV3.0 Fan Kit
2. UPS PIco HV3.0 Relay Kit
3. LiPO Battery 4000 mAh
4. LiPO Battery 8000 mAh
5. LiFePO4 Battery 4000 mAh
6. LiFePO4 Battery 8000 mAh
7. LiFePO4 or LiPO 12000 mAh (only on special order)
8. Terminal Blocks PCB
9. Dedicated UPS PIco HV3.0 Plexiglas Case LF or LP

What is in the BOX?

This package comes with everything you need to start using the **UPS PIco HV3.0** right out of the box. It is assembled, tested and contains all required accessories. A little work is necessary in order to setup the complete Raspberry® and **UPS PIco HV3.0** in a single full operating system, and this is instructed below.

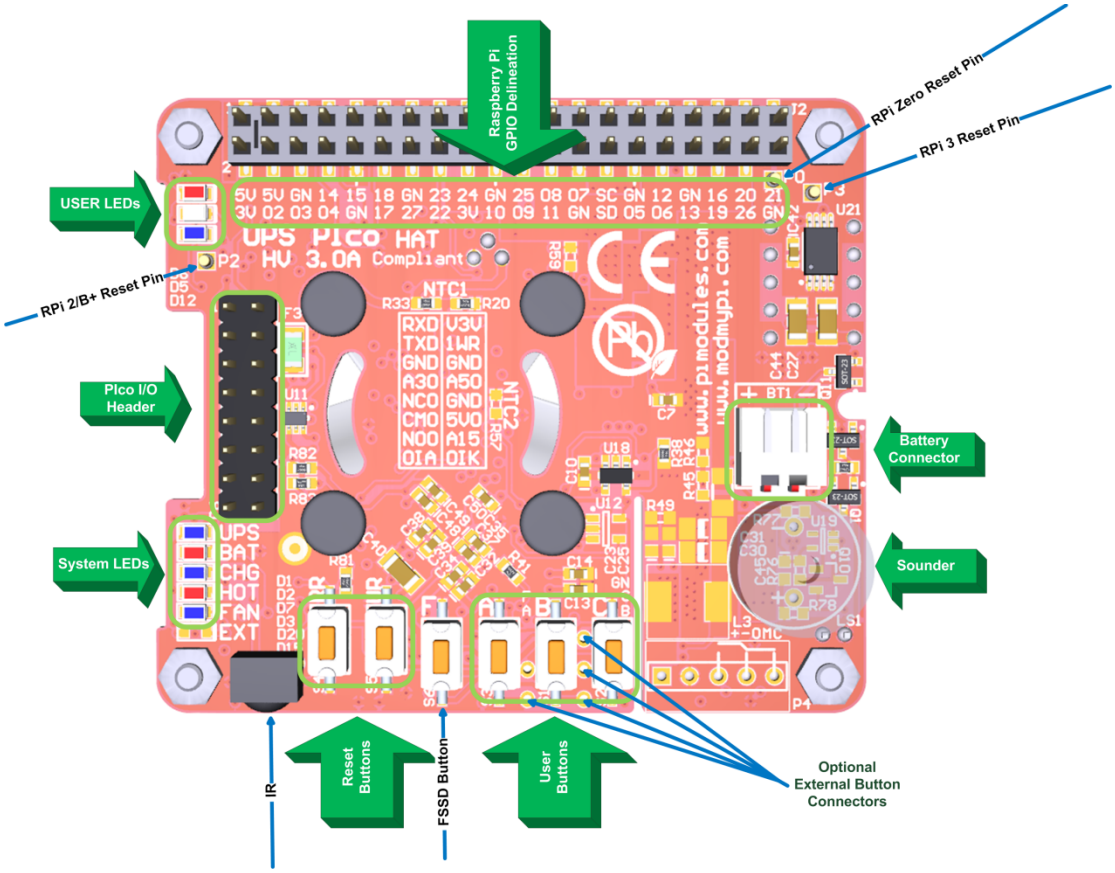
Each Box contains the following parts:

- 1 x The **UPS PIco HV3.0** module
- 1 x 40 THT Header (Stack or Top End)
- 1 x Dual layer wide temperature adhesive tape (used for battery mounting) stuck on the bottom side of **UPS PIco HV3.0** battery, or left free in the box
- 1 x Set of spacers (plastic spacers, rubber stick, or screws and plastic spacer tubes, depending of production lot)
- 1 x Separate packed ultra high current LiPO battery 450 mAh, 6A current
- 1 x Gold Reset Pin (POGO pin)
- 1 x Electromagnetic Sounder
- 2 x 8 pins Headers Strips

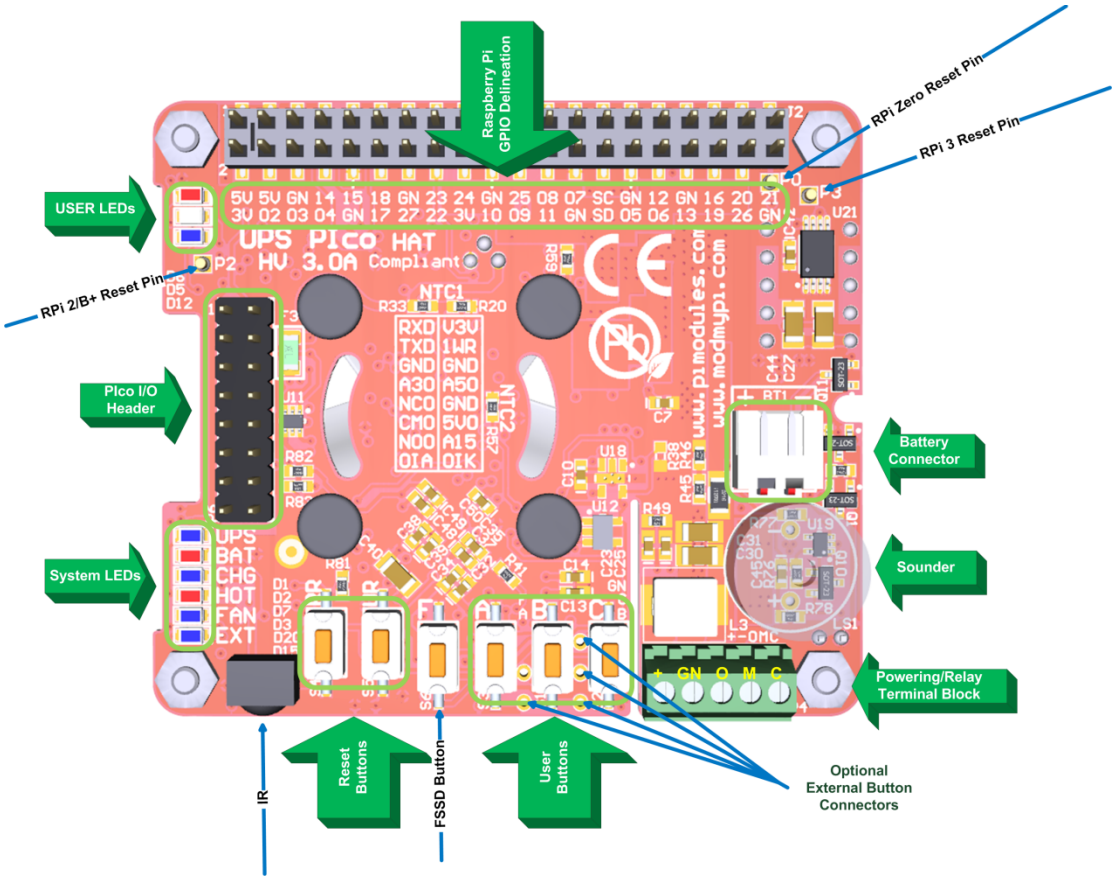
Please kindly notice that, due to shipping regulations, LiPO batteries are packed in the same box but are physically or electrically separated and not connected to the **UPS PIco HV3.0** module. It must be connected by the user, and it is a part of the installation procedure.

Setting up Procedure

Hardware Setup fro UPS PIco HV3.0 Stack/TopEnd



Hardware Setup fro UPS PIco HV3.0 Plus/PPoE



Software Setup for UPS PIco HV30 Stack/TopEnd/Plus/PPoE

Software Installation Procedure of the UPS PIco HV3.0 Daemons

and UPS PIco HV3.0 email broadcasting System

1. Install Raspberry Pi Operation System (i.e. NOOBs)
   - **Disable the serial port (only if you need to upload a new firmware) via raspi-config**
   - **Enable the I2C**

2. Ensure that Python is installed and updated, by using the following command

   sudo apt-get install python-rpi.gpio

3. Ensure to run below line

   sudo apt-get install git python-dev python-serial python-smbus python-jinja2 python-xmltodict python-psutil python-pip

   (Take note of the line-wrapping above, it should all be on one line)

4. Note that some of the above can also be install with pip as below:

   sudo pip install jinja2
   sudo pip install xmltodict

   (Obviously after python-pip has been installed)

5. Clone Raspberry Pi daemons and email broadcasting system from the GitHub using the following command

   sudo git clone https://github.com/modmypi/PiModules.git

6. Move to the required directories where software has been copied.
7. First to the email broadcasting system (package)

   sudo cd PiModules/code/python/package

8. Then proceed with the installation of the email package software

   sudo python setup.py install

more information about the package usage and details are available at

https://github.com/modmypi/PiModules

9. Second to the System Monitoring and File Safe Shutdown Daemons (picofssd)

For interaction of UPS PIco HV3.0 is used fixed Raspberry Pi GPIOs:
GPIO_GEN27 and GPIO_GEN22
These GPIOs are used for sending Train Pulse as also to initiate the Files Safe System Shutdown (FSSD) and should be not used in any other applcations

cd ../upspico/picofssd

10. Then proceed with the installation of the picofssd daemons software

sudo python setup.py install

11. Once the script has been installed, it can be installed to the `SysVInit` system with the following command

sudo update-rc.d picofssd defaults

12. Enable to run at boot time with the following command

sudo update-rc.d picofssd enable

13. Now when the Pi is rebooted the daemon should start automatically.

The Daemons can be started and stopped in the usual way for SysVInit:

sudo /etc/init.d/picofssd start

sudo /etc/init.d/picofssd stop

Important Notices:

(1) Both PIco packages must be installed even if not used

(2) It is very important to start/stop the Daemons Service when doing Hardware Reset of the PIco HV3.0 in order to avoid undefined situations with pulse train recognition procedure by the system. Resetting the PIco with Not Stopped the Daemon Service can cause an unexpected system shutdown (however without card corruption - system will just safety shutdown)

Installation Procedure of the UPS PIco HV3.0 Hardware RTC

1. Proceed with the installation of the i2c-tools using the following command

   sudo apt-get install i2c-tools

2. Now edit the /etc/modules file

   sudo nano /etc/modules

3. Make sure to have the following items in the file and add what is missing

   i2c-bcm2708
   i2c-dev
   rtc-ds1307

4. Now edit the file /etc/rc.local

   sudo nano /etc/rc.local

5. and Add the following lines, before "**exit 0**"

   echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
   (sleep 4; hwclock -s) &

Firmware updates Procedure of the UPS PIco HV3.0 (on RPi3)

To disable serial communication over the UART long enough to do a Firmware update. You need to do two things.

1. Disable Bluetooth. Add this to the end of /boot/config.txt:

dtoverlay=pi3-disable-bt

(which disables Bluetooth)

Do this to stop Bluetooth trying to use UART:

sudo systemctl disable hciuart

2. Disable the serial console. Do this to stop the serial console on the UART:

sudo systemctl stop serial-getty@ttyAMA0.service

To stop it from starting again when rebooted:

sudo systemctl disable serial-getty@ttyAMA0.service

To re-enable it when you have finished updating firmware:

sudo systemctl enable serial-getty@ttyAMA0.service

To start it without rebooting:

sudo systemctl start serial-getty@ttyAMA0.service

You could also disable the console by editing /boot/cmdline.txt and rebooting

The UPS PiCo features an embedded serial boot loader which allows users to manually update the unit's firmware. The firmware can be uploaded using a dedicated python script, called **9600_picofuHV3.0.py**

Bootloader is small piece of firmware stored permanently in the micro controller flash memory of the UPS Pico, located in the special protected area. It can not be erased by user without dedicated hardware tools. In order to upload new firmware, an invocation of the bootloader routine is needed. It can be done manually or automatically if I2C-tools is running and installed.

The bootloader functionality ensures that the UPS PIco is up-to-date, and allows users to report various changes that can be implemented on the user's side. It is extremely useful functionality, and ensures that the product has longevity.

It is mandatory to have previously installed python and I2C-tools on the Raspberry Pi. You will install these during initial PiCo setup outlined previously in this document. Please install smbus support for python to enable additional functionality. Simply run the following command (with an internet connection):

sudo apt-get install python-smbus

As the bootloader uses the Raspberry Pi Serial Port (RS232), it is mandatory to have it free on the Raspberry Pi (without any hardware occupying it). It is also important that you ensure that there is no software using it. As well If minicom has been used, please restart the Raspberry Pi, as minicom keeps the RS232 interface occupied.

The first task which is done by the UPS PIco after reset is to check if bootloader has been requested. If not, then the rest of the firmware runs. Otherwise, the UPS PIco lights the big RED LED and waits for the firmware upload from the Raspbeerry Pi.

There are two ways to invoke the bootloader mode and to upload the new firmware:

Automatic Initiation

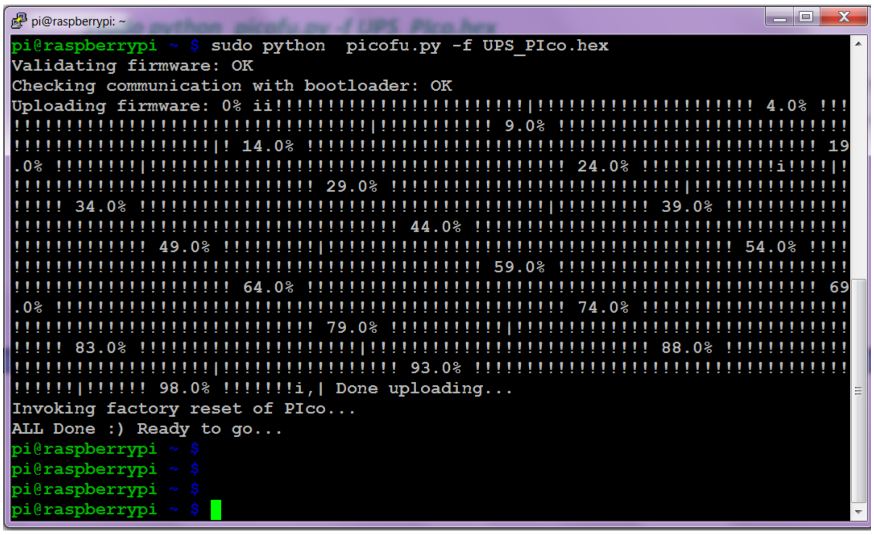
The boot-loader is invoked by running the following command line

sudo i2cset -y 1 0x6b 0x00 0xff

sudo python 9600_picofuHV3.0.py -v -f UPS_PIco_HV3.0_main.hex

The **UPS_PIco_HV3.0_main.hex** should be replaced with the name of the last firmware update, or the firmware you wish to use.

When firmware starts the upload procedure, the Orange User LED will lit, and then when firmware starts uploading the User Blue Led will Lit and UPS LED will be blinking.



Once complete the system with output **ALL Done :) Ready to go. . .**

We would recommend that you now shutdown your Pi and UPS PiCo completely in order to ensure that all changes are integrated. Once you've rebooted your system, you can check the UPS PiCo firmware version using the following command:

sudo i2cget -y 1 0x69 0x26

In this case the system should output **0xXX**, signifying that the firmware has updated correctly.

Manual Initiation

The UPS PiCo has the ability to invoke the bootloader manually, via the on board buttons. You can do this instead of using the automatic initiation outlined above.

It is very important to start/stop the Daemons Service when doing Hardware Reset of the PIco HV3.0 in order to avoid undefined situations with pulse train recognition procedure by the system. Resetting the PIco with Not Stopped the Daemon Service can cause an unexpected system shutdown (however without card corruption)

The following procedure needs to be followed:

- Press and hold the **UR** button
- Continue to hold the **UR** button, and press and hold the **F** button.
- Release the **UR** button, but keep holding the **F** button
- Release the **F** button
- The User Orange LED will light, and system will be able to receive the firmware update
- Then write the following command on the Raspberry Pi command line

sudo python 9600_picofuHV3.0.py -v -f UPS_PIco_HV3.0_main.hex

If within 16 second after boot loader initiation the firmware will be not initiated the UPS PIco HV3.0 will be reset to normal working conditions by internal Watch Dog mechanism. This has been implemented for security reasons, if firmware is uploaded remotely.

UPS PIco HV3.0 HAT LEDs

The **UPS Pico HV3.0 HAT** is equipped with 6 LEDs (that offers information about the **UPS Pico HV3.0 HAT** system status. Three of them are dedicated for user applications and can be handled by the **PICo** (I^2C) interface. One of them is **Orange**, the second one is **Green**, and the third is **Blue.** A detailed description of the system LEDs and their usage is provided on below table.

System LEDs Indications		
UPS LED		
	OFF	System is not running or is in Low Power Mode (only HW RTC is running)
	Lighting continuously	System (PIco + RPi) is booting or shutting down
	Blinking every 400 ms for 400 ms	System (PIco + RPi) is running on cable powering (after booting time)
	Blinking every 1200 ms for 400 ms	System (PIco + RPi) is running on battery powering
BAT LED		
	OFF	Battery level is **above** warning thresholds: - For LiPO Battery **3.5V** - For LiFePO4 **2.95V**
	Lighting continuously	Battery level is **below** warning thresholds: - For LiPO Battery **3.5V** - For LiFePO4 **2.95V**
CHG LED		
	OFF	Battery is not Charged
	Lighting continuously	Battery is Charged (and current is flowing to the battery) If battery is Full, even if Charger is ON, current is not flowing to the battery, so CHG LED is OFF
HOT LED (INF LED)		
FAN LED		
	OFF	FAN is not running
	Lighting continuously	FAN is running
EXT LED		
	OFF	External Cable powering is disconnected (7-28VDC)
	Lighting continuously	External Cable powering is connected (7-28VDC)

Accessing of the User LEDs can be done by the following **PIco** Commands.

Example of use

sudo i2cset -y 1 0x6b 0x09 0x01 for ON of the Orange LED

sudo i2cset -y 1 0x6b 0x09 0x00 for OFF of the Orange LED

sudo i2cset -y 1 0x6b 0x0A 0x01 for ON of the Green LED

sudo i2cset -y 1 0x6b 0x0A 0x00 for OFF of the Green LED

sudo i2cset -y 1 0x6b 0x0b 0x01 for ON of the Blue LED

sudo i2cset -y 1 0x6b 0x0b 0x00 for OFF of the Blue LED

0x09	User LED Orange	Byte	Common	R/W	User LED Orange **ON - Write:** 0x01 User LED Orange **OFF - Write:** 0x00
0x0A	User LED Green	Byte	Common	R/W	User LED Green **ON - Write:** 0x01 User LED Green **OFF - Write:** 0x00
0x0B	User LED Blue	Byte	Common	R/W	User LED Blue **ON - Write:** 0x01 User LED Blue **OFF - Write:** 0x00

UPS Pico HV3.0 HAT Buttons

System Buttons

The **UPS Pico HV3.0 HAT** is equipped with 6 buttons that can be used in various ways. Three of them are dedicated for user applications and can be handled by user through the **PICo** (I^2C) interface or **@commands** (RS232) system, all other are specific for various **UPS Pico HV3.0 HAT** functionalities. All of them can be used for some start-up functionalities when **UPS Pico HV3.0 HAT** is reset. A detailed description of all buttons and their usage is provided on below table.

It is very important to start/stop the Daemons Service when doing Hardware Reset (UR) of the PIco HV3.0 in order to avoid undefined situations with pulse train recognition procedure by the system. Resetting the PIco with Not Stopped the Daemon Service can cause an unexpected system safe shutdown (however without card corruption)

Button	Description	Usage	Additional Functionalities
RR	**R**aspberry Pi® Hardware **R**eset	Make Raspberry Pi Hardware Reset when pressed. To be used need installed (soldered) the Gold Plated Reset Pin. **NOTE1**: Resetting of the Raspberry Pi®, can corrupt files on the SD card if used **NOTE2**: Resetting of the Raspberry Pi®, does **not** affect the **UPS PIco** (including PIco RTC)	NONE
UR	**U**PS Pico HV3.0 HAT Hardware **R**eset	Make the **UPS Pico HV3.0 HAT** Hardware Reset when pressed. **NOTE1**: Resetting of the **UPS PIco** does not reset the	When pressed with combination with other buttons activate various start-up functionalities. The procedure is to press first the **UR** button, and then another one, then release the **UPSR** button and then release the other button (**A, B, C**).

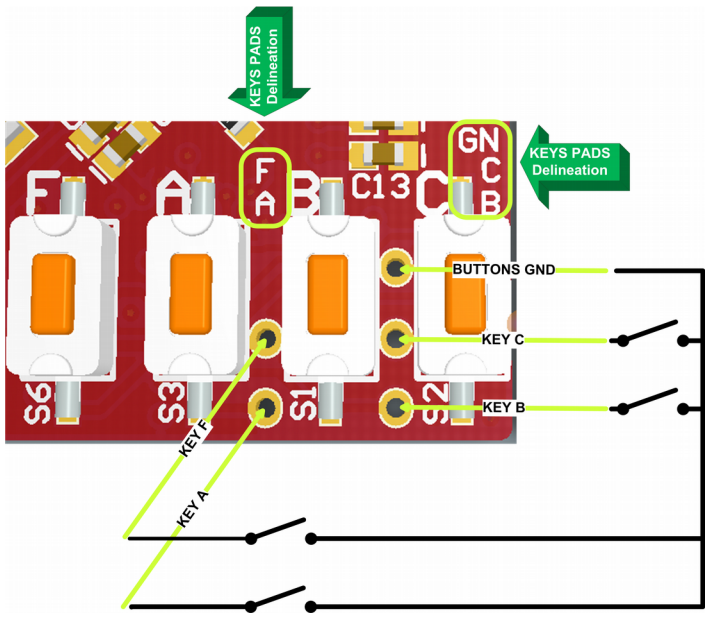
		Raspberry Pi® **only** if cable powered. **NOTE2**: Resetting of the **UPS PIco** does **NOT reset** the Integrated Hardware RTC.	
F	**F**ile Safe Shut Down (FSSD)	When pressed initiate the File Safe Shutdown Procedure. If Raspberry Pi® + **UPS Pico HV3.0 HAT** system is battery powered, after FSSD finished **UPS Pico HV3.0 HAT** will cut the power. Pressed again (need to have installed the Gold Plated Reset Pin for the restart option), start the Raspberry Pi® + **UPS Pico HV3.0 HAT** system again. In the battery powered System can be used as ON/OFF (files safe) button	When used with **UR** button, invokes the bootloader (light the Red User LED). The bootloader can be invoked also from the PICo interface.
A	**User Key A**	Can be used for User Application – Read the status via PICo (I^2C) or RS232 interface	NONE
B	**User Key B**	Can be used for User Application – Read the status via PICo (I^2C) or RS232 interface	When used with **UR** button, Set the **Hardware RTC** to default values
C	**User Key C**	Can be used for User Application – Read the status via PICo (I^2C) or RS232 interface	When used with **UR** button, sets system default values.

User Buttons (Keys)

User buttons **A**, **B** and **C** in the **UPS Pico HV3.0 HAT** are analog buttons, that means when pressed change the level voltage that is read by integrated A/D converter, value of its is interpreted by the firmware and the result is loaded to a proper system variable. The **F** button is a digital interrupt driven button and his value can not be read by the user

Each User Key and **FSSD** Key (buttons) has an additional THT pads that allows to be used by user for their own mounted buttons outside of the **UPS PIcoHV3.0 HAT or case**. Each of such user added buttons (keys) need just short to the system ground when pressed**.**

Reading of User Buttons values can be done by accessing the system variable by any software or by command line.



In order to make working external keys (buttons), user need to solder cables to the THT key pads. It is not recommended to use a very long cables (due to analog implementation of the keyboard), their length should not be longer than 100 – 150 mm. However we never tested longer cable and if user need to use longer cable should test it on their site. To make external keys workable user need to short each one with **GN THT** pad when pressed. In example if user need to have external access to the FSSD **(F)** button, need to install button that short the **F** pad with the **GN** pad when pressed. Similar approach should be followed with other keys. Above picture

show how external key (buttons) need to be connected. It is not needed to connect all of them.

The **key** register holds the latest value of pressed key, so user need to write i.e. **0x00** after reading, in order to recognize that new key has been pressed (when pressed again, even the same key). Current implementation require timed (pooling) reading of this register to recognize that a key has been pressed. In the future it is planned to implement interrupt driven handler for this and other functionalities.

Example of use

sudo i2cget -y 1 0x69 0x1A should return 1, 2 or 3

The key value will remain in the register until new value will be written (when key (new one or the same) will be pressed. Therefore, user should write zero to this register after read in order to recognize and read again the new (or the same) key if (when) pressed.

sudo i2cset -y 1 0x69 0x1A 0x00

UPS PIco HV3.0 Battery Type Selection

The **UPS PIco HV3.0** is supporting 2 different chemistry battery types:

- the **LiPO**
- and the **LiFePO4**

Both chemistry batteries are available in 2 capacities. Therefore the UPS PIco HV3.0 can be supplied with the following batteries:

- The standard LiPO battery 450 mAh which comes with the UPS PIco HV3.0
- The enhanced LiPO battery with capacity 4000 mAh
- The enhanced LiPO battery with capacity 8000 mAh
- The enhanced LiFePO4 battery with capacity 4000 mAh
- The enhanced LiFePO4 battery with capacity 8000 mAh

Batteries with different chemistry offers different unique features, and needs to be specified on the system setup when changed. The core differences between both chemistry batteries are listed here below. The battery type setting declare the chemistry and not the capacity of the battery. Declaration of the battery chemistry is needed due to different threshold voltages and slightly different charging algorithm. It is mandatory to have declared a proper battery chemistry for a proper system

functionality. The default battery chemistry is the LiPO and if not changed it is not needed to change the declaration. Only if the user use the enhanced LiFePO4 batteries it is need to proceed with changed of the battery chemistry declaration.

0x6B -> UPS PIco Module Commands

0x07	battype	Byte	Common	R/W	Defines used battery chemistry type: 0x46 – LiFePO4 (ASCII : F) used in version Stack/TopEnd 0x51 – LiFePO4 (ASCII: Q) used in version Plus 0x53 – LiPO (ASCII: S) used in version Stack/TopEnd 0x50 – LiPO (ASCII: P) used in version Plus **Other codes are not allowed**

Example of use

sudo i2cset -y 1 0x6b 0x07 0x46 ***LiFePO4 (ASCII : F) used in version Stack/TopEnd***

sudo i2cset -y 1 0x6b 0x07 0x51 ***LiFePO4 (ASCII: Q) used in version Plus***

sudo i2cset -y 1 0x6b 0x07 0x53 ***LiPO (ASCII: S) used in version Stack/TopEnd***

sudo i2cset -y 1 0x6b 0x07 0x50 ***LiPO (ASCII: P) used in version Plus***

Caution: The **UPS PIco HV3.0** has declared always the default battery chemistry type, and when firmware update is executed the battery chemistry is always changed to PCB defaults, therefore it is need to re-declare the battery type after firmware update to the used one by the system. Some industrial customers have default declared the LiFePO4 in their systems.

UPS PIco HV3.0 System Monitoring

The **UPS PIco HV3.0** offer to user an extended monitoring system that measure and report a large number of system parameters trough installed sensors. Each sensor is reporting the **UPS PIco HV3.0** status via dedicated variables. In addition there is access to the integrated 12 bits 3 x A/D converters. All monitoring system data are collected in a single entity called **PIco Status** and exists at the I^2C address **0x69**. Detailed specifications for each variable (register) as also examples are provided in next pages.

UPS PIco HV3.0 12 bit A/D converers

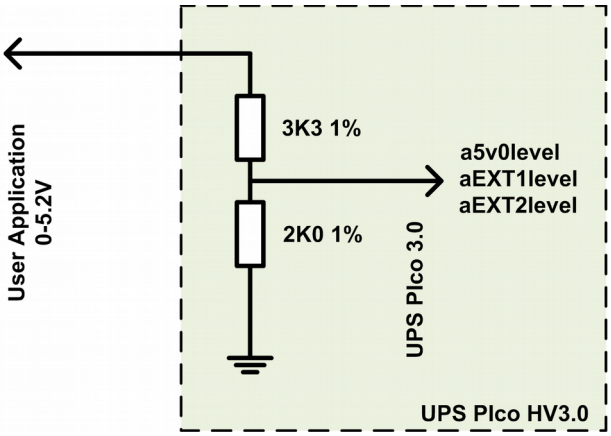
The **UPS PIco HV3.0** is equipped with 3 x 12 bits A/D converters. Access to their conversion data is possible via dedicated registers placed at the I^2C address **0x69 (0x14, 0x16, 0x18)**. Those A/D converters read continuously data every 250 uS with conversion time of 3.5 uS per sample. However due to implemented low noise software simple filtering in the firmware the effective rate data rate is around of 0.001 sec per reading (each A/D register values is refreshed every 1ms).

Each of the A/D converters is pre scaled to measure voltage 0-5.2V with implemented on the **UPS PIco HV3.0** resistor divider. However there is also a possibility for the user (if use Terminal Block PCB or additional external resistor) to use two of them as pre scaled of 0-10V, 0-20V, or 0-30V. These two A/D converters are named **aEXT1level** and **aEXT2level**.

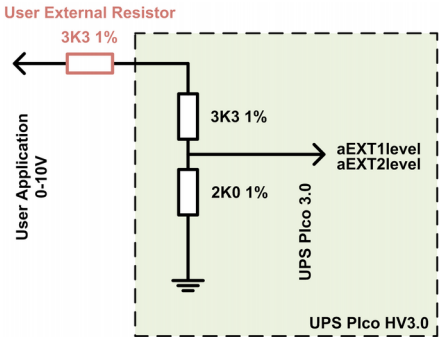
Due to electrical requirements of the integrated A/D converters the impedance is set to low values, therefore some sensors can not be read properly as could require higher impedance of A/D converter interface. Therefore it is recommended to use on such cases Voltage Follower that converts the sensor high impedance to A/D converters lower one.

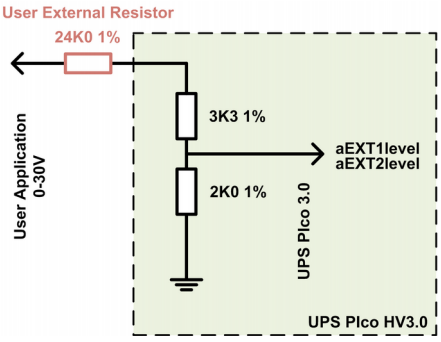
This functionality (of the **Voltage Follower**) has been implemented on the Terminal Blocks PCB, where on one of the A/D's converters (the **a5v0level)** a **Voltage Follower** has been assigned and allows to convert high impedance of any possible used sensor to low impedance on the A/D side.

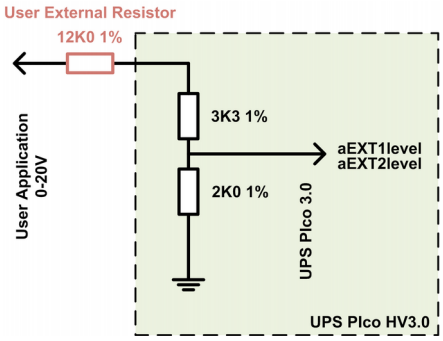
The basic circuit of all A/D converters (the resistor dividers) are shown here below, it is same to all implemented A/D converters in the **UPS PIco HV3.0.**



If user decide to use Higher Voltage Interface as pre scaled of 0-15V, 0-20V, or 0-30V the Terminal Blocks PCB should be used, or an additional Resistors need to be added externally, like in the below picture. In addition the register **setA_D (0x08)** at address **0x6b** should be set to a proper value according the below table.







Caution: The **UPS PIco HV3.0** has implemented an ESD protection on each A/D converter input. This protection protect the system from ESD discharges and **not** from continuously high voltage applied.

Therefore it is very important if High Voltage used, to be sure that a proper resistor(s) has been used. If smaller resistor(s) that required will be used, then the A/D input will be destroyed permanently, and very possible the whole UPS PIco HV3.0

Therefore user need to take care to be sure that a proper values of resistor(s) has been used.

setA_D Values	a5v0level Scale	AEXT1level Scale	AEXT2level Scale	AEXT1 Resistor	AEXT2 Resistor
0x00	5.2V	5.2V	5.2V	0K	0K

0x01	5.2V	5.2V	10V	0K	3K3
0x02	5.2V	5.2V	20V	0K	12K0
0x03	5.2V	5.2V	30V	0K	24K0

setA_D Values	**a5v0level Scale**	**AEXT1level Scale**	**AEXT2level Scale**	**AEXT1 Resistor**	**AEXT2 Resistor**
0x00	5.2V	5.2V	5.2V	0K	0K
0x10	5.2V	10V	5.2V	3K3	0K
0x20	5.2V	20V	5.2V	12K0	0K
0x30	5.2V	30V	5.2V	24K0	0K

Any combination of data provided on above table is allowed. The register **setA_D** is 8 bit. The 4th MSB bits are responsible for the **AEXT1level** pre-scale, and the 4th LSB bits are responsible for the **AEXT2level** pre-scale.

UPS PIco HV3.0 Serial Port(s)

The **UPS PIco HV3.0** is equipped with 2 serial ports. One of them is connected directly to the Raspberry Pi ® Serial Port, and the second one is available for user applications. Both Serial Ports are full programmable and the data rate can be set by user, as also can be enabled or disabled. In addition there is an internal routing option where one port can receive and send data over the other one. This allows connecting the Raspberry Pi® serial port to the external RS232 12V interface (via Terminals Blocks PCB) or to 5V tolerant without any additional Jumpers or Cables. In addition the second Serial port of the **UPS PIco HV3.0** can be used as a second serial port routed directly to the I2C interface (this option is not unlocked yet, and will be available within one of the next firmware update). The **UPS PIco HV3.0** by default is set OFF and Raspberry Pi® Serial port can be used for any other applications. If it is needed it can be set ON, as also the set the data rate. Setting the data rate sets it for both **UPS PIco HV3.0** serial ports.

After any firmware update the **UPS PIco HV3.0** Serial Ports(s) must be set again. It is done via PIco variable **rs232_rate**. The following settings are available:

Setting Value	Meaning
0x00	**UPS PIco HV3.0** Serial Port is **OFF** **Default value**
0x01	**UPS PIco HV3.0** Serial Port is **ON** and data rate is set to **4800** pbs
0x02	**UPS PIco HV3.0** Serial Port is **ON** and data rate is set to **9600** pbs
0x03	**UPS PIco HV3.0** Serial Port is **ON** and data rate is set to **19200** pbs
0x04	**UPS PIco HV3.0** Serial Port is **ON** and data rate is set to **38400** pbs
0x05	**UPS PIco HV3.0** Serial Port is **ON** and data rate is set to **57600** pbs
0x0F	**UPS PIco HV3.0** Serial Port is **ON** and data rate is set to **115200** pbs

Example of use

sudo i2cset -y 1 0x6b 0x02 0x00 ***Disable PIco RS232 and set tri state the TXD and RXD pins***

sudo i2cset -y 1 0x6b 0x02 0x05 ***Enable the PIco RS232 and set the data rate to 57600 bps***

sudo i2cset -y 1 0x6b 0x02 0x0F ***Enable the PIco RS232 and set the data rate to 115200 bps***

Ultra Short description of the UPS PIco HV3.0 Programmers Registers

The PICo (I^2C) Interface - Peripherals I2C Control Interface

The **P**eripherals **I**2C **Co**ntrol – The **PICo Interface** – is an implementation of **I2C** interface adapted to easy control of the peripherals connected to the Raspberry Pi® via simple command line or trough programming language. By using human understandable simple commands, control of the **UPS PIco HV3.0** peripherals is made extremely simple. Control at programming language level is also possible and easy. The core concept of the **UPS PIco HV3.0 interface** is that all peripheral device control and data exchange between it and Raspberry Pi® variables are common for the **I^2C interface** as also for the peripheral itself. Therefore any change of them by either party, Raspberry Pi® and the peripheral, causes immediate update and action.

Two types of variables are available:

- **Common**, where data are stored in the same place and any change on it will cause action on the **UPS PIco** Module
- **Mirror**, where are copy of data stored on internal variables of the **UPS PIco HV3.0** Module, they are protected, so changes on it will not implies the **UPS PIco HV3.0** Module functionality and will be overwritten immediately when **UPS PIco HV3.0** Module recognized changes on them

There have been implemented the following **PICo** addresses assigned to the following entities:

0x69 ->UPS PIco HV3.0 Module Status Registers Specification

Address	Name	Size	Type	R/W	Explanation
0x00	mode	Byte	Mirror	Read	Powering Mode – Read ONLY, Writing has no effect on the system and will be overwritten by UPS PIco HV3.0 with the new value **Read:** **0x01** - RPI_MODE (means cable powering mode USB or EPR) **0x02** - BAT_MODE
0x08	batlevel	Word	Mirror	Read	Means value of Battery Voltage in 10^{th} of mV in BCD format
0x0a	rpilevel	Word	Mirror	Read	Means value of Voltage supplying RPi on J8 5V Pin in 10^{th} of mV in BCD format
0x0c	eprlevel	Word	Mirror	Read	Means value of Extended Voltage supplying RPi on Extended Voltage input (7-28VDC) in 10^{th} of mV in BCD

					format
0x14	a5v0level	Word	Mirror	Read	Means value of the first A/D converter pre scaled to 5.2V. Higher voltage could not be supplied without a resistor divider. Readings are in 10th of mV in BCD format
0x16	a15v0level	Word	Mirror	Read	Means value of the first A/D converter pre scaled to 5.2V. Higher voltage could not be supplied without a resistor divider. Readings are in 10th of mV in BCD format. **If added an extra resistor or used as pre scaled to 10, 20 or 30V.**
0x18	a30v0level	Word	Mirror	Read	Means value of the first A/D converter pre scaled to 5.2V. Higher voltage could not be supplied without a resistor divider. Readings are in 10th of mV in BCD format. **If added an extra resistor or used as pre scaled to 10, 20 or 30V.**
0x1a	key	Byte	Common	R/W	User Kay Pressed information **Read: 0x01** – Pressed key **A** **Read: 0x02** – Pressed key **B** **Read: 0x03** – Pressed key **C** **Write: 0x00** – Reset (clear) after the current reading and prepare for the next one.
0x1b	ntc	Byte	Mirror	Read	Temperature in Celsius degree of the embedded NTC1 sensor placed on the top of PCB. Values in BCD format.
0x1c	TO92	Byte	Mirror	Read	Temperature in Celsius degree of the TO-92 sensor placed on the bottom of PCB. It is valid only if this sensor is soldered. It is available in the PIco Fan Kit. Values in BCD format.
0x20	charger	Byte	Mirror	Read	Information about charger IC status. **Read: 0x00** – Charger IC is OFF and battery is not charged **Write: 0x01** – Charger IC is ON and battery is charged For Version UPS PIco HV3.0 Stack/TopEnd the charging current is fixed to 300 mA. For Version UPS PIco HV3.0 Plus the charging current is dynamically changed based on powering conditions on micro USB powering input or External Powering Input. The charging current on that version is from 100 mA – 800 mA. With maximum of the system to be 1200 mA (will be activated in the future). If the **charger** register is set ON, meant that charger circuits has been activated, however if battery is really charged depends to the internal conditions of the charger IC. When current is flowing to the battery (charging) then CHG LED is lighting continuously. It is possible that **charger** register can be read as 0x01, but CHG LED will be off, because system set to charge battery however buttery is full, so no current is flowing. The CHG LED always shows if current is flowing to battery or not. This is valid for all batteries chemistry types.
0x22	pico_is_running	Word	Mirror	Read	It is a 16 bit unsigned variable that value of it, is changing every 1 ms within the main loop of the firmware. Reading two times of this variable must

					return a different value (with interval longer than 1 ms), if not, means that system hangs-up, and need to be reset, if not restarted by other PIco protection internal mechanism (watch-dog, and supervising watch dog). As these protection mechanisms are always restarting the system when something goes wrong, reason of existence of this variable is just to confirm to the remote user that everything is working well and give feedback to the remote user that system is running properly. As it is a mirror variable, writing to it nothing change, will be again re-written with the newer internal value.
0x24	pv				PCB Version - current available versions: A
0x25	bv				Bootloader Version - current available versions: S - BL_PIco HV 3.0A Stack/TopEnd default LP Battery F - BL_PIco HV 3.0A Stack/TopEnd default LF Battery P - BL_PIco HV 3.0A Plus default LP Battery Q - BL_PIco HV 3.0A Plus default LF Battery
0x26	fv				Firmware Version: current 0x24 dated 22/01/2017

0x6A -> UPS PIco Hardware RTC Registers Direct Access Specification

Address	Name	Size	Type	R/W	Explanation
0x00	seconds	Byte	Mirror	Read	seconds in BCD
0x01	minutes	Byte	Mirror	Read	minutes in BCD
0x02	hours	Byte	Mirror	Read	hours in BCD
0x03	wday	Byte	Mirror	Read	week day in BCD
0x04	mday	Byte	Mirror	Read	month day in BCD
0x05	month	Byte	Mirror	Read	month in BCD
0x06	year	Byte	Mirror	Read	year in BCD

0x6B -> UPS PIco Module Commands

Address	Name	Size	Type	R/W	Explanation
0x00	pico_state	Byte	Common	R/W	**Write: 0xcc** – Unconditional File Safe Shutdown and (and Power OFF when battery powered) **Write: 0xdd** - then restore factory defaults Will stay in the values of 0xdd until factory defaults restored, and then will be set to 0x00 **Write: 0xee** - Reset the UPS PIco CPU, it cause start-up values i.e. RTC will be set to 01/01/2000 **Write:** 0xff - Call the UPS PIco Bootloader, **Orange** Led will be light. Recover from this state can be done **only** by pressing the RST button, new firmware upload or automatically after 16 seconds if nothing happen. All interrupts are disabled during this procedure. It should be used with RPi Uploading firmware script. Use it very carefully and only when is needed – when firmware uploading. Do not play with it; this is not toy functionality. **Powering of the pair UPS PIco+RPi must be done via RPi micro USB socket during boot loading process due to following UPS PIco Resets after firmware uploading or when returning from this mode.** **Due to required protection for the RPi from the unconditional reset (files corruption), it is not possible to enter to this mode when system is powered in a different way than in RPI Powering Mode.**
0x01	bat_run_time	Byte	Common	R/W	**On Battery Powering Running Time when cable power loses or not exist. After that time a File Safe Shut Down Procedure will be executed and System will be shut downed without restart. Battery power will be disconnected. System is in sleep mode (LPR) and RTC is running.** **If Raspberry Pi cable power returns again system will be start automatically.** **If during the sleep mode (LPR) the F button will be pressed for longer time than 2 seconds (with battery or cable powering) Raspberry Pi will re-start again.** **Value of 0xff (255) disable this timer, and system will be running on battery powering until battery discharge to 3.4V for LP battery and 2.8V fro LF Battery type.** **Factory default value is 60 seconds** **Each number represent 1 minute of Battery Running. Default Value is 0, and the highest Value is 0xFE. If user will enter i.e. 2, the Battery**

					Running time will be 60 seconds + 2 x 60 seconds = 180 seconds. After that time system will be shutdown. If user after that will press again F button system will restart and run for 180 seconds again and then shutdown. **Read:** Anytime, Return actual **fssd_timeout** value **Write:** 0x00 – 0xFF **Any change on this register will cause immediate writing of the new value to the PIco EEPROM**
0x02	rs232_rate	Byte	Common	R/W	Writing to this register sets the UPS PIco HV3.0 Serial Port to the following settings: **0x00 - UPS PIco HV3.0** Serial Port is OFF **Default value** **0x01 - UPS PIco HV3.0** Serial Port is ON and data rate is set to 4800 pbs **0x02 - UPS PIco HV3.0** Serial Port is ON and data rate is set to 9600 pbs **0x03 - UPS PIco HV3.0** Serial Port is ON and data rate is set to 19200 pbs **0x04 - UPS PIco HV3.0** Serial Port is ON and data rate is set to 34600 pbs **0x05 - UPS PIco HV3.0** Serial Port is ON and data rate is set to 57600 pbs **0x0F - UPS PIco HV3.0** Serial Port is ON and data rate is set to 115200 pbs
0x07	battype	Byte	Common	R/W	Defines used battery chemistry type: 0x46 – LiFePO4 (ASCII : F) used in version Stack/TopEnd 0x51 – LiFePO4 (ASCII: Q) used in version Plus 0x53 – LiPO (ASCII: S) used in version Stack/TopEnd 0x50 – LiPO (ASCII: P) used in version Plus **Other codes are not allowed**
0x08	setA_D	Byte	Common	R/W	Defines the pre scaler of the **AEXT1level** and the **AEXT2level** registers. The 4th MSB bits are responsible for the **AEXT1level** pre-scale, and the 4th LSB bits are responsible for the **AEXT2level** pre-scale. **Read:** Anytime, Return actual **setA_D** value **Write:** 0x00 – 5.2V prescale for the **AEXT2level** **Write:** 0x01 – 10V prescale for the **AEXT2level** **Write:** 0x02 – 20V prescale for the **AEXT2level** **Write:** 0x03 – 30V prescale for the **AEXT2level** **Write:** 0x00 – 5.2V prescale for the **AEXT1level** **Write:** 0x10 – 10V prescale for the **AEXT1level** **Write:** 0x20 – 20V prescale for the **AEXT1level** **Write:** 0x30 – 30V prescale for the **AEXT1level** **(not implemented in the current version of firmware yet)**
0x09	User LED Orange	Byte	Common	R/W	**User LED Orange ON - Write:** 0x01 **User LED Orange OFF - Write:** 0x00
0x0A	User LED Green	Byte	Common	R/W	**User LED Green ON - Write:** 0x01 **User LED Green OFF - Write:** 0x00
0x0B	User LED Blue	Byte	Common	R/W	**User LED Blue ON - Write:** 0x01 **User LED Blue OFF - Write:** 0x00
0x0C	brelay	Byte	Common	R/W	**Zero Power Bi Stable Relay**

					Write: 0x01 Set **Write:** 0x01 Reset
0x0D	bmode	Byte	Common	R/W	**Integrated Sounder Mode** **Read:** Anytime, Return actual **bmode** value **Write:** 0x00 – Unconditional Disable the Sounder **Write:** 0x01 – Unconditional Enable the Sounder
0x0E	bfreq	Word	Common	R/W	**Frequency of sound in Hz**
0x10	bdur	Byte	Common	R/W	**Duration of sound in 10th of ms (10 = 100 ms)**
0x11	fmode	Byte	Common	R/W	**Integrated Fan Running Mode** **Read:** Anytime, Return actual **fmode** value **Write:** 0x00 – Unconditional Disable the FAN with selected speed from the **fspeed** **Write:** 0x01 – Unconditional Enable the FAN FAN with selected speed from the **fspeed** When UPS PIco is going down to the LPR mode, the FAN is automatically disabled, and enabled again when the UPS PIco returns to normal work
0x12	fspeed	Byte	Common	R/W	**Integrated Fan Speed** **Read:** Anytime, Return actual **fspeed** value **Write:** 00 – Selected speed when ON is 0% (not running) **Write:** 100 – Selected speed when ON is 100% (full speed running) **Any other (0-100) number is allowed and means % of speed and current consumption**
0x13	fstat	Byte	Mirror	Read	**Read:** Anytime, Return actual **if FAN** is actually running or not (for remote users)

Events Triggered RTC Based System Actions Scheduler

Not Unlocked in the firmware yet

0x6c -> Start Time Stamp

Address	Name	Size	Type	R/W	Explanation
0x00	active	Byte	Common	R/W	Activation Stamp 0x00 not active(Start), 0xff active (Start)
0x01	minute	Byte	Common	R/W	Starting Minute of hour in BCD - 2 digits (0-59) i.e. 22
0x02	hour	Byte	Common	R/W	Starting Hour of the Day in BCD - 2 digits (0-23) i.e. 22
0x03	mday	Byte	Common	R/W	Starting Day of the Month in BCD - 2 digits (1-31) i.e. 22
0x04	month	Byte	Common	R/W	Starting Month in BCD - 2 digits (1-12) i.e. 12
0x05	year	Byte	Common	R/W	Starting Year in BCD - 2 digits (0-99) i.e. 16

0x6d -> Running Time Stamp

Address	Name	Size	Type	R/W	Explanation
0 or **0x00**	D_repetition	Byte	Common	R/W	Days repetition in BCD 2 digits (0-99) every XX days: 00 – not repeated (only once if all repetitions are 0) 1-99 – every 1-99 days i.e. 7 means every week (i.e. every Monday) i.e. 10 means every 10 days i.e. 1 means every day
1 or **0x01**	H_repetition	Byte	Common	R/W	In BCD 2 digits (0-23) every XX days: 00 – not repeated (only once if all repetitions are 0) 1-23 – every 1-23 hours i.e. 7 means every 7 hours
2 or **0x02**	M_repetition	Byte	Common	R/W	In BCD 2 digits (1-59) every XX minutes: 00 – not repeated (only once if all repetitions are 0) 1-59 – every 1-59 minutes i.e. 7 means every 7 minutes
3 or **0x03**	H_Duration	Byte	Common	R/W	In BCD 2 digits hours 1-24 hours
4 or **0x04**	M_Duration	Byte	Common	R/W	In BCD 2 digits minutes 0-59

0x6e -> Events Stamp (NOT implemented yet)

Address	Name	Size	Type	R/W	Explanation

0x6f -> Actions Stamp (NOT implemented yet)

Currently Implemented only Power Up System – permanently selected.

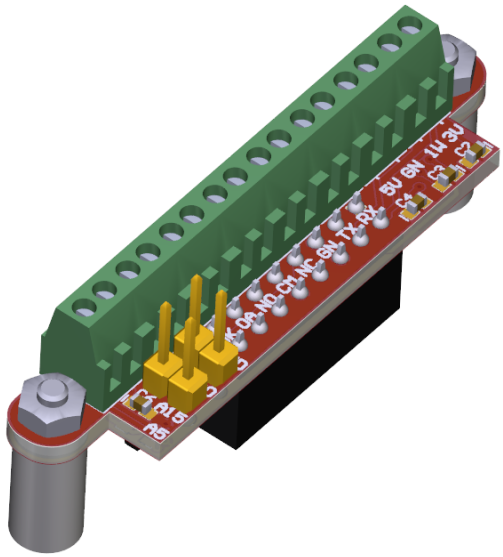
Future implementation: FAN, Charger, Relay

Address	Name	Size	Type	R/W	Explanation

Terminal Blocks PCB (Optional)

UPS PIco Terminal Block PCB

Designed for the UPS PIco HV3.0



User Guide

Designed for the Raspberry Pi® 3

Compatible with

Raspberry Pi® 2, Pi Zero, A+, B+,

HAT Compliant

Designed and Manufactured by PiModules and ModMyPi

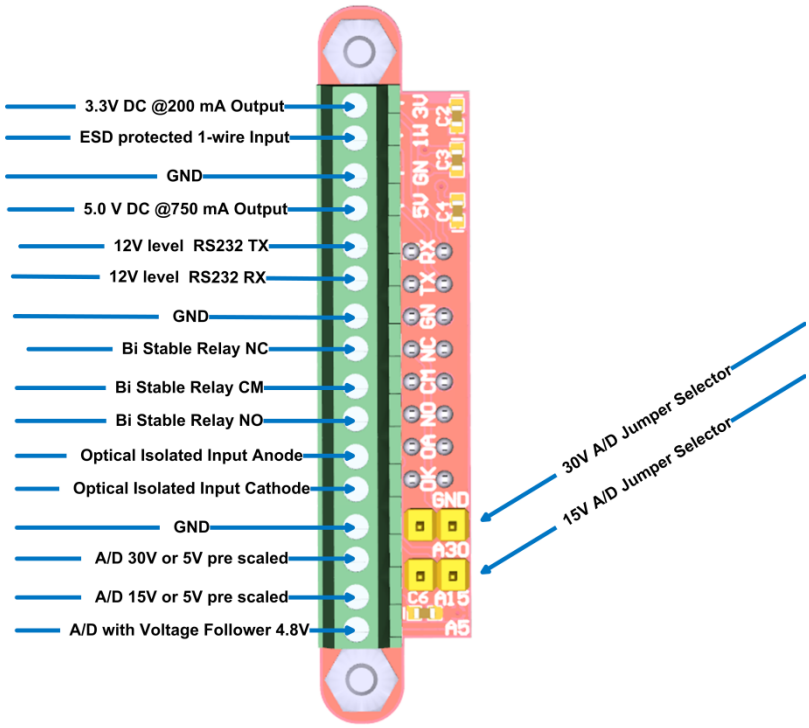
www.pimodules.com www.modmypi.com

Introduction

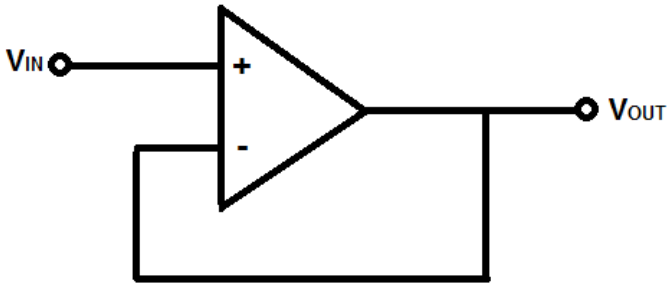
The **UPS PIco HV3.0 Terminal Blocks PCB** is an advanced Terminal Blocks PCB that adds a wealth of extra functionality and development features to the innovative **UPS PIco HV3.0**!

The following listed features are offered by the **UPS PIco HV3.0 Terminal Blocks PCB**:

- Terminal Block Connectivity on Independent from Raspberry Pi, and battery backed-up 3.3 V 200 mA supply (available also when Raspberry Pi is not powered)
- Terminal Block Connectivity on ESD protected 1-wire interface
- Terminal Block Connectivity on Independent from Raspberry Pi and battery backed-up 5V source 750 mA (available also when Raspberry Pi is not powered)
- 12V RS232 Interface Level Converter connectable to the Raspberry Pi Primary Serial Port or Independent Secondary Serial Port offered by UPS PIco HV3.0A with Terminal Block Connectivity
- Terminal Block Connectivity on Auxiliary interface to the bi-stable (zero power) Relay offered by the UPS PIco HV3.0A
- Terminal Block Connectivity on Optical Isolated Interface – readable as digital or analog input offered by the UPS PIco HV3.0A
- Terminal Block Connectivity on ESD Protected 12 bit A/D converters pre-scaled to: 5V, 15V and 30V (user selectable) accessed by I^2C on Raspberry Pi®
- Voltage Follower and scaled of 0 - 4.8V A/D with Terminal Block Connectivity

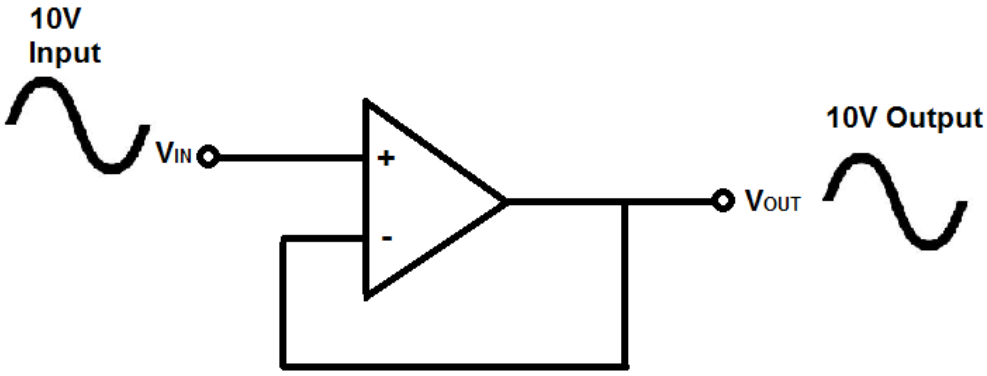

3.3V DC @200 mA Output
ESD protected 1-wire Input
GND
5.0 V DC @750 mA Output
12V level RS232 TX
12V level RS232 RX
GND
Bi Stable Relay NC
Bi Stable Relay CM
Bi Stable Relay NO
Optical Isolated Input Anode
Optical Isolated Input Cathode
GND
A/D 30V or 5V pre scaled
A/D 15V or 5V pre scaled
A/D with Voltage Follower 4.8V
30V A/D Jumper Selector
15V A/D Jumper Selector

What is a Voltage Follower?



A **voltage follower** (also called a unity-gain amplifier, a buffer amplifier, and an isolation amplifier) is a op-amp circuit which has a voltage gain of 1.

This means that the op amp does not provide any amplification to the signal. The reason it is called a voltage follower is because the output voltage directly follows the input voltage, meaning the output voltage is the same as the input voltage. Thus, for example, if 10V goes into the op amp as input, 10V comes out as output. A voltage follower acts as a buffer, providing no amplification or attenuation to the signal.



What is the Purpose of a Voltage Follower

One may ask then, what is the purpose of a voltage follower? Since it outputs the same signal it inputs, what is its purpose in a circuit? This will now be explained.

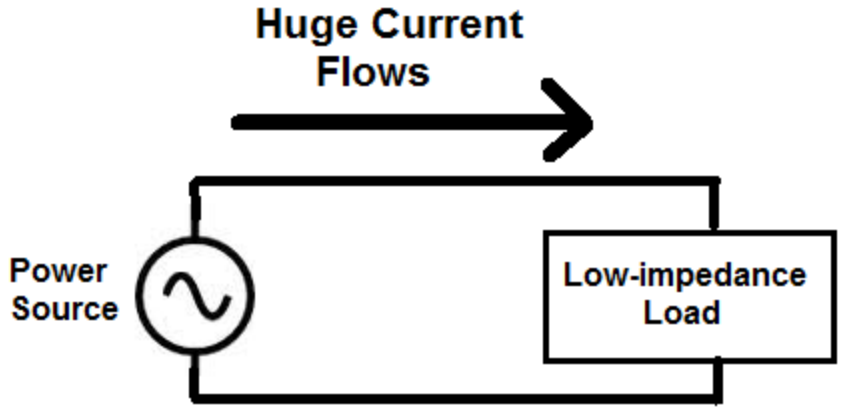
An op amp circuit is a circuit with very high input impedance. This high input impedance is the reason voltage followers are used. This will now be explained.

Voltage Followers Draw Very Little Current

When a circuit has a very high input impedance, very little current is drawn from the circuit. If you know ohm's law, you know that current, I=V/R. Thus, the greater the resistance, the less current is drawn from a power source. Thus, the power of the circuit isn't affected when current is feeding a high impedance load.

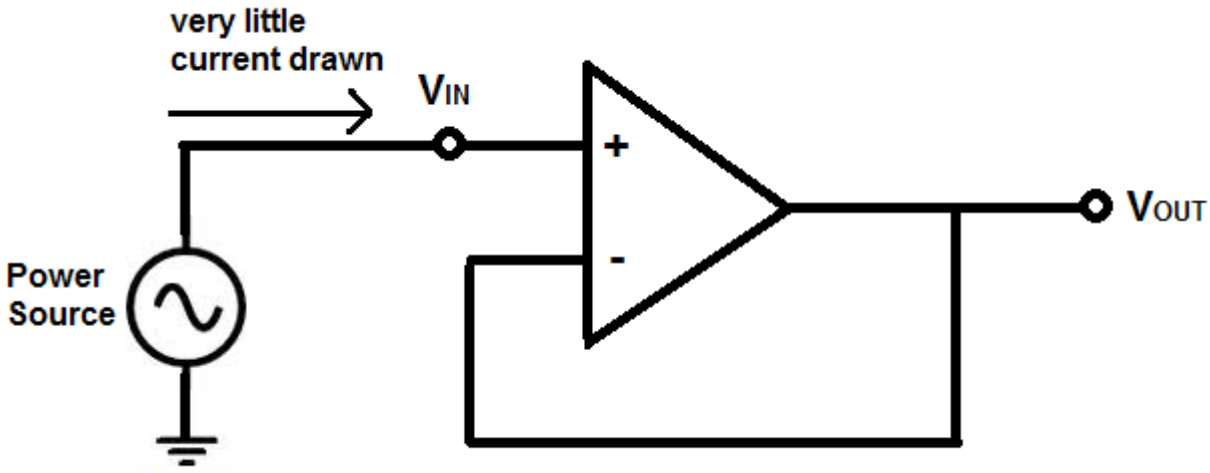
Let's look at both illustrations below:

The below circuit is a circuit in which a power source feeds a low-impedance load.



In this circuit above, the load demands and draws a huge amount of current, because the load is low impedance. According to ohm's law, again, current, I=V/R. If a load has very low resistance, it draws huge amounts of current. This causes huge amounts of power to be drawn from the power source and, because of this, causes high disturbances and use of the power source powering the load.

Now let's look at the circuit below, connected to an op-amp voltage follower:



This circuit above now draws very little current from the power source above. Because the op amp has such high impedance, it draws very little current. And because an op amp that has no feedback resistors gives the same output, the circuit outputs the same signal that is fed in.

This is one of the reasons voltage followers are used. They draw very little current, not disturbing the original circuit, and give the same voltage signal as output. They act as isolation buffers, isolating a circuit so that the power of the circuit is disturbed very little.

Voltage Followers Consume Practically All of the Voltage from a Voltage Divider Circuit

So current, as explained above, is one of the reasons voltage followers are used. They simply don't draw a lot of current, so they do not load down the power source.

Another reason voltage followers are used is because of voltage dividers. This again deals with ohm's law. According to ohm's law, voltage= current x resistance (V=IR).

In a circuit, voltage divides up or is allocated according to the resistance or impedance of components.

Because an op amp has a very high input impedance, the majority of voltage will fall across ut, since it's so high impedance. So it's very valuable when used in a voltage divider because it guarantees that it will receive the majority of voltage if placed in a voltage divider circuit.

This will now be illustrated so you can see.

So let's say we have a circuit shown below which represents a voltage divider with a load attached to the output.