Sonal Uithoornseweg 42A 5701 CB Helmond Netherlands www.sarperonal.com	**Document No:** SNL000003 **Project Group:** High Voltage	Modified Date:28/02/2023 Status: For Review Type:Quick Start Guide

Title:

Quick Start Guide of HV BMS

Summary

This document describes how to quick start in order to prove the function and performance of the following subsystem:

Subsystem name: High voltage battery management system Controller

The main tasks of the subsystem are:

- Reading TPL signals from NXP 6-14 cell bms systems
- Controlling outputs depending on the cell safety status
- Sending Voltage and temperature data over Rs485 Modbus

Link for the item: https://www.sarperonal.com/product/high-voltage-bms/

It supports the following interfaces:

- Wifi
- RS485 Modbus
- I2C
- TPL
- UART

Maintainer:	Revievers	For info
SONAL	Steven T.	Public

Open Issues

ID	Short Description	Action By	Status
001	Hardware Creation	SONAL	Done
002	HW Documentation	SONAL	Done
003	TPL Connection	STVTN	Done
003	RS485 Protocol	STVTN	Done
~~004~~	~~CanBus Protocol~~	~~SONAL~~	~~Open~~
005	Wifi Protocol	SONAL	Open
006	Modbus	STVTN	Done
007	Shunt connection	STVTN	Open
008	Input verification	SONAL	Open
009	Output Verification	SONAL	Done
010	GUI	SONAL	Done

Refferance Documents

Title	Author	Link

Abbreviatons

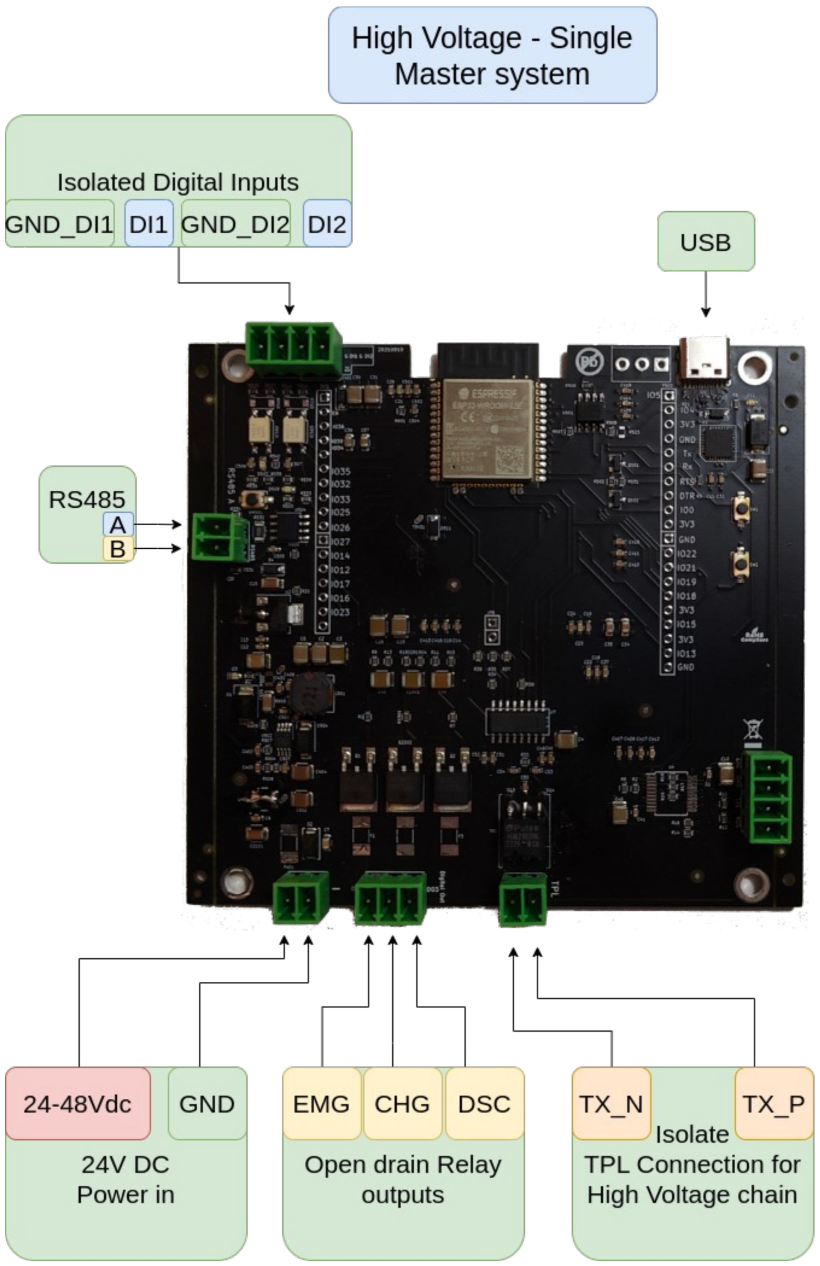
Abbreviation	Description
TPL	Transformer Physical Layer
OV cut	Over Voltage cut
UV cut	Under Voltage cut
OT cut	Over Temperature cut

1. Introduction

1. Scope

As a BMS producer, it is essential to ensure that the BMS systems we provide are thoroughly tested and meet the highest quality standards. This document outlines how to quick start with the system which is a crucial part of our customer service for NXP's individual 6-cell or 14-cell BMS systems.

Our BMS systems are designed to provide efficient and reliable management of both new and recovered batteries, including those that have been scraped, recycled, or recovered from crashed EV batteries. The subsystem's primary function is to ensure that these cells can be safely and effectively reused, reducing waste and contributing to a more sustainable future.



Drawing 1: Controller Pinout Diagram

2. Power:

Our BMS system can be powered in two ways:

- **USB connector:** This connector can be powered from a USB-C port. It consumes a steady 250mA of power, with a maximum of 500mA for 10ms.

- **Power Connector:** Our system can accept a wide range of power inputs from 12Vdc to 50Vdc over the power connector. The power is completely isolated from the high voltage signals, so you won't be connected to the HV part of the battery.

  Please note that if you power the BMS system from both connectors simultaneously, it will draw power from the first connector that was connected. If you need a wider range of input DC power, just let us know. We can customize your system to meet your specific needs.

3. Digital Inputs:

Our HV BMS system has two digital inputs that are completely isolated from any other sources. They can accept input sources ranging from 16Vdc to 50Vdc. You can use any kind of input source, such as fire/water/gas alarms or any kind of emergency signal, with the input modules.

If you need a special triggering mechanism for input signals, please consult with us. For instance, we can easily implement charge cutoff or discharge cutoff for digital inputs.

4. Open Drain Digital Outputs:

Our high voltage BMS system features 3 contactor outputs: one for Emergency, one for Charge, and another for Discharge. You can use all 3 or just 1, depending on your project's requirements.

Refer to the diagram below for an example of how to connect the contactor output. For any of the 3 digital output connections, we recommend using "**Normally Open**" (NO) type contactors to ensure proper functionality.

Please note that each output requires a separate contactor to control the HV signal. If you have any questions about the connection or configuration of the contactor outputs, please consult with our support team.

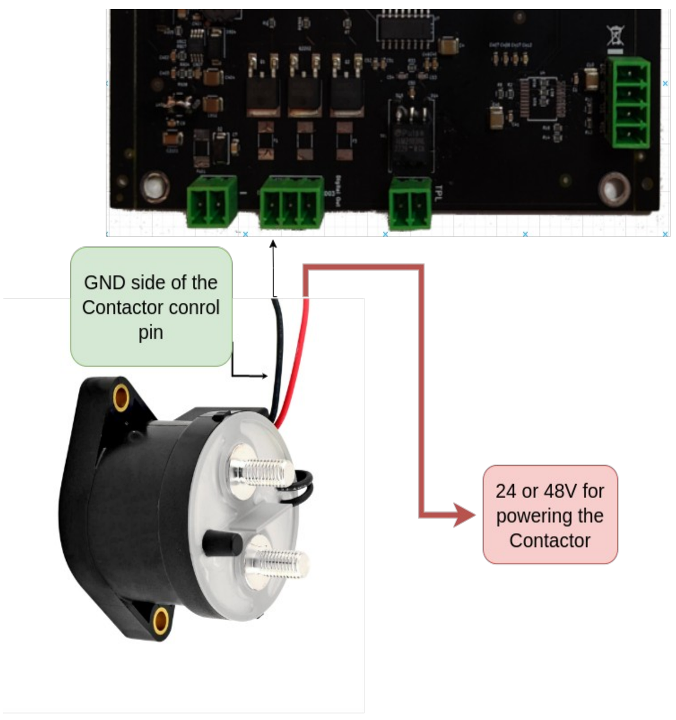

GND side of the Contactor conrol pin
24 or 48V for powering the Contactor

Figure 1: Contactor Output connection diagram

States of contactor outpus:

	Emergency Out	Charge output	Discharge output
Emergency OFF	Emergency contactor will be off state and current wont pass throught its contacts	Charge contactor will be off state and current wont pass throught its contacts	Discharge contactor will remain on and current will pass
OverVoltage Cut	4.2V Example setValue	4.2V	4.2V
Emergency ON	Emergency contactor will remain on and current will pass	Charge contactor will remain on and current will pass	Discharge contactor will remain on and current will pass
Under Voltage Cut	3.0V Example setValue	3.0V	3.0V
Emergency OFF	Emergency contactor will be off state and current wont pass throught its contacts	Charge contactor will remain on and current will pass	Discharge contactor will be off state and current wont pass throught its contacts
0V	0V	0V	0V

Figure 2: States of Contactor Outputs

5. TPL connection

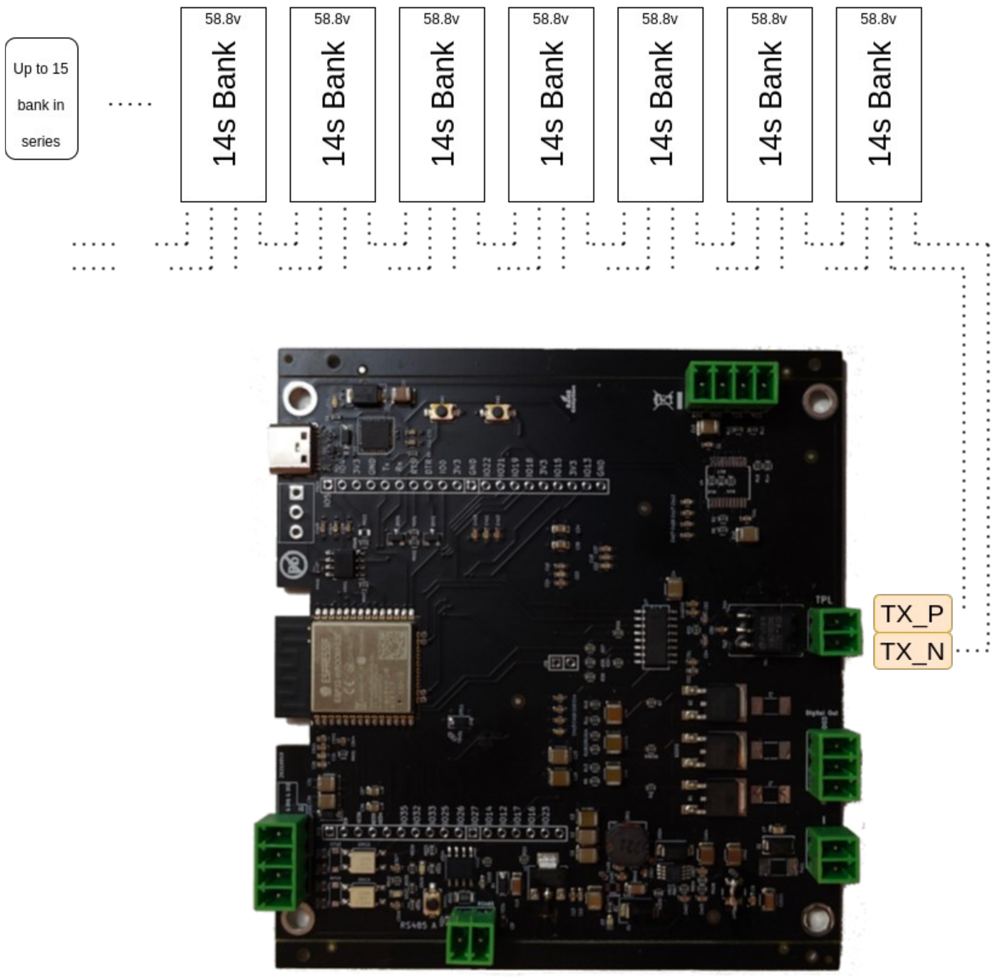


Figure 3: Simplified connection diagram of TPL connection

Connecting Multiple Banks:

Our high voltage BMS system is designed to handle multiple battery banks. You can connect up to 15 slave connections to the TX_P and TX_N connection points. This means that you can attach a maximum of 15 banks, with each bank being capable of supplying up to 14 cells at 4.2V each.

Multiple Master Modules:

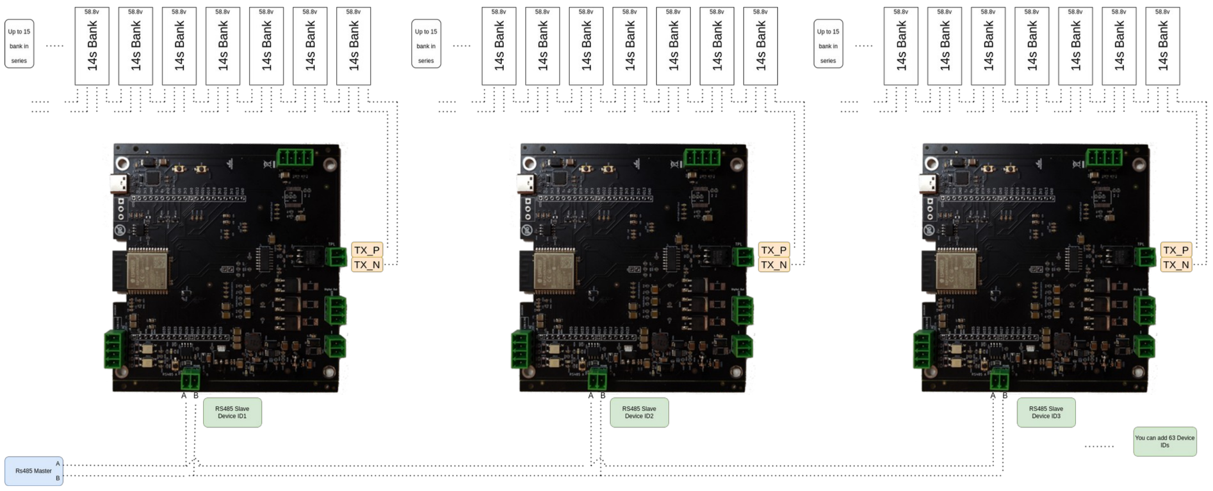
If you are using multiple master modules in your system, you can power them all using the same 24V power supply. Whether you're connecting the master modules in parallel or series to your HV banks, a single power supply can be used to power all the master modules.

If you have any questions about connecting multiple banks or master modules, please don't hesitate to contact our support team. We're here to help you get the most out of our high voltage BMS system.

6. RS485 Modbus connection:

Our high voltage BMS system comes with a modbus comm connector on each module, allowing you to read all cell voltage and temperature states over Modbus. Each device has a soft-coded ID, which can be changed along with other Modbus parameters by accessing the Settings page of the device from the web GUI.

To connect multiple devices to the same RS485 line with different device IDs, you can attach 1 to 14 slaves to each BMS master. This will allow you to easily reach all the data over Modbus and monitor the status of your battery banks.



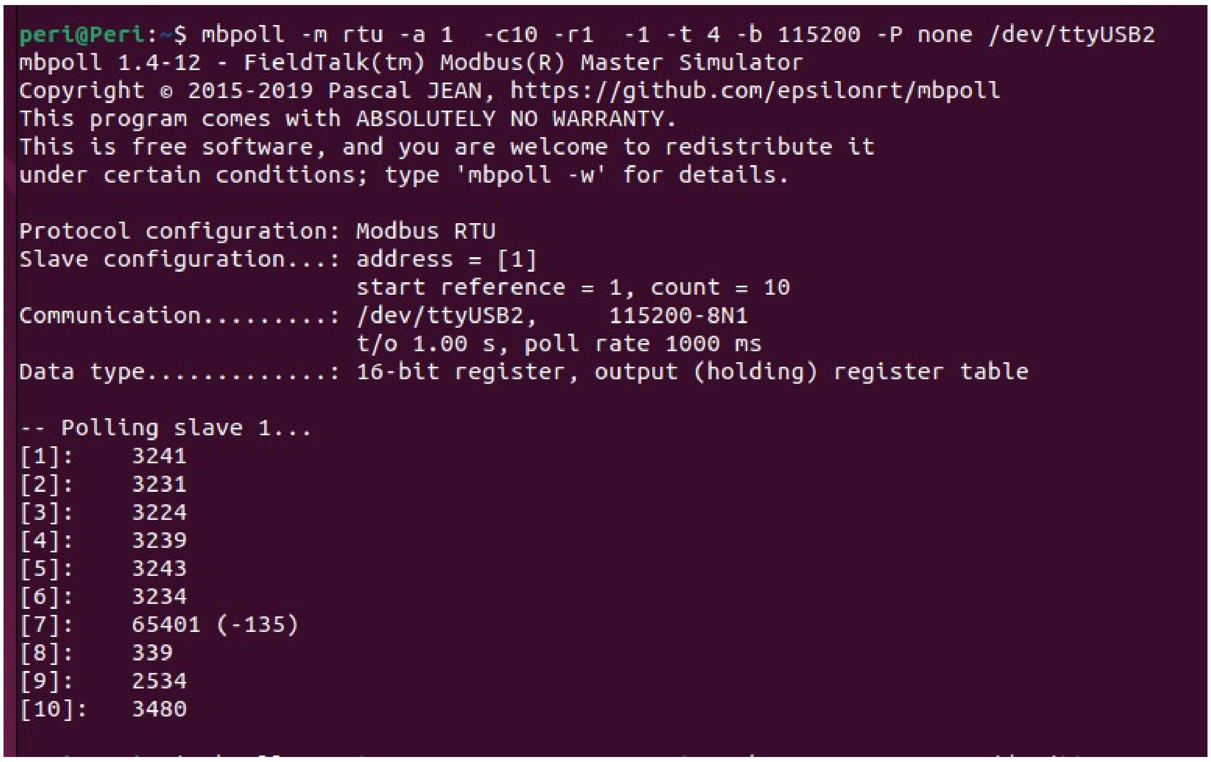
If you have any questions about setting up the RS485 Modbus connection, please don't hesitate to contact our support team. We're here to help you get the most out of our high voltage BMS system.

Example code or multiple addresses:
Polling data from device 1:
mbpoll -m rtu -a 1 -c9 -r10 -1 -t 4 -b 115200 -P none /dev/ttyUSB2

Polling data from device 2:
mbpoll -m rtu -a 2 -c9 -r10 -1 -t 4 -b 115200 -P none /dev/ttyUSB2

Polling data from device 3:
mbpoll -m rtu -a 3 -c9 -r10 -1 -t 4 -b 115200 -P none /dev/ttyUSB2

detailed explanation of the code:
-a 1 : it reffers to Device 1
-c9 : dump 9 register
-r10: start form 40010
-t 4 : 40000
-b: defines baudrate, in our case it is 115200



```
peri@Peri:~$ mbpoll -m rtu -a 1  -c10 -r1  -1 -t 4 -b 115200 -P none /dev/ttyUSB2
mbpoll 1.4-12 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright © 2015-2019 Pascal JEAN, https://github.com/epsilonrt/mbpoll
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'mbpoll -w' for details.

Protocol configuration: Modbus RTU
Slave configuration...: address = [1]
                        start reference = 1, count = 10
Communication.........: /dev/ttyUSB2,     115200-8N1
                        t/o 1.00 s, poll rate 1000 ms
Data type.............: 16-bit register, output (holding) register table

-- Polling slave 1...
[1]:    3241
[2]:    3231
[3]:    3224
[4]:    3239
[5]:    3243
[6]:    3234
[7]:    65401 (-135)
[8]:    339
[9]:    2534
[10]:   3480
```

To retrieve data from the BMS over Modbus, you can use the provided example code that dumps 10 registers starting from 40001 up to 40010. Please note that the code is configured for a Ubuntu computer with a RS485 USB converter connected. However, there may be different approaches to getting the data with a PLC. This document does not cover the PLC side of communication. Nonetheless, the provided code should assist you in communicating with the device.

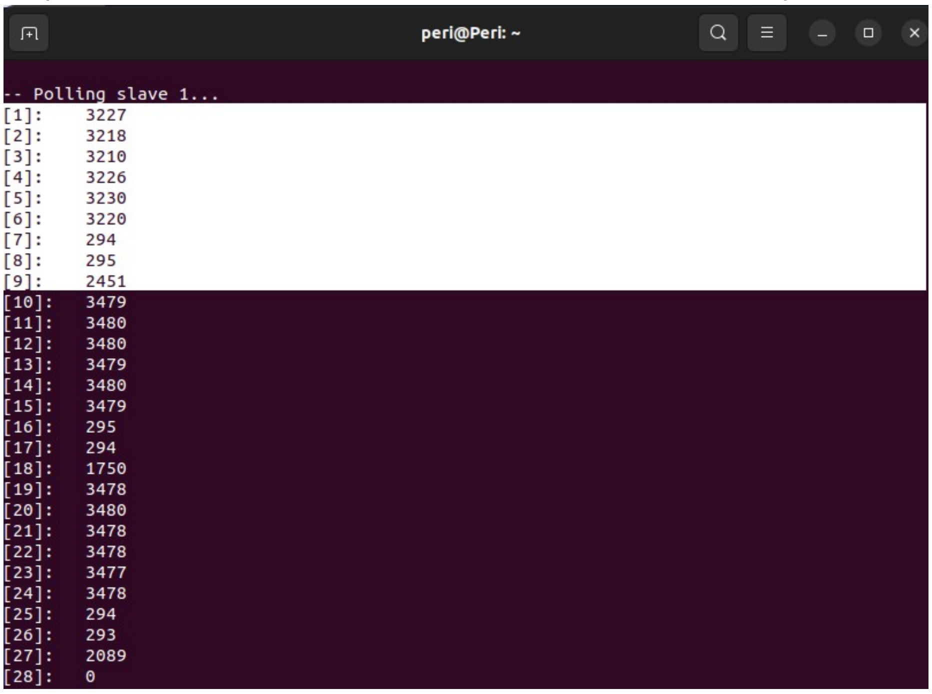
For example, to dump 50 registers starting from 40001, you can use the following code in our testing environment:

mbpoll -m rtu -a 1 -c 50 -r 1 -1 -t 4 -b 115200 -P none /dev/ttyUSB3

Please adjust the code accordingly to fit your specific environment and requirements.

In our testing environment: Device used for Modbus communication:
DSD TECH SH-U10 USB to RS485 Converter

Code example for dumping 50 registers starting from 40001:
mbpoll -m rtu -a 1 -c50 -r1 -1 -t 4 -b 115200 -P none /dev/ttyUSB3



peri@Peri: ~

```
-- Polling slave 1...
[1]:    3227
[2]:    3218
[3]:    3210
[4]:    3226
[5]:    3230
[6]:    3220
[7]:    294
[8]:    295
[9]:    2451
[10]:   3479
[11]:   3480
[12]:   3480
[13]:   3479
[14]:   3480
[15]:   3479
[16]:   295
[17]:   294
[18]:   1750
[19]:   3478
[20]:   3480
[21]:   3478
[22]:   3478
[23]:   3477
[24]:   3478
[25]:   294
[26]:   293
[27]:   2089
[28]:   0
```

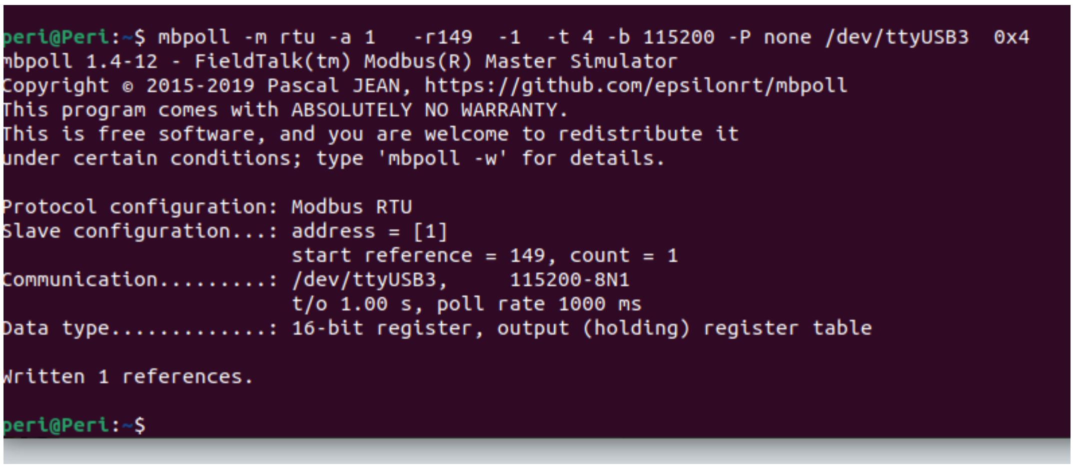
As you can see 7th register is a NTC temperature and needs to be divided to 10.
8th temperature is a MCU temperature and needs to be devided to 10

Code example for defining Bank Count over RS485:

```
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB3  0x1
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0x2
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0x4
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0x7
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0x8
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0x9
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0xa
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0xb
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0xc
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0xd
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0xe
mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB6  0xf
```

0xf will give acces to 15 banks in daisy chain

You can attach as many banks you want up to 15. Daisy chain supports max 15 devices(address register limit)



```
peri@Peri:~$ mbpoll -m rtu -a 1   -r149  -1  -t 4 -b 115200 -P none /dev/ttyUSB3  0x4
mbpoll 1.4-12 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright © 2015-2019 Pascal JEAN, https://github.com/epsilonrt/mbpoll
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'mbpoll -w' for details.

Protocol configuration: Modbus RTU
Slave configuration...: address = [1]
                        start reference = 149, count = 1
Communication.........: /dev/ttyUSB3,      115200-8N1
                        t/o 1.00 s, poll rate 1000 ms
Data type.............: 16-bit register, output (holding) register table

Written 1 references.

peri@Peri:~$
```

7. Cable connections

We assume you made the harness of the slave BMSs with the supplied connector and pre attached wires. All the cells needs to be connected to the cables while the connector is not attached to the Slave BMS.

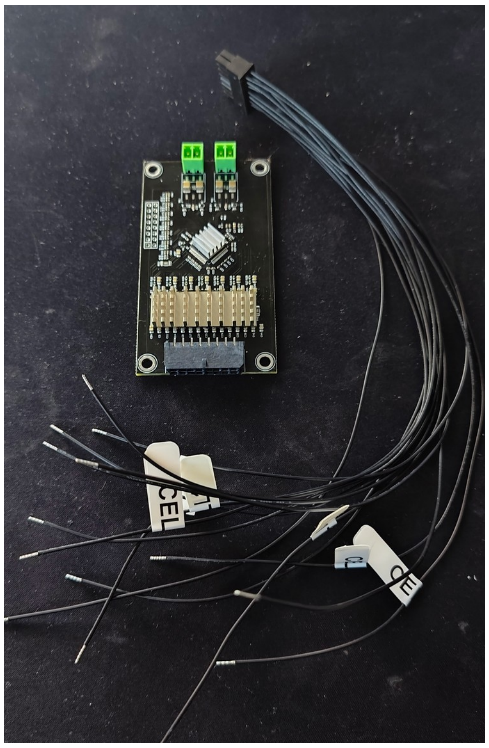


Figure 4: Cable wire connector

After connecting all cables and connections to the battery cells please measure the voltage on connector end side before connecting to the Slave BMS

Silk screen on the PCB will help you to understand if you are measuring the correct cell. Wrong connection can result continuous balancing and this can damage the PCB and also the battery. Please avoid wrong connection by mast measurement before connection

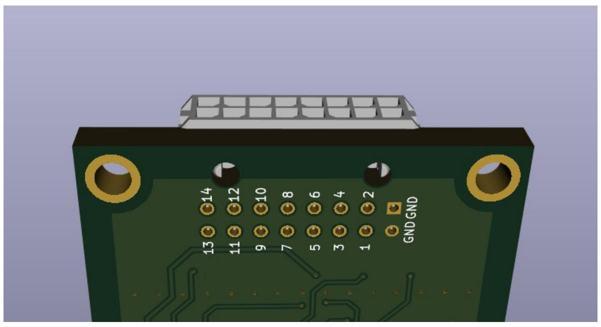


Figure 5: Silk Screen of cell numbers on PCB

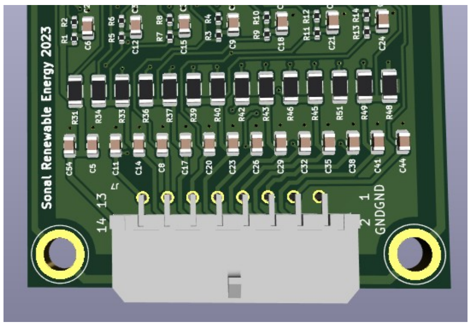


Figure 6: Silk Screen of cell numbers on PCB

➔ **Connecting daisy chain cables inbetween Master module and Slave modules.**

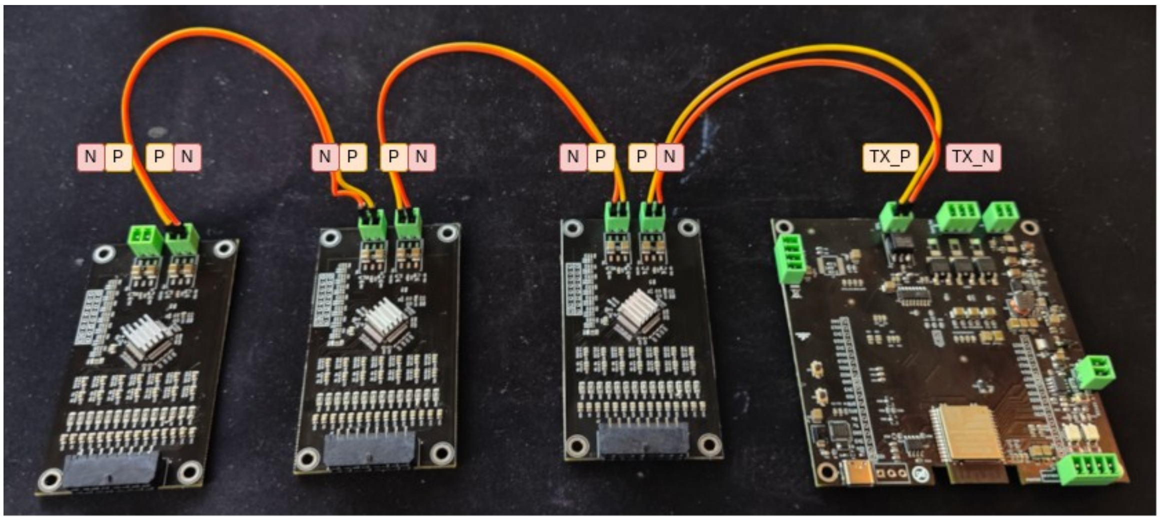


Figure 7: Daisy Chain Connection

Be sure that daisy chain connection has been made correctly inbetween master and slave modules and also inbetween slave modules.

8. Wifi connection and Settings

We assume you made the harness of the slave BMSs
Link for Slave BMS modules:
https://www.sarperonal.com/product/high-voltage-slave-14s-to-210s-system-can-be-

connected/

15 slave module can be connected to single master board.Please connect all Slave modules to the related cell voltage reading connectors.

Then please connect all the slave modules to eachother. And at the end before powering the master module you should connect the TPL cable to the master module

After this step we can power up HV BMS master module. Either from Wide input power socket or over USB.

Master module will check how many slave boards are connected to itself and will determine the cell count by itself.

Please open a hotspot with cridentials with MoonitorBMS
SSID: **MoonitorBMS**
Password: **MoonitorBMS**

then master module will connect this hotspot, after this poing you should open the IP address of the master module from your browser.

9. Emergency Cuts

a. OverVoltage cut

b. UnderVoltage cut

c. OverTemperature cut

"We are in the process of writing this section and it will be available soon."