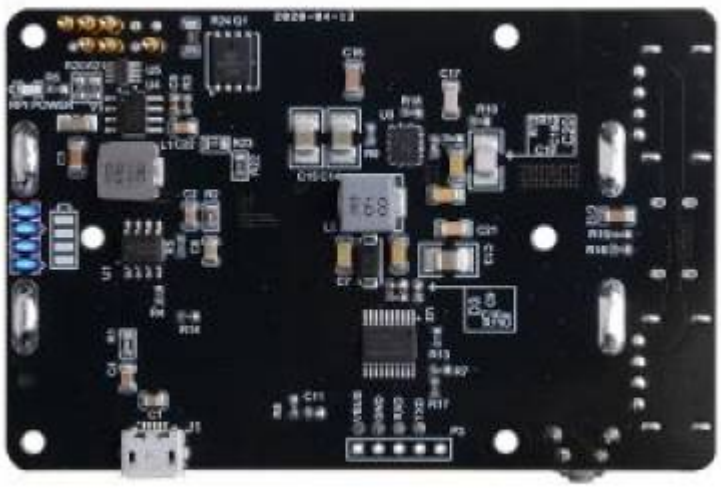
UPS EP-0118

Source :
https://wiki.52pi.com/index.php/UPS_(With_RTC_%26_Coulometer)_For_Raspberry_Pi_SKU:_EP-0118

UPS (With RTC & Coulometer) For Raspberry Pi 4B/3B+/3B



Description

This is a new version of UPS Board.It not only supports Raspberry Pi 4B but also adds type C output. It provides enough power for the Raspberry Pi to make your Raspberry Pi still work while moving, and its design is so smart that you can get rid of troubled of a mass wire. This UPS can provide you with the operation of replacing the battery yourself. This new UPS not only supports an independent RTC module, but also provides a coulomb counter. You can obtain UPS OUTPUT voltage and current information through I2C in the system. Combined with different programming methods, you can inherit the power information into the system. It is also convenient to detect the state of the battery, which can be detected and warned through the program. It can use most of the 18650 standard batteries and fully comply with the battery characteristics in terms of battery life. In addition, the power display provided on the circuit board is also very user-friendly. The illuminated LED light can quickly show the remain power and support the discharging while charging. You can directly connect to the external power supply for charging. At the same time, the Raspberry Pi will not be turned off. The Coulometer can not detect batteries voltage and current due to different batteries has different specifications Please install the battery first, then connect the UPS to the Raspberry Pi. Connecting too many high-power peripherals will lead to insufficient power supply. Please pay attention to the power

demand of peripherals.

Caution

Please do not reverse the battery! Only for 18650 batteries and the discharge rate should be less than 10A !!!

```
This product belongs to the power supply device.
Please keep it out of the reach of children. Please strictly refer to the
instructions for the UPS  product. If the battery polarity is reversed, it
will directly burn the power management chip of the device and even cause
fire or personal injury. Do not add any Components by yourself, all
accidental damage caused by the cause is at your own risk.
```

Make sure your power adapter capacity can cover lithium battery charging and raspberry pi use case, typically is 5V 3A or better should be used.

```
 (ex:Battery charge may require up to 2.1A, Raspberry Pi typically requires
0.6-0.8A, but up to 2.4A, at which if the power supply current is less than
4.5A, it may not be able to continue use)
```

ESD Warning When connecting the battery, please do not touch the area where the yellow frame is located. ESD may cause damage to your device, so please avoid it.

Battery specifications

In order to prevent damage to the UPS equipment, please pay attention when purchasing the battery: The discharge rate of each battery cannot exceed 2C, the battery voltage is 4.2V when fully charged, the parameters of the two batteries must be the same, and the batteries cannot be mixed.

ChangeLog

update: 2020-9-8

Limited battery capacity and discharge rate

update: 2020-4-30

Fixed some bugs. Upgrade the module functions, add RTC moudle and coulometer voltameter function. Adding more pogo-pin to support new features. Update the firmware version. Changed new conductor to support the new features.

Features

New Features

RTC Independent Clock Current and Voltage Reading Battery Indicator Programmable reading output voltage in OS（Not Batteries voltage） Charging Status LED indicator

Old Features

LED remaining battery prompt Replaceable battery solution Support Raspberry Pi 4B/3B+/3B Charging up to 2.1A Rated discharge power is 15W Extended Two USBA port power output Extended type C port power output Output overcurrent protection Tap to boot, long press to shut down Support the charge and discharge power display The battery holder can be quickly replaced Synchronous switch charge and discharge Built-in power path management, support side-loading Support load high current line compensation function Adaptive charge current regulation to match all adapters Support LED power display Button boot: UPS will start when the button is pressed, Long Press to force turn off Low power consumption Multiple protection, high reliability Output overcurrent, overvoltage, short circuit protection Input overvoltage, overcharge, overdischarge, overcurrent discharge protection Machine over temperature protection Gallery EP 0114 UPS 9.jpg EP 0114 UPS 11.jpg EP 0114 UPS 12.jpg Function area definition EP 0114 UPS 13.jpg EP 0114 UPS 14.jpg EP 0114 UPS 15.jpg Package includes 1* UPS (With RTC & Coulometer) For Raspberry Pi 4B/3B+/3B 4* M2.5* copper stick 4* M2.5 screws 4* M2.5* long copper stick 4* M2.5 nuts 1* Acrylic shield 1* Instructions EP 0114 UPS 6.jpg

Mechanical Drawing UPS 1.png

How to setup EP 0114 UPS 1.jpg

LED light status definition Discharge Status Power C(%) LED1 LED2 LED3 LED4 C >= 75% ON ON ON ON 50% ⇐ C <75% ON ON ON OFF 25% ⇐ C <50% ON ON OFF OFF 3% ⇐ C < 25% ON OFF OFF OFF 0% ⇐ C < 3% 1.5Hz Blink OFF OFF OFF EP 0114 UPS 8.jpg

Charging Status Power C(%) LED1 LED2 LED3 LED4 100% ON ON ON ON 75% ⇐ C ON ON ON 1.5Hz Blink 50% ⇐ C <75% ON ON 1.5Hz Blink OFF 25% ⇐ C < 50% ON 1.5Hz Blink OFF OFF C < 25% 1.5Hz Shining OFF OFF OFF How to Read Voltage and Current Flash the latest Raspbian OS to TF card and boot up Raspberry Pi. Make sure Raspberry Pi connect to internet and update the repository's information with command: “sudo apt-get update” Make sure I2C function has been turned on. Install the python liberary: sudo pip3 install pi-ina219 Create a new .py file and paste following demo code. # IMPORT THE LIBERARY. from ina219 import INA219 from ina219 import DeviceRangeError SHUNT_OHMS = 0.05

def read():

```
    """Define method to read information from coulometer."""
    ina = INA219(SHUNT_OHMS)
    ina.configure()
    print("Bus Voltage: %.3f V" % ina.voltage())
    try:
        print("Bus Current: %.3f mA" % ina.current())
        print("Power: %.3f mW" % ina.power())
        print("Shunt voltage: %.3f mV" % ina.shunt_voltage())
    except DeviceRangeError as e:
```

```
      print(e)
```

if name == “main”:

```
  read()
```

Run the demo code: python demo.py How to setup and check the RTC module Using Overlays Overlays are loaded using the “dtoverlay” config.txt setting. As an example,consider I2C Real Time Clock drivers. In the pre-DT world these would be loaded by writing a magic string comprising a device identifier and an I2C address to a special file in /sys/class/i2c-adapter, having first loaded the driver for the I2C interface and the RTC device - something like this: PS: sudo su means use root user to operate the system file, otherwise you may encount an error: “Permission denied”

```
  modprobe i2c-bcm2835
  modprobe rtc-ds1307
  sudo su
  echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

With DT enabled, this becomes a line in config.txt: Edit /boot/config.txt file via vim.tiny tool or nano in a terminal: sudo vim.tiny /boot/config.txt And then add this parameter:

```
  dtoverlay=i2c-rtc,ds1307
```

This causes the file /boot/overlays/i2c-rtc.dtbo to be loaded and a “node” describing the DS1307 I2C device to be added to the Device Tree for the Pi. By default it usees address 0x68, but this can be modified with an additional DT parameter:

```
  dtoverlay=i2c-rtc,ds1307,addr=0x68
```

Parameters usually have default values, although certain parameters are mandatory. See the list of overlays below for a description of the parameters and their defaults.

For Raspberry Pi 4B Here we assume that you have already burned the Raspbian Image into TF card and connect to your PC and logged in. Open a terminal and modify /boot/config.txt file using what you favorate editor such as vim.tiny or nano, add parameters as following picture:

DS1307-4B(1).png

You can read /boot/overlay/README and find this info to add support for ds1307 I2C Real Time Clock device.

DS1307-4B(2).png

please ensure that /boot/config.txt file include two paramaters:

dtoverlay=i2c-rtc,ds1307 dtparam=i2c_arm=on After that, please make sure you have disabled the “fake hwclock” which interferes with the 'real' hwclock sudo apt-get -y remove fake-hwclock sudo update-rc.d -f fake-hwclock remove Now with the fake-hw clock off, you can start the original 'hardware clock' script. Edit the script file /lib/udev/hwclock-set with nano or vim editor and comment out these three lines: if [-e /run/systemd/system] ; then exit 0 fi Finally result like this: DS1307-4B(6).png

save and reboot your RPi.

For Raspberry Pi 3B Here we assume that you have already burned the Raspbian Image into TF card and connect to your PC and logged in. Open a terminal and modify /boot/config.txt file using what you favorate editor such as vim.tiny or nano, add parameters as following picture:

Ds13076.png

You can read /boot/overlay/README and find this info to add support for ds1307 I2C Real Time Clock device.

Name: i2c-rtc Info: Adds support for a number of I2C Real Time Clock devices Load: dtoverlay=i2c-rtc,<param>=<val> Params: ds1307 Select the DS1307 device please ensure that /boot/config.txt file include those three paramaters:

device_tree=bcm2710-rpi-3-b.dtb dtoverlay=i2c-rtc,ds1307 dtparam=i2c_arm=on After that, please make sure you have disabled the “fake hwclock” which interferes with the 'real' hwclock sudo apt-get -y remove fake-hwclock sudo update-rc.d -f fake-hwclock remove Now with the fake-hw clock off, you can start the original 'hardware clock' script. Edit the script file /lib/udev/hwclock-set with nano or vim editor and comment out these three lines: if [-e /run/systemd/system] ; then exit 0 fi Finally result like this: Raspberry pi hwclock-set.png

save and reboot your RPi.

How to Check it After reboot and log in, open a terminal and typing this command to check if RTC module is functional. dmesg |grep rtc if you can see this picture means that your RTC module is working properly.

Ds13077.png

and then you can adjust your system clock as following command:

Ds13078.png

Note: 051014302016.20 is equal to mmDDHHMMYYYY.ss, more information please using ‘man date’ command. Last step, set the Hardware Clock to the current System Time. Ds13079.png

ok, finished. Have fun.

GitHub Github:[https://github.com/geeekpi/upsv3] Video Tutorial Youtube.jpeg

Please follow the link: [UPS With RTC and Coulometer For Raspberry Pi Usage Tutorial | https://youtu.be/HCyXJtXGaHw]

FAQ Q: Dose it support Raspberry Pi 4B with my 7 inch LCD? A: It depends on how much current will your 7 inch LCD cost and how many external device that you connect to. Q: It is safe for my home assistant project? A: Yes, It's safe. Q: Why does my RTC module running a wrong time when i reboot my Pi ? A: Please do remember to disable the “fake hwclock” which interferes with the 'real' hwclock as following commands:

sudo apt-get -y remove fake-hwclock sudo update-rc.d -f fake-hwclock remove Keywords battery, 18650, lipo battery, UPS, current, power supply,raspberrypi 4B FAQ Q: what will your UPS do when the power will restored? A: It will do nothing when the power is restored. Q: How can I restart my

Raspberry automatically? (without press the power-on button) A: Sorry for that, it has no function like that, you need to press the power-on button to start it. Q: Your UPS will restart the Raspberry automatically? A: No, It will not restart the Raspberry automatically unless you press the power-on button.