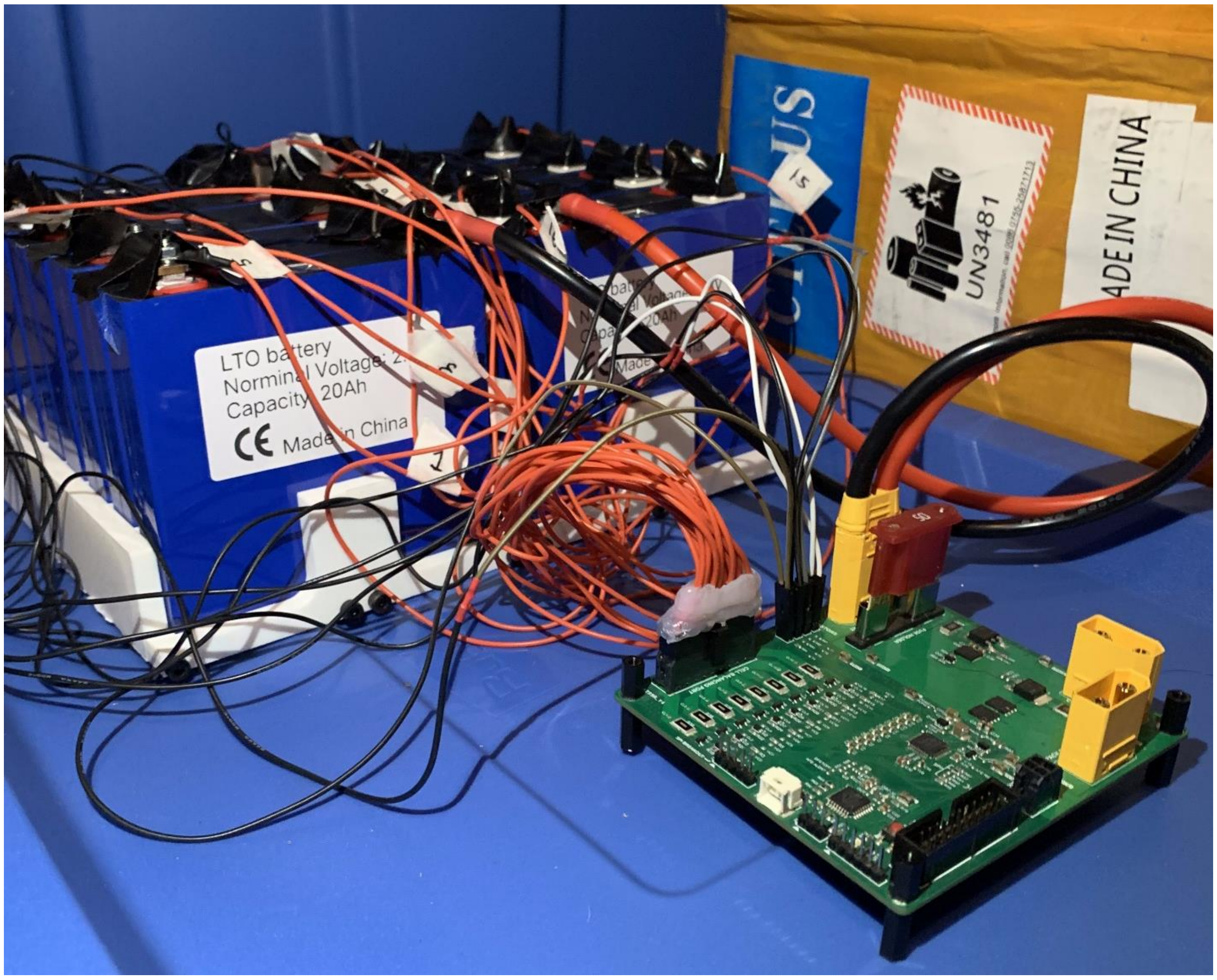


Custom Lithium-Titanate (LTO) Battery Pack and Battery Management System (BMS)

Design Whitepaper

Prepared by: Farris Matar

December 17, 2024

Table of Contents

1 Background

This project is part of a larger overall objective to assemble an e-scooter with all electronics built from scratch. The primary motivation of this endeavor is to learn about all the components that go into designing a small-scale electric vehicle and attempt to turn this knowledge into practical experience by building the components in order to create an e-scooter that is comparable to most consumer-range e-scooters. In order to verify the final build's performance, it should be able to achieve the following specifications:

- Top speed of 30km/h
- Minimum range of 30km (approximately 1 hour of continuous use)
- Adjustable speed via a user-operated throttle
- Display to show power consumption, battery's remaining capacity, and current speed

The battery and battery management system (BMS) were the first components to be designed for the e-scooter. The specifications above were used to guide the defining requirements of these components.

At this stage, the only other component to be selected was the motor. The motor intended for use in the e-scooter is a 36V, 720W brushless DC motor. Since the motor will consume the most power by far compared to any other component in the e-scooter, it was important to have these parameters to further define this project's requirements.

2 Project Overview

The battery's role in the e-scooter is to store energy to be supplied for the motor and other electronic components when the e-scooter is being driven. The BMS's role is to act as a layer of protection for the battery, constantly monitoring the voltage, current, temperature, and other metrics for both telemetry purposes and to activate protections that will disable battery charging and/or discharging when appropriate.

2.1 Battery Requirements

A summary of the main requirements for the battery is provided in Table 2.1 below.

Table 2.1: Requirements for the e-scooter battery

Requirement	Reasoning
Battery capacity at least 720 Wh	720Wh is a sizeable amount that will keep a balance between total capacity and battery weight. 720Wh allows powering the 720W motor continuously at max. power for about an hour, ensuring the minimum range is met.
Battery nominal voltage is around 36V	This ensures the motor voltage rating is not exceeded. Although a DC-DC converter can be used to regulate the motor voltage, keeping the nominal battery voltage around the rated motor voltage can reduce the need for this, improving efficiency.
Battery discharging rate is at least 30A	30A of discharging allows outputting up to roughly 1080W of power with a nominal voltage around 36V, which is well beyond the expected power draw of the motor and provides a healthy margin for powering auxiliary electronics.
Battery charging rate is at least 10A	10A of charging current will allow the battery to be fully charged in just over 2 hours, based on the nominal voltage and capacity requirements, which is a reasonable amount of time to wait while not stressing on the capabilities of the battery.
Battery can operate in ambient temperatures between -15ºC and 30ºC.	Because the e-scooter is to be used outside, it should be capable of running in a wide range of temperatures without a significant change in performance.

2.2 Selection of Battery Cells

In addition to the specifications above, a battery chemistry that was relatively stable and safe to use was prioritized. After careful consideration, the battery chemistry selected was lithium titanate (LTO). Lithium titanate is known for being more safe and thermally stable compared to other lithium-ion battery chemistries. Additionally, it is capable of very high charging and discharging rates, which are appropriate given the high power consumption of the e-scooter and the benefit of having a shorter charge time. LTO batteries also have excellent low-temperature discharging capabilities, typically being able to retain over 80% capacity at -30ºC. The major downside compared to other chemistries is the energy density – for the same capacity, LTO batteries will typically be heavier and larger than their counterparts using other chemistries. Although this isn’t ideal for an e-scooter, it was accepted as a necessary tradeoff for its safety and high performance in the required specifications.

For the specific cells to be used to build the battery, the Toshiba SCiB 20Ah cells were chosen. These cells have a nominal voltage of 2.3V, meaning with a 16s1p battery pack the total capacity would be approximately 736 Wh. The energy density of the Toshiba SCiB cells were also higher than most commercially available LTO cells, reducing the weight disadvantage.

2.3 BMS Requirements

A summary of the main requirements for the BMS is provided in Table 2.2 below.

Table 2.2: Requirements for the BMS PCB

Requirement	Reasoning
BMS PCB and component ratings are sufficiently high to support the voltage, current, cell count, and temperature range the battery will operate with	This is self-explanatory – the BMS should not limit the battery from achieving its required specifications.
BMS is capable of measuring cell voltages, battery current, and battery temperature with less than 1% error.	1% error ensures sufficient accuracy for the BMS to report the battery's status and especially to accurately determine when to enable protections
BMS is capable of performing cell balancing, supporting at least 500mA of balancing current.	Given the fairly high capacity requirement, a decently high balancing current should be supported to ensure the final stage of charging does not take excessively long in the event of unbalanced cell voltages.
BMS will send telemetry about battery voltage, current, and temperature over UART	UART was selected for its simple connection and reliability over medium-length cables. Additionally, if signal integrity becomes an issue due to the cable length, it can be improved easily by converting it into a differential protocol such as RS422 through a pair of transceivers

2.4 BMS Component Selection

The two key components that would guide the BMS PCB design were the analog front-end (AFE) and the microcontroller unit (MCU).

2.4.1 AFE Selection: TI BQ76952

The AFE's purpose is to provide high-accuracy and precision measurements of the cell voltages, charging/discharging current, and battery temperature. An AFE can be given additional control in the BMS, such as controlling the charging and discharging MOSFETs and the cell balancing MOSFETs based on its measurements.

For this BMS, the TI BQ76952 AFE was selected. It uses 24-bit ADCs for measuring voltages, temperatures (via external thermistors), and current, reporting excellent accuracy under test conditions. It also offers a full suite of protections including over/undervoltage, over/undercurrent during both charge and discharge, and over/undertemperature, in addition to supporting autonomous cell-balancing. TI also provides extensive documentation in its technical reference manual on how to configure the AFE, providing an extensive variety of options based on the configuration of the battery and the level of control desired for the AFE in terms of operation and protection. All of these features provided good confidence in the BQ76952 allowing the BMS to achieve all the specified requirements.

2.4.2 MCU Selection: STM32L412K8T6

The MCU's primary role is to program and communicate with the BQ76952 over I2C or SPI, as well as summarize and report information on the battery status given by the BQ76952 on a UART channel. These requirements are fairly simple and plenty of microcontrollers would have been able to accomplish them, however the STM32L412K8T6 was chosen for 2 specific reasons. The first is that STM's microcontrollers are very commonly used both in industry and by hobbyists, providing plenty of reference material to assist with programming and debugging it for use on the BMS. The second is for its extremely low power consumption. The STM32L4 series of microcontrollers are capable of operating with very low operating current consumption (in the range of a few hundred microamps), as well as allowing several modes of operation for reducing the current draw of the MCU down to tens of nanoamps. This is perfect for minimizing the power consumed from the battery while the e-scooter is not in use.

3 BMS PCB Design

All design files were produced in Altium and can be found on GitHub here:
https://github.com/far-mat42/custom-escooter/tree/main/BMS%20PCB/LTO%2016S%20BMS

A high-level overview of the PCB design is depicted in Figure 3.1 below.

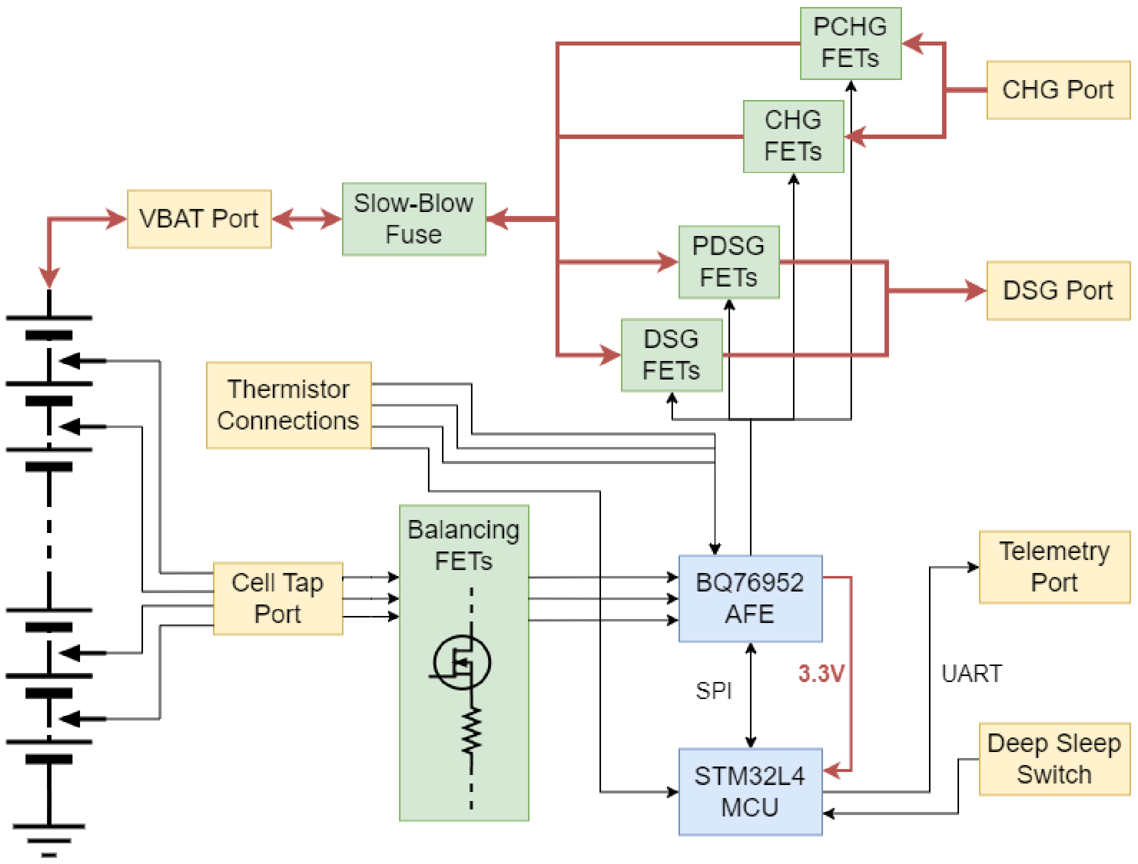


Figure 3.1: Block diagram representing the electrical architecture of the BMS PCB

3.1 Charge/Discharge Paths

Separate charging and discharging paths are used to reduce conduction losses (and thus improve efficiency) as well as reduce the total number of MOSFETs needed, as fewer MOSFETs were needed for the charge path due to the lower charging current required. Additionally, predischarge and precharge FETs were included in the design. The predischarge FET allows gradually raising the load voltage through a high-resistance path, preventing high inrush currents that could damage the FETs or trip the AFE's protection.

3.2 Cell Balancing FETs

The BQ76952 is capable of performing cell balancing using internal FETs and resistances. However, the current it is capable of balancing is too low for the 500mA requirement. As such, external FETs and resistors were used to increase the cell-balancing current sufficiently.

Low-pass filters are used on the cell-voltage inputs between the battery cell taps and the BQ76952 pins. This not only filters noise out from the voltage measurements, but also increases the series resistance on each cell voltage input to greatly reduce the amount of current and power dissipated within the BQ76952 when performing cell balancing. The reduced power dissipation allows for a greater number of cells to be balanced at once without overheating the BQ76952.

3.3 Thermistor Connections

Four thermistors are connected to the BMS PCB. Three are directly connected to the BQ76952's temperature sensing pins, with two to be affixed to the battery to measure the battery temperature and one to be attached to the DSG FETs to monitor their temperature. The BQ76952's internal logic allows for configuring independent protections against both the battery temperature and the FET temperature, ensuring neither exceed the maximum temperature setting.

The fourth thermistor is instead connected to one of the MCU's 12-bit ADC inputs, and is to be affixed to the battery as well. Although less precise than the AFE, it provides additional redundancy for measuring the battery temperature, which is one of the critical measurements to keep track of.

3.4 PCB Layout

A photo of the finished BMS PCBA is shown in Figure 3.2 below. Large sections of copper were dedicated to the main charge and discharge paths to ensure the PCB would be capable of conducting the high currents required by the design. The outer layers were built with 2oz copper weight to further improve the performance of the PCB, as well as provide plenty of thermal mass connected to the FETs to reduce the temperature rise at higher currents.

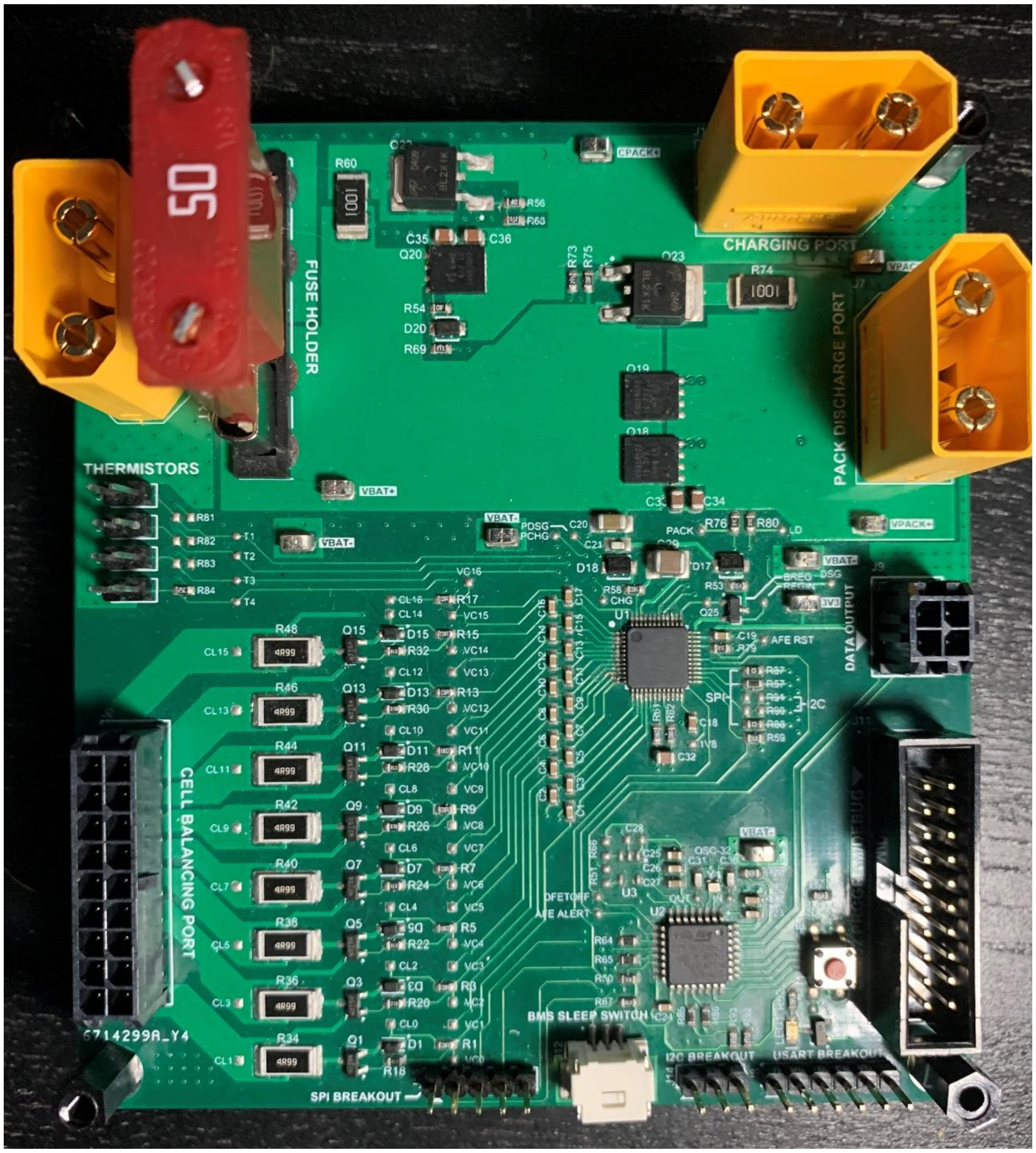


Figure 3.2: A fully assembled BMS PCBA

3.5 Resistor Ladder Companion PCB

In addition to the BMS PCB, a simple companion PCB was also designed that implemented a resistor ladder using potentiometers to provide an easy way of simulating the cell voltages for the BQ76952. A CAD model of this PCB is shown in Figure 3.3 below. The potentiometers allowed easily adjusting individual cell voltages to test the BQ76952's detection of over/undervoltage events and cell-balancing thresholds.

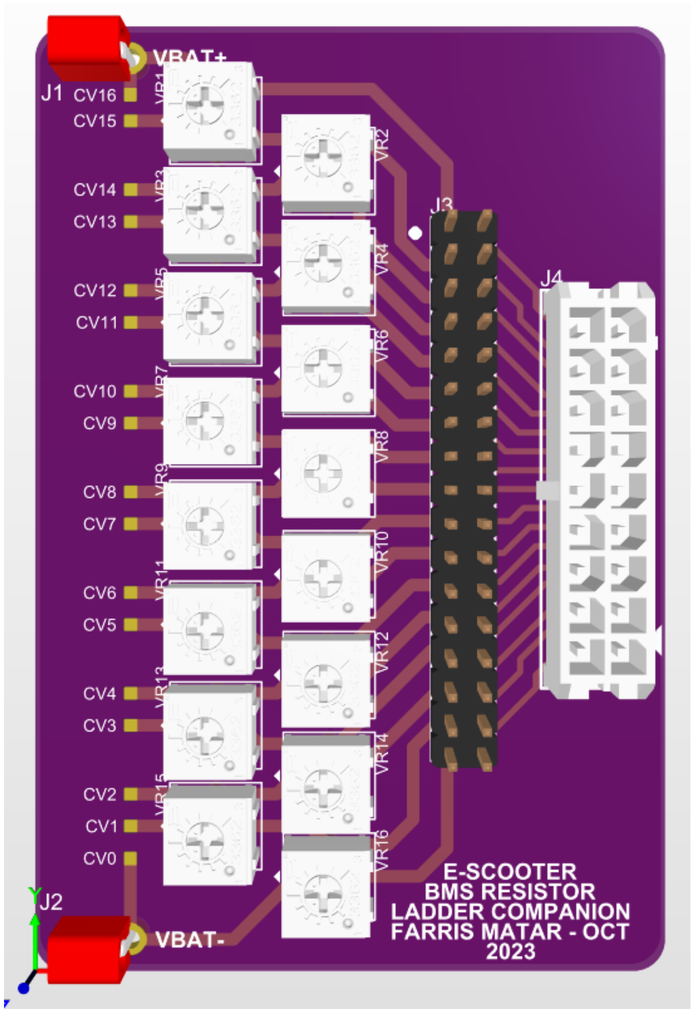
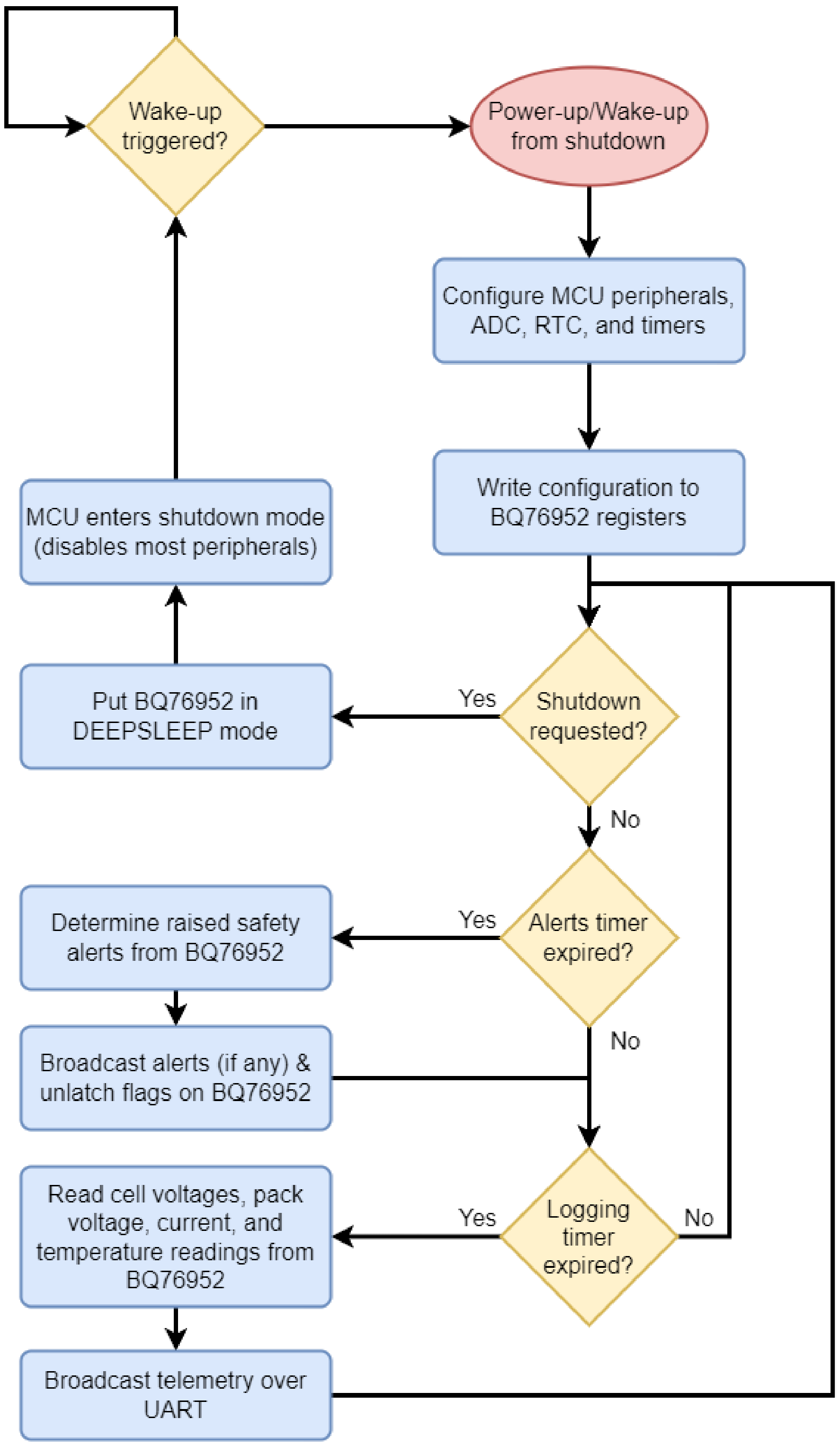


Figure 3.3: CAD model of the resistor ladder companion PCB

4 Software Design

All software for the BMS was written in C using STM32's CubeIDE. The source code can be found on GitHub here: https://github.com/far-mat42/custom-escooter/tree/main/BMS%20PCB/Testing/STM32%20Testing%20Ground/LTO-16S-BMS-SW

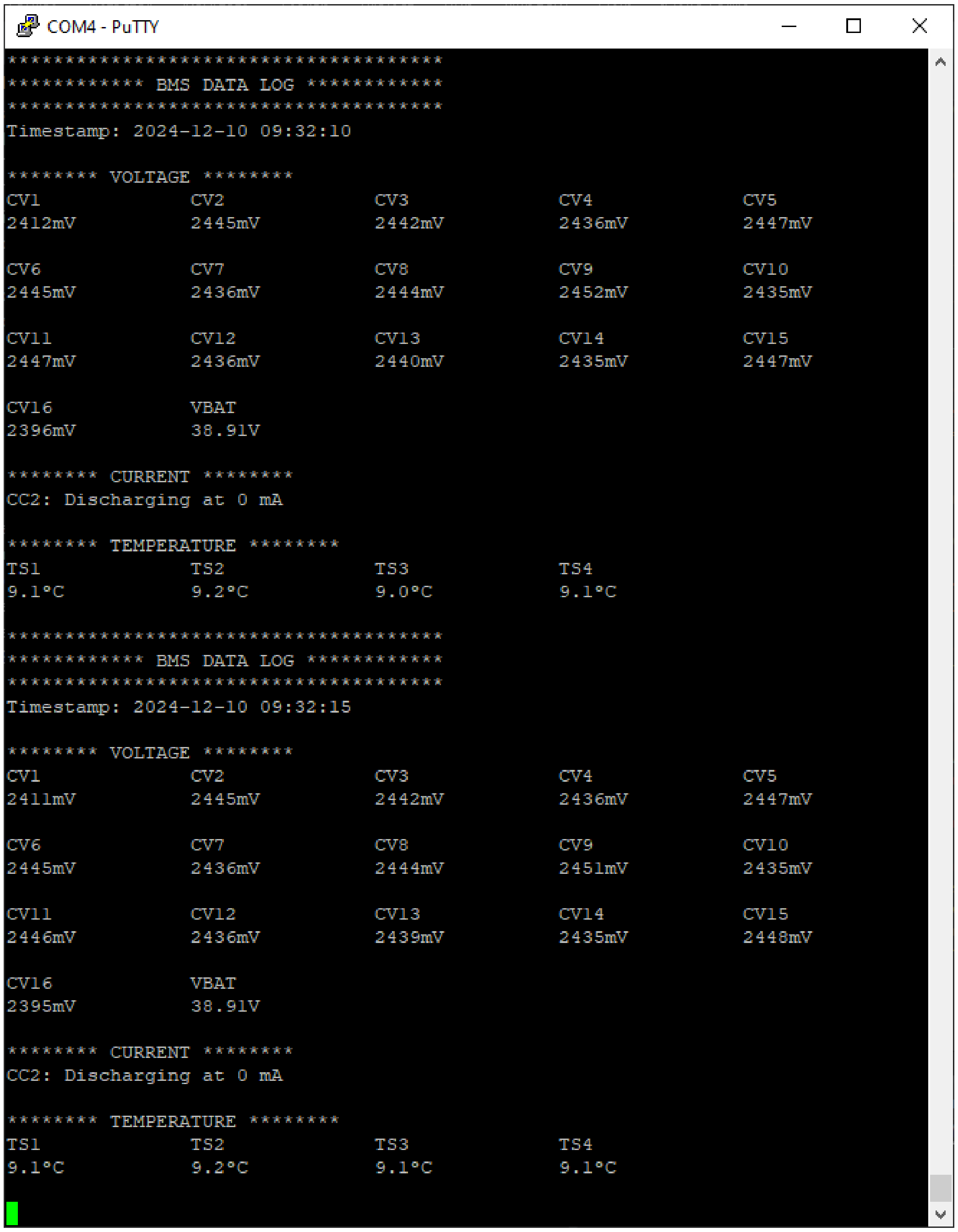
An overview of the software written for the BMS is shown in Figure 4.1 below. The vast majority of the BMS's operations are handled by the BQ76952 AFE. This includes measurements for voltage, current, and temperature, enabling the charge/discharge paths, and activating protections. As such, after setting up the configuration for the AFE, the STM32's is to simply retrieve the measurements and any detected safety alerts from the AFE and broadcast them over UART for downstream devices. The STM32 uses timers to check for safety alerts every 0.5 seconds, and retrieve measurements every 5 seconds. It also uses a real-time clock (RTC) to add timestamps to information broadcasted over UART, as well as an external interrupt to signal when to put the BMS in a low-power shutdown mode, deactivating most peripherals on the STM32 and AFE to save power until woken up again.



Wake-up triggered?
Power-up/Wake-up from shutdown
Configure MCU peripherals, ADC, RTC, and timers
Write configuration to BQ76952 registers
MCU enters shutdown mode (disables most peripherals)
Put BQ76952 in DEEPSLEEP mode
Yes
Shutdown requested?
No
Determine raised safety alerts from BQ76952
Yes
Alerts timer expired?
No
Broadcast alerts (if any) & unlatch flags on BQ76952
Read cell voltages, pack voltage, current, and temperature readings from BQ76952
Yes
Logging timer expired?
No
Broadcast telemetry over UART

Figure 4.1: Flowchart of the STM32's software

An example of what the broadcasted data looks like is shown in Figure 4.2 below. Note that for clearer representation of the data, additional formatting and stylization is added. The actual message transmitted over UART in the final design would be more simple and concise to reduce the time taken to transmit the data.



```
**************************************
************ BMS DATA LOG ************
**************************************
Timestamp: 2024-12-10 09:32:10

******** VOLTAGE ********
CV1             CV2             CV3             CV4             CV5
2412mV          2445mV          2442mV          2436mV          2447mV

CV6             CV7             CV8             CV9             CV10
2445mV          2436mV          2444mV          2452mV          2435mV

CV11            CV12            CV13            CV14            CV15
2447mV          2436mV          2440mV          2435mV          2447mV

CV16            VBAT
2396mV          38.91V

******** CURRENT ********
CC2: Discharging at 0 mA

******** TEMPERATURE ********
TS1             TS2             TS3             TS4
9.1°C           9.2°C           9.0°C           9.1°C

**************************************
************ BMS DATA LOG ************
**************************************
Timestamp: 2024-12-10 09:32:15

******** VOLTAGE ********
CV1             CV2             CV3             CV4             CV5
2411mV          2445mV          2442mV          2436mV          2447mV

CV6             CV7             CV8             CV9             CV10
2445mV          2436mV          2444mV          2451mV          2435mV

CV11            CV12            CV13            CV14            CV15
2446mV          2436mV          2439mV          2435mV          2448mV

CV16            VBAT
2395mV          38.91V

******** CURRENT ********
CC2: Discharging at 0 mA

******** TEMPERATURE ********
TS1             TS2             TS3             TS4
9.1°C           9.2°C           9.1°C           9.1°C
```

Figure 4.2: Example of two sets of logging data retrieved from the AFE and sent over UART

5 Testing and Validation

5.1 AFE-MCU SPI Communication

The HAL APIs for the STM32 were used to implement SPI communication with the BQ76952. According to the BQ76952 Technical Reference Manual, the BQ76952 communicates using the following 24-bit data frame in Figure 5.1. Note that the MSB of the register address byte is used as the R/W bit.

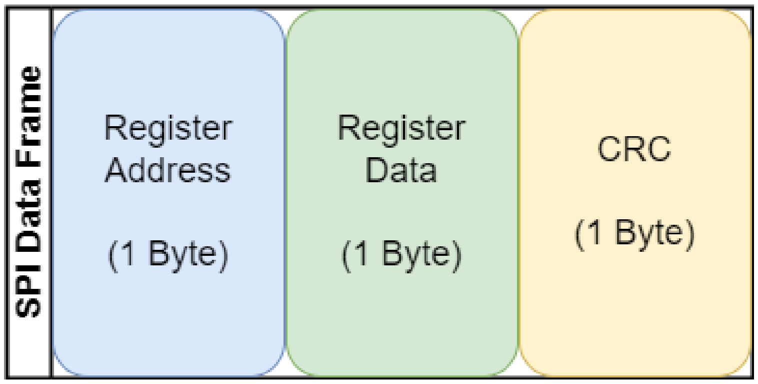


Figure 5.1: SPI data frame for BQ76952's SPI communication

The BQ76952 acts as a SPI slave and communicates in one of 3 ways. The first is through direct commands, in which a command address is written and the relevant data is read/written all within a single SPI data frame. The second is through subcommands, in which over 2 SPI data frames, a 2-byte subcommand address is written to the BQ76952, after which the data (if any) is written/read in subsequent SPI data frames from the BQ76952's data buffer. The third is through accessing the RAM registers, in which over multiple SPI data frames, a 2-byte RAM register address is written to the BQ76952, after which the register data is written to/read from the data buffer, followed by the checksum and data length of the RAM data. A depiction of the data frames for each of these methods is shown in Figure 5.2.

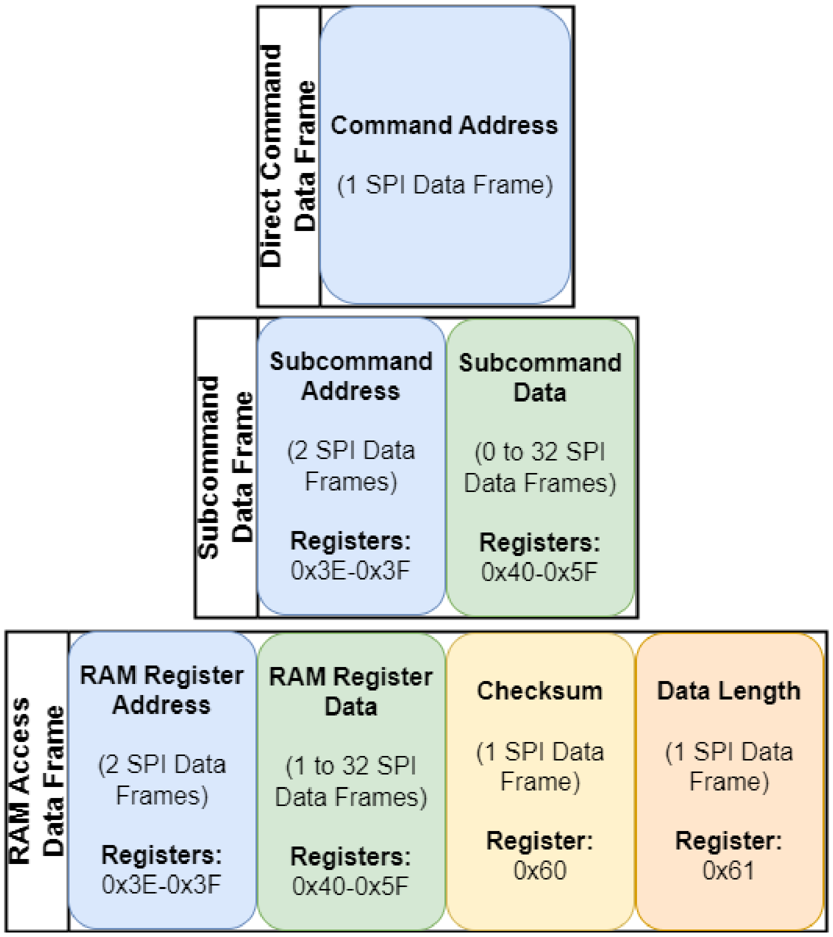


Figure 5.2: Data frames for the direct commands, subcommands, and RAM access communications with the BQ76952

The STM32 was programmed to send and receive SPI messages according to the specified data frames, calculating the CRC for each SPI data frame as well as the checksum for RAM access communications. To validate the communication between the STM32 and BQ76952, Saleae's Logic 8 logic analyzer was used to analyze the SPI communication (https://www.saleae.com/). The Saleae was connected to the BMS as shown in Figure 5.3, with the clips easily connecting to the SPI breakout headers on the BMS.

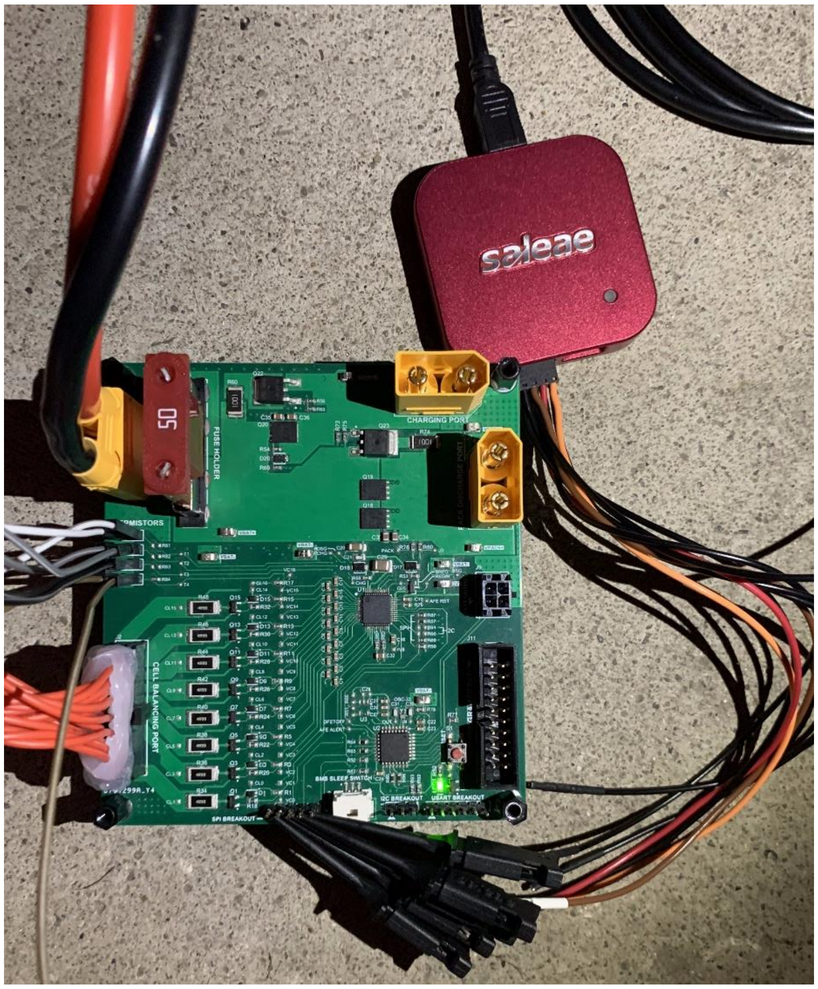


Figure 5.3: Connection of the Saleae Logic 8 to the BMS

Using Saleae's Logic 2 software, multiple samples of the SPI communication between the STM32 and BQ76952 were recorded. Figure 5.4 below shows an example of 2 attempts at sending a single SPI data frame. A direct command is sent to the BQ76952 twice – the first time, the BQ76952 replies with the data frame processed previously, the second time the BQ76952 repeats the data frame. The repeated data frame indicates the BQ76952 has received the command and processed it, replying with the relevant data in the data byte (0x00). The Saleae Logic 2 software easily helps with verifying the data by displaying the written bytes on the MISO and MOSI lines according to the CLK line. Figure 5.5 shows another example with a series of direct commands in a row, with the STM32 re-attempting each command until the BQ76952 confirms it receives it and sends the requested data.

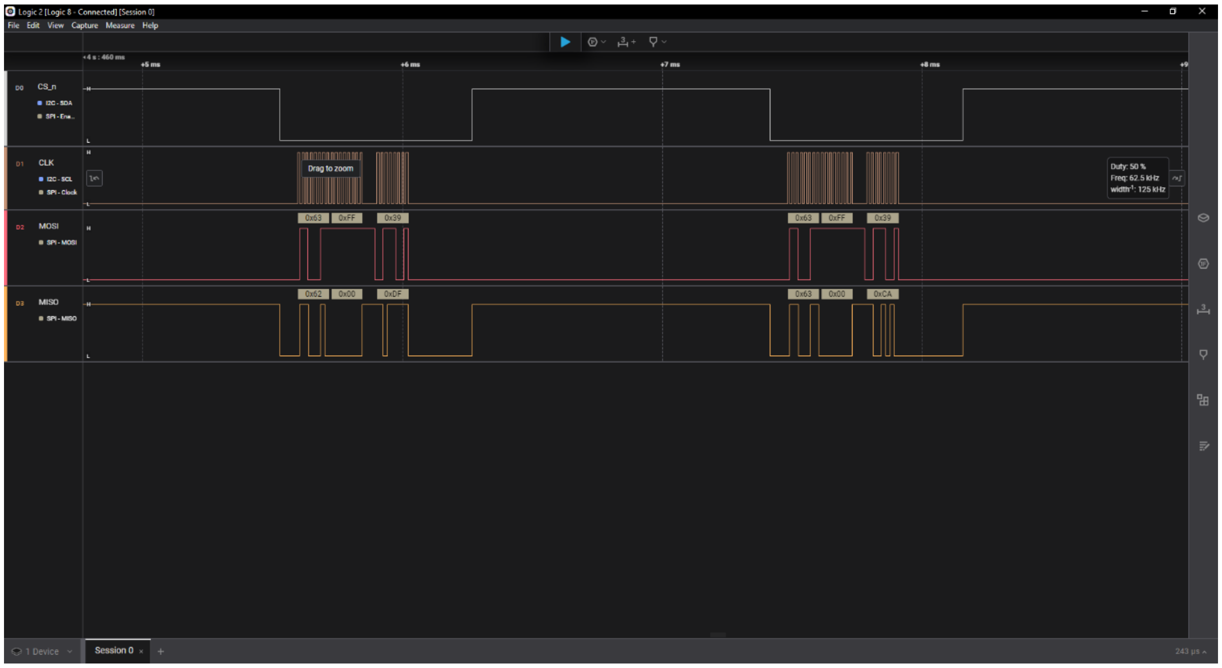


Figure 5.4: Example of a direct command between the STM32 and BQ76952

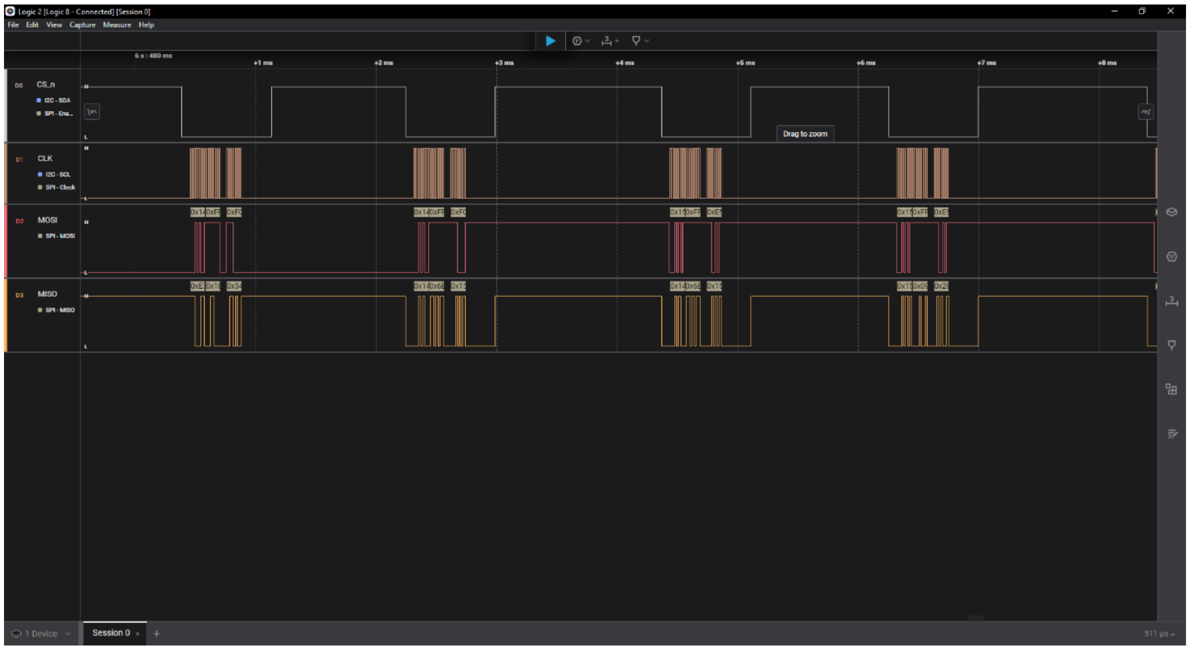
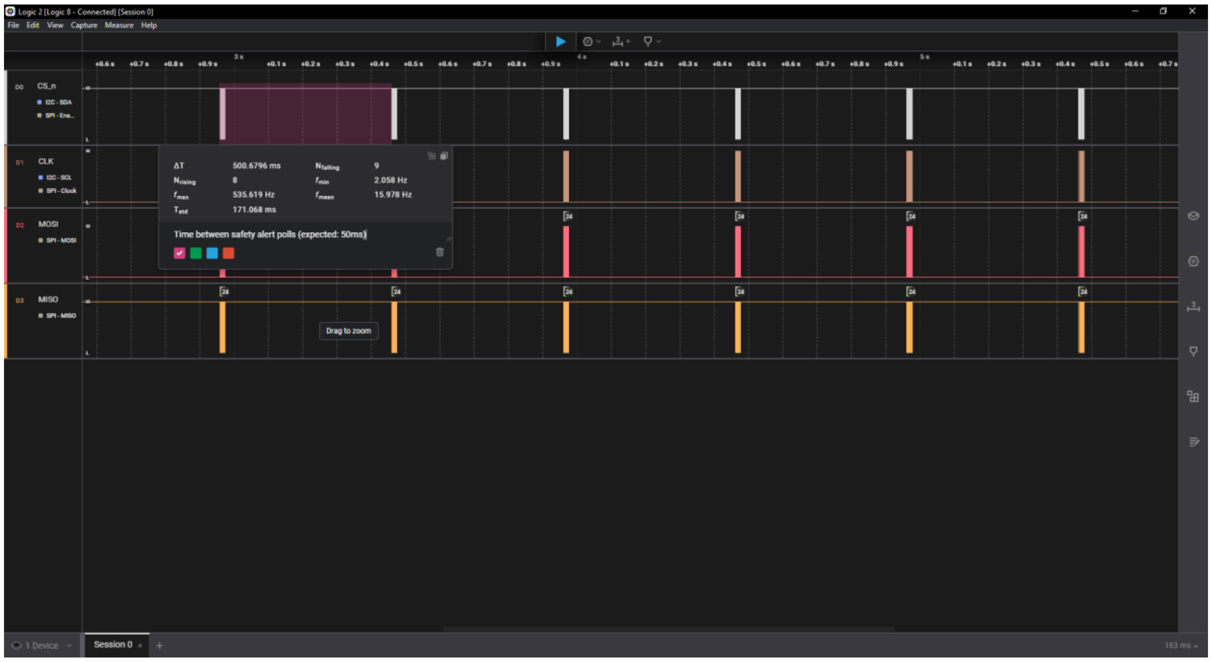


Figure 5.5: Example of a series of direct commands

The Logic 2 software also allowed verifying the timing for the polling of the safety alerts and measurements data with its measurement tool. Figure 5.6 shows an example of measuring the polling period for safety alerts.



ΔT	500.6796 ms	$N_{falling}$	9
N_{rising}	8	f_{min}	2.058 Hz
f_{max}	535.619 Hz	f_{mean}	15.978 Hz
T_{std}	171.068 ms		

Time between safety alert polls (expected: 50ms)

Figure 5.6: Measuring the polling period between polls for safety alerts from the BQ76952

Using the Saleae logic analyzer and software in the future is highly recommended. Using the logic analyzer is very straightforward and it easily connects to the headers on the PCBA. The software is also easy to use and offered plenty of helpful tools and analysis for validating the STM32's configuration for SPI communication and the successful communication of data between the STM32 and BQ76952.

5.2 Cell Voltage Measurement Accuracy

To verify the accuracy of the cell voltage measurements, the BMS was used in conjunction with the Resistor Ladder Companion with all potentiometers set to roughly the same resistance. A power supply was used to mimic the battery voltage and the Resistor Ladder Companion divided the supply down into the 16 cell voltage inputs for the BQ76952.

The accuracy was tested by individually measuring each cell voltage with a Keysight 34461A 6.5 digit DMM and comparing this measured voltage with the cell voltage reported by the BQ76952. The DMM reading was taken as the correct value, and the error in the BQ76952's measurement was calculated based on this. Up to 10 samples were recorded for each cell voltage, and the test was run with the power supply (i.e. the battery voltage) set to 30V, 35V, 40V, and 45V to check the accuracy across the expected operating range of the battery voltage.

A plot summarizing the results of the accuracy tests is shown below in Figure 5.7.

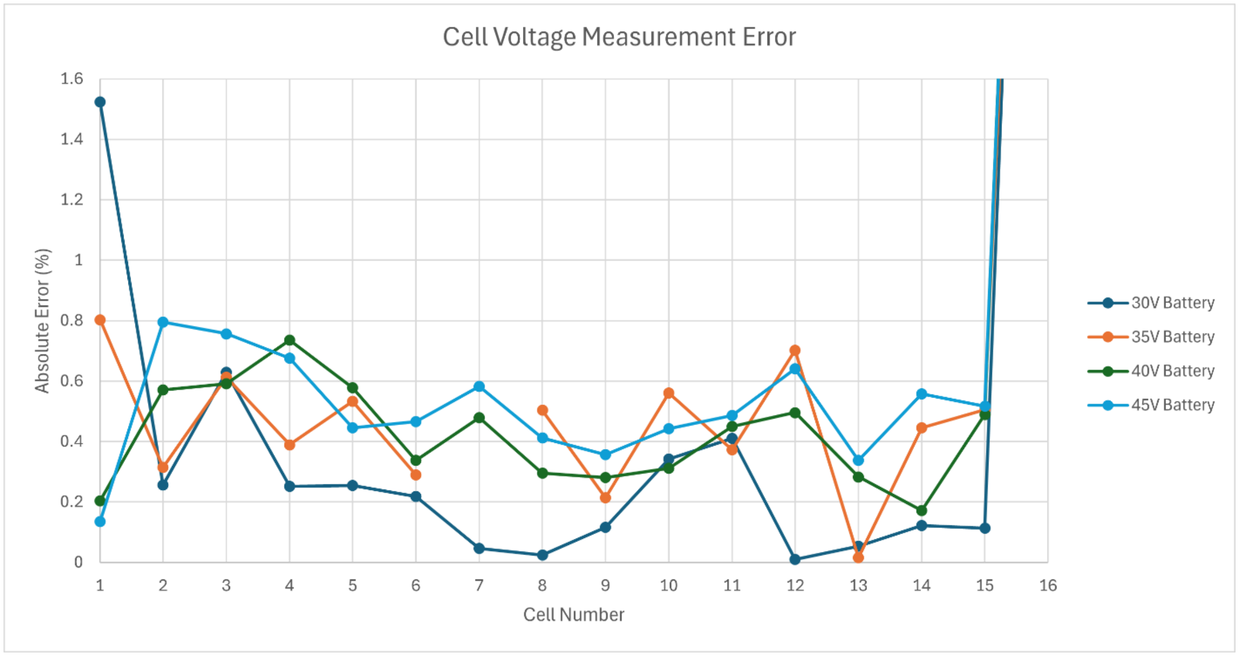


Figure 5.7: Plot of the average error for each cell's voltage measurement at different battery voltages

As seen in the plot, the average error for cells 1 to 15 is comfortably below the 1% requirement for the BMS. The only exception was cell 16, which consistently had about +5-6% of error on the measurement. This was resolved by calibrating the gain for cell 16's voltage measurements to be slightly lower than the other cells. After calibration, the voltage measurements for cell 16 were within the 1% error requirement as well, successfully achieving the requirement set for this BMS.

5.3 Discharge PCB Thermal Tests

In order to verify the BMS PCBA was capable of supporting the peak discharge current requirement, the discharge terminal was connected to an e-load to assess how high of a load the PCBA could handle, while a thermal camera was used to examine the temperature of the FETs and PCB. To start, a thermal image of the BMS with no load applied was taken to give a reference point of the idle/ambient temperature. The picture is shown in Figure 5.8, with ambient temperature being around 27ºC.

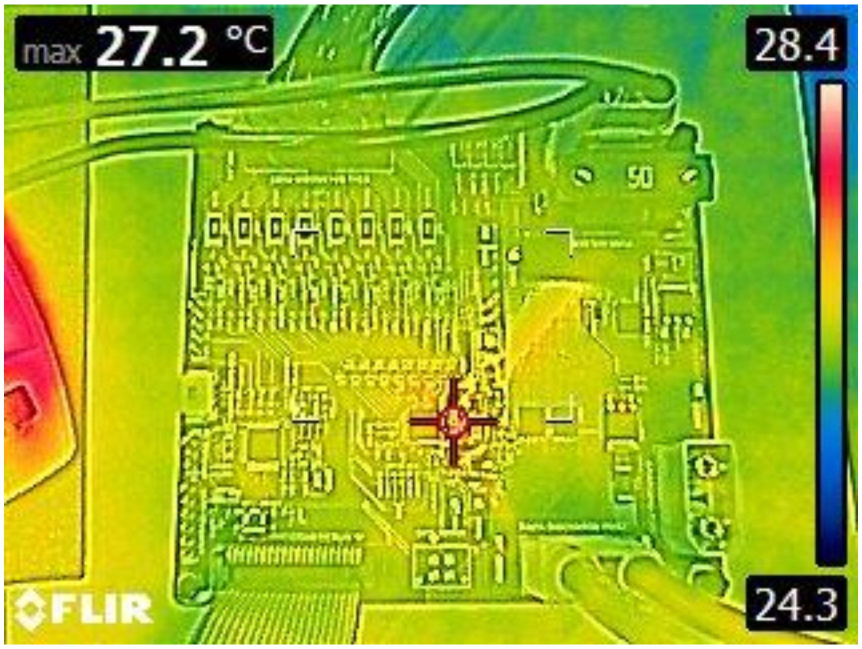


Figure 5.8: Thermal overlay of the BMS when idle (no load)

The load was gradually stepped up in small increments all the way up to 36A, the maximum value tested. The thermal image of the BMS after supplying this load for 5 minutes continuously is given below in Figure 5.9. As shown, the PCBA handles the high current with no issues, with the hottest components being the DSG FETs which only get up to about 78ºC.

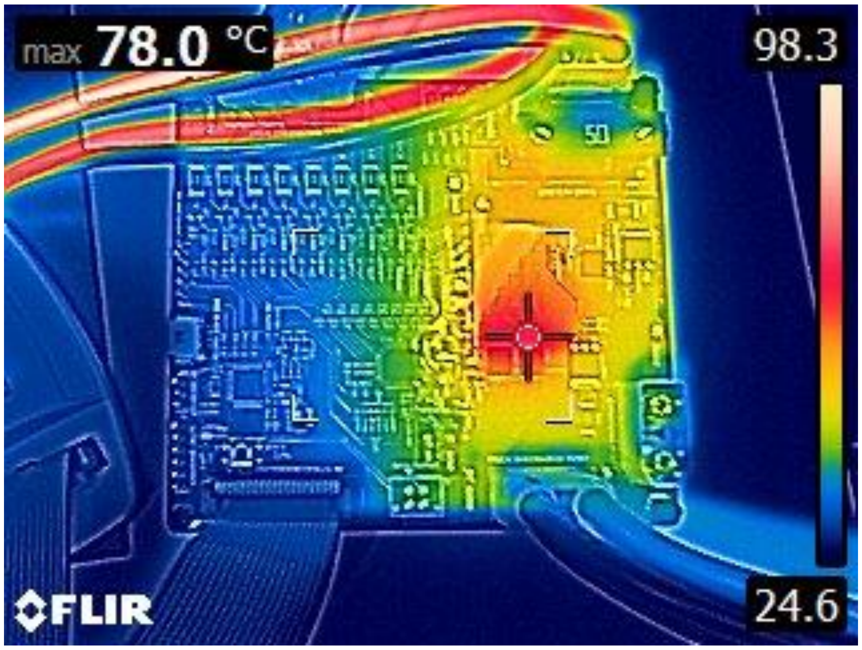


Figure 5.9: Thermal overlay of the BMS after discharging 36A for 5 minutes

6 Potential Improvements

6.1 Real-time Clock Circuit

Currently, the STM32’s RTC requires an initial starting date and time before it can keep track of time while running. The RTC resets every time the STM32 enters shutdown to conserve power, which means it doesn’t accurately keep track of time. In a future revision of the BMS, it would be good to include a circuit solely responsible for acting as the RTC, so that the STM32 could update its own RTC’s starting time on startup. The DS1307 for example could fill this role, consuming very low current when in battery backup mode, which would ensure the power consumption of the BMS remains low when in shutdown.

6.2 Unseparated Charge and Discharge Paths

Although the separation of the charge and discharge paths reduces inefficiency, it also forces the system to only work unidirectionally. This causes issues with components that can operate with bidirectional power flow. A clear example of how this would be beneficial is with the motor, as allowing for bidirectional power flow would allow for the use of regenerative braking to conserve battery power. Thus, it would be good to have future revisions use a combined charge and discharge path through two sets of MOSFETs, as many BMS circuits do.

6.3 Support for Lower Cell Counts

The current BMS only supports 16s batteries, and although the BQ76952 can be configured to work with different voltage and current ranges, it would be extremely difficult to modify the PCBA to work with a battery with fewer series cells. The BQ76952 can work with batteries with as few as 3 cells in series, but the cell voltage connections must be made properly to do so. To make the BMS more modular and allow it to adapt to changes in the battery sizing, it would be nice to provide the option to easily configure the hardware to work with lower cell counts, such as with DNP components and footprints.

6.4 Additional Thermistor Inputs

In the final design, the BQ76952’s DCHG, DDSG, ALERT, and DFETOFF pins were unused. The BQ76952 allows using these pins as additional thermistor inputs, so it would be beneficial in future revisions to implement this to acquire more temperature samples, whether they be for the battery, FETs, or other components.

6.5 Coulomb Counting

The BQ76952 has a coulomb counter for tracking the total amount of charge remaining in the battery. However, this information is not currently being recorded by the STM32 software, as properly implementing it will require understanding how the coulomb counter responds to battery charging and the logic for resetting the counter when the battery is fully charged requires information from the charger. Once the battery charger is designed and built, the coulomb counting can be implemented for accurate tracking of the battery's remaining capacity.

6.6 PCB Errata

This section outlines some minor mistakes in the PCB design that should be corrected in future revisions.

6.6.1 Flipped Diodes/FETs in Gate Drive Circuitry

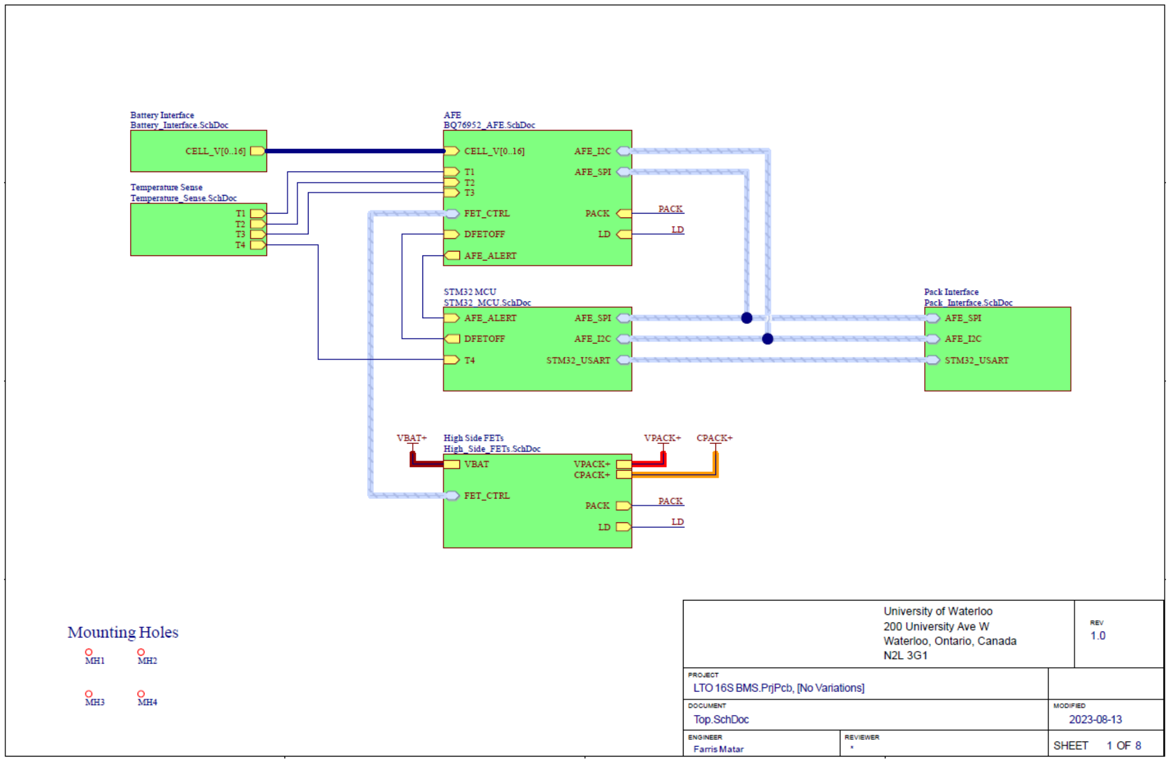
Some of the Zener diodes and one of the MOSFETs in the gate drive circuitry are in the wrong orientation. These are called out in the High Side Power FETs schematic in Appendix 7.1.

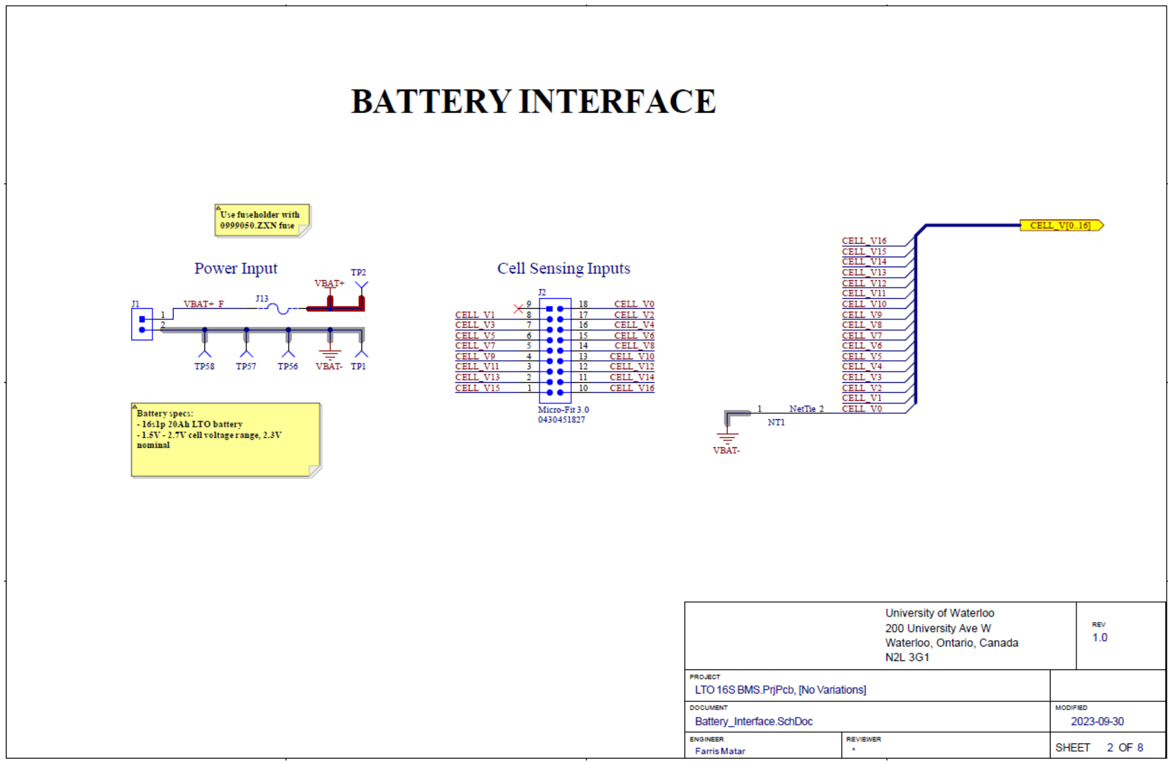
6.6.2 STM32 Heartbeat Pull-down

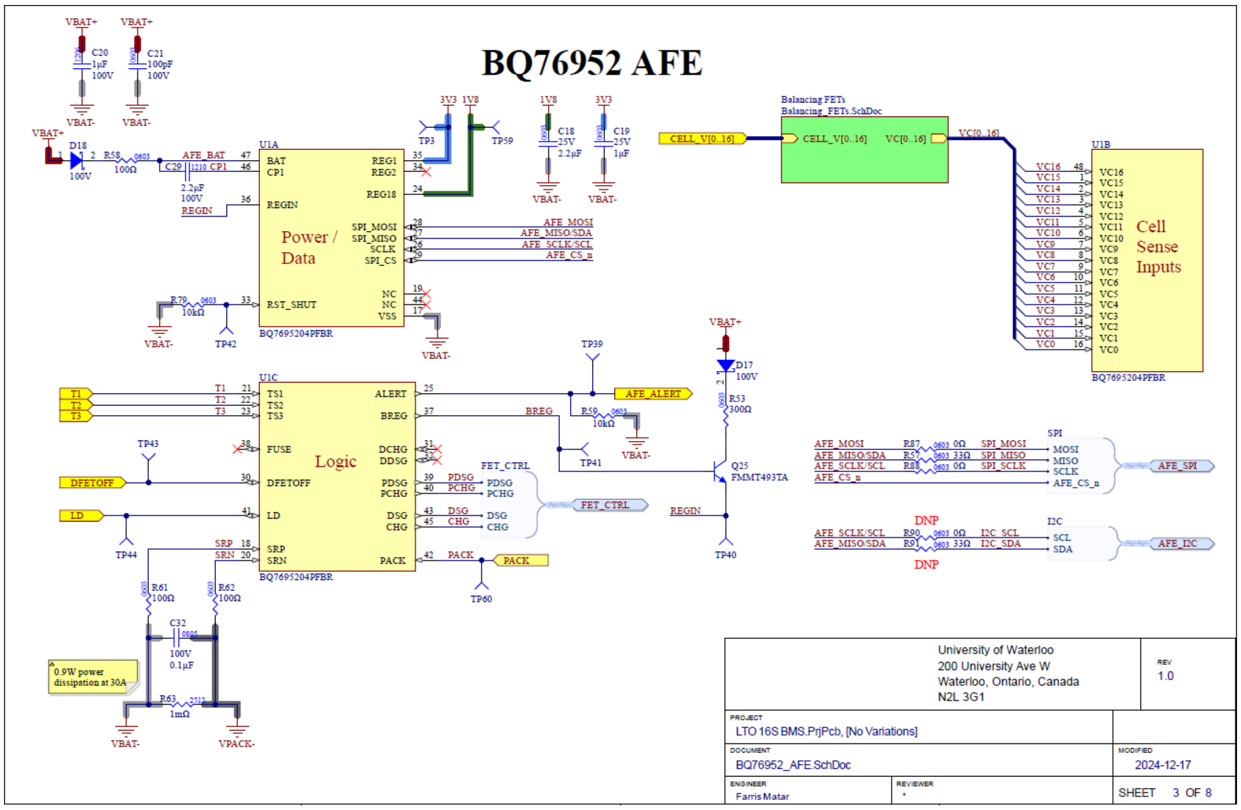
It was noticed that when the STM32 was shutdown, the heartbeat signal (which activated a FET to blink a status LED) went floating. The voltage would gradually rise (likely due to charge leaking from somewhere) and slowly turn the FET and LED on, greatly increasing the current consumption of the BMS when it is supposed to be shutdown. Future revisions should add a weak pull-down on this line to prevent the LED from turning on this way and consuming power unnecessarily.

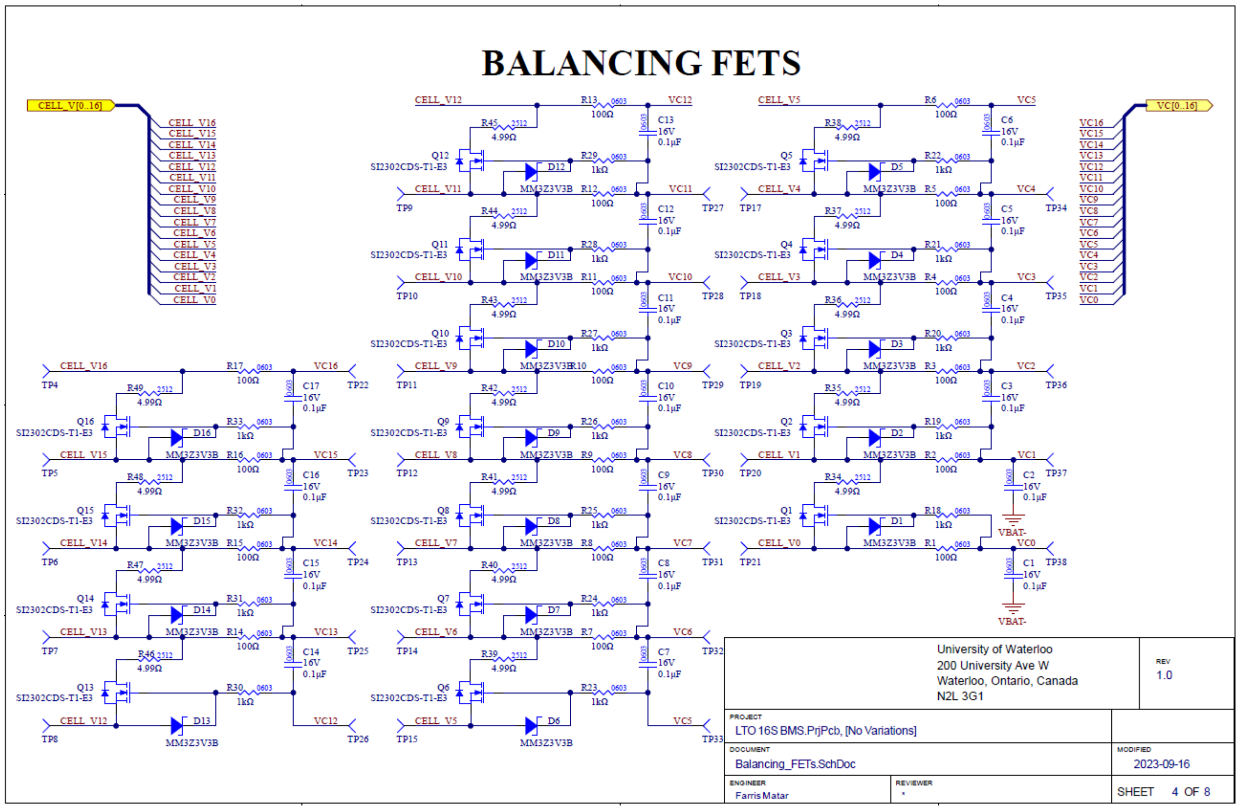
7 Appendix

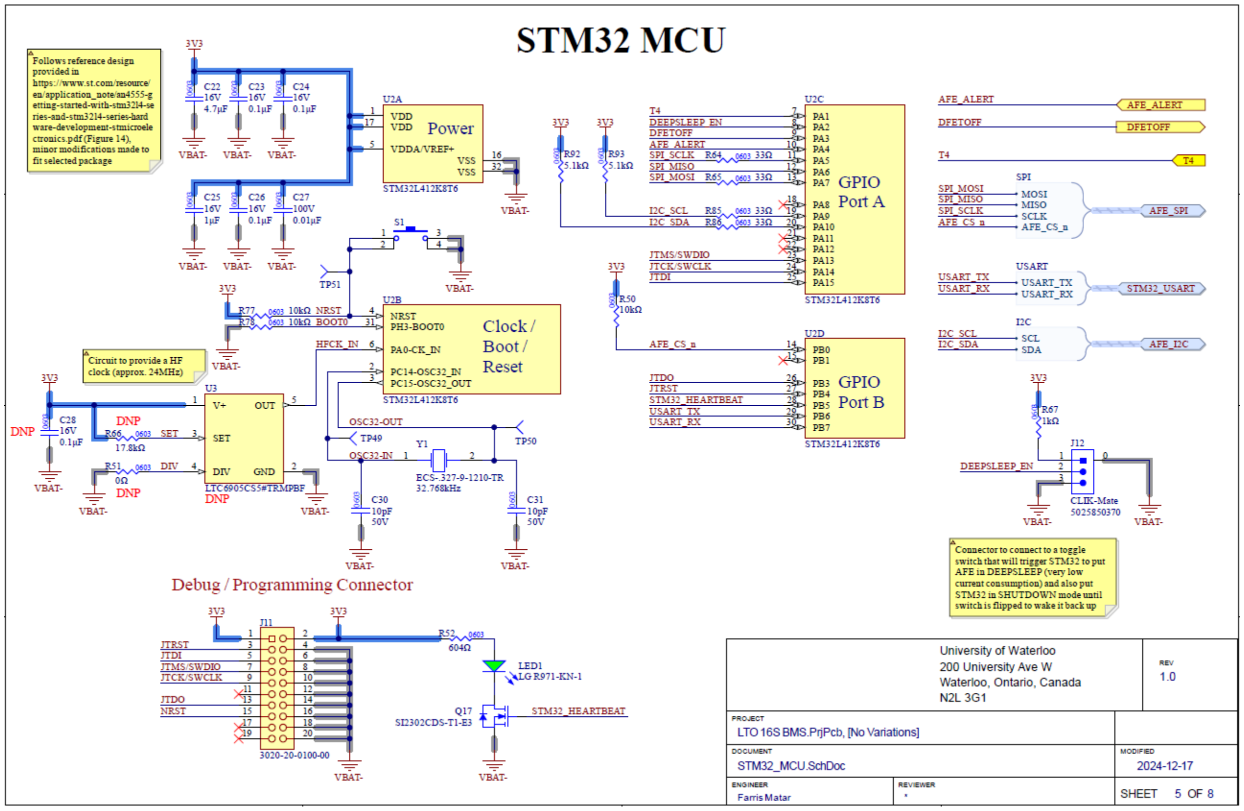
7.1 BMS Schematic

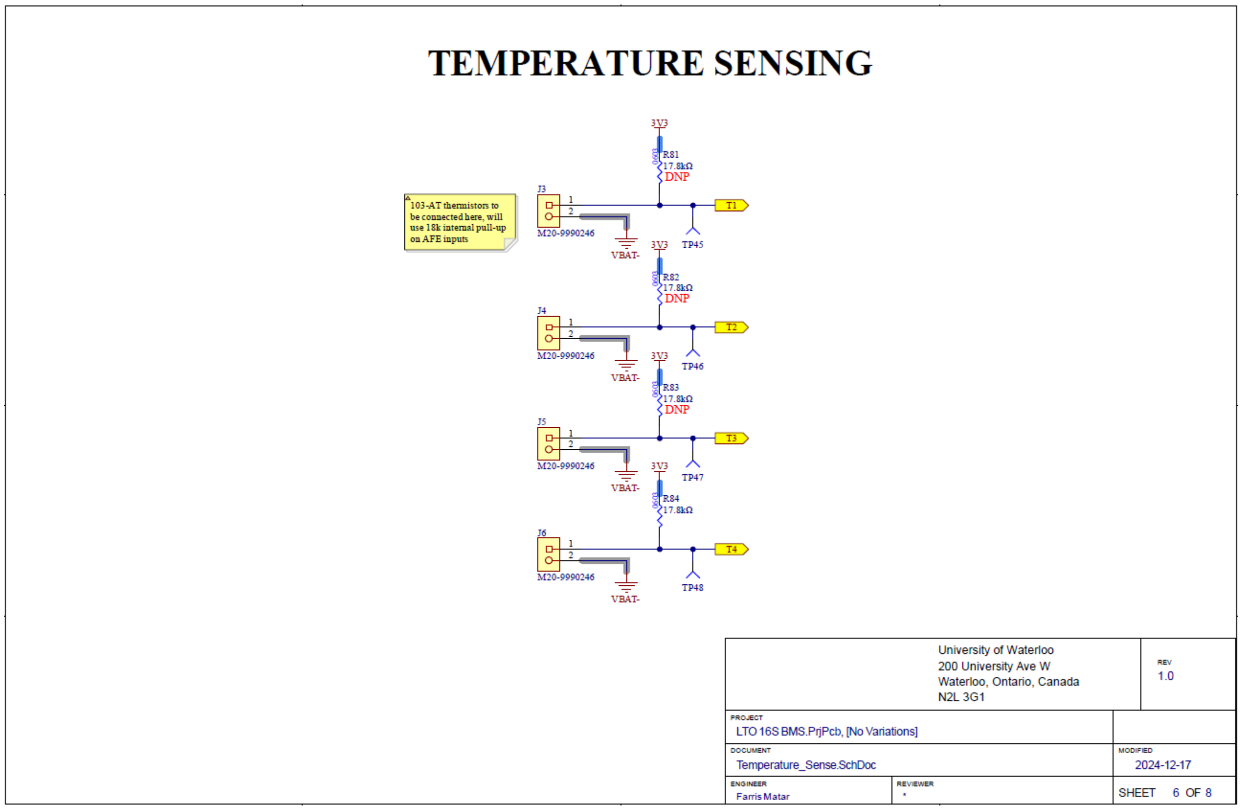


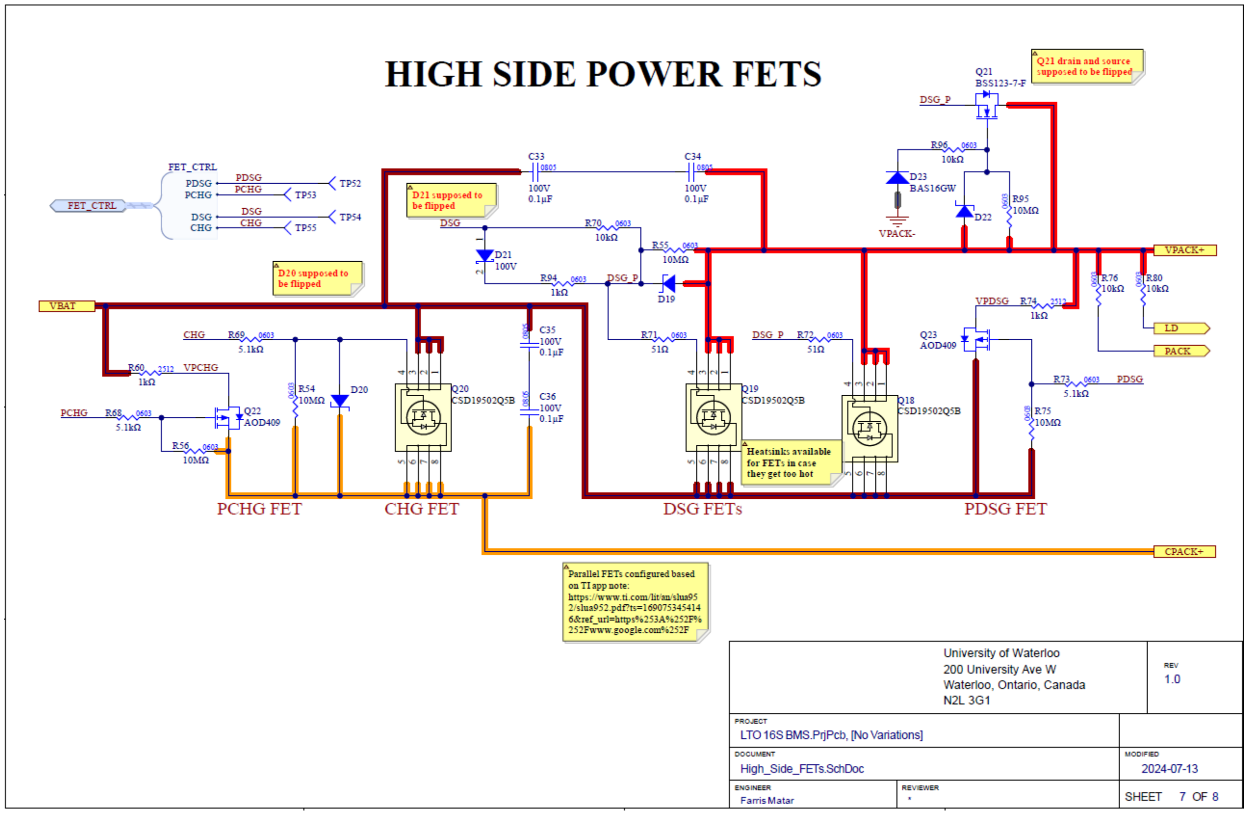



BQ76952 AFE
Power / Data
Logic
Cell Sense Inputs
Balancing FETs
Balancing_FETs.SchDoc
U1A
U1B
U1C
BQ7695204PFBR
FET_CTRL
AFE_ALERT
AFE_SPI
AFE_I2C
DNP
0.9W power dissipation at 30A
University of Waterloo
200 University Ave W
Waterloo, Ontario, Canada
N2L 3G1
REV 1.0
PROJECT LTO 16S BMS.PrjPcb, [No Variations]
DOCUMENT BQ76952_AFE.SchDoc
MODIFIED 2024-12-17
ENGINEER Farris Matar
REVIEWER *
SHEET 3 OF 8

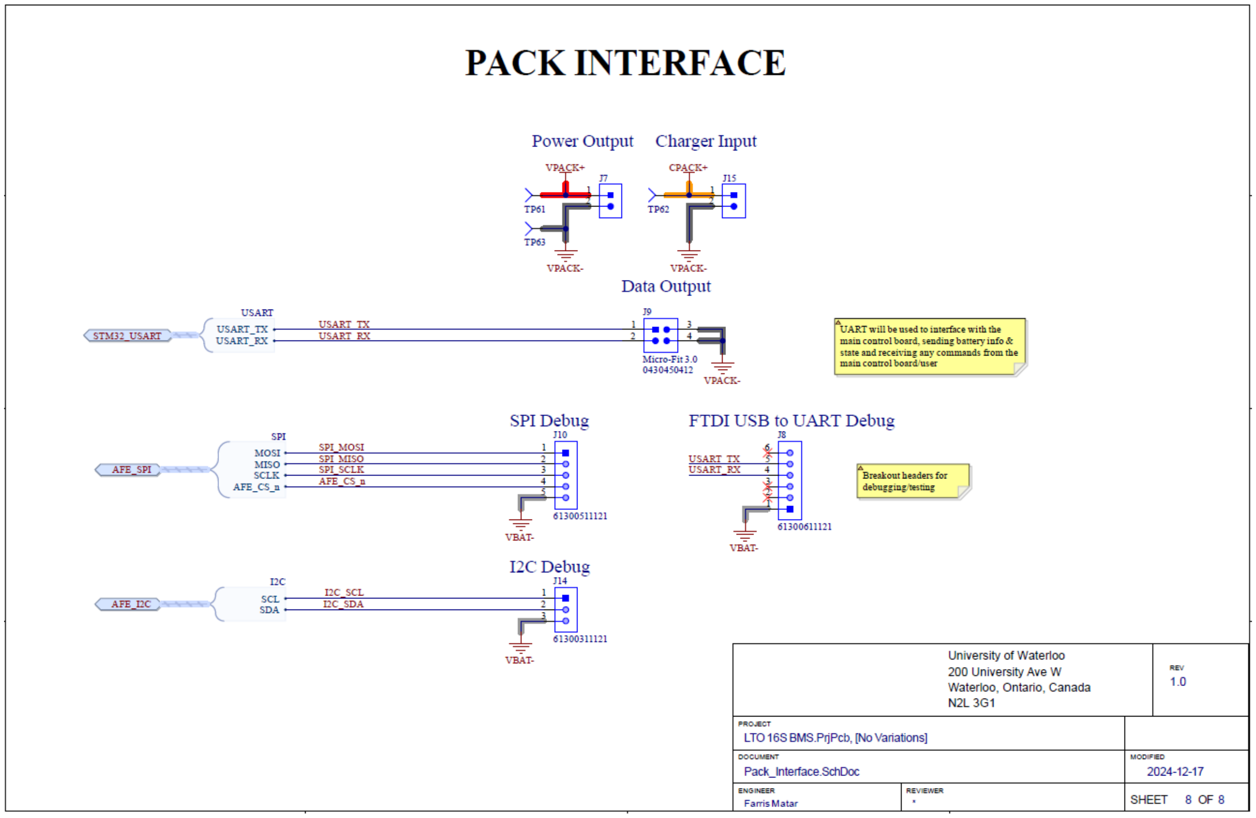

BALANCING FETS
CELL_V[0..16]
VC[0..16]
SI2302CDS-T1-E3
MM3Z3V3B
University of Waterloo
200 University Ave W
Waterloo, Ontario, Canada
N2L 3G1
REV 1.0
PROJECT LTO 16S BMS.PrjPcb, [No Variations]
DOCUMENT Balancing_FETs.SchDoc
MODIFIED 2023-09-16
ENGINEER Farris Matar
REVIEWER *
SHEET 4 OF 8


STM32 MCU
Follows reference design provided in https://www.st.com/resource/en/application_note/an4555-getting-started-with-stm32l4-series-and-stm32l4-series-hardware-development-stmicroelectronics.pdf (Figure 14), minor modifications made to fit selected package
Power
GPIO Port A
GPIO Port B
Clock / Boot / Reset
Circuit to provide a HF clock (approx. 24MHz)
STM32L412K8T6
LTC6905CS5#TRMPBF
DNP
ECS-.327-9-1210-TR 32.768kHz
Debug / Programming Connector
3020-20-0100-00
LED1 LG R971-KN-1
Q17 SI2302CDS-T1-E3
STM32_HEARTBEAT
AFE_ALERT
DFETOFF
T4
AFE_SPI
STM32_USART
AFE_I2C
J12 CLIK-Mate 5025850370
Connector to connect to a toggle switch that will trigger STM32 to put AFE in DEEPSLEEP (very low current consumption) and also put STM32 in SHUTDOWN mode until switch is flipped to wake it back up
University of Waterloo
200 University Ave W
Waterloo, Ontario, Canada
N2L 3G1
REV 1.0
PROJECT LTO 16S BMS.PrjPcb, [No Variations]
DOCUMENT STM32_MCU.SchDoc
MODIFIED 2024-12-17
ENGINEER Farris Matar
REVIEWER *
SHEET 5 OF 8


TEMPERATURE SENSING
103-AT thermistors to be connected here, will use 18k internal pull-up on AFE inputs
R81 17.8kΩ DNP
R82 17.8kΩ DNP
R83 17.8kΩ DNP
R84 17.8kΩ
M20-9990246
T1
T2
T3
T4
TP45
TP46
TP47
TP48
University of Waterloo
200 University Ave W
Waterloo, Ontario, Canada
N2L 3G1
REV 1.0
PROJECT LTO 16S BMS.PrjPcb, [No Variations]
DOCUMENT Temperature_Sense.SchDoc
MODIFIED 2024-12-17
ENGINEER Farris Matar
REVIEWER *
SHEET 6 OF 8



HIGH SIDE POWER FETS
FET_CTRL
PDSG
PCHG
DSG
CHG
TP52
TP53
TP54
TP55
Q21 drain and source supposed to be flipped
Q21 BSS123-7-F
DSG_P
R96 1kΩ
D23 BAS16GW
D22
R95 10MΩ
VPACK-
D21 supposed to be flipped
D20 supposed to be flipped
C33 100V 0.1µF
C34 100V 0.1µF
R70 10kΩ
D21 100V
R55 10MΩ
R94 1kΩ
D19
VPACK+
R76 10kΩ
R80 10kΩ
LD
PACK
VBAT
VPDSG
R74 1kΩ
Q23 AOD409
R69 5.1kΩ
C35 100V 0.1µF
R71 51Ω
R72 51Ω
R60 1kΩ
VPCHG
R54 10MΩ
D20
Q20 CSD19502Q5B
Q19 CSD19502Q5B
Q18 CSD19502Q5B
R73 5.1kΩ
R68 5.1kΩ
Q22 AOD409
C36 100V 0.1µF
R75 10MΩ
R56 10MΩ
Heatsinks available for FETs in case they get too hot
PCHG FET
CHG FET
DSG FETs
PDSG FET
CPACK+
Parallel FETs configured based on TI app note: https://www.ti.com/lit/an/slua952/slua952.pdf?ts=1690753454146&ref_url=https%253A%252F%252Fwww.google.com%252F
University of Waterloo
200 University Ave W
Waterloo, Ontario, Canada
N2L 3G1
REV 1.0
PROJECT LTO 16S BMS.PrjPcb, [No Variations]
DOCUMENT High_Side_FETs.SchDoc
MODIFIED 2024-07-13
ENGINEER Farris Matar
REVIEWER *
SHEET 7 OF 8



PACK INTERFACE
Power Output
Charger Input
VPACK+
CPACK+
J7
J15
TP61
TP62
TP63
VPACK-
Data Output
USART
STM32_USART
USART_TX
USART_RX
J9
Micro-Fit 3.0 0430450412
UART will be used to interface with the main control board, sending battery info & state and receiving any commands from the main control board/user
SPI Debug
FTDI USB to UART Debug
SPI
AFE_SPI
MOSI
MISO
SCLK
AFE_CS_n
SPI_MOSI
SPI_MISO
SPI_SCLK
J10
61300511121
J8
61300611121
Breakout headers for debugging/testing
VBAT-
I2C Debug
I2C
AFE_I2C
SCL
SDA
I2C_SCL
I2C_SDA
J14
61300311121
University of Waterloo
200 University Ave W
Waterloo, Ontario, Canada
N2L 3G1
REV 1.0
PROJECT LTO 16S BMS.PrjPcb, [No Variations]
DOCUMENT Pack_Interface.SchDoc
MODIFIED 2024-12-17
ENGINEER Farris Matar
REVIEWER *
SHEET 8 OF 8