

This tutorial assumes you've already set up a Raspberry Pi with Raspbian12(bookworm). For help installing the Debian-based OS on your Pi, check out the docs on [Raspberrypi.org](https://www.raspberrypi.org).

## 1.0 Editing the EEPROM configuration

1.1 From the command line or Terminal window start by running the following command :

```
pi@raspberrypi ~ $ sudo rpi-eeprom-config -e
```

Change the setting of **POWER\_OFF\_ON\_HALT** from **0** to **1** ,  
Add **PSU\_MAX\_CURRENT=5000** at the end of the file that reads like this:

```
GNU nano 7.2 /tmp/tmpf9i3trrp/boot.conf
[all]
BOOT_UART=1
POWER_OFF_ON_HALT=1
BOOT_ORDER=0xf41
PSU_MAX_CURRENT=5000
```

1.2 Reboot your Raspberry Pi 5 to make the change take effect.

```
pi@raspberrypi ~ $ sudo reboot
```

## 2.0 Configuring the Raspberry Pi for I2C

2.1 From the command line or Terminal window start by running the following command :

```
pi@raspberrypi ~ $ sudo raspi-config
```

This will launch the raspi-config utility. Select "Interfacing Options"

```
Raspberry Pi Software Configuration Tool (raspi-config)

 1 System Options          Configure system settings
 2 Display Options        Configure display settings
 3 Interface Options       Configure connections to peripherals
 4 Performance Options    Configure performance settings
 5 Localisation Options   Configure language and regional settings
 6 Advanced Options       Configure advanced settings
 8 Update                 Update this tool to the latest version
 9 About raspi-config     Information about this configuration tool

<Select>                                <Finish>
```

2.2 Highlight the "I2C" option and activate "<Select>"

## Raspberry Pi Software Configuration Tool (raspi-config)

```
I1 SSH          Enable/disable remote command line access using SSH
I2 VNC          Enable/disable graphical remote desktop access
I3 SPI          Enable/disable automatic loading of SPI kernel module
I4 I2C          Enable/disable automatic loading of I2C kernel module
I5 Serial Port Enable/disable shell messages on the serial connection
I6 1-Wire       Enable/disable one-wire interface
I7 Remote GPIO Enable/disable remote access to GPIO pins
```

<Select>

<Back>

### 2.3 Select and activate "<Yes>"

Would you like the ARM I2C interface to be enabled?

<Yes>

<No>

### 2.4 Highlight and activate "<Ok>"

The ARM I2C interface is enabled

<Ok>

2.5 Activate "<Finish>", this will exist the raspi-config utility.

```
Raspberry Pi Software Configuration Tool (raspi-config)
1 System Options      Configure system settings
2 Display Options    Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options   Configure advanced settings
8 Update             Update this tool to the latest version
9 About raspi-config Information about this configuration tool

<Select>                                <Finish>
```

2.6 Update your Pi and Install the required dependencies

```
pi@raspberrypi ~ $ sudo apt-get update
```

```
pi@raspberrypi ~ $ sudo apt-get upgrade -y
```

```
pi@raspberrypi ~ $ sudo apt-get install python3-pip
```

2.7 Download the required scripts

```
pi@raspberrypi ~ $ git clone https://github.com/suptronics/x120x.git
```

2.8 Reboot the Raspberry Pi

```
pi@raspberrypi ~ $ sudo reboot
```

The Raspberry Pi will reboot and the interface will be enabled.

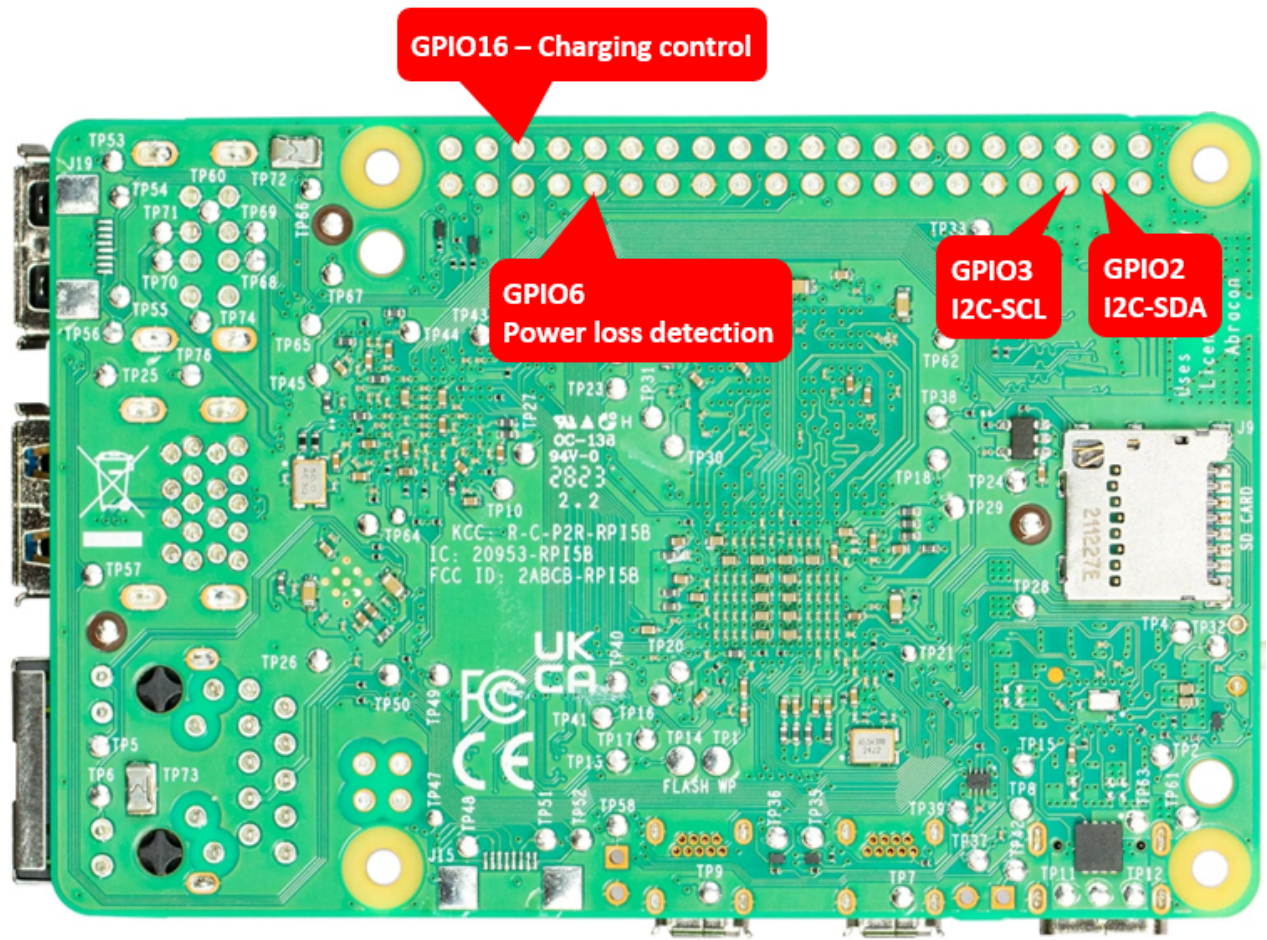
2.9 Once you have logged into your Raspberry Pi from the command line, run the command to see all the connected devices

```
pi@raspberrypi ~ $ 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:  --  --  --  --  --  36  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

#36 - the address of the battery fuel gauging chip

\*The UPS connects to the Pi GPIO via pogo pins, If the I2C address (0x36) is not detected, Please clear the leads of pin 3 and pin 5 on the GPIO header from the bottom of the Raspberry Pi PCB and install again.



### 3.0 Reading UPS status

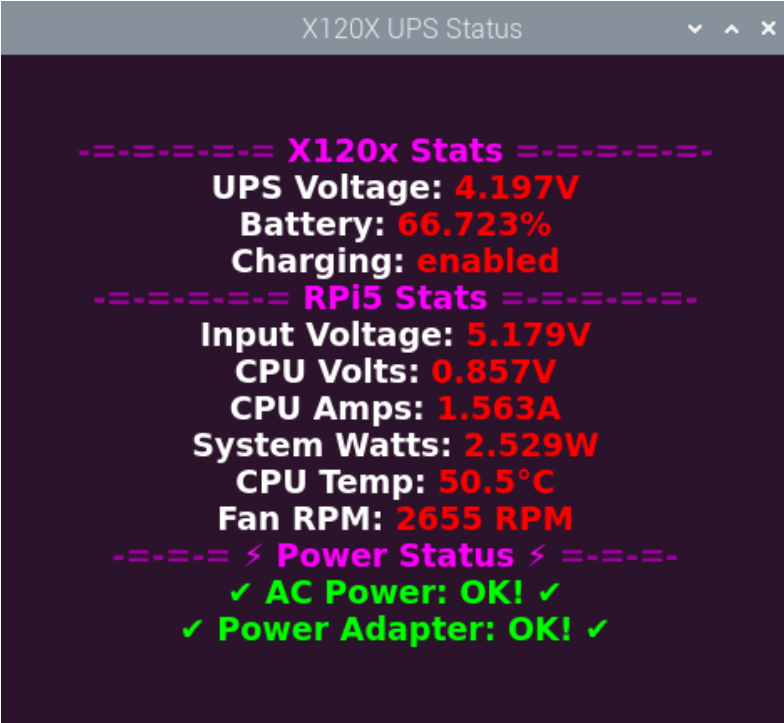
Command line for reading UPS status

```
pi@raspberrypi:~/x120x $ sudo python3 merged.py
```

```
pi@raspberrypi:~ $ sudo python3 merged.py
Capacity: 74.05% (Full), AC Power State: Plugged in, Voltage: 4.20V
```

Desktop interface for reading UPS status

```
pi@raspberrypi:~/x120x $ sudo python3 qtx120x.py
```



```
X120X UPS Status
----- X120x Stats -----
UPS Voltage: 4.197V
Battery: 66.723%
Charging: enabled
----- RPi5 Stats -----
Input Voltage: 5.179V
CPU Volts: 0.857V
CPU Amps: 1.563A
System Watts: 2.529W
CPU Temp: 50.5°C
Fan RPM: 2655 RPM
----- ⚡ Power Status ⚡ -----
✓ AC Power: OK! ✓
✓ Power Adapter: OK! ✓
```

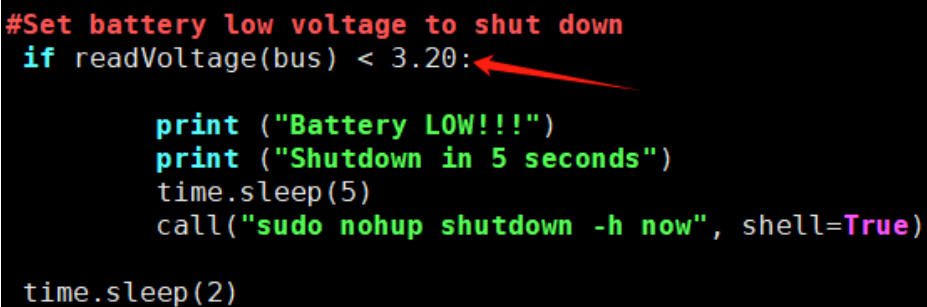
#### 4.0 Manual reading battery voltage & percentage

4.1 Run the command to read battery voltage and percentage

```
pi@raspberrypi:~/x120x $ sudo python3 bat.py
```

4.2 Change the battery low voltage to implement safe shutdown. default is less than 3.20Vdc.

```
pi@raspberrypi:~/x120x $ sudo nano bat.py
```



```
#Set battery low voltage to shut down
if readVoltage(bus) < 3.20:
    print ("Battery LOW!!!")
    print ("Shutdown in 5 seconds")
    time.sleep(5)
    call("sudo nohup shutdown -h now", shell=True)

time.sleep(2)
```

Note: the voltage range must be 3.00~4.10vdc.

4.3 Save and exit. In nano, you do that by hitting CTRL + X, answering Y and hitting Enter when prompted.

#### 5.0 Manua testing AC Power loss or power adapter failure detection (PLD)

5.1 Execute the command to initiate power loss detection (PLD)

```
pi@raspberrypi:~/x120x $ sudo python3 pld.py
```

Disconnect the power adapter and observe the detection of the power loss.

## 6.0 Control battery charging - for advanced users only

6.1 To disable battery charging

```
pi@raspberrypi ~ $ pinctrl set 16 op dh
```

6.2 To enable battery charging

```
pi@raspberrypi ~ $ pinctrl set 16 op dl
```