Rebuilding the Boosted Rev

Documentation

Roland Stollnberger
stollnberger@gmx.at
10.10.2024

Photo: Levi Jacob Price

Disclaimer

The steps and procedures described in this guide for replacing the electrical components of an e-scooter are intended only for experienced and knowledgeable individuals. Replacing and using modified electrical components can be dangerous and requires a high level of technical understanding and craftsmanship.

Important Notes:

Legal Regulations: Please observe local laws and regulations regarding modifying or the use of modified e-scooters on public roads.
Disclaimer: The author of this guide assumes no liability for damages or injuries resulting from the replacement or use of the described electrical components or the provided code. All information is provided without warranty.
Electrical Safety: Handling electrical components can be dangerous. Replacing and using modified electrical components is at your own risk. Ensure that all electrical connections are properly insulated and secured to avoid short circuits and electric shocks.

By using this guide, you agree to the above terms and confirm that you understand the risks.

Unfortunately, a simple battery replacement is not possible with the Rev because the communication between the battery's BMS and the controller has not been cracked yet. The original controller does not start when a retrofit BMS is connected.
To bring the Rev back to live and enhance the range it is necessary to replace the controller, the battery and the dashboard.
There is no plug & play solution. Three main steps are necessary to rebuild the Rev:

- Build a dashboard, in this case with a Lilygo T-Display S3 Touch
- Make a 12s5p Battery
- Buy a VESC based Controller - the Lacroix Stormcore D60+ was used here
  (Lacroix went out of business, but there are some alternatives like the VESC HD-60T)

All necessary Files are provided here:
https://github.com/stolliroli/revolution-dashboard

Step 1 - Dashboard

Part list:

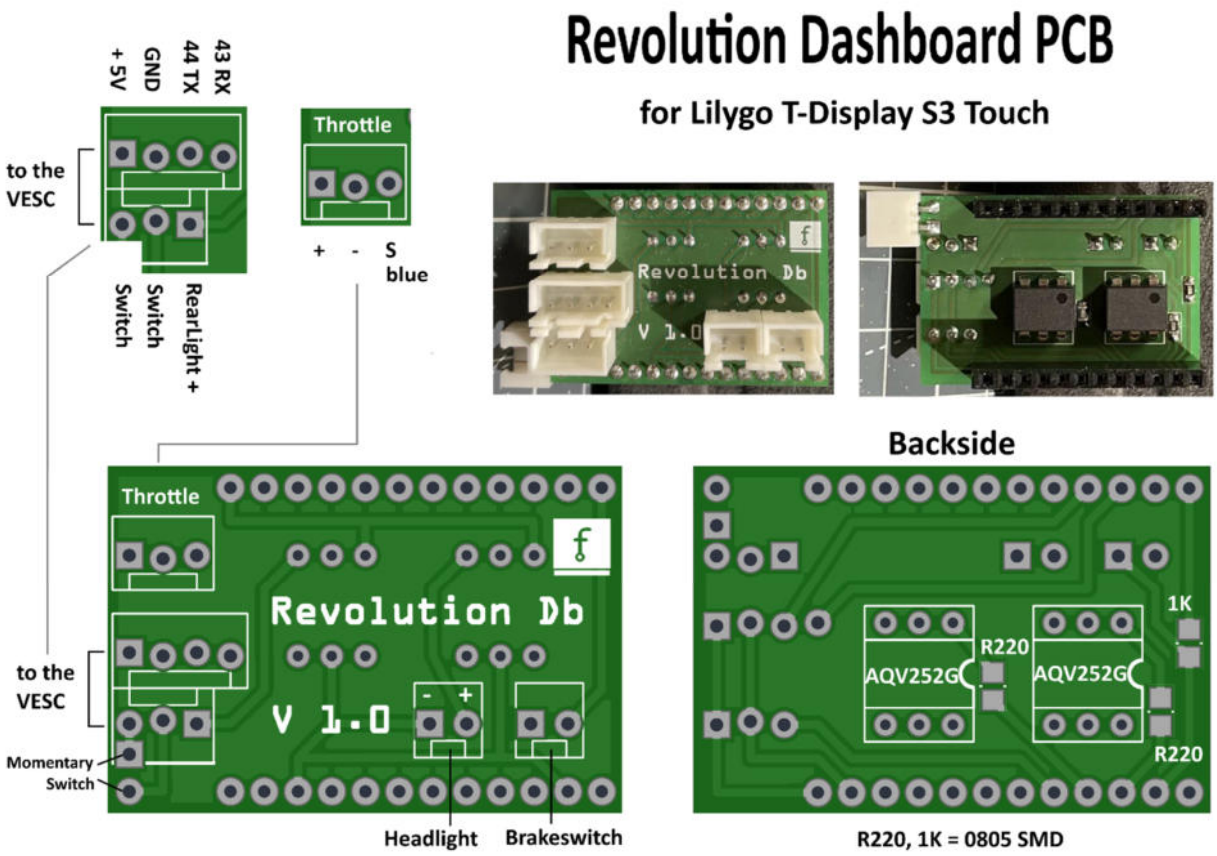
Lilygo T-Display S3 Touch	https://www.lilygo.cc/products/t-display-s3?variant=42589373268149
PCB	https://aisler.net/stolliroli/rolis-projects/revolution-dashboard Or use any other fabricator, the files are located in the provided link above.

2x AQV252G PhotoMOS
2x 220 Ohm Resistor, 0805 SMD
1x 1 kOhm Resistor, 0805 SMD
1x 12mm Momentary Switch (short version)
2,54mm JST-XH connector (3x 2Pin, 2x 3Pin, 1x 4Pin) It's best to get a set
3D printed parts (PETG or any other outdoor suitable filament)
Some 2mm screws
2.54mm 2x 12 Pin Female socket Board
Thin wire
4x M3 brass thread inserts

The easiest way to write the code to the Lilygo is using the Arduino IDE.
Following the instruction on Github https://github.com/Xinyuan-LilyGO/T-Display-S3 will help to install the libraries and do the correct settings for the Lilygo display.
Further instructions are in the code “Revolution_Dashboard_1.0.ino” of the shared link.

Once the code is opened in Arduino IDE, **there are some settings to check in the Config.h tab** (Step 6 – Dashboard config). To program the board later over WIFI **it is important to provide your WIFI SSID and Password before installing the Lilygo in the housing**. After installation it becomes difficult to connect USB.
When the code has been uploaded, next time the Lilygo is connected to USB Power and any of the unlock codes (for driving modes) has been typed in, “WIFI connected” will appear below the battery bar.
Now the port settings have to be changed in Arduino IDE to the IP address of your Lilygo to test the OTA programming.

PCB assembly:

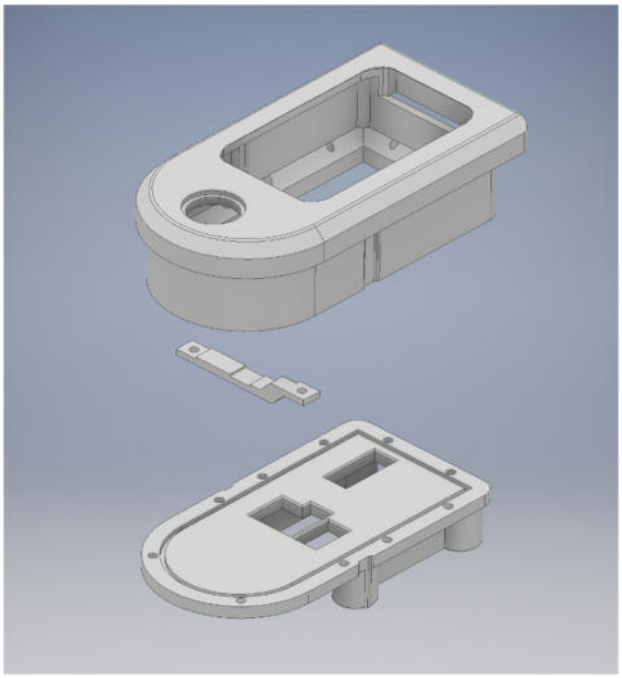


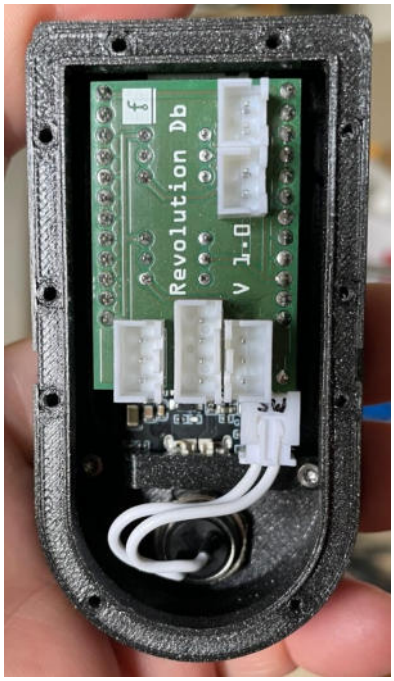
Before assembly, the housing can be coated with matt clear varnish to make it waterproof.

The Lilygo display is attached with the bracket and two 2mm flat head screws. None of the SMD components should get pinched (If the Lilygo board layout has been changed). Now the PCB can be attached to the Lilygo Display.

Note the following steps if you want to seal the display.
I used “Kafuter K-586 Black Silicone Free Gasket” to seal the display.
First, cover the display with thin plastic packaging tape and insert it into the housing.
Using a craft knife to careful cut along the display recess in the housing. Don't worry, the display glass is so hard and won't get any scratches.
Remove the display again and peel off the tape around the display.
Also cover the housing with the packaging tape and carefully cut out the recess for the display.
Apply the sealant to the inside edge where the display will later rest.
After attaching the display, excess sealing can be wiped off and the packaging tape removed.
Allow the seal to cure overnight.

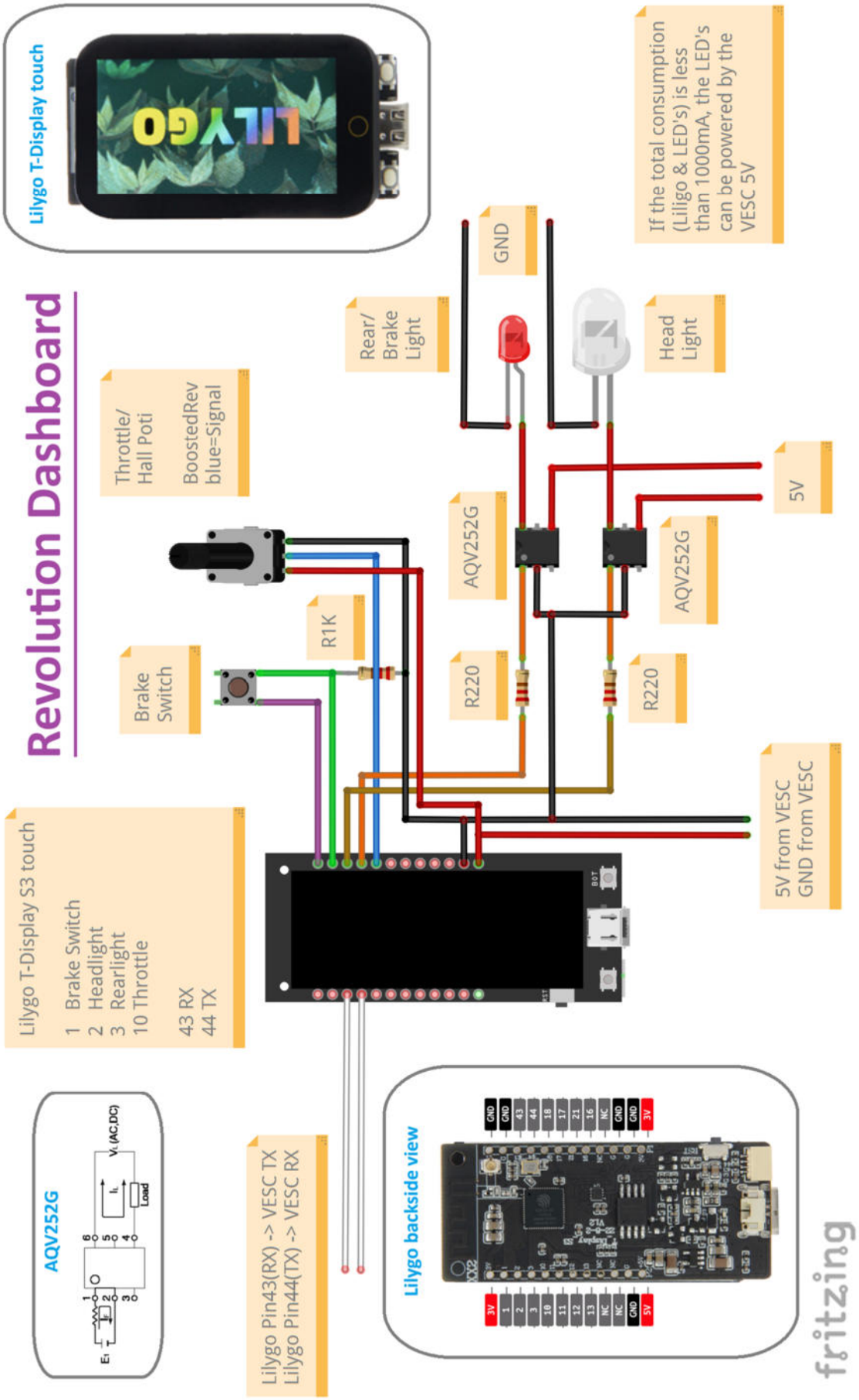
Solder the thin wires to the momentary switch and connect them to the PCB with a 2Pin JST-XH plug.





Screw in 10 M2 countersunk head screws and melt in the M3 brass thread inserts to finish the dashboard.

Wiring diagram:

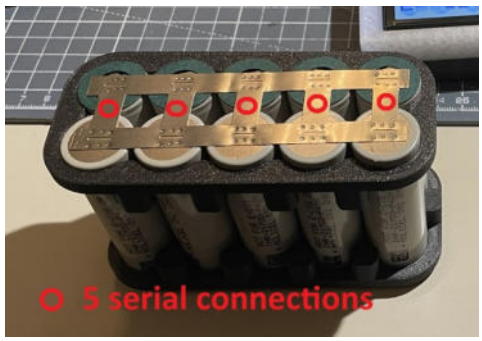


Step 2 - Battery build

Part list:
60x INR21700-P42A https://www.nkon.nl/de/molicel-21700a-4200mah-30a.html
JBD Bluetooth Smart BMS 10S-17S 50A
https://www.aliexpress.com/item/1005005997653586.html?spm=a2g0o.order_list.order_list_main.67.34b61802zEt qc9
pure nickel strips (take a look at the chart below)
10 AWG Silicon Wire red/black ~30cm (main cable)
16 AWG Silicon Wire red/black ~1m (charging cable)
14 AWG Silicon Wire ~12cm (bus bar)
XT90 female connector (to Stormcore)
XT30 connector pair (charging cable)
60x Cell insulator rings (gasket paper) for 21700 cells 
Battery Insulation Gasket Barley Paper 
Kapton tape 50mm
Shrink tube 180mm x 1m

Keep in mind that your new setup will draw up to 40-50 Ampere. But not all spot welders are able to weld 0.3mm nickel stripes. Multiply the amperage in the table by the number of cells connected in parallel to determine your requirements. Example: 9,6A (0,2mm) x 5 (Cells) = 48A
You can use wider stripes as well. It depends on what you want to do with your Rev.
My measurement was max. 40 Amps on the original Rev ESC.

Acceptable current levels for pure nickel strips			
Size	Optimal	Acceptable	Poor
0.1 mm x 5 mm	< 2.1 A	~ 3.0 A	> 4.2 A
0.1 mm x 7 mm	< 3.0 A	~ 4.5 A	> 6.0 A
0.15 mm x 7 mm	< 4.7 A	~ 7.0 A	> 9.4 A
0.2 mm x 7 mm	< 6.4 A	~ 9.6 A	> 12.8 A
0.3 mm x 7 mm	< 10 A	~ 15 A	> 20 A

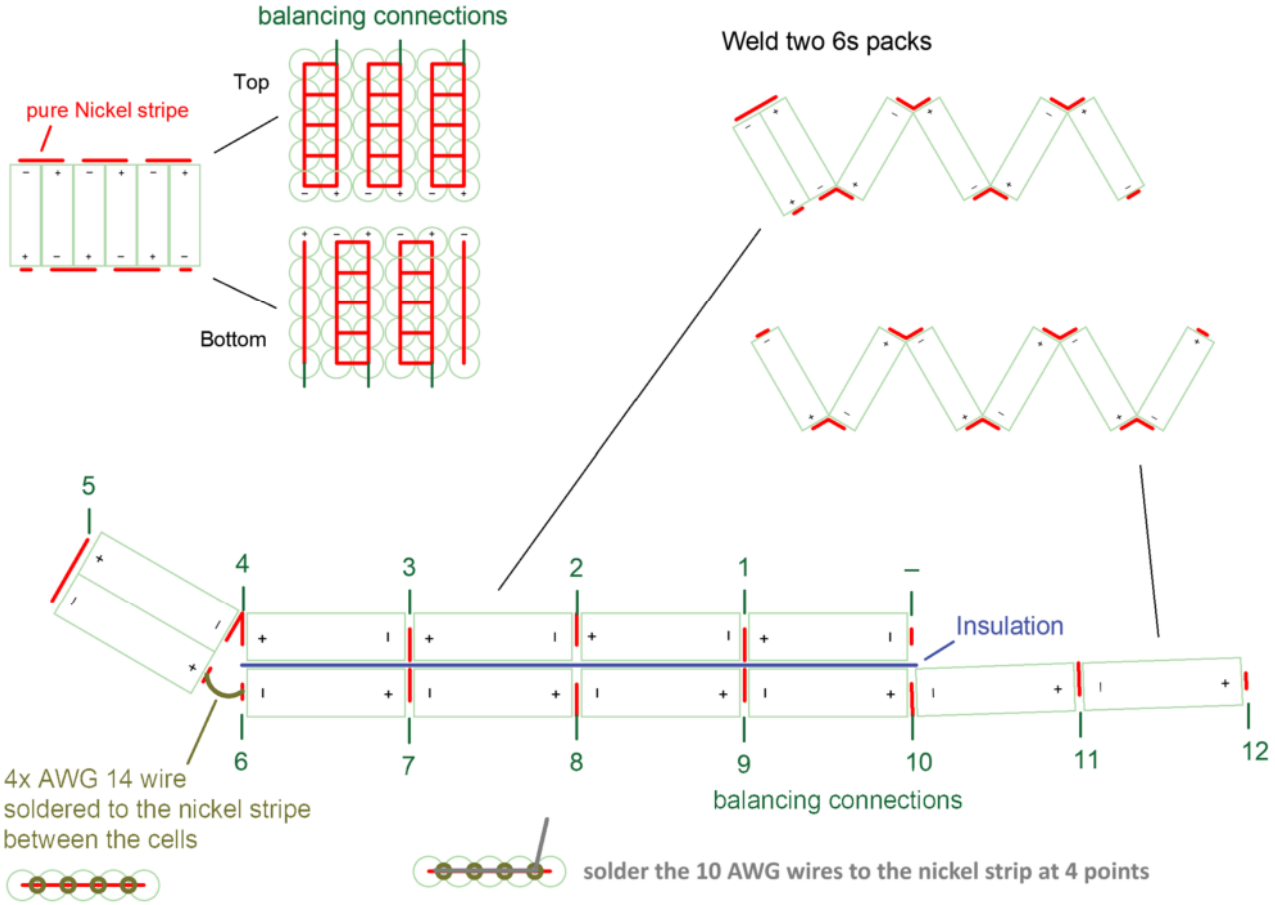


In order to build a good battery, it is important to **test the cells**. Faulty cells affect battery performance and lifespan. Many tests can take a lot of time. You should at least check the internal resistance of the individual cells. Outliers should be eliminated. The **YR1035+** would be a good choice for this task. **Furthermore, it is very important that the cells have the same voltage before welding the cells in parallel to avoid excessive compensating currents.**
Calculation: Ampere = V diff / (mOhm/1000). 1-2 A will be OK.
For balancing the cells you need a charger. My V diff was below 0,01V.

After a lot of research, I decided on the kWeld spot welder.
It draws 1280 Ampere per weld.
This thing rocks.
https://www.keenlab.de/index.php/product/kweld-complete-kit/

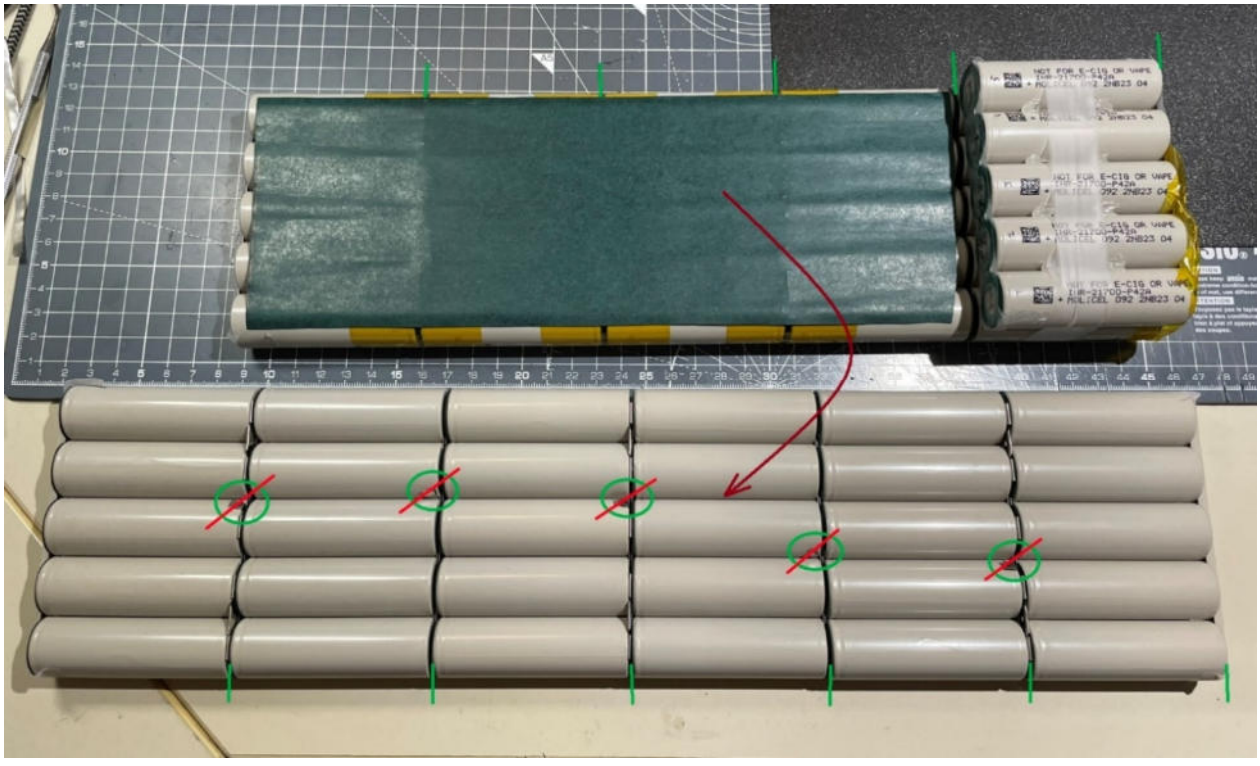
Attention: Always cover all unused contacts to avoid short circuits when working on the battery.

Cell arrangement:



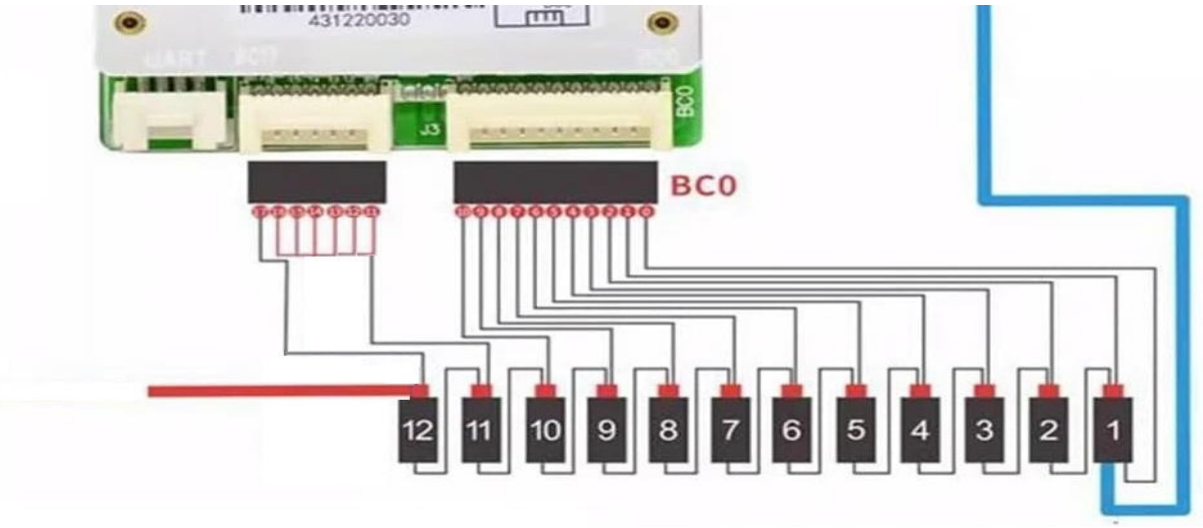
Since I wanted to make the battery as compact as possible, I routed the balancer cables between the two rows of cells. It turns out, there is enough space in the Rev to route the cables along the side of the battery.

Make sure the nickel strips cannot touch each other when joining the rows of cells. A few layers of insulation paper or a layer of epoxy glass is recommended.



Remove the plug from the BMS first to solder the BMS cables to the battery. This must be connected later.

The BMS can handle 17 cells in serial. Since we are building a 12s battery, the balancer cables 11-16 must be connected together to the battery connection 11. The 17th cable is connected to cell 12.

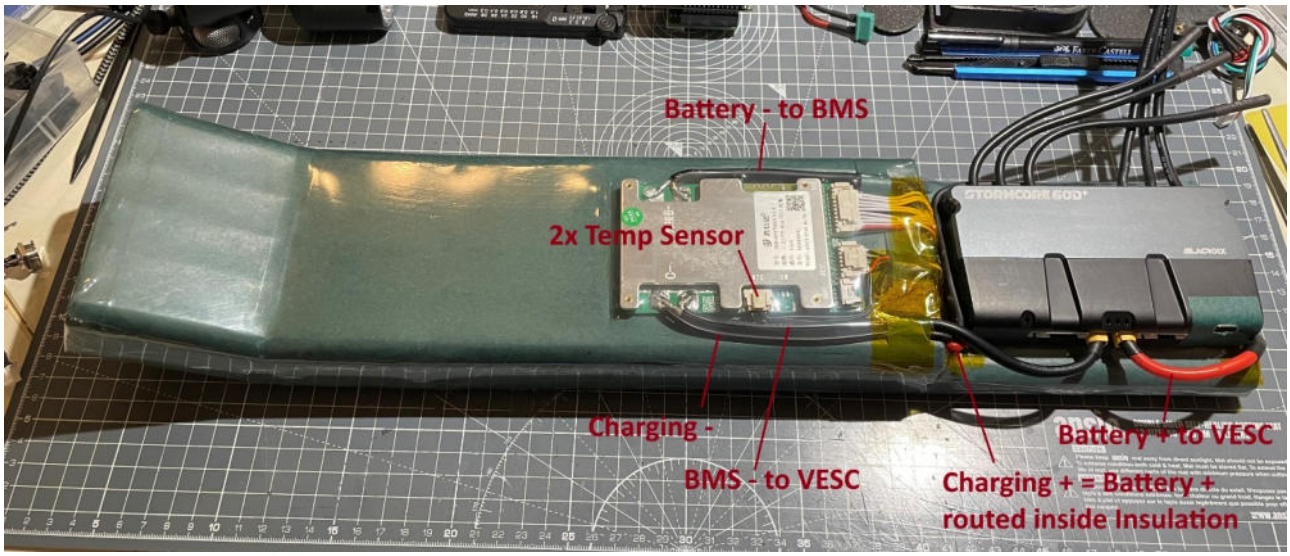


Building this battery is a bit difficult because the floor inside the Rev is not flat.
To avoid puncturing the battery, I glued foam rubber to important places. Make sure that the battery does not press on the screws of the skid plates.

Before wrapping the battery in insulation paper, I taped off all the contacts and placed the battery in the Rev to measure the angles of the bends. To do this I used a small electronic protractor. Then I recreated the angles on the workbench using wedges. Once the insulation paper is glued to the battery (start with the side walls first), the shape is maintained and the battery should fit into the Rev.

My Rev doesn't need to be waterproof, but I wanted to keep most of the dust out. That's why I used a 3D printed plate at the back and closed the holes at the floor.
Later I closed the front and rear openings of the side channels with foam rubber.

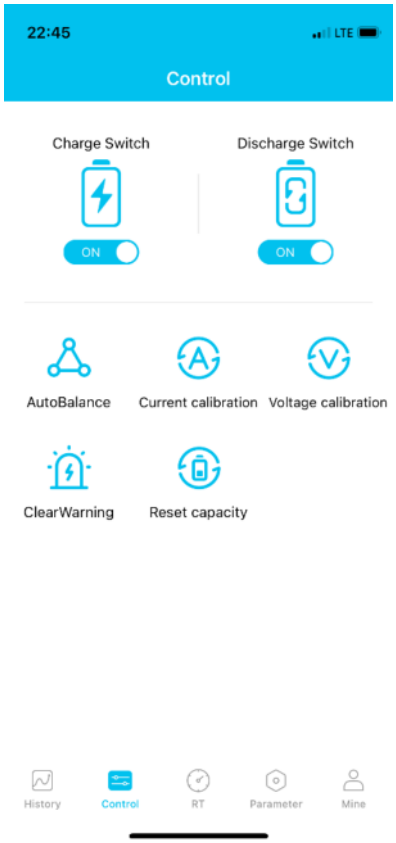
The BMS has been attached with double-sided adhesive tape to the battery. To avoid heat problems, it is recommended to insert a sheet of epoxy glass in between. Connect the main cables and the charging cable and also the temperature sensors. The final step is to connect the BMS cables and **setup the BMS**.
Attach the shrink tubing.

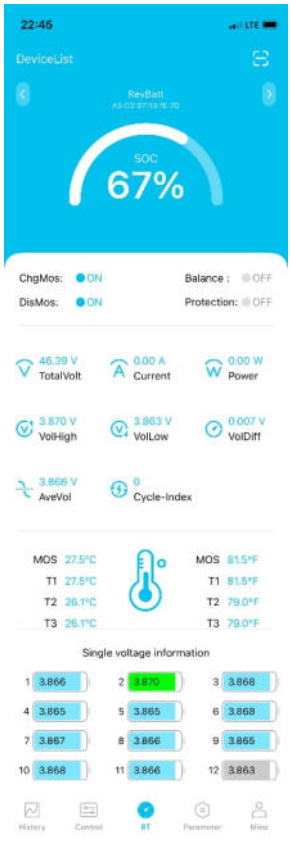


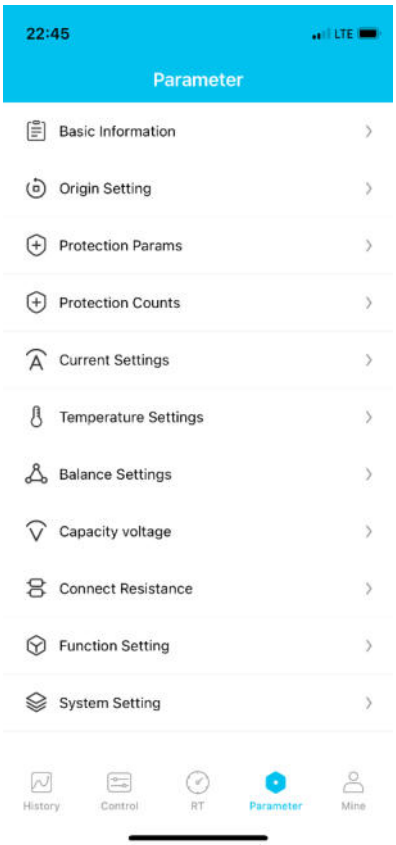
I taped the ESC on top of the lower cell-row. The ESC could reach higher temperature. A sheet of epoxy glass and thicker double-sided tape is recommended to get a bit of air circulation.

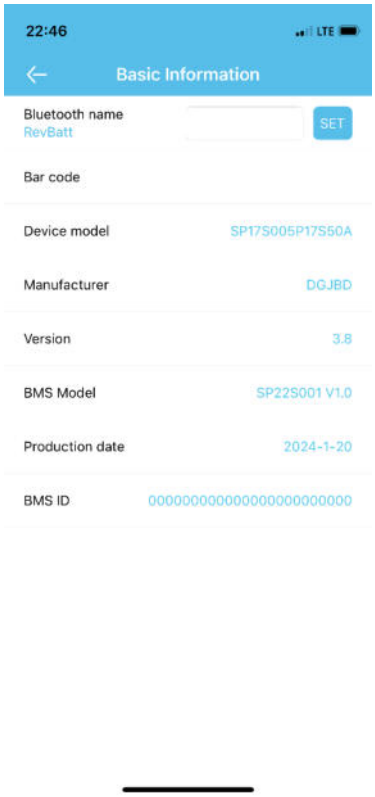
Step 3 – Setup the BMS

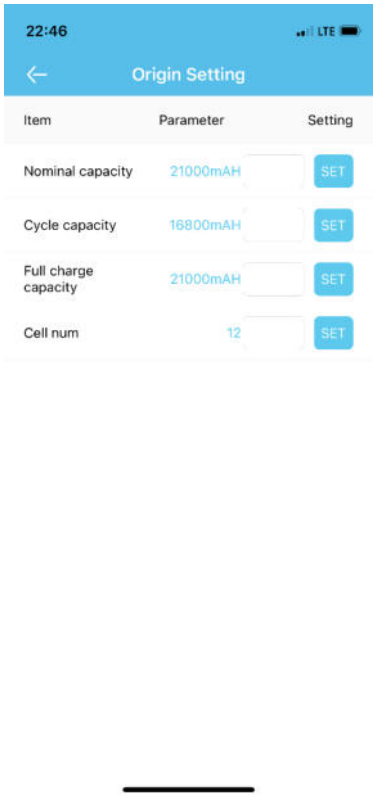
The output and charging function were turned off as default at my JBD BMS. To setup the BMS download the smartphone app. There will be a recommendation from the seller for the associated app to use.
In my case I installed the XiaoXiang iOS app.
You must create a user account to have fully access to the software. Otherwise you can only view data but not change any settings.

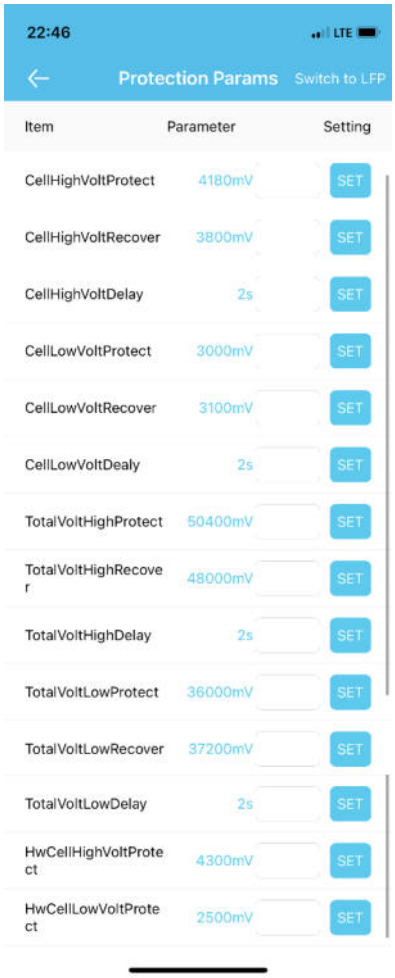


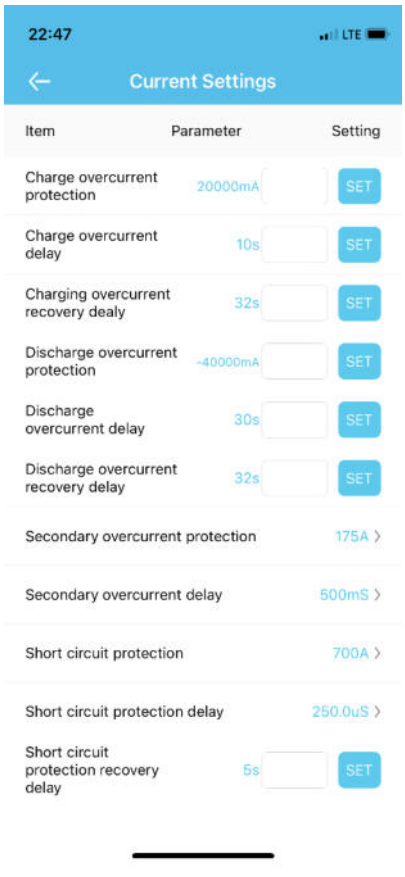




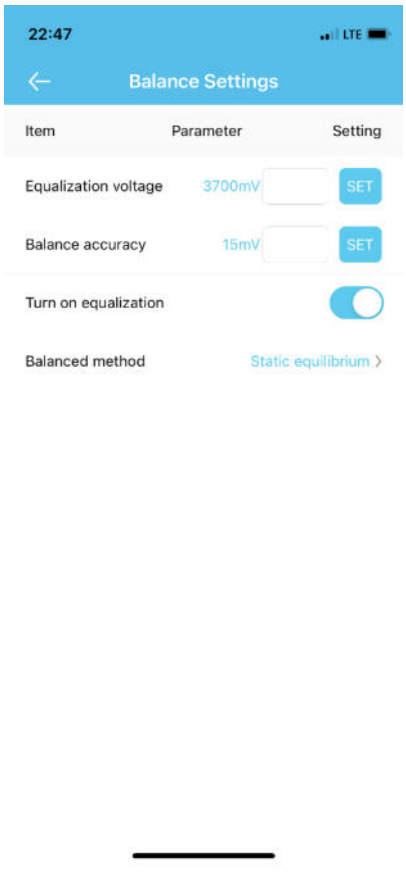

22:46
LTE
Basic Information
Bluetooth name
RevBatt
SET
Bar code
Device model
SP17S005P17S50A
Manufacturer
DGJBD
Version
3.8
BMS Model
SP22S001 V1.0
Production date
2024-1-20
BMS ID
00000000000000000000000

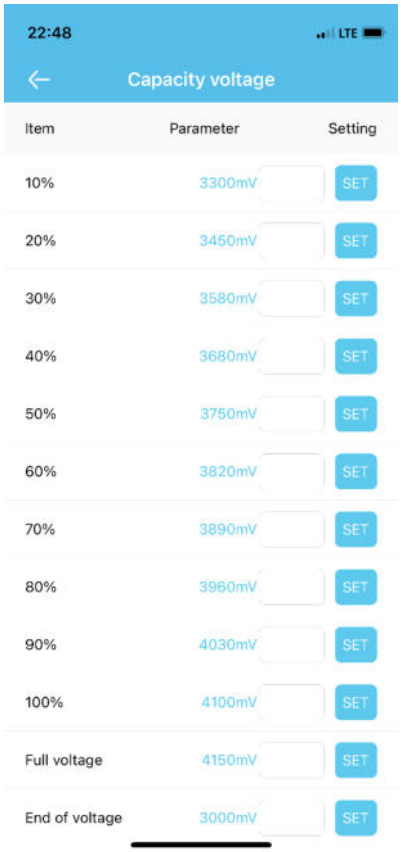

22:46
LTE
Origin Setting
Item
Parameter
Setting
Nominal capacity
21000mAH
SET
Cycle capacity
16800mAH
SET
Full charge capacity
21000mAH
SET
Cell num
12
SET

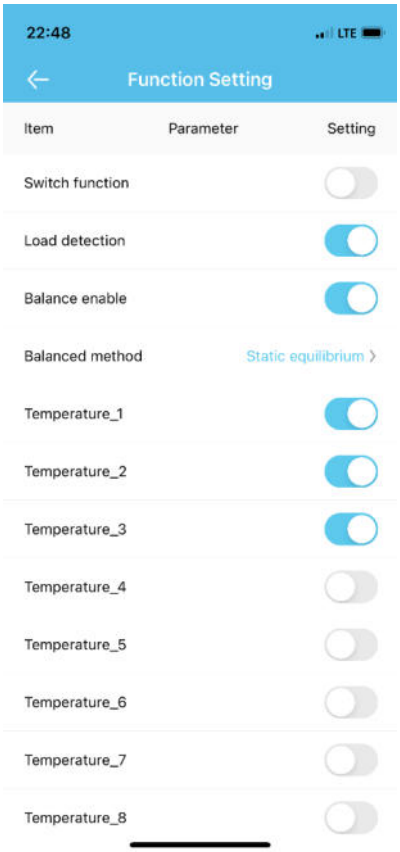

22:46
LTE
Protection Params
Switch to LFP
Item
Parameter
Setting
CellHighVoltProtect
4180mV
SET
CellHighVoltRecover
3800mV
SET
CellHighVoltDelay
2s
SET
CellLowVoltProtect
3000mV
SET
CellLowVoltRecover
3100mV
SET
CellLowVoltDealy
2s
SET
TotalVoltHighProtect
50400mV
SET
TotalVoltHighRecover
48000mV
SET
TotalVoltHighDelay
2s
SET
TotalVoltLowProtect
36000mV
SET
TotalVoltLowRecover
37200mV
SET
TotalVoltLowDelay
2s
SET
HwCellHighVoltProtect
4300mV
SET
HwCellLowVoltProtect
2500mV
SET

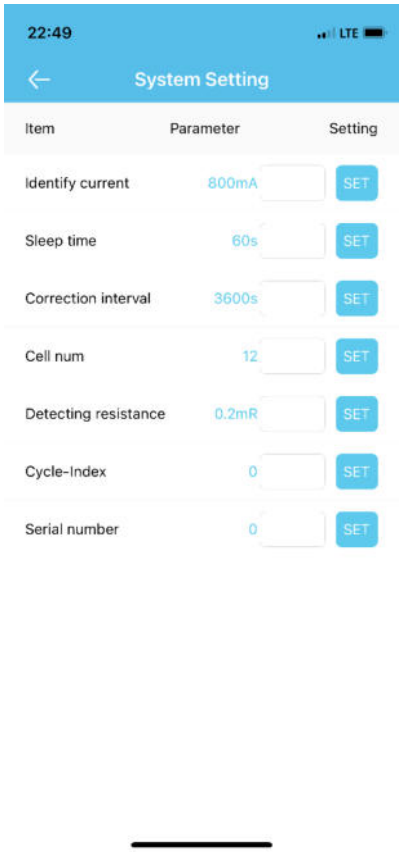

22:47
LTE
Current Settings
Item
Parameter
Setting
Charge overcurrent protection
20000mA
SET
Charge overcurrent delay
10s
SET
Charging overcurrent recovery dealy
32s
SET
Discharge overcurrent protection
-40000mA
SET
Discharge overcurrent delay
30s
SET
Discharge overcurrent recovery delay
32s
SET
Secondary overcurrent protection
175A
Secondary overcurrent delay
500mS
Short circuit protection
700A
Short circuit protection delay
250.0uS
Short circuit protection recovery delay
5s
SET


22:47
LTE
Temperature Settings
Item
Parameter
Setting
Charging high temperature protection
65°C
SET
Charging high temperature recovery
55°C
SET
Charge high temperature delay
5s
SET
Charging low temperature protection
-30°C
SET
Charging low temperature recovery
-25°C
SET
Charging low temperature delay
5s
SET
Discharge high temperature protection
75°C
SET
Discharge high temperature recovery
65°C
SET
Discharge high temperature delay
5s
SET
Discharge low temperature protection
-30°C
SET
Discharge low temperature recovery
-25°C
SET
Discharge low temperature delay
5s
SET


22:47
LTE
Balance Settings
Item
Parameter
Setting
Equalization voltage
3700mV
SET
Balance accuracy
15mV
SET
Turn on equalization
Balanced method
Static equilibrium





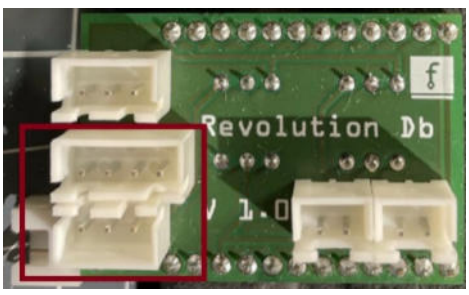


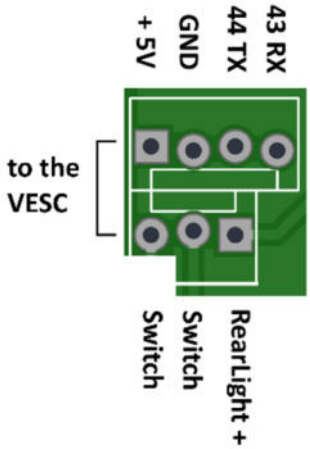
The BMS should show 12 cells if it is connected correctly to the battery.
Take care about the protection parameters.
Unfortunately I couldn't find any instructions for this specific BMS. In the instructions for a similar BMS, at least the principle is well explained:
https://macsbatteries.com/images/bms_manuals/Xiaoxiang_BMS_Manual-20220714.pdf

Step 4 – put it together

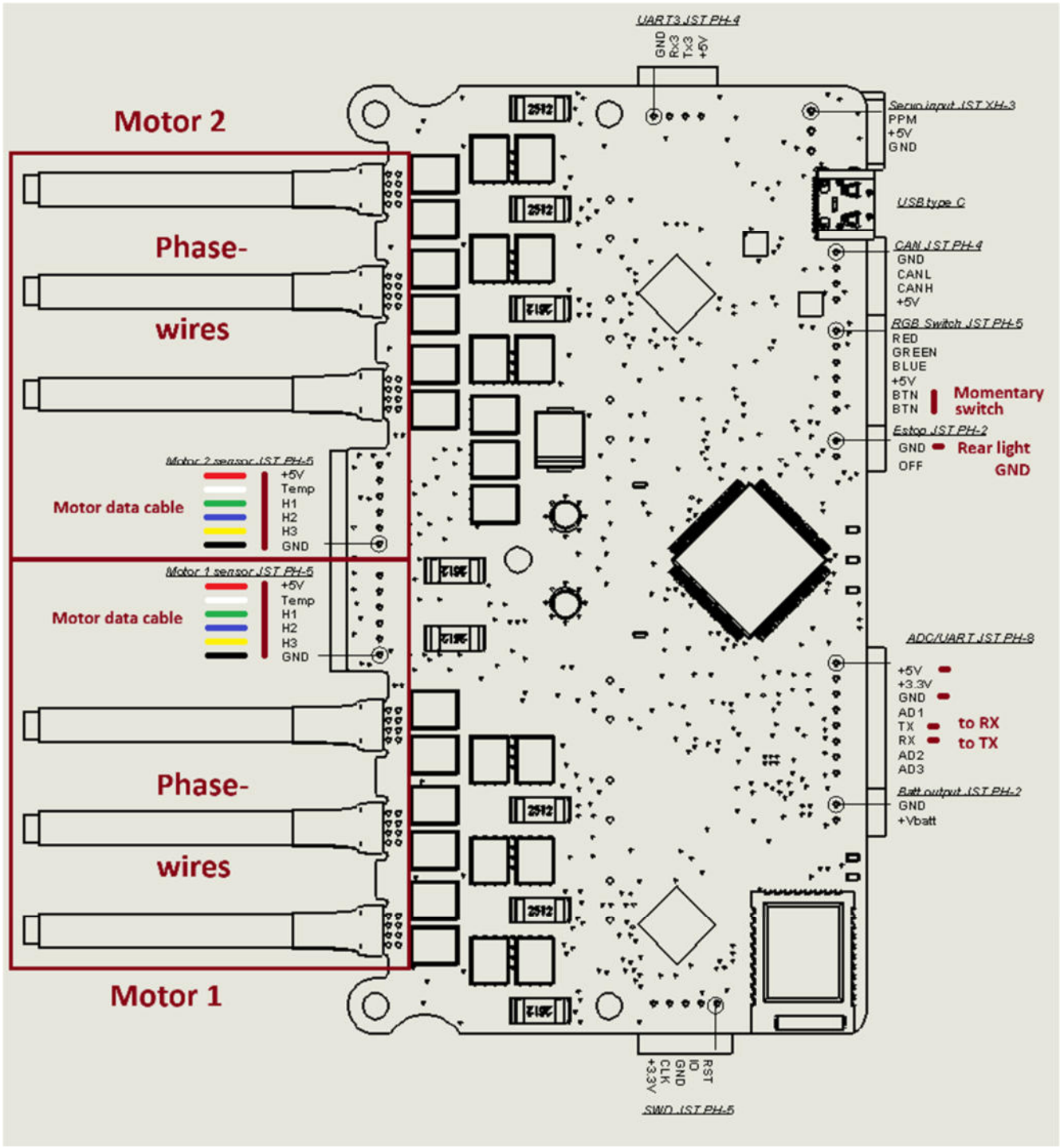
Part list:
VESC based Controller
Cable 24AWG, 7Cores, 2m
2,54mm JST-XH connector (3x 2Pin, 2x 3Pin, 1x 4Pin)
2.00mm JST-PH connector (for connecting the Stormcore, other ESC may have other connectors)
6x 4mm Gold Bullet Connector Female
XT30 connector pair (charging cable)

To strip out the original Data cable, gently pull out the two rubber sealings on the steering stem.
Build a new Data cable from the Dashboard to the VESC and put it into the stem.
Ad one 3 Pin and one 4 Pin JST-XH Plug to the cable to connect the dashboard.





Connection diagram Stormcore D60+



Route the new cable in the side channel till the end of the battery compartment. Cut it for your needs and crimp the wires as shown in the VESC connection diagram. Use the widest connectors as possible from your set to fit well. Use the + Rear light wire from the dashboard cable and the GND from the esc to connect the Rear light.

Cut the Motor Wires from the original Rev ESC and determine the appropriate length you need.
Solder the Gold Connectors to the thick wires and crimp the JST-PH plugs to the motor data cables.
Always add shrink tubes to all connectors. Never leave a cable unprotected!
There is no order for connecting the motor phase wires. The VESC will detect it when running the initialization.
Do not change the connection order after the VESC motor setup. Otherwise you have to repeat this.

Routing the cables:

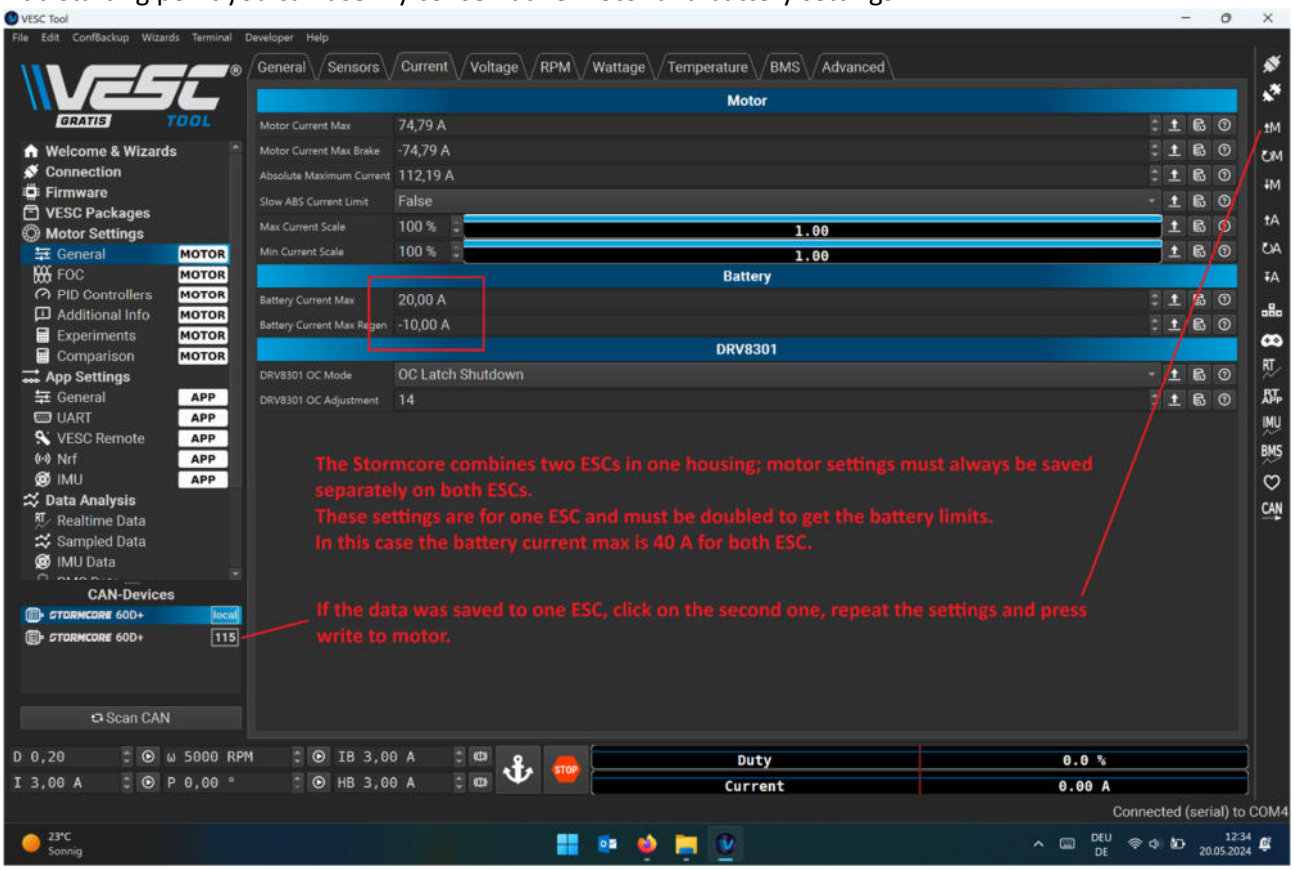
Charging cable:

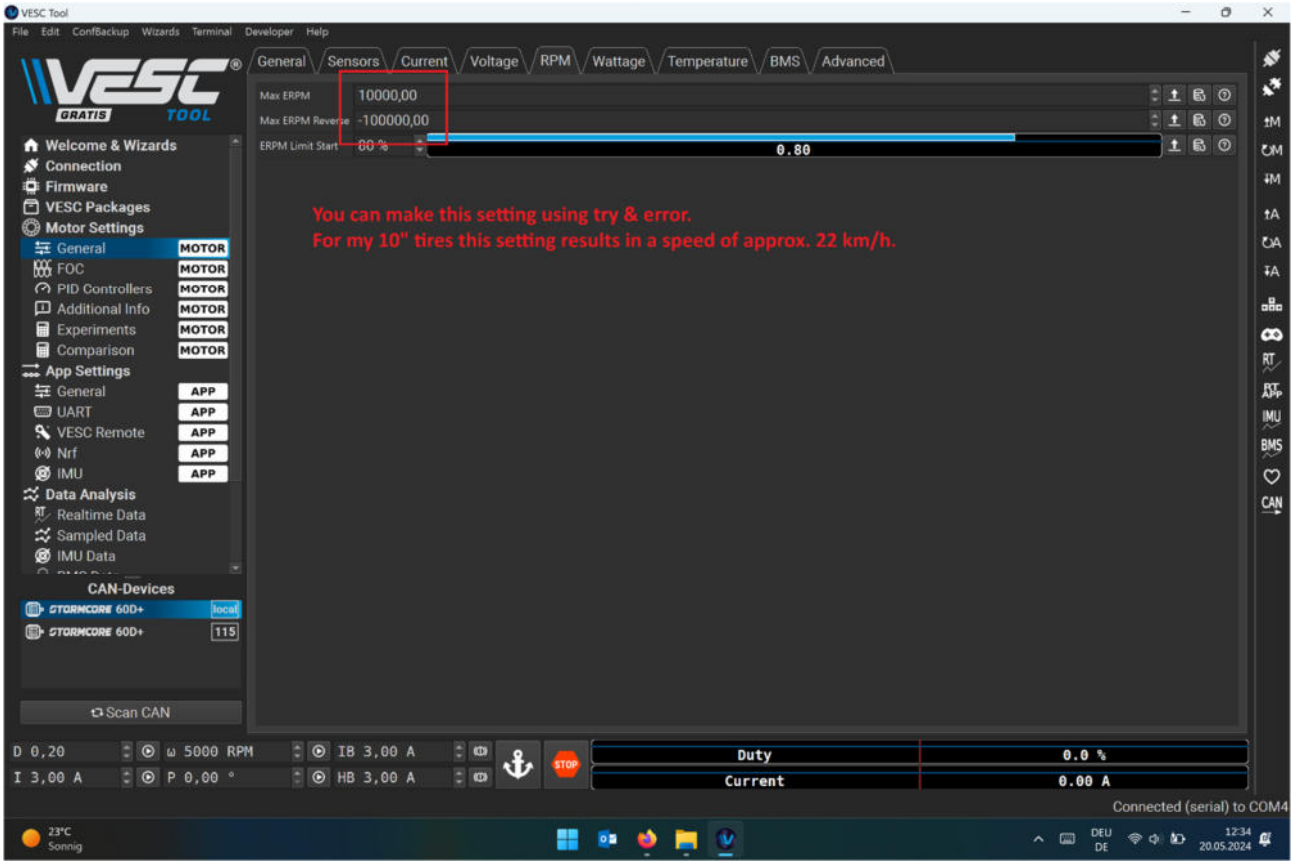
I've routed the motor cable, the charging cable and the dashboard data cable inside the side channels.
I added foam rubber to the side walls of the battery compartment and every corner touched by cables.
At the front I took the rubber edge protection that came with the Rev to protect the battery.

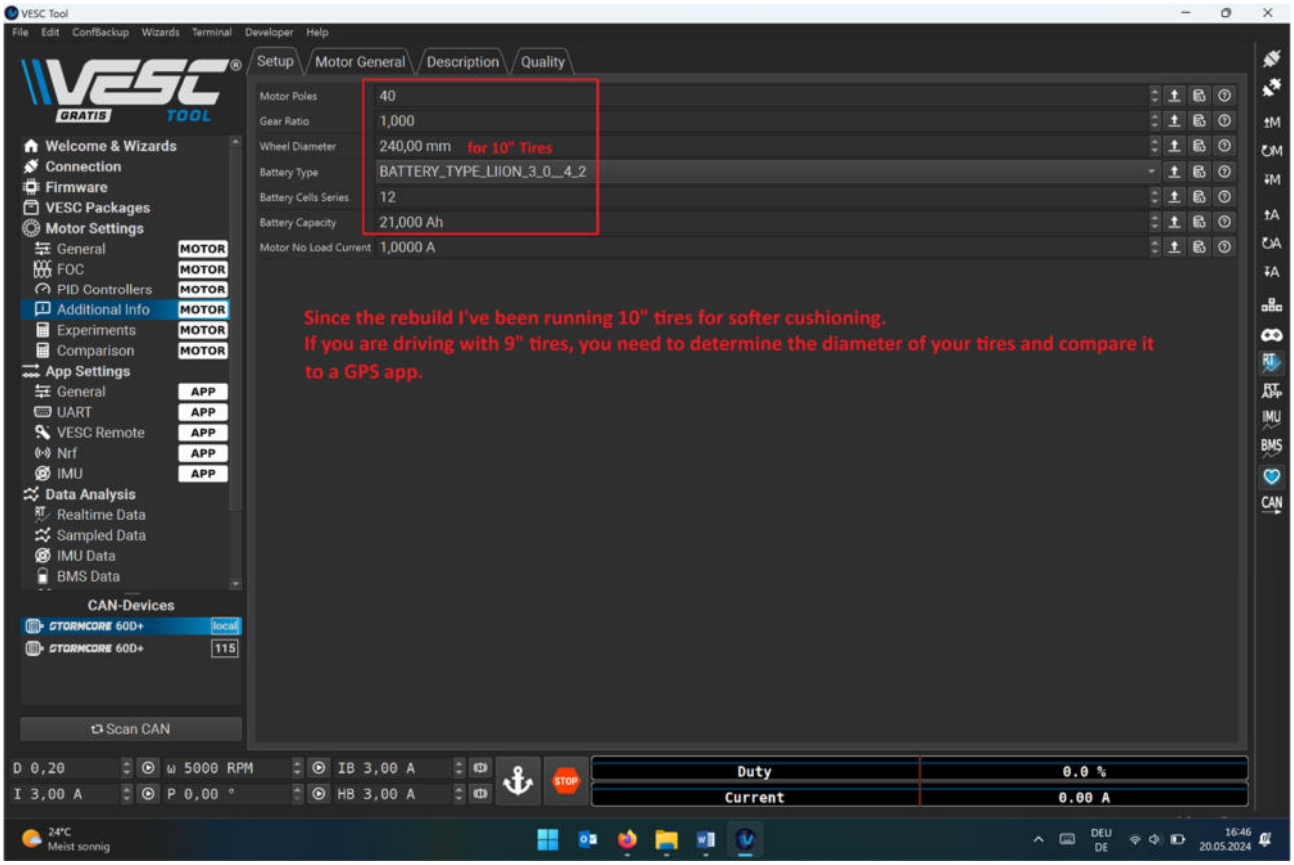
Step 5 – Setup the VESC (Stormcore)

Follow this guide: https://www.youtube.com/watch?v=R6opwxURPDE (Ignore the Part with the Receiver for the Remote and follow the settings of my screenshots). The dashboard works fine with Firmware 6.02. For motor detection use “Large outrunner”. The motor current should not exceed 75A.

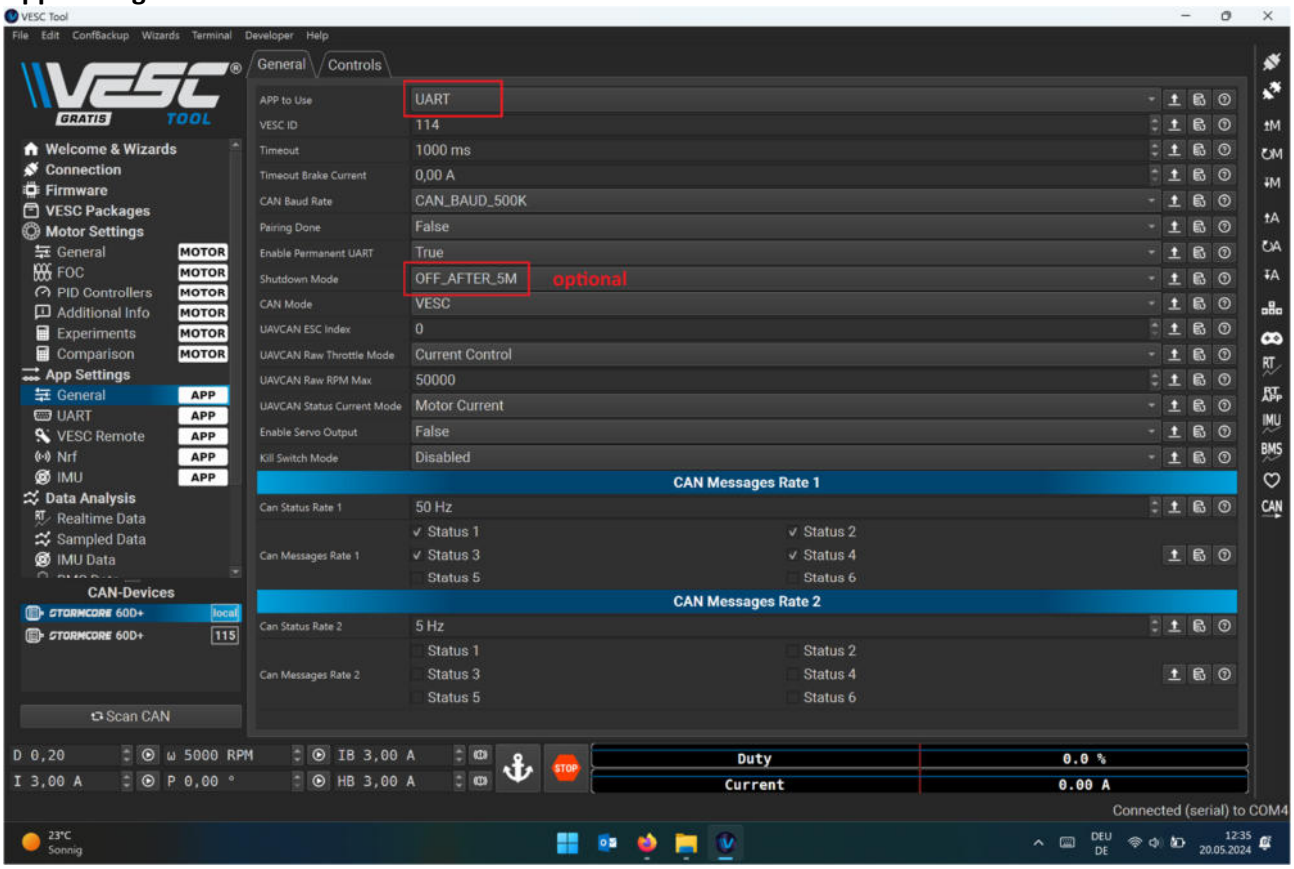
At a starting point you can use my conservative motor and battery settings:

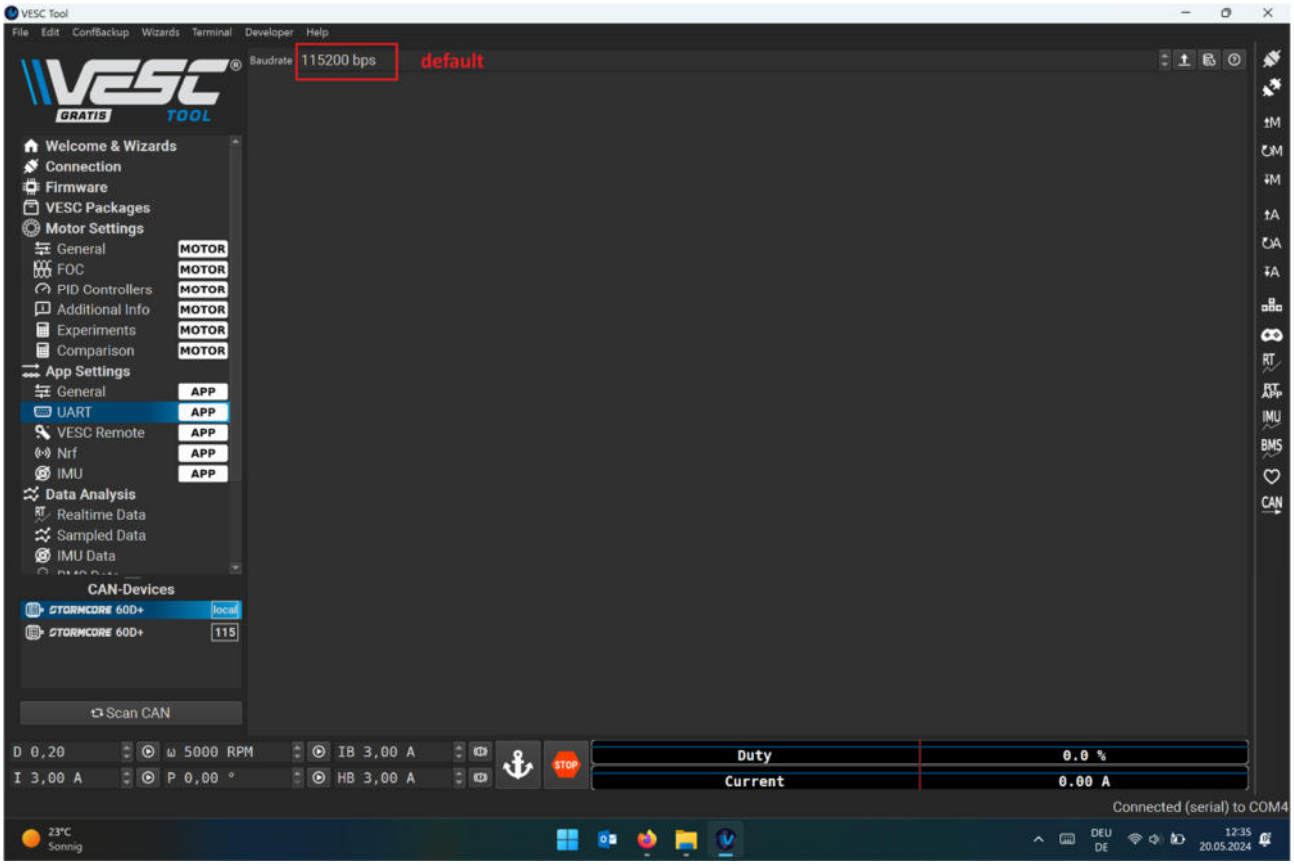




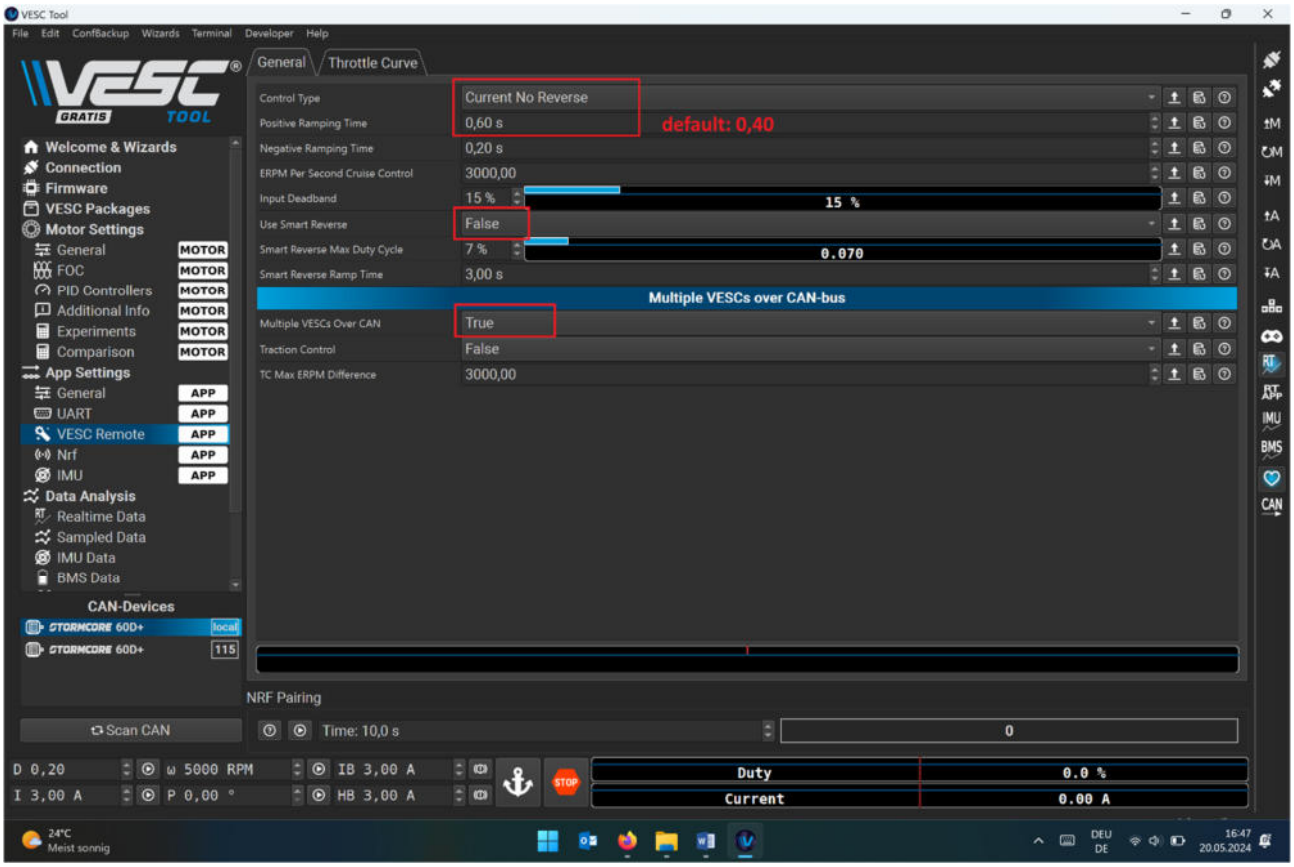


App Settings:





Important: Setup VESC Remote for communication with the dashboard!



Learn more about the VESC settings: https://vesc-project.com/node/183

Step 6 – Dashboard configuration

Take your time to adjust the settings in the cofig.h tab. Most things are simple.

In order to create a somewhat street legal profile, the **thPercentage** maps the mechanical gas path from zero to the entered value. This function is intended for mode 1 to make the throttle response less aggressive. The value must be adjusted to its own needs. The smaller the value, the smoother the acceleration, but also the power is reduced (problem for steep hills). With a value of 100, the throttle path is passed on to the VESC linearly from 0-100% (as in mode 2).
The max speed for mode 1 has to be set independently in the VESC Tool -> Motor Settings – General - RPM.
You could alternatively leave thPercentage at 100% and also reduce the watts in VESC Tool.
(The settings for RPM and Watt will be temporary overwritten when activate mode 2).

I haven't found out if you can set imperial units in the VESC tool. With **setMi**, the speedometer and trip are switched to miles. I think the mobile app will work this way too. It gets the metric values from the VESC and converts it to imperial units.

tachComp: In theory, with the correct wheel diameter, the speedometer and trip should be displayed correctly. With this multiplier you can also set the trip value exactly on a case-by-case basis.

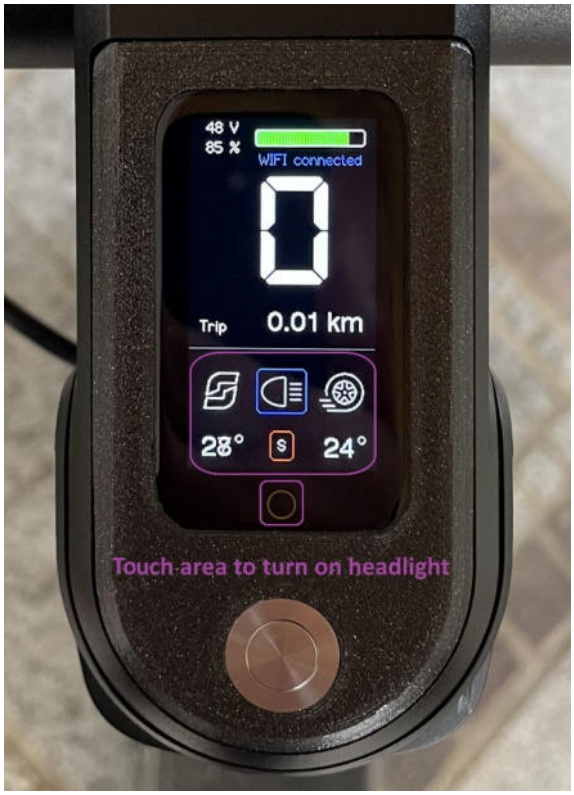
voltdropcomp: Due to the thin cables from the VESC to the dashboard, a voltage drop of approx. 0.3V occurs when the headlight is switched on. This can hardly be avoided as considerably thicker cables would have to be used. Unfortunately, the output value of the Hall sensor on the throttle lever reacts very sensitively to changes in voltage, so that the min-neutral-max value shifts.
With **showThReading** the RAW value of the throttle can be read. This is shown on the display above the text Trip. With the throttle in neutral you can see the difference when the light turns on and off. This difference can be compensated with the variable **thComp** and will be added to the RAW Value when the light is switched on.

I can't remember whether the original REV cuts out engine power when pulling the brake lever. With **stopOnBrake** you can decide if you want to use this function.

To program the dashboard over WiFi, one of the mode 1 or 2 unlock codes must be entered.

Dashboard features:

- Show the Battery level and voltage
- Velocity, Trip, ESC and Motor temp
- Shows if the WiFi is connected, the Headlight is turned on and mode S is activated
- Turn the headlight on and off
- Calibrate the throttle lever
- Always on Rear light / shines brighter when braking
- Two driving modes (is activated with different unlock codes)



Calibrate the throttle lever first:
Turn on the Rev with a short press to the momentary switch.
Enter the code “1000” and follow the instructions on the display. Press OK to save the throttle values permanently.
The dashboard restarts.

How does setting profiles work:
Profiles are not saved in the VESC, some values are temporarily transferred to the VESC (until restart).
In this code the max RPM will be raised in sport mode only.
The smoother acceleration in Mode 1 is achieved by reducing the max throttle value in the Arduino code.

Step 7 – Test ride

Type in the code for Mode 1 and test all functions.
Take an easy test drive.
When everything goes well, shutdown the Rev by pressing the momentary switch 4 seconds.
Power it on again and test Mode 2 (aka Sport Mode).

LONG LIVE THE REV