RasPi-EVSE User Guide

1 OVERVIEW

The RasPi-EVSE HAT provides it various interfaces for its functions. This user guide briefly describes these interfaces and its associated function. Where applicable some short usage examples are laid out as well.

Figure 1 shows an overview of the circuit board with the various components grouped into function blocks.

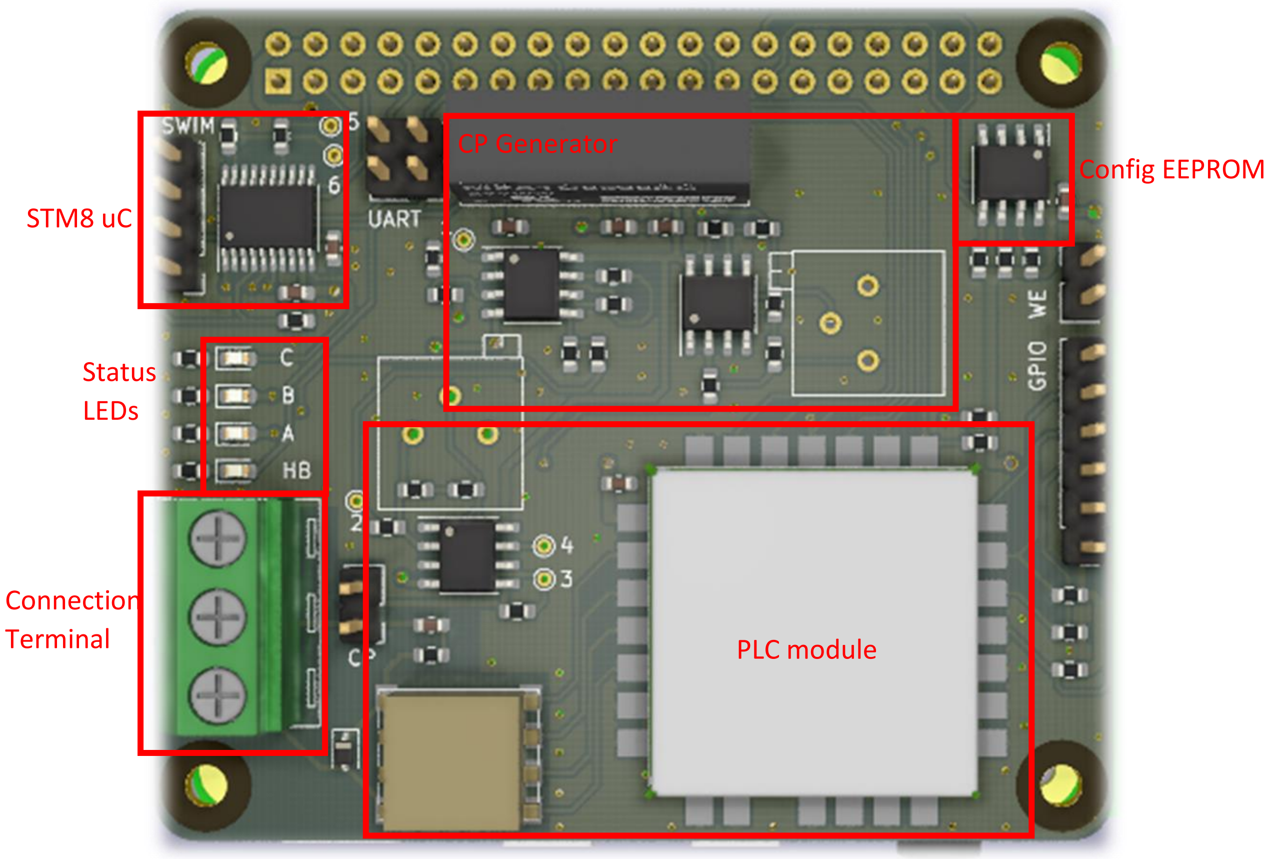

Figure 1 Overview

2 CONTENTS

3 Setup

The Raspi-EVSE HAT adheres to the Raspberry Pi Foundations HAT specification and includes a configuration EEPROM, therefore loading the relevant drivers automatically when you boot up the Raspbian OS.

Specifically, this includes three drivers:

1. I^{2}C Master Driver for I^{2}C 1
2. QCA700x Driver, most likely as eth1
3. Heartbeat LED

3.1 Checking I^{2}C

The I^{2}C driver should show up in **`/dev`** folder as **`/dev/i2c-1`**, as can be checked for example the following way

```
pi@babspi:~ $ ls /dev | grep i2c
i2c-1
```

If this works, you can also check that the STM8 Co-Processor on the RasPi-EVSE HAT also works correctly by scanning the I^{2}C bus for devices. Then, this should show up with the I^{2}C device address 0x60. This can be checked for example with the **`i2cdetect`** command the following way with:

```
pi@babspi:~ $ 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: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

3.2 Checking for QCA PLC

The QCA PLC device driver registers a network interface driver, most commonly as eth1.

This can be checked for example with the **ifconfig** command:

```
pi@babspi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether dc:a6:32:b3:ee:a3  txqueuelen 1000  (Ethernet)
        RX packets 38  bytes 5856 (5.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 73  bytes 11281 (11.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.83.156  netmask 255.255.0.0  broadcast 169.254.255.255
        inet6 fe80::f0b8:c760:529c:f0f9  prefixlen 64  scopeid 0x20<link>
        ether f2:75:e0:11:79:5c  txqueuelen 100  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 201  bytes 35579 (34.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

In order to check that eth1 is the correct device and the driver can successfully communicate with the physical chip you can use the plctool command from the open-plc-utils (for more information see chapter Open-PLC-Utils).

```
pi@babspi:~/open-plc-utils/plc $ sudo ./plctool -r -i eth1
eth1 00:B0:52:00:00:01 Request Version Information
eth1 00:01:87:10:EF:93 QCA7000 MAC-QCA7000-1.2.5.3207-00-20180927-CS
```

4 STATUS LEDS

The RasPi-EVSE HAT provides 4 status LEDs marked below and described in detail in the following chapters.

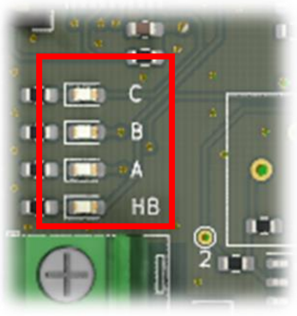

Figure 2 Status LEDs

4.1 HB LED

This LED is connected to the Raspberry Pi PIN 15 (GPIO22) directly and by default automatically configured on bootup via the configuration EEPROM as heartbeat (HB). Therefore, it should flash twice quickly about every second in red.

You can reconfigure the LED temporarily via **/sys/class/leds/raspi_evse_hb** or permanently via configuration EEPROM update.

```
pi@babspi:/sys/class/leds/raspi_evse_hb $ cat trigger
none rc-feedback kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kbd-
shiftlock kbd-altgrlock kbd-ctrllock kbd-altlock kbd-shiftllock kbd-
shiftrlock kbd-ctrlllock kbd-ctrlrlock timer oneshot [heartbeat] backlight
```

```
gpio cpu cpu0 cpu1 cpu2 cpu3 default-on input panic actpwr mmc1 mmc0
rfkill-any rfkill-none rfkill0 rfkill1
```

4.2 A LED

This LED is connected to the STM8 Co-Processor and lights up permanently in **red** whenever the CP status is in state A. This signifies that the RasPi-EVSE HAT has not detected a connected vehicle.

4.3 B LED

This LED is connected to the STM8 Co-Processor and lights up permanently in **orange** whenever the CP status is in state B. This signifies that the RasPi-EVSE HAT has detected a vehicle connected but the vehicle has not commanded to start the charging (yet).

4.4 C LED

This LED is also connected to the STM8 Co-Processor directly and lights up permanently in **green** whenever the RasPi-EVSE HAT has detected that the connected vehicle is connected and commands to start delivering power for charging to the RasPi-EVSE HAT.

5 JUMPERS

5.1 UART

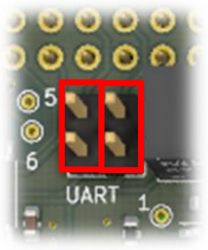

Figure 3 UART Jumper

These two Jumpers allow the UART lines from the Raspberry Pi to be connected to the STM8 Co-Processor. Currently this is not being used by the application on the STM8 but can easily be programmed for future function extension.

You could also easily connect an UART to USB bridge here in order to access the Raspberry Pi via serial terminal.

5.2 CP (Control Pilot)

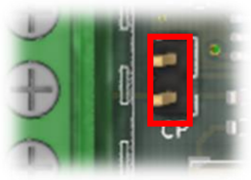

Figure 4 CP Jumper

This jumper allows the CP circuit to be connected or disconnected. For example, you should remove this jumper when you want to sniff on an existing charging station in order to not have the CP generation of the RasPi-EVSE HAT interfere with the CP generation of the real charging station.

For all other use cases this Jumper should be set in order to have the RasPi-EVSE HAT generate the CP signal in order to have the car detect the RasPi-EVSE HAT as a charging station.

5.3 WE (Write Enable)

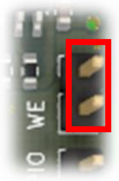

Figure 5 WE Jumper

This jumper, when connected, enables writing to the configuration EEPROM from the Raspberry PI. If removed, the Raspberry Pi cannot write a new configuration to the configuration EEPROM. This should only be connected when you want to update configuration EEPROM with a custom configuration.

Instructions on how to do this can be found here:
https://github.com/raspberrypi/hats/tree/master/eepromutils

6 PIN HEADERS

6.1 SWIM

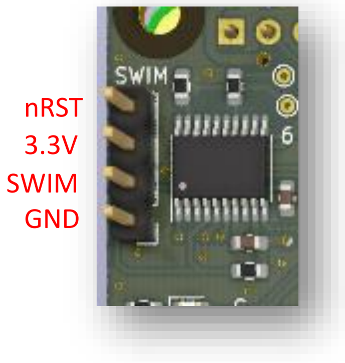

Figure 6 SWIM Header

This pin header is for programming the STM8 Co-Processor via the SWIM protocol. Cheap ST-Link V2 USB dongles are plentiful available on Amazon, eBay, etc and is not included with the RasPi-EVSE HAT.

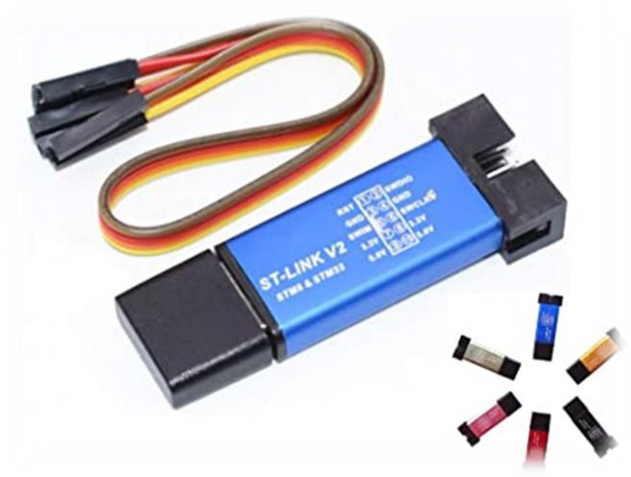

Figure 7 ST-LINK USB Dongle

When the RasPi-EVSE HAT is connected to the Raspberry Pi it is suggested to not connect the 3.3V Pin because the Raspberry Pi is already providing this voltage or the programmer could back feed the Raspberry Pi which is not recommended.

Either connect 3.3V to the USB programmer dongle when the RasPi-EVSE HAT is not connected to the Raspberry Pi, or disconnect the 3.3V Pin from the USB programmer and let the Raspberry Pi provide the 3.3V power to the STM8 Co-Processor.

6.2 GPIO

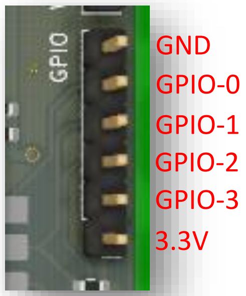

Figure 8 GPIO Header

The GPIO pin header provides access to the GPIO pins on the PLC chip. In addition, it provides GND and 3.3V.

The PLC module is connected with pull-up and pull-down resistor in order to bootstrap the PLC chip as seen in Figure 9 GPIO Connections. Detailed description of the functions can be found in the Datasheet of the PLC module.

https://in-tech-smartcharging.com/products/powerline-communication-modules/plc-stamp-micro-2

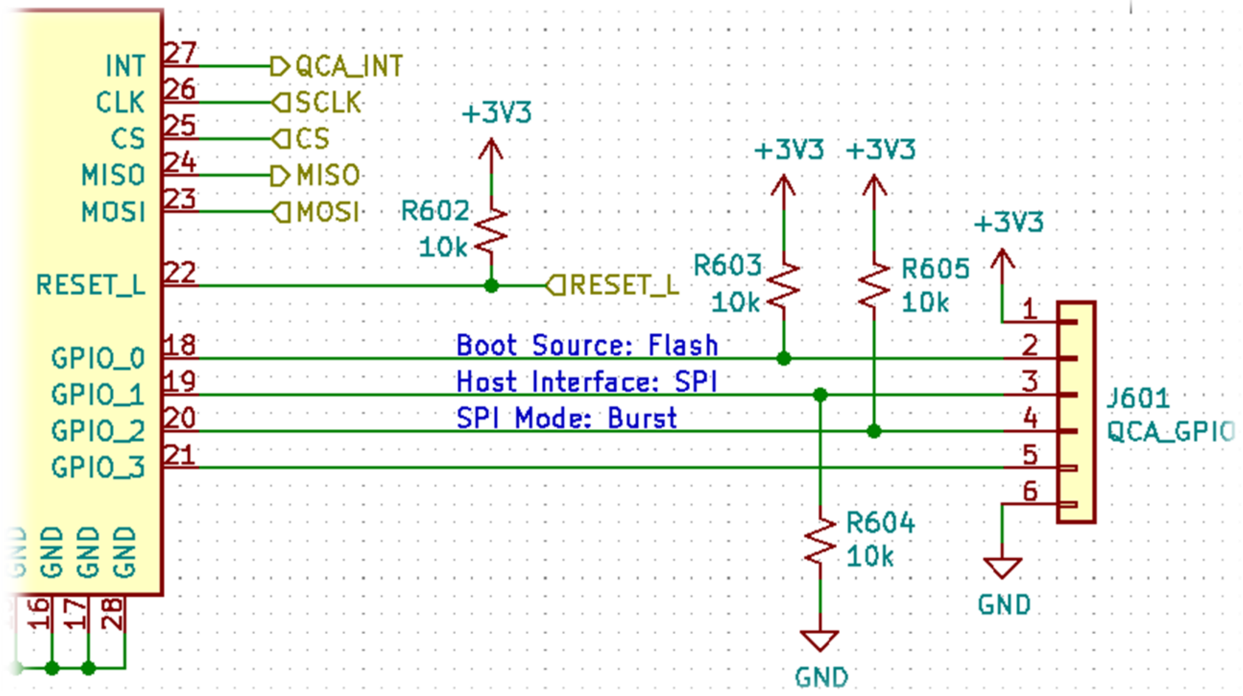

Figure 9 GPIO Connections

7 Raspberry Pi Header

Figure 10 Raspberry Pi Header

In order to provide additional functionality, for example switching relays or communicating with other SPI or I2C devices, the RasPi-EVSE HAT provides a stackable pin header.

Figure 11 Used Pins, gives a brief overview over which pins of the Raspberry Pi are being used by the RasPi-EVSE HAT. The following table lists the detailed usage of the Raspberry Pi 40-pin header:

Used Raspberry Pi Pin	RasPi-EVSE HAT Function	Description
GPIO-0	EEPROM SDA	Config EEPROM
GPIO-1	EEPROM SCL	Config EEPROM
GPIO-2	STM8 I2C SDA	STM8 Connection
GPIO-3	STM8 I2C SCL	STM8 Connection
GPIO-8	SPI-CS	PLC SPI
GPIO-9	MISO	PLC SPI
GPIO-10	MOSI	PLC SPI
GPIO-11	SCLK	PLC SPI
GPIO-23	PLC-INT	PLC Interrupt
GPIO-24	PLC-RESET	PLC Reset

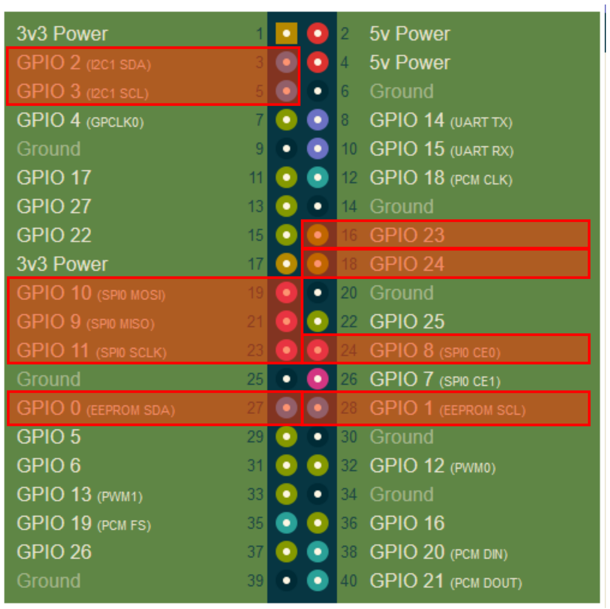

Figure 11 Used Pins

8 Screw Terminals

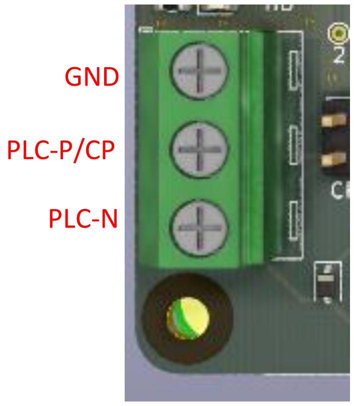

Figure 12 Screw Terminals

The screw terminals are the connection to the electric vehicle or charging station as follows:

- The **GND** terminal should be connected to the PE line or chassis of the electric vehicle.
- The **PLC-P/CP** terminal should be connected to the CP line of the plug going to the vehicle's charging socket. It usually should be internally connected to the CP generator circuit via jumper 'CP' as described in chapter CP, in order to simulate a charging station.
- The **PLC-N** terminal should be connected to GND or PE in order to simulate a DC charging station that provides PLC communication according to DIN SPEC 70121 or ISO 15118. When the RasPi-EVSE HAT is acting as a AC charging station without PLC communication this terminal can be left unconnected.

Figure 13 Terminal Connection shows the internal connection of the screw terminals for further clarification.

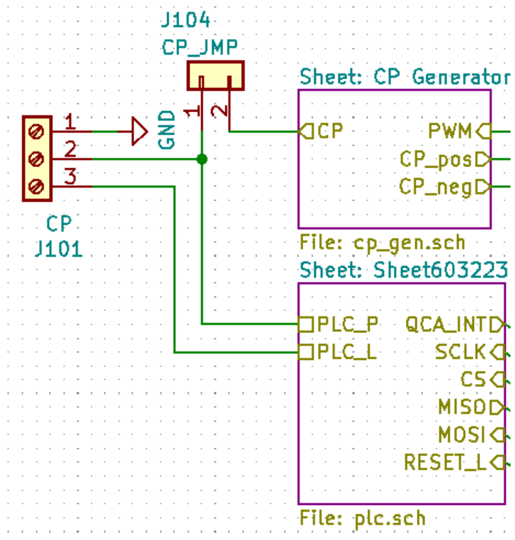

Figure 13 Terminal Connection

9 STM8 CO-PROCESSOR I^{2}C PROTOCOL

The STM8 operates at a clock frequency of **10kHz** and its I^{2}C address is **0x60**.

It provides the following functions via registers:

- Setting CP line's PWM percentage
- Reading CP line's negative voltage level
- Reading CP line's positive voltage level

9.1 REGISTER MAP

Name	Address	Read/ Write	Reset Value	Type	Description
adc_pos	0x00	Read	-	int16	Contains the raw adc value of the CP's positive period
adc_neg	0x02	Read	-	int16	Contains the raw adc value of the CP's negative period
pwm_percent	0x04	Write	0	uint8	Sets the CP lines PWM Percentage. Valid values are 0-100

9.2 ADC VOLTAGE EXAMPLE

In order to calculate the measured voltage value from the raw adc value of I^{2}C registers adc_pos and adc_neg, the following formula should be applied:

$$voltage = adc * \frac{13.2}{1024}$$

This is valid also for the negative values because the adc value is of type signed integer.

These registers can be read or written, for example, via command line functions **i2cget** as demonstrated here:

```
pi@babspi:~ $ i2cget -y 1 0x60 2 w
0x57fc
```

This command is reading from the STM8 (address 0x60) and the register address 2 (adc_neg) two bytes (w) and returns a value of 0x57FC. This corresponds to the signed integer value of 0xFC57 or -937 decimal. According to the above formula this gives a voltage level of roughly -12.07.

9.3 PWM EXAMPLE

The PWM percentage can also be set via command line function i2cset, as demonstrated like this:

```
pi@babspi:~ $ sudo i2cset -y 1 0x60 4 80
```

This command is setting the STM8 (address 0x60) and register address 4 (pwm_percentage) to the value of 80 decimal. This relates to a 80% low and 20% high PWM (between +/-12V).

10 Open-PLC-Utils

The PLC chip of the RasPi-EVSE HAT has a number of functions and configuration parameters that unfortunately are not well documented and neither is an official datasheet provided from the chip manufacturer.

Fortunately, there is an open-source project that gives you access to most of the functionality: open-plc-utils.

Therefore, in order to control the PLC chip you can just use this Github repository that provides access to most of the functions and configuration of the PLC chip via small command line utilities or program your own software based on the source code.

https://github.com/qca/open-plc-utils

This can be easily installed on a Raspberry Pi by following the build instructions in the README.

All commands interact with the PLC chip via the ethernet interface provided by the driver. For example, the command line utility of the open-plc-utils repository for checking the connection to the PLC chip and the firmware version is the following:

```
pi@babspi:~/open-plc-utils/plc $ sudo ./plctool -r -i eth1
eth1 00:B0:52:00:00:01 Request Version Information
eth1 00:01:87:10:EF:93 QCA7000 MAC-QCA7000-1.2.5.3207-00-20180927-CS
```

11 Node-Red example application

In order to easily control the CP lines and react on various conditions node red can be conveniently used. Node-Red is easily installed on the Raspberry Pi by following instructions of the Node-Red Project for Raspberry Pi: https://nodered.org/docs/getting-started/raspberrypi

The following example application show how to set the PWM percentage and read back the measured voltage.

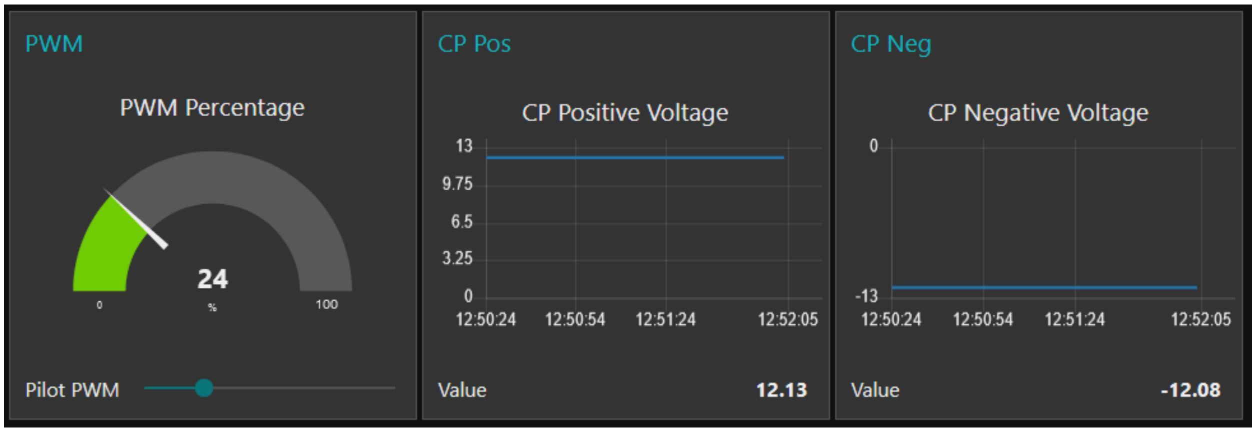

Figure 14 Node Red EVSE Dashboard

Something like this can be created with a simple node-red flow:

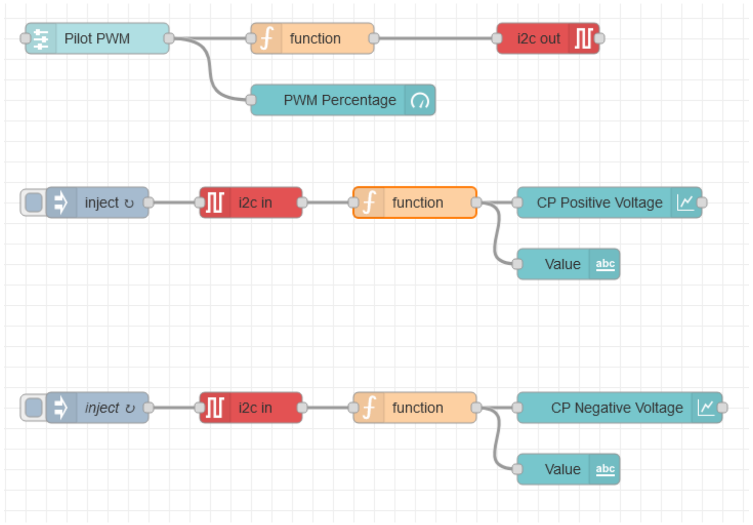

Figure 15 Node Red EVSE Flow

11.1 Demo Flow

This flow can be imported into node red via copy and paste of the following code:

```
[{"id":"42a191ae.3209e","type":"tab","label":"Flow
1","disabled":false,"info":""},{"id":"7e3dbf32.43b8b","type":"i2c
scan","z":"42a191ae.3209e","name":"","busno":"1","x":380,"y":100,"wires":[[
"e18bf3fc.2a0558"],["f5f71dcf.75b21"]]},{"id":"e18bf3fc.2a0558","type":"deb
ug","z":"42a191ae.3209e","name":"","active":true,"tosidebar":true,"console"
:false,"tostatus":false,"complete":"false","statusVal":"","statusType":"aut
o","x":590,"y":60,"wires":[]},{"id":"f5f71dcf.75b21","type":"debug","z":"42
a191ae.3209e","name":"","active":true,"tosidebar":true,"console":false,"tos
tatus":false,"complete":"false","statusVal":"","statusType":"auto","x":590,
"y":140,"wires":[]},{"id":"36675026.e229b","type":"inject","z":"42a191ae.32
09e","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat":
"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","paylo
adType":"date","x":200,"y":100,"wires":[["7e3dbf32.43b8b"]]},{"id":"7db7417
b.1c1208","type":"i2c
out","z":"42a191ae.3209e","name":"","busno":"1","address":"96","command":"4
","payload":"payload","payloadType":"msg","count":"1","x":630,"y":280,"wire
s":[[]]},{"id":"c250e97.2348c98","type":"ui_slider","z":"42a191ae.3209e","n
ame":"","label":"Pilot
PWM","tooltip":"","group":"d8e40fbf.935f28","order":2,"width":0,"height":0,
"passthru":true,"outs":"all","topic":"topic","topicType":"msg","min":0,"max
":"100","step":1,"x":190,"y":280,"wires":[["a63d7329.acb068","58ec5535.cdde
fc"]]},{"id":"a63d7329.acb068","type":"function","z":"42a191ae.3209e","name
":"","func":"msg.payload = 100-msg.payload;\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":400,"y":280,"
wires":[["7db7417b.1c1208"]]},{"id":"58ec5535.cddefc","type":"ui_gauge","z"
:"42a191ae.3209e","name":"","group":"d8e40fbf.935f28","order":1,"width":0,"
height":0,"gtype":"gage","title":"PWM
Percentage","label":"%","format":"{{value}}","min":0,"max":"100","colors":[
"#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":430,"y":340,"wires":
[]},{"id":"845c21b1.88d28","type":"i2c
in","z":"42a191ae.3209e","name":"","busno":"1","address":"96","command":"0"
,"count":"2","x":340,"y":440,"wires":[["96bfd73.37f1928"]]},{"id":"96bfd73.
37f1928","type":"function","z":"42a191ae.3209e","name":"","func":"var x =
(msg.payload[0] << 8) + msg.payload[1];\nmsg.payload =
(x*13.2/1024).toFixed(2);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":500,"y":440,"
wires":[["a4dbf22b.8b2958","2534da6d.1f0d46"]]},{"id":"a9eedb50.e702a8","ty
pe":"i2c
in","z":"42a191ae.3209e","name":"","busno":"1","address":"96","command":"2"
,"count":"2","x":340,"y":640,"wires":[["92c2e6ee.dd622"]]},{"id":"92c2e6ee.
dd622","type":"function","z":"42a191ae.3209e","name":"","func":"var x =
(((msg.payload[0] << 8) | msg.payload[1]) << 16) >> 16;\nmsg.payload = (x *
13.2/1024).toFixed(2);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":500,"y":640,"
wires":[["306e118e.e1e876","ddd8e215.8dd148"]]},{"id":"a4dbf22b.8b2958","ty
pe":"ui_chart","z":"42a191ae.3209e","name":"","group":"491a4af2.950874","or
der":1,"width":0,"height":0,"label":"CP Positive
Voltage","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpol
ate":"bezier","nodata":"","dot":false,"ymin":"0","ymax":"13","removeOlder":
"100","removeOlderPoints":"","removeOlderUnit":"1","cutout":0,"useOneColor"
```

:false,"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"outputs":1,"useDifferentColor":false,"x":690,"y":440,"wires":[[]]},{"id":"306e118e.e1e876","type":"ui_chart","z":"42a191ae.3209e","name":"","group":"3af4c094.77f89","order":1,"width":0,"height":0,"label":"CP Negative Voltage","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"bezier","nodata":"","dot":false,"ymin":"0","ymax":"-13","removeOlder":"100","removeOlderPoints":"","removeOlderUnit":"1","cutout":0,"useOneColor":false,"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"outputs":1,"useDifferentColor":false,"x":700,"y":640,"wires":[[]]},{"id":"63b9fb72.05c9cc","type":"inject","z":"42a191ae.3209e","name":"","props":[],"repeat":"1","crontab":"","once":true,"onceDelay":"1","topic":"","x":190,"y":440,"wires":[["845c21b1.88d28"]]},{"id":"61ddda95.d588cc","type":"inject","z":"42a191ae.3209e","name":"inject","props":[{"p":"payload"}],"repeat":"1","crontab":"","once":true,"onceDelay":"1","topic":"","payload":"","payloadType":"date","x":190,"y":640,"wires":[["a9eedb50.e702a8"]]},{"id":"2534da6d.1f0d46","type":"ui_text","z":"42a191ae.3209e","group":"491a4af2.950874","order":2,"width":0,"height":0,"name":"","label":"Value","format":"{{msg.payload}}","layout":"row-spread","x":650,"y":500,"wires":[]},{"id":"ddd8e215.8dd148","type":"ui_text","z":"42a191ae.3209e","group":"3af4c094.77f89","order":2,"width":0,"height":0,"name":"","label":"Value","format":"{{msg.payload}}","layout":"row-spread","x":650,"y":700,"wires":[]},{"id":"d8e40fbf.935f28","type":"ui_group","name":"PWM","tab":"cb528dc9.7100b8","order":1,"disp":true,"width":"6","collapse":false},{"id":"491a4af2.950874","type":"ui_group","name":"CP Pos","tab":"cb528dc9.7100b8","order":2,"disp":true,"width":"6","collapse":false},{"id":"3af4c094.77f89","type":"ui_group","name":"CP Neg","tab":"cb528dc9.7100b8","order":3,"disp":true,"width":"6","collapse":false},{"id":"cb528dc9.7100b8","type":"ui_tab","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]

12 Disclaimer

This product is not to be directly used on mains or other high voltage AC or DC lines.

13 Version History

Version	Date	Changes
0.1	12. March 2021	Initial Release