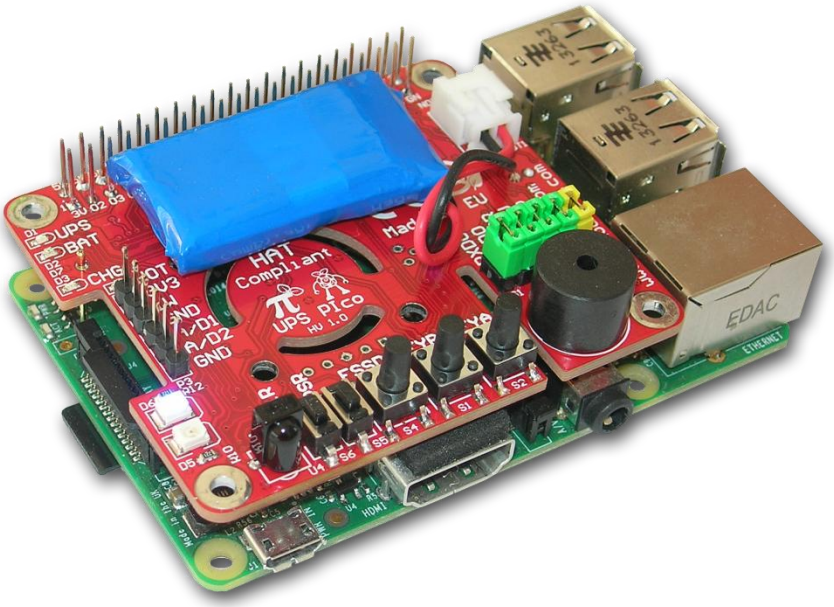
UPS PIco

Uninterruptible Power Supply

with Peripherals and I²C control Interface

for use with

Raspberry Pi® Pi2, B+, A+, B, and A



HAT Compliant

Simple Setting Guide for the UPS PIco

Intelligent Modules for your Raspberry Pi®

Document Revisions

Version	Date	Modified Pages	Modified Sections	Comments
1.0	15/11/2015			First Public Document Release
1.1	27/01/2016			Updated Various Instructions

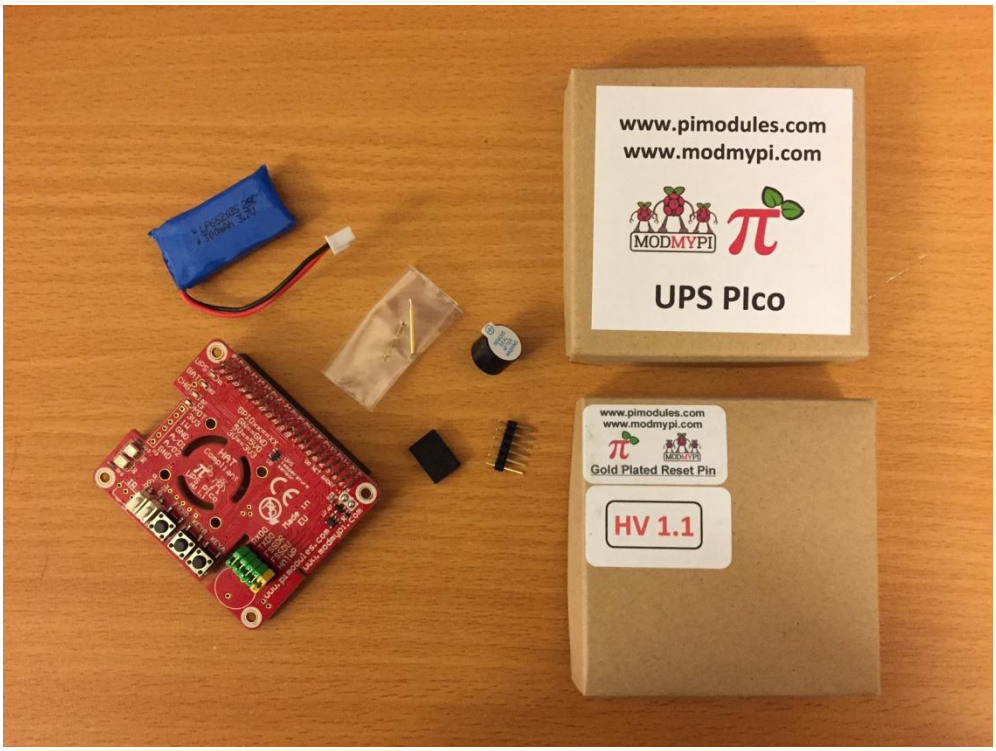
What is in the BOX?

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

Each Box contains the following parts:

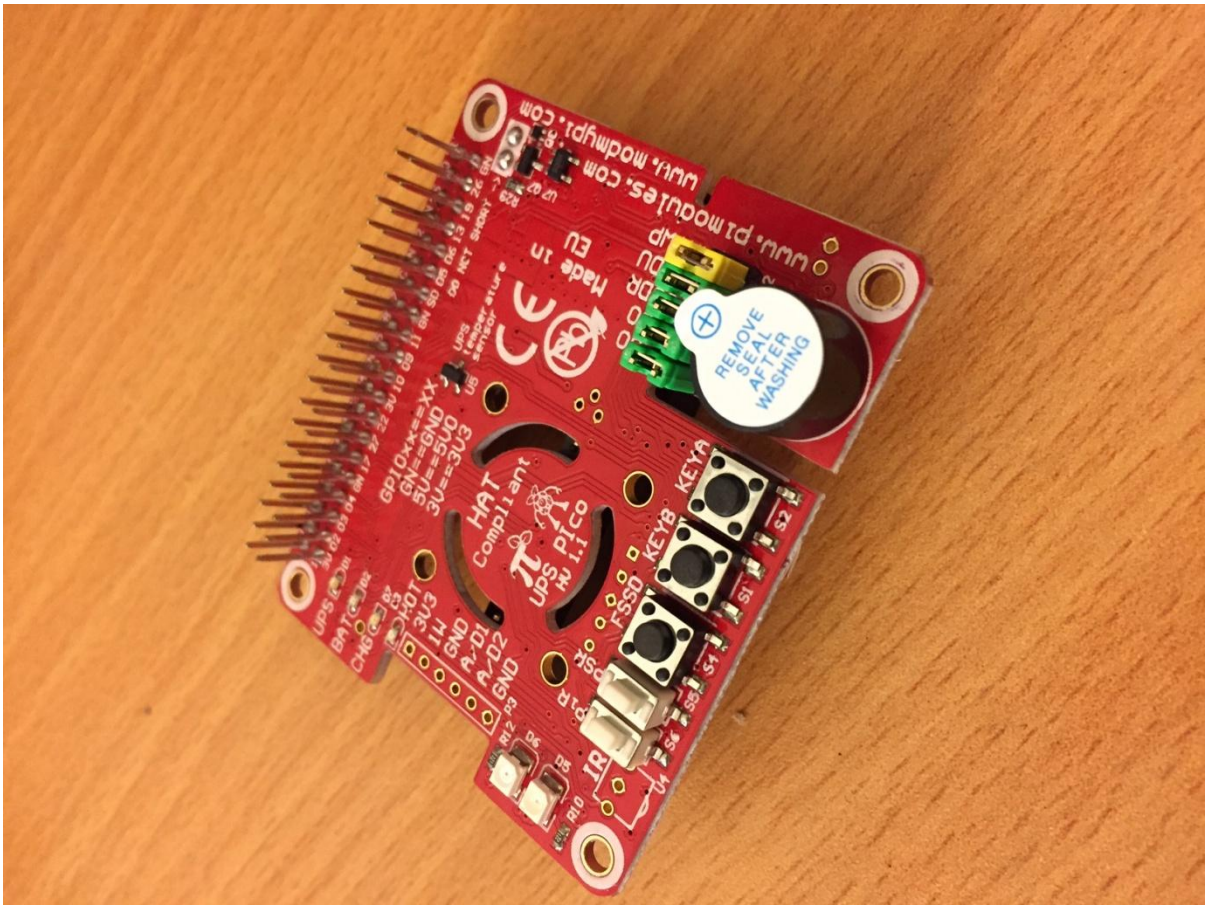
- 1 x The **UPS PIco** module assembled and tested
- 1 x Dual layer wide temperature adhesive tape (used for battery mounting) stuck on the bottom side of **UPS PIco battery**
- 1 x Set of installed jumpers
- 1 x Separate packed LiPO battery
- 1 x Gold Reset Pin
- 1 x Buzzer
- 1 x Sticky Pad
- 1 x Header Strip

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



Setting up Procedure

- Ensure that all jumpers are in place on the **UPS PIco** module. There are 4 green, and 1 yellow.
- If you would like to use the buzzer, you can solder it on now. **Ensure that this is done with the correct polarity**. Positive "+" on the board should match the positive "+" on the buzzer.



- If you would like to install the Gold Plated Reset Pin, please do so now following the instructions below. It is not mandatory to install the pin at this time, but it will enable full function of the **UPS PIco** module.

  http://www.pimodules.com/_pdf/UPS_PIco_Reset_Pin_Assembly_Manual.pdf

- If you would like to install the PIco fan kit, please do so now following the instructions below. It is not mandatory to install the fan kit at this time, but it will enable full function of the **UPS PIco** module.

  http://www.pimodules.com/_pdf/UPS_PIco_Fan_Kit_Assembly_Manual.pdf

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

UPS PIco Hardware Installation

There are two different iterations of **UPS PIco** boards available at this time HV1.0 & HV1.1. Please follow the guide for your specific board below to ensure correct installation. To find out which board you have, check the version number under the text “HAT Compliant”.

Please ensure that you have a good quality power supply available for powering the Raspberry Pi. 5V 2A is recommended. This will ensure that there is enough current to recharge the PiCo’s battery.

The **UPS PIco** uses two GPIO pins to communicate with the Raspberry Pi:

- GPIO_22 – Generates pulse trains that are recognised by the **UPS PIco**. This allows the PiCo to judge whether the Pi is running or not.
- GPIO_27 – Operates the initiation of the File Safe Shutdown Process (FSSD)

When the **UPS PIco** hardware is installed correctly, and the **UPS PIco** recognises the pulse train, the **UPS** LED will twice once per second. This will only occur after a proper software set-up. If the **UPS** LED is not blinking, or blinking very fast, your system is not correctly set-up, and the power back-up will not work. The following LED blinking patterns are available from the UPS LED:

- Very Fast (200ms) – Indicates Raspberry Pi is booting
- Very Fast (200ms) – Indicates cold boot (when the UPSR button has been pressed)
- Very Fast (200ms) – Indicates Raspberry Pi is shutting down
- Normal (500ms) – Indicates the Raspberry Pi is powered and UPS has a stable pulse train
- Slow (2000ms) – Indicates Raspberry Pi is being powered by the UPS battery
- Off – UPS is in low power mode (LPR) and the Raspberry Pi is off

If you wish to remove the **UPS PIco** board from your Raspberry Pi, please ensure that the Raspberry Pi is **NOT** cable powered, and that the battery connector on the UPS has been removed from the white socket. **In other words, ensure that the Raspberry Pi is completely disconnected from all power sources, and completely shut-off, before removing the UPS PIco. For the HV1.1, it is not needed to remove the battery cable, it is enough to press the UPSR button for about 1 second. This will electrically disconnect the battery from the system.**

HV1.0 Installation

- Ensure that the **UPS PIco** battery is **NOT** plugged into the UPS.
- Glue the sticky pad onto the top of the Raspberry Pi's HDMI connector. This acts as a small spacer to keep the board in place.
- Ensure that the Raspberry Pi is **NOT** powered e.g. the unit is completely off.
- Plug the **UPS PIco** onto the Raspberry Pi's GPIO
- Turn your Raspberry Pi on by installing the power cord to the micro USB connector
- The UPS will react to the power:
    - The red & blue LEDs will flash 10 times every second, and the buzzer will beep every second. If the fan kit is installed, this will spin during the cold start up procedure.
- Now, plug the battery onto the white battery connector on the top of the **UPS PIco**.
- That is hardware installation complete! Please note. Software installation must be completed before the **UPS PIco** will function.

HV1.1 Installation

- Plug the UPS battery into the white battery socket connector on the **UPS PIco**
- Pass the battery cable through the cable slot on the **UPS PiCo** board so that the battery sits on top of the **UPS PIco**.
- **Press the UPSR button on the UPS PIco for about 1 second. This disconnects the battery electrically from the system for initial boot.**
- Ensure that the Raspberry Pi is **NOT** powered e.g. the unit is completely off.
- Plug the **UPS PIco** onto the Raspberry Pi's GPIO
- Turn your Raspberry Pi on by installing the power cord to the micro USB connector
- The **UPS PIco** will react to the power:
    - The red & blue LEDs will flash 10 times every second, and the buzzer will beep every second. If the fan kit is installed, this will spin during the cold start up procedure.
- That is hardware installation complete! Please note. Software installation must be completed before the **UPS PIco** will function.

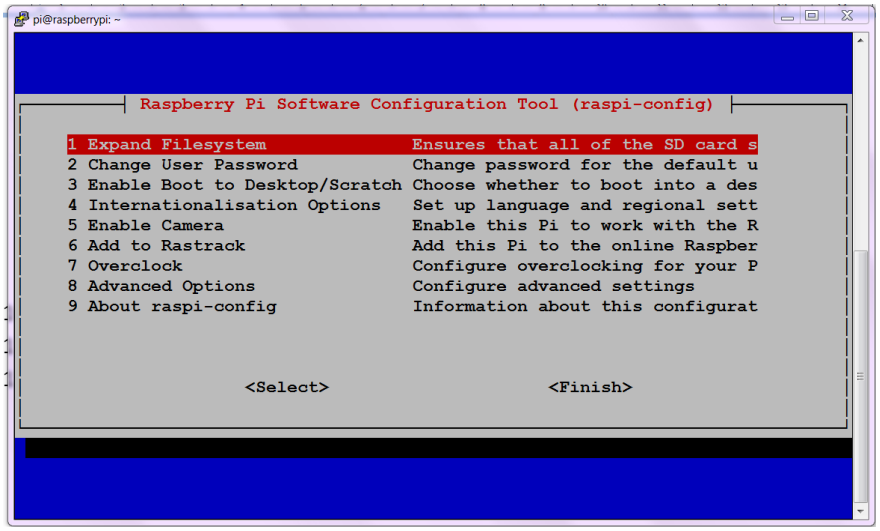
UPS PIco Software Installation

The UPS PiCo will not function unless the following software is installed, and the Pi correctly set-up for UPS use.

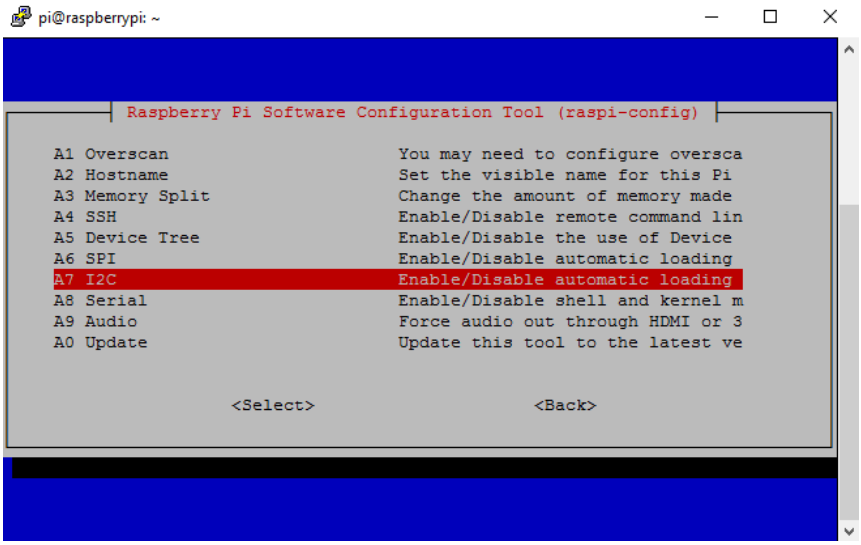
- Ensure that you have the latest version of your chosen operating system. The **UPS PIco** is designed for use on Raspian.
- Ensure that you have a stable and valid internet connection.
- Ensure that your Pi is powered by PSU with at least 5V 2A output.
- From terminal, enter the Raspberry Pi configuration menu using the following command:

  sudo raspi-config

- Scroll to **Advanced Options**

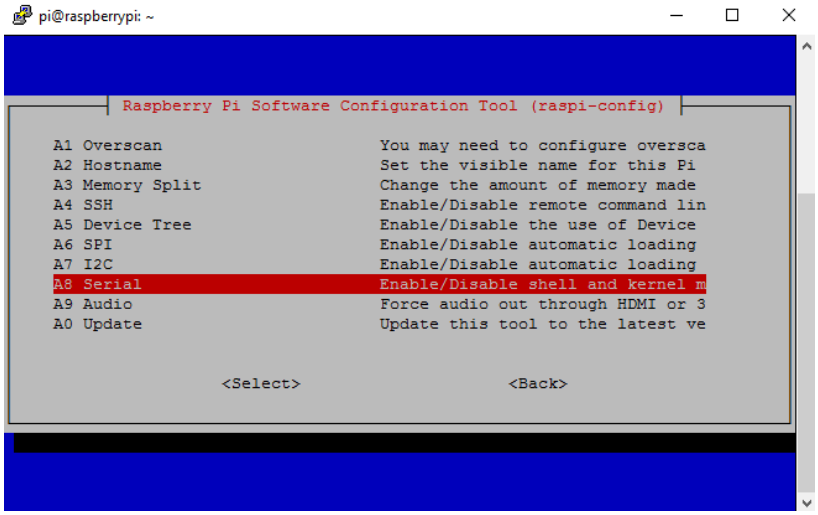


- Then to A7 I2C interface and set this to enabled.



- Select Yes, to set the kernel to be loaded by default.

- Then scroll to A8 Serial, and ensure this is enabled (it is usually enable by default anyway).



- Ensure that this is enabled (it is usually enable by default anyway).

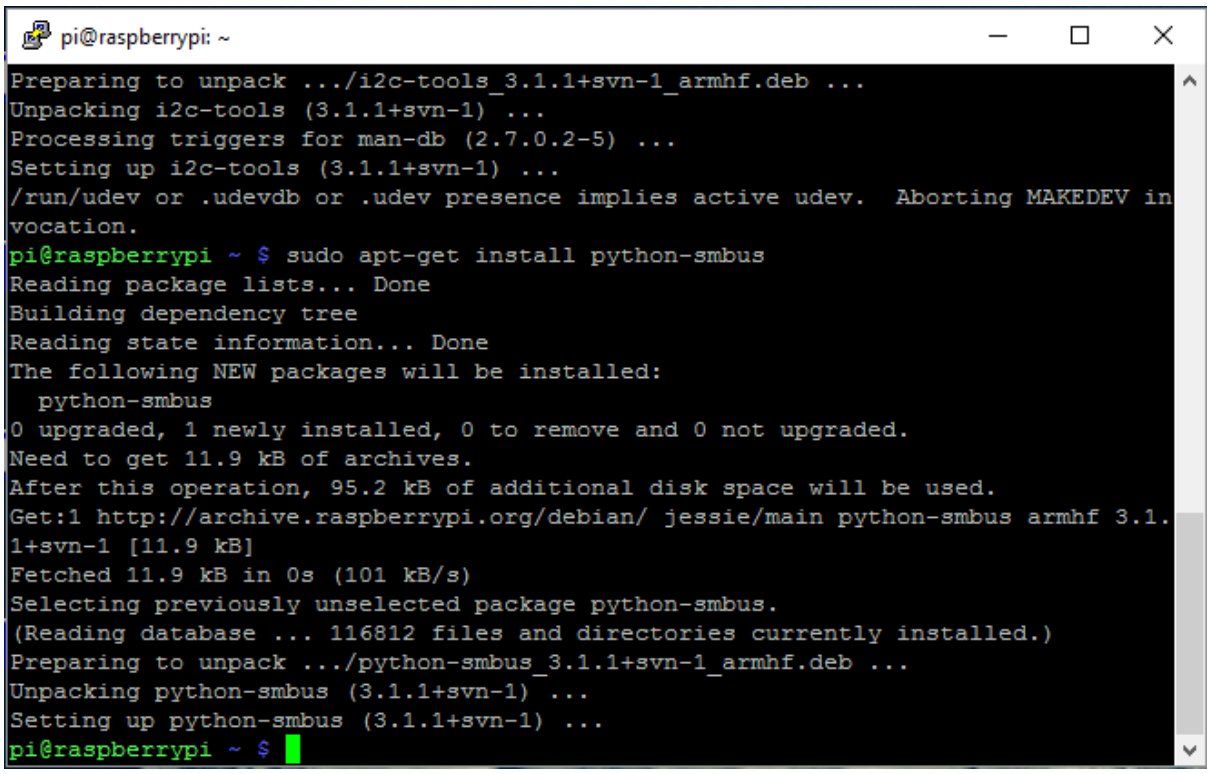
- **If the system requests a reboot, please do so, then continue set-up.**
- Next, ensure that Python is installed and updated, by using the following command

sudo apt-get install python-rpi.gpio

```
pi@raspberrypi ~ $ sudo apt-get install python-rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-rpi.gpio is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

- Install the python-smbus package, by using the following command

sudo apt-get install python-smbus



- Then, ensure that i2c is installed and updated, by using the following command

sudo apt-get install i2c-tools

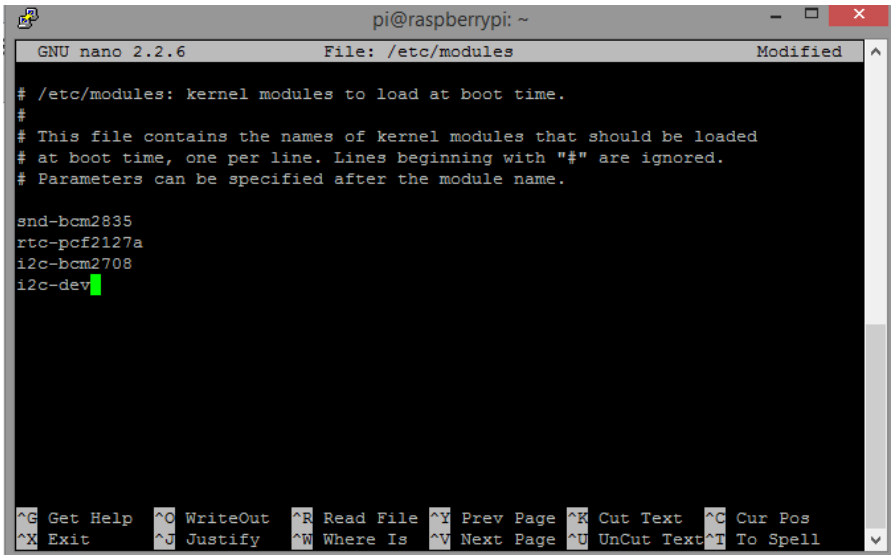
```
pi@raspberrypi ~ $ sudo apt-get install i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

- Now edit the “/etc/modules” file by using the following command

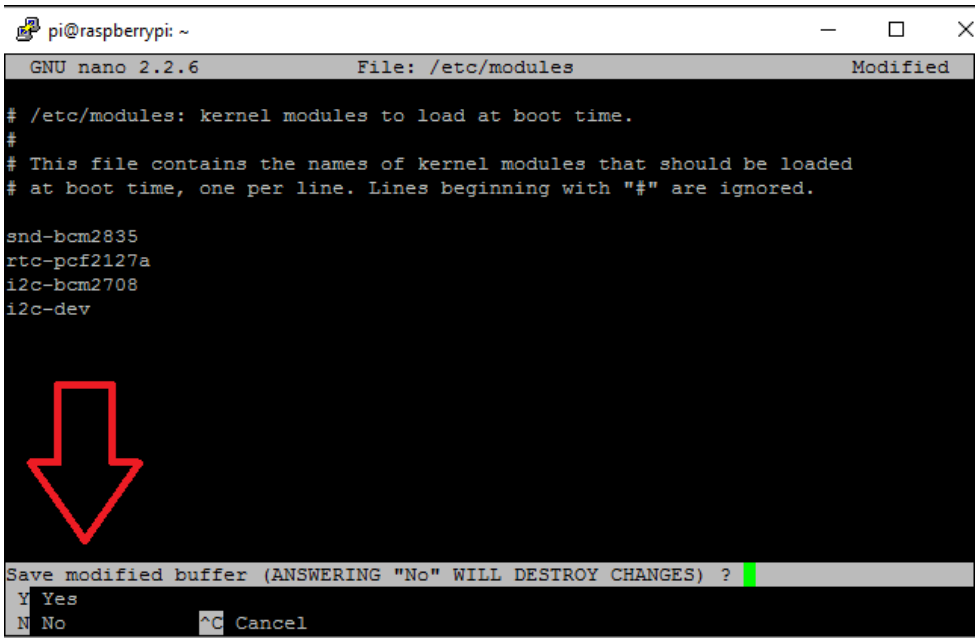
  sudo nano /etc/modules

- Add the following lines at the end of the file:

  i2c-bcm2708
  i2c-dev



- Please Note. There may, or may not, be other kernel modules present. For example, in this file we have “snd-bcm2835” and “rtc-pcf2127a”. These are are for other devices using different kernel modules. You should leave these present to ensure that all devices continue to function. **Ensure that each line is present only once.**
- To exit Nano press CRTL+X
- **Ensure that you save the file when you exit, by saying YES when it prompts**



- Please reboot your Raspberry Pi at this point:

  sudo reboot

- When the Pi re-starts, check that i2c is running by using the following command. The PiCo will list as a device in Line 60, Col 8:9:a:b

  sudo i2cdetect -y 1

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- UU -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 69 6a 6b -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi ~ $
```

- We now need to download the PiCo supporting files, available here. You can acess the files however you want, but this guide will download them directly onto the Pi. **Please Note. Commands are case sensitive!**

  http://www.pimodules.com/_zip/UPS_PIco_Supporting_Files.zip

- You can download it to your home directory using the following command:

  wget http://www.pimodules.com/_zip/UPS_PIco_Supporting_Files.zip

```
pi@raspberrypi ~ $ wget http://www.pimodules.com/_zip/UPS_PIco_Supporting_Files.zip
--2016-01-27 10:10:36--  http://www.pimodules.com/_zip/UPS_PIco_Supporting_Files.zip
Resolving www.pimodules.com (www.pimodules.com)... 74.208.89.193, 2607:f1c0:1000:2090:fad9:fde
0:a949:8821
Connecting to www.pimodules.com (www.pimodules.com)|74.208.89.193|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 307005 (300K) [application/zip]
Saving to: 'UPS_PIco_Supporting_Files.zip'

UPS_PIco_Supporting_Fil 100%[=============================>] 299.81K   704KB/s   in 0.4s

2016-01-27 10:10:37 (704 KB/s) - 'UPS_PIco_Supporting_Files.zip' saved [307005/307005]

pi@raspberrypi ~ $ ls
Desktop    Downloads  Pictures  python_games  UPS_PIco_Supporting_Files.zip
Documents  Music      Public    Templates     Videos
pi@raspberrypi ~ $
```

- If you run the list command ***ls***, you will now see the .zip folder listed in your home directory. This is show above.

- Now we need to unzip the folder:

  unzip UPS_PIco_Supporting_Files.zip

```
pi@raspberrypi ~ $ unzip UPS_PIco_Supporting_Files.zip
Archive:  UPS_PIco_Supporting_Files.zip
  inflating: pico_status.py
  inflating: picofssd.py
  inflating: picofu3.py
  inflating: rc.local
  inflating: UPS_PIco_0x58_18_01_2016.hex recent changes.pdf
pi@raspberrypi ~ $
```

- You can check that the folder unzipped correctly by using the ***ls*** command again.

```
pi@raspberrypi ~ $ ls
Desktop    picofssd.py     Public         UPS_PIco_0x58_18_01_2016.hex recent changes.pdf
Documents  picofu3.py      python_games   UPS_PIco_Supporting_Files.zip
Downloads  pico_status.py  rc.local       Videos
Music      Pictures        Templates
pi@raspberrypi ~ $
```

- The folder should contain the following files:
  - **picofssd.py** – PiCo run script
  - **picofu3.py –** Firmware update script
  - **pico_status.py** – Pico status monitor script
  - **rc.local** – Edited Pi "Run at Boot" script.
  - **UPS_PIco_0x58_18_01_2016.hex recent changes.pdf** – Revision change log doc.

- Now execute the PiCo run script and check for errors:

  sudo python /home/pi/picofssd.py

```
pi@raspberrypi ~ $ sudo python /home/pi/picofssd.py
```

- The UPS LED should start flashing on the UPS, twice per second to indicate that the UPS is working correctly. To exit the script, hit CTRL+C.

- Now that our UPS is working, we can set the script to run at boot! **There are two ways of doing this**. If you've already edited your rc.local for another device or programe you will need to edit the rc.local file manually. Otherwise, if you're using a clean install of your operating system you can simply use the rc.local file we downloaded earlier.

- To simply use the rc.local file we downloaded, you just need to move it to the /etc/ folder using the following command.

  sudo mv rc.local /etc/

```
pi@raspberrypi ~ $ sudo mv rc.local /etc/
pi@raspberrypi ~ $
```

- **Skip this step if you've completed above.** If you need to edit your rc.local file, you can do so using the following command.

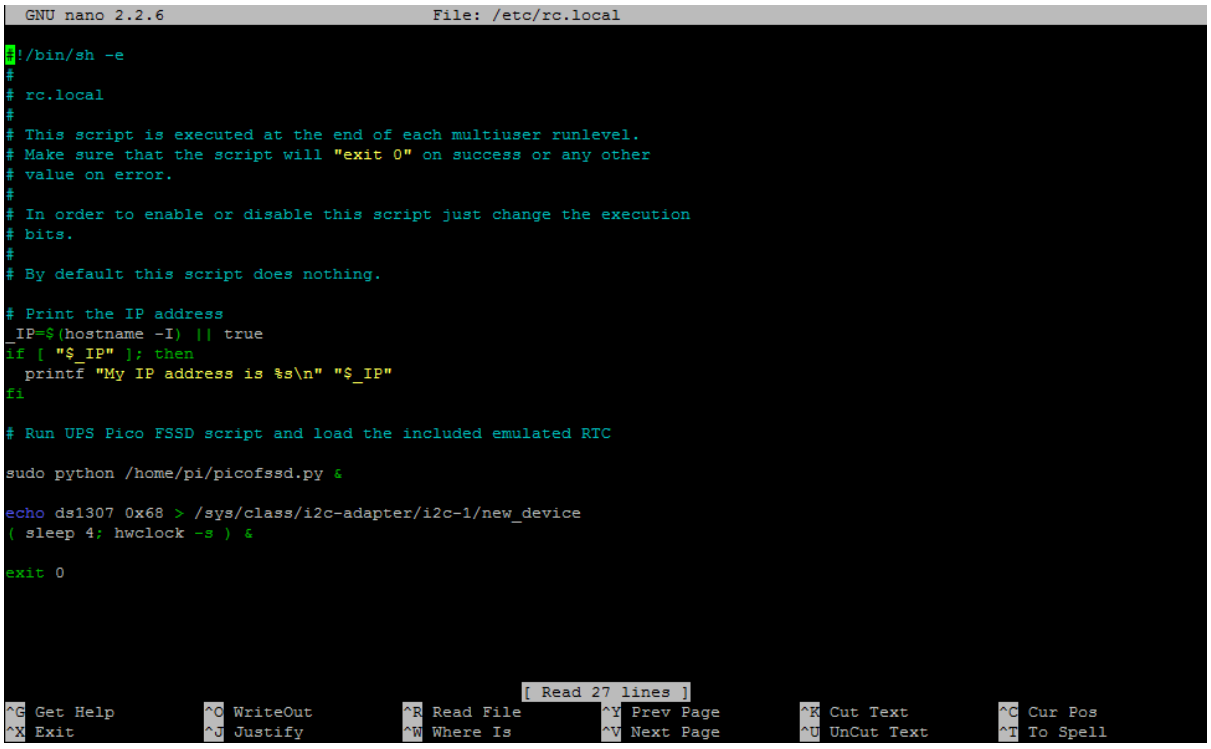
  sudo nano /etc/rc.local

- **Skip this step if you've completed above.** The PiCo needs you to add the following lines, before "exit 0"

  sudo python /home/pi/picofssd.py &

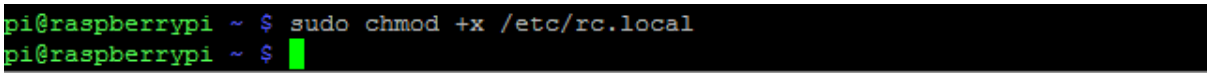
  echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
  (sleep 4; hwclock -s) &

- **Skip this step if you've completed above.** Ensure that you save the file before you exit



- Now we need to ensure that this file is set to be executable. So run the following command:

  sudo chmod +x /etc/rc.local



- Now restart your Raspberry Pi to ensure all software changes are made.

  sudo reboot

- When the system restarts, you should see the following UPS LED blinking process:
    - During boot – LED blinks very fast (200ms)
    - After boot – LED blinks normal (500ms) which indicates that the Raspberry Pi is recognised and the UPS is working correctly.
    - The CHG LED might also light continuously, to indicate that the battery is charging. If this LED is off, it means that the battery is fully charged.
- To test the device - Remove the power supply from the Raspberry Pi (just unplug it). The Pi should stay active, and the UPS LED should flash slower at once per second.

- If your reconnect the power, the UPS should automatically revert back to standby mode.

- If you wait 30 seconds the Pi will shut down. Once complete, the UPS LED will flash (twice per second) for 2 - 3 seconds. After this, all LEDs on the Pi & UPS will turn off.

- When power is restored, e.g. by plugging the power supply back in, the Raspberry Pi should boot again.

- The UPS Pico is now set up and ready to use! We recommend that all users check that the Pico firmware is up to date – There is a guide below under **UPS PIco Updating Firmware.** There are also some basic user commands that can be accessed, which are listed below under **Using the UPS Pico**.

Using the UPS Pico

Removal

If you need to remove the PiCo for any reason, ensure that the Raspberry Pi is switched off, completely disconnected from power, and the PiCo is switched off before removal.

- To shutdown the Pi cleanly, use the following command:

  sudo shutdown -h now

- Once the Pi has shutdown, the UPS PiCo status light will start to flash quickly.
- Make sure you then remove power from your Pi by turning off the Pi at the plug socket.
- If you leave the PiCo at this point, it will turn itself off after 30 seconds.
- If you want to interrupt the wait period, simply press the UPSR button on the PiCo. This will completely turn off the PiCo.

Status Check

To check various status features of the PiCo, you can use the pico_status.py script. The scripts is downloaded as part of the supporting files .zip we downloaded during installation.

- To run, simply type in the following command:

  sudo python pico_status.py

```
pi@raspberrypi ~ $ sudo python pico_status.py

       pico status V1.0
*********************************
  UPS PIco Firmware: 59
  Powering Mode: RPi
  BAT Volatge: 3.97 V
  RPi Voltage: 4.96 V
  SOT23 Temperature: 21 C
  TO-92 Temperature: 00 C
  A/D1 Voltage: 0.0 V
  A/D2 Voltage: 0.0 V
*********************************

pi@raspberrypi ~ $
```

- This will list various status aspects of the PiCo.

Battery Run Time

The PiCo has a factory set battery run time of 30 seconds. This is the time between your Pi losing power, and the PiCo safely shutting the Pi down into low power mode. This value can be edited using the following command. Please Note. There is a minimum run time of 15 seconds.

You can set this value to anything you want by changing the red number in the following command e.g. for 30 seconds:

i2cset -y 1 0x6b 9 30

Or for 120 seconds:

i2cset -y 1 0x6b 9 120

```
pi@raspberrypi ~ $ i2cset -y 1 0x6b 9 120
pi@raspberrypi ~ $
```

To set the run time to maximum e.g. the run time of the batttery, use the following command. This will keep the Pi powered until the battery discharges to 3.5V, at which point it will shut the Pi down.

i2cset -y 1 0x6b 9 0xff

UPS PIco Updating Firmware

The UPS PiCo features an embedded serial bootloader which allows users to manually update the unit's firmware. Bootloader is small piece of firmware stored permanently in the micro controller flash memory of the UPS Pico, located in the special protected area. It can not be erased by user without dedicated hardware tools. In order to upload new firmware, an invokation of the bootloader routine is needed. It can be done manually or automatically if I2C-tools is running and installed.

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

The firmware can be uploaded using a dedicated python script, called **picofu.py**. This file may change with firmware revisions. At the writing of this guide, the current firmware is **Version 59** and is updated using **picofu3.py.**

The latest and most up to date firmware is always available from the following link.

http://www.pimodules.com/_zip/UPS_PIco_Firmware_Update.zip

The PiCo supporting files (including picofu.py) are always available from the following link.

http://www.pimodules.com/_zip/UPS_PIco_Supporting_Files.zip

It is mandatory to have previously installed python and I2C-tools on the Raspberry Pi. You will install these during initial PiCo setup outlined previously in this document.

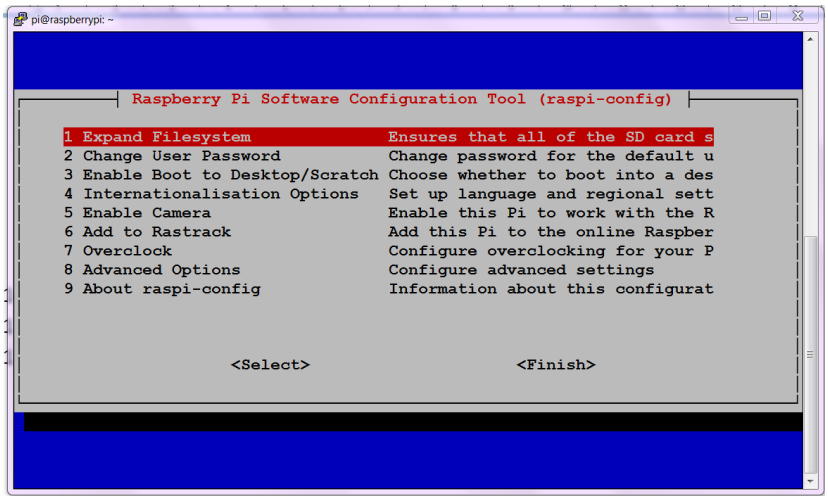
As the bootloader uses the Raspberry Pi Serial Port (RS232), it is mandatory to have it free on the Raspberry Pi (without any hardware occupying it). It is also important that you ensure that there is no software using it. If minicom has been used, please restart the Raspberry Pi, as minicom keeps the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

To free the serial port, we need to disable the login shell in configuration.

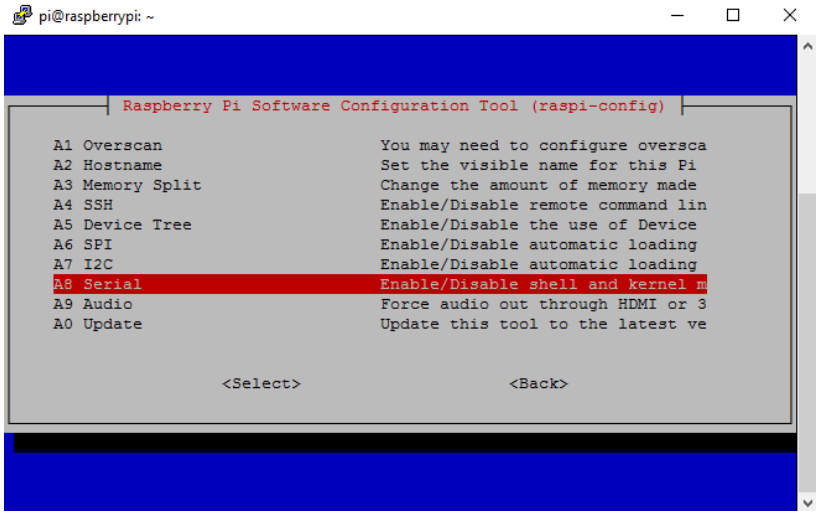
- From terminal, enter the Raspberry Pi configuration menu using the following command:

  sudo raspi-config

- Scroll to **Advanced Options**



- Then scroll to A8 Serial



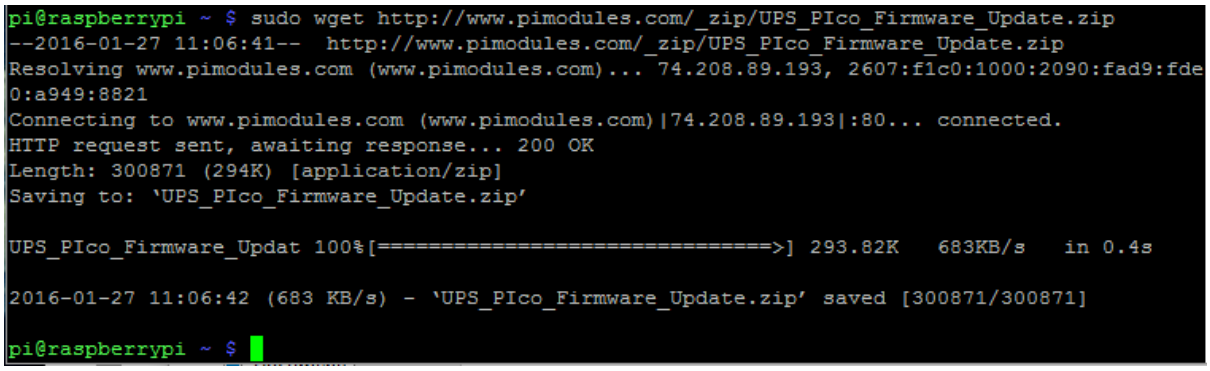
- Ensure set it to disabled.

- Now restart your Raspberry Pi to ensure all changes are made. The configuration should prompt a reboot anyway.

  sudo reboot

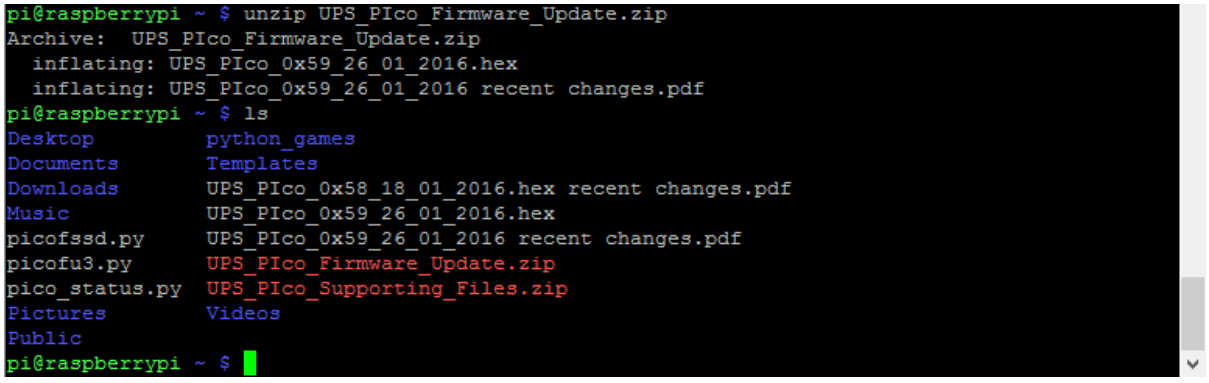
- Now download the UPS PiCo firmware update python script by using the following command.

  sudo wget http://www.pimodules.com/_zip/UPS_PIco_Firmware_Update.zip



- Unzip the file we just downloaded:

  unzip UPS_PIco_Firmware_Update.zip



- This zip folder will contain two files:
  - **UPS_PIco_0x59_26_01_2016.hex –** Firmware update hex file.
  - **UPS_PIco_0x59_26_01_2016 recent changes.pdf** - Revision change log doc.

- **Please Note. The UPS PiCo is actively worked on and regularly updated. It is likely that the hex revision will change, so please use the correct filename when updating.**

- **The bootloader can be initiated in two ways. The automatic initiation should always be used first. Manual should only be used in the event of an automatic initiation failure.**

Automatic Initiation

The bootloader is invoked by running the following command line

sudo i2cset -y 1 0x6b 0x00 0xff && sudo python picofu3.py -v -f UPS_PIco.hex

The **UPS_PIco.hex** should replaced with the name of the last firmware update, or the firmware you wish to use - Please remember that linux is case sensitive. The ***picofu3.py*** part should be replaced with the latest version of the picofu.py file.

Using the firmware and picofu file outlined in this document, that we just downloaded, you would run the following command:

sudo i2cset -y 1 0x6b 0x00 0xff && sudo python picofu3.py -v -f UPS_PIco_0x59_26_01_2016.hex

When firmware starts the upload procedure, the BIG BLUE LED will blink very fast, and the BIG RED LED with turn off.

```
pi@raspberrypi ~ $ sudo i2cset -y 1 0x6b 0x00 0xff  &&  sudo python picofu3.py -v -f UPS_PIco_0x59_26_01_2016.hex
Validating firmware: OK
Checking communication with bootloader: OK
Uploading firmware: 0% ..... 4.0% .....
..... 9.0% .....
..... 14.0% .....
..... 19.0% ..... 23.0% .....
..... 28.0% .....
..... 33.0% .....
..... 38.0% ..... 43.0% .....
..... 47.0% .....
..... 52.0% .....
..... 57.0% ..... 62.0% .....
..... 67.0% .....
..... 71.0% .....
..... 76.0% ..... 81.0% .....
..... 86.0% .....
..... 91.0% .....
... 95.0% ..... Done uploading...
Invoking factory reset of PIco...
ALL Done :) Ready to go...
pi@raspberrypi ~ $
```

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

We would recommend that you now shutdown your Pi and UPS PiCo completely in order to ensure that all changes are integrated.

- To shutdown cleanly, use the following command:

  sudo shutdown -h now

- Once the Pi has shutdown, the UPS PiCo status light will start to flash quickly.
- Make sure you then remove power from your Pi by turning off the Pi at the plug socket.
- If you leave the PiCo at this point, it will turn itself off after 30 seconds.
- If you want to interrupt the wait period, simply press the UPSR button on the PiCo. This will completely turn off the PiCo.

- Once you've rebooted your system, you can check the UPS PiCo firmware version using the following command:

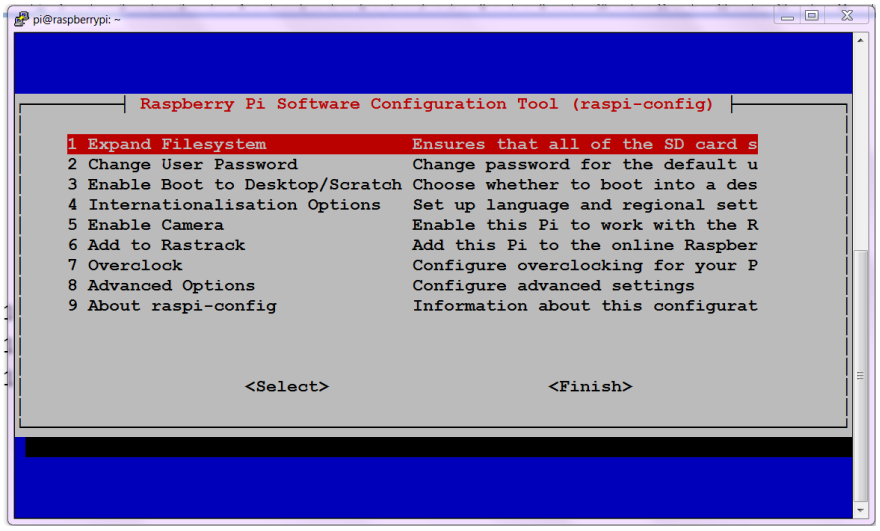
sudo i2cget -y 1 0x6b 0x00

```
pi@raspberrypi ~ $ sudo i2cget -y 1 0x6b 0x00
0x59
pi@raspberrypi ~ $
```

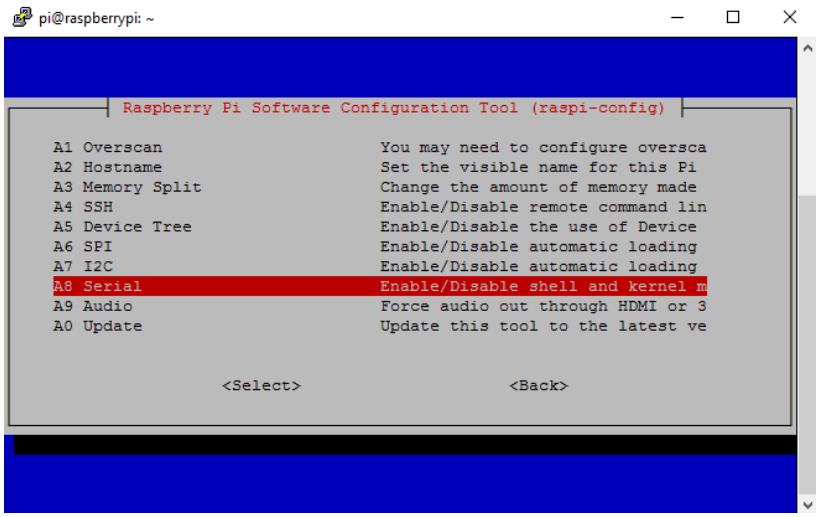
- In this case the system should output 0x59, signifying that the firmware has updated correctly.
- We now need to re-enable serial, so enter the Pi configuration menu.

sudo raspi-config

- Scroll to **Advanced Options**



- Then scroll to A8 Serial, and ensure this is enabled.



- Ensure that this is enabled.

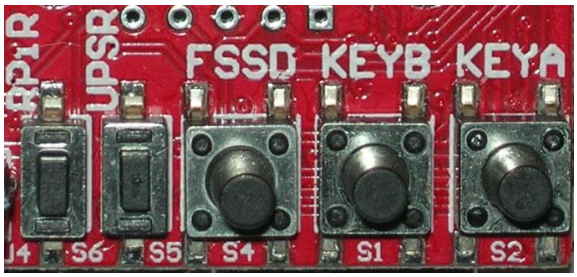
- **If the system requests a reboot, please do so.**

Manual Initiation

The UPS PiCo has the ability to invoke the bootloader manually, via the on board buttons. You can do this instead of using the automatic initiation outlined above. **The manual method should only be used if the automatic method fails for some reason.**

The following procedure needs to be followed:

- Press and hold the **UPSR** button
- Continue to hold the **UPSR** button, and press and hold the **KEYA** button.
- Release the **UPSR** button, but keep holding the **KEYA** button
- Release the **KEYA** button
- The big RED LED will light, and system wil be able to receive the firmware update



- Then write the following command on the Raspbeery Pi command line

sudo python picofu3.py -v -f UPS_PIco.hex

The **UPS_PIco.hex** should replaced with the name of the last firmware update, or the firmware you wish to use - Please remember that linux is case sensitive. The ***picofu3.py*** part should be replaced with the latest version of the picofu.py file.

Using the firmware and picofu file outlined in this document, that we just downloaded, you would run the following command:

sudo python picofu3.py -v -f UPS_PIco_0x59_26_01_2016.hex

When firmware starts the upload procedure, the BIG BLUE LED will blink very fast, and the BIG RED LED with turn off.

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

Then follow on from the instructions above.

Further Information

The full user guide is available here, this details further functions of the UPS PiCo:

http://www.pimodules.com/_pdf/UPS_PIco_Manual.pdf