Student research logbook

Battery system hardware, Julian Meeuwis

Introduction

This logbook is regarding the development of a battery emulator to safely test existing or experimental battery management systems. This logbook is life and will get updated.

Table of Contents

Logbook

14-02-2025

Goal: “deliver BMS”

1. Improve emulator
2. Than BMS

Problems:

- Matlab(ESP32 not compatible) GUI design in matlab, allowed to change to C#
- Oscillation in cell board.

Start plan of approach

16-02-25

Notes:

Research:

“Cell pcbs” are based on LT3080:

Adjustable1.1A Single Resistor Low Dropout Regulator:

Form of linear regulator

Heat problem!

As suggested increase pad. Easy?

“Increase heat pad bottom side to at least 2500mm2”

“Host”:

Includes low drop out regulator?

ESP32

Because they used it before.

Cheap

Logbooks:

Powerpoint?

Gantt chart.

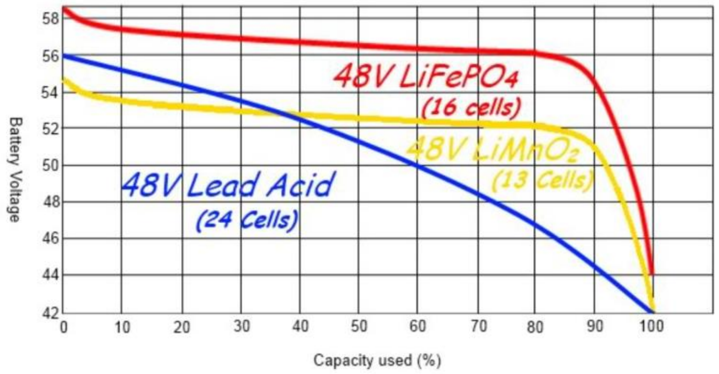
Points:

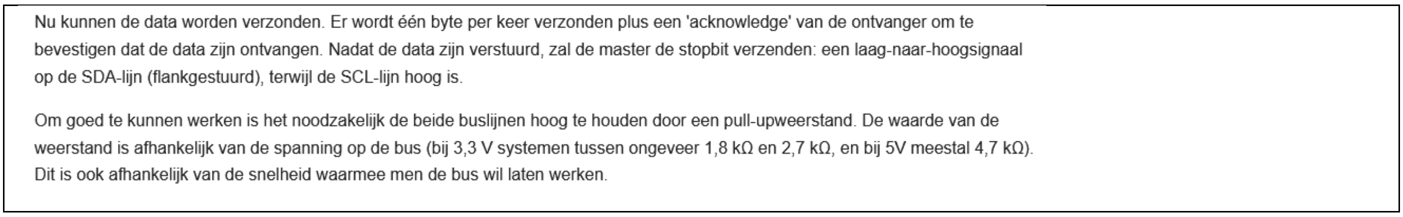
- Need to improve this document with a clear overview of work!
- Use proper diagrams since easy to overview.
- Read testing in final report.

Questions:

- Is the “host board” a bms on its own? Or what is it purpose.
  “Host device is necessary to emulate batteries, it is not a BMS”
- Most regulators are linear low drop out. Why not buck?

20-02-2025




Nu kunnen de data worden verzonden. Er wordt één byte per keer verzonden plus een 'acknowledge' van de ontvanger om te bevestigen dat de data zijn ontvangen. Nadat de data zijn verstuurd, zal de master de stopbit verzenden: een laag-naar-hoogsignaal op de SDA-lijn (flankgestuurd), terwijl de SCL-lijn hoog is.

Om goed te kunnen werken is het noodzakelijk de beide buslijnen hoog te houden door een pull-upweerstand. De waarde van de weerstand is afhankelijk van de spanning op de bus (bij 3,3 V systemen tussen ongeveer 1,8 kΩ en 2,7 kΩ, en bij 5V meestal 4,7 kΩ). Dit is ook afhankelijk van de snelheid waarmee men de bus wil laten werken.

https://docs.arduino.cc/learn/communication/wire/

https://learn.sparkfun.com/tutorials/i2c/all

Feedback 21-02-2025:

Vcontrol lt3080.

Esp32 regarding communication gui

Gantt chart with phases & tasks.

- Simple
- Phases
    - Research
    - Troubleshooting previous hardware
    - Software / communication
    - Fix imple
    - Test
    - Refix

Risks

- Oscillation
- Heat
- Communication

Duration of research. Chris & Kay.

Testing after vacation.

Test plan*

*from old testing.

Planning

23-02-2025

To does:	**Priority**
- Make plan of approach including gantt chart	1
- Clean up document	2
- Constant scheduling	1

Notes:

I2C only used to send commands? Are the results analog, so yes why is this not using I2C

More research regarding linear region mosfet/transistor, biasing.

Vcontrol only allows 1.7% of output current.

PCB “should” work, little fan does wonders, explain about raspberry pi benchmarks

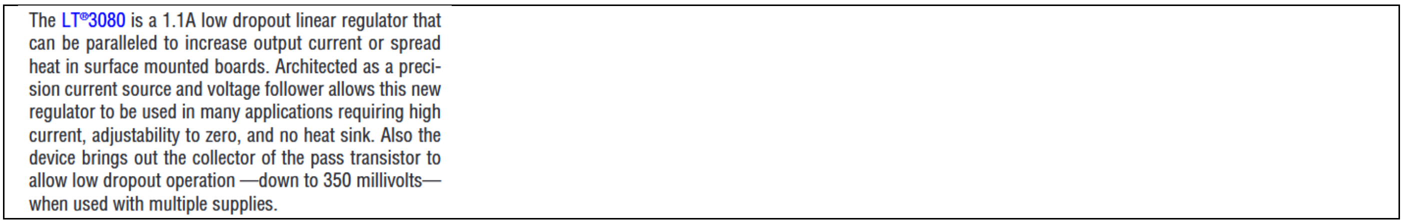
Drop should be over mosfet, most power dissipation too?

12V supply is not good regarding temperature generation.

Research:

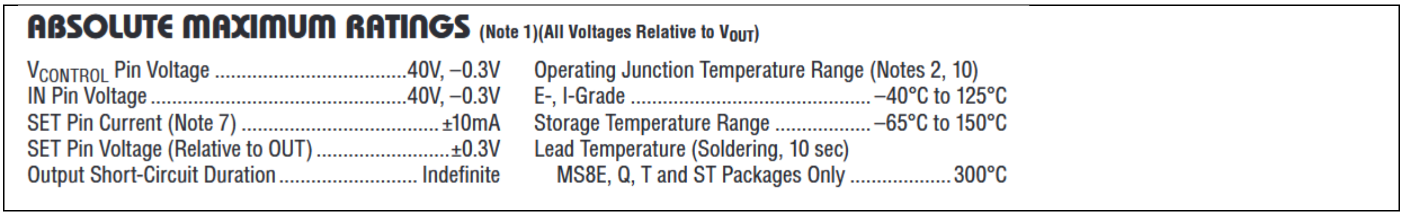
I2C quote: “Since the microprocessor lacks sufficient pins to handle all these inputs and because some input ranges may exceed the microprocessor's 3.3V rating, multiplexers are used. The multiplexers gather all measurements from the cell boards, regulate the signals, and send the specific data of each measurement to the microcontroller (MCU) based on the commands issued to the multiplexers”.

Questions regarding the “1.5V offset” that is being quoted since datasheet shows:



The LT®3080 is a 1.1A low dropout linear regulator that can be paralleled to increase output current or spread heat in surface mounted boards. Architected as a precision current source and voltage follower allows this new regulator to be used in many applications requiring high current, adjustability to zero, and no heat sink. Also the device brings out the collector of the pass transistor to allow low dropout operation —down to 350 millivolts— when used with multiple supplies.

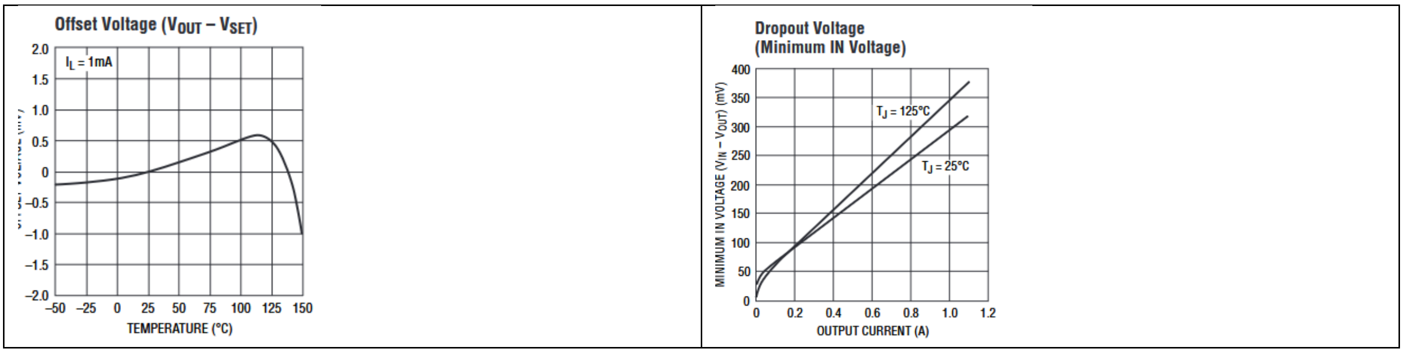
But it need to have an offset, could this not being done digitally?



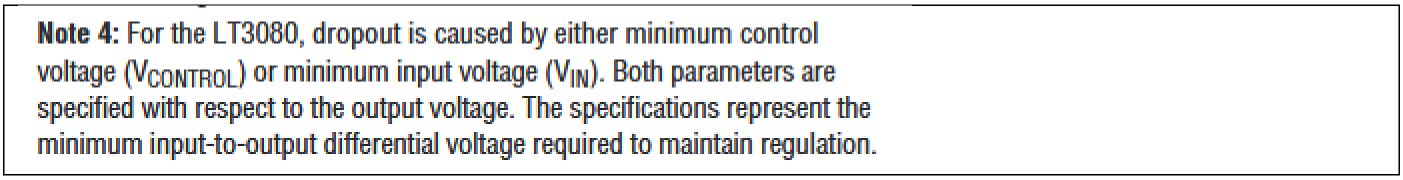
ABSOLUTE MAXIMUM RATINGS (Note 1)(All Voltages Relative to V_{OUT})

$V_{CONTROL}$ Pin Voltage	40V, –0.3V	Operating Junction Temperature Range (Notes 2, 10)	
IN Pin Voltage	40V, –0.3V	E-, I-Grade	–40°C to 125°C
SET Pin Current (Note 7)	±10mA	Storage Temperature Range	–65°C to 150°C
SET Pin Voltage (Relative to OUT)	±0.3V	Lead Temperature (Soldering, 10 sec)	
Output Short-Circuit Duration	Indefinite	MS8E, Q, T and ST Packages Only	300°C

Offset is temperature depended!



There is a good chance that the offset is used to prevent drop out. If this is the case, it can be better prevented by scaling this digitally.

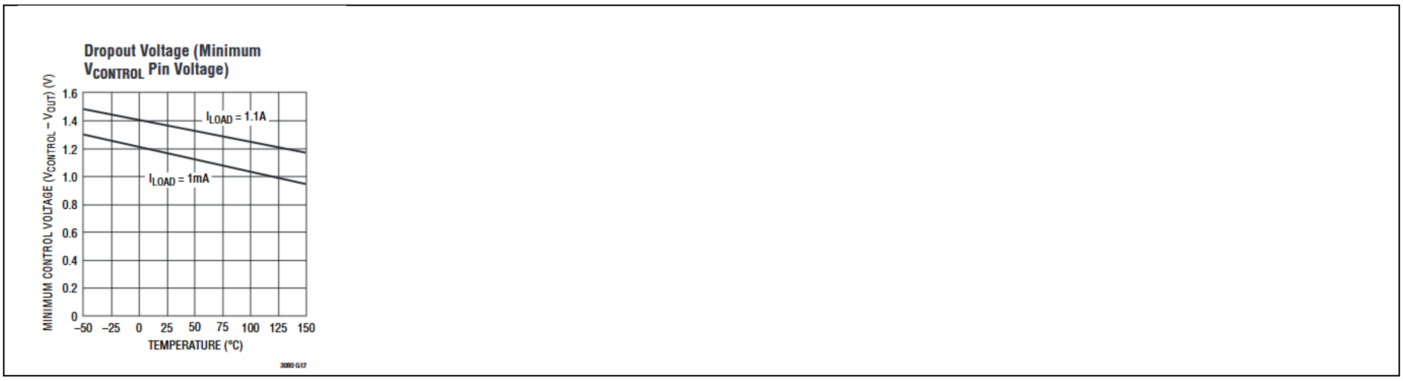


Note 4: For the LT3080, dropout is caused by either minimum control voltage ($V_{CONTROL}$) or minimum input voltage (V_{IN}). Both parameters are specified with respect to the output voltage. The specifications represent the minimum input-to-output differential voltage required to maintain regulation.

https://www.analog.com/media/en/technical-documentation/data-sheets/lt3080.pdf

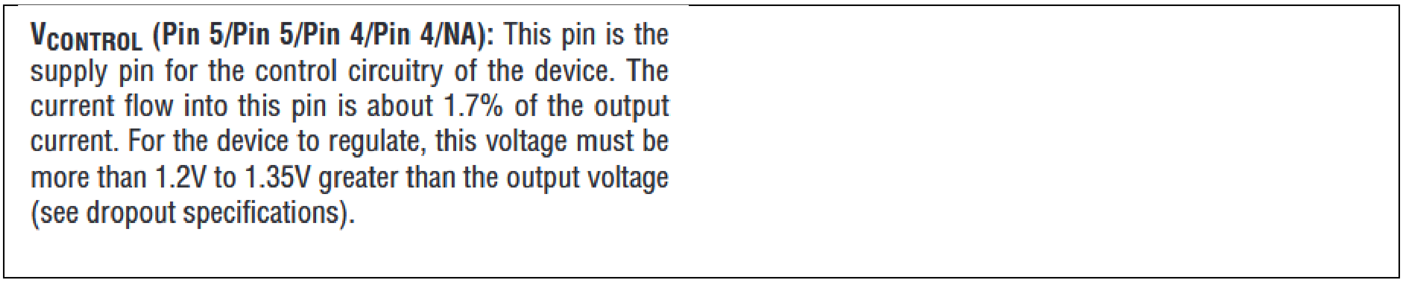
weird behavior in ltspice sim, explain about offset, explain about set current.

On this chart the sim is based:



Matlab part:

“MATLAB and their GUIDE system. Typing guide within the MatLab workspace opens the GUI Designer”



$V_{CONTROL}$ (Pin 5/Pin 5/Pin 4/Pin 4/NA): This pin is the supply pin for the control circuitry of the device. The current flow into this pin is about 1.7% of the output current. For the device to regulate, this voltage must be more than 1.2V to 1.35V greater than the output voltage (see dropout specifications).

24-02-2025

To does:	**Priority**
- Make plan of approach including gantt chart	1
- Clean up document	2

Notes:

First PCB looks solid, but extra bjt is removed since extra current is not necessary.

First PCB looks unfinished

Research:

Huge voltage drop over linear regulators because input voltage is 12V. Possible fix is implementing an interal buckconverter to stepdown this voltage to either 6V or 9V. But LDO drop could be 1.5V possible output is 6V. But even from 12 to 9 volt saves 3V*2A = 6 Watt per cell.

27-02-2025

To does: **Priority**

- Ask about first project

Notes:

“we realized that, for some reason, the ESP32 chip needs to be flashed from time to time. We used an online tool from GitHub to flash it. We don’t exactly know why this is happening—it seems like the ESP32 memory gets full after some time, and we need to reflash the chip. We will investigate this issue later, but for now, we’ll focus on the main goal. So, we can say that the whole functionality of the hardware work just fine” Take in account during testing.

Continued working on plan of action, scheduled where possible.

“New” testplan is to test old hardware to verify points of interests(risks).

Conclusion:

Nmos removed, testing, esp32 failures

To do list for points in gantt

5-10 points

Test plan previous group.

07-03-2025

3.1 Introduction

During the introduction, the way of working regarding student research position is introduced. In this phase, it becomes clear what is expected of the student. The goal of the research is discussed in this.

- Doing research about the project itself
- Diving into previous project work
- Organizing way of working
- Start plan of approach
- Communicate goals

3.2 Research / Testing previous work

The functioning of a battery emulator and BMS system is studied, after which the system of the previous group is critically examined. After sufficient knowledge has been acquired, the previous system is tested with the corresponding test plan. During testing, the already known problems are carefully examined and it becomes clear which functions work well or need more attention.

- Test functionality of previous hardware
- Verify real life working with previous simulations
- Implement fix for know problems and troubleshoot new problems

3.3 Troubleshooting previous work

Some known issues and those identified in testing are being resolved on existing hardware where possible. If this is not possible, solutions are looked at. Some problems emerged during the tests of the previous group. Some of these problems include:

Gui

Communication is one of the problems that has become known from the tests of the previous group. The ESP32 is not compatible with the written GUI (matlab).

Heat

The tests also showed that the previous prototype generated a lot of heat.

Oscillation

Previous testing and simulations have shown that an error was made during the design of the cell modules that resulted in oscillation of the output voltage.

- Testplan executed
- Modify existing hardware?

intermediate results

- Discuss possibilities with client
- Why power resistor?
- 12V input, why not buck?(Test with powersupply jan)

- Decided whenever to continue with previous hardware(Plus fixed) or to redesign hardware.
  - Estimate time consumption(Decision with clients)

3.5 Revise design

Once the problems have become clear, a revision can be considered. Either a complete new system can be designed or previous can be improved.

-

3.6 Testing new implementation

With the introduction of new hardware, of course, new problems come. These errors are discovered through testing. The system is tested with the usability for the client in mind.

10-03-2025

To does:

- Take micro usb cable
- Go to Chris regarding drive he made for me, looks already being used by other group
- Sort items to upload

Notes:

Make new testplan(in excel)? Figure out input voltage cellboard and master board.

Did more research regarding existing PCBs seems to be complete but why DC offset?

12-03-2025

To does:

- Import code in VScode

Notes:

1. Matlab GUI is there but does not seem to function. Converted the old Arduino.ino to work with VScode. Old Arduino code is not responsible since use of a lot of delays. When converting to VScode problem since this type of esp32 not available as template(esp32-s3-wroom-1-N4R8). Old code not responding to input but cell boards seems to function properly.
2. After using the code in Arduino IDE, it seem to function properly.
3. Check if main board need external powersupply, doesn't look like it.

The matlab doesn't seem to function. It is a good idea to built the GUI in matlab since most engineers have this installed.

Tried to convert existing code to Vscode for easy of use but encounter problems with uploading the code. It seems this ESP32 model has no template.

13-02-2025

To does:

Done/Notes:

Emulator seems to work as expected from previous videos. In video it is not visible if cell is connected straight to powersupply or gets it power from main board. If the second is the case, how is this possible? Since cell boards use 2 A each? The high voltage drop results in a very high loss. A fix for this could be an onboard buck converter. But what is smarter:

- To place 1 on each cell board?
- To place 1 on main board?
- To place multiple on main board?

20-03-2025

To does:

- Continue POA

Notes/done:

New cellboards are used. Everytime hitting current limit.

Steps:

1. Observe Schematic
2. Fulfill testplan
3. Write conclusions

Host board:

Conclusion from previous testing is that wiper is reversed. My conclusion is that current wiper works as expected but Voltage wiper is inversed

Isolated DC/DC is used for -12V supply.

I2C line is very very unstable. Mostly work when connected cells after booting the whole system:

1. Power Esp32
2. Power whole system with 12V
3. Than connect cells

Cell board:

Voltage board A doesn't drop to 1 V stays at 1.1V even when wiper is 243(coded maximum)

Voltage board B does reach 0.9V when maximum allowed current.

BIG problem. Schematic doesn't match with PCB +12V and -12V are mixed.

Global:

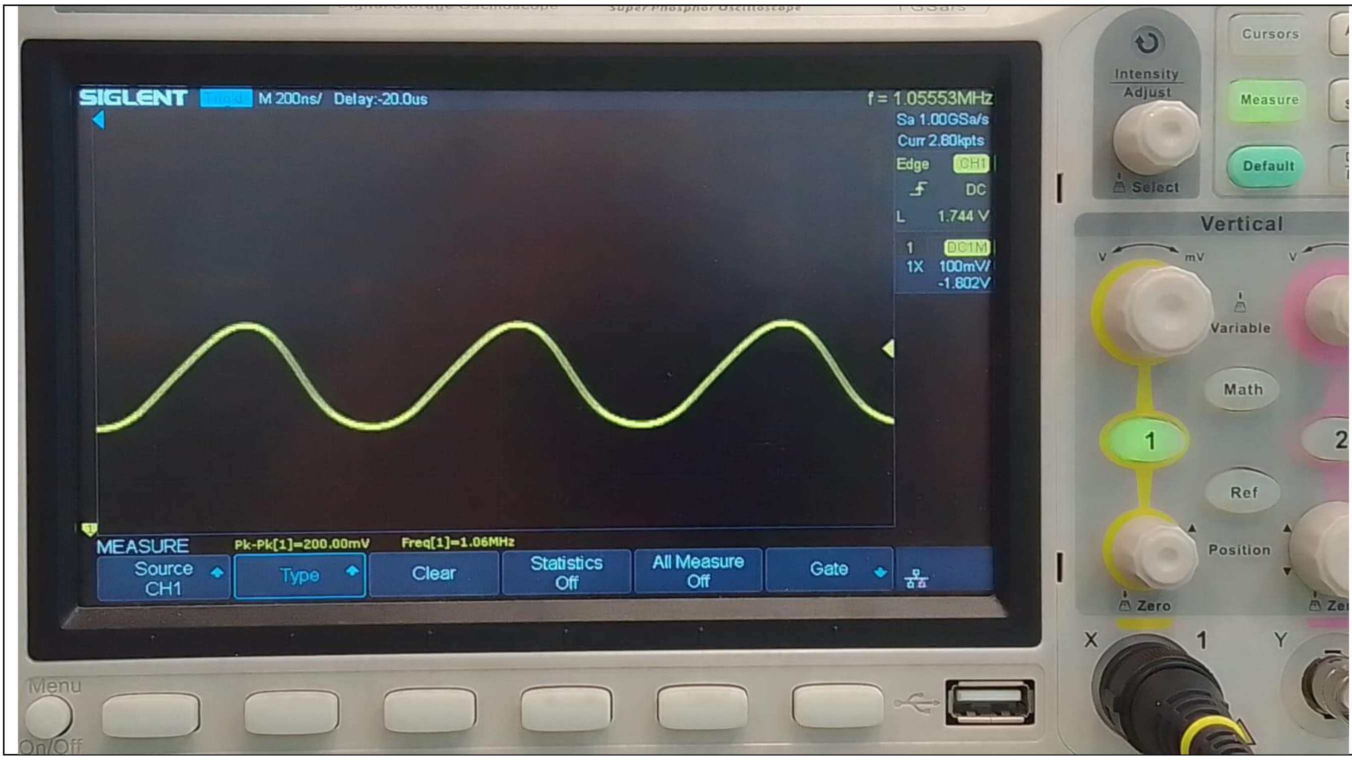
Boards are damaged alot because boot causes a lot of current rush(Wipers reset). The heat bends the PCBs. This is visible as white powder on top? You can even hear this!

Board seem to function fine. Is removing offset circuit possible**?**

Extra DC/DC could be replaced with DC for powering cellboard

I2C does have some problems!

Verified oscillation

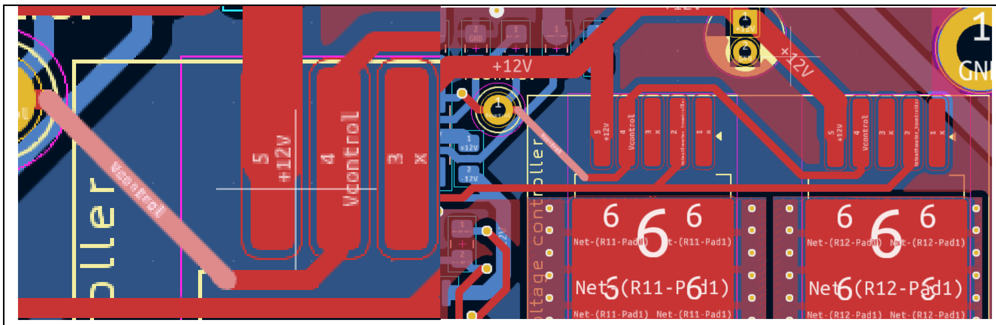


Check wiper configuration with series resistance.

Ask about special days.

Make two tests 1. Lower input voltage at 12V rail, and than remove offset and short.

Correction to myself, -12V IS used. Opamp reference?



Cut and short

27-03-2025

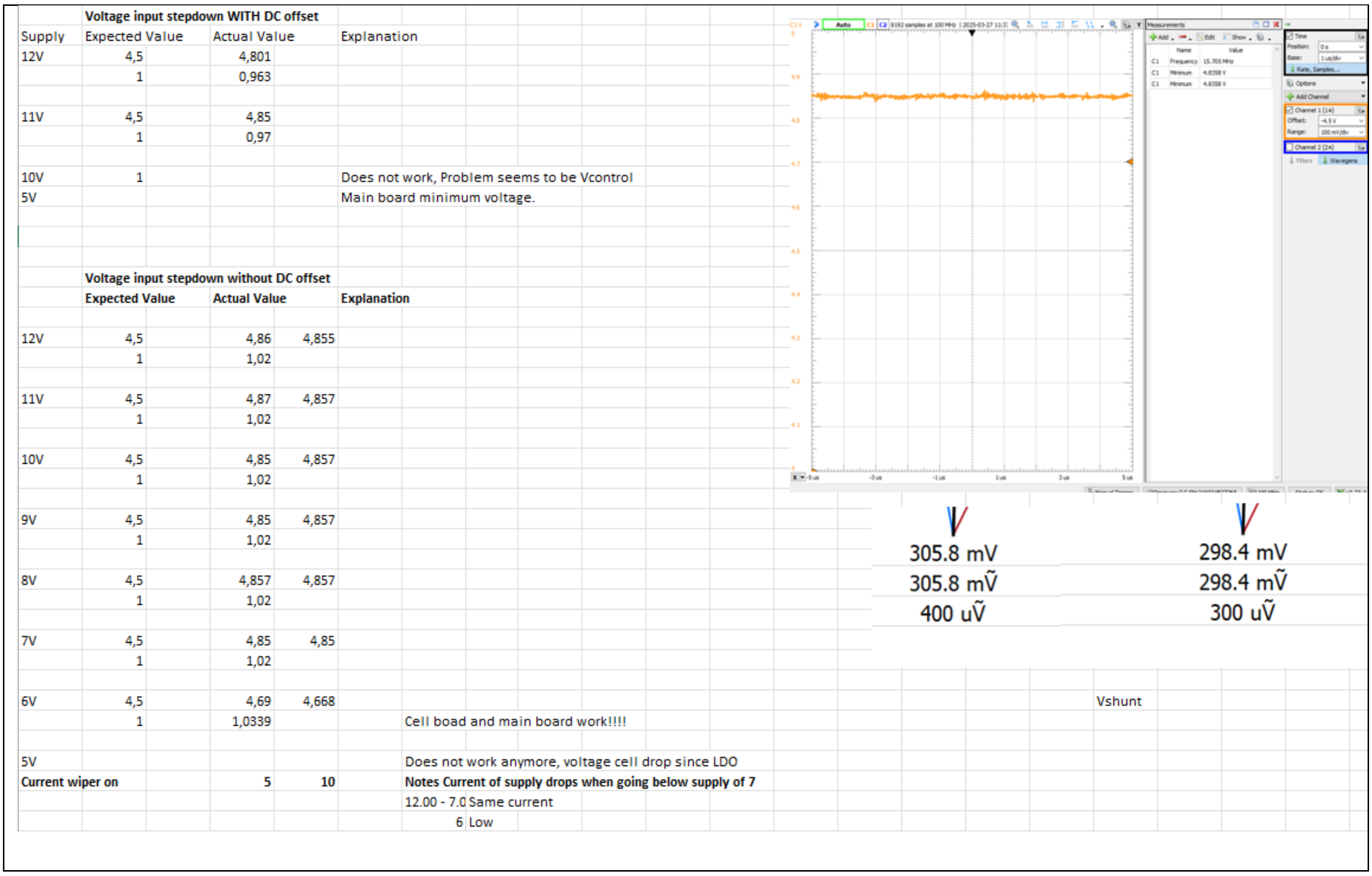
To does:

- Continue POA
- Check wiper configuration with series resistance.
- Try I2C with delay

Notes/done:

-12V line is actually used, only for opamps. Why is this necessary?

Some emails about special days with Chris?



Voltage input stepdown WITH DC offset

Supply	Expected Value	Actual Value	Explanation
12V	4,5	4,801	
	1	0,963	
11V	4,5	4,85	
	1	0,97	
10V	1		Does not work, Problem seems to be Vcontrol
5V			Main board minimum voltage.

Voltage input stepdown without DC offset

	Expected Value	Actual Value		Explanation	
12V	4,5	4,86	4,855		
	1	1,02			
11V	4,5	4,87	4,857		
	1	1,02			
10V	4,5	4,85	4,857		
	1	1,02			
9V	4,5	4,85	4,857		
	1	1,02			
8V	4,5	4,857	4,857		
	1	1,02			
7V	4,5	4,85	4,85		
	1	1,02			
6V	4,5	4,69	4,668		
	1	1,0339		Cell boad and main board work!!!!	
5V				Does not work anymore, voltage cell drop since LDO	
Current wiper on		5	**10**	**Notes Current of supply drops when going below supply of 7**	
				12.00 - 7.0	Same current
				6	Low

305.8 mV	298.4 mV
305.8 mṼ	298.4 mṼ
400 uṼ	300 uṼ

Vshunt

Hardware with “improvements” are tested. First the power input is lowered till the point that the system reacts on it (in a negative way). The voltage was lowered and problems occurrent already at 10V, the voltage (cell board)dropped to 300 mV.

Then offset circuit is cut off and Vcontrol is shorted with power input. After this the tests were repeated. The system reacted way better when input voltage was dropped. Till 7V the system reacted exactly the same as 12V input.

At 6V the system seamed to be affected by the lower voltage, but this is minimal. When designing new PCB test again with 7V and 6V and check of this deviation is neglectable.

All the tests were done twice but with a different wiper setting. Conclusion is that amount of current through system does not affect the outcome (voltage).

28-03-2025

Notes:

Quick notes

Meeting:

Recreate system in LTspice

I2C

Heatsink

Think about I2C increasing devices

04-04-2025

To does:

- Continue POA
- Upload files
- Check wiper configuration with series resistance.
- Update / improve ltspice
- Try I2C with delay

Notes/done:

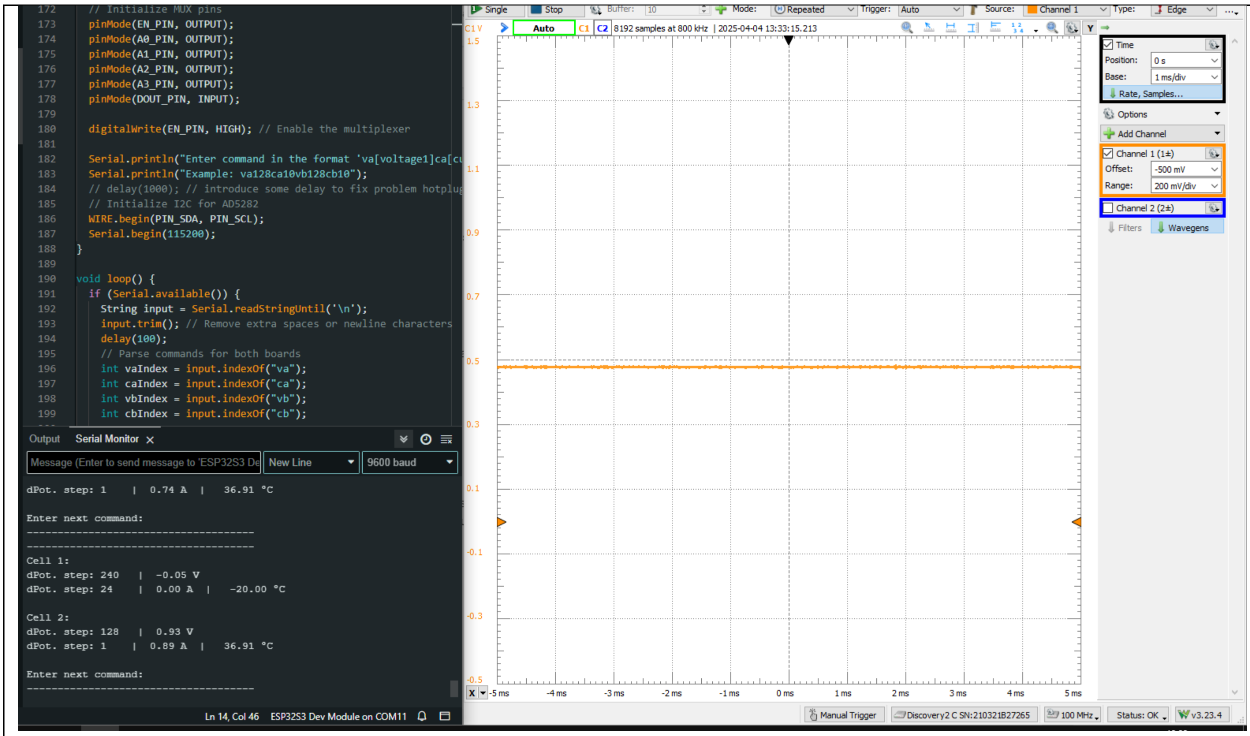
Plan is to first adjust and “improve” cellboards and than continue with the rest.

1. LTspice
2. Testin I2C

Testing:

During testing the maximum output voltage was reached with a supply of 7V. However the current maximum was not corresponding to the set value. When increasing supply to 7.2V it seemed to work normal.

Continued testing with 8V supply



```
  // Initialize MUX pins
  pinMode(EN_PIN, OUTPUT);
  pinMode(A0_PIN, OUTPUT);
  pinMode(A1_PIN, OUTPUT);
  pinMode(A2_PIN, OUTPUT);
  pinMode(A3_PIN, OUTPUT);
  pinMode(DOUT_PIN, INPUT);

  digitalWrite(EN_PIN, HIGH); // Enable the multiplexer

  Serial.println("Enter command in the format 'va[voltage1]ca[cu
  Serial.println("Example: va128ca10vb128cb10");
  // delay(1000); // introduce some delay to fix problem hotplug
  // Initialize I2C for AD5282
  WIRE.begin(PIN_SDA, PIN_SCL);
  Serial.begin(115200);
}

void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim(); // Remove extra spaces or newline characters
    delay(100);
    // Parse commands for both boards
    int vaIndex = input.indexOf("va");
    int caIndex = input.indexOf("ca");
    int vbIndex = input.indexOf("vb");
    int cbIndex = input.indexOf("cb");
```

Message (Enter to send message to 'ESP32S3 De   New Line   9600 baud

```
dPot. step: 1    |  0.74 A  |   36.91 °C

Enter next command:
-------------------------------------
-------------------------------------
Cell 1:
dPot. step: 240  |  -0.05 V
dPot. step: 24   |  0.00 A  |   -20.00 °C

Cell 2:
dPot. step: 128  |  0.93 V
dPot. step: 1    |  0.89 A  |   36.91 °C

Enter next command:
-------------------------------------
```

Ln 14, Col 46   ESP32S3 Dev Module on COM11

Buffer: 10   Mode: Repeated   Trigger: Auto   Source: Channel 1   Type: Edge

Auto   C1   C2   8192 samples at 800 kHz | 2025-04-04 13:33:15.213

☑ Time   Position: 0 s   Base: 1 ms/div

☑ Channel 1 (1±)   Offset: -500 mV   Range: 200 mV/div

☐ Channel 2 (2±)

Discovery2 C SN:210321B27265   100 MHz   Status: OK   v3.23.4

Vshunt

Continued with longer testing maximum current → 1 LDO got extremely hot, other seemed fine.

Quick notes

- Voltage sensor works correct
- Current sensor not!
- I2C channel looks fine but not functioning
- Rush current!
- Continue matlab gui since useful for self-development and industrial standard.
- When redesigning PCB, ONLY REMOVE unnecessary parts, dont move / add parts. Also order new main bord since damage done testing.
- Continue POA
- Upload files
- Check agenda & logbook

Idea to make ADC on cellboards and info over I2C.

Testing

System specification	Rational	Test description (Tests are compared example)	Expected result	Actual Result
Supply/Input voltage 12V - 5V	Lower power consumption	1. Set up normal environment 2. Cycle down from 12V to 5V	System should function same as 12V input	Vout of cellboard start deviating from example under 6-7V supply. Iout of cellboard start deviating from example under 7.2V
1.5V offset	Solve oscillating problem	1. Remove offset by removing trace 2. Test working	System should function as normal	System works OK without offset circuit
Volt sensor	Verify working	1. Measure output 2. Compare to value in software	Measured output and value in software are the same	As expected.
Current sensor	Verify working	1. Measure output 2. Compare to value in software	Measured output and value in software are the same	As expected.
Output voltage	Verify working	1. Measure output 2. Compare to simulation	Measured output compared to simulations are identical	As expected,
Output current	Verify working	1. Measure output 2. Compare to simulation	Measured output compared to simulations are identical	As expected till 1A, maximum current is 1.5A and 2 A is expected, reason **not** found yet. This wasn't tested before

Problems

Function	Problem	Rational	Impact	Solution
Thermal	Runaway		Critical	Series resistance (few hundred milli ohms) ~~Mosfet instead of transistor.~~
Potentiometer	Rush current	Board and component damage.	Critical	Memory (how much rewriting?).
Potentiometer / I2C	No response after “full” reboot		Critical	-
Thermal	Heat generation	Heat decreases overall efficiency.	Medium	Lower supply voltage, less drop over LDO. Increase pad size and improve overall thermal surface (Vias)
Sensor	Wrong current scaling		Small	Adjust software scaling. OR Adjust hardware scaling.
Potentiometer	System does not reach designed maximum current		Small/Medium	-

Tips:

- (Mosfet resistance lower? Gate source voltage, this is regarding current control)

Continue POA with intermediate results. Upload in drive, and send ltspice

Other improvements (non-functional)

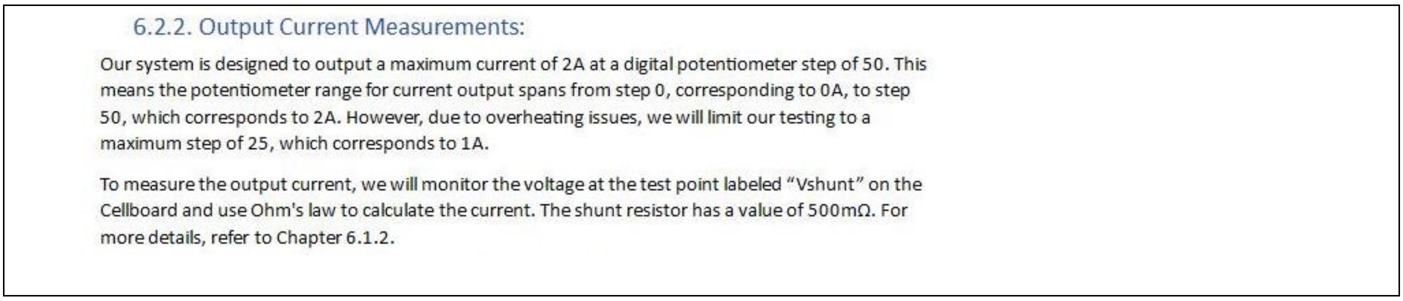
Function	Idea	Rational	Impact
User-interface	Functional GUI	Easy use for customer	Large
Potentiometer	Change stepsize current controller	Accurate and suggesting from previous group	Small
Supply	Remove -12V	Simplify design	Medium
Multiplexer	Get rid of multiplexer	Simplify design	Large (lot of time)

06-04-2025

To does:

- Continue POA
- Upload files
- Check wiper configuration with series resistance.
- Update / improve ltspice
- ~~Try I2C with delay~~
- Check current sensor / shunt Since change in supply voltage result in not able to reach maximum current.

Notes/done:

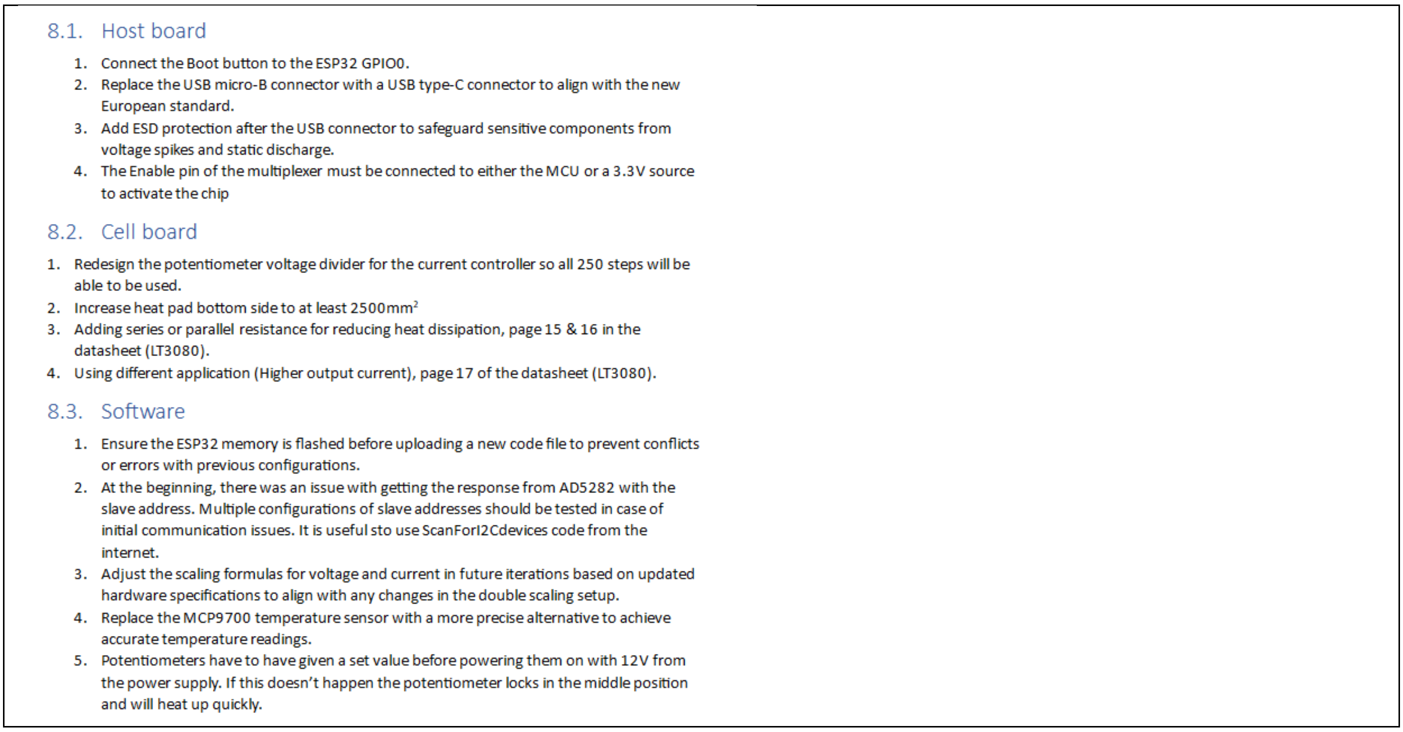


6.2.2. Output Current Measurements:

Our system is designed to output a maximum current of 2A at a digital potentiometer step of 50. This means the potentiometer range for current output spans from step 0, corresponding to 0A, to step 50, which corresponds to 2A. However, due to overheating issues, we will limit our testing to a maximum step of 25, which corresponds to 1A.

To measure the output current, we will monitor the voltage at the test point labeled "Vshunt" on the Cellboard and use Ohm's law to calculate the current. The shunt resistor has a value of 500mΩ. For more details, refer to Chapter 6.1.2.

Test board at absolute maximum.



8.1. Host board

1. Connect the Boot button to the ESP32 GPIO0.
2. Replace the USB micro-B connector with a USB type-C connector to align with the new European standard.
3. Add ESD protection after the USB connector to safeguard sensitive components from voltage spikes and static discharge.
4. The Enable pin of the multiplexer must be connected to either the MCU or a 3.3V source to activate the chip

8.2. Cell board

1. Redesign the potentiometer voltage divider for the current controller so all 250 steps will be able to be used.
2. Increase heat pad bottom side to at least 2500mm^2
3. Adding series or parallel resistance for reducing heat dissipation, page 15 & 16 in the datasheet (LT3080).
4. Using different application (Higher output current), page 17 of the datasheet (LT3080).

8.3. Software

1. Ensure the ESP32 memory is flashed before uploading a new code file to prevent conflicts or errors with previous configurations.
2. At the beginning, there was an issue with getting the response from AD5282 with the slave address. Multiple configurations of slave addresses should be tested in case of initial communication issues. It is useful sto use ScanForI2Cdevices code from the internet.
3. Adjust the scaling formulas for voltage and current in future iterations based on updated hardware specifications to align with any changes in the double scaling setup.
4. Replace the MCP9700 temperature sensor with a more precise alternative to achieve accurate temperature readings.
5. Potentiometers have to have given a set value before powering them on with 12V from the power supply. If this doesn't happen the potentiometer locks in the middle position and will heat up quickly.

07-04-2025

To does:

- Continue POA
- Upload files
- Check wiper configuration with series resistance.
- Update / improve ltspice
- ~~Try I2C with delay~~
- Check current sensor / shunt Since change in supply voltage result in not able to reach maximum current.

09-04-2025

Notes/Done:

The schematic of the cellboards indicates in a controller voltage supply with a controlled input(charge) current. However the output current seems to be left uncontrolled.

Is this the purpose? So yes, what is **the end goal for this emulator**. Like what will it be used for.

Question regarding capacitor placed close to LT3080 in combination with bigger R for runaway control.

Roadplan compared to gantt chart

10-04-2025

To does:

- Continue POA
- Upload files
- Check wiper configuration with series resistance.
- Update / improve ltspice
- ~~Try I2C with delay~~
- Check current sensor / shunt Since change in supply voltage result in not able to reach maximum current.

Notes/Done:

Matlab has a GUI design toolbox but also an app design toolbox. Which one is more suitable for my use case. Inbuilt matlab app designer seems to have most support.

	App designer (Matlab)	GUIDE (Matlab)	C#	C++
Compatibility (client & developer)	Good	OK	OK	OK
Implementation time (With my experience)	OK	OK	OK	Bad
Ease of use (Client)	GOOD (Assumption client has Matlab)	OK? (Possible incompatible)	OK (windows security error?)	OK
Available documentation	Good	OK (Outdated)	GOOD	GOOD

Communication with ESP32

	Serial	UART	Bluetooth / WIFI	1-wire
Ease of use (Client)	Good	Bad (Need for extra adapter)	OK	Bad
Compatibility (client & developer)	Good	Good	OK	Bad
Implementation time (With my experience)	Good	OK	Bad	Bad
Available documentation	Good	Good	Good	OK
Backwards compatible with Arduino IDE (console)	Good	-	-	-

Quicknotes:

What i did:

- POA
- Testing + proper notation of it
- Software solution I2C
- GUI solution
- Simplify LTspice simulation with small improvements
- Uploaded files
- Organised LOG

What i tried (regarding I2C)

- Change pullup resistors (smaller and bigger value)
- Delay when initialising

Biggest problems right now (which influence progress)

- Limited output current
- I2C connection difficulties.

What would be next? Also regarding output current(explain with diagram please)

Gant chart?

Risico’s (hoofdstuk 7): Heat => Overheating. Action: redesign the passive or active cooling solution.

14-04-2025

To does:

- Continue POA
- Upload files
- Check wiper configuration with series resistance.
- Update / improve ltspice
- Not able to reach maximum current.

Notes/Done:

Check if mosfet sim is same as PCB. Test Thursday afternoon.

Micro usb power line

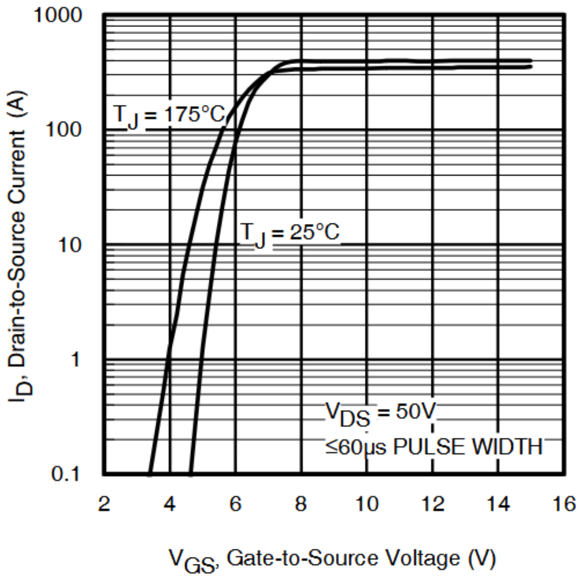
17-04-2025

To does:

- Continue POA
- Upload files
- Check wiper configuration with series resistance.
- Update / improve ltspice
- Not able to reach maximum current.
- Reach out to Ruud or Jochem (Broken boards & I2C problems)
- Test I2C with Arduino

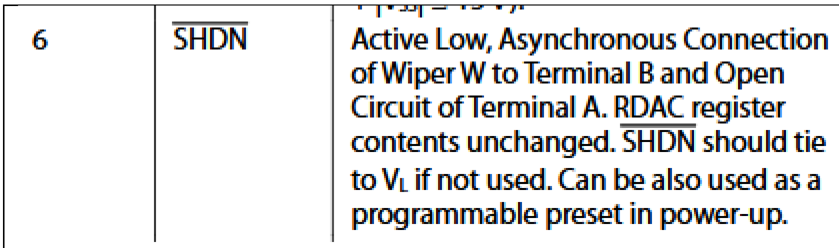
Notes/Done:

Fig 1. Typical Output Characteristics



IRFS4115

Doesn't seem to fully open. Test on opamp (TP9 & TP10).



6	$\overline{SHDN}$	Active Low, Asynchronous Connection of Wiper W to Terminal B and Open Circuit of Terminal A. RDAC register contents unchanged. $\overline{SHDN}$ should tie to V_L if not used. Can be also used as a programmable preset in power-up.

Boot up wiper setting

Tested to maximum current with supply voltage of 8 – 12 V

Old cellboard (8V supply)

Shunt	Step	Mosfet
1.5A	30	6.02V
1.6A	40	6.2V
1.6S	50	6-7 V

Notes:

Started to oscilate.

New cellboard (8V supply)

Shunt	Step	Mosfet
1.5A	30	6.0V
1.7A	40	~ 6.4V
2A	50	~ 6.5V

During testing the board for it maximum capacity it seems to be malfunctioning after 1.5A sinking current.

To verify this, other board modified (removed offset) and conclusion is that testing board is defect.

To properly scale current wiper, inverted current setup

Memory jumper*

Schedule meeting Harold Tuesday

Why broken boards?

Next testing:

- Add series R to PCB on “broken pcb”
- Memory jumper
- Pull up resistors are not optimal

23-04-2025

To does:

Notes:

I2C scan didn’t work because wrong pin assignment

I2C:

- Hard reset hardware
- Hard reset software
- Pull ups? 1K?

24-04-2025

To does:

- Make block diagram regarding frame (I2C)
-

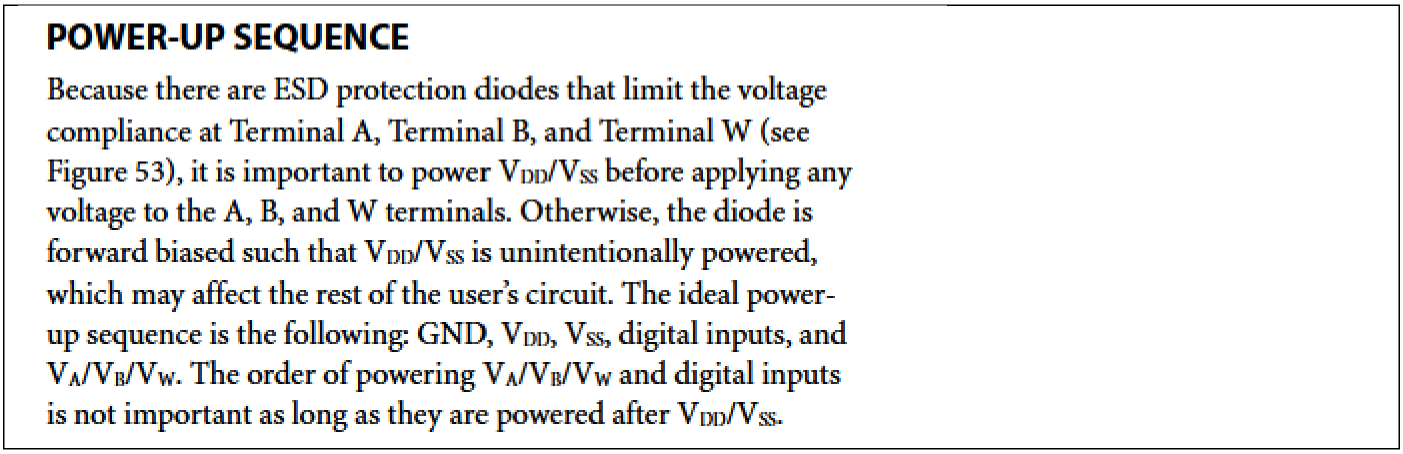
Notes:

Tried software reset → see code. This isn't working. Chip seems to not respond to shutdown.

After powering on potentiometer resets to closed state which result in inrush current. This "short" causes a drop in the 5V voltage regulator.

This drop in the 5V regulator may result in an unstable I2C bus.

Switch on 12V line with a delay seems to solve the problem, which is weird since i would have expected that the 5V startup could lead for instabilities. See datasheet AD5282



POWER-UP SEQUENCE

Because there are ESD protection diodes that limit the voltage compliance at Terminal A, Terminal B, and Terminal W (see Figure 53), it is important to power V_{DD}/V_{SS} before applying any voltage to the A, B, and W terminals. Otherwise, the diode is forward biased such that V_{DD}/V_{SS} is unintentionally powered, which may affect the rest of the user's circuit. The ideal power-up sequence is the following: GND, V_{DD}, V_{SS}, digital inputs, and $V_A/V_B/V_W$. The order of powering $V_A/V_B/V_W$ and digital inputs is not important as long as they are powered after V_{DD}/V_{SS}.

Explain more about the three frame "protocol"

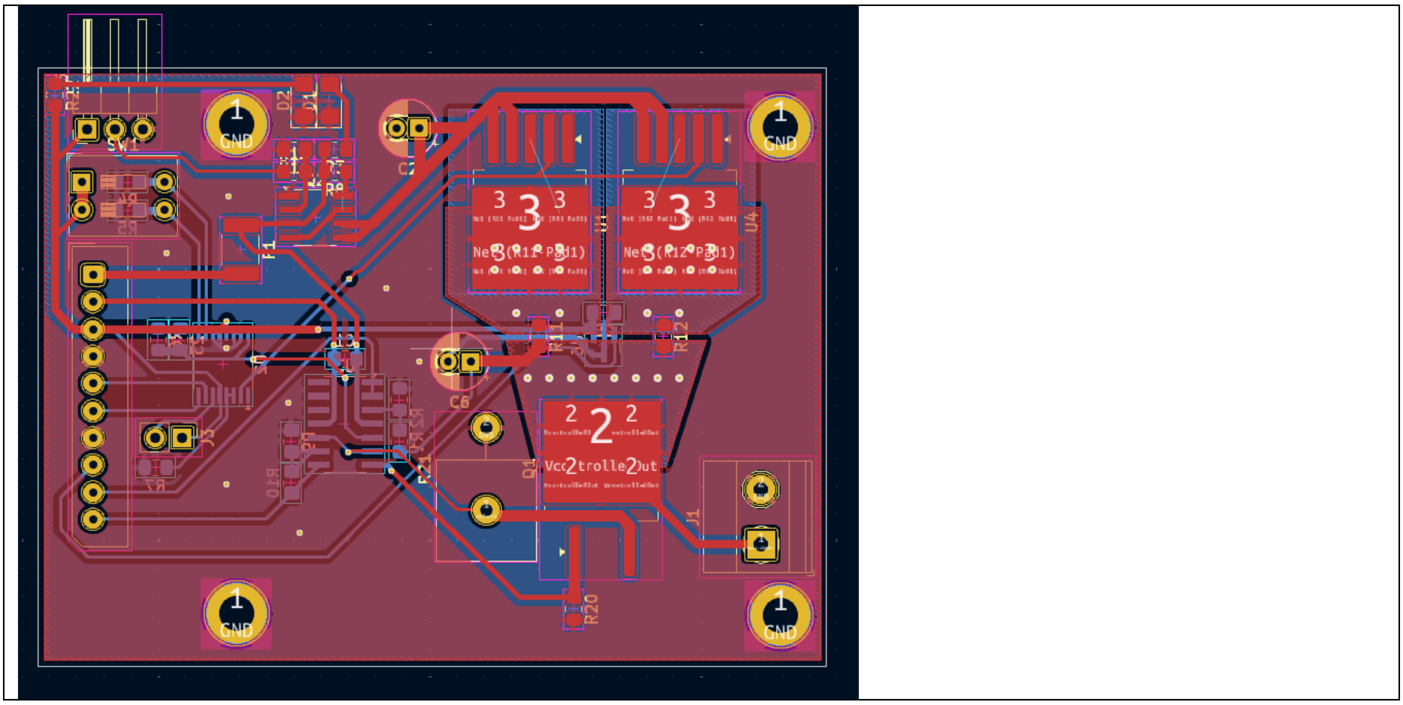
09-05-2025

To does:

Notes:

Started with PCB. Keep extra opamp as “spare” part. Power and ground on top, rest on bottom.

Routing is very difficult since large amount of current and different traces.



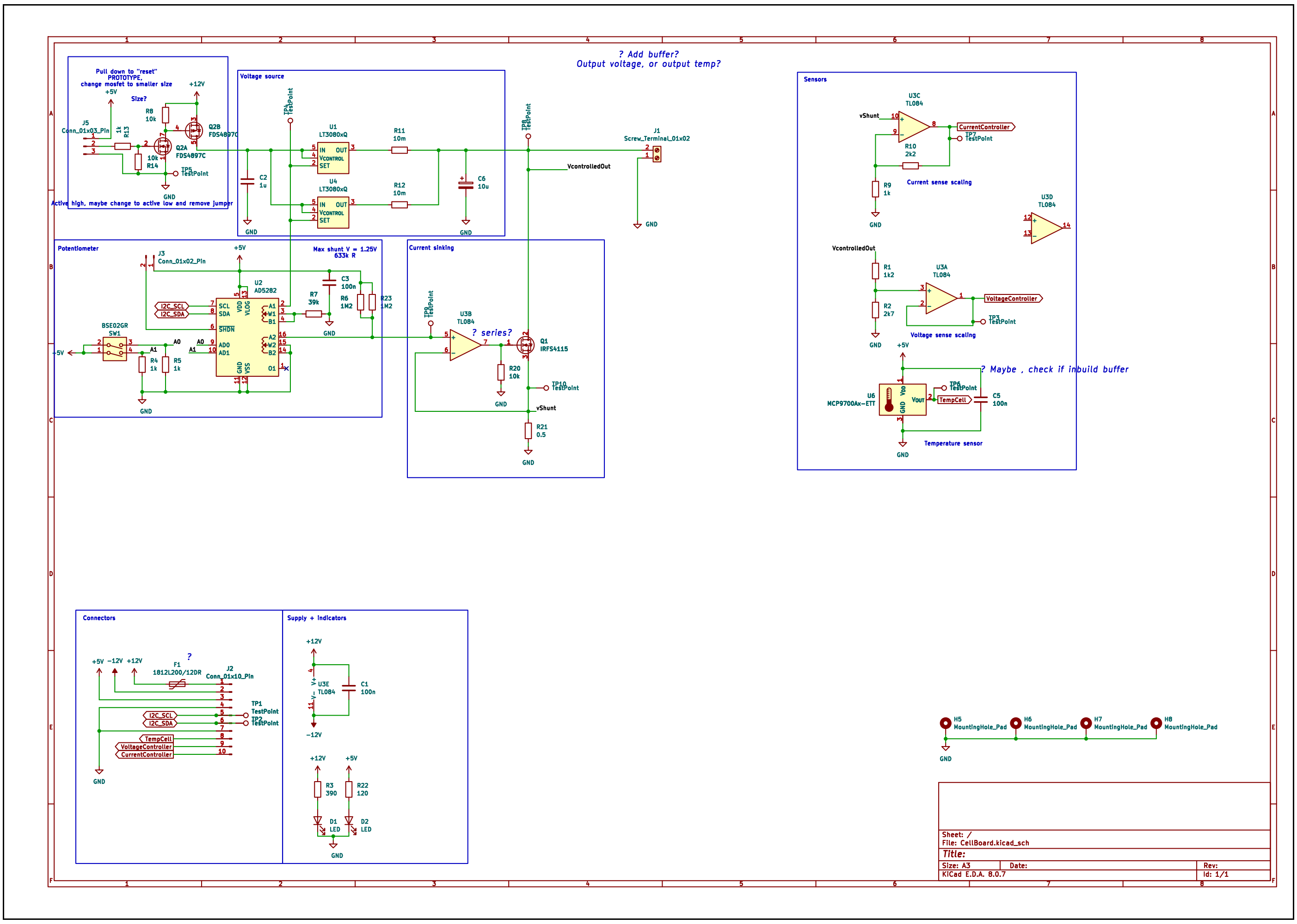
13-04-2025

To does: **Priority**

- Start GUI
- Refine PCB

Notes:

Finished routing PCB, need for improvement. Start designing GUI.



16-05-2025

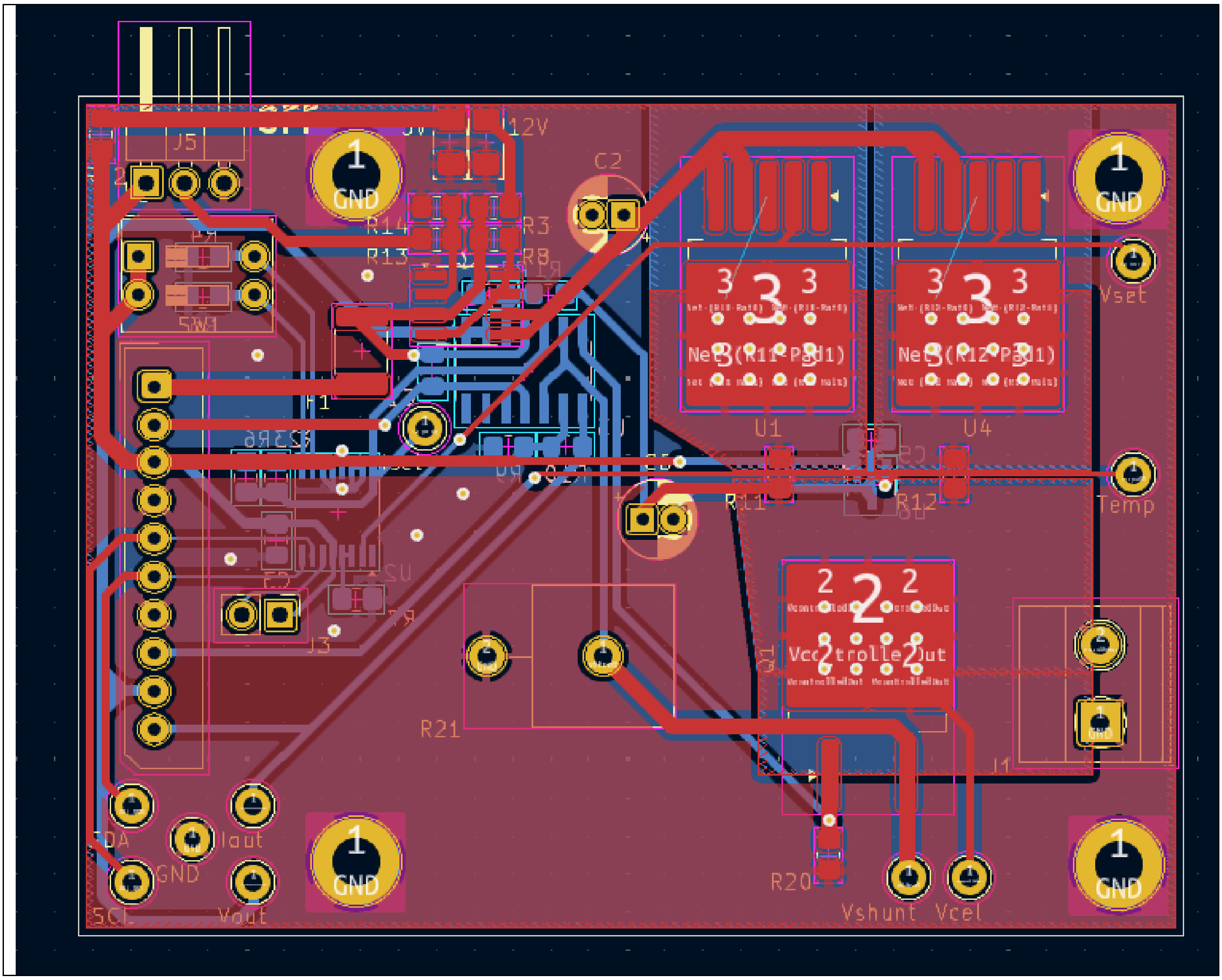
To does: **Priority**

- Start GUI
- Refine PCB

Notes:

Finished routing PCB, need for improvement. Start designing GUI.

Increase trace thickness and verify design with piet. Move load resistor for space heatsink.



21-05-2025

To does:	**Priority**
Start GUI	
Refine PCB	

Notes:

Checked with piet, he was happy. No big remarks regarding PCB.

Ordered PCB

26-05-2025

To does:	**Priority**
Start GUI	1
Improve ESP32 Code	2
Update documents	1
After success first Aisler order, Ask to order new Main board	2
New test plan	1

Notes:

Continued with documenting the work of previous weeks(Because of extra workload).

Plan coming weeks regarding GUI and new hardware testing.

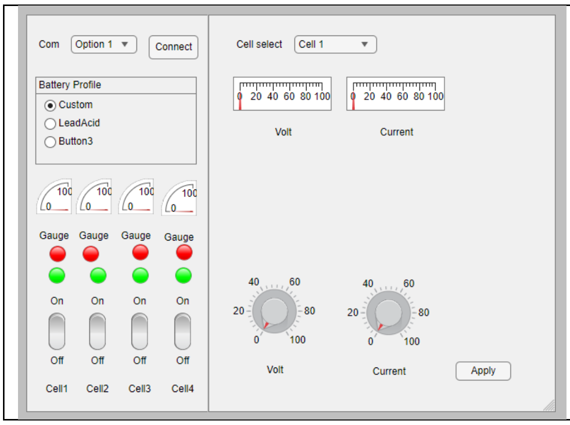
New testplan?

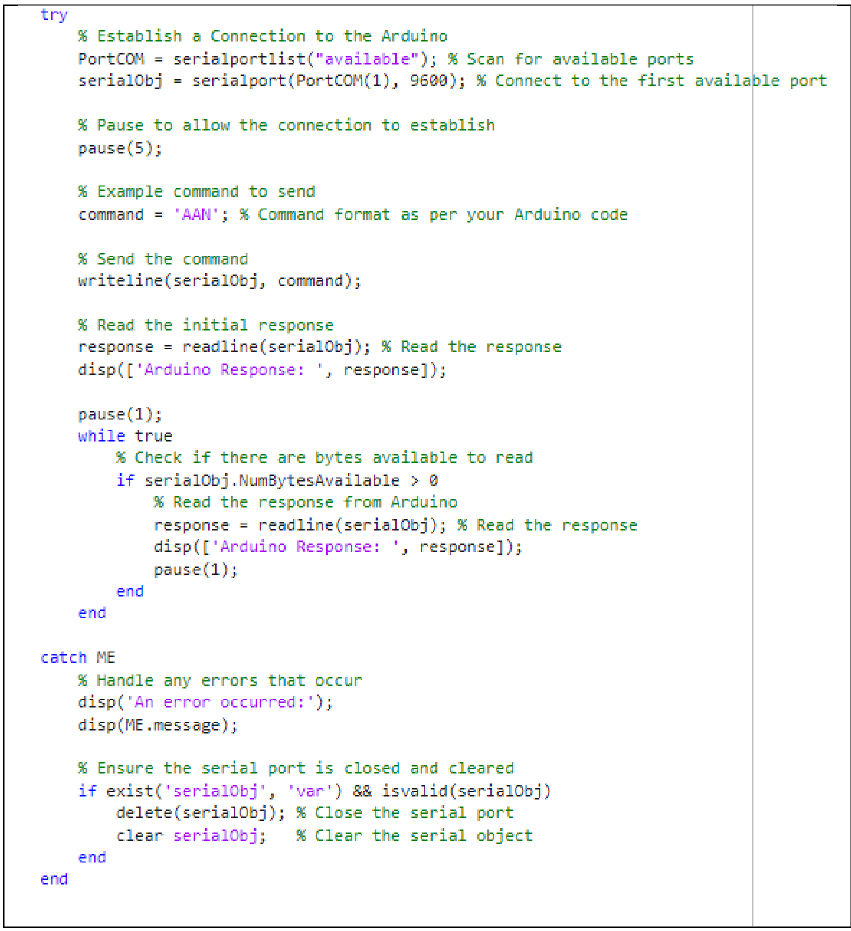
01-06-2025

To does:	Priority
Start GUI	1
Improve ESP32 Code	2
Update documents	1
After success first Aisler order, Ask to order new Main board	2
New test plan	1

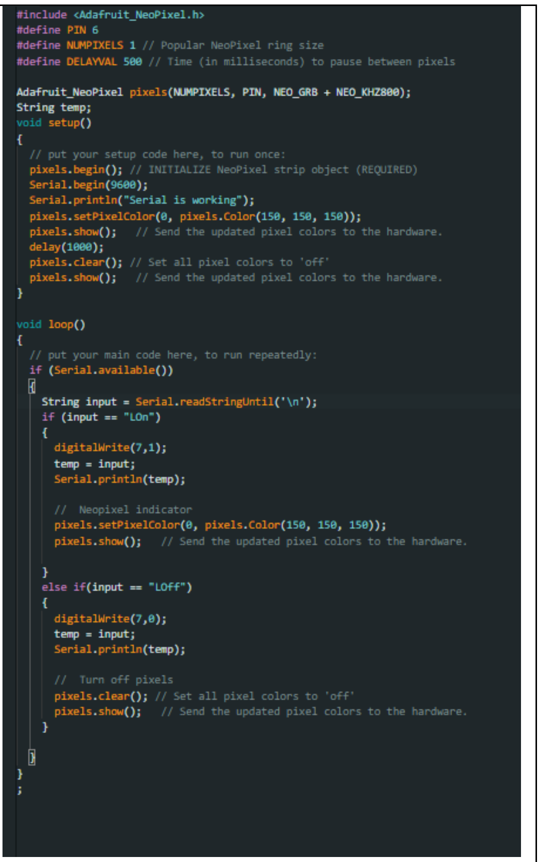
Notes:

Successfully established connection over UART serial in matlab environment





```
try
    % Establish a Connection to the Arduino
    PortCOM = serialportlist("available"); % Scan for available ports
    serialObj = serialport(PortCOM(1), 9600); % Connect to the first available port

    % Pause to allow the connection to establish
    pause(5);

    % Example command to send
    command = 'AAN'; % Command format as per your Arduino code

    % Send the command
    writeline(serialObj, command);

    % Read the initial response
    response = readline(serialObj); % Read the response
    disp(['Arduino Response: ', response]);

    pause(1);
    while true
        % Check if there are bytes available to read
        if serialObj.NumBytesAvailable > 0
            % Read the response from Arduino
            response = readline(serialObj); % Read the response
            disp(['Arduino Response: ', response]);
            pause(1);
        end
    end

catch ME
    % Handle any errors that occur
    disp('An error occurred:');
    disp(ME.message);

    % Ensure the serial port is closed and cleared
    if exist('serialObj', 'var') && isvalid(serialObj)
        delete(serialObj); % Close the serial port
        clear serialObj;   % Clear the serial object
    end
end
```



```
#include <Adafruit_NeoPixel.h>
#define PIN 6
#define NUMPIXELS 1 // Popular NeoPixel ring size
#define DELAYVAL 500 // Time (in milliseconds) to pause between pixels

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
String temp;
void setup()
{
  // put your setup code here, to run once:
  pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
  Serial.begin(9600);
  Serial.println("Serial is working");
  pixels.setPixelColor(0, pixels.Color(150, 150, 150));
  pixels.show();   // Send the updated pixel colors to the hardware.
  delay(1000);
  pixels.clear(); // Set all pixel colors to 'off'
  pixels.show();   // Send the updated pixel colors to the hardware.
}

void loop()
{
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    String input = Serial.readStringUntil('\n');
    if (input == "LOn")
    {
      digitalWrite(7,1);
      temp = input;
      Serial.println(temp);

      //  Neopixel indicator
      pixels.setPixelColor(0, pixels.Color(150, 150, 150));
      pixels.show();   // Send the updated pixel colors to the hardware.

    }
    else if(input == "LOff")
    {
      digitalWrite(7,0);
      temp = input;
      Serial.println(temp);

      //  Turn off pixels
      pixels.clear(); // Set all pixel colors to 'off'
      pixels.show();   // Send the updated pixel colors to the hardware.
    }

  }
}
;
```

Simple code to verify incoming data

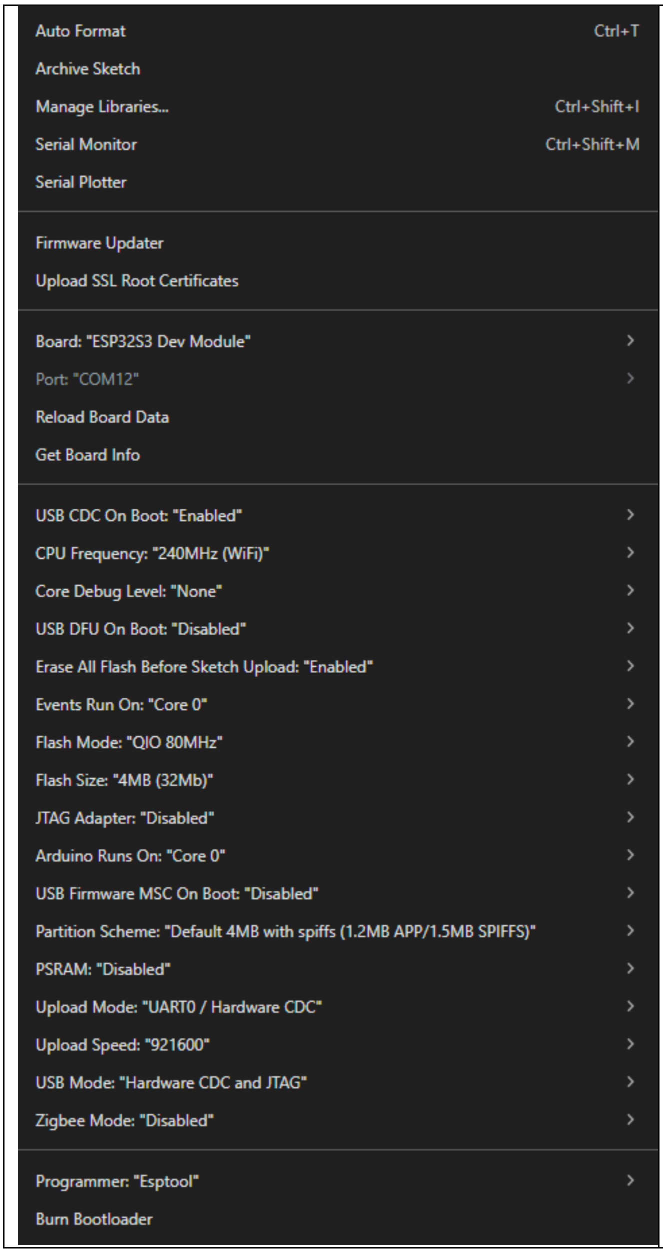
06-06-2025

To does:	**Priority**
- Start GUI	1
- Improve ESP32 Code	2
- Update documents	1
- After success first Aisler order, Ask to order new Main board	2
- New test plan	1

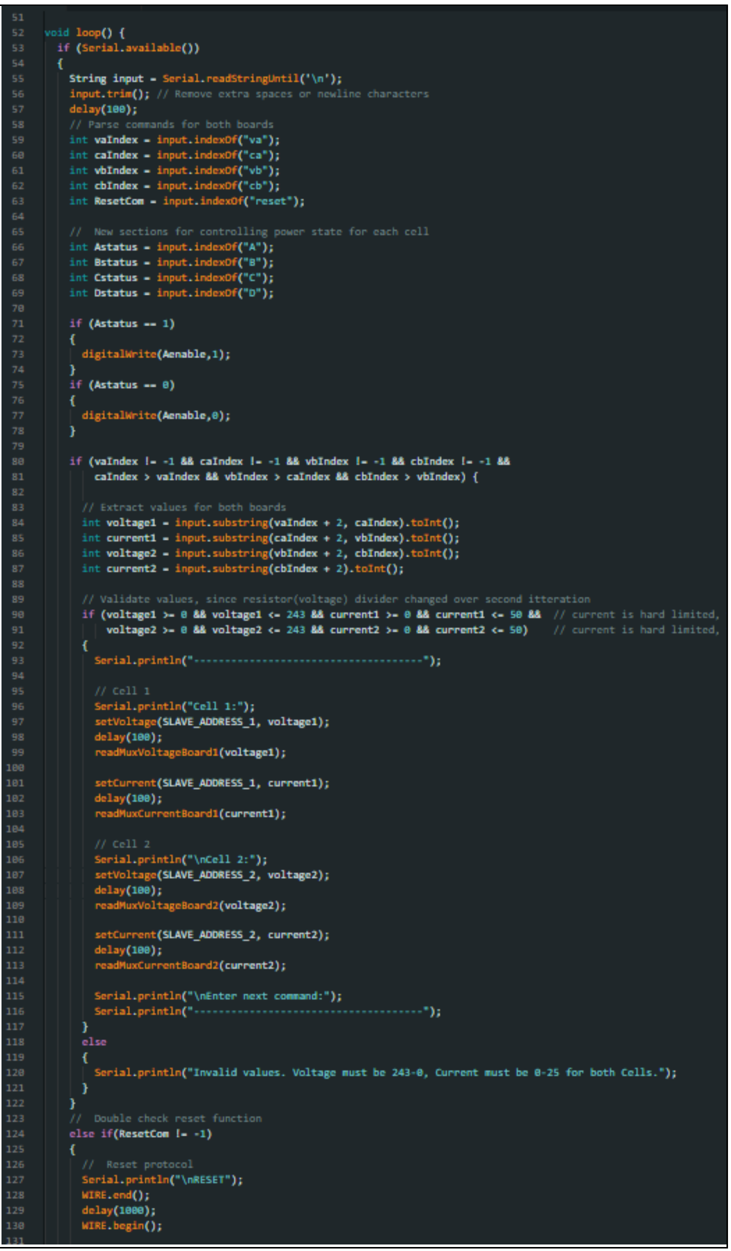
Notes:

Integrated simple program to verify working of GUI and to create a better understanding regarding the app designer. ORDER components for BMS

Improved main code(Loop) and ESP 32FLASH:



```
Auto Format                                                  Ctrl+T
Archive Sketch
Manage Libraries...                                    Ctrl+Shift+I
Serial Monitor                                         Ctrl+Shift+M
Serial Plotter

Firmware Updater
Upload SSL Root Certificates

Board: "ESP32S3 Dev Module"                                       >
Port: "COM12"                                                     >
Reload Board Data
Get Board Info

USB CDC On Boot: "Enabled"                                        >
CPU Frequency: "240MHz (WiFi)"                                    >
Core Debug Level: "None"                                          >
USB DFU On Boot: "Disabled"                                       >
Erase All Flash Before Sketch Upload: "Enabled"                   >
Events Run On: "Core 0"                                           >
Flash Mode: "QIO 80MHz"                                           >
Flash Size: "4MB (32Mb)"                                          >
JTAG Adapter: "Disabled"                                          >
Arduino Runs On: "Core 0"                                         >
USB Firmware MSC On Boot: "Disabled"                              >
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" >
PSRAM: "Disabled"                                                 >
Upload Mode: "UART0 / Hardware CDC"                               >
Upload Speed: "921600"                                            >
USB Mode: "Hardware CDC and JTAG"                                 >
Zigbee Mode: "Disabled"                                           >

Programmer: "Esptool"                                             >
Burn Bootloader
```



```

void loop() {
  if (Serial.available())
  {
    String input = Serial.readStringUntil('\n');
    input.trim(); // Remove extra spaces or newline characters
    delay(100);
    // Parse commands for both boards
    int vaIndex = input.indexOf("va");
    int caIndex = input.indexOf("ca");
    int vbIndex = input.indexOf("vb");
    int cbIndex = input.indexOf("cb");
    int ResetCom = input.indexOf("reset");

    //  New sections for controlling power state for each cell
    int Astatus = input.indexOf("A");
    int Bstatus = input.indexOf("B");
    int Cstatus = input.indexOf("C");
    int Dstatus = input.indexOf("D");

    if (Astatus == 1)
    {
      digitalWrite(Aenable,1);
    }
    if (Astatus == 0)
    {
      digitalWrite(Aenable,0);
    }

    if (vaIndex != -1 && caIndex != -1 && vbIndex != -1 && cbIndex != -1 &&
        caIndex > vaIndex && vbIndex > caIndex && cbIndex > vbIndex) {

      // Extract values for both boards
      int voltage1 = input.substring(vaIndex + 2, caIndex).toInt();
      int current1 = input.substring(caIndex + 2, vbIndex).toInt();
      int voltage2 = input.substring(vbIndex + 2, cbIndex).toInt();
      int current2 = input.substring(cbIndex + 2).toInt();

      // Validate values, since resistor(voltage) divider changed over second itteration
      if (voltage1 >= 0 && voltage1 <= 243 && current1 >= 0 && current1 <= 50 &&  // current is hard limited,
          voltage2 >= 0 && voltage2 <= 243 && current2 >= 0 && current2 <= 50)    // current is hard limited,
      {
        Serial.println("------------------------------------");

        // Cell 1
        Serial.println("Cell 1:");
        setVoltage(SLAVE_ADDRESS_1, voltage1);
        delay(100);
        readMuxVoltageBoard1(voltage1);

        setCurrent(SLAVE_ADDRESS_1, current1);
        delay(100);
        readMuxCurrentBoard1(current1);

        // Cell 2
        Serial.println("\nCell 2:");
        setVoltage(SLAVE_ADDRESS_2, voltage2);
        delay(100);
        readMuxVoltageBoard2(voltage2);

        setCurrent(SLAVE_ADDRESS_2, current2);
        delay(100);
        readMuxCurrentBoard2(current2);

        Serial.println("\nEnter next command:");
        Serial.println("------------------------------------");
      }
      else
      {
        Serial.println("Invalid values. Voltage must be 243-0, Current must be 0-25 for both Cells.");
      }
    }
    //  Double check reset function
    else if(ResetCom != -1)
    {
      //  Reset protocol
      Serial.println("\nRESET");
      WIRE.end();
      delay(1000);
      WIRE.begin();

```

Disassemble good parts(Connectors, etc) from broken boards.

13-06-2025

To does:	**Priority**
- Start GUI	1
- Improve ESP32 Code	2
- Update documents	1
- After success first Aisler order, Ask to order new Main board	2
- New test plan	1

Notes:

Continued functional GUI.

22-06-2025

To does:	**Priority**
- Start GUI	1
- Improve ESP32 Code	2
- Update documents	1
- After success first Aisler order, Ask to order new Main board	2
- New test plan	1

Notes:

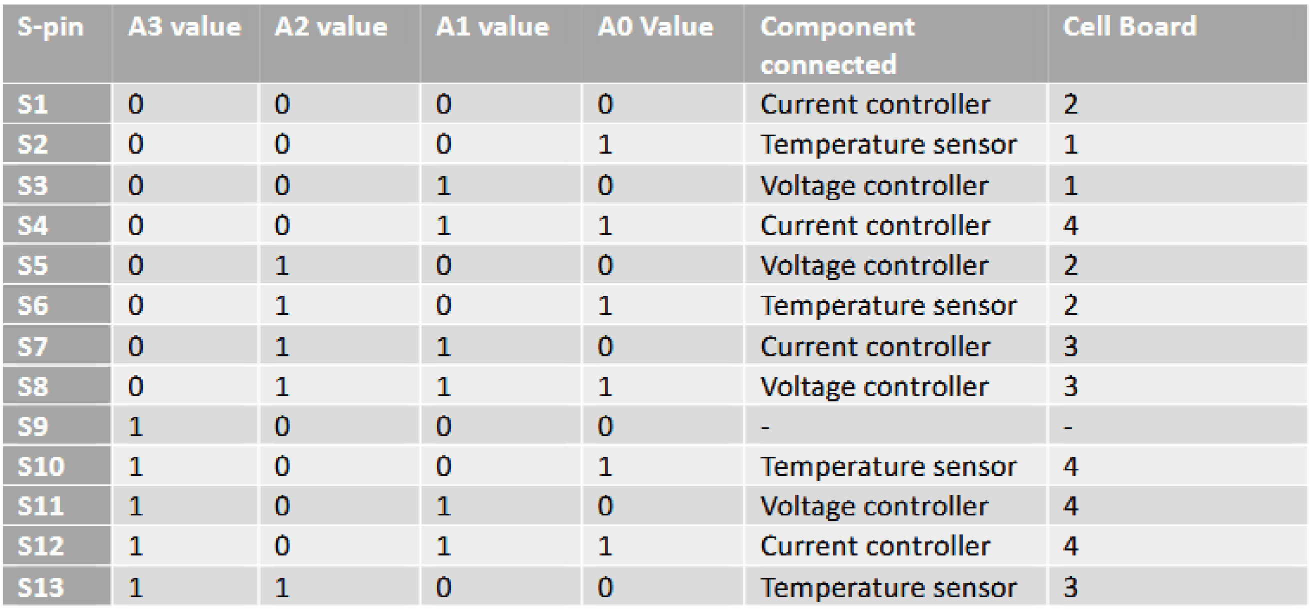
Continued functional GUI. Started extending base code for 4 cells

23-06-2025

To does:	**Priority**
- Improve GUI	2
- Improve ESP32 Code	2
- Update documents	2
- Soldering	1
- New test plan	2

Notes:

Be aware of chaos and wrong use of code in main code: spagethi code



S-pin	A3 value	A2 value	A1 value	A0 Value	Component connected	Cell Board
S1	0	0	0	0	Current controller	2
S2	0	0	0	1	Temperature sensor	1
S3	0	0	1	0	Voltage controller	1
S4	0	0	1	1	Current controller	4
S5	0	1	0	0	Voltage controller	2
S6	0	1	0	1	Temperature sensor	2
S7	0	1	1	0	Current controller	3
S8	0	1	1	1	Voltage controller	3
S9	1	0	0	0	-	-
S10	1	0	0	1	Temperature sensor	4
S11	1	0	1	0	Voltage controller	4
S12	1	0	1	1	Current controller	4
S13	1	1	0	0	Temperature sensor	3

Changed values for cellboard 3 & 4. Solve chaos by introducing header file.

Verify working code and than intruduce generic functions:

```
// Function to read temperature from a specified board
void readTemperature(int board) {
  // Set the address pins based on the board number
  digitalWrite(A3_PIN, (board & 0b1000) ? HIGH : LOW);
  digitalWrite(A2_PIN, (board & 0b0100) ? HIGH : LOW);
  digitalWrite(A1_PIN, (board & 0b0010) ? HIGH : LOW);
  digitalWrite(A0_PIN, (board & 0b0001) ? HIGH : LOW);
  delay(100);

  int adcValue = analogRead(DOUT_PIN);
  float voltage = adcValue * (3.3 / 4095.0);
  float temp = 1.25 * ((voltage - V0C) / TC);

  Serial.print("Board ");
  Serial.print(board);
  Serial.print(" Temperature: ");
  Serial.print(temp, 2);
  Serial.println(" °C");
}

// Function to read voltage from a specified board
void readMuxVoltage(int board, int dPotStep) {
  // Set the address pins based on the board number
  digitalWrite(A3_PIN, (board & 0b1000) ? HIGH : LOW);
  digitalWrite(A2_PIN, (board & 0b0100) ? HIGH : LOW);
  digitalWrite(A1_PIN, (board & 0b0010) ? HIGH : LOW);
  digitalWrite(A0_PIN, (board & 0b0001) ? HIGH : LOW);
  delay(100);

  long totalAdcValue = 0;
  for (int i = 0; i < SAMPLE_COUNT; i++) {
    totalAdcValue += analogRead(DOUT_PIN);
    delay(1);
  }
  int averagedAdcValue = totalAdcValue / SAMPLE_COUNT;
  float voltage = ((((averagedAdcValue * (3.3 / 4095)) * 2) / 0.65) - 0.05);

  Serial.print("Board ");
  Serial.print(board);
  Serial.print(" dPot. step: ");
  Serial.print(dPotStep);
  Serial.print("   |  ");
  Serial.print(voltage, 2);
  Serial.println(" V");
}

// Function to read current from a specified board
void readMuxCurrent(int board, int dPotStep) {
  // Set the address pins based on the board number
  digitalWrite(A3_PIN, (board & 0b1000) ? HIGH : LOW);
  digitalWrite(A2_PIN, (board & 0b0100) ? HIGH : LOW);
  digitalWrite(A1_PIN, (board & 0b0010) ? HIGH : LOW);
  digitalWrite(A0_PIN, (board & 0b0001) ? HIGH : LOW);
  delay(100);

  float currentSum = 0;
  for (int i = 0; i < SAMPLE_COUNT; i++) {
    int adcValue = analogRead(DOUT_PIN);
    float current = ((((adcValue * (3.3 / 4095)) * 2) / 3.3) / 0.5);
    currentSum += current;
  }
  float currentAverage = currentSum / SAMPLE_COUNT;

  Serial.print("Board ");
  Serial.print(board);
  Serial.print(" dPot. step: ");
  Serial.print(dPotStep);
  Serial.print("    |  ");
  Serial.print(currentAverage, 2);
  Serial.print(" A  |   ");
  readTemperature(board);
}
```

```
// Inside the loop function, after validating the input
if (voltage1 >= 0 && voltage1 <= 243 && current1 >= 0 && current1 <= 50 &&
   voltage2 >= 0 && voltage2 <= 243 && current2 >= 0 && current2 <= 50 &&
   voltage3 >= 0 && voltage3 <= 243 && current3 >= 0 && current3 <= 50 &&
   voltage4 >= 0 && voltage4 <= 243 && current4 >= 0 && current4 <= 50) {

  Serial.println("------------------------------------");

  // Cell 1
  Serial.println("Cell 1:");
  setVoltage(SLAVE_ADDRESS_1, voltage1);
  delay(100);
  readMuxVoltage(1, voltage1);  // Call the generic function for voltage
  setCurrent(SLAVE_ADDRESS_1, current1);
  delay(100);
  readMuxCurrent(1, current1);   // Call the generic function for current

  // Cell 2
  Serial.println("\nCell 2:");
  setVoltage(SLAVE_ADDRESS_2, voltage2);
  delay(100);
  readMuxVoltage(2, voltage2);  // Call the generic function for voltage
  setCurrent(SLAVE_ADDRESS_2, current2);
  delay(100);
  readMuxCurrent(2, current2);   // Call the generic function for current

  // Cell 3
  Serial.println("\nCell 3:");
  setVoltage(SLAVE_ADDRESS_3, voltage3);
  delay(100);
  readMuxVoltage(3, voltage3);  // Call the generic function for voltage
  setCurrent(SLAVE_ADDRESS_3, current3);
  delay(100);
  readMuxCurrent(3, current3);   // Call the generic function for current

  // Cell 4
  Serial.println("\nCell 4:");
  setVoltage(SLAVE_ADDRESS_4, voltage4);
  delay(100);
  readMuxVoltage(4, voltage4);  // Call the generic function for voltage
  setCurrent(SLAVE_ADDRESS_4, current4);
  delay(100);
  readMuxCurrent(4, current4);   // Call the generic function for current

  Serial.println("\nEnter next command:");
  Serial.println("------------------------------------");
}
else
{
  Serial.println("Invalid values. Voltage must be 243-0, Current must be 0-50 for both Cells.");
}
```

Even another approach with libraries

CellBoardControl.h

```
#ifndef CellBoardControl_h
#define CellBoardControl_h

#include <Arduino.h>

class CellBoardControl {
public:
   CellBoardControl(int sdaPin, int sclPin);
   void readTemperature(int board);
   void readMuxVoltage(int board, int dPotStep);
   void readMuxCurrent(int board, int dPotStep);
   void setVoltage(uint8_t slaveAddress, uint8_t value);
   void setCurrent(uint8_t slaveAddress, uint8_t value);

private:
   int _sdaPin;
   int _sclPin;
};

#endif
```

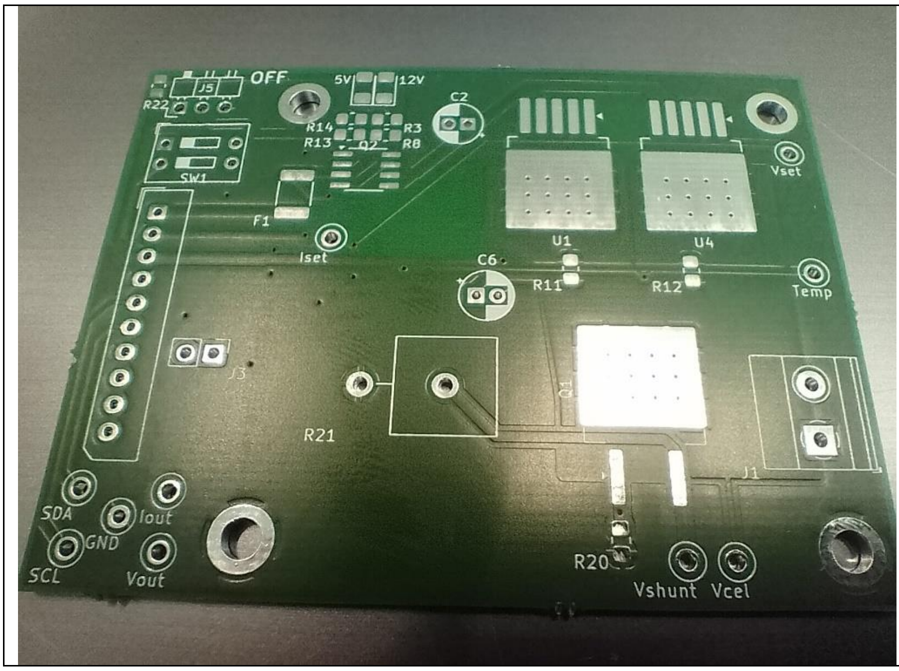
CellBoardControl.cpp

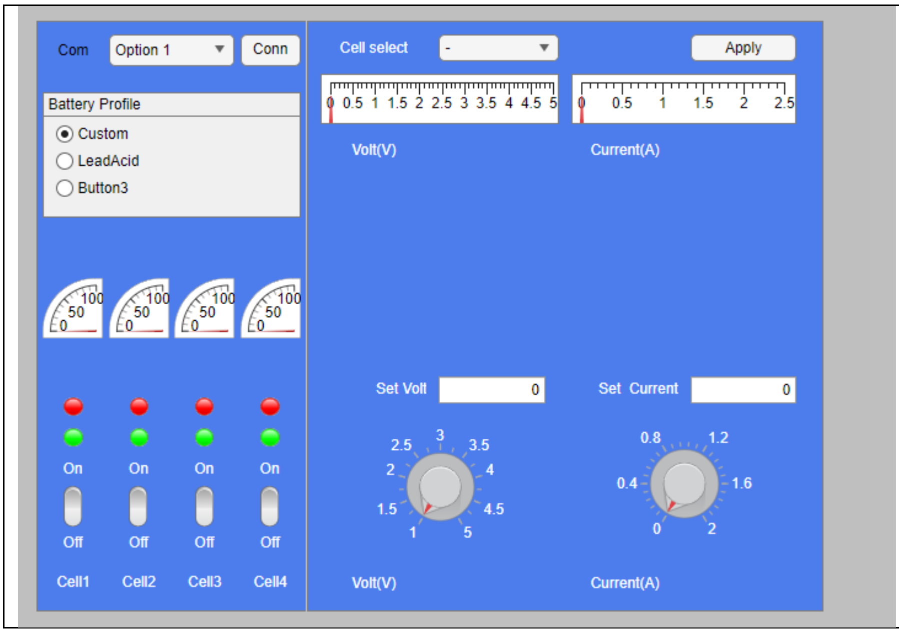
```
#include "CellBoardControl.h"
#include <Wire.h>

CellBoardControl::CellBoardControl(int sdaPin, int sclPin) {
  _sdaPin = sdaPin;
  _sclPin = sclPin;
  Wire.setPins(_sdaPin, _sclPin);
  Wire.begin();
}

void CellBoardControl::readTemperature(int board) {
  // Set the address pins based on the board number
  digitalWrite(A3_PIN, (board & 0b1000) ? HIGH : LOW);
  digitalWrite(A2_PIN, (board & 0b0100) ? HIGH : LOW);
  digitalWrite(A1_PIN, (board & 0b0010) ? HIGH : LOW);
  digitalWrite(A0_PIN, (board & 0b0001) ? HIGH : LOW);
  delay(100);

  int adcValue = analogRead(DOUT_PIN);
  float voltage = adcValue * (3.3 / 4095.0);
  float temp = 1.25 * ((voltage - V0C) / TC);

  Serial.print("Board ");
  Serial.print(board);
  Serial.print(" Temperature: ");
  Serial.print(temp, 2);
  Serial.println(" °C");
}

void CellBoardControl::readMuxVoltage(int board, int dPotStep) {
  // Set the address pins based on the board number
  digitalWrite(A3_PIN, (board & 0b1000) ? HIGH : LOW);
  digitalWrite(A2_PIN, (board & 0b0100) ? HIGH : LOW);
  digitalWrite(A1_PIN, (board & 0b0010) ? HIGH : LOW);
  digitalWrite(A0_PIN, (board & 0b0001) ? HIGH : LOW);
  delay(100);

  long totalAdcValue = 0;
  for (int i = 0; i < SAMPLE_COUNT; i++) {
    totalAdcValue += analogRead(DOUT_PIN);
    delay(1);
  }
  int averagedAdcValue = totalAdcValue / SAMPLE_COUNT;
  float voltage = ((((averagedAdcValue * (3.3 / 4095)) * 2) / 0.65) - 0.05);

  Serial.print("Board ");
  Serial.print(board);
  Serial.print(" dPot. step: ");
  Serial.print(dPotStep);
  Serial.print("   |  ");
  Serial.print(voltage, 2);
  Serial.println(" V");
}

void CellBoardControl::readMuxCurrent(int board, int dPotStep) {
  // Set the address pins based on the board number
  digitalWrite(A3_PIN, (board & 0b1000) ? HIGH : LOW);
  digitalWrite(A2_PIN, (board & 0b0100) ? HIGH : LOW);
  digitalWrite(A1_PIN, (board & 0b0010) ? HIGH : LOW);
  digitalWrite(A0_PIN, (board & 0b0001) ? HIGH : LOW);
  delay(100);

  float currentSum = 0;
  for (int i = 0; i < SAMPLE_COUNT; i++) {
    int adcValue = analogRead(DOUT_PIN);
    float current = ((((adcValue * (3.3 / 4095)) * 2) / 3.3) / 0.5);
    currentSum += current;
  }
  float currentAverage = currentSum / SAMPLE_COUNT;

  Serial.print("Board ");
  Serial.print(board);
  Serial.print(" dPot. step: ");
  Serial.print(dPotStep);
  Serial.print("    |  ");
  Serial.print(currentAverage, 2);
  Serial.print(" A |  ");
  readTemperature(board); // Call the temperature reading function
}

void CellBoardControl::setVoltage(uint8_t slaveAddress, uint8_t value) {
  Wire.beginTransmission(slaveAddress);
  Wire.write(0); // Choose Wiper 1 (Voltage control)
  Wire.write(value);
  Wire.endTransmission();
}

void CellBoardControl::setCurrent(uint8_t slaveAddress, uint8_t value) {
  Wire.beginTransmission(slaveAddress);
  Wire.write(128); // Choose Wiper 2 (Current control)
  Wire.write(value);
  Wire.endTransmission();
}
```

Main.ino EXAMPLE!

```
#include <CellBoardControl.h>

// Define your SDA and SCL pins
#define PIN_SDA 48
#define PIN_SCL 47

// Create an instance of the CellBoardControl class
CellBoardControl cellBoardControl(PIN_SDA, PIN_SCL);

void setup() {
  Serial.begin(9600);
  // Other setup code...
}

void loop() {
  // Example usage of the library functions
  int voltage1 = 128; // Example voltage value
  int current1 = 10;  // Example current value

  // Set voltage and current for Cell 1
  cellBoardControl.setVoltage(SLAVE_ADDRESS_1, voltage1);
  cellBoardControl.setCurrent(SLAVE_ADDRESS_1, current1);

  // Read voltage and current for Cell 1
  cellBoardControl.readMuxVoltage(1, voltage1);
  cellBoardControl.readMuxCurrent(1, current1);

  // Add similar calls for other cells as needed...

  delay(1000); // Delay for demonstration purposes
}
```

Started solderen PCB. Started with one and verify working. After that order more components to solder a second and maybe even 3 & 4. But it is important to make clear it takes long to solder 1.




Com: Option 1

Battery Profile
● Custom
○ LeadAcid
○ Button3

Cell select: -

Set Volt: 0

Set Current: 0

Red light goes on when overcurrent or high temperature

24-06-2025

Wiper deviate because wrong r23 r6s

Another deviation is no R11 R12

25-06-2025

To does:	**Priority**
Improve GUI	2
Improve ESP32 Code	1
Update documents	1
Maybe order new components for second PCB	2
Make 3d Printed fan casing	2

Notes:

I think previous digital pot was defective because new works out of the box.

Switching system works

Voltage works

Max current goes to 1.74A, Iset stuck at 0.87 V

Current → 0 when wiper 255

Voltage → 0.8 when wiper 255

So first fix scaling in Arduino. Than fix scaling in matlab gui by inverting the code now.

DECLARIABLE DEVIATION SINCE PULLUP 2X 1M2 WERE NOT AVAILBLE

Conclusion:

Shortcomings:

- Missing 2x 10mOhms resistors for balance LDO
- Wrong R23 & R6 1Mohms instead of 1m2 → Maximum current a little off

A lot of heat generation on bottom and top which was expected, heat is good divided between bottom and top. Maybe fan structure would solve heat.

AFTER VERIFYING OPAMP DOESNT SEEM TO GET HIGHER, NOT VERY DECLARIABLE..........

26-06-2025

Make test plan and test voltage drop over current divider system

01-07-2025

Notes

Device runs noticeable cooler with 5015 fan running at low speed.

Device functions OK with 9V supply. (REMEMBER mainboard still designed to handle 12V)

After checking set resistors, fixed current problem

Conclusion:

Because of incorrect resistors hardware limit for current is to HIGH.

System works!

Hard limit 60-70

Problems tackled:

- I2C
- Heat+-
- Wipers

To improve

Hardware:

- Better heatsink on tops
- Better case with better flow
- More text on PCB, Like indications.
- Even bigger heatpads for heat dissipation
- Standard PCB sizes that actually matter(Logical steps, etcs)

02-07-2025

Finish report for 18-07-2025 (MDD) & not plan of approach.

Think about continuing vacation???

Finish Arduino code and extensively testing

Finish kicad notes

Only small steps

Target 10 July, Send to Ralph, Harold

10-07-2025

To does,

- Add notes to software & kicad
- Documentation

Notes:

Worked on docu0

Componentenlijst

Why Arduino ide over Vscode

Future work:

- Sofware
  - Matlab
  - Arduino
  - Optimize
- PCBs
- Testing with hardware bms (preferred NXP)

Steps	First connected everything Power on Power supply totalsystem Connect USB Send Command to zero all Power on cells Continue regular use

Voltage read isn't correct. Previous group used magic numbers

INSERT PICTURE

STUPID ME CONNECT NOTHING TO THE POWER LINE FOR THE CELLBOARD BECAUSE THIS CAN INTERFERE

Something about using 12V instead of 9, because mainboard isn't made for less

Add pinheaders for fans.

Make system to function with lower supply voltage because better efficiency. This includes adjusting the main board, and all the scalers to function properly with a lower voltage

15-07-2025

Dont forget to make BOM for PCB

Add comments kicad

Wire impedance, shorter wires

Bom in appendices