2022

UPS PIco HV4.0 B/C/D HAT

www.pimodules.com

1/1/2022

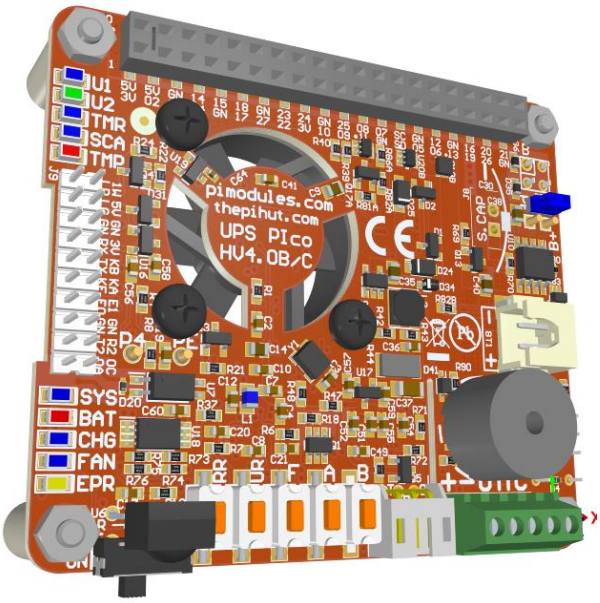
UPS PIco HV4.0 B/C/D HAT

Versions: Stack/ Advanced/ Passive PoE

The Ultimate Power Management System with RTC, Enhanced Peripherals and I²C control Interface
Intelligent Mobile Power Bank

Ultra-High Current Extended Buck Supply of 4.0A

External Powering up to 32 VDC and up to 24V DC Passive PoE



User Guide

Especially designed for the Raspberry Pi® 4 Model B

Compatible with

Former models of Raspberry Pi 3 B/B+

Document Revisions

Version	Date	Modified Sections	Comments
N.A.	01/01/2022	N.A.	First Preliminary Public Document Release
N.A.	11/10/2022	SOFTWARE SETUP FOR UPS PICO HV4.0	Updated Automatic setup script
N.A.	11/10/2022	Associated Software and Scripts	Added Associated Software Entity to current version of manual - links

Unlocked Firmware Features

Version	Date	Unlocked Features
Initial	Initial	

Firmware Bug Fixes/Add-on

Version	Date	Unlocked Features
Initial	Initial	
010600	01062022	Improved various non-significant internals
010601	01062022	Corrected Bug with self-restarting
010602	01062022	Corrected Bug with EPR supply non starting
010603	01062022	Added Description and Activated Handler for Various Powering Modes
010604	01062022	Added The Enhanced 'F' Key Behaviors
010605	01062022	Corrected and enhanced BAT, SCAP and MIXED Modes
120600	12062022	Improved various non-significant internals
120601	12062022	Activated Serial Port Support and their Enhanced Features
120602	12062022	Activated DMA handler on both Serial Ports instead of former Interrupt based

Daemons Bug Fixes/Add-on

Version	Date	Unlocked Features
Initial	Initial	

Schematics/PCB Updates

Version	Date	Unlocked Features
Initial	Initial	

Table of Contents

System Overview

Introduction

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** is an The **U**ltimate **P**ower **Management S**ystem **HAT** that adds a wealth of innovative powerful and development features to the Raspberry Pi® microcomputer! It has been designed especially for the Raspberry Pi® 4 and considering all enhanced power and cooling requirement of the Raspberry Pi® 4 models, however in addition it is still compatible with most of former Raspberry Pi models. The core functionality of the **UPS PIco HV4.0B/C/D** is to protect and automatically shut-down your Raspberry Pi if there is a cable power failure and can be set to automatically monitor and reboot your Pi once cable power has been restored! However, it is only a small part of plenty and powerful functionalities that are implement on this small HAT.

If used as Mobile Power Bank it is equipped with an Intelligent Externally Accessed (with Files Safe Shutdown functionality) Power Slide Switch that allows to safety System Switch ON/OFF whenever you like, without worrying about files corruption as it is always properly shutdown the system before battery will be disconnected (keep battery connected until system shutdown)!

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** features **a 5.2V 3.2A total current output** when battery powering, designed especially for use on the latest **Raspberry Pi® 4 Model B** as also most of the former Raspberry Pi ® modules!

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** offers now 2 User Programmable Keys, 2 separate User programable LEDs with different colors, support for multiple and different chemistry of a high-capacity batteries, Support for Super Capacitor, SPDT relay, as also 2 x A/D 12-bit converters with pre-adjustable readings to 5.2V. Now, with number of embedded sensors (inbound current, outbound current on high (32VDC) and low (5VDC) powering voltages, temperature, voltages), true 5V 1-wire interface, standard high voltage RS232 interface and many, many additional features!!

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** is standard equipped with a 450mAh 15C LiPO battery (able to supply up to 6.5A) specially designed to enable safe shutdown when cable power cuts. Additionally, this can be easily upgraded to the extended 1500mAh, 4000mAh, 8000mAh or even 12000mAh (on Special Order) capacities, which enables prolonged use of a Raspberry Pi for **more than 24 hours** without a power supply connected! Embedded charger is automatically adjusting charging current to existing cable power supply capabilities and can charge up to 1A current continuously.

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** design support now batteries with different chemistry like: LiPO, Li-Ion, LiFePO4, NiMH and SAL. 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 or 10 years lifetime!!

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** also support a unique feature the **Supercapacitor** power backup in various configurations **100F, 300F, 500F** and finally **800F**. The **Supercapacitor** power backup has a plenty and unique feature. First, the lifetime of **Supercapacitor** is minimum **500000 charges cycles!!**, and working temperature is -20 +80 Celsius Degree. Secondly

as **Supercapacitor** power backup is a separated circuit can be used independently or combined with existing battery offering an additional unique feature like double power backup (on short power losses runs on Super Capacitor and on longer one automatically switches to battery).

Now, with new add-on board (PIco LP/LF Li-Ion 18650 Battery Holder) you can use all Li-Ion 18650 batteries from electronic cigarettes wide available on the local markets approaching total capacity of 7200mAh, as also 18650 Li-Ion and LiPo4Fe (called also 123).

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** with additional External Supply Powering Input; that has implemented Dynamic Power Tracking (based on Voltage Proportional Charge Control – especially designed for Solar Cells); automatically adjust battery charging current according to power availability from 80mA – 1000 mA, to use all available energy from the Solar Panel in case of use. 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, which is varying due to unstable sunning conditions. The External Supply Powering Input can accept power from 6.5 V DC up to 32 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 protection, ESD protection as also with Zero Voltage Drop Inverse Polarity Protection protecting Raspberry Pi® System from improper usage, but also offers, due to zero voltage drop, usage of most of available energy from the Solar Panel in case of use.

Especially combination of Supercapacitor Power backup and Solar Panel make the system "**set it and forget it**"

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** is powered, and the Battery Pack intelligently charged via the GPIO pins on the Raspberry Pi®, therefore no additional cabling or power supply is required (if used Raspberry Pi® PSU 5V supply). Due to that fact the **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** requires no external cable powering and fits within the footprint of the Raspberry Pi®, it is compatible with most of available cases. If powered via External Power Input (6.5V-32VDC) the there are cases available to hold your designed system.

Also, in case the **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** is powered from the **Extended Power Input**, it allows to charge the battery even if Raspberry Pi® is not powered. Thus, functionality in combination with Events Scheduler make the system always full of energy when needed to be running.

The **embedded double 12V RS232** serial driver combined with embedded multiplexer allows to select and use any of the existing Raspberry Pi® existing Serial Port for communication with external world.

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** can be equipped (optionally) during ordering process with **TCXO** (**1.5 ppm stability**) instead of standard Crystal. That feature offers to the user ultra-stable RTC for time critical applications.

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** can also be equipped with an optional Infra-Red Receiver interface which is routed directly to GPIO18 if used.

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.

The IoT developers will find very useful the 2 independent 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, Raspberry Pi Voltage, Inbound/Outbound Current measure, System Temperature and true 5V 1-wire interface.

Professional developers often need to protect their Applications Intellectual Properties. To support them the **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** offers the XTEA dual path encryption engine that protect the developed software with the unique secure code assigned by Application developer.

The new PCB with 2 oz copper and 6 layers, is designed especially for high current powering systems offering Multilayer Copper Thermal Pipes for increased System Thermal Response and better passive cooling!!

The **UPS PIco HV4.0B/C/D Stack/Advanced/Passive PoE** has been designed and simulated with Altium Designer CAM/CAM tool.

Available UPS PIco HV4.0 HAT Models

The **UPS PIco HV4.0** is offered in 3 different Models covering all User Application Requirements:

- UPS PIco **HV4.0 Stack**
- UPS PIco **HV4.0 Advanced**
- UPS PIco **HV4.0 Passive PoE**

Each model can be Assembled as Model B/BC and D. The assembly defined the handling of the Supercapacitor. Therefore, assemblies are defined as:

- Assembly **B** that supports Both Power Sources (Supercapacitor and Battery), but only one back up power source can be used at once
- Assembly **BC** that supports Both Power Sources (Supercapacitor and Battery), and both can be used at the same time
- Assembly **D** that supports only Battery as power Back-up, and does not support Supercapacitor at all

Therefore, in examples **UPS PIco HV4.0B Passive PoE** can be powered by PoE (ethernet) however only one power backup can be used Super Capacitor (any capacity) or Battery. Cannot be used both at the same time (due to constriction of PCB)

Therefore, in examples **UPS PIco HV4.0BC Passive PoE** can be powered by PoE (ethernet) and both power backup can be used Super Capacitor (any size) or Battery at the same time.

Therefore, in examples **UPS PIco HV4.0D Passive PoE** can be powered by PoE (ethernet) and only battery backup can be used.

UPS PIco HV4.0 HAT Core Features

UPS PIco HV4.0B/C/D is an **U**ltimate **P**ower Management **S**ystem HAT that adds a wealth of innovative powerful and development features to the Raspberry Pi® microcomputer! It has been designed especially for the **Raspberry Pi® 4** and considering all enhanced power and cooling requirement of the **Raspberry Pi® 4** models, however in addition it is still compatible with all former Raspberry Pi models. The core functionality of the **UPS PIco HV4.0B/C/D Advanced** is to protect and automatically shut-down your Raspberry Pi if there is a cable power failure and can be set to automatically monitor and reboot your Pi once cable power has been restored! However, it is only a small part of plenty and powerful functionalities that are implement on this small HAT.
The **UPS PIco HV4.0B/C Advanced** is the only one on the market that offers such enhanced set of features at once:

General

- **HAT Dimensions Compliant**
- **Email broadcasting** on events (Cable Power loss/return, Wake-up, User Button, A/D etc)
- **Plug and Play**
- Ultra-light **System.d Daemon** based on threading
- **GPIO free** (all GPIOs are available for user application) **interaction with Raspberry Pi® via I^2C**
- Enhanced System Monitoring and Programming API
- **Labeled J8 Raspberry Pi® GPIO Pins** for Easy Plug & Play of experimental cables
- Standard **THT 40 Pin** connector (not soldered)
- Remote bootloader for Live Firmware Update on remote locations
- Local bootloader for Live Firmware Update

Powering Options

- Protected (ESD, over current (PPTC fuse), invert polarity) powering input **6.5-32V, 4A Supply @5V**
- **2 independent power back-up** sources: **Battery** and **Super Capacitor** automatically switching between them according to powering condition, used together or separately.
- Enhanced **Line Interactive UPS** functionality (25us response time) if powered via **Raspberry Pi® 5V GPIO**
- **Automatic Advanced Power Spikes Tracking Algorithm**
- **On-Line UPS** functionality if powered via **6.5-32V EPR Powering Input**
- Both cable powering sources (**GPIO 5V** and **EPR 5.5-32V/Passive PoE)** can be connected at the same time
- **Additional File Safe Shutdown** and **Start-up** Functionality on a Single Button
- **Continuously 3.2A@5.25V** supply on battery power backup
- Programmable Battery backed-up of independent power sources of 200@5V and 3V3
- **Intelligent Mobile Power Bank** functionality with safe shutdown/start-up, with Internal or External Slide Switch and RTC

Supported Battery Types, Super Capacitors and Capacities

- Supports a wide range of different Chemistry and capacities batteries and Super Capacitors (LiPO/LiFePO4/Li-Ion/NiMH/SAL/Super Capacitor)
- **Support for Li-Ion 18650 low-cost batteries** (from Electronic Cigarettes) with **dedicated mounting base PCB HAT screwed on top**
- **Support for LiPO 18650 batteries** with **dedicated mounting base PCB HAT screwed on top**

- **Support for LiFePO4 18650 batteries** with **dedicated mounting base PCB HAT screwed on top**
- Support for Super Capacitor 100F soldered to PCB
- Support for Super Capacitor 300F as a separate **PCB HAT screwed on top**
- Support for Super Capacitor 500F as a separate **PCB HAT screwed on top**
- Support for Super Capacitor 800F as a separate **PCB HAT screwed on top**
- Both Back-up Power Sources (Battery and Super Capacitor) can be used at once (with automatic selection and switching) or one of them, thanks to ultra-low voltage High Current Boost Converter

Embedded Peripherals and Interfaces

- **2 User Programmable buttons**
- **2 User Programmable LEDs** (with mapping capability of the system behavior LEDs)
- **System File Safe Shutdown/Start-up button** with additional cable connectivity to external button
- **2 x 200 ks/s** ESD protected **A/D** with voltage follower (high impedance)
- ESD Protected **3V3/5V0 1-wire** interface
- **Infra-Red Receiver** Sensor Interface (IR Not Included) directly connected to the GPIO18
- Integrated **Hardware Real Time Clock (RTC)** with Battery Back-Up
- Optionally **TCXO** with Ultra Stable **RTC** (1.5 ppm @+25°C) – Error Over Time ±0.432sec/day; ±12.960sec/month; ±2.628min/year
- SPDT 2A **RELAY** on the same PCB
- Programmable **Integrated PWM Sounder** (programmable by user API or Automatic), able to play music
- Programmable embedded quad **+/-12V RS232 interface** (2 at once time).
- **Programmable Integrated PWM FAN** (integrated within the basic order) running based on Raspberry Pi Core temperature

Embedded Sensors

- **Double High-side bi-directional hardware current sensing monitor** with power calculation on 5V supply and on EPR (32V) supply
- **onboard** temperature sensor
- Battery Voltage Level
- Super Capacitor Voltage Level
- Raspberry Pi GPIO 5V level Voltage
- EPR 5.5V-32V level Voltage

User/Programmer Interface

- **I²C PIco API Interface** for Control and Monitoring, with over 50 programming registers
- Support for **4 different** users selectable I²C addresses sets:
    - **DEFAULT:** 0x68, 0x69, 0x6A, 0x6B, 0x6C, 0x6D, 0x6E, 0x6F
    - **NO_RTC:** 0x69, 0x6B
    - **ALTERNATE1:** 0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F
    - **ALTERNATE2:** 0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F

RTC Support and System Scheduler

- **RTC** Scheduler
- **Enhanced shutdown and start-up** system based on various internal/external events: Time stamp, A/D level, RS232 data, Cable Powering/Loss (UPS), Battery Level, I/O Level, etc

Case Compatibility

- **If powered via Raspberry Pi GPIO** then is compatible with 99% of all existing cases
- **If Powered via EPR (6.5-32V)** need dedicated **PiBlock Case**

System Monitoring

- **Status Monitoring** – Powering Mode, Inbound current (on 5V and 32V), Outbound current (on 5V and 32V), Powering Voltage, Battery Voltage, Super Capacitor Voltage, Temperature (Raspberry Pi Core and UPS PIco), Timer State
- **Events Pi** Log feature
- **System LEDs** – SYS, BAT, CHG, INF, FAN, SCA, TMR, TMP (optionally selected can be mapped to User LEDs)
- **System Healthy**, that informs user remotely if Raspberry Pi and UPS Pico HV4.0B are running properly and system is power protected (based on various internal system triggers)

User Applications Security

- (optional) **2**-way **XTEA** Based **Encryption Engine** for **User Intellectual Properties** protection

System Protection

- Direct **Raspberry Pi® Hardware Reset Button** via Spring Loaded Pogo Pinpoint to **RUN**
- Direct **Raspberry Pi® Hardware ON/OFF** via Spring Loaded Pogo Pin pointing to **PE**, Even if Powering Cable is entered to **Raspberry Pi®**
- **Programmable Watch-Dog Hardware** feature (**St**ill **A**live Timer)
- **PPTC 3A@5V fuse**
- **ZVD circuit** on all 5V GPIO connections
- **Micro-controller**-based **watchdog**
- Over Temperature protection
- Over Current protection

System Design

- Designed and Simulated with PDA Analyzer with one of the most advanced CAD/CAM Tools – Altium Designer
- Design Based on Microchip 16-bit 16 MIPS micro controller
- Industrial Originated

PCB Construction

- **2 oz copper** PCB manufactured for proper high current supply and cooling
- 6 mils track/6 mils gap technology **6 layers PCB**
- PCB Surface Finishing – Immersion Gold
- **Multi-layer Copper Thermal Pipes** for increased System Thermal Response and better passive cooling

UPS PIco HV4.0 HAT 450 Technical Specifications

UPS PIco HV4.0 HAT Versions			
Features	**UPS PIco HV4.0 HAT Stack**	**UPS PIco HV4.0 HAT Advanced**	**UPS PIco HV4.0 HAT Passive PoE (6.5V-24V)**
Raspberry Pi®			
Raspberry Pi® System Compatibility			
Compatible Raspberry Pi Models	Especially Designed for **Raspberry Pi® 4** Compatible with Pi3B+, Pi3, Pi Zero/W	Especially Designed for **Raspberry Pi® 4** Compatible with Pi3B+, Pi3, Pi Zero/W	Especially Designed for **Raspberry Pi® 4** Compatible with Pi3B+, Pi3, Pi Zero/W
Cases Compatibility			
Cases	Most of the cases Pi Modules PIco cases No Need for additional Power input	Most of the cases Pi Modules PIco cases If used for powering 6.5-32VDC need special case	Most of the cases Pi Modules PIco cases No Need for additional Power input
Raspberry Pi® HAT Compliant			
HAT EEPROM	Not Exist	Not Exist	Not Exist
HAT Dimensions	Compliant	Compliant	Compliant
Raspberry Pi® GPIO Usage (occupation)			
Permanent use of I²C for Raspberry Pi® interaction **User selectable addresses**	GND, SDA0, SCL0 I²C Addresses 1: **68 69 6a 6b 6c 6d 6e 6f** I²C Addresses 2: **58 59 5a 5b 5c 5d 5e 5f** I²C Addresses 3: **48 49 4a 4b 4c 4d 4e 4f** I²C Addresses 4: **69 6b**	GND, SDA0, SCL0 I²C Addresses 1: **68 69 6a 6b 6c 6d 6e 6f** I²C Addresses 2: **58 59 5a 5b 5c 5d 5e 5f** I²C Addresses 3: **48 49 4a 4b 4c 4d 4e 4f** I²C Addresses 4: **69 6b**	GND, SDA0, SCL0 I²C Addresses 1: **68 69 6a 6b 6c 6d 6e 6f** I²C Addresses 2: **58 59 5a 5b 5c 5d 5e 5f** I²C Addresses 3: **48 49 4a 4b 4c 4d 4e 4f** I²C Addresses 4: **69 6b**
Selectable use of Raspberry Pi® RS232 serial0 **For Raspberry Pi®4 selectable for additional 3 ports**	GND, TXD0, RXD0 OFF(HiZ) GND, TXD, RXD of each selected serial port OFF(HiZ)	GND, TXD0, RXD0 OFF(HiZ) GND, TXD, RXD of each selected serial port OFF(HiZ)	GND, TXD0, RXD0 OFF(HiZ) GND, TXD, RXD of each selected serial port OFF(HiZ)
Selectable use of Raspberry Pi® GPIO	I²C only for interaction with PIco GPIO_GEN18 (if IR receiver is used) GPIO_GEN4 (if 1-wire is used)	I²C only for interaction with PIco GPIO_GEN18 (if IR receiver is used) GPIO_GEN4 (if 1-wire is used)	I²C only for interaction with PIco GPIO_GEN18 (if IR receiver is used) GPIO_GEN4 (if 1-wire is used)
Passive Cooling	6 layers PCB 2oz copper with enhanced ground and cooling planes, covered with huge number of thermal vias Number of dedicated cooling holes for air circulation	6 layers PCB 2oz copper with enhanced ground and cooling planes, covered with huge number of thermal vias Number of dedicated cooling holes for air circulation	6 layers PCB 2oz copper with enhanced ground and cooling planes, covered with huge number of thermal vias Number of dedicated cooling holes for air circulation
Automatic Advanced Adaptive Active Cooling	(Optional – but strongly recommended) Embedded FAN with automatic speed regulation Placed exactly above heating Raspberry Pi CPU	Standard Included Embedded FAN with automatic speed regulation Placed exactly above heating Raspberry Pi CPU	Standard Included Embedded FAN with automatic speed regulation Placed exactly above heating Raspberry Pi CPU
Official Raspberry Pi PoE HAT support	Compatible to be working with PoE HAT	Compatible to be working with PoE HAT	None Contains own Passive PoE interface
Dedicated Passive PoE on Pico PCB	none	none	HV4.0 Passive PoE Module 12V/24V DC or less with reverse polarity, 2 level ESD and PPTC fuse protection Continuously current measure over Passive PoE supply on 7V/24V
UPS Specifications			
Power Monitoring			
UPS Type	Line Interactive on Raspberry Pi® 5V GPIO	Line Interactive on Raspberry Pi® 5V GPIO On-line on EPR	Line Interactive on Raspberry Pi® 5V GPIO On-line on EPR On-line on PPoE
UPS Response time	Line Interactive Maximum 25 uS	Line Interactive Maximum 25 uS On-Line 0uS	Line Interactive Maximum 25 uS On-Line 0uS
Automatic Restart on Cable Power Return	YES	YES	YES
Raspberry Pi Battery Backup	Standard – 5.25V@3A current continuous supply to Raspberry Pi via 5V GPIO Pins	Standard – 5.25V@3A current continuous supply to Raspberry Pi via 5V GPIO Pins	Standard – 5.25V@3A current continuous supply to Raspberry Pi via 5V GPIO Pins
Cable Power Input	Raspberry Pi® GPIO 5V	Raspberry Pi® GPIO 5V UPS Pico External Power Input 6.5V-32V DC supplying Pi with 5V@4A Can be used both (GPIO 5V and EPR) at the same time (isolated with ZVD)	Raspberry Pi® GPIO 5V Passive PoE Power Input 24 VDC UPS Pico External Power Input 6.5V-24V DC supplying Pi with 5V@2A Can be used both (GPIO 5V and EPR or PPoE) at the same time (isolated with ZVD)
Cable Power Monitoring Point	Raspberry Pi® GPIO 5V	Raspberry Pi® GPIO 5V/EPR	Raspberry Pi® GPIO 5V/EPR/PPoE
UPS Activation Powering Triggers/Thresholds	Proprietary Algorithm of Falling Power Peak Analysis	Proprietary Algorithm of Falling Power Peak Analysis	Proprietary Algorithm of Falling Power Peak Analysis

	Programmable by user Self-learning system	Programmable by user Self-learning system	Programmable by user Self-learning system
Cable Powering Reactivation	After 5s of continuously cable powering (without power spikes)	After 5s of continuously cable powering (without power spikes) on any cable power source (GPIO or External)	After 5s of continuously cable powering (without power spikes) on any cable power source (GPIO or External or Passive PoE)
Auxiliary 5V and 3V3 Battery Backed Supply on PIco I/O Pins	Standard 5V@200 mA current and 3V3@200 mA continuous supplies on PIco I/O Pin battery backed, with possibility to continuous supply auxiliary devices with Raspberry Pi disconnected	Standard 5V@200 mA current and 3V3@200 mA continuous supplies on PIco I/O Pin battery backed, with possibility to continuous supply auxiliary devices with Raspberry Pi disconnected	Standard 5V@200 mA current and 3V3@200 mA continuous supplies on PIco I/O Pin battery backed, with possibility to continuous supply auxiliary devices with Raspberry Pi disconnected
Power Back-up			
Total Back-up Power	5.25V 3.2A continuously Supply	5.25V 3.2A continuously Supply	5.25V 3.2A continuously Supply
Number Power Back-up Sources	Up to 2	Up to 2	Up to 2
Power Back-up Sources Types	Battery or/and Super Capacitor	Battery or/and Super Capacitor	Battery or/and Super Capacitor
Power Back-up Sources Interaction	Short Power losses Power Back-up on Super Capacitor, Longer Power Losses Power Back-up on Battery Automatic switching/monitoring of Back-up Power Sources Works with Battery only, Super Capacitor only and Both	Short Power losses Power Back-up on Super Capacitor, Longer Power Losses Power Back-up on Battery Automatic switching/monitoring of Back-up Power Sources Works with Battery only, Super Capacitor only and Both	Short Power losses Power Back-up on Super Capacitor, Longer Power Losses Power Back-up on Battery Automatic switching/monitoring of Back-up Power Sources Works with Battery only, Super Capacitor only and Both
Supported Super Capacitors Power Back-up			
On Board	100F/2.8V	100F/2.8V	100F/2.8V
Off Board (additional HAT)	300F/2.8V or 500F/2.8V or 800F/2.8V	300F/2.8V or 500F/2.8V or 800F/2.8V	300F/2.8V or 500F/2.8V or 800F/2.8V
Super Capacitor Charger Type	Adaptive PWM	Adaptive PWM	Adaptive PWM
Super Capacitor Lifetime	1 million charge/discharge cycles	1 million charge/discharge cycles	1 million charge/discharge cycles
Super Capacitor operating Temperatures	-20/+85 Celsius Degrees	-20/+85 Celsius Degrees	-20/+85 Celsius Degrees
Supported Batteries Power Back-up			
Supported Batteries Chemistry	LiPO (standard Supported with system), LiFePO4, Li-Ion, SAL (2.8V), NiMH (2.4V/3.6V), NiCD (2.4V/3.6V)	LiPO (standard Supported with system), LiFePO4, Li-Ion, SAL (2.8V), NiMH (2.4V/3.6V), NiCD (2.4V/3.6V)	LiPO (standard Supported with system), LiFePO4, Li-Ion, SAL (2.8V), NiMH (2.4V/3.6V), NiCD (2.4V/3.6V)
Supported Batteries Types			
Default (standard)	PCM Protected LiPO 3.7V, 450 mAh 15C with silicone high current cables	PCM Protected LiPO 3.7V, 450 mAh 15C with silicone high current cables	PCM Protected LiPO 3.7V, 450 mAh 15C with silicone high current cables
Optional LiPO 3.7V (with additional plastic base)	(1) 1500 mAh 3C 3.7V with silicone high current cables (2) 2550 mAh 2C 3.7V with silicone high current cables (3) 4000 mAh 2C 3.7V with silicone high current cables (4) 8000 mAh 2C 3.7V with silicone high current cables	(1) 1500 mAh 3C 3.7V with silicone high current cables (2) 2550 mAh 2C 3.7V with silicone high current cables (3) 4000 mAh 2C 3.7V with silicone high current cables (4) 8000 mAh 2C 3.7V with silicone high current cables	(1) 1500 mAh 3C 3.7V with silicone high current cables (2) 2550 mAh 2C 3.7V with silicone high current cables (3) 4000 mAh 2C 3.7V with silicone high current cables (4) 8000 mAh 2C 3.7V with silicone high current cables
Optional LiFePO4 3.2V (with additional plastic base)	(1) 3000 mAh 2C 3.2V with silicone high current cables (2) 4000 mAh 2C 3.2V with silicone high current cables (3) 8000 mAh 2C 3.2V with silicone high current cables	(1) 3000 mAh 2C 3.2V with silicone high current cables (2) 4000 mAh 2C 3.2V with silicone high current cables (3) 8000 mAh 2C 3.2V with silicone high current cables	(1) 3000 mAh 2C 3.2V with silicone high current cables (2) 4000 mAh 2C 3.2V with silicone high current cables (3) 8000 mAh 2C 3.2V with silicone high current cables
Optional Li-Ion 3.7V (with additional plastic base)	(1) 5500 mAh 2C 3.7V with silicone high current cables (2) 8000 mAh 2C 3.7V with silicone high current cables (3) 11000 mAh 2C 3.7V with silicone high current cables	(1) 5500 mAh 2C 3.7V with silicone high current cables (2) 8000 mAh 2C 3.7V with silicone high current cables (3) 11000 mAh 2C 3.7V with silicone high current cables	(1) 5500 mAh 2C 3.7V with silicone high current cables (2) 8000 mAh 2C 3.7V with silicone high current cables (3) 11000 mAh 2C 3.7V with silicone high current cables
SAL (Acid, non-maintenance) Batteries 2.4V	(1) 2500 mAh 2C 2.4V with silicone high current cables (2) 5000 mAh 2C 2.4V with silicone high current cables (3) 8000 mAh 2C 2.4V with silicone high current cables	(1) 2500 mAh 2C 2.4V with silicone high current cables (2) 5000 mAh 2C 2.4V with silicone high current cables (3) 8000 mAh 2C 2.4V with silicone high current cables	(1) 2500 mAh 2C 2.4V with silicone high current cables (2) 5000 mAh 2C 2.4V with silicone high current cables (3) 8000 mAh 2C 2.4V with silicone high current cables
Optional bigger Capacities Batteries	Supported in all Chemistries	Supported in all Chemistries	Supported in all Chemistries
Optional 18650 Battery Holder (with protections) HAT			
18650 Li-Ion 3.7V Support	Li-Ion 3.7V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 7200 mAh	Li-Ion 3.7V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 7200 mAh	Li-Ion 3.7V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 7200 mAh

18650 LiPO 3.7V Support	LiPO 3.7V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 7200 mAh	LiPO 3.7V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 7200 mAh	LiPO 3.7V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 7200 mAh
18650 LiFePO4 3.2V Support	LiFePO4 3.2V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 3000 mAh	LiFePO4 3.2V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 3000 mAh	LiFePO4 3.2V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 1000mAh up to 3000 mAh
18650 NiMH 2.4V Support	NiMH 2.4V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 4000mAh up to 8000 mAh	NiMH 2.4V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 4000mAh up to 8000 mAh	NiMH 2.4V batteries are supported via the Single/Double 18650 HAT Batteries Capacity from 4000mAh up to 8000 mAh
Batteries Lifetime			
LiPO	450 cycles	450 cycles	450 cycles
LiFePO4	2000 cycles	2000 cycles	2000 cycles
Li-Ion	300 cycles	300 cycles	300 cycles
NiMH	600 cycles	600 cycles	600 cycles
SAL	600 cycles	600 cycles	600 cycles
Super Capacitor	500000-1 million cycles	500000-1 million cycles	500000-1 million cycles
Batteries Working - Charging Temperature			
LiPO	(0/+50) - (0/+40) Celsius Degrees	(0/+50) - (0/+40) Celsius Degrees	(0/+50) - (0/+40) Celsius Degrees
LiFePO4	(0/+55) - (0/+45) Celsius Degrees	(0/+55) - (0/+45) Celsius Degrees	(0/+55) - (0/+45) Celsius Degrees
Li-Ion	(0/+60) - (0/+50) Celsius Degrees	(0/+60) - (0/+50) Celsius Degrees	(0/+60) - (0/+50) Celsius Degrees
NiMH	(0/+60) - (0/+50) Celsius Degrees	(0/+60) - (0/+50) Celsius Degrees	(0/+60) - (0/+50) Celsius Degrees
SAL	(-20/+60) - (-20/+60) Celsius Degrees	(-20/+60) - (-20/+60) Celsius Degrees	(-20/+60) - (-20/+60) Celsius Degrees
Super Capacitor	(-20/+85) - (-20/+85) Celsius Degrees	(-20/+85) - (-20/+85) Celsius Degrees	(-20/+85) - (-20/+85) Celsius Degrees
Batteries - Charging Type			
LiPO/LiFePO4/Li-Ion/NiMH/ SAL/Super Capacitor	Automatic Selected: Full Charging Cycle Trickle Charging	Automatic Selected: Full Charging Cycle Trickle Charging	Automatic Selected: Full Charging Cycle Trickle Charging
Batteries Protection			
Standard LiPO 450 mAh	PCM and On board cut-off protection system overcharge or over discharge, over voltage and under voltage PCB temperature monitoring (if battery is placed on PCB) Battery Temperature monitoring (optional – on customer request)	PCM and On board cut-off protection system overcharge or over discharge, over voltage and under voltage PCB temperature monitoring (if battery is placed on PCB) Battery Temperature monitoring (optional – on customer request)	PCM and on board cut-off protection system overcharge or over discharge, over voltage and under voltage PCB temperature monitoring (if battery is placed on PCB) Battery Temperature monitoring (optional – on customer request)
High Capacity Li-Ion, LiPO, LiFePO4	PCM and On board cut-off protection system overcharge or over discharge, over voltage and under voltage PCB temperature monitoring (if battery is placed on PCB)	PCM and On board cut-off protection system overcharge or over discharge, over voltage and under voltage PCB temperature monitoring (if battery is placed on PCB)	PCM and On board cut-off protection system overcharge or over discharge, over voltage and under voltage PCB temperature monitoring (if battery is placed on PCB)
Super Capacitor	Over Voltage protection (2.8V) Max charging current 1.2A	Over Voltage protection (2.8V) Max charging current 1.2A	Over Voltage protection (2.8V) Max charging current 1.2A
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 6.5V-32V from EXT	Battery is Electrically Isolated (however cable connected) until system start up for the first time and receive 5V from GPIO (or Passive PoE or 6.5V-24V from EXT)
Optional	Slide ON/OFF switch (external or internal), OFF always with File Save shutdown functionality	Slide ON/OFF switch (external or internal), OFF always with File Save shutdown functionality	Slide ON/OFF switch (external or internal), OFF always with File Save shutdown functionality
Intelligent Battery Power Bank			
Direct Battery Powering (Intelligent Power Bank) with Internal/External ON/OFF Slide Switch	ON/OFF Slide Switch with File Safe Shutdown functionality when switching to OFF (keep battery powering ON until system shutdown)	ON/OFF Slide Switch with File Safe Shutdown functionality when switching to OFF (keep battery powering ON until system shutdown)	ON/OFF Slide Switch with File Safe Shutdown functionality when switching to OFF (keep battery powering ON until system shutdown)
Interfaces			
User Applications Hardware Interfaces			
ESD Protected True 5V0/3V3 1-wire interface	Directly connected to Raspberry Pi ® (if used only) GPIO04	Directly connected to Raspberry Pi ® (if used only) GPIO04	Directly connected to Raspberry Pi ® (if used only) GPIO04
Independent from Raspberry Pi ® 3V3 battery backed supply @200 mA With battery Back-up (Raspberry Pi ® can be OFF when the power Auxiliary 3V3 source is available)	Yes On separated pins	Yes On separated pins	Yes On separated pins
Independent from Raspberry Pi ® 5V0 supply @200 mA With battery Back-up (Raspberry Pi ®	Yes On separated pins	Yes On separated pins	Yes On separated pins

can be OFF when power Auxiliary 5V0 source is available)			
2 x 12 Bit A/D converters ESD protected, pre-scaled to 5V with Sampling rate 200K SPS, DMA buffered, Low Pass Software filtered/ Nonfiltered Both A/D with Voltage Follower buffer	Yes	Yes	Yes
3V3/5V0 RS232 Port that can be programmed as: primary Raspberry Pi® Port Secondary (independent from the existing on Raspberry Pi®)	Yes	Yes	Yes
Double +/- 12 V RS232 converter attached to primary and one of 3 other serial Ports (RPi 4)	Yes	Yes	Yes
Optical Isolated Interface (readable as digital or analog)	optional	Yes	Yes
Primary 3 Pin Relay Rating (resistive) Maximum Switching Current/Voltage on Terminal Block	Yes (Optional) SPDT 2A/32V	Yes (Standard) SPDT 2A/32V	Yes (Standard) SPDT 2A/32V
IR Interface	Directly connected to Raspberry Pi ® (if used only) GPIO18	Directly connected to Raspberry Pi ® (if used only) GPIO18	Directly connected to Raspberry Pi ® (if used only) GPIO18
Opto-Coupler Interface	Optional (only pre-order base)	Yes, Standard Included	Yes, Standard Included
System LEDs Indicators	Timer (Scheduler Activity) - **TMR** (Optional if installed) Super Cap Level - **SCA** System Temperature (RPi Core) - **TMP** System Status - **SYS** Integrated Battery Level - **BAT** Integrated Battery Charger Status - **CHG** (Optional if installed) FAN activity Status - **FAN**	Timer (Scheduler Activity) - **TMR** (Optional if installed) Super Cap Level - **SCA** System Temperature (RPi Core) - **TMP** System Status - **SYS** Integrated Battery Level - **BAT** Integrated Battery Charger Status - **CHG** FAN activity Status - **FAN** External Power Supply Status - **EXT**	Timer (Scheduler Activity) - **TMR** (Optional if installed) Super Cap Level - **SCA** System Temperature (RPi Core) - **TMP** System Status - **SYS** Integrated Battery Level - **BAT** Integrated Battery Charger Status - **CHG** FAN activity Status - **FAN** Passive PoE Supply Status - **EXT**
User LEDs Indicators	Green, Red With capability to connected external LEDs Possibility of Mapping of System Events to User LED(s)	Green, Red With capability to connected external LEDs Possibility of Mapping of System Events to User LED(s)	Green, Red With capability to connected external LEDs Possibility of Mapping of System Events to User LED(s)
System Keys	RPiR, UPSR, FSSD	RPiR, UPSR, FSSD	RPiR, UPSR, FSSD
User programmable Keys	AKEY, BKEY	AKEY, BKEY	AKEY, BKEY
External Connectivity to PIco Keys	FSSD, AKEY, BKEY With capability to connected external KEYs) ON/OFF slide Switch	FSSD, AKEY, BKEY With capability to connected external KEYs) ON/OFF slide Switch	FSSD, AKEY, BKEY With capability to connected external KEYs) ON/OFF slide Switch
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
Independent to Raspberry Pi® Watchdog (Still Alive)	Yes	Yes	Yes
Battery Backed Hardware Real Time Clock and Calendar	Yes When UPS (power cycling is used)	Yes When UPS (power cycling is used)	Yes When UPS (power cycling is used)
Ultra-Stable TCXO Clocking	Optional (1.5 ppm stability on RTC)	Optional (1.5 ppm stability on RTC)	Optional (1.5 ppm stability on RTC)
System Switch ON/OFF	Supported with Embedded or external ON/OFF slide switch or external one (even is system is powered by USB C)	Supported with Embedded or external ON/OFF slide switch or external one (even is system is powered by USB C)	Supported with Embedded or external ON/OFF slide switch or external one (even is system is powered by USB C)
Solar Panel Supply	Yes, on 5V	Though EPR (5.5-32VDC) 5V@4A battery charging current adopted to existing solar conditions	Though EPR (5.5-24VDC) 5V@4A battery charging current adopted to existing solar conditions
Cooling Capabilities			
Advanced Automatic Active Cooling System (FAN)	Yes (optional if FAN installed) based on temperature of the Raspberry Pi® read directly from Raspberry Pi Core	Yes (optional if FAN installed) based on temperature of the Raspberry Pi® read directly from Raspberry Pi Core	Yes (optional if FAN installed) based on temperature of the Raspberry Pi® read directly from Raspberry Pi Core
Passive Cooling	Thought extended hole system supporting air circulation, extended cooling copper planes	Thought extended hole system supporting air circulation, extended cooling copper planes	Thought extended hole system supporting air circulation, extended cooling copper planes
Vital System Information	Thought Registers Set accessed via I^2C interface	Thought Registers Set accessed via I^2C interface	Thought Registers Set accessed via I^2C interface
Selected Vital System Information	Thought Commands accessed via RS232 interface (optional)	Thought Commands accessed via RS232 interface (optional)	Thought Commands accessed via RS232 interface (optional)
XTEA Protection Encryption for User Application	Yes Thought I^2C or Serial Port	Yes Thought I^2C or Serial Port	Yes Thought I^2C or Serial Port
Programmable/Accessible all the system parameters via I^2C PIco	YES, System is full programmable and parameterized	YES, System is full programmable and parameterized	YES, System is full programmable and parameterized

Programmer Interface			
SysInfo Register	Yes, Provide core information about the system: System FSSD Reason - System Wakeup Reason - Pico Restart Reason -	Yes, Provide core information about the system: System FSSD Reason - System Wakeup Reason - Pico Restart Reason -	Yes, Provide core information about the system: System FSSD Reason - System Wakeup Reason - Pico Restart Reason -
Pico Running Register	YES, provide information to remote user if system is running properly	YES, provide information to remote user if system is running properly	YES, provide information to remote user if system is running properly
Remote and local Bootloader for Live Firmware Update	YES	YES	YES
e-mail sending on event	YES	YES	YES
Former UPS Pico HV3.0A/B/B+ Compatibility			
PIco Programmer Registers	100% compatibility + additional due to extra features	100% compatibility + additional due to extra features	100% compatibility + additional due to extra features
Measuring and Monitoring System			
Real Time Raspberry Pi® System current measure	Dual High-side bi-directional Hardware Current Sensing Monitor with power calculation (5V0 path)	Dual High-side bi-directional Hardware Current Sensing Monitor with power calculation (5V0 path) (6.5-32V DC path)	Dual High-side bi-directional Hardware Current Sensing Monitor with power calculation (5V0 path) (6.5-24V DC path) (Passive PoE Path)
Powering Mode Status	YES	YES	YES
Battery Level	YES	YES	YES
Raspberry Pi® 5V level	YES	YES	YES
External Powering Level	NO	YES	YES
Passive PoE Level	NO	NO	YES
Raspberry Pi® Core Temperature	YES	YES	YES
UPS Pico HV4.0 Temperature	YES	YES	YES
A/D(s) Level	YES	YES	YES
Charger Status	YES	YES	YES
Event Driven Scheduler			
Time/Calendar Scheduler	YES	YES	YES
Shut-down/Weak-up on	Time/Calendar Event Low Battery/Super Capacitor Event ON/OFF Slide Switch Event FSSD Button Event Loss of Cable Powering Event External Serial Activity (any) Data Event External Serial Activity (dedicated) Data Event I/O Pin change level Event Opto-Isolated Change Event (optional) A/D Event Raspberry Pi Core Temperature Event Raspberry Pi Shutdown Command Event	Time/Calendar Event Low Battery/Super Capacitor Event ON/OFF Slide Switch Event FSSD Button Event Loss of Cable Powering Event External Serial Activity (any) Data Event External Serial Activity (dedicated) Data Event I/O Pin change level Event Opto-Isolated Change Event A/D Event Raspberry Pi Core Temperature Event Raspberry Pi Shutdown Command Event	Time/Calendar Event Low Battery/Super Capacitor Event ON/OFF Slide Switch Event FSSD Button Event Loss of Cable Powering Event External Serial Activity (any) Data Event External Serial Activity (dedicated) Data Event I/O Pin change level Event Opto-Isolated Change Event A/D Event Raspberry Pi Core Temperature Event Raspberry Pi Shutdown Command Event
Manufacturing			
PCB Manufacturing	6 Layers, 2 OZ Copper, 6mils/6mils Immersion Gold Plated PB Free alloy assembly	6 Layers, 2 OZ Copper, 6mils/6mils Immersion Gold Plated PB Free alloy assembly	6 Layers, 2 OZ Copper, 6mils/6mils Immersion Gold Plated PB Free alloy assembly

UPS PIco HV4.0 HAT Add-On equipment

The **UPS PIco HV4.0B/C/D** is an Ultimate Power Management System HAT itself is fully and independent system. However, it can be upgraded with some add-ons that that adds a wealth of additional innovative powerful and development features to the Raspberry Pi® microcomputer!

There are grouped to simplify decision making for customers, and there are:

Batteries

The minimum current for simple Raspberry Pi 4 draw from battery is about 2 A. User according to his application needs should select the proper battery. All batteries used by the **UPS PIco HV4.0B/C/D** MUST be protected by PCM (PCB) that protect battery from overcharge, over discharge, and shortcut. Any other battery can be used, as far contains protection PCB and a properly made cable and connector. Standard batteries offered by our company are:

- PIco LiPO Battery 450 mAh 15 C (Max Supply Current 7A)
- PIco LiPO Battery 1500 mAh 2 C (Max Supply Current 3A)
- PIco LiPO Battery 2550 mAh 2 C (Max Supply Current 5.1A)
- PIco LiPO Battery 4000 mAh 2 C (Max Supply Current 8A)
- PIco LiPO Battery 8000 mAh 2 C (Max Supply Current 16A)
- PIco LiFePO4 Battery 4000 mAh 2 C (Max Supply Current 8A)
- PIco LiFePO4 Battery 8000 mAh 2 C (Max Supply Current 16A)

Battery Holders

Another solution for Battery Power Backup is Battery Holder. There are 4 types offered by our company. Each battery holder can be supplied with appropriate rechargeable battery. Especially 18650 Li-Ion Batteries from Electronic Cigarettes is a very good choice.

- PIco Single LP/LF Li-Ion 18650 Battery Holder HAT
- PIco Double Li-Ion 18650 Battery Holder HAT
- 3xAA NiMH Battery Holder HAT
- 3xAAA NiMH Battery Holder HAT

Super Capacitors

The Super Capacitor is a very attractive Backup power sources, due to operating temperature, internal resistance as also to huge number or charging/discharging cycles if compared with any battery. Typically, it is between 500 000 – 1 million cycles. The negative point of the Super Capacity is the size limited capacity compared with batteries. However due to extremely extended life cycle are an ideal source for short or even semi extended working times. The **UPS PIco HV4.0B/C** offers a very interesting functionality that use Super Capacitor for a short power losses and battery for longer one, automatically selected. The offered Super Capacitors and Bank of Super Capacitors are the following:

- Super Capacitor 100F (offers workimg time of 30 seconds, if cable power loss)
- Super Capacitor Bank 300F HAT (offers workimg time of 90 seconds, if cable power loss)
- Super Capacitor Bank 500F HAT (offers workimg time of 150 seconds, if cable power loss)

- Super Capacitor Bank 800F HAT (offers workimg time of 240 seconds, if cable power loss)

However must be noted that charging time is long and for the 100F Super Capacitor is about 3-8 minutes. Charging current for this 100F capacitor is about 1.2A generated on **UPS PIco HV4.0BC** PCB. The 300F/500F/800F HAT have own independent buck converter that charges them with current of 3A. Therefore, the 300F Super Capacitor Bank charging time is comparable with the charging time of the 100F placed on the **UPS PIco HV4.0BC** PCB.

Through Hole Add-on Parts

Some Used parts are Trough Hole Parts (THT). If they are used need to be soldered by user or ordered of the soldering very low-cost service offered by our company. They are:

- Infrared Receiver
- ON/OFF micro Slide Switch
- 3 pin 2mm RS232 THT socket and header (without cable)
- 2x10 pins 2mm PIco header
- 80 dB Sound Electromagnetic Transducer

Application Specific Add-on Extensions

Some very specific application requires an extra part. Except of TCXO all other can be used without any manufacturer involvement. They are:

- TCXO - Temperature Compensated Ultra Stable Clock Generator (1.5 ppm) for the embedded RTC (need to be ordered from froommanufacturer and can not be updated by user)
- 3 pin 2mm RS232 socket/header and 100 mm cable (ended with 9 pins D-Sub)
- UPS PIco HV4.0 Terminals Blocks Additional PCB (20 I/O are directed to Terminals Block)
- UPS PIco HV4.0 2x DPDT Realys Terminals Blocks Additional PCB (a PCB with two SPDT Relays Serial Port I/1 and one A/D directed to Terminals Block)

Setting up Procedure

What is in the Package?

This package comes with everything you need to start using the **UPS PIco HV4.0B/C/D HAT** right out of the box. It is assembled, tested, and contains all required accessories. A little work is necessary to setup the complete Raspberry Pi® and **UPS PIco HV4.0B/C/D HAT** in a single full operating system, and this is instructed below. Each Package contains the following parts:

Model / Parts	UPS PIco HV4.0B/C/D HAT Stack	UPS PIco HV4.0B/C/D HAT Advanced	UPS PIco HV4.0B/C/D HAT Passive PoE
Core Pico HV4.0 HAT PCB	1	1	1
2 x 20 Pin THT header	1 x Stack (long pins)	1 x Stack (long pins)	1 x Stack (long pins)
2 x 2 Pin PoE THT header	none	none	1 pc
Dual layer wide temperature adhesive tape (used for battery mounting)	1 pc	1 pc	1 pc
Set of spacers (plastic spacers, rubber stick, or screws and plastic spacer tubes, depending on production lot)	1 set of dedicated to high profile case mounting	1 set of dedicated to high profile case mounting	1 set of dedicated to low profile case mounting
Ultra-high current LiPO battery 450 mAh, with 6A current draw (except if ordered different configuration)	1 pc	1 pc	1 pc
Gold Reset Pins (POGO pin)	2 pcs	2 pcs	2 pcs
2mm Jumper used when Pico HV3.0B Reset Pins are configured for former Raspberry Pi 3	1 pc	1 pc	1 pc
Terminal Blocks (5 way) 2.54 pith	none	1 pc	1 pc
Electromagnetic SPDT Relay	none	1 pc Depending to availability, Relay can be assembled on TOP or BOTTOM of the PCB. It is clearly marked on the package what version user has.	1 pc Depending to availability, Relay can be assembled on TOP or BOTTOM of the PCB. It is clearly marked on the package what version user has.
Ultra-Low Noise FAN 20x20mm	none	1 pc	1 pc

Please kindly notice that, due to shipping regulations, Lithium batteries are packed in the same package but are physically or electrically separated (disconnected) and not connected to the **UPS PIco HV4.0B/C/D HAT** module. It must be connected by the user, and it is a part of the installation procedure.

Some few parts need to be assembled (soldered), it is extremely easy to be done by the end user himself. However, our e-shop (**Homepage - Pi Modules**) is offering in addition the assembly service with a very low just symbolic price, for customer that are not equipped with soldering tools or do not have a proper soldering skill. However, a detailed instructions how to solder these few parts are provided within this User Guide

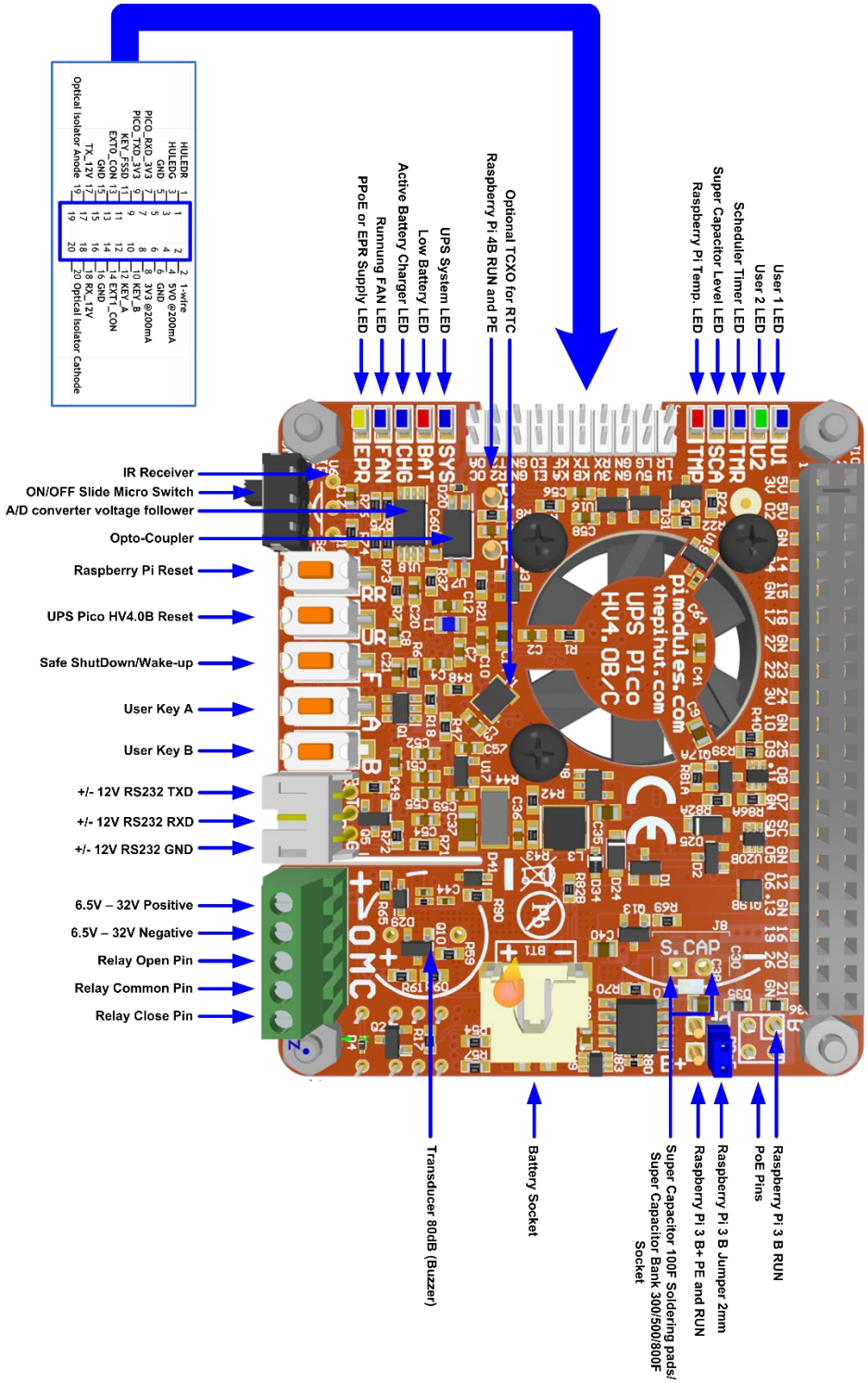
Hardware Setup for the UPS PIco HV4.0B/C/D HAT Stack/Advanced/Passive PoE

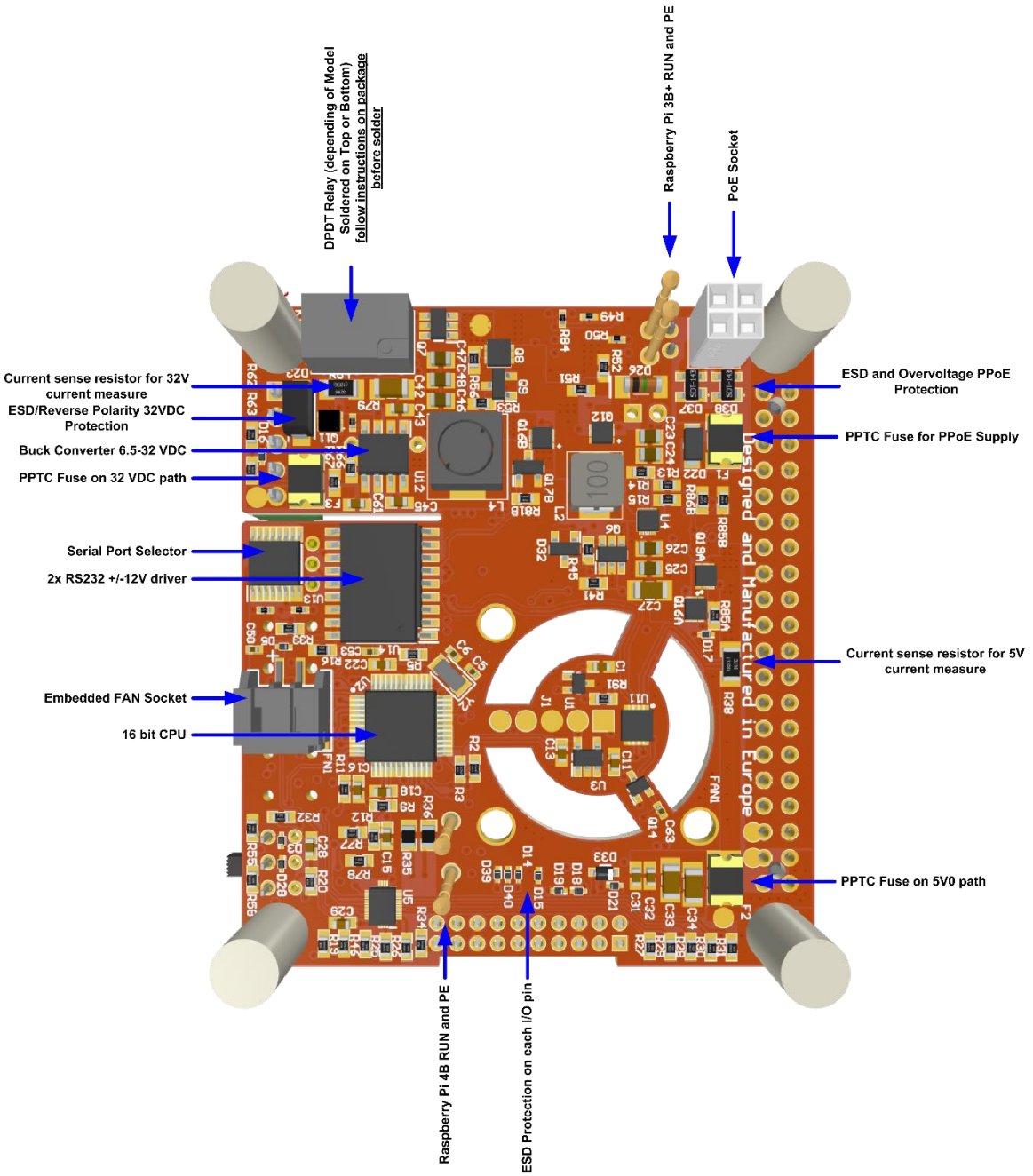
All **UPS PIco HV4.0B/C/D HAT** modules (each version) are based on the same PCB and differ only on SMD and THT parts assembly options. Therefore, users should know that on each board some components are missing or replaced by another one depending to the version.

The differences between Version Stack/Advanced/PPoE are mainly in the following points:

- The SPDT Relay is offered as a standard in the version Advanced/PPoE (need to be ordered optionally for the versions Stack)
- The High Voltage Supply 6.5-32 VDC is present only in the version Advanced and 6.5-28VDC on version PPeE (Passive PoE)
- The opto-coupler is offered as a standard in the version Advanced/PPoE, and not offered in the Stack. However, the ESD protected I/O pin logic level in is existing instead on the version Stack.
- The FAN is offered as a standard in the version Advanced/PPoE (need to be ordered optionally for the versions Stack)

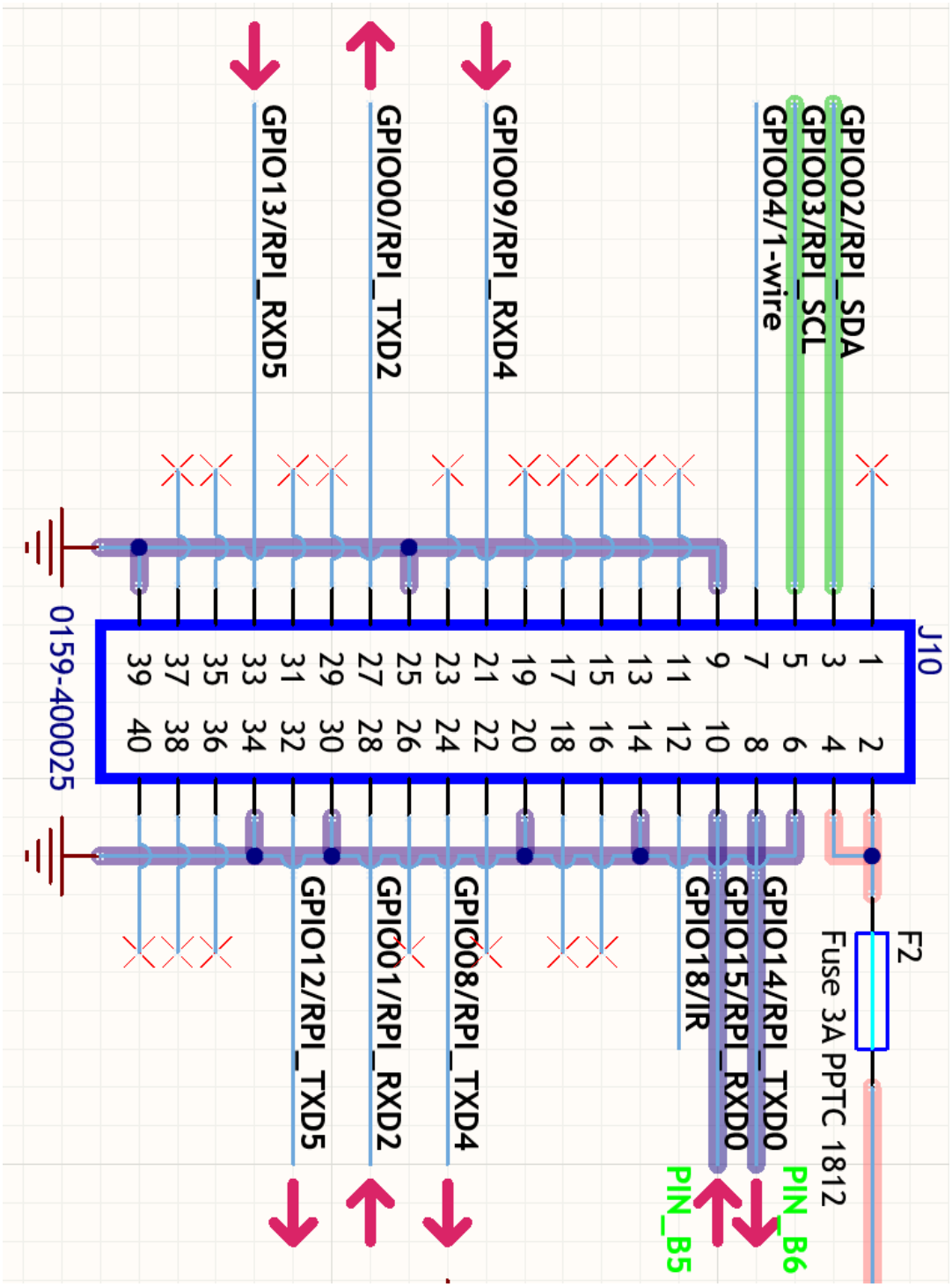
On each of **UPS PIco HV4.0B/C/D HAT** are plenty of I/Os, and other User Interfaces (Keys, Sounder, etc). The below pictures show each version I/Os.


HULEDR
HULEDG
GND
PICO_RXD_3V3
PICO_TXD_3V3
KEY_FSSD
EXTD_CON
GND
TX_12V
Optical Isolator Anode
1-wire
5V0 @200mA
GND
3V3 @200mA
KEY_B
KEY_A
EXTI_CON
GND
RX_12V
Optical Isolator Cathode
PPoE or EPR Supply LED
Active Battery Charger LED
Low Battery LED
Raspberry Pi 4B RUN and PE
Super Capacitor Level LED
Raspberry Pi Temp. LED
User 2 LED
User 1 LED
IR Receiver
ON/OFF Slide Micro Switch
A/D converter voltage follower
Opto-Coupler
Raspberry Pi Reset
UPS Pico HV4.0B Reset
Safe ShutDown/Wake-up
User Key A
User Key B
+/- 12V RS232 TXD
+/- 12V RS232 RXD
+/- 12V RS232 GND
6.5V – 32V Positive
6.5V – 32V Negative
Relay Open Pin
Relay Common Pin
Relay Close Pin
Transducer 80dB (Buzzer)
Battery Socket
Super Capacitor 100F Soldering pads/
Super Capacitor Bank 300/500/800F
Socket
Raspberry Pi 3 B+ PE and RUN
Raspberry Pi 3 B Jumper 2mm
PoE Pins
Raspberry Pi 3 B RUN


DPDT Relay (depending of Model Soldered on Top or Bottom) follow instructions on package before solder
Raspberry Pi 3B+ RUN and PE
PoE Socket
Current sense resistor for 32V current measure
ESD/Reverse Polarity 32VDC Protection
Buck Converter 6.5-32 VDC
PPTC Fuse on 32 VDC path
ESD and Overvoltage PPoE Protection
PPTC Fuse for PPoE Supply
Serial Port Selector
2x RS232 +/-12V driver
Embedded FAN Socket
16 bit CPU
Current sense resistor for 5V current measure
PPTC Fuse on 5V0 path
Raspberry Pi 4B RUN and PE
ESD Protection on each I/O pin
Designed and Manufactured in Europe

Interface	Name on PCB	Functionality
40 Pin SMD Connector with delineation	**J2** (Black one placed on the bottom side)	Used for Pass Through the Stack Connector. Delineation helps users to find a proper GPIO if needed
User LEDs	**U1**, **U2** placed on the left-up corner of PCB	2 color LEDs (Blue, Green) accessed via I^2C used for user applications
Gold plated Reset Pin (POGO pin)	**B, B+, P4**	Used for hardware reset of the Raspberry Pi® or Power Enable, each place is specified by the number, therefore: **B** - RUN on the Raspberry Pi 3 Model B **B+** - RUN on the Raspberry Pi 3 Model B+ **P4** - RUN on the Raspberry Pi 4 Model B User should solder one of the POGO pins on this place according to Model of Raspberry Pi used
Gold plated Power Enable Pin (POGO pin)	**PE, PE**	Each one is used for the Power Enable Raspberry Pi 3 Models B+ and Raspberry Pi 4 Model B
PIco I/O 20 pin (2x10) header	**J9**	Used for various I/O handled by **UPS PIco HV4.0**, detailed described in next chapters
SPDT Relay	**K1**	SPDT Relay soldered on Bottom/Top, used only for various user applications. User MUST check fi it is soldered on Bottom or Top of PCB according to production
Battery Connector	**BT1**	Battery connector, here should be plug in the battery (any type or capacity)
System LEDs	**TMR, SCA, TMP, SYS, BAT, CHG, FAN, EPR**	System LEDs used by **UPS PIco HV4.0** for messaging to the user on various conditions. Detailed described on next chapters
Sounder	None (inside of circle, just marked '**+**' and '**-**' for soldering)	Used for Sound Generation on various **UPS PIco HV4.0** conditions or user

		applications
Infra-Red Receiver	**IR** U6	If soldered, then interface the Raspberry Pi® with IR receiver, used for the any IR application – connected directly to the GPIO18 on Raspberry Pi
Hardware Reset Buttons	Buttons **RR** and **UR**	Hardware Reset Buttons: **RR** – **R**aspberry Pi® Hardware **R**eset **UR** – **U**PS PIco HV4.0 hardware **R**eset
FSSD Button	Button **F**	**F**ile **S**afe **S**hut **D**own **B**utton – detailed description is in next chapters
User Application Buttons	Buttons **A**, **B**	Buttons used for User Applications
Extended Power Supply (6.5-32 VDC)	**+, -**	Extended Power Supply (6.5-32 VDC) for version UPS PIco HV4.0 Advanced Extended Power Supply (6.5-28 VDC) for version UPS PIco HV4.0 PPoE
SPDT Relay	**O**, **M**, **C**	Contacts for the SPDT Relay O – Open M – Common C – Closed Reset Depending to availability Relay can be assembled on TOP or BOTTOM of PCB. It is clearly marked on the package what version user has.
Connector for the FAN	**FN1**	Used to connect FAN when mounted the FAN kit (placed on bottom)
RS232 +12/-12 Interface drivers	**J6 (T, R, G)**	RS232 +12/-12 Interface drivers (selecting only one)
Passive PoE connector	**PPoE**	Used to connect Raspberry Pi PoE for powering
Super Capacitor	**S.CAP**	Used for Connection of 100F Super Capacitor directly soldered or Socket for Super Capacitors Bank 300F/500F/800F. Mind the sing "–"


GPIO02/RPI_SDA
GPIO03/RPI_SCL
GPIO04/1-wire
GPIO09/RPI_RXD4
GPIO00/RPI_TXD2
GPIO13/RPI_RXD5
J10
0159-400025
F2
Fuse 3A PPTC 1812
GPIO14/RPI_TXD0
GPIO15/RPI_RXD0
GPIO18/IR
GPIO08/RPI_TXD4
GPIO01/RPI_RXD2
GPIO12/RPI_TXD5
PIN_B6
PIN_B5

GPIO (Pin #)	Activity	Functionality

UPS Pico HV4.0 Stack, Advanced and Passive PoE handmade components assembly

Most of the Parts used on the **UPS Pico HV4.0B HAT** are SMD technology and are assembled, however few parts that are different for each support version of the Raspberry Pi, are THT and need to be assembled by user. It is very simple procedure and easy to be done by even non experienced to soldering user. However, in order to cover also users that cannot do those simple tasks, our company is offering a very low cost THT assembly service that need to be selected (paid) when doing the ordering.

The following parts need to be hand assembled (depending on **UPS Pico HV4.0B HAT** model is used):

- Gold Plated (POGO) Reset Pin **RUN** (this pin is placed on different places depending to model of Raspberry Pi used
- Gold Plated (POGO) Power Enable Pin **PE** (this pin is placed on different places depending to model of Raspberry Pi used
- Green, 5 ways THT Terminal Block on Version **UPS Pico HV4.0B HAT Advanced** and **PPoE**
- 4 Way, THT PPoE connector on Version **UPS Pico HV4.0B HAT PPoE**
- SPDT THT Relay on Version **UPS Pico HV4.0B HAT Advanced** and **PPoE,** or optional on version **Stack**
- 40 pin (2x20) 2.54 Pass Through Header

The following parts need to be hand assembled (depending on if have been ordered separately as not included by default is used):

- 20 pin I/O 2mm THT Header
- IR Receiver
- ON/OFF Micro Slide Switch (Magic Switch)
- 2mm RS232 Socket
- Buzzer
- Supercapacitor 100F
- Socket for the Supercapacitor plug (300F-500F-800F)
- and only for screwing, the FAN

The Newest Model of the **UPS Pico HV4.0B HAT** has been designed especially for the **Raspberry Pi 4 Model B.** However, it is compatible with most of the former models of the Raspberry Pi (those that have built with 40 pins connector). The main difference between all models of the Raspberry Pi is the place of the **RUN/PE** (Hardware Reset/Power Enable) pin location. It is placed on different places on each model or Raspberry Pi. These different places of it, is mapped on various position on the **UPS Pico HV4.0B HAT**. This **RUN/PE** pins are touching by the Gold-Plated Reset/PE (POGO Pin) pin of the **UPS Pico HV4.0B HAT** and activate some functionalities of the **UPS Pico HV4.0B HAT**. It is strongly recemented to use these pins in order have full functional **UPS Pico HV4.0B HAT**. Therefore,

user need to place and solder this pin on specified by the Raspberry Pi Model place like show below picture.

Here below we are describing, how (depending on model) those parts need to be assembled.

Usage of the Reset Pin (RUN) and Power Enable Pin (PE)

The **Gold Plated** the Reset Pin (RUN) and Power Enable Pin (PE) are used to provide various additional functionalities to the the **UPS Pico HV4.0B HAT**. It is not necessary, however strongly recomened as additional functionalities covered by it make the the **UPS Pico HV4.0B HAT** system more co-operative. It is used with the following functionalities already implemented in the the **UPS Pico HV4.0B HAT**, they are:

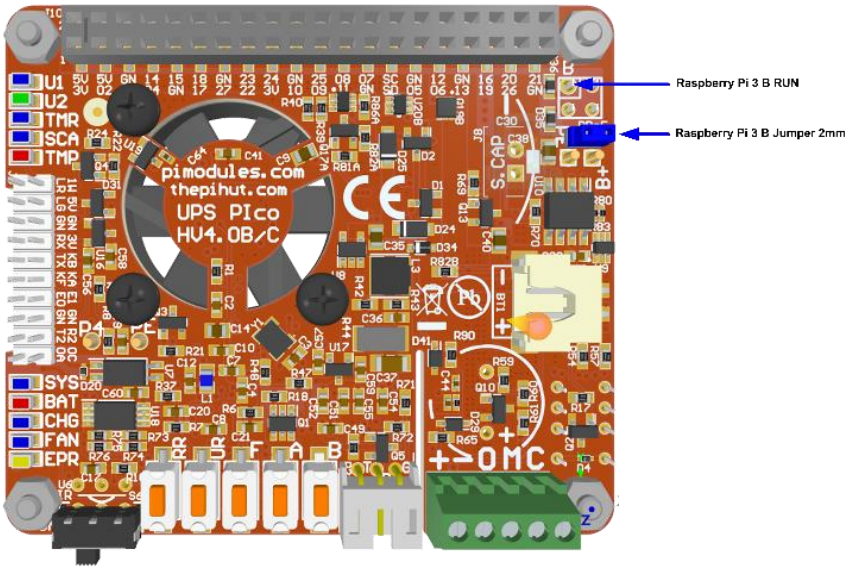
- Button for Hardware Reset of Raspberry Pi®
- Watch Dog ("Still Alive?") functionality - Automatically Resetting (Restarting) of the Raspberry Pi® when hung-up
- Resetting (Restarting) of the Raspberry Pi® when cable power returns during shutting down process.

There is a very simple hand work needed to solder this pin to the Raspberry Pi®

Place on your desk the **Raspberry Pi®** the **UPS PIco HV3.0 HAT** and the **Gold-Plated Reset Pin**.

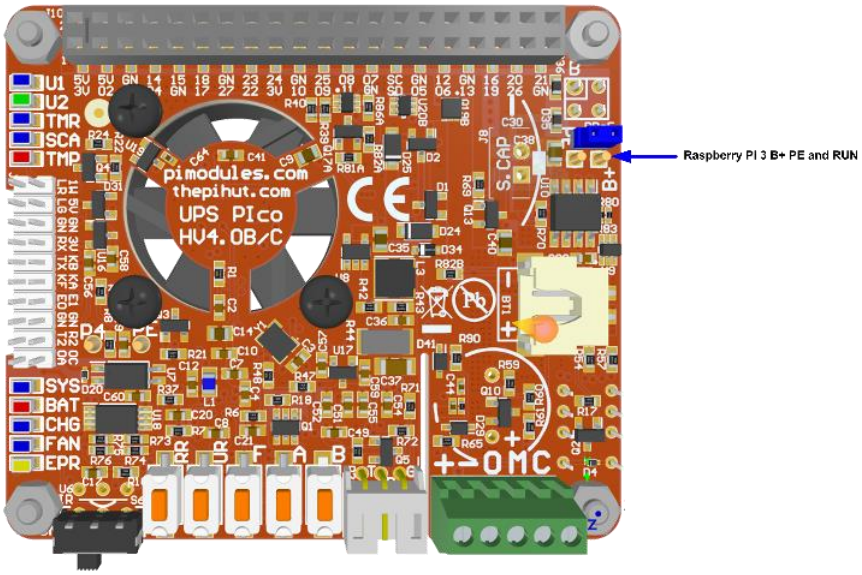
Configuring UPS Pico HV4 HAT to be assembled for the Raspberry Pi 3 Models B Gold Plated Reset Pin – POGO Pin (RUN)

The Raspberry Pi 3 Model B is supported only with RUN (Reset Pin). To solder the RUN pin is used one of the PPoE holes. For this model user need also to solder the 2mm Jumper as show on the below picture.



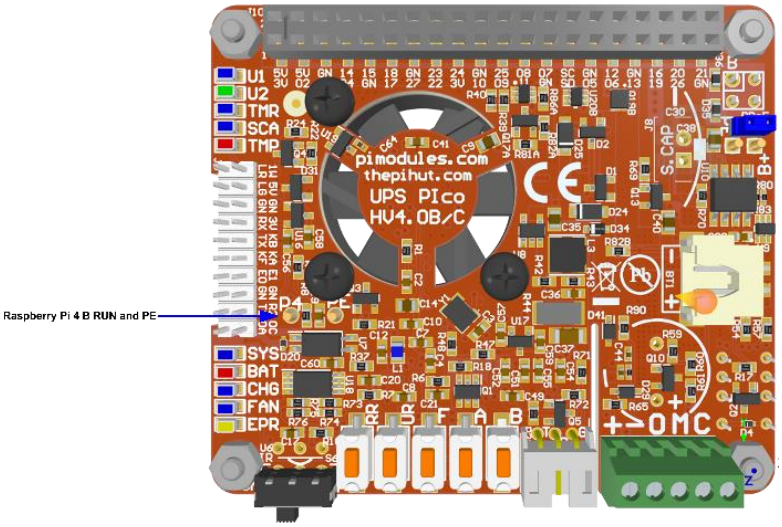
Configuring UPS Pico HV4 HAT to be assembled for the for the Raspberry Pi 3 Model B+ Gold Plated Pins (POGO): Reset Pin (RUN) and Power Enable Pin (PE)

The Raspberry Pi 3 Model B+ has implemented a unique feature Power Enable Pin – the **PEN**. This pin is placed near to the **RUN** pin. This **PEN** feature allows to switch ON/OFF the raspberry Pi 3 Model B+ even if the micro-USB cable is connected. This feature is used in various functions implemented in the **UPS Pico HV4.0B HAT.** To use them, user need to solder the second Gold-Plated Pin on the **PE** position like show below picture. Therefore, the **UPS Pico HV4.0B HAT** need to have soldered 2 x Gold-Plated Pins. **Soldering of the 2mm Jumper is not needed**. Placement of these 2 ins is shown on below picture:



Configuring UPS Pico HV4 HAT to be assembled for the for the Raspberry Pi 4 Model B Gold Plated Pins (POGO): Reset Pin (RUN) and Power Enable Pin (PE)

The Raspberry Pi 4 Model B has implemented a unique feature Power Enable Pin – the **PEN**. This pin is placed near to the **RUN** pin. This **PEN** feature allows to switch ON/OFF the Raspberry Pi 4Model B even if the USB cable is connected. This feature is used in various functions implemented in the **UPS Pico HV4.0B HAT.** To use them, user need to solder the second Gold-Plated Pin on the **PE** position like show below picture. Therefore, the **UPS Pico HV4.0B HAT** need to have soldered 2 x Gold-Plated Pins. **Soldering of the 2mm Jumper is not needed**. Placement of these 2 ins is shown on below picture:


Raspberry Pi 4 B RUN and PE
pimodules.com
thepihut.com
UPS PIco
HV4.0B/C
U1
U2
TMR
SCA
TMD
SYS
BAT
CHG
FAN
EPR
S.CAP
OMC

Configuring UPS Pico HV4 HAT to be assembled with Supercapacitor 100F

TBC

Power Supply Unit Recommendations

Please ensure that you are using a good quality Power Supply Unit available for powering of the Raspberry Pi and **UPS PIco HV4.0 HAT**. A PSU 5.2V@3.0A is recommended. This will ensure that there is enough current to recharge the PIco's battery. Low quality PSUs, or PSUs with bad quality of supply cables cause a voltage drops on the Raspberry Pi® 5V GPIOs that are recognized by the PIco and force a wrong functionality. It is also mandatory to have good quality USB powering cable. Please avoid PSUs that use dual USB connectors as there are double voltage drops on both USB connections (micro-USB, and USB socket).

Once you have all parts correctly installed, we're ready to proceed with software installation

SOFTWARE SETUP FOR UPS PICO HV4.0 Stack/Advanced/Passive PoE

Here below is described step by step Installation Procedure of Daemons, email Broadcasting System, Supercapacitor HAT charger, Setting RTC. The system now is supported by automatic setup procedure therefore user do not need to care about anything just run the proper script and everything will be done automatically.

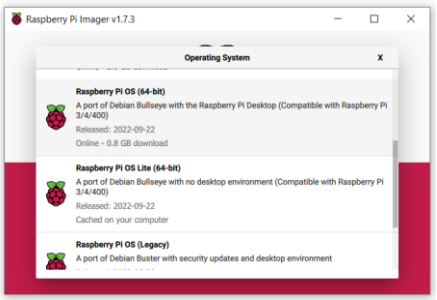
Automatic Installation Procedure of Daemons, Setup of Raspberry Pi OS, Supercapacitor HAT charger, and Setting RTC of UPS Pico HV4

Before user start installing the assosiated of Daemons, email Broadcasting System, Supercapacitor HAT charger, Setting RTC; it is needed to prepare the SD card with preferred OS. To do that we that we recommend using a dedicated Raspberry Pi ® Imaging tool. This tool is preparing the SD card ultra-fast and set up properly any required settings.

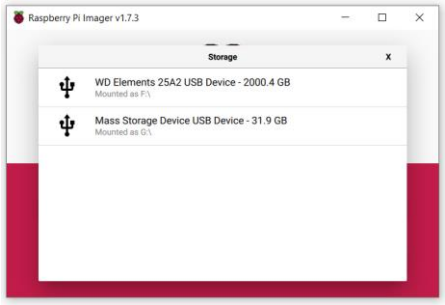
It can be downloaded from here:

https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/

After running it, user need to select the preferred OS from a plenty available there:

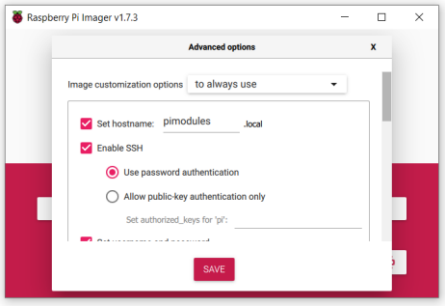


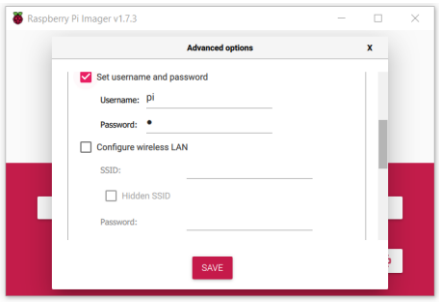
The user needs to select the storage where the new setup OS will be stored. It needs to be done very carefully to avoid possible mixing with existing USB based external HDD/SSDs (as all data will be erased).



After selection of the storage media, user need to set some parameters that are needed (suggested) for UPS Pico HV4 system usage. There are:

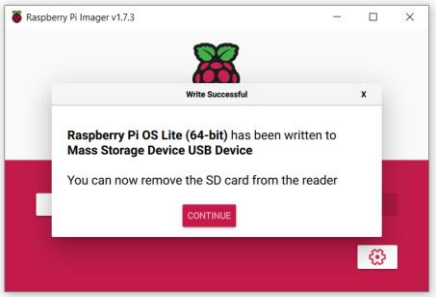
- Host name (keep default or set user preferred)
- Enable SSH (it is very usefully for future access to the system with having keyboard and monitor connected (locally or remotely) – therefore strongly recommended
- Username
- And Password
- Configure if needed (and activate WiFi)





Then start writing the OS image to selected SD card.

The whole procedure is very fast, and after few minutes fresh SD card is ready for use.



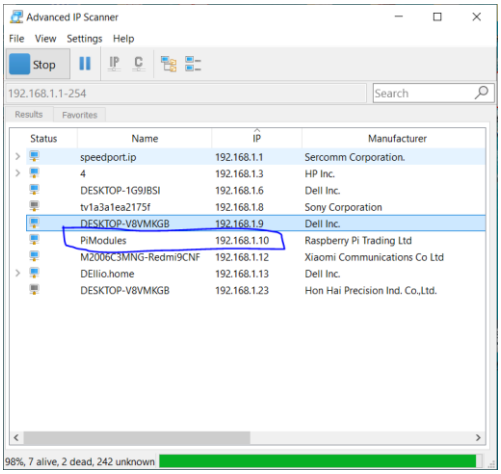
To access the system, we recommended to use SSH, locally or remotely, as this will save from the needs to use of Monitor and keyboard.

In order to use SSH user need to indicate the Raspberry Pi **IP address**. This can be achieved with a very useful too (free for amateurs) called Advanced IP Scanner.

It can be downloaded from the following link:

https://www.advanced-ip-scanner.com

After Installation run it and the IP Scan show the Raspberry Pi address, like in below example.

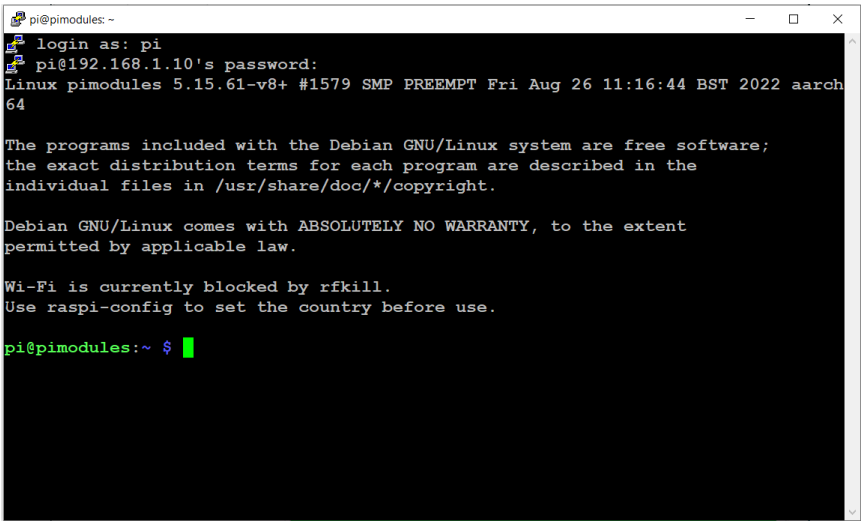


Based on the Raspberry Pi IP address user can log in with any SSH client, in our example we are using the PuTTY, that can be downloaded from here:

https://www.putty.org

After Installation run the PuTTY, providing the proper IP address, Username as also Password.

Logging to the system will show the following screen

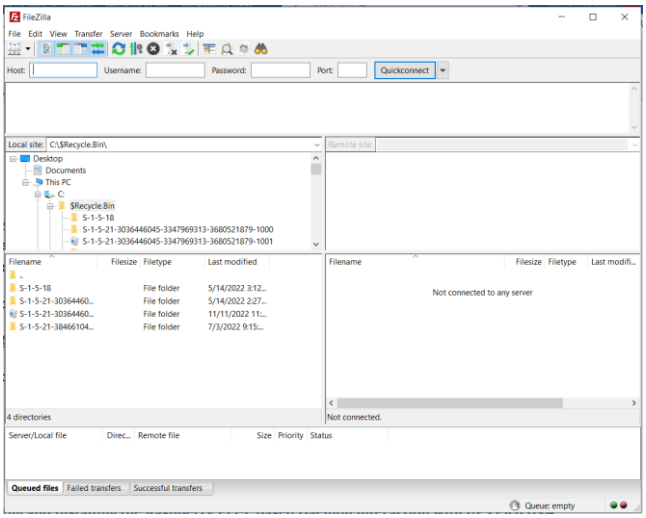


In order to do the automatic installation of UPS Pico HV4 Daemons and etc., user need to copy one file to the OS SD card. This can be done easy using FTP. As in the future user may need to have a wider view of the contents of the SD card, we suggest using a very good free tool FileZilla.

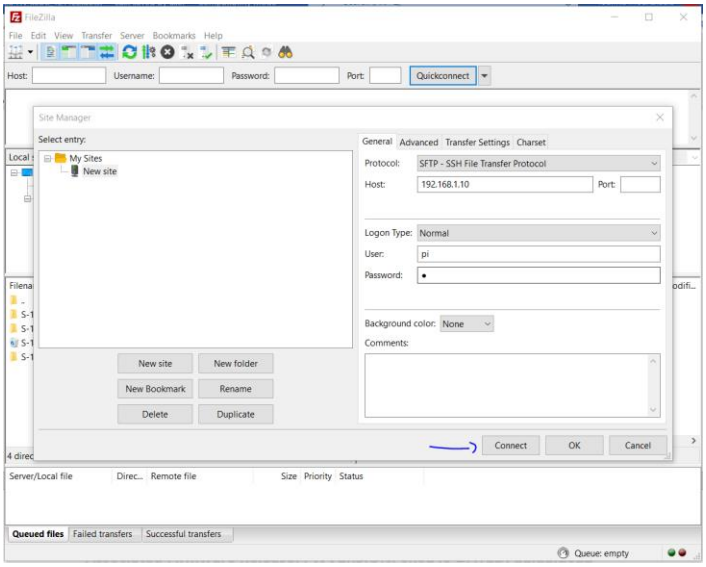
It can be downloaded from here:

https://filezilla-project.org

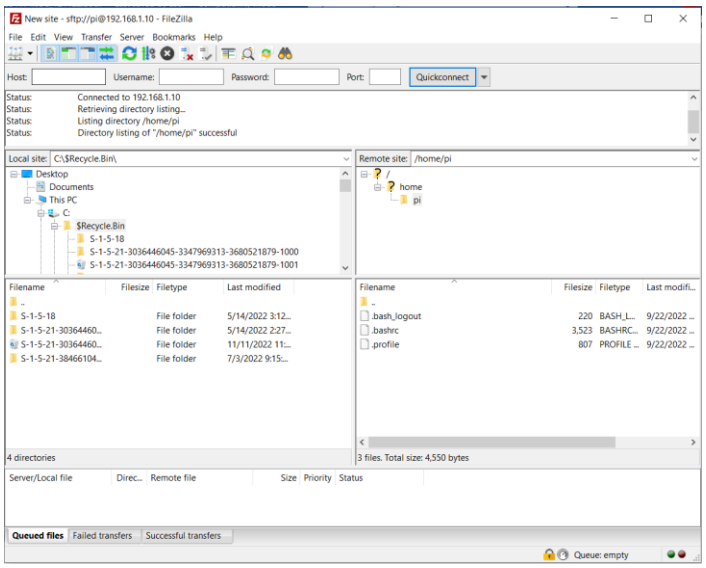
After Installation run the PuTTY, providing the proper IP address, Username as also Password.



Then press connect to have access to the Raspberry Pi SD card.



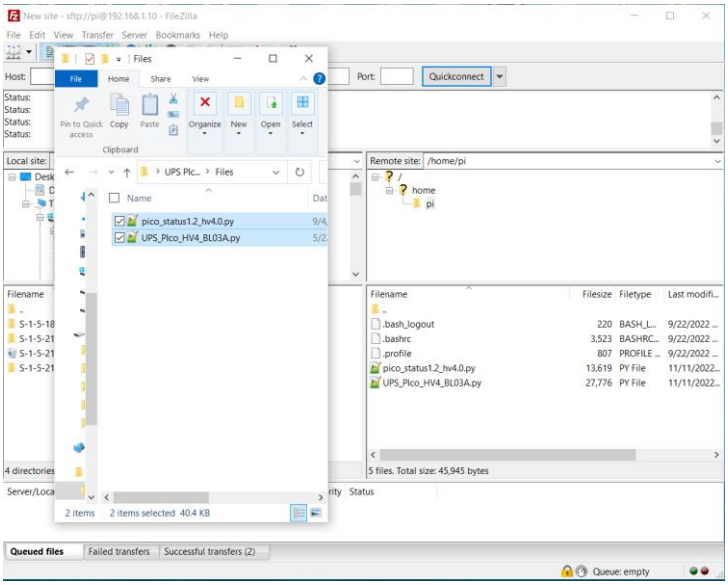
User will see the content of it, like below picture.

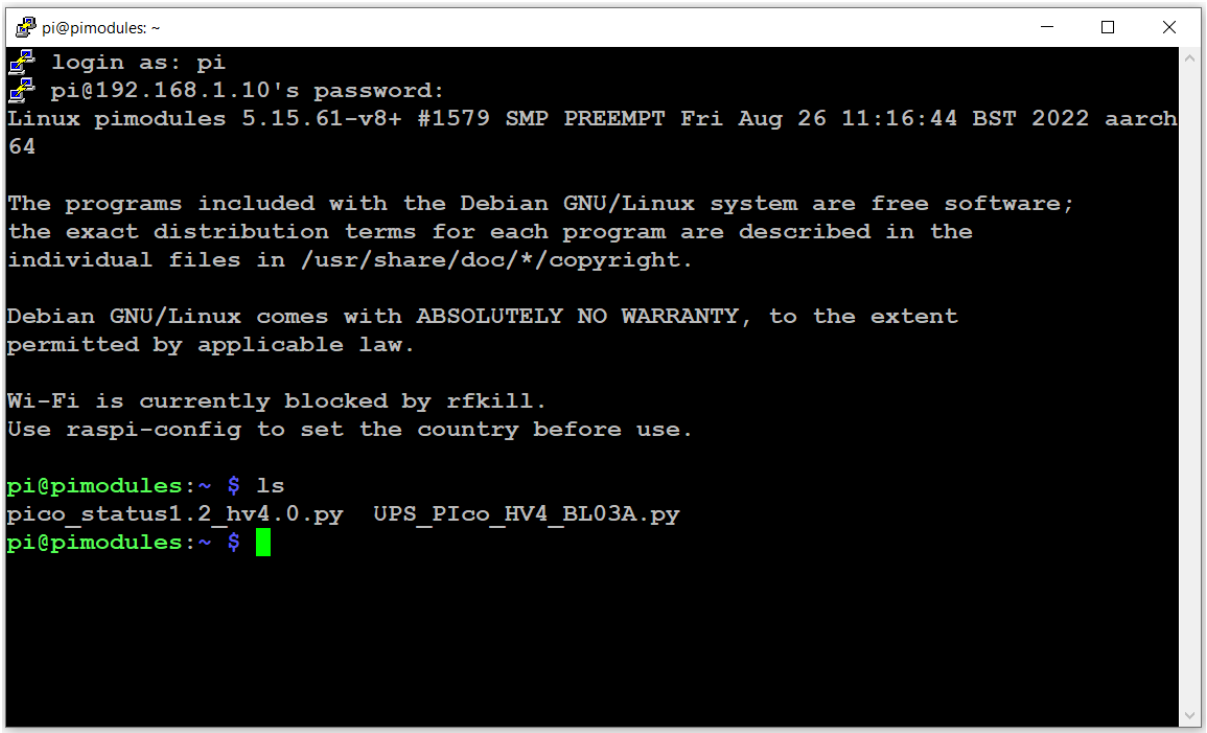


Two files need to be copied to the SD card, in order to have system ready to use, these are:

- UPS_PIco_HV4_BL03A.py
- pico_status1.2_hv4.0.py

User can just drag them to the Raspberry Pi SD card with a mouse like show on below picture.



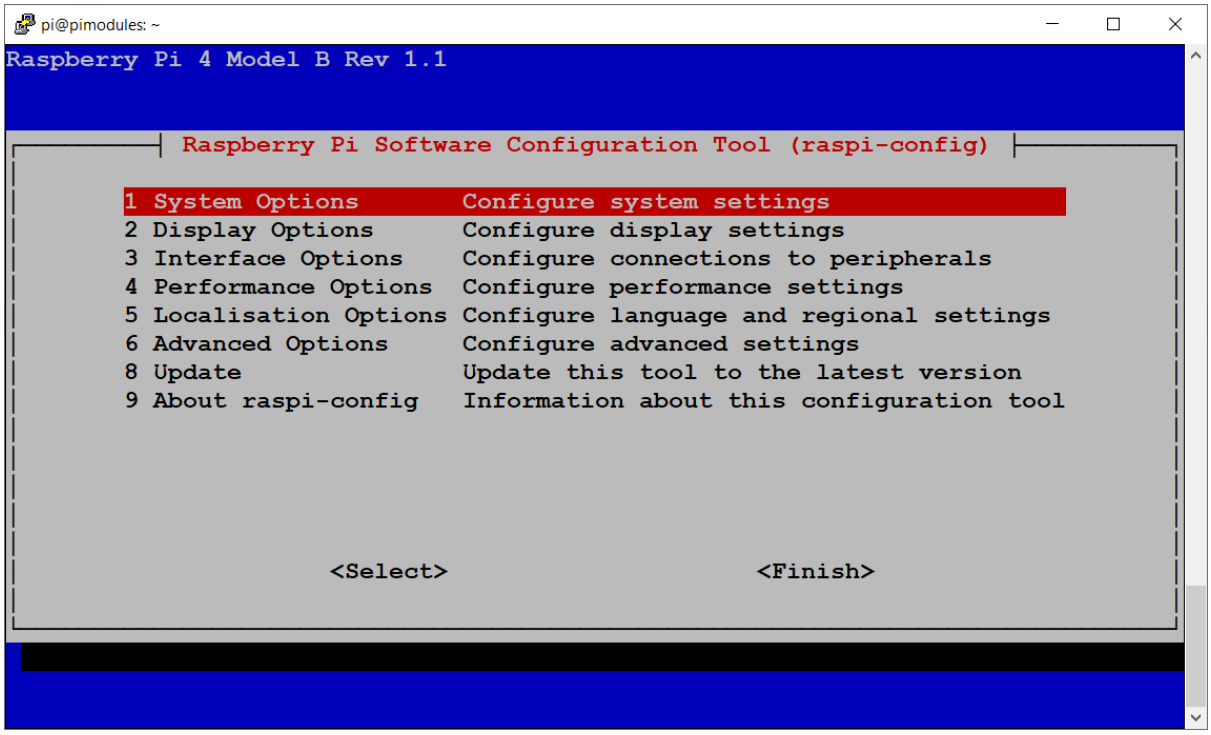


Final Setup/Check with '*raspi-config*' of OS settings for the UPS Pico HV4

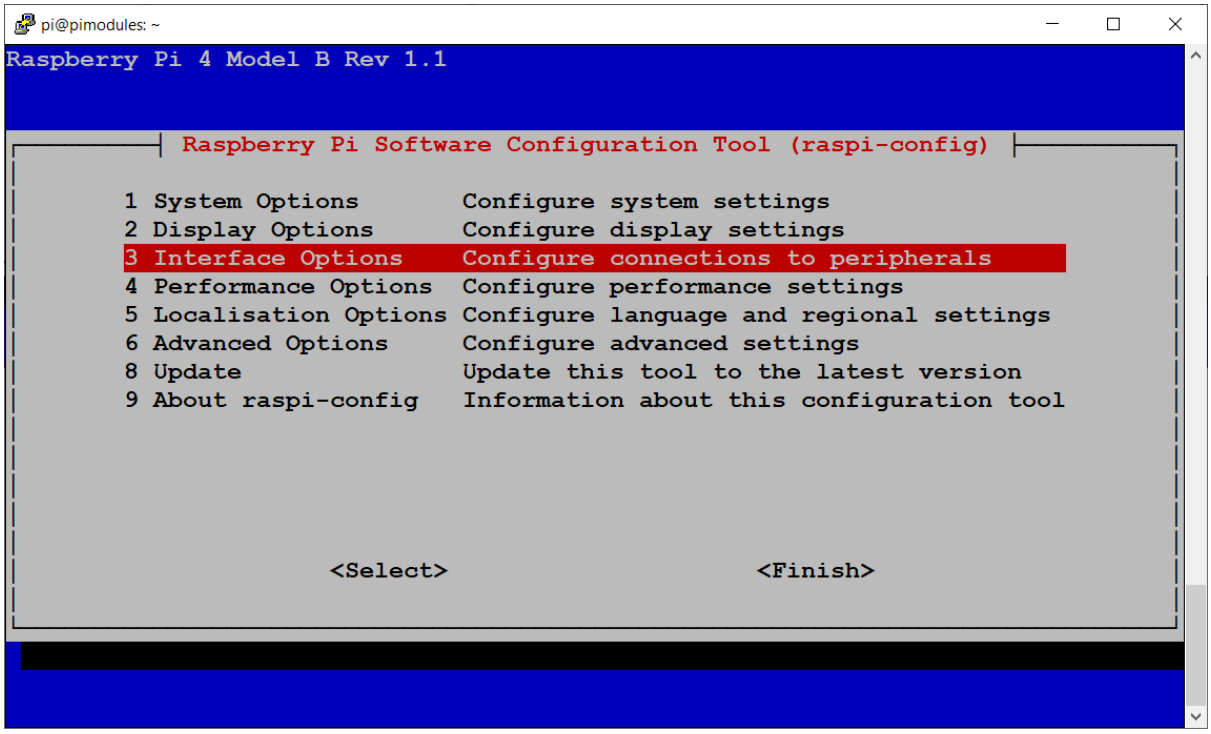
To run the UPS Pico HV4 system on the Raspberry Pi, user need to check or change the following parameters on the Raspberry OS by using the **raspi-config** utility.

1. Run the raspi-config utility

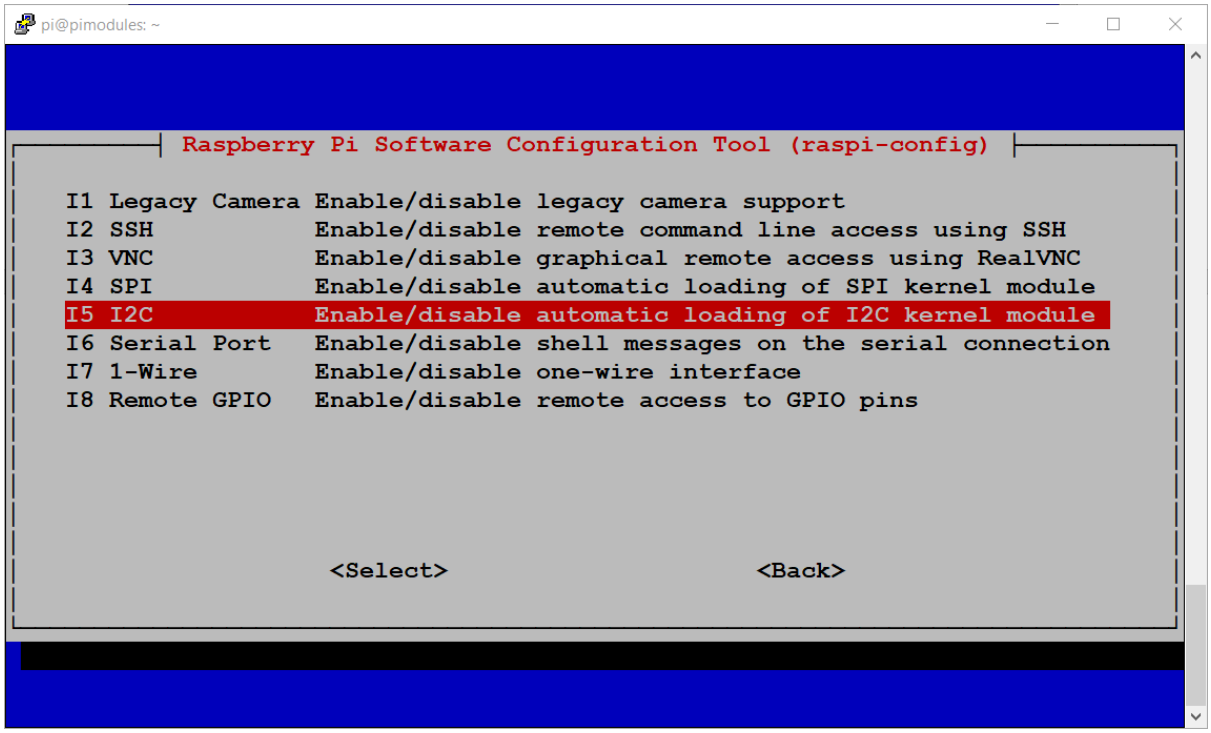
sudo raspi-config



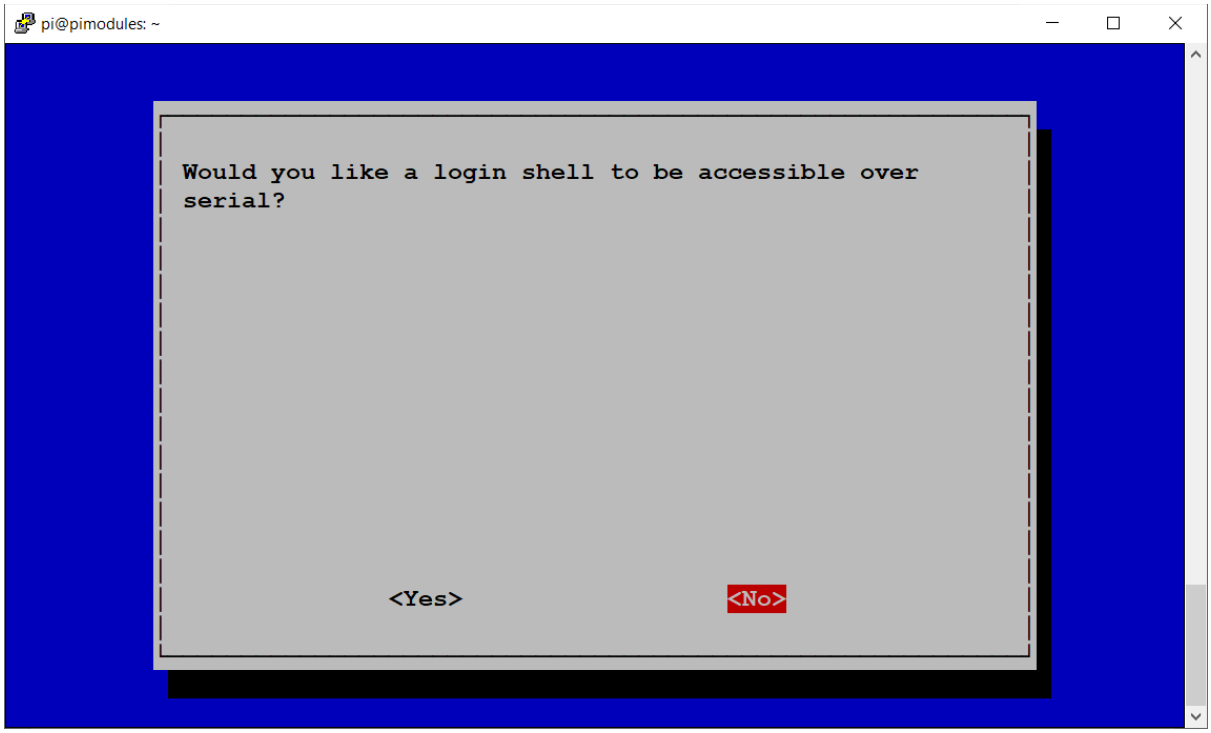
1. Select Interface Options

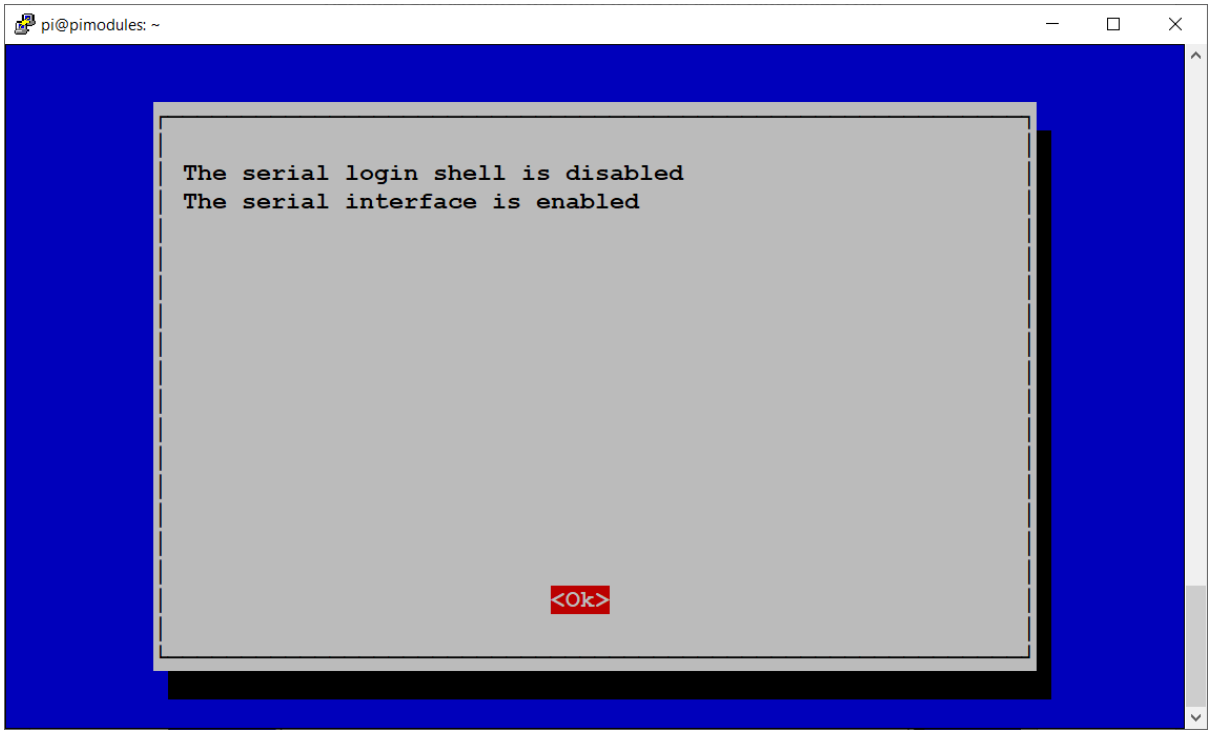


2. Activate the I²C interface



3. Activate the Serial Port interface for User Applications





Then Reboot the Raspberry Pi System

Using and Installing the Raspberry Pi I²C based Daemon interaction with UPS Pico HV4

Using and Installing the Raspberry Pi I²C based Daemon interaction with UPS Pico HV4

The **UPS PIco HV4.0 HAT** need to "know" what the status of Raspberry Pi is. If it is running, shutting down, restarting or hang-up. For this interaction is used I²C interface in order to leave all other pins free for user applications. The interaction is achieved by using of dedicated scrip that is started when Raspberry Pi is starting and running all the time. It is called **pico_pi2c.py**, and it is used also for email broadcasting, transferring the Raspberry Pi Temperature, Safety Shutting down the Raspberry pi, as also for Supercapacitors Bank charging. Two types of this script are available, slightly differed in written in Python2 and Python3. They have extensions *.2py and *.3py. Users need to rename one of them to *.py and follow below installation instructions.

To be sure that a proper pico_i2c.py script has been selected, user can run it on command line and see if there are any errors reported as also if the SYS LED is blinking properly. Just run the following command and check. SYS LED should be blinking and UPS running. Then user can follow below instruction to install the Daemon and made the system running automatically. Always remember that use a proper **pico_pi2c.py** depending to python version installed.

sudo pico_i2c.py

Installing/Enable the Daemon

1. Copy provided python pico_i2c.py script to the root directory, if not already done so.
2. Create a configuration file that tells **System.D** what we want it to do and when:

sudo nano /lib/systemd/system/pico_i2c.service

3. Add the text below to this file, and then exit by saving it:

[Unit]

Description=UPS PIco GPIO Free Raspberry Pi Interaction Service

After=multi-user.target

[Service]

Type=idle

ExecStart=/usr/bin/python /home/pi/pico_i2c.py

StandardOutput=inherit

StandardError=inherit

Restart=always

[Install]

WantedBy=multi-user.target

4. Setup the file permissions

sudo chmod 644 /lib/systemd/system/pico_i2c.service

5. Reload, enable, start the daemon

sudo systemctl daemon-reload

sudo systemctl enable pico_i2c.service

sudo systemctl start pico_i2c.service

6. Reboot the Pi and your custom service should run:

sudo reboot

Decreasing the Raspberry Pi I²C rate

The Raspberry Pi I²C driver has a bug, well known – as a stretching problem. This because that Raspberries Pi are not waiting for the peripheral if it is not fast enough. It was not so important for the former interaction way, as the access to the I²C, was only if user requests data from the UPS Pico. The UPS Pico is requiring the stretching in most of the cases, as it is slower that Raspberry Pi. The new driver is using the I²C as an interaction way with UPS Pico, therefore the possibility of missed access is important. To overcome this problem user, need to slow down the rate of the Raspberry Pi to lower one from the 100K as it is by default. We propose to slowdown to 50K or even less i.e. to 25. This will not affect any other peripheral using the I²C interface. This can be done by editing the **config.txt** and adding the following lines

sudo nano /boot/config.txt

[PICO]
 # Added for PIco
enable_uart=1
dtoverlay=i2c-rtc,ds1307
dtparam=i2c_arm=on
dtparam=i2c1_baudrate=5000

Installing/Enable email broadcasting system

TBC

Installation Procedure of the UPS PIco HV4.0 Hardware RTC

1. Ensure to run below line

   sudo apt-get -y install i2c-tools

2. Edit by running the following line

   sudo nano /etc/modules

   and check, make sure to have the following items in the file and add what is missing:

   i2c-bcm2708

   i2c-dev

   rtc-ds1307

3. Edit by running the following line

   sudo nano /boot/config.txt

4. and add the following to this file:

   enable_uart=1

   dtoverlay=i2c-rtc,ds1307

5. Edit by running the following line

   sudo nano /etc/rc.local

6. and add the following line before "**exit 0**"

   sleep 4; hwclock -s &

7. Reboot system by

   sudo reboot

8. Remove the **fake-hwclock** which interferes with the RTC **hwclock**

   sudo apt-get -y remove fake-hwclock

   sudo update-rc.d -f fake-hwclock remove

9. Run

   sudo nano /lib/udev/hwclock-set

10. and comment out these three lines**:**

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

11. Run **date** to verify the time is correct.

12. Plug in Ethernet or WiFi (if not plugged before) to let the Pi sync the right time from the Internet. Once that's done, run:

    sudo hwclock -w

13. to write the time, and another

    sudo hwclock -r

13. to read the time

That's it! Next time you boot the time will automatically be synced from the RTC module.

Ready to use 16 GB SD card Images

In order to support our customers, we are usually releasing a ready image of the latest OS with everything installed. Therefore, user need just to download such image and program their own SD card and use it. These images are available at:

TBC

Bootloader Feature – Keep the firmware up to date

The **UPS PIco HV4.0 HAT** is a very flexible hardware platform that offers an ultra-wide range of features. Most of them are software programable however based on existing hardware on the PCB. Therefore, during the time, new versions with additional features are released supporting more and more features. It is mandatory for the user, to have the ability to upload the newest firmware version whenever it is released, to keep the **UPS PIco HV4.0 HAT** up to date. The firmware upload to the **UPS PIco HV4.0 HAT** is done by running a small piece of software located in dedicated and protected memory part in the micro-controller called boot sector and python code running on the Raspberry Pi. This micro-controller memory part is protected from any erase, so even if uploading of the new firmware procedure fails, this bootloader will never fail.

The execution (the invoking) of the bootloader can be done from a software level by running of some dedicated commands, or manually by pressing of dedicated key sequence. When system is in bootloader mode, is not acting as protection for power losses, so please use it with extremely care. However, there is activated protection mechanism for powering loses but are very simple and run only on USB powering mode. Please make sure that battery is fully charged if used with battery, if used with Supercapacitor then make sure that system is USB powered. The bootloader functionality

ensures that the **UPS PIco HV4.0 HAT** 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.

There are 3 ways to enter the bootloader mode:

a. By Software executing command 0xff called Local Bootloader Invocation
b. By Software executing command 0xbb called Remote Bootloader Invocation
c. By Hand pressing dedicated keys pressing called Hand Bootloader Invocation

The Local Bootloader Invocation

The main difference between **Local** and **Remote Bootloader** is that the Local one, once invoked cannot be exited until UR (reset) button pressed. Therefore, system will continue waiting for the firmware uploading until it will start, or UR pressed. This condition on Remote system could cause a problem as exit of this condition is only possible if new firmware will be properly uploaded or UR button pressed, event that can be done only locally by human, so there is the name Local Bootloader. The bootloader is invoked by running the following command line:

sudo python UPS_PIco_HV4_BL.py -l -f ups_pico4_main_XXXX.hex

The TMP (RED) and U2 (GREEN) LEDs will lit confirming that your system entered the bootloader mode, and system starts uploading the new firmware. The name **ups_pico4_main_XXXX** of the new firmware must be replaced with current version of firmware.

The uploading process is supported with following visualization to give proper feedback to the user:

2022-05-10 19:48:04,703 [WARNING] PICO PCB version: 0x41 PICO model: BC PPoE

2022-05-10 19:48:04,704 [WARNING] RPi model: Raspberry Pi 4 Model B Rev 1.1

2022-05-10 19:48:04,705 [WARNING] MD5 of the firmware file: cdf323b7197ec496d2aff0f899db670e

2022-05-10 19:48:04,846 [WARNING] Firmware file verification OK

2022-05-10 19:48:04,851 [WARNING] Current firmware release: 0x129

2022-05-10 19:48:06,958 [WARNING] Serial link with PICO UPS verified

Uploading firmware |##################################| 100.0% - 0s

2022-05-10 19:50:50,722 [WARNING] New firmware release: 0x12A

2022-05-10 19:50:50,723 [WARNING] Firmware update completed

The Remote Bootloader Invocation

The Remote Bootloader once invoked will remain on this stage for 8 seconds and if no firmware start uploading cause automatic system restart and "normal" running. It can be also exited from the bootloader mode by pressing the UR key. These conditions guarantee to the user that if any reason

system does not start firmware uploading, return to “normal” functionality. The bootloader is invoked by running the following command line:

sudo python UPS_PIco_HV4_BL.py -f ups_pico4_main_XXXX.hex

The TMP (RED) and U2 (GREEN) LEDs will lit confirming that your system entered the bootloader mode, and system starts uploading the new firmware. The name **ups_pico4_main_XXXX** of the new firmware must be replaced with current version of firmware.

The uploading process is supported with following visualization to give proper feedback to the user:

2022-05-10 19:48:04,703 [WARNING] PICO PCB version: 0x41 PICO model: BC PPoE

2022-05-10 19:48:04,704 [WARNING] RPi model: Raspberry Pi 4 Model B Rev 1.1

2022-05-10 19:48:04,705 [WARNING] MD5 of the firmware file: cdf323b7197ec496d2aff0f899db670e

2022-05-10 19:48:04,846 [WARNING] Firmware file verification OK

2022-05-10 19:48:04,851 [WARNING] Current firmware release: 0x129

2022-05-10 19:48:06,958 [WARNING] Serial link with PICO UPS verified

Uploading firmware |##################################| 100.0% - 0s

2022-05-10 19:50:50,722 [WARNING] New firmware release: 0x12A

2022-05-10 19:50:50,723 [WARNING] Firmware update completed

The Hand Bootloader Invocation

The Hand Bootloader invoking must be done by hands therefore need system to be accessible by human. Once system entered in the bootloader mode, will remain on this mode for about 8 seconds and return to normal conditions if no firmware uploading process will be stared.

The following procedure must be followed to enter the bootloader mode:

- 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 TMP (RED) and U2 (GREEN) LEDs will lit confirming that your system is in bootloader mode.

sudo python UPS_PIco_HV4_BL.py -j -i -g -f ups_pico4_main_XXXX.hex

2022-05-09 21:01:39,499 [WARNING] Serial link with PICO UPS verified

```
Uploading firmware |################################| 100.0% - 0s

2022-05-09 21:04:44,897 [WARNING] Firmware update completed
```

REMARK:

The Hand Bootloader Invocation is used also (and usually) when firmware uploading faults and system is not responding to the commands, as this is the only piece of firmware that cannot be destroyed and protected from any re-writing.

Names of **Local** and **Remote** Bootloader procedures are done based on our experience and user can use both on Local and Remote boot loading according to their needs.

There is a list of commands that can be used by user in various specific cases. They are listed in the Python Code, and are:

- -f --fw-file
- -p --serial-port
- -b --baudrate
- -g --skip-fw-verify
- -G --fw-verify-only
- -H --skip-fw-md5
- -I --skip-i2c-fw
- -j --skip-i2c-bl
- -k --skip-i2c-reset
- -l --i2c-bl-local

Post-Firmware Update procedure

After firmware uploading system will be set to factory defaults, cancelling any user setup. The factory setup is stored in the new firmware, any new firmware can have different default values, therefore users can ask for this to the manufacturer if bulk orders are made. These parameters can be upgraded time to time within the new firmware procedure, however some of them cannot as are stored in the boot sector of the micro-controller.

NOTE: After new firmware uploading, system automatically recognize it and set the default values, therefore is some specific values has been used, user need to set them again.

0x6B -> UPS PIco Commands Default Values (Factory Reset) - Current Firmware Version

UPS PIco HV4 Parameter	Register Address	Default Value
Raspberry Pi Serial Port Communication	0x02	OFF (0x00)
UPS PIco HV4 Serial Port Communication	0x02	OFF (0x00)
System on Hold	0x03	OFF (0x00)
RS232 12V Driver selection	0x04	OFF (0x00)
Default Battery	0x07	Ready from Boot Sector or set by the new firmware usually “L”
Buzzer Mode	0x0c	0x01
FAN Mode	0x11	0x03
FAN Speed	0x12	50
FAN Threshold Temperature	0x13	50 Celsius
System Running time when Battery Powered	0x01	0x01 (60 seconds)
User 5V0 and 3V3 additional Powering when Battery Powered	0x06	OFF
User LED U1 (Green)	0x09	OFF
User LED U1 (Blue)	0x0a	OFF
System LEDs ON/OFF	0x15	ON
Magic Switch Functionality	0x16	OFF (0x00)
Raspberry Pi on Hold when starting up	0x03	0x00
FSSD Duration	0x1a	30 seconds
Embedded Battery Charger	0x19	ON
Raspberry Pi OFF time before Enter to the Low Powering Mode	0x1b	5 minutes
I^2C Address Selection	0x00	I2C_Default (0x60) cannot be read directly only by command execution: *sudo i2cdetect -y 1*
STA Timer	0x05	OFF (0xff)
RTC Setup	none	Can be read only by reading register at 0x6a The default setup is: Sec=00 Min=00 Hour=00

		Wday=07 Mday=01 Month=01 Year=22

Using the UPS PIco HV4.0 HAT

The **UPS PIco HV4.0 HAT** is a complete and flexible cable/battery power management system, that also provides a protection from cable powering losses and save the SD card from corruption (the UPS functionality). In addition, it is offering a plenty of additional features that make it unique on the market. Compared with other similar Raspberry Pi® UPS or Powering Systems is the most advanced than any other. The usage and their capabilities will be described here below. There have been divided in following entities:

- Running the System for the first time
- System Functionality and Features
- The **UPS PIco HV4.0** HAT Cable Powering Inputs
- Enhanced Battery/Supercapacitor Power Back-up System
- The **PICo** I^2C Registers **Interface**
- Enhanced **RS232** Features
- User Applications Hardware Interfaces
- Measuring and Monitoring System
- Basic System Scheduler
- Events Triggered RTC Based System Actions Scheduler

Running the System for the first time

Once proceeded with Hardware and Software installation, user can start using of the complete system. Ensure that **UPS PIco HV4.0 HAT** is properly placed on the Raspberry Pi® top, and spacers are screwed. Plug-in the battery to the **BT1** socket (battery can be plugged/unplugged also when system is running - cable powered, however we recommend to plug-it from the beginning) and apply power to the Cable Power Inputs. They can be the Raspberry Pi® micro-USB, USB type C, the EXT (6.5-32V DC) power or the Passive PoE. Most powering inputs can be supplied at the same time. The only restriction is **do NOT supply Passive PoE and EXT** at the same time. **UPS PIco HV4.0 HAT** is protected with ZVD circuits and both powering sources can be supplied at the same time without any problem. If system is used in “on the go mode”, as exclusively battery powered system (as an Intelligent Power Bank), battery should be plugged-in before system will be switched with Magic Switch.

The Magic Switch can be used ONLY if before of use a proper register have been setup. This procedure is detailed described in next chapter. Using of Magic Switch before a proper setup it, can cause unexpected effects like (absence of possibility to switch OFF with Magic Switch, or absence of File Save Shutdown). Therefore, it is required to setup system for the first-time using cable powering, setting the Magic Switch Register (if planned to be used) and then use system as an Intelligent Power Bank powered.

After cable power applying Raspberry Pi® will start booting and during that time the UPS Blue LED will lit continuously. After about 30-40 seconds when Raspberry Pi® boots-up and properly installed Daemons starts running the UPS LED should be blinking about every 750 ms as far Cable Power is still connected. If the UPS LED is not blinking, that means the Daemons are wrong installed, and user

need to check the installation process again. If the UPS LED is blinking properly remove any cable power applied and the UPS LED should be blinking much slower – once every 2 seconds. These two steps ensure you that the Daemons are installed correctly and **UPS PIco HV4.0 HAT** running properly. Your system is ready and protected. If you will not apply the cable power again, after 60 seconds of running on battery, your system will be forced by **UPS PIco HV4.0 HAT** to safe shutdown. If Cable power will be applied, your system will boots-up and start running again. This is the basic usage, and if you have recognized all stages, you are ready. Enjoy your new **UPS PIco HV4.0 HAT** installed and protecting your system. For furthermore advanced usage you need to follow the next chapters.

If the Magic Switch is used (if previously is setup the appropriate register) - switch it ON without cable power applying. The system starts booting up, Raspberry Pi® will start booting and during that time the UPS Blue LED will lit continuously. After about 30-40 seconds when Raspberry Pi® boots-up and properly installed Daemons starts running the UPS LED should be blinking about every 2 seconds as far Magic Switch is ON. If the UPS LED is not blinking, that means the Daemons are wrong installed, and user need to check the installation process again. If you move the Magic Switch to position OFF again, then Safe Shutting Down will be started, UPS LED will light continuously, and after 30-40 seconds system shutdown and disconnect battery source.

System Functionality and Features

The **UPS PIco HV4.0 HAT** core functionality is to provide powering battery back-up and protect the Raspberry Pi® system from micro-SD card corruption if power loss occurs during writing to micro-SD card as also supplying the Raspberry Pi® based systems as Intelligent Power Bank with Safe Shutdown when OFF.

However, due to implementation of enhanced battery powering system it can be used for any kind of Battery or Cable Powered Application.

The **UPS PIco HV4.0 HAT** is plugged on top of the Raspberry Pi® and it is continually monitoring the GPIO 5V Pins. The proprietary self-learning algorithm analyzes the powering status on these 5V GPIO's and recognizes when cable powering is going to be lost. If so, then within 10 us applies the Battery Back-Up power and when cable power returns release it. The **UPS PIco HV4.0 HAT** powering analyzer check the stability of the cable powering and only if it is stable for more than 5 seconds release the battery power Back-up returning to Cable powering.

In case of usage as Intelligent Power Bank (without Cable Power Source) the checks the system power and when is switching OFF ensure that system will be properly shuttled down, without SD card corruption

All functionality of the **UPS PIco HV4.0 HAT** can be monitored or changed/forced via enhanced set of System Variables (System Registers) accessed through the I^2C interface. This Interface is described in detail in another chapter. It is called **P**eripherals **I**^{2}C **Co**ntrol Interface - the **PICo** Interface - and practically allows user to change most of system parameters via command line (if SSH or Terminal is used) or via any language interface (Python, C, C++, etc.). Some of the System Parameters can be also monitored via Raspberry Pi® using minicom® by using of the Raspberry Pi® Serial Port (if it is released for other applications) or again higher-level language interfaces.

The **PICo Interface** is occupying pre-defined (with possibility to change their location) addresses on the Raspberry Pi® address I^2C space. By default, they are 0x68, 0x69, 0x6A, 0x6B, 0x6C, 0x6D, 0x6E, 0x6F. In next chapters will be analyzed how to use of these System Registers. There are specified in the Table **UPS PIco HV4.0 HAT** I^2C addresses.

The installed software for interaction with the Raspberry Pi® (Daemons), is using the ONLY the I^2C GPIOs (informing that the Raspberry Pi® is running and initiating the Safe Shutdown if/when needed). No others GPIO are used and allows to be used to other application. The Daemon is monitoring these I^2C GPIOs and fire-up and interrupt on the Raspberry Pi® side. This approach is very flexible and does guarantee that interaction even if huge files are copied and/or Raspberry Pi® is ultra-busy with other tasks. It is allowed to use all GPIOs for any other Raspberry Pi® functionality.

The UPS PIco HV4.0 HAT Cable Powering Inputs

The **UPS PIco HV4.0 HAT** can be powered with various cable power sources:

1. Raspberry Pi Micro USB socket
2. Raspberry Pi USB Type C socket
3. Directly GPIO 5V0 input (user need to take **increased care** about polarity)

4. External Powering Input 7VDC – 32 VDC
5. Passive PoE Input 7VDC – 24VDC
6. External Powering on PPoE Devices Input 7VDC – 24VDC

Powering with any sources (1-3) and (4-5) can be done at the same time as powering sources are electrically isolated with ZVD circuits. Powering with (4) and (6) is not allowed to be done at the same time as there is risk for a damage of the devise. **Therefore, if system is powered via Passive PoE it is not allowed to provide at the same time cable power at the same time via EPR**. However, it is allowed to be powered via Raspberry Pi's USB socket at the same time.

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

The **P**eripherals **I**^{2}C **Co**ntrol – The **PICo Interface** – is an implementation of I^2C 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 HV4.0 HAT** peripherals are made extremely simple. Control at programming language level is also possible and easy. The core concept of the **UPS PIco HV4.0 HAT** 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.

There are two types of registers available:

- **Common**

  where data are stored in the same place and any change on it will cause action on the **UPS PIco HV4.0 HAT**.

- **Mirror**

  where are copy of data stored on internal variables of the **UPS PIco HV4.0 HAT**, they are protected, so changes on it will not implies the **UPS PIco HV4.0 HAT** functionality and will be overwritten immediately when **UPS PIco HV4.0 HAT** recognized changes on them.

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

0x68 -> **UPS PIco HV4.0 HAT** RTC access via HWCLOCK

0x69 -> **UPS PIco HV4.0 HAT** Status Registers Specification

0x6A -> **UPS PIco HV4.0 HAT** Hardware RTC Registers Direct Access Specification (only for reading)

0x6B -> **UPS PIco HV4.0 HAT** Commands

Events Triggered RTC Based System Actions Scheduler Commands

0x6c -> Start Time Stamp

0x6d -> Actions Running Time Stamp

0x6e -> Events Stamp

0x6f -> Actions Stamp

The location address of them can be changed. This procedure is described in next chapters.

System Cold Start, Warm Start, Default Start, "on the Go" Start and UPS LEDs behaviors

Cold Start

Cold Start is called when Cable Power is applied for the first time to the System after battery connection, the Raspberry Pi ® is starting up, and **UPS PIco HV4.0 HAT** is using all parameters stored in the internal EEPROM (default or user changed).

This start-up is called Cold Start and means that System is starting up for the first time without power cycling as battery is connected for the first time.

Note: If you are doing a Cold Start, and battery is connected, as the Raspberry Pi® is protected also during the booting process, it is enough to connect cable power just for 2 seconds. The System will continue starting-up with battery power back-up (without cable power applied).

Warm Start

This is the most used, and normal type of System Start-up. It happens when System is Cable Power or FSSD button is pressed, after Safe Shutdown of the system, and **UPS PIco HV4.0 HAT**. This start-up is called Warm Start and means that System is starting up from Low Power Mode (Power Cycling), RTC is running as system is battery powered and is in Low Powering Mode.

Note: If system is Warm Started, the Cable Power need to be applied for minimum 8 seconds to be recognized. This is the most used System Startup.

Default Start

When System is Cable Power user has a possibility to restore the factory defaults. To do that the following steps need to be followed:

- Press and hold the UR button
- Continue to hold the UR button, and press and hold the B button.
- Release the UR button, but keep holding the B button
- Release the B button

Then "walking LEDs" will be visible (for about 2 seconds) during that time the Internal **UPS PIco HV4.0 HAT** Flash Memory will be erased and written with factory default values, including factory default battery. After that the system will start running normally with new (default) settings.

"On the Go" Start

This is the Start when **Magic Switch** is programmed and put on the position ON. On this start system behavior as Intelligent Power Bank. System is starting with putting the Magic Switch to position ON and shutting down when put it to position OFF.

Battery Powering Protection

Due to shipping regulations in some countries, it is required to ship the **UPS PIco HV4.0 HAT** with battery connected, however without system to be powered. Therefore, to cover this requirement, a dedicated battery connectivity protection system has been implemented. It works in the following way:

When system is not cable powered (via Raspberry Pi ® or via External Powering) connecting of battery does not cause system powering, as connected to the **UPS PIco HV4.0 HAT** battery is in fact electrically disconnected. It has been implemented by using a high current/ultra-low resistance MOSFET switch (16 mOhm/10A) in default (hardware forced to OFF condition).

There is no possibility to start the system (even if battery remain connected to their socket) until External Cable (to Raspberry Pi® USB socket, External Power to **UPS PIco HV4.0 HAT**, PPoE, GPIO 5V) Power applied, or Magic Switch is switched to position ON.

During External Cable Power, battery can be connected or disconnected by user at any time. Only if battery is connected system is offering SD card protection, and UPS functionality.

If user wish to disconnect electrically the battery from the system, should press the R button for more than 2 seconds, after system FSSD (File Safe System Shutdown) with disconnected Cable Power powering. This will cause an electrical disconnection of the battery from the system. Note that RTC will be not working after that. Restarting system in such condition need to apply **Cable Power** powering again.

Power Monitoring Automatic Algorithm over GPIO (5V) pins

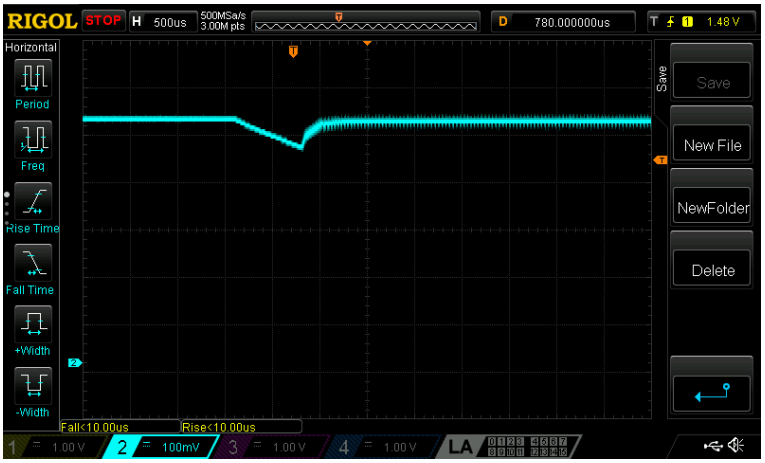
All **UPS PIco HV4.0 HAT** models have implemented a power monitoring algorithm in order to detect power loses or instability. This algorithm is monitoring every 16 us the 5V over GPIO pins (independently if **UPS PIco HV4.0 HAT** is power via Raspberry's USB, EPR, PPoE or GPIO 5V input). The feature that allows to monitor powering via GPIO offering a huge flexibility in enclosures selection as practically any enclosure can be used, as no extra holes are required.

This Power Monitoring Algorithm is fully automatic, self-learning, and adopting to very specific powering needs.

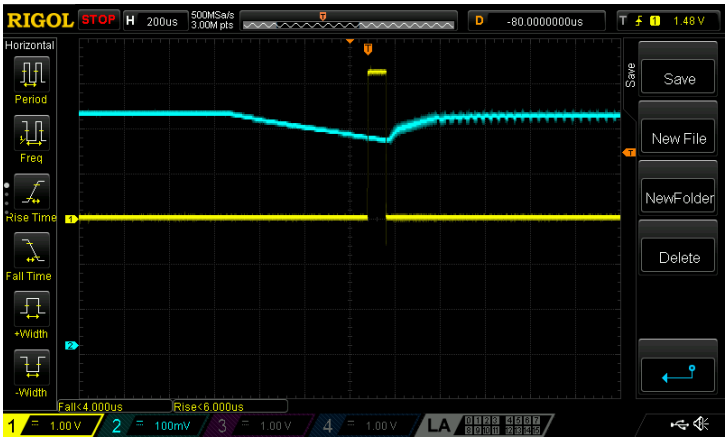
There is no need to any user action. Just take it and use it. If the powering conditions are unique after a time of 15-30 minutes system will learn them and adopt to these specific conditions or inform user that cable powering conditions cannot be handled by the system i.e. USB voltage is too low (4.5V)

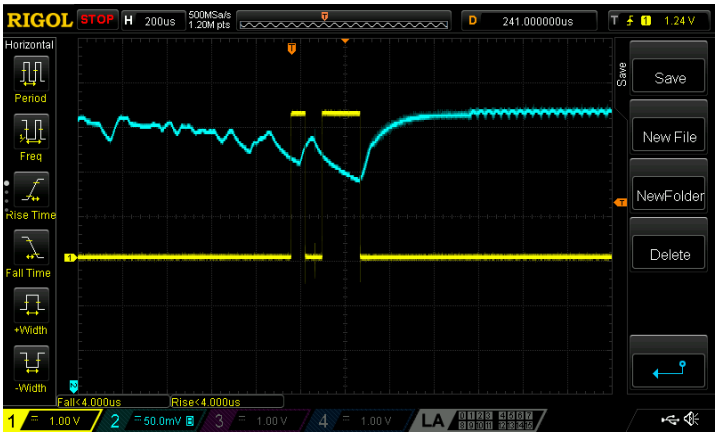
How Power Monitoring Works?

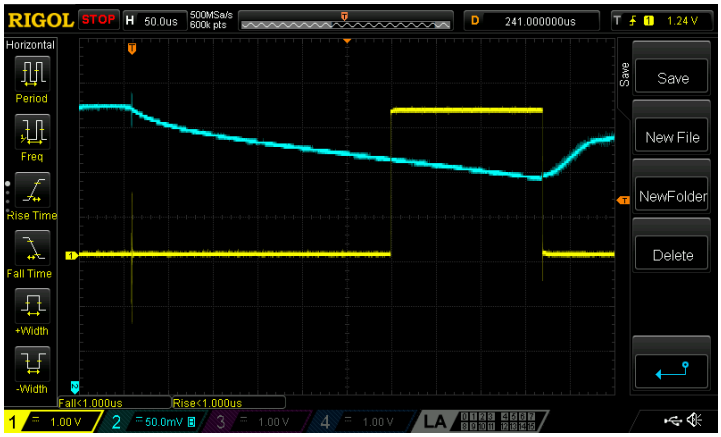
For power monitoring is used an automatic A/D conversion with thresholds, that is checking every 16 us (within an interrupt) the 5V GPIO pins. Power loses, or instabilities are continuously monitored and if are dangerous for system running are immediately recovered by battery backup. This back-up is working until powering of the system stabilize and automatically return to cable powering after shorter or longer time needed. When **UPS PIco HV4.0 HAT** detects power fluctuation automatically switches to ultra-fast 8us checking and monitor it within pre-defined time window.

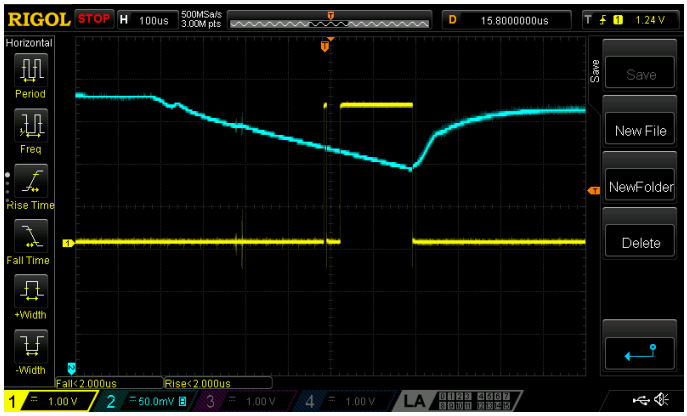


Here above is shown typical power loose event, that has been recovered within 120 us. The dropping down powering voltage on GPIO pins was detected and filtered (from possible unwanted powering spikes), and in specific time recovered by switching ON to the 5V boost supplied form integrated battery. On below picture is visible (the yellow line) how **UPS PIco HV4.0 HAT** was checking the powering status and when time frame passed switches immediately to battery powering backup.


RIGOL
STOP
H 200us
500MSa/s
3.00M pts
D -80.0000000us
T 1.48 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
Fall<4.000us
Rise<6.000us
1 1.00 V
2 100mV
3 1.00 V
4 1.00 V
LA


RIGOL
STOP
H 200us
500MSa/s
1.20M pts
D 241.000000us
T 1.24 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
Fall<4.000us
Rise<4.000us
1 1.00 V
2 50.0mV
3 1.00 V
4 1.00 V
LA


RIGOL
STOP
H 50.0us
500MSa/s
600k pts
D 241.000000us
T 1.24 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
Fall<1.000us
Rise<1.000us
1 1.00 V
2 50.0mV
3 1.00 V
4 1.00 V
LA



Disabling the UPS PIco HV4.0 HAT Battery Back-up functionality

The **UPS PIco HV4.0 HAT** in some cases (mainly when located/installed remotely) need to be disabled (permanently or temporary) the Battery Back-up functionality. The current firmware offers this feature. When battery backup disables, Raspberry Pi® is then not protected on cable power lose. If such cable power loss happens, then Raspberry Pi® will stop working immediately. It is possible that micro-SD card will be destroyed. Therefore, users using this feature need to do it very carefully. This battery back-up feature does not imply other **UPS PIco HV4.0 HAT** functionality like Relay, A/D Converter, Buzzer, User LEDs, RS232, etc. The embedded RTC will be not working in case of disabling of battery back-up feature.

If this feature will be activated, the integrated battery will be electrically disconnected from the system.

If this feature will be activated, the integrated battery will be electrically disconnected from the system.

The Magic ON/OFF (Slide) Switch functionality

The **UPS PIco HV4.0 HAT** is equipped with an ON/OFF **Magic** Slide **Switch**. This Switch is called Magic, as it is multifunctional, programmable, and adding a huge difference in powering schemes of the Raspberry Pi®. It is allowing to use Raspberry Pi ® as an independent device powered exclusively from battery, without any cable powering source. This powering feature of **Intelligent Power Bank**, is called "**on the Go**"

First, the Magic Switch is not soldered by default, and it is not necessary to be soldered. It is required only if user is planning to use the **UPS PIco HV4.0 HAT** as a mobile battery powered application. In all other cases, absence of this switch does not imply the standard functionality of the **UPS PIco HV4.0 HAT**. If for any reason, the ON/OFF Magic Switch has been soldered, but user does not need his additional functionality, should be placed **on position OFF** and cannot be programmed as active feature (0xAA). Also, if user need to have external access to Intelligent ON/OFF functionality, then an external ON/OFF Switch will longer cables can be soldered instead of the micro switch.

The following functionalities are assigned to the Magic Switch:

- Intelligent ON/OFF when ON, without Cable Power Present
- If during the ON cable power inserted, then battery is charged.
- Intelligent ON/OFF with Files Save Shutdown when OFF without battery power cut (Integrated RTC is running, schedulers are running). Battery consumption is not zero (not implemented yet)
- Intelligent ON/OFF with Files Save Shutdown when OFF and battery power cut (Integrated RTC is not running, schedulers are not running). Energy consumption from Battery is zero

Here below are shown the **Magic Switch** functionalities based on Switch position and programming registers values. The following PICo registers are associated to the Magic Switch functionality:

Magic Switch State	Assigned Functionality	Programming Register values	UPS Pico HV4.0 HAT Version	Initial State	System Powering Behaviors
Absent or not soldered	Standard Power Cycling. System must be initialized with inserting of the cable powering	Not Affected Do not program any value (keep with default value of 0x00 or 0xFF)	UPS Pico HV4. HAT Stack UPS Pico HV4. HAT Advanced UPS Pico HV4. HAT PPoE	Default State	Standard FSSD functionality with other keys and Low Powering as described in other chapters
Soldered original slide micro switch or external slide switch in position OFF	Standard Power Cycling. System must be initialized with inserting of the cable powering	Not Affected Do not program any value (keep with default value of 0x00)	UPS Pico HV4. HAT Stack UPS Pico HV4. HAT Advanced UPS Pico HV4. HAT PPoE	Default State	Standard FSSD functionality with other keys and Low Powering as described in other chapters
Soldered original slide micro switch or external slide switch in position ON	Allowed ONLY when activated or during programming (activating)	Not Affected on_the_go=0xAA	UPS Pico HV4. HAT Stack UPS Pico HV4. HAT Advanced UPS Pico HV4. HAT PPoE	Default State	Standard FSSD functionality on low battery

Soldered original slide micro switch or external slide switch in position OFF	FSSD of the system and battery power OFF	Not Affected on_the_go=0xAA	UPS Pico HV4. HAT Stack UPS Pico HV4. HAT Advanced UPS Pico HV4. HAT PPoE	Default State	FSSD of the system and battery power OFF

Setting up (activating) the Magic Switch

With Cable Powering connected (USB or EPR or PPoE) do the following:

1. Make sure that **Magic Switch** is on Position ON (on the LEDs side)

2. Check the Magic Switch Register

sudo i2cget -y 1 0x6b 0x16

3. should be 0x00 or 0xff

4. Then activate (program) it

sudo i2cset -y 1 0x6b 0x16 0xaa

5. Remove Powering Cable

6. System is still running, but on battery mode

7. Switch the Magic Switch to OFF position

8. System start shutdown, and after short time cut the power

9. Whenever you like can make it ON/OFF (of is always with system shutdown first)

IMPORTANT NOTICE1: When the Magic Switch Register is activated, the battery running register is automatically set to 0xff, so system is running on battery until it is low, then system automatically shut down. If you like to have it running shorter, you need to reprogram the register for a shorter time with just after Magic Switch programming

IMPORTANT NOTICE2: It is possible to charge the battery with cable power with the Magic Switch, but before entering the cable system must be running – Magic Switch put to ON state.

The UPS PIco HV4.0 HAT Battery Type/Profile Selection

The **UPS PIco HV4.0 HAT** is supporting the following chemistry battery types:

- the LiPO with nominal voltages 3.7V and charging 4.2V
- the LiFePO4 with nominal voltages 3.2V and charging 3.65V
- the Li-Ion with nominal voltages 3.65/3.7V and charging 4.2V
- the NiMH/NiCd with nominal voltages 3.6V (3 internal cells) and charging 4.2V
- the SAL (Lead Acid) with nominal voltages 2.2V (1 internal cells) and charging 2.45V

The **UPS PIco HV4.0 HAT** is sold by default with the small LiPO 450 mAh battery. However, for dedicated applications user can choose one of the other batteries' chemistries supporting by **UPS PIco HV4.0 HAT**. The main differences between various batteries chemistries are:

- Charing temperature range
- Discharging Temperature Range
- Number of charging/discharging cycles
- Battery Cost
- Power density (how big is battery with dedicated capacity)
- C factor (how much current battery can provide when discharging)

Battery Chemistry	Charing temperature range	Discharging Temperature Range	Number of charging/discharging cycles	Power density
LiPO	0°C to 40°C	0°C to 40°C	400-450	Very high
LiFePO4	0°C to 40°C	0°C to 45°C	2000	High
Li-Ion	0°C to 45°C	–20°C to 60°C	300-400	Highest
NiMH/NiCd	0°C to 45°C	–20°C to 65°C	200	low
SAL (Lead Acid)	–20°C to 50°C	–20°C to 50°C	200	Very low

From Electrical point of view the most important factors for battery are the working temperature and C factor. These two factors are defining if the battery is good for dedicated application or not. The working temperature (Charging/Discharging) is defined based on application working conditions. Users need to consider all scenarios to be sure that battery will be all the time within the specified temperature conditions. In example, if system is placed in an isolated plastic case, and temperature (due to Raspberry Pi) is all the time increasing, then the internal temperature (so also the battery) will be high. Especially for Lithium based batteries temperature is a very important factor. The second and very important is the C factor.

Maximum Current supplying by battery (C Factor)

The C factor for any battery defines how much current can be drawn from battery when discharged. It is different for each battery chemistry and type. In example the standard 450 mAh battery that comes with **UPS PIco HV4.0 HAT** are 15C. That means the max current provided by this battery when discharged is 15 x 0.450A=6.75A!!! Indeed, this small battery can provide near to 7A of current.

User Application Current consumption calculation example

When designing battery powered system application (even with battery backup), it is very important to know/estimate the power/current requirements. The below calculations need to be used as a guideline for any Battery Powered Application. For the calculations we will make some assumptions that can be adjustment to specific User Application if/when needed.

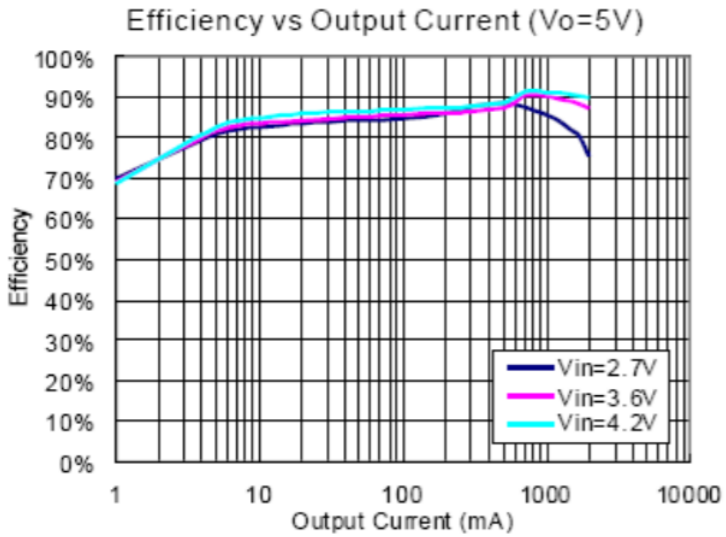
Below calculations will be based on power consumption of the Raspberry Pi 4 Model B. This model has the following power consumption estimations:

- 575 mA while idling
- 885 mA while LXDE is being loaded
- 600 mA to view 1080p video
- 640 mA to record 1080p video

USB devices (HDD, LTE, GSM, etc) connected to the Raspberry Pi draws their current @5V. User need to take into considerations the maximum (peak) current required by this hardware.

Our calculations will be done for the standard LiPO 450 mAh battery of 15C

For our calculation we assume that Raspberry Pi and all other connected devices draw totally 1A@5V



Calculation Algorithm

Calculate the total power required for 5V.

For our example it is 5V x 1A = 5W @ 5V

Calculate the total power required if system is powered by 3.7V (in our example LiPO battery)

5WA/3.7 V = 1.35 A @ 3.7V

Calculate the power losses due boost converter used. The **UPS PIco HV4.0 HAT** for generating the 5V from battery is using technology called boost converter that converts the battery voltage to 5.2V. We use the best technology provided in the market; however, this conversion always causes a power loss that are loosed on thermal (a part of supplied energy is converter to thermal). The Boost converter loses are changing depending to battery voltage. In example when battery is full changed, and their voltage is 4.2V, loses are small (about 10%) if voltage drop down when battery is discharging, loses increases. We propose to use mean value of 20% for 3.7V and 25% for 3V. Therefore, for our calculations case we will use 20% of thermal loses (called also boost converter efficiency – indicated with Greek letter of "η"). We need to multiply the current consumption by loses in order to know the max current requirements from battery.

1.35 A * 1.2 = 1.62A @ 3.7V when battery powered, or 5.99 W @3.7V

This result means that if we are powering Raspberry Pi that totally uses at 1A @5V (5W) we need to supply by battery 1.62A @ 3.7V.

As our battery is 0.45 A 15C can supply the system with 1.62A @ 3.7V. (max current of this battery is 0.45A x 15 = 6,75A)

This result allows us to easy calculate how long last 0.45 A 15C 3.7V battery when system is powered by this battery.

It is 0.45A/1.62A = 2.7 minutes -> 1.5-2.0 minutes of real working time (please always use lower numbers, as battery is not going to be completely discharged, we have also some loses on battery cables, PCB tracks, MOSFET switches etc.)

Selecting of a proper battery for your application is one of the core requirements when designing a new system.

User can select one of supported batteries, however, always need to consider the worst case for current consumption and working temperature. In addition, need to use 22AGW (existing already in all our batteries or battery holders) cables and relatively short to avoid voltage drops on them.

Our company is offering a plenty of different batteries that are supporting the **UPS PIco HV4.0 HAT** applications.

We always offer the best quality batteries with the best prices. There are 4 new add-ons for that can hold different batteries chemistry supply. They are:

- PIco Double Li-Ion 18650 Battery Holder
- PIco Single LP/LF, Li-Ion 18650 Battery Holder
- PIco triple NiHM/NiCd AA Battery Holder
- PIco triple NiHM/NiCd AAA Battery Holder

In addition, our company supports with various batteries capacities and various chemistries, that can be directly connected to the **UPS PIco HV4.0 HAT**

There are:

- The standard LiPO battery 450 mAh which comes with the **UPS PIco HV4.0 HAT** – 15C
- The enhanced LiPO battery with capacity 4000 mAh - 2C
- The enhanced LiPO battery with capacity 8000 mAh – 2C
- The enhanced LiFePO4 battery with capacity 3000 mAh 2C
- The enhanced LiFePO4 battery with capacity 4000 mAh 2C
- The enhanced LiFePO4 battery with capacity 8000 mAh – 2C
- The Li-Ion Battery 10400 mAh – 2C
- The LiPO battery with capacity 1500 mAh – 2C

Batteries with different chemistry offers different unique features and needs to be specified on the system setup when changed. This ensure that **UPS PIco HV4.0 HAT** will adjust the proper charging/working profile. It is not allowed to use different batteries with not related profile to them. There is no protection from this action, therefore using of a wrong profile can cause an unexpected result. Exceptions on this rule is LiPO and Li-Ion batteries where differences are very small, however it is also to them recommended to use a proper battery profile. It is strongly recommended to use LiPO, Li-Ion, and LiFePO4 batteries equipped with protection PCM (PCB) that protect batteries from over current, overcharge and over discharge in addition to protections provided by the **UPS PIco HV4.0 HAT**.

The Supercapacitors Support – a unique feature of UPS PIco HV4

The **UPS PIco HV4.0 HAT** is equipped with unique feature that support of Supercapacitors as a power backup. This feature is unique as the Supercapacitor(s) can be used alone (as the only Power Backup source) or together with battery, and on short power absence can be used the Supercapacitor and for longer one Battery Backup.

What is Supercapacitor?

A supercapacitor is like a capacitor except for the bigger area of its plates and the smaller distance between these plates. The plates are metallic and are soaked in electrolytes and are separated by a very thin insulator. An electric double layer is created in the supercapacitor as opposite charges are formed on both sides of the separator when the plates are charged. This results in a supercapacitor with greater capacitance. In other words, the combination of plates and the larger effective surface area enables a supercapacitor to have greater capacitance and higher energy density. Unlike a battery, a supercapacitor has an unlimited life cycle, with little wear and tear on long-term use. Thus, it can be charged and discharged an unlimited number of times (in real word about 500K – 1M times).

A supercapacitor has many advantages. It can deliver high power and enable high load currents owing to its low resistance. Its charging mechanism is simple and fast and is not subject to

overcharging. Compared to a battery, a supercapacitor has excellent high- and low-temperature charge and discharge performance. It is also highly reliable and has low impedance.

A supercapacitor has certain limitations including its high cost and the high self-discharge involved. Moreover, unlike a regular battery, it has low specific energy, and its use of the full energy spectrum is hindered by linear discharge voltage.

Because of their properties, supercapacitors are used in many applications. They are widely deployed to deliver power and bridge power gaps. They are a replacement for batteries in certain settings such as in battery-free devices.

Comparing a Supercapacitor and a Battery

There are unique differences between the battery and the supercapacitor. The battery's chemistry determines the operating voltage, electrochemical reactions control charge and discharge. The capacitor is not electrochemical, and the maximum allowable voltage is determined by the dielectric material type which separates the plates. Since the supercapacitor is non-chemical, the voltage is free to keep rising till the dielectric fails. This usually occurs as a short circuit, so it is advisable not to go above the specified voltage. The supercapacitor is not a battery replacement to store long-term energy. When the charge and discharge times are more than 60 seconds, a battery should be used; if less, then the optimum solution is a supercapacitor. Supercapacitors are perfect for quick charge and to fill a short-term power requirement, whereas batteries are better for providing long term energy. Combining the two in a hybrid battery can satisfy both requirements while reducing the battery stress, which leads to a longer service life.

The following Supercapacitors are offered and available for **UPS PIco HV4.0 HAT** to be used:

- Super Capacitor of 100F (it is soldered just on top of the **UPS PIco HV4.0 HAT**)
- Super Capacitors Bank of 300F (a separate HAT with its own charger)
- Super Capacitors Bank of 500F (a separate HAT with its own charger)
- Super Capacitors Bank of 800F (a separate HAT with its own charger)

Depending on the application and model of the **UPS PIco HV4.0 HAT** is offering various configurations of power backup source.

Basically, there are 2 Mods of use of the Power Backup source:

- The Single Mode
- The Mixed Mode

In the **Single Mode** only one type of source can be selected (any battery chemistry type or Supercapacitor), and this one will be used for the power back up if cable powering is lost.

In the **Mixed Mode** two types of sources can be selected (any battery chemistry type and Supercapacitor), and both will be used in power backup process. Selection what source is used at the current mode depends to level of charging of Supercapacitor, duration of cable power lost and is automatically done by the **UPS PIco HV4.0 HAT** based on their firmware.

The selection of Backup Power Source can be done via writing power source to dedicated PICo register:

sudo i2cset -y 1 0x6b 0x07 ***XX***

Where the **XX** is the Power Backup source. The following values are allowed:

Single Mode			
LiPO	'L'	0x4C	
Li-Ion	'I'	0x49	
LiFePO4	'F'	0x46	
NiMH	'H'	0x48	
SAL	'A'	0x41	
ISC	'C'	0x43	Internal Supercapacitor (100F)
ESC	'D'	0x44	External Supercapacitor Bank (300F-500F-800F) HAT
Mixed Mode			
LiPO + Supercapacitor	'l'	0x6C	Internal Supercapacitor (100F)
Li-Ion + Supercapacitor	'i'	0x69	Internal Supercapacitor (100F)
LiFePO4 + Supercapacitor	'f'	0x66	Internal Supercapacitor (100F)
NiMH + Supercapacitor	'h'	0x68	Internal Supercapacitor (100F)
SAL + Supercapacitor	'a'	0x61	Internal Supercapacitor (100F)

NOTE: In the Mixed Mode only 100F Supercapacitor is allowed in the current version of firmware.

Each model of the **UPS PIco HV4.0 HAT** can be Assembled as Model B, BC and D. The assembly defined the handling of the Supercapacitor. Therefore, assemblies are defined as:

- Assembly **B** that supports Both Power Sources (Supercapacitor and Battery), but only one back up power source can be used at once
- Assembly **BC** that supports Both Power Sources (Supercapacitor and Battery), and both can be used at the same time
- Assembly **D** that supports only Battery as power Back-up, and does not support Supercapacitor at all

If a wrong selection of Power Source is done (not supported by appropriate model) it will be by firmware automatically rejected. The firmware is common for all models; however, it is smart and know on what model of **UPS PIco HV4.0 HAT** is running.

However, it is not allowed to have connected Supercapacitor/Bank and Battery at the same time on Assembly B, due to hardware implementation. Only one Power Backup Source can be connected at once. Having soldered Supercapacitor and battery can cause damage of the Supercapacitor.

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x07 0x4C

Selects power source the LiPO Battery (default selection of the firmware) - (**Single Mode**)

sudo i2cset -y 1 0x6b 0x07 0x43

Selects power source the 100F Supercapacitor as the ONLY Power Backup source - (**Single Mode**)

sudo i2cset -y 1 0x6b 0x07 0x63

Selects power source the 100F Supercapacitor and LiPO Battery as a Power Backup source (**Mixed Mode**). Therefore, depending to level of charging of the Super Capacitor and duration of Cable Power loose system decide to use Supercapacitor and Battery

Supported Supercapacitor Types

Due to enhanced boot converter Integrated Circuit implemented in the **UPS PIco HV4.0 HAT**, it is producing 5V2 already from voltage level of 0V5 and is start from 1V8. Therefore, the biggest level of energy stored in the Supercapacitor is used. However, firmware allows to start up the boost converter only if Supercapacitor level is higher than 2V5. This is indicated by the SCA Led lit above the 2V5 level of Supercapacitor. The Supercapacitors supported by the **UPS PIco HV4.0 HAT** are rated to 2V8 and all threshold of the system are designed based on this. It is not allowed to use different type if Supercapacitors than this one approved by manufacturer of the **UPS PIco HV4.0 HAT.**

If used the 100F supercapacitor soldered on the **UPS PIco HV4.0 HAT** directly, it is charged with maximum current of 1A2 @2.8V. Using PWM technology when needed during the charging process. The charging time takes about 3-5 minutes.

If used the 300F, 500F or 800F supercapacitor HAT, it is charged with maximum current of 3A0 @2.8V. Using PWM technology when needed during the charging process. The charging time takes about 3-8 minutes. For the PWM is used one of the GPIO PWM pins. Users have possibility to select by jumper which one will be used:

- GPIO18
- GPIO12
- GPIO13
- GPIO19

If used If used the 300F, 500F or 800F supercapacitor HAT user need to activate the appropriate lines in the **pico_i2c.py** and reload the service.

LED INDICATIONS:

The **SCA LED** lit if the level is higher than 2V5

Selecting the proper Backup Powering Mode

There are 2 Mods of use of the Power Backup source:

- The Single Mode
- The Mixed Mode

In the **Single Mode** only one type of source can be selected (any battery chemistry type or Supercapacitor), and this one will be used for the power back up if cable powering is lost.

In the **Mixed Mode** two types of sources can be selected (any battery chemistry type and Supercapacitor), and both will be used in power backup process.

They can be defined according to use of backup sources as:

- Battery Mode (Only Battery is used)
- Supercapacitor Mode (Only Supercapacitor is used)
- Mixed Mode (Supercapacitor and Battery Mode together with automatic switching)

These Backup Powering Modes can be used depending to Assembly PCB Version. What Version of PCB user has can be determined by reading the following PICo register at **0x69@0x36** by using the following command:

sudo i2cget -y 1 0x69 0x36

The answer will show to the user what backup powering modes are available, following below table. Please kindly notice that system is smart enough to reject wrong modes (if not allowed), however soldering Supercapacitor if not allowed can destroy it (on current version of firmware). Changing of Backup Powering Modes is very simple and can be done by simple changing of battery/Supercapacitor type. According to it **UPS PIco HV4.0 HAT** will automatically select a proper charging profile.

The selection of Backup Power Source can be done via writing power source to dedicated PICo register:

sudo i2cset -y 1 0x6b 0x07 ***XX***

Where the **XX** is the Power Backup source. The following values are allowed:

Single Mode			
LiPO	'L'	0x4C	
Li-Ion	'I'	0x49	
LiFePO4	'F'	0x46	
NiMH	'H'	0x48	
SAL	'A'	0x41	
ISC	'C'	0x43	Internal Supercapacitor (100F)
ESC	'D'	0x44	External Supercapacitor Bank (300F-500F-800F) HAT
Mixed Mode			
LiPO + Supercapacitor	'l'	0x6C	Internal Supercapacitor (100F)
Li-Ion + Supercapacitor	'i'	0x69	Internal Supercapacitor (100F)
LiFePO4 + Supercapacitor	'f'	0x66	Internal Supercapacitor (100F)
NiMH + Supercapacitor	'h'	0x68	Internal Supercapacitor (100F)
SAL + Supercapacitor	'a'	0x61	Internal Supercapacitor (100F)

- Assembly **B** that supports Both Power Sources (Supercapacitor and Battery), but only one back up power source can be used at once
- Assembly **BC** that supports Both Power Sources (Supercapacitor and Battery), and both can be used at the same time
- Assembly **D** that supports only Battery as power Back-up, and does not support Supercapacitor at all

Assembly Version	**UPS PIco HV4.0 Model** *sudo i2cget -y 1 0x69 0x36*	**Backup Sources** *sudo i2cget/**set** -y 1 0x6b 0x07 **XX***	**Allowed Backup Mode(s)** *sudo i2cget -y 1 0x69 0x02*
Assembly **B**	**UPS PIco HV4.0 Stack** called **T** **UPS PIco HV4.0 Advanced** called **B** **UPS PIco HV4.0 PPoE** called **Q**	Battery LiPO 'L' 0x4C Battery Li-Ion 'I' 0x49 Battery LiFePO4 'F' 0x46 Battery NiMH 'H' 0x48 Battery SAL 'A' 0x41 Internal Super CAP 100F 'C' 0x43 External Super CAP 'D' 0x44 : 300F 'D' 0x44 500F 'D' 0x44 800F 'D' 0x44	BAT_BACKUP (0x01) SCA_BACKUP (0x03)
Assembly **BC**	**UPS PIco HV4.0 Stack** called **S** **UPS PIco HV4.0 Advanced** called **A** **UPS PIco HV4.0 PPoE** called **P**	Battery LiPO 'L' 0x4C Battery Li-Ion 'I' 0x49 Battery LiFePO4 'F' 0x46 Battery NiMH 'H' 0x48 Battery SAL 'A' 0x41 Internal Super CAP 100F 'C' 0x43 External Super CAP 'D' 0x44 : 300F 'D' 0x44 500F 'D' 0x44 800F 'D' 0x44 Battery LiPO+SC 'l' 0x6C Battery Li-Ion+SC 'i' 0x69 Battery LiFePO4+SC 'f' 0x66 Battery NiMH+SC 'h' 0x68 Battery SAL+SC 'a' 0x61	BAT_BACKUP (0x01) MIX_BACKUP (0x02) SCA_BACKUP (0x03)

Assembly **D**	**UPS PIco HV4.0 Stack** called **U** **UPS PIco HV4.0 Advanced** called **C** **UPS PIco HV4.0 PPoE** called **R**	Battery LiPO 'L' 0x4C Battery Li-Ion 'I' 0x49 Battery LiFePO4 'F' 0x46 Battery NiMH 'H' 0x48 Battery SAL 'A' 0x41	BAT_BACKUP (0x01)

Selecting the Power Backup Mode is done by storing a selected Battery/SCAP on the PICo register **0x6b@0x07** line below:

sudo i2cset -y 1 0x6b 0x07 ***XX***

If not allowed mode for current Assembly version system will recognize and not store it.

The Benefits and Differences of various Backup Sources

The pioneering construction of the **UPS PIco HV4.0 HAT** is offering to the user all possible powering backup sources in all possible configurations. Therefore, user can select one of them that cover exactly their application needs. However, there are some important differences between each Power Backup Mode that will be described here below.

BAT_MODE (Battery Mode) is the most common used Power Backup Mode and depending to what battery is used offers longer or shorter battery lifetime. It is usefully for applications where is needed to have long battery runtime (i.e. some hours). However, it is limited with battery chemistry and environmental conditions it is running. A special care needs to be taken with battery chemistry selection according to environmental conditions. This mode is recommended for application on controlled environment conditions however where cable power absence is often and stay for longer time.

SCA_MODE (Supercapacitor Mode) is the most resistive used Power Backup Mode to extreme environmental conditions. Due to extreme conditions environment of the Supercapacitor. It can be used with relatively small on-board Supercapacitor of 100F or additional HAT of 300F/500F/800F. The running time is limited due to stored energy as also charging time is long. System is protected after Supercapacitor is charged above 2.5V. However, once changed is protecting the system for unlimited time. This mode is recommended for application on difficult environment conditions however where cable power absence is rather occasional and not so often.

MIX_MODE (Mixed Mode) is the most advanced Power Backup Mode and provide an additional protection for the battery life. Due to embedded intelligence, system on cable power absence automatically switches to Supercapacitor Power Backup, and when level of Supercapacitor is below the 2.2V automatically switches to battery. This provides an enhanced lifetime for battery as used only on longer (usually more than 5 seconds) cable power absence. If, Supercapacitor is not charged yet, system automatic uses battery for Power Backup.

Battery Charger Monitoring and Control

The Embedded Battery Charger is controlled by **UPS PIco HV4.0 HAT** automatically, and according to cable powering condition is adjusting charging current as also switching charger ON/OFF. The implemented dynamic charging current adjustment is changing automatically charging current accruing to long time cable powering conditions. The state when **UPS PIco HV4.0 HAT** is switched charger ON/OFF can be monitored on the Charger Status Register that is located at the address **0x69 @ 0x1e, 0x1f**. A detailed description of it is provided here below:

0x1e	**BAT_chg_status**	**Byte**	**Mirror**	**Read**	**Information about charger IC program status.** Read: 0x00 – Charger IC is OFF and battery is not charged Read: 0x00 – 0x80 – Charger IC is ON and battery and charging current is set to 0x01-0x80 in 10th of mA NOTE: Programmed charging current is different from the real charging current, as real depends to how full battery is.
0x1f	**BAT_chg_real**	**Byte**	**Mirror**	**Read**	**Information about charger IC real status.** Read: 0x00 – Charger IC is OFF and battery is not charged Read: 0x00 – 0x80 – Charger IC is ON and battery and charging current is set to 0x01-0x80 in 10th of mA NOTE: Real charging current is different from the programmed charging current, as real depends to how full battery is. NOT Implemented YET

Some very specific applications, requiring having the ability to disable the charger, and enable only when some specific conditions met. In order to cover such applications a special Charger Control Register has been implemented. With this Register user can Enable/Disable charging feature. This Register us placed at the address **0x6b@0x19** and can be set only when system is cable powered. Under Normal Conditions charger is active when Raspberry Pi is running (SYS LED flashing fast), however user can set the charger to be active also when system is in Sleep Mode.

0x19	**CHG_ctrl**	**Byte**	**Mirror**	**Read**	**Information about charger IC program control** Write: 0x00 – Charger IC is OFF and battery is

					not charged Write: 0x01 – Charger IC is ON and battery is charged Read: Current Value at the time

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x19 0x00 **Disable the battery charger**

sudo i2cset -y 1 0x6b 0x19 0x01 **Enable Automatic Mode of the battery charger**

Low Battery LED and Beeper

By selecting battery type (chemistry) **UPS PIco HV4.0 HAT** firmware automatically sets the threshold for charging as also the threshold for the system shutting down on low battery. There is an information on user by switching ON the **BAT LED** and Start regular Beeping. This level is slightly higher of any low battery shutdown threshold selected by system or user. It is 0.2V higher than level of low battery level selected. When battery is low, user will see the BAT LED lit, and short beeping, after a short time, system will be automatic shutdown

Powering Modes

UPS PIco HV4.0 HAT powering selection and monitoring functionality is based on internal firmware-based **State Machine**. This state machine is deciding on Powering State (called also Powering Mode) based on various parameters like powering source, battery level, current level, RTC etc. The actual Powering Mode each time is stored in internal register and can be accessed by **PICo** interface over address **0x69@0x00.**

The following Powering Modes are available:

- CBL (which consist of all sub modes EPR (6.5-32VDC) and RPi (5V)
- BAT (which consist of all sub modes BAT and LPR)

User can at any time check the powering mode the system is from command line, or software interface, remotely or on site. The meaning is:

- 0x01 Powering from Cable (Raspberry Pi® or PPoE or External)
- 0x02 Powering from Battery (or Supercapacitor)

EXAMPES OF USE:

sudo i2cget -y 1 0x69 0x00

User should receive response 0x00 or 0x01.

UPS PIco HV4.0 HAT Low Powering functionality – Power Cycling

One of the most important features of the **UPS PIco HV4.0 HAT** is the **Power Cycling**. Power Cycling as specified before is the core firmware **State Machine** that is handling the whole system powering behaviors. The Power Cycling feature is handling the System Shutdown, System Start-up as also Battery/Supercapacitor Charging.

The following scenarios has been implemented in the current firmware version that are covering 100% of possible cases. The following Powering Sources and Backup Modes are available depending to model of the **UPS PIco HV4.0 HAT.**

Raspberry Pi® Shutdown/Wakeup Scenarios

The **UPS PIco HV4.0 HAT** is entering Low Powering Mode in the following scenarios

The Raspberry Pi based System running scenario is when System is powered by cable (via powering USBPR/PPoE Interface). If this cable powering fails system is automatic switching to the battery/supercapacitor backup and System (based on Raspberry Pi, PIco and possible additional

hardware) continues working. This change of powering source from cable to battery/supercapacitor is indicated to user by “beep” (if sounder is implemented), as also by changing of the UPS LED blinking speed (which is going to blink much slower than with a cable powering). User can indicate the powering mode by reading the

EXAMPES OF USE:

sudo i2cget -y 1 0x69 0x00

and see that it has been changed to battery powered. If this condition continues (lack of cable supply power) system after 60 seconds (user can program the battery running time) will initiate the **File Safe Shut Down Procedure** - **FSSD** and as a result System will enter to the **Low Powering Mode** – **LPR**. This is the basic and usual way to enter the **LPR** Mode (by absence of the Cable Powering)

Another way to entering the **LPR** mode is by pressing the F button. This will active immediately the **FSSD** procedure, and System after a required time to shutdown will enter the **LPR** mode. It can be done on all Cable Powering Sources, and as the result system will switch OFF the Raspberry Pi embedded powering system (by using the PE POGO pin) or will switch OFF the UPS PIco 4 Buck Converter.

The **UPS PIco HV4.0 HAT – Raspberry Pi System** can enter the **Low Powering Mode** (LPR) on the following cases:

- By pressing the F button
- By absence of Cable Powering (after programmed time frame or battery low)
- By Raspberry Pi® (*sudo halt*) command after 5 minutes of not running of Raspberry Pi®
- By Event from the Basic Scheduler or SAS ETR scheduler

During the **LPR** mode, the embedded **RTC** will continue working. The **wake-up** from the **LPR** mode can be done in the following ways:

- By pressing the F button
- By Applying the Cable Power to powering USB socket (or changing the powering conditions)
- By Applying the Cable Power to EPR/PPoE power input (or changing the powering conditions)
- By Event from the Basic Scheduler or SAS ETR scheduler
- By re-running the Raspberry pi if stopped once

User can also change powering conditions, by removing the EPR Cable, plug on the powering USB cable or re-entering the EPR Cable. On all that cases, due to ultra-low power implementation recognizing of the Cable Power re-entering is done every 10 seconds. Therefore, the longest time when re-entering cable will be recognized is 10 seconds, after this time system will be restarted and continue working.

It is not needed to re-enter the power cable during the LPR mode to restart the System. User can just press the F key and system will wake-up and run on the battery power. On this case (due to specific interrupt-based F key implementation) the wake-up will be immediate.

The Enhanced 'F' Key Behaviors

The '**F**' Key is a very critical for the User Interface. If pressed initiates the system FSSD (File Safe Shut Down) and if pressed again initiate the system restart from the LPR mode. Here below are presented its features analyzed according to the Powering Mode and Power Backup Source.

PIco HV4.0 Model/Assembly	Backup Source	Powering Source	Powering Mode	F Key Action	LEDs	System Behaviors
UPS Pico HV4.0 BC Stack UPS Pico HV4.0 B Stack UPS Pico HV4.0 D Stack	Battery (Any chemistry battery)	USB on Raspberry Pi®	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® RED LED is ON	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC Stack UPS Pico HV4.0 B Stack UPS Pico HV4.0 D Stack	Battery (Any chemistry battery)	USB on Raspberry Pi®	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® RED/Green LED is ON	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)
UPS Pico HV4.0 BC Stack UPS Pico HV4.0 B Stack	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	USB on Raspberry Pi®	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® RED LED is ON	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC Stack UPS Pico HV4.0 B Stack	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	USB on Raspberry Pi®	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® RED/Green LED is ON	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced UPS Pico HV4.0 D Advanced	Battery (Any chemistry battery)	USB on Raspberry Pi®	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® RED LED is ON	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced UPS Pico HV4.0 D Advanced	Battery (Any chemistry battery)	USB on Raspberry Pi®	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® RED/Green LED is ON	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	USB on Raspberry Pi®	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® RED LED is ON	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced	Super Capacitor 100F or Super Capacitors Bank (300F/500F/	USB on Raspberry Pi®	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® RED/Green LED is	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)

	800F)				ON	
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced UPS Pico HV4.0 D Advanced	Battery (Any chemistry battery)	EPR on UPS PIco HV4.0	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® LEDs OFF	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced UPS Pico HV4.0 D Advanced	Battery (Any chemistry battery)	EPR on UPS PIco HV4.0	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® LEDs OFF	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	EPR on UPS PIco HV4.0	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® LEDs OFF	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC Advanced UPS Pico HV4.0 B Advanced	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	EPR on UPS PIco HV4.0	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® LEDs OFF	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)
UPS Pico HV4.0 BC PPoE UPS Pico HV4.0 B PPoE UPS Pico HV4.0 D PPoE	Battery (Any chemistry battery)	USB on Raspberry Pi®	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® RED LED is ON	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC PPoE UPS Pico HV4.0 B PPoE UPS Pico HV4.0 D PPoE	Battery (Any chemistry battery)	USB on Raspberry Pi®	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® RED/Green LED is ON	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)
UPS Pico HV4.0 BC PPoE UPS Pico HV4.0 B PPoE	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	USB on Raspberry Pi®	CBL_MODE	Pressed	All UPS Pico HV4.0 LEDs are OFF after FSSD Raspberry Pi® RED LED is ON	After F pressing, System is starting FSSD, then after 30 seconds, entering LPR Mode. The Raspberry Pi® is in low powering Mode by PE and RUN pins. This functionality allows to shut down the system and keep OFF however with cable connected
UPS Pico HV4.0 BC PPoE UPS Pico HV4.0 B PPoE	Super Capacitor 100F or Super Capacitors Bank (300F/500F/ 800F)	USB on Raspberry Pi®	CBL_MODE	Pressed again	Proper UPS Pico HV4.0 LEDs are ON after wake-up Raspberry Pi® RED/Green LED is ON	After F pressing again on LPR Mode, System will wake up, then after some seconds, entering CBL Mode. System will be available for normal using This functionality allows to shut down and wake up the system with USB Cable Connected (like ON/OFF)

FAQ

Question: What is the usability of the F Key for my day-to-day UPS PIco HV4.0 usage?

Answer: You use the F Key as intelligent ON/OFF the system independent from the Powering Source or Power Backup. You can i.e., Switch ON/OFF system supplied by PPoE, or Solar Cells without care about cable disconnection.

System Information – SysInfo

System information information's about running/booting/shutting down of the **UPS PIco HV4.0 HAT** is important for Application Developer. They are stored on the Pico Register Called **SysInfo** located at address **0x69@0x22**. It is 16 bits wide, and there are bitwise allocated to different meaning.

0x22	SysInfo	word	Mirror	Read	Read the System Information
					Read: 0x---X bits 3:0 Means System FSSD Reason: ○ 0x1 - FSSD button ○ 0x2 - low battery ○ 0x3 - Timed FSSD ○ 0x4 - Timed Simple Scheduler ○ 0x5 - Timed ETR Scheduler ○ 0x6 - Event Read: 0x--X- bits 7:4 Means System Wakeup Reason: ○ 0x1 - FSSD button ○ 0x2 – RPi Voltage Applied ○ 0x3 - Running RPI (reset/reboot) ○ 0x4 – EPR/PPoE Voltage Applied ○ 0x5 - Timed Simple Scheduler ○ 0x6 - Timed ETR Scheduler ○ 0x7 - Event Read: 0x-X-- bits 11:8 Means Pico Restart Reason: **TBD** Write: 0x0000 – Clearing the variable

EXAMPES OF USE:

sudo i2cget -y 1 0x69 0x22 w **to get stored data**

sudo i2cset -y 1 0x69 0x22 0x0000 w **to clear stored data after read**

"PIco is Running" Feature

Many users are using the Raspberry Pi® in remote places where is difficult to access the UPS LED and see it blinking. Therefore, it is needed to check and confirm that **UPS PIco HV4.0 HAT** is running and protecting the Raspberry Pi®. For that reason, a dedicated PIco register has been implemented that allows remote user to proof that PIco is working properly. This register is placed on the I^2C address 0x69 @ 0x04. If the **UPS PIco HV4.0 HAT** is working (properly) this register value is updated every 1 millisecond. To proof that UPS PIco HV3.0A/B/B+ HAT is working need to read 2 times with time difference bigger than 1 millisecond. The read values need to be different.

EXAMPES OF USE:

sudo i2cget -y 1 0x69 0x04 w && i2cget -y 1 0x69 0x04 w

User should receive response like this (two different 16-bit numbers)

0x823f

0x8247

UPS PIco HV4.0 HAT STill Alive (STA) Functionality

The **UPS PIco HV4.0 HAT**, offers to the user a protection mechanism for the possibility of the Raspberry Hang-up (freeze of it). In a case that Raspberry Pi® freeze, the **UPS PIco HV4.0 HAT**, will automatically hardware reset it, using Gold Plated Reset Pin (POGO Pin) that must be soldered to have such functionality. The default state is that Still Alive functionality is disabled.

The **Still Alive** functionality is based on 8-bit timer located at address **0x6b @ 0x05** that his value is decreasing every second when his value is different from 0xff. If it reaches 0x00 **UPS PIco HV4.0 HAT** resets hardware the Raspberry Pi®. The default value after restart/start of the **UPS PIco HV4.0 HAT** is 0xff (disabled).

To activate it, user need to write to this register value different than 0xff, and rewrite new value every defined time (by its written value).

The following options are available, and can be used at any time:

0x05	STA_timer	Byte	Mirror	Read	Information about STA timer
					Write: 0xff – Disable STA Timer Write: 0x00 – Cause immediate Reset of the system Write: 0x01-0xfe – Start counting for the system Reset, by one second each time Read: Current Value at the time

Writing of 0xff cause disable of this STA timer

Writing of 0x01 – 0xfe cause start of down counting (every second) of this STA timer until it reaches the 0x00 when the Raspberry Pi® will be hardware Reset

Writing of 0x00 cause immediate and unconditional Raspberry Pi® hardware Reset

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x05 0x00 **unconditional resets the Raspberry Pi®**

sudo i2cset -y 1 0x6b 0x05 0x0f **Sets the STA timer to 15 seconds (if within 15 seconds software running on Raspberry Pi® not write new values, Raspberry Pi® will be hardware reset by PIco.**

sudo i2cset -y 1 0x6b 0x05 0xff **disable the STA timer**

UPS PIco HV4.0 HAT System_on_Hold Functionality

Sometimes, dedicated hardware used with Raspberry Pi® based System require the Raspberry Pi® to start with delay, after dedicated hardware starts. This feature has been implemented with the **UPS PIco HV4.0 HAT**, offers to the user this feature called **System_on_Hold**. This keep the system on hold (with PE POGO Pin) however the 5V is delivered and allows to start other hardware before the Raspberry Pi®. After dedicated time, Raspberry Pi is released and startup.

The **System_on_Hold** functionality is based on 8-bit timer located at address **0x6b@0x03** that his value is decreasing every second, when his value is different from 0x00. If it reaches 0x00 **UPS PIco HV4.0 HAT** starts hardware the Raspberry Pi® (enable Powering of it with PE). The default value after restart/start of the **UPS PIco HV4.0 HAT** is 0x00 (disabled – starts immediately).

The following options are available, and can be used at any time:

0x03	System_on_Hold	Byte	Mirror	Read	Information about System on Hold Timer
					Write: 0x00 – Disable System_on_Hold Timer Write: 0x00 – Cause immediate start of the system Write: 0x01-0xff – Start counting for the system start, by one second each time Read: Current Value at the time

Writing of 0x00 cause immediate start of the Raspberry Pi®

Writing of 0x01 – 0xff cause start of down counting (every second) of this **System_on_Hold** timer until it reaches the 0x00 when the Raspberry Pi® will start

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x03 0x0f **Sets the System_on_Hold timer to 15 seconds (Raspberry Pi® will not start for the 15 second after power applying)**

sudo i2cset -y 1 0x6b 0x03 0x00 **disable the System_on_Hold timer**

User Selectable UPS PIco HV4.0 HAT I²C addresses

The **UPS PIco HV4.0 HAT** interacts with Raspberry Pi® via **PICo I²C** Registers Set interface. There are pre-selected (default) addresses that are used by **UPS PIco HV4.0 HAT**. However, user may need to change them to adopt the system to different address area in their application. In addition, the integrated Hardware RTC may be not used and the address of the 0x68 that is assigned to it, will be used by another external RTC provided by user. The **UPS PIco HV4.0 HAT** offers a mechanism that allows to change these addresses to different ones.

The following **4 different** users selectable I²C addresses are available:

- **DEFAULT:** 0x68, 0x69, 0x6A, 0x6B, 0x6C, 0x6D, 0x6E, 0x6F
- **NO_RTC:** 0x69, 0x6B
- **ALTERNATE1:** 0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F
- **ALTERNATE2:** 0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F

Addresses DEFAULT	Addresses NO_RTC	Addresses ALTERNATE1	Addresses ALTERNATE2	Address Usage
0x68	not used	0x58	0x48	Used for UPS PIco HV4.0 Hardware RTC. If RTC is activated in the Raspberry Pi will be visible as UU
0x69	0x69	0x59	0x49	Used for system monitoring
0x6A	not used	0x5A	0x4A	Contains RTC registered accessible independently by user
0x6B	0x6B	0x5B	0x4B	Used for System Setting up
0x6C	not used	0x5C	0x4C	Used for the RTC Scheduler
0x6D	not used	0x5D	0x4D	Used for the RTC Scheduler
0x6E	not used	0x5E	0x4E	Used for the RTC Scheduler
0x6F	not used	0x5F	0x4F	Used for the RTC Scheduler

Programming/Changing of the I²C addresses is possible via writing to the **PICo I²C** Registers Set **0x6b @0x00** the following values:

- For **DEFAULT** write the value 0x60
- For **NO_RTC** write the value 0x68
- For **ALTERNATE1** write the value 0x50
- For **ALTERNATE2** write the value 0x40

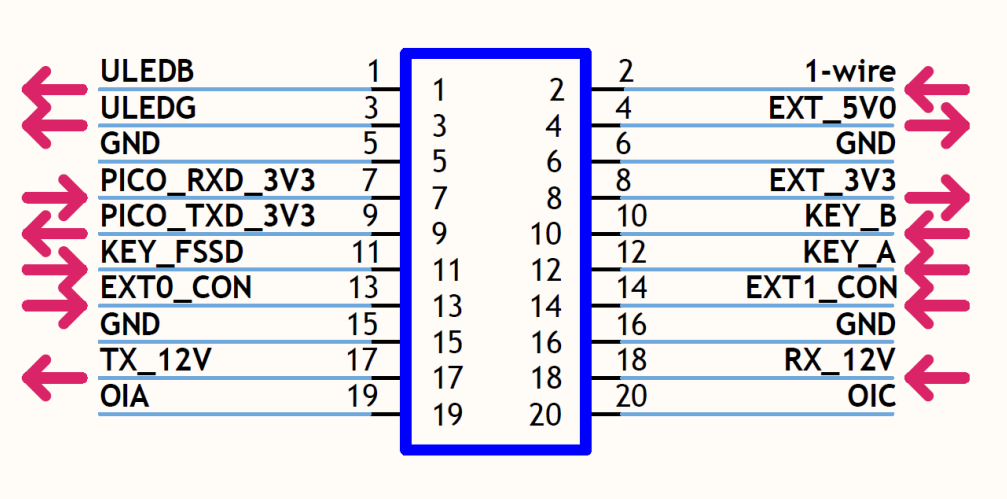
UPS PIco HV4.0 HAT User Applications Hardware Interfaces

The **UPS PIco HV4.0 HAT** is equipped with a set of User Applications Hardware Interfaces that allows to rapid setting-up of various applications without necessity of other additional HAT HATs (PCBs). It contains:

- 2mm Header Hardware Interfaces
- System and User LEDs
- System and User Buttons
- Sound Generation System
- Single SPDT Relay
- ON/OFF Magic Switch
- Opto-Coupler interface
- Programmable ON/OFF Auxiliary 5V@150mA mA and 3.3V@150 mA Powering Sources
- RS232 Interface dedicated to wakeup/shutdown on command
- Multiple/Multiplexed RS232 Interfaces Support (with +/-12V driver)
- User Selectable PIco HV4.0 I^2C addresses (NORMAL, NO_RTC, ALTERNATE1, ALTERNATE2)
- IR Receiver Interface
- ESD protected 1-wire Interface

2mm Header Hardware Interfaces

All the hardware interfaces that must be accessed externally are routed to the 2mm Header. These interfaces can be accessed by simple soldering cable or by using the Terminal Block additional PCB. The Terminal Block PCB offers Screwed interface to each pin of the 2mm header. They are:



Name	Pin Number	Label on PCB	Meaning	Usage
ULEDB	1	LR	User LED Blue	
ULEDG	3	LG	User LED Green	
GND	5	GN	Ground	
PICO_RXD_3V3	7	RX	PICO RXD @3V3	
PICO_TXT_3V3	9	TX	PICO TXD @3V3	
KEY_FSSD	11	KF	Key FSSD	
EXT0_CON	13	E0	A/D Converter Channel 0	
TX_12V	17	T2	PICO TXD @12V	
OIA	19	OA	Opto-Coupler Anode Input	
1-wire	2	1W	1-wire interface	
EXT_5V0	4	5V	Supply 5V0	
EXT_3V3	8	3V	Supply 3V3	
KEY_B	10	KB	Key B	
KEY_A	12	KA	Key A	
EXT1_CON	14	E1	A/D Converter Channel 1	
RX_12V	18	R2	PICO RXD @12V	
OIC	20	OC	Opto-Coupler Cathode Input	

UPS PIco HV4.0 HAT LEDs

The **UPS PIco HV4.0 HAT** is equipped with 10 LEDs (that provides information about the **UPS PIco HV4.0 HAT** system status and user information). There are 7 System LEDs and 2 User Application LEDs. The System LEDs are described here on below table:

System LEDs Indications		
SYS LED BLUE		
	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 750 ms for 100 ms	System (PIco + RPi) is running on cable powering (after booting time)
	Blinking every 2000 ms for 100 ms	System (PIco + RPi) is running on battery powering
BAT LED RED		
	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 BLUE		
	OFF	Battery is not Charged (full)
	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, then CHG LED is OFF
FAN LED BLUE		
	OFF	FAN is not running
	Lighting continuously	FAN is running
EXT LED BLUE		
	OFF	External Cable powering or PPoE is disconnected (6.5-32VDC)
	Lighting continuously	External Cable powering or PPoE is

		connected (6.5-32VDC)
TMR LED BLUE		
	OFF	Timer of the Scheduler is OFF
	Blinking every 1000 ms for 10 ms	Timer of the Scheduler is Active
SCA LED BLUE		
	OFF	Supercapacitor level is below 2V5
	ON	Supercapacitor level is above 2V5
TMP LED RED		
	OFF	Raspberry Pi core temperature is below programmed threshold (default is 50 Celsius)
	ON	Raspberry Pi core temperature is above programmed threshold (default is set to 50 Celsius)

The **User LEDs** are dedicated for user applications and can be handled by the **PICo I²C** Registers Set interface. One of them is Green, and the second is Blue.

In the **UPS PIco HV4.0 HAT** there is an additional option to use external LEDs connected in parallel with the User LEDs. Thus, allows user to make their own applications where Pico User LEDs will be used. This option is used in the **PiBlock Case** offered by our company.

Connectivity of external User LEDs can be done via cables using the 2mm header as shown here below. It is not needed to use in series a resistor however in any case the Pico LED resistor is used in this connectivity.

Activating of the User LEDs can be done by the following PICo Commands.

0x09	User LED Green	Byte	Common	R/W	User LED Green ON - Write: 0x01 User LED Green OFF - Write: 0x00
0x0a	User LED Blue	Byte	Common	R/W	User LED Blue ON - Write: 0x01 User LED Blue OFF - Write: 0x00

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x09 0x01 **for ON the Green LED**

sudo i2cset -y 1 0x6b 0x09 0x00 **for OFF the Green LED**

sudo i2cset -y 1 0x6b 0x0a 0x01 **for ON the Blue LED**

sudo i2cset -y 1 0x6b 0x0a 0x00 **for OFF the Blue LED**

UPS PIco HV4.0 HAT System-Users LEDs Mapping

The **UPS PIco HV4.0 HAT** is equipped with User LEDs as described above that can be used for any user information. However, in the models there are ready soldering pads that allow to solder external (via cables) LEDs and use them as an external indicator. Sometimes it is usefully to map to this User LEDs system functionalities LEDs and have some or most of them available for viewing if system based on **UPS PIco HV4.0 HAT** and Raspberry Pi® is placed in closed cases. The LEDs Mapping allows to "map" (copy) selected system LEDs indications to any User LED. The mapping does not change the "normal" system LEDs indication as it is defined.

0x18	Mapping_User_LED_Green	Byte	Common	R/W	There is single byte register 0b76543210 where the corresponding bits when written with "1" are: • Bit 0th is mapping SYS LED • Bit 1st is mapping BAT LED • Bit 2nd is mapping CHG LED • Bit 3rd is mapping TMP LED • Bit 4th is mapping SCA LED • Bit 5th is mapping REL LED • Bit 6th is mapping TMR LED • Bit 7th is mapping PWR LED
0x19	Mapping_User_LED_Blue	Byte	Common	R/W	There is single byte register 0b76543210 where the corresponding bits when written with "1" are: • Bit 0th is mapping SYS LED • Bit 1st is mapping BAT LED • Bit 2nd is mapping CHG LED • Bit 3rd is mapping TMP LED • Bit 4th is mapping SCA LED • Bit 5th is mapping REL LED • Bit 6th is mapping TMR LED • Bit 7th is mapping CBL LED

Mapped User LEDs can be used with their command for ON/OFF however when updated status if the "System" the value of it will be overwritten. It is possible also to map 2 or more system LEDs to a single User LED, however it makes reading more complicated.

Example of use

sudo i2cset -y 1 0x6b 0x18 0x01 (0b00000001) Mapping of System SYS LED on the User Green LED

sudo i2cset -y 1 0x6b 0x19 0x02 (0b00000010) Mapping of System BAT LED on the User Blue LED

UPS PIco HV4.0 HAT System-Users LEDs ON/OFF

The **UPS PIco HV4.0 HAT** can turn most of the LEDs OFF, except of these ones connected directly to hardware, by setting the proper register. LEDs that cannot switch OFF are:

- EPR
- FAN
- CHG

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x15 0x01 **for ON the LEDs**

sudo i2cset -y 1 0x6b 0x15 0x00 **for OFF the LEDs**

UPS PIco HV4.0 HAT Buttons

The **UPS PIco HV4.0 HAT** is equipped with 5 buttons that can be used in various ways. Two of them are dedicated for user applications and can be handled by user through the **PICo I²C** Registers Set, all other are specific for various **UPS PIco HV4.0 HAT** functionalities. All of them can be used for some start-up functionalities when **UPS PIco HV4.0 HAT** is reset. Access to vale of User Keys is available via **0x69@0x18** register. This is the only Register that need to be write after reading. Once User Key is pressed, its value stays in this register until read.

To make working external keys (buttons), user need to solder cables to the appropriate 2mm Header pads. It is not recommended to use a very long cable (due to analog implementation of the keyboard), their length should not be longer than 100 – 200 mm. To make external keys workable, user need to short each one with GND 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 GND pad when pressed. Similar approach should be followed with other keys.

The key register holds the latest value of pressed key, so user need to write 0x00 after reading, to recognize that new key has been pressed (when pressed again, even the same key).

A detailed description of all buttons and their usage is provided on below table.

Button	Description	Usage	Additional Functionalities
RR	Raspberry Pi® Hardware Reset	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)	
UR	UPS Pico HV4.0 HAT Hardware Reset	Make the UPS Pico HV4.0 HAT Hardware Reset when pressed. **NOTE1:** Resetting of the UPS Pico HV4.0 HAT does not reset the Raspberry Pi® only if USB cable powered. **NOTE2:** Resetting of UPS Pico HV4.0 HAT does NOT reset the Integrated Hardware RTC.	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 UR button and then release the other button (A, B).
F	File Safe Shut Down (FSSD)	When pressed initiate the File Safe Shutdown Procedure. • If Raspberry Pi® and UPS Pico HV4.0 HAT system is battery powered, after FSSD finished UPS Pico HV3.0 HAT will cut the power, and only RTC is running • If Raspberry Pi® and UPS Pico HV4.0 HAT system is USB cable powered, after FSSD finished the UPS Pico HV4.0 HAT will disable the Raspberry Pi PE the power supply, and only RTC is running • If Raspberry Pi® and UPS Pico HV4.0 HAT system is EPR/PPoE cable powered, after FSSD finished the UPS Pico HV4.0 HAT will cut the power, and only RTC is running Pressed again (need to have installed the Gold-Plated Reset Pin for the restart option), start the Raspberry Pi® + UPS Pico HV4.0 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 Ready Value: 0x01	Can be used for User Application – Read the status via PICo I^2C Registers Set or RS232 interface	NONE
B	User Key B Ready Value: 0x02	Can be used for User Application – Read the status via PICo I^2C Registers Set or RS232 interface	When used with UR button, Set the UPS Pico HV4.0 HAT to default values
No Key Pressed	Ready Value: 0x00	NONE	NONE

EXAMPES OF USE:

sudo i2cget -y 1 0x69 0x18 **for user keys pressed read**

should return 0x00, 0x01 or 0x02

sudo i2cset -y 1 0x69 0x18 0x00 **for reset keys pressed after read**

IMPORTANT NOTICE: Pressed User Key Value will remain until read by user and reset its value.

UPS PIco HV4.0 HAT Sound Generation System

The **UPS PIco HV4.0 HAT** is equipped with Enhanced Sound Generation System. It is providing a user audio interface on various states of **UPS PIco HV4.0 HAT** conditions, but it is also available for dedicated user applications offering the whole range of acoustic frequencies full programmable by user.

There are 3 registers that are responsible for the generating sound **bmode, bfreq** and **bdur** located at **0x6B**. To generate sound user, need to program first the required frequency and then the required duration is 10th of ms.

Current implementations need to program one be one sound when generated. The maximum duration is 255 x 10 ms = 2.55 seconds

Additionally, it is possible to deactivate it permanently, by setting the **bmode** register to 0x00.

The default value is active

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 Default Value: 0x01
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)

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x0d 0x00 **Deactivate permanently the buzzer (no sounds will be played)**

sudo i2cset -y 1 0x6b 0x0d 0x01 **Activate permanently the buzzer (default value)**

In order to play sound buzzer, need to be activated firstly.

sudo i2cset -y 1 0x6b 0x0e 0x417 w **Set the frequency to C (1047 Hz) note**

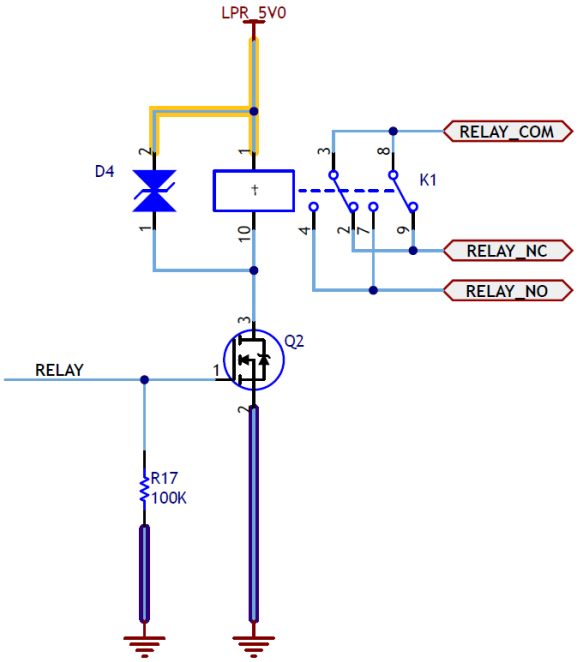
Sound will be not generated until **bdur** will be set, as the **bfreq** just set the frequency. Therefore, the activation and duration of sound is done via register **bdur**.

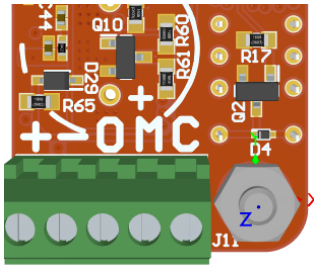
sudo i2cset -y 1 0x6b 0x10 100 **Set the duration to 1 second**

After Sound execution, the **bdur** register is 0 again.

UPS PIco HV4.0 HAT SPST Relay

The **UPS PIco HV4.0 HAT** can be equipped with embedded Relay with single coil. This Relay is standard offered with version **UPS PIco HV4.0 HAT Advanced/PPoE**, it can be also ordered separately and added to the **UPS PIco HV4.0 HAT Stack**. In both cases this Relays is not mounted on the PCB and user need to do it by himself. The assembly (soldering) of the Relay on the **UPS PIco HV4.0 HAT** it is very easy task and can be done by anybody using simple soldering tool. However, it is also possible that this assembling can be ordered to be done by our company, if customer order directly on the shop or any other eshop that offer such service. The Relay is DPDT however in order to have higher current immunity pins have been connected in parallel, therefore it is acting as SPDT. The Relay is handled from the **PICo I²C** Registers Set **0x6b@0x0B**, by writing 0x01 to set or 0x00 to reset.





Relay Contact description	Label on PCB	Meaning
Normally Close Contact	Close	On Set State this contact is closed and have connection with Common. Relay NC (Normal Close)
Common Contact	coMmon	On Set States this contact is switching between NCO/NOO
Normally Open Contact	Open	On Set State this contact is Open and does not have connection with Common. Relay NO (Normal Open)

Relay Basic Technical Specifications

Arrangement	2 form C
Contacts Material	Gold overlay silver alloy
Contacts Resistance (initial)	Maximum 50 mΩ (at 1 A 6 VDC)
Contacts Rating (resistive)	0.5A 125 VAC or 2A 30 VDC
Contacts Maximum Carrying Current	2 A
Contacts Maximum Switching Power	62.5 AV/30 W
Contacts Maximum Switching Voltage	250 VAC, 220 VDC
Contacts Operate (at nominal voltage)	Maximum 3 ms
Contacts Release (at nominal voltage)	Maximum 3 ms

Due to construction of **UPS PIco HV4.0 HAT** PCB we do not allow to use Integrated Relay for switching of higher voltages/currents other than 32 VDC/2A per switching contacts.

Due to PCB construction, it is not allowed to use the Integrated Relay for switching

220 VAC at any current, even very low.

EXAMPES OF USE:

sudo i2cset -y 1 0x6B 0x0B 0x00 **should Reset the Relay**

sudo i2cset -y 1 0x6B 0x0B 0x01 **should Set the Relay**

Each time when Relay is changing his state a characteristic “tick” is audible. Multiple execution of the same command is not changing anything.

0x0B	relay	Byte	Common	R/W	Action on Relay Write: 0x01 Set Write: 0x00 Reset

A very usefully and nice usage of User LEDs mapping functionality is to set User LED to be active when Relay is active using below command:

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x18 0x20

With this command each time when Relay is activated a User LED will lit, so user will know about Relay just looking to the User LED.

UPS PIco HV4.0 HAT Programmable Auxiliary 5V@150 mA and 3.3V@150 mA Powering Sources

The **UPS PIco HV4.0 HAT** is equipped with Auxiliary 5V@150mA and 3.3V@150mA Power Sources. The 3.3V@150mA is produced with LDO that are independent from the 5V of the Raspberry Pi®. There are programmable and battery backed up (if programmed/activated by user), provide continuously supply even if Raspberry Pi® is switched OFF. The Auxiliary 5V@150mA is reverse current draw with Schottky diode. Therefore, due to small voltage drop the final voltage is about 4.85V, instead of the 5.0V. The 3.3V@150mA is only protected with LDO itself embedded over current protection. These Auxiliary Power Sources are addressed to supply devices that need to be running even if Raspberry Pi® is switched OFF i.e. USB HUB, PIR Sensor, additional external high current relay, addon PCBs with extra hardware etc.

Both Auxiliary Powering Sources are controlled with the same PICo Register set **0x6b@0x06**, by writing 0x01 to activate during LPR or 0x00 to de-activate during LPR.

0x06	enable5V	Byte	Common	R/W	Defines usage of the Auxiliary 5V@150mA

					0x00 – Auxiliary 5V and 3.3V are not battery backed 0x01 – Auxiliary 5V and 3.3V are battery backed Default Values is OFF

EXAMPES OF USE:

sudo i2cset -y 1 0x6B 0x06 0x00

The Auxiliary 5V0 and 3V3 will be not battery backed-up and stop working when power will be cut-off on the GPIO 5V, therefore will be present until Raspberry Pi® is powered by cable power or battery. When system, shutdown and **UPS PIco HV4.0 HAT** enter to Lower Power Mode these Auxiliary Powering Sources will cut out.

sudo i2cset -y 1 0x6B 0x06 0x01

The Auxiliary 5V and 3.3V will be battery backed-up and will continue supply also when 5V will be not available on the GPIO 5V, therefore will be present also after Raspberry Pi® is shut down and not powered, and **UPS PIco HV4.0 HAT** enter to Lower Power Mode.

On that case the power usage on Low Powering Mode is increased by LDO and boost converter quiescent current (around of 1 mA in total) and current drawn by connected devices.

UPS PIco HV4.0 HAT IR Receiver Interface

The **UPS PIco HV4.0 HAT** is equipped with **IR** receiver interface. It is directly routed to the GPIO18. It can be used for any application like Media Player. If IR Receiver is not soldered, then the GPIO18 is free for any other application. The **UPS PIco HV4.0 HAT** is offering ONLY the interface (resistor and Powering) to the IR, and do not interact with it in any way. In order to use the IR receiver, user do not need to add anything. An excellent tutorial how to use IR is provided by www.thepihut.com at below link:

https://thepihut.com/blogs/raspberry-pi-tutorials/raspberry-pis-remotes-ir-receivers

UPS PIco HV4.0 HAT Serial Port(s)

The **UPS PIco HV4.0 HAT** support to the serial ports is very advanced and contains:

- 2x12V Multiplexed Serial Port driver
- An independent (monitored) serial Port of the **UPS PIco HV4.0 HAT** micro-controller that is used to various applications
- Multiplexed Interface to existing Serial Ports of the Raspberry Pi 4 Model B that can be connected to all supported by Raspberry Pi 4 Model B Serial Prts
- Programmable rate @command line interface over selected serial port(s)
- Shutdown and wakeup on serial data
- Send dedicated serial data on wakeup

Both Serial Ports Rate can be programmed. This can be done with a single command written to the PICo Command register **RS232_rate** at **0x6b@02**. Only 4 bits are assigned to each Serial Port Rate definition:

7th bit	**6th bit**	**5th bit**	**4th bit**	**3rd bit**	**2nd bit**	**1st bit**	**0th bit**
Dedicated to Pico Serial Port				**Dedicated to Raspberry Pi Serial Port**			
Port is OFF (HiZ)		0		Port is OFF (HiZ)		0	
Port is 4800 BPS		1		Port is 4800 BPS		1	
Port is 9600 BPS		2		Port is 9600 BPS		2	
Port is 19200 BPS		3		Port is 19200 BPS		3	
Port is 38400 BPS		4		Port is 38400 BPS		4	
Port is 57600 BPS		5		Port is 57600 BPS		5	
Port is 115200 BPS		F		Port is 115200 BPS		F	

EXAMPES OF USE:

sudo i2cset -y 1 0x6B 0x02 0x00 Sets both Serial Ports OFF

sudo i2cset -y 1 0x6B 0x02 0x0F Sets Pico Serial Port OFF, Raspberry Pi Port to 115200 bps

sudo i2cset -y 1 0x6B 0x02 0xFF Sets both Serial Ports to 115200 bps

UPS PIco HV4.0 HAT Serial Port(s) Multiplexer

UPS PIco HV4.0 HAT Serial Port(s) Router

UPS PIco HV4.0 HAT Serial Port(s) @commands

```
sudo apt-get install device-tree-compiler
```

sudo apt-get install mc

UPS PIco HV4.0 HAT FAN Control (Active Cooling System)

The **UPS PIco HV4.0 HAT** can be equipped with Active Cooling System based on micro-FAN. The PIco FAN is full low rate PWM controlled rotation speed from 0% up to 100%. It can be manually set ON or OFF on per-selected speed, as also automatically based on preset temperature threshold. It can be done via the following registers placed at the **PICo I²C** Registers Set address **0x6b @0x11**, **@0x12**, and **@0x13**.

When UPS PIco is going down to the LPR mode or running with Supercapacitor power backup, the FAN is automatically disabled, and enabled again when the UPS PIco returns to normal work.

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 **Write:** 0x02 – Automatic ON/OFF with defined speed in the fspeed, ON when temperature read directly from Raspberry Pi s higher than fttemp threshold, OFF when lower. **Default Value** is set to 0x02 – Automatic ON/OFF
0x12	**fspeed**	**Byte**	**Common**	**R/W**	**Integrated Fan Speed** **Read:** Anytime, Return actual fspeed value **Write:** 00 – Selected speed when OFF 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 Default speed is set to 50%
0x13	**fttemp**	**Byte**	**Common**	**R/W**	**Integrated Fan Temperature Threshold in Automatic Mode** BCD Fan Running threshold temperature in Celsius, 2 digits i.e. 35, means 35 Celsius. In order to be used (automatic FAN ON/OFF) need to set fmode to 0x02. **Write:** Anytime, with threshold value in BCD format **Read:** Anytime, Return actual fspeed value

					Default value is set to 50 Celsius

There are also available 2 Registers in **PICo I²C** Registers Set address **0x69 @0x1A**, **@0x21**. Both can be read. The first one **@0x1A** contains the Raspberry Pi® Core temperature and is used to adjust FAN activation or FAN Speed. The second one **@0x21** when read show if FAN is running or not. In addition, the LED FAN lit, if FAN is running.

EXAMPES OF USE:

sudo i2cset -y 1 0x6b 0x13 100 **Set the FAN speed to 100**

sudo i2cset -y 1 0x6b 0x12 0x01 **Set the FAN ON**

sudo i2cset -y 1 0x6b 0x12 0x00 **Set the FAN OFF**

sudo i2cset -y 1 0x6b 0x12 0x02 **Set the FAN ON as an Automatic**

The default setup is Automatic Mode with 50 Celsius and 50% of FAN speed, so user do not need to change anything if like just to use the FAN. If higher cooling performance is needed (however with more noise) then the fspeed should be set to 100 (100%), similar with temperature threshold fttemp. However please kindly notice that FAN speed and temperature threshold have been set in order to have best performance with lowest noise.

UPS PIco HV4.0 HAT Measuring and Monitoring System

The **UPS PIco HV4.0 HAT** offer to the user an extended Measuring and Monitoring System that measure and report many systems parameters trough installed sensors. Each sensor is reporting the **UPS PIco HV4.0 HAT** status via dedicated variables (PICo Registers). In addition, there is access to the integrated 12 bits 2x Voltage Follower buffered A/D converters. All monitoring system data are collected in a single entity called **PICo Status** and exists at the **PICo I²C** Registers Set address **0x69**. Detailed specifications for each variable (register) as also examples are provided in next pages. They are:

List them

Powering Mode

This Register (called also Variable) contains information about current powering source. They are:

- 0x00 Cable Powering RPI, EPR, PPoE
- 0x01 Battery (Supercapacitor) Powering BAT, SCA

It is changed automatically when powering source change, and can be used for by user to fire up various application i.e. data backup if system is battery powered. If buzzer is assembled and activated (default) then an audible signal is informed the user when powering source is changed.

Backing Mode

This Register (called also Variable) contains information about current powering source if Mixed Mode is used. In the Mixed Mode (Supercapacitor and Battery) system on short cable power losses is backed up (powered) via Supercapacitor and if the level of it is not enough or cable power lose is too long backed up via Battery. This temporary status of powering source is stored in the Register to inform user about it.

Running Time

Not Activated YET

PIco is Running

Supercapacitor Level

Battery Level

Raspberry Pi® 5V0 GPIO Level

EPR (External Powering) or PPoE Level

Incoming Current Level

Not Activated YET

The **UPS PIco HV4.0 HAT** has implemented Single and Dual High-Side Current-Sense Monitor with Power Calculation Integrated Circuit. This IC is a single or dual high-side bidirectional current sensing monitors with precision voltage measurement capabilities. Each sensor measures the voltage developed across an embedded sense resistor to represent the high-side current of a battery or voltage regulator. The implemented IC also measures the SENSE+ pin voltage and calculates average power over the integration period. The implemented IC is also measuring dynamic power. The long integration time allows for extending system polling cycles without losing any power consumption information. It measures the SENSE1+ and SENSE2+ pin voltages (VSOURCE) and calculates average power over the integration period.

Outcoming Current Level

Not Activated YET

The **UPS PIco HV4.0 HAT** has implemented Single and Dual High-Side Current-Sense Monitor with Power Calculation Integrated Circuit. This IC is a single or dual high-side bidirectional current sensing monitors with precision voltage measurement capabilities. Each sensor measures the voltage developed across an embedded sense resistor to represent the high-side current of a battery or voltage regulator. The implemented IC also measures the SENSE+ pin voltage and calculates average power over the integration period. The implemented IC is also measuring dynamic power. The long integration time allows for extending system polling cycles without losing any power consumption information. It measures the SENSE1+ and SENSE2+ pin voltages (VSOURCE) and calculates average power over the integration period.

External Powering or PPoE Incoming Current Level

Not Activated YET

The **UPS PIco HV4.0 HAT** has implemented Single and Dual High-Side Current-Sense Monitor with Power Calculation Integrated Circuit. This IC is a single or dual high-side bidirectional current sensing

monitors with precision voltage measurement capabilities. Each sensor measures the voltage developed across an embedded sense resistor to represent the high-side current of a battery or voltage regulator. The implemented IC also measures the SENSE+ pin voltage and calculates average power over the integration period. The implemented IC is also measuring dynamic power. The long integration time allows for extending system polling cycles without losing any power consumption information. It measures the SENSE1+ and SENSE2+ pin voltages (VSOURCE) and calculates average power over the integration period.

Buffered 12-bit A/D converters

The **UPS PIco HV4.0 HAT** is equipped with 2 x 12 bits multichannel A/D converter. Access to their conversion data is possible via dedicated PICo Registers placed at the **PICo I²C** Registers Set address **0x69 @0x14** and **@0x16** separately for each channel. Those A/D converters samples continuously A/D channels every 400 uS with conversion time of 3uS per sample. However due to implemented low noise software enhanced filtering in the firmware the effective rate data rate is 0.001 sec per reading.

Each of the A/D converters is ESD protected for 5V0. They are also buffered with **Voltage Follower**. They are named aEXT0, aEXT1. Read data are rounded and converted to BCD format in order to allow user easy reading from command line. The maximum reading voltage is 3V3.

EXAMPES OF USE:

sudo i2cget -y 1 0x69 0x14 w **should return value of the aEXT0level in BCD format**

sudo i2cget -y 1 0x69 0x16 w **should return value of the aEXT1level in BCD format**

User Key Pressed

UPS PIco HV4.0 HAT PCB Temperature

Not Activated YET

Raspberry Pi® Core Temperature

Opto-Coupler Level (status)

Not Activated YET

Embedded Charger Programmed Charging Current and Status

Embedded Charger Real Charging Current

Not Activated YET

FAN PWM Status

SYSINFO Variable

PCB Version

UPS PIco HV4.0 HAT Model

UPS PIco HV4.0 HAT PCB default battery

Firmware Version

UPS PIco HV4.0 HAT System Time Schedulers

The **UPS PIco HV4.0 HAT** has implemented 2 independent, Time Schedulers. There are:

- The **B**asic Time **S**cheduler (**BS**)
- The **E**vent **T**riggered **R**TC Based **S**ystem **A**ctions **S**cheduler (**ETR SAS**)

Both schedulers cannot be used at the same time, and if the first one is selected, the second is deselected and vice versa. The default value is selecting the **B**asic **S**cheduler. Running of Time Schedulers increasing the current consumption during the sleep (LPR) mode.

The Register responsible for the Scheduler selection is located in the **0x6b** Registers Set

0x6B -> UPS PIco **0x20** Time Scheduler Selector

This register is used to select the System Time Scheduler. Three values are possible 0x00 (default value) – which means deselected all schedulers, 0x01 – which means **Basic Scheduler** selected, or 0x02 – which means **Event Triggered RTC Based System Actions Scheduler**. Setting of it is necessary to select the proper System Time Scheduler.

sudo i2cset -y 1 0x6b 0x20 0x01 to select **BS** (default value)

or

sudo i2cset -y 1 0x6b 0x20 0x02 to select **ETR SAS**

Basic Scheduler

This scheduler is basically used when **UPS PIco** simple scheduler is needed, just to make **ON/OFF** the Raspberry Pi (so no needed to use the complex settings of the ETR SAS). There are only few registered involved in this scheduler and setting up of them is rapid. User need just to set how long Raspberry Pi® should be running, after that, how long Raspberry Pi® should be not running, and how many times it should happen. The time resolution of the **BS** is based on **1 minute**, however everything is adjusted with 1 second accuracy, as each start/stop action is executed at the beginning (first second) of internal RTC counted minute (even if the internal RTC is not set, it is always running – however some features need to have it set). Below picture explain the logic behind of this **B**asic **S**cheduler. In order rapid to use BS it is not needed to have setup the RTC, however some features need to have it set-up.

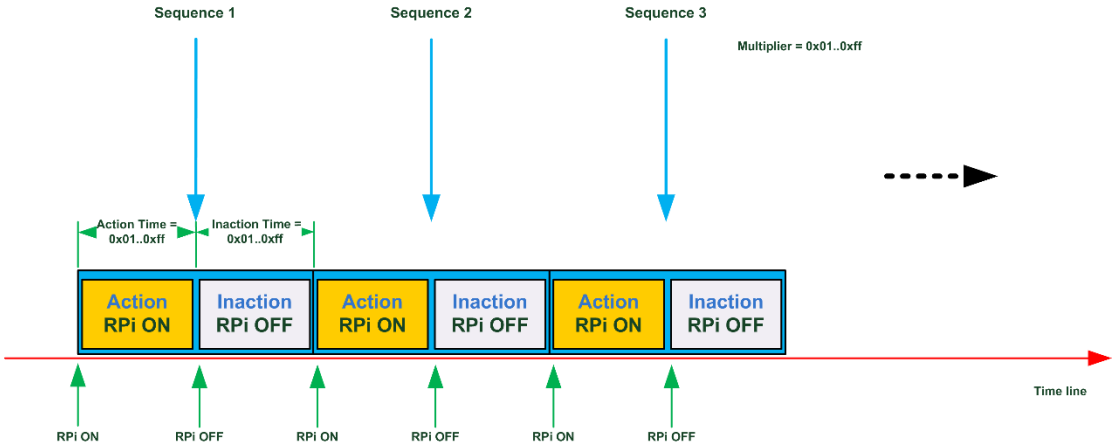


Figure 1 Single Basic Scheduler

BS Definitions

There are some definitions that need to be specified to have better understanding of the **Basic Scheduler** functionality. There are basically similar to the **ETR SAS** definitions (described in the next chapter), however are simplified due to adaptation to this simple **Basic Scheduler**. There are:

BS Action – It is **ON** the Raspberry Pi® only

BS Inaction – It is **OFF** the Raspberry Pi® only

BS Action Time – Specify time between **BS Action** (ON of the Raspberry Pi®) and their opposite state (OFF of the Raspberry Pi®). In example if Raspberry Pi® **Power ON**, the opposite state is Raspberry Pi® **Power OFF**. Therefore, **BS Action Time** is time between ON and OFF of the Raspberry Pi®. So, simply saying "how long the Raspberry Pi will be ON"

BS Inaction Time – Specify time between **BS Inaction** (OFF of the Raspberry Pi®) and their opposite state (ON of the Raspberry Pi®). In example if Raspberry Pi® **Power OFF**, the opposite state is Raspberry Pi® **Power ON**. Therefore, **BS Inaction Time** is time between OFF and ON of the Raspberry Pi®. So, simply saying "how long the Raspberry Pi will be OFF"

BS Sequence – The **BS Sequence** defines both states (**ON** and **OFF**) together as single entity. This definition is usefully for further explanation of **BS Multiplier**. User can have such sequence multiplied or run infinitely.

BS Start Time – The **BS Start Time** defines starting time in 2 BCD bytes hours and minutes. However by default this value is set to 0xFFFF, and means start in next minute (immediately) after activation with **BS_RUN** register.

BS Multiplier – Defines how many times **BS Sequences** will be repeated from the **BS Sequence**, up to 254 times or infinitely which is 255 (0xFF).

Basic Scheduler Involved PICo Registers

The following PICo Registers are involved in the **Basic Scheduler** programming. There are:

- **Time_scheduler_Selector** - placed address **0x6B** and location **0x20** (default 0x01)
- **BS_action_time** - placed address **0x6B** and location **0x21** (default 0x01 minute)
- **BS_inaction_time** - placed address **0x6B** and location **0x22** (default 0x01 minute)
- **BS Start Time** - placed address **0x6B** and location **0x24** (default 0xFFFF)
- **BS_multipier** - placed address **0x6B** and location **0x26** (default 0xFF infinitive)
- **BS_RUN** - placed address **0x6B** and location **0x27** (default 0x00 OFF)

Detailed description of all Registries related to the **Basic Scheduler** are located at the **0x6B -> UPS PIco Module Commands** and there are:

0x6B -> UPS PIco **0x21** BS_action_time (in minutes)

This register defines how the Raspberry Pi® long will be ON (called Action) and running after starting up of Basic Scheduler. The default value is 0x01. Each value is in minutes. The maximum time is 0xff (255 minutes). This register can be read when Raspberry Pi® is running, user will see the decreased value as time is passed. The value cannot be lower than 0x01.

0x6B -> UPS PIco **0x22** BS_inaction_time (in minutes)

This register defines how long the Raspberry Pi will be OFF (called inaction) and not running after starting up of Basic Scheduler. The default value is 0x01. Each value is in minutes. The maximum time is 0xff (255 minutes). This register can be read when Raspberry Pi® is running, user will see the decreased value as time is passed. The value cannot be lower than 0x01.

0x6B -> UPS PIco **0x23** BS_Start Time (4 digits in BCD)

0x6B -> UPS PIco **0x24** BS_multiplier

This register defines how many times the **Basic Scheduler Sequence** (set of one Action and one Inaction - Raspberry Pi® ON/OFF) will be running. It can make it running counted times (from 1 up to 254), or if programmed 0xff (255) then the Action will be executed unlimited times (repeated continuously). This register can be read when Raspberry Pi® is running, user will see the decreased value as counter is passed.

0x6B -> UPS PIco **0x25** BS_RUN

This register is used to make **Basic Scheduler** running (start) or not (stop). However, there are some smart variations that allows the Basic Scheduler to be used more flexible and easier. Therefore, with this register setting **BS** can start immediately, on date changing (00:00) – here required is to have set-up the RTC properly the time, as also start with Active state and then go to Inactive or the opposite, Start with Inactive state and then go to Active one.

0x20	Time_Scheduler_Selector	Byte	Common	R/W	Selects which **Scheduler** is used: - 0x01 Basic Scheduler (default) - 0x02 ETR SAS Only one can be selected, and each programming is referred to it. **Default value : 0x01**
0x21	BS_action_time	Byte	Common	R/W	**Basic Scheduler Action Time** in minutes. Allowed values are 0x01 – 0xff. **Default value : 0x01**
0x22	BS_inaction_time	Byte	Common	R/W	**Basic Scheduler Inaction Time** in minutes. Allowed values are 0x01 – 0xff. **Default value : 0x01**
0x22	BS_start_time	word	Common	R/W	**Basic Scheduler Start Time** in BCD. **Default value : 0xffff**
0x25	BS_multipier	Byte	Common	R/W	**Basic Scheduler Sequence Multiplier**. Allowed values are 0x01 – 0xff. **Default value : 0xff** Value of 0xff means running (repeating) unlimited times after starting.
0x27	BS_ RUN	Byte	Common	R/W	Specify when **Basic Scheduler** is starting and running or not. Allowed values are: - 0xff BS Starts in first defined start time with resolution to minute, with executing first the action state - 0x00 BS Stops immediately in next minute, after finishing last state **Default is 0x00 (not running). When changing the BS to 0xff (running), system automatically update the battery running time to unlimited (0xff) and if**

Table 1 Basic Scheduler involved Registers

Basic Scheduler Optical Indications

When **B**asic **S**cheduler is activated (BS_ RUN=0xFF, 0xFA, 0xFB), the **TMR LED** is blinking for 50 ms every 1 second. This happens as far the Raspberry Pi is cable powered and running. In the Low Powering Mode, the **TMR LED** as all others are OFF. This feature allow user to know when **BS** is active or not.

All programmed **BS** values are stored in the internal **EEPROM**, therefore if your system reset/restart the **BS** will continue running until **BS_ RUN=0x00.**

Activating of Basic Scheduler UPS PIco automatically changes the battery_run_time to unlimited (0xFF) to avoid collisions with programmed schedulers.

Deactivating of Basic Scheduler UPS PIco automatically changes the battery_run_time to 70 second (0x01)

Programming (setting-up) of Basic Scheduler is not allowed when system is Battery Powered.

The Basic Scheduler can be used when system is supplied only with battery

If during the Basic Scheduler execution powering condition changed (i.e. enter the powering Cable) system will behavior as without Basic Scheduler therefore will start-up.

All other functionalities related to Raspberry Pi running functionalities (i.e. STA or Low Battery) are still active when the Basic Scheduler is running

BS Example 1st - Simple Raspberry Pi® ON/OFF executed infinitive times for 1 minutes (ON/OFF every minute), starting immediately

We need to start up - set ON - the Raspberry Pi®, keep it running for 1 minutes, shutdown it, and after 1-minute start it again. This will be repeated infinitely.

Time_Scheduler_Selector = 0x01;	Select the Basic Scheduler
BS_action_time = 0x01;	Sets duration of Action (ON) time to 1 minute (Raspberry Pi® will run for 1 minute and then shutdown)
BS_inaction_time =0x01;	Sets duration of Inaction (OFF) time to 1 minute (Raspberry Pi® will be sleeping for 1 minute)
BS_multipier = 0xff;	This will be repeated forever
BS_ RUN =0xff;	When user decide, just activate the Basic Scheduler, and it is start immediately

The data entering should looks like below (it important to follow the below order to avoid any mistake in programming):

1. Make sure to select the **BS** as a current scheduler

   sudo i2cset -y 1 0x6b 0x20 0x01 for making BS as selected scheduler

2. Enter **BS Action Time**, on our case it is 1 minute

   sudo i2cset -y 1 0x6b 0x21 0x01 for duration time 1 minute

3. Enter **BS Inaction Time**, on our case it is 1 minutes

   sudo i2cset -y 1 0x6b 0x22 0x01 for repetition time 2 minutes

4. Enter **BS Multiplier**, on our case it is infinitive

   sudo i2cset -y 1 0x6b 0x23 0xff for infinitive running

5. Check if programmed values are OK, by running the below python script

   sudo python status_bs.py

If everything is as expected, run the Basic Scheduler to start immediately

sudo i2cset -y 1 0x6b 0x24 0xff

BS Example 2nd- Simple Raspberry Pi® ON/OFF executed 100 times for 1 minutes (ON/OFF every minute), started beginning next day (00:00)

We need to start up - set ON - the Raspberry Pi®, keep it running for 1 minutes, shutdown it, and after 1-minute start it again. This will be repeated 100 times.

Time_Scheduler_Selector = 0x01;	Select the Basic Scheduler
BS_action_time = 0x01;	Sets duration of Action (ON) time to 1 minute (Raspberry Pi® will run for 1 minute and then shutdown)
BS_inaction_time =0x01;	Sets duration of Inaction (OFF) time to 1 minute (Raspberry Pi® will sleeping for 1 minute)
BS_multipier = 0x64;	This will be repeated 100 times
BS_ RUN =0xfb;	When user decide, just activate the Basic Scheduler, and it is start beginning next day

The data entering should looks like below (it important to follow the below order to avoid any mistake in programming):

The data entering should looks like below (it important to follow the below order to avoid any mistake in programming):

1. Make sure to select the **BS** as a current scheduler

   sudo i2cset -y 1 0x6b 0x20 0x01 for making BS as selected scheduler

2. Enter **BS Action Time**, on our case it is 1 minute

   sudo i2cset -y 1 0x6b 0x21 0x01 for duration time 1 minute

3. Enter **BS Inaction Time**, on our case it is 1 minutes

   sudo i2cset -y 1 0x6b 0x22 0x01 for repetition time 2 minutes

4. Enter **BS Multiplier**, on our case it is infinitive

   sudo i2cset -y 1 0x6b 0x23 0xff for infinitive running

5. Check if programmed values are OK, by running the below python script

   sudo python status_bs.py

If everything is as expected, run the Basic Scheduler to start beginning next day

sudo i2cset -y 1 0x6b 0x24 0xfb

Events Triggered RTC Based System Actions Scheduler

Not activated yet in current firmware version

The **E**vents **T**riggered **R**TC Based **S**ystem **A**ctions **S**cheduler (**ETR SAS**) is a very advanced functionality that allows user to implement a simple timed Actions (usually ON/OFF) of the Raspberry Pi®, but also a very complicated Actions Schedules depended to External Events and Time without or with involvement of Raspberry Pi®. This functionality can be perfectly combined with **IoT** or any other time dependent applications. The time resolution of the **ETR SAS** is based on **1 minute**, however everything is adjusted with 1 second accuracy, as each action start/stop is executed at the beginning (first second) of internal RTC counted minute. There are implemented 4 parallel working **ETR SAS** running with different Set-up's. That means that i.e. user can set the 1st **ETR SAS** running at night (00:00 – 06:00) every 10 minutes (repeated 20 times), the 2nd **ETR SAS** to run at morning time (06:00 – 10:00) every 30 minutes, and the rest of the day every 1 minute based on 3rd **ETR SAS**. The **ETR SAS** can be based on the **RTC**, but can be also time independent and execute Action triggered by external Event (i.e. A/D). The **Action** can be simple ON/OFF the Raspberry Pi® but also independent of the Raspberry PI® (without switching it ON) just activate the Auxiliary 5V@750mA or Bi-Stable Relay switching. Combination of all **ETR SAS** produce in the result a very complicated state machine able to implement practically any schedule is needed by user.

Associated Software

Each Version of manual is related to the valid version of firmware released with it, and any other released after that. The Valid Version of Firmware as also Valid Scripts can be downloaded from below links.

Automatic System Setup and Firmware Upgrade Python Script

The current version of this script is ***UPS_PIco_HV4_BL03A.py*** (click to download it)

Associated Monitoring Scripts

The current version of this script is ***pico_status1.2_hv4.0.py*** (click to download it)

Daemons Scripts

The current version of this script is ***pic_i2c.zip*** (click to download it)

Latest Firmware related to current manual

The current version of this script is ***ups_pico4_main_0144_280922.zip*** (click to download it)

A Complete Description of the UPS PIco HV4.0 HAT Programming Registers

0x69 ->UPS PIco HV4.0 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** - CBL_PWR (means cable powering mode USB or EPR) **0x02** - BAT_MODE (means backup powering mode BAT or SCAP)
0x01	backing_mode	byte	mirror	read	Current backup source if mixed mode used
0x02	rtime	word	mirror	read	Available running time on Battery with current consumption
0x04	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 (watchdog, 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.
0x06	scatlevel	word	mirror	read	Means value of Super Capacitor Voltage in 10^{th} of mV in BCD format
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 GPIO 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 (6.5-28VDC) in 10^{th} of mV in BCD format
0x0e	curilevel	word	mirror	read	Incoming Current to the system via 5VDC source 10^{th} of mA in BCD format
0x10	curolevel	word	mirror	read	Outgoing Current from the system via 5VDC source 10^{th} of mA in BCD format
0x14	ecilevel	word	mirror	read	External Powering Ingoing Current to the system via 32VDC source 10^{th} of mA in BCD format
0x16	aEXT0level	word	mirror	read	Means value of the second A/D converter pre-scaled to 3.3V. Higher voltage could be supplied with an external resistor divider. Readings are in 10^{th} of mV in BCD format.
0x18	aEXT1level	word	mirror	read	Means value of the second A/D converter pre scaled to 3.3V. Higher voltage could be supplied with an external resistor divider. Readings are in 10^{th} of mV in BCD format.

<table>
<tr><td>0x1a</td><td>key_pressed</td><td>byte</td><td>Common</td><td>R/W</td><td>User Kay Pressed information

Read: 0x01 – Pressed key A
Read: 0x02 – Pressed key B

Write: 0x00 – Reset (clear) after the current reading and prepare for the next one.</td></tr>
<tr><td>0x1b</td><td></td><td></td><td></td><td></td><td>NO Implemented YET</td></tr>
<tr><td>0x1c</td><td>RPI_temp</td><td>byte</td><td>mirror</td><td>read</td><td>Temperature in Celsius degree of Raspberry Pi Core</td></tr>
<tr><td></td><td>optlevel</td><td>word</td><td>mirror</td><td>read</td><td>NO Implemented YET</td></tr>
<tr><td></td><td>BAT_chg_stat</td><td>byte</td><td>mirror</td><td>read</td><td>TBS</td></tr>
<tr><td></td><td>BAT_chg_set_cur</td><td>byte</td><td>mirror</td><td>read</td><td>Charging max current (could be lower if battery is charged)</td></tr>
<tr><td></td><td>FAN_status</td><td>byte</td><td>mirror</td><td>read</td><td>TBS</td></tr>
<tr><td></td><td>sysinfo</td><td>word</td><td>mirror</td><td>read</td><td>System Information</td></tr>
<tr><td></td><td>RS232_status</td><td>byte</td><td>mirror</td><td>read</td><td>TBS</td></tr>
<tr><td></td><td>RS232_data</td><td></td><td></td><td></td><td>TBS</td></tr>
<tr><td>0x35</td><td>PCB_version</td><td>byte</td><td>mirror</td><td>read</td><td>PCB Version: PCB/Bootloader Version "0x40", "0x41"</td></tr>
<tr><td>0x36</td><td>PIco_model</td><td>byte</td><td>mirror</td><td>read</td><td>Below Version can be powered by Battery and Super Capacitor at the same time
"S" BC Stack
"A" BC Advanced
"P" BC PPoE

Below Version can NOT be powered by Battery or Super Capacitor at the same time
"T" B Stack
"B" B Advanced
"Q" B PPoE

Below Version can be powered only by Battery
"U" D Stack
"C" D Advanced
"R" D PPoE</td></tr>
<tr><td>0x37</td><td>default battery</td><td>byte</td><td>mirror</td><td>read</td><td>PCB default Battery Type. Battery Type can be changed at any time, however after factory default (i.e., new firmware update) will be set again this hard defined. It is stored in the PCB bootloader protected memory
<table>
<tr><td colspan="4">Single Mode</td></tr>
<tr><td>LiPO</td><td>'L'</td><td>0x4C</td><td></td></tr>
<tr><td>Li-Ion</td><td>'I'</td><td>0x49</td><td></td></tr>
<tr><td>LiFePO4</td><td>'F'</td><td>0x46</td><td></td></tr>
<tr><td>NiMH</td><td>'H'</td><td>0x48</td><td></td></tr>
<tr><td>SAL</td><td>'A'</td><td>0x41</td><td></td></tr>
<tr><td>ISC</td><td>'C'</td><td>0x43</td><td>Internal Super Cap (100F)</td></tr>
<tr><td>ESC</td><td>'D'</td><td>0x44</td><td>External Super Cap Bank (300F-500F-800F)</td></tr>
<tr><td colspan="4">Mixed Mode</td></tr>
<tr><td>LiPO+SC</td><td>'l'</td><td>0x6C</td><td></td></tr>
<tr><td>Li-Ion+SC</td><td>'i'</td><td>0x69</td><td></td></tr>
<tr><td>LiFePO4+SC</td><td>'f'</td><td>0x66</td><td></td></tr>
<tr><td>NiMH+SC</td><td>'h'</td><td>0x68</td><td></td></tr>
<tr><td>SAL+SC</td><td>'a'</td><td>0x61</td><td></td></tr>
</table></td></tr>
<tr><td>0x38</td><td>firmware version</td><td>word</td><td>mirror</td><td>read</td><td>Firmware Version</td></tr>
</table>

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	weekday 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: 0x60** – Set I2C base address to 0x60. Therefore, all parameters will be related to this available base address: 0x68 (RTC and is set then UU), 0x69, 0x6A, 0x6B, 0x6C, 0x6E, 0x6E, 0x6F **Write: 0x68** – Set I2C base address to 0x60. However, the RTC is free and all related to this register's addresses. Therefore, all parameters will be related to this available base address: 0x69, 0x6B **Write: 0x50** – Set I2C base address to 0x50. Therefore, all parameters will be related to this available base address: 0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5E, 0x5E, 0x5F. User need to take care and change the appropriate Daemon addresses and reload Daemon to have system properly running. **Write: 0x40** – Set I2C base address to 0x40. Therefore, all parameters will be related to this available base address: 0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4E, 0x4E, 0x4F. User need to take care and change the appropriate Daemon addresses and reload Daemon to have system properly running. **Write: 0xaa** – Unconditional File Safe Shutdown and (and Power OFF when battery powered) **Write: 0xdd** - then restore factory defaults. Battery Type will be set according to what has been stored in the original setup. 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 causes 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 happens. 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**

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td>Mode.</td></tr>
<tr><td>0x01</td><td>bat_run_time</td><td>Byte</td><td>Common</td><td>R/W</td><td>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.

<u>Factory default value is 70 seconds</u>

Each number stands for 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</td></tr>
<tr><td>0x02</td><td>rs232_rate</td><td>Byte</td><td>Common</td><td>R/W</td><td>Writing to this register sets the UPS PIco HV4.0 Serial Ports to the following settings:
RED is the Default Value

<table>
<tr><th>Serial Port</th><th>Value</th><th>Comm. Rate/Function</th></tr>
<tr><td>RPi</td><td>0x00</td><td>Serial Port is OFF (HiZ)</td></tr>
<tr><td>RPi</td><td>0x01</td><td>Serial Port is 4800</td></tr>
<tr><td>RPi</td><td>0x02</td><td>Serial Port is 9600</td></tr>
<tr><td>RPi</td><td>0x03</td><td>Serial Port is 19200</td></tr>
<tr><td>RPi</td><td>0x04</td><td>Serial Port is 38400</td></tr>
<tr><td>RPi</td><td>0x05</td><td>Serial Port is 57600</td></tr>
<tr><td>RPi</td><td>0x06</td><td>Serial Port is 115200</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>PIco HV4.0</td><td>0x00</td><td>Serial Port is OFF (HiZ)</td></tr>
<tr><td>PIco HV4.0</td><td>0x10</td><td>Serial Port is 4800</td></tr>
<tr><td>PIco HV4.0</td><td>0x20</td><td>Serial Port is 9600</td></tr>
<tr><td>PIco HV4.0</td><td>0x30</td><td>Serial Port is 19200</td></tr>
<tr><td>PIco HV4.0</td><td>0x40</td><td>Serial Port is 38400</td></tr>
<tr><td>PIco HV4.0</td><td>0x50</td><td>Serial Port is 57600</td></tr>
<tr><td>PIco HV4.0</td><td>0x60</td><td>Serial Port is 115200</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td>Users need to set both Serial Ports at once i.e. by writing 0x66 means both ports set to 115200, by writing 0x06 means one port is set to OFF and second port set to 115200</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0x05</td><td>STA_timer</td><td>Byte</td><td>Common</td><td>R/W</td><td>Still Alive Timeout Counter in seconds

Read: Anytime, Return actual sta_timer value

Write: 0xff – Disable the counter (default value)

Write: 0x01 – 0xfe Enable and Start down counting of the Still Alive Timer in resolution of 1 second, until reaches value of 0x00 which initiate Unconditional Hardware Reset Procedure

Write:0x00 – Initiate immediately File Safe Shutdown Procedure and system restart with similar conditions as described below

In order to use it as Still Alive (type of watchdog) timer, user needs to upload value from 0x01 to 0xfe earlier than defined time of seconds. Not uploading of this value will cause System Unconditional Hardware Reset (so System to be Restarted)

In order to have this feature working the Gold Plated Reset Pin must be instaled

This feature is working on Battery or Cable powering

After execution of the STA Restart the sta_timer is set again to 0xff (disabled).</td></tr>
<tr><td>0x06</td><td>enable5V</td><td>Byte</td><td>Common</td><td>R/W</td><td>Defines usage of the Auxiliary 5V@750mA:
0x00 – Auxiliary 5V and 3.3V are not battery backed-up
0x01 – Auxiliary 5V and 3.3V are battery backed-up
Default Values is OFF
Other codes are not allowed</td></tr>
<tr><td>0x07</td><td>battype</td><td>Byte</td><td>Common</td><td>R/W</td><td>Defines used battery chemistry type:
<table>
<tr><td colspan="4">Single Mode</td></tr>
<tr><td>LiPO</td><td>'L'</td><td>0x4C</td><td></td></tr>
<tr><td>Li-Ion</td><td>'I'</td><td>0x49</td><td></td></tr>
<tr><td>LiFePO4</td><td>'F'</td><td>0x46</td><td></td></tr>
<tr><td>NiMH</td><td>'H'</td><td>0x48</td><td></td></tr>
<tr><td>SAL</td><td>'A'</td><td>0x41</td><td></td></tr>
<tr><td>ISC</td><td>'C'</td><td>0x43</td><td>Internal Super Cap (100F)</td></tr>
<tr><td>ESC</td><td>'D'</td><td>0x44</td><td>External Super Cap Bank (300F-500F-800F)</td></tr>
<tr><td colspan="4">Mixed Mode</td></tr>
<tr><td>LiPO+SC</td><td>'l'</td><td>0x6C</td><td></td></tr>
<tr><td>Li-Ion+SC</td><td>'i'</td><td>0x69</td><td></td></tr>
<tr><td>LiFePO4+SC</td><td>'f'</td><td>0x66</td><td></td></tr>
<tr><td>NiMH+SC</td><td>'h'</td><td>0x68</td><td></td></tr>
<tr><td>SAL+SC</td><td>'a'</td><td>0x61</td><td></td></tr>
</table>
Other codes are not allowed</td></tr>
<tr><td>0x08</td><td>setA_D</td><td>Byte</td><td>Common</td><td>R/W</td><td>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.</td></tr>
</table>

					Read: Anytime, Return actual **setA_D** value **Write:** 0x00 – 5.2V pre-scale for the AEXT**2level** **Write:** 0x00 – 5.2V pre-scale for the AEXT**1level** **Write:** 0xFF – all A/D registers will contain raw data
0x09					
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:** 0x00 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 **Default Value:** 0x01
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** **Write:** 0x02 – Automatic ON/OFF 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 Default value is set to 0x02 – Automatic ON/OFF
0x12	fspeed	Byte	Common	R/W	**Integrated Fan Speed** **Read:** Anytime, Return actual **fspeed** value **Write:** 00 – Selected speed when OFF 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** Default speed is set to 50% **Any data written to this register are stored in the internal EEPROM. So, even if UPS PIco HV3.0 will be reset, will be recovered.**
0x13	fstat	Byte	Mirror	Read	**Read:** Anytime, return actual **if FAN** is actually running or not (for remote users) When FAN is set to be running (even if not

					connected physically) the FAN LED is lighting. The intensity of the FAN LED is depending of the FAN Speed (PWM)
0x14	fttemp	Byte	Mirror	R/W	**Integrated Fan Temperature Threshold in Automatic Mode** BCD Fan Running threshold temperature in Celsius, 2 digits i.e., 35, means 35 Celsius. To be used (automatic FAN ON/OFF) need to set **fmode** to 0x02. Maximum temperature is 60 Celsius. Higher values will be ignored. FAN will start at 36 Celsius and stop at 35 Celsius. **Read:** Anytime, Return actual **fspeed** value **Write:** 00 – 60 Sets the temperature Threshold for the Automatic FAN Start/Stop **Default value is set to 50 Celsius**
0x15	LED_OFF	Byte	Common	R/W	Added LED OFF, that switch OFF all software controlled LEDs. CHG, FAN, EXT can not be switched off as are connected to the hardware and controlled by it. By writing the 0x00 to LEDOFF disable the LEDs. Default is 0x01 (means LED ON)

Events Triggered RTC Based System Actions Scheduler Commands

0x6c -> Start Time Stamp

Address	Name	Size	Type	R/W	Explanation
0x00	active	Byte	Common	R/W	Activation Stamp 0x00 not active (Stop), 0xff active (Start) of current SAS
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
0x06	error	Byte	Mirror	Read	ETR SAS errors

0x6d -> Actions Running Time Stamp

Address	Name	Size	Type	R/W	Explanation
0x00	error	Byte	Mirror	Read	ETR SAS errors
0x01	Duration	Word	Common	R/W	In BCD 4 digits minutes 1-9999
0x03	Repetition	Word	Common	R/W	In BCD 2 digits (1-9999) every XXXX minutes: 0x0000 – not repeated (only once for Duration time, on programed time) 0x0001 – 0x9999 – every 1-9999 minutes i.e. 0x0010 means every 10 minutes

0x6e -> Events Stamp

Address	Name	Size	Type	R/W	Explanation

0x6f -> Actions Stamp

Address	Name	Size	Type	R/W	Explanation
0x00	RPi_PON	Byte	Common	R/W	Raspberry Pi Power ON Activate: 0x01 Deactivate: 0x00 Default: 0x00 Write 0x01 to have this Action Active and switch Raspberry Pi® Powering ON
0x01	5V_PON	Byte	Common	R/W	Auxiliary 5V@750mA and 3V3 Power ON Activate: 0x01 Deactivate: 0x00 Default: 0x00

					Write 0x01 to have this Action Active and switch Auxiliary 5V@750mA and 3V3 Powering ON
0x02	BR_Set	Byte	Common	R/W	Bi-Stable Relay Set Activate: 0x01 Deactivate: 0x00 Default: 0x00 Write 0x01 to have this Action Active and Set the Bi-Stable Relay