1981 Honda CB750c Electric Conversion

Dino Mastropietro with Mihir Juvvadi and Luke Wing
Faculty Advisorship by Jason Siegel

May 7, 2022

Abstract

This report serves as a progress update on the electric conversion of a 1981 Honda CB750c. In it, I will outline our system level design choices and discuss implementation details of all components related to the electrical system. Having completed most major design work, we are currently assembling the battery pack and intend to have a fully compliant and road-worthy vehicle completed in the coming months.

Motivation

With ambitions high and a shared love of motorcycles/automobiles held in our hearts, we set out to build this motorcycle as a challenge to our engineering abilities. Current industry trends in electrification and connectivity make our work timely as all major OEMs race to electrify their fleets and connect their vehicles to cloud infrastructure.

Many electric conversions have been done on classic cars and bikes as a way to breathe new life into these beautiful machines which house outdated technology (dirty internal combustion engines). We want to separate ourselves from these projects by including connected vehicle technology that will bring visibility and flexibility into our build and hopefully lay a foundation for many builds to come.

Goals

When completed, we would like to have a legally-compliant, roadworthy motorcycle that is connected to the cloud. In general, compliant means we will be able to have the bike inspected, registered, insured, and upholding NHTSA and ISO 26262 safety standards.[1, 2] Roadworthy means we would like to reach highway speeds, quickly accelerate and brake in safety-critical situations, and traverse over any terrain the original bike could have traversed. Connecting the vehicle to the cloud will be explicated in later sections.

Systems

Every part of the electrical system must work in tandem with one another to produce the safe, reliable operation desired. In the following sections I explain our design choices, component selection, and implementation details along with challenges we have faced and analyses of current difficulties that we are actively trying to solve. Sections are ordered with respect to their relative importance for safely operating the motorbike. On the next page, find a high-level diagram of the major electrical components.

---

[1] https://www.nhtsa.gov/DOT/NHTSA/Rulemaking/Articles/Associated%20Files/mcpkg002.pdf

[2] https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-2:v1:en

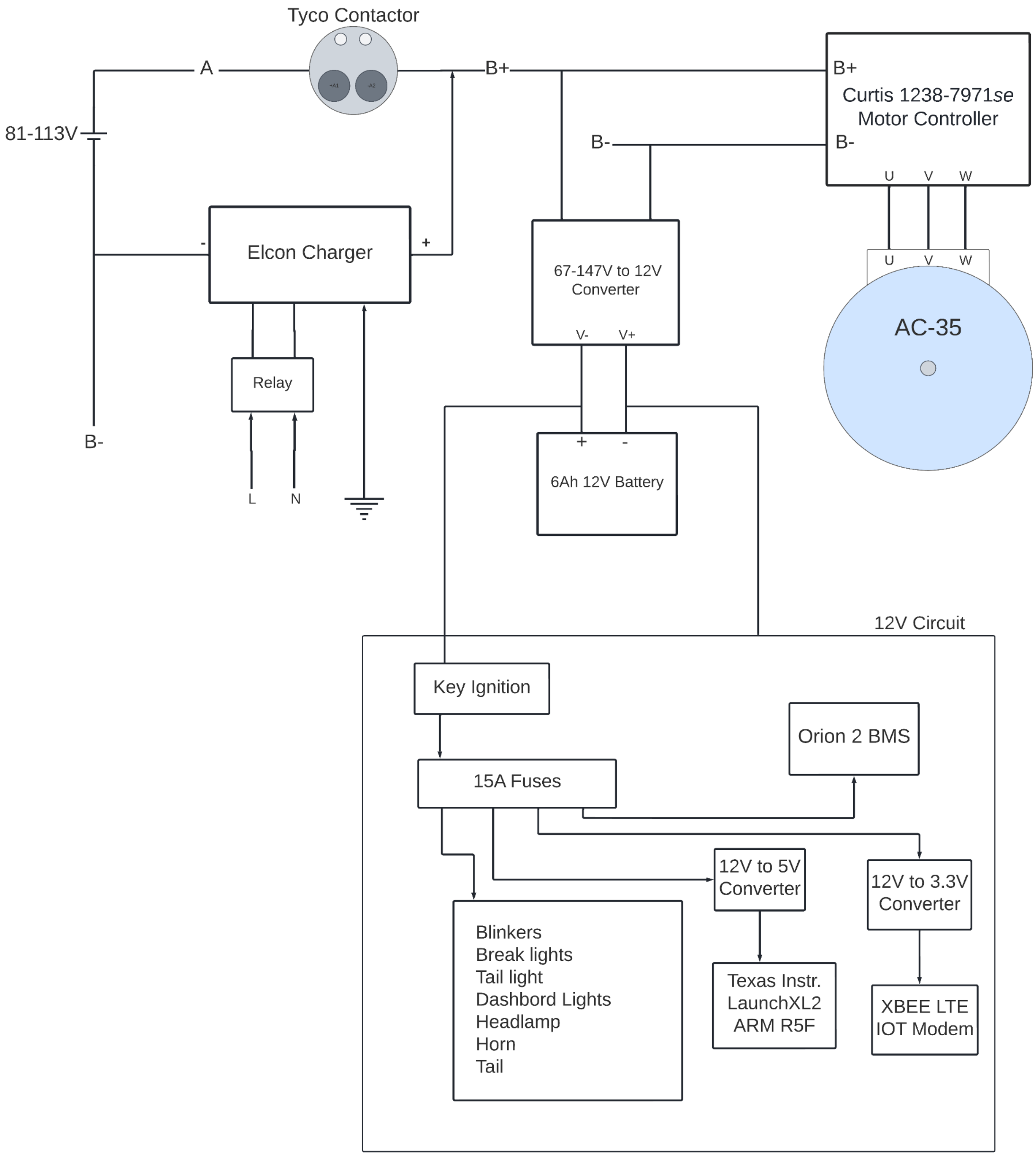


Figure 0: High level schematic of electrical system

Batteries

Selection & Configuration

Our major concerns when selecting which battery cells to use are foremost safety, price then discharge limits and capacity. Originally, we looked for Lithium Iron Phosphate (LiFePo) cells due to their relative stability and long life when compared with more conventional Lithium Manganese, Cobalt or Nickel cells (Lithium ion, henceforth abbreviated: Li^+). However, due to prohibitively high prices for LiFePo cells and their availability in larger prismatic structure, we selected a 7.6kWh pack made up of 3.7V (nominal) 26Ah Li^+ cells which came out of a 2010 Ford Fusion Energi - give or take a few years. These prismatic cells are small enough so we can fit the form factor of the bike well and are rated to a maximal discharge of 10C, which helped us meet our maximum continuous discharge requirements.

To meet the performance requirements of our motor, we need to produce a 96V pack that can handle a maximum discharge rate of about 650A. We configured a 28S3P to give a nominal pack voltage of 103.6V and a maximal discharge of 780A. By initial estimates this configuration gave us ~90 miles of range with the ability to pull 0-60mph in 3.8 seconds.[3] Unfortunately, we found these numbers optimistic once we began testing.

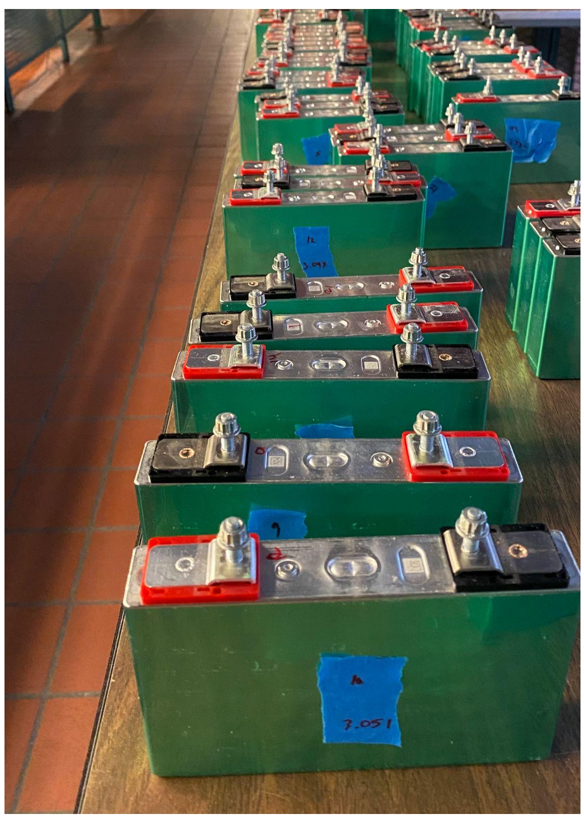


Figure 1: Above, the individual cells outside their packaging

[3] Mihir's Report

Performance & Degradation

After procuring the cells, we ran them through a series of tests to ensure that the age of the pack did not impact their performance too significantly. At a minimum, we needed to ensure the cells were capable of operating in the ballpark of their original specifications, knowing that Li^{+} cells tend to degrade quickly in the early stages of life then level off, with their health not decreasing as rapidly in the later stages of their life. With the cells in their OEM packaging, we ran tests pulsing the batteries at 5, 10, 25, and 50 amps to find an average cell resistance. We concurrently ran tests on 4 individual cells, to be used as representatives for the whole, to find their capacity; figures 2 and 3 below show our setup.

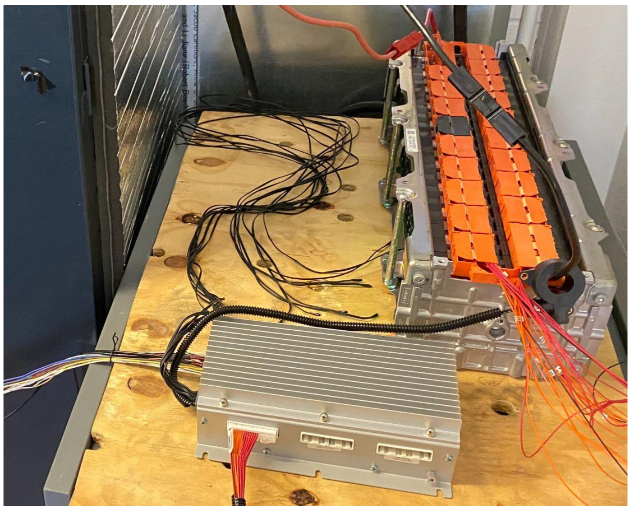


Figure 2: Ford 21 Cell Pack, upper right, connected to Orion 2 BMS, center front, and battery cycler, left

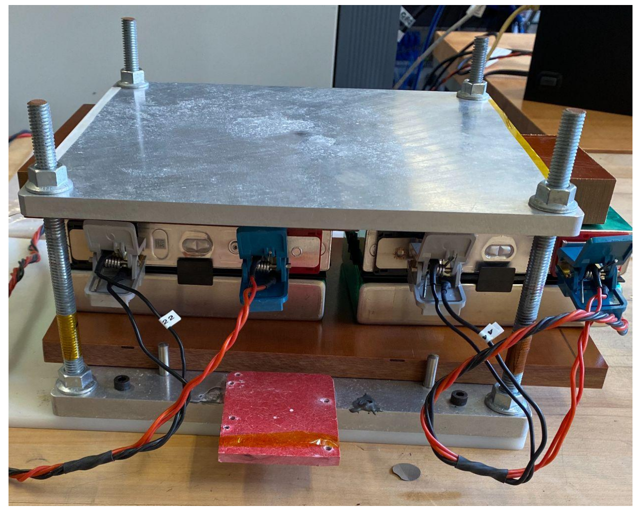


Figure 3: Four cells under compression and individually connected to a multi-channel battery cycle

$$V_{pack} = \varepsilon - I_{pack} \bullet R_{internal} \bullet num.\ cells\ in\ series \qquad \text{Eq. 1}$$

I calculated the average resistance of each of our battery cells to be 4.055mΩ using the experimental setup in figure 2 and solving for $R_{internal}$ in equation 1, where ε is the sum of all the open circuit voltages of the cells as given by the Orion BMS. Figure 4 on the next page shows the average cell resistance during the pulse tests done with 42 cells in series. The first pulse draws from the cells at 10A for one minute, then we rest the cells for two minutes. The second pulse draws 25A, and the rest of the pulses draw 50A. It is worth noting that my calculation for internal resistance is 0.015mΩ less than that of the Orion BMS. I attribute this difference to the length of the cell tap wires, which are 6ft long and whose resistance would be incorporated by the BMS.

It is clear that 4.055mΩ is a *very* high resistance for high discharge cells like these and as such, would lead to severe heating and energy loss during discharge. For example, during peak current draw through a single cell, of ~200 amps, there would be 162 watts lost per cell due to coulomb heating. This type of heat would wreak havoc on our ability to accelerate fast and often.

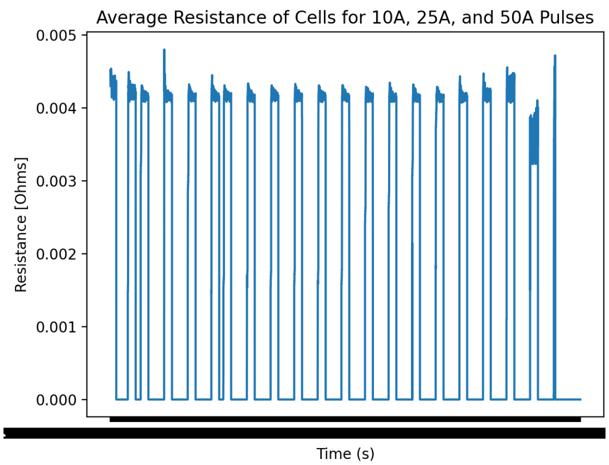


Figure 4: Average internal resistance of a cell in 42S pack during pulse testing

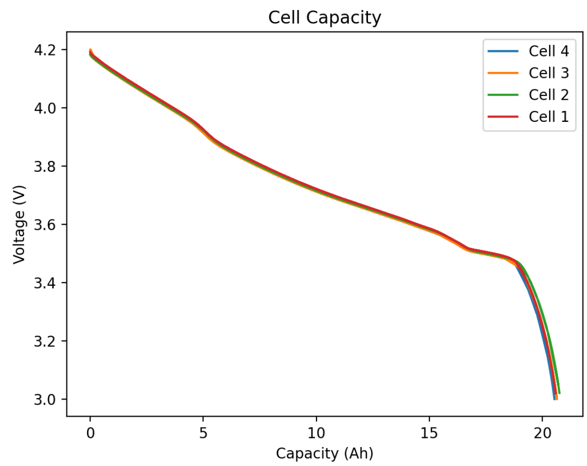


Figure 5: Capacity of four representative cells

Even though this calculation matches that of the Orion 2 BMS, I was still wary of this internal resistance - it being so incredibly burdensome for our performance - and could not accept that our batteries had deteriorated so significantly, thus I calculated the resistance of the four individual cells shown in figure 3. To do so, I used a slightly different, more direct method of finding the internal resistance. In this test setup, there was no BMS connected to the batteries to collect data about open circuit voltage. So I used the instantaneous drop in voltage of the cell when current went from 0 to 5A to determine internal resistance. To explain, when there is no load applied to terminals of a battery, we can read the true open circuit voltage since the terminals are not connected. The instant a load is connected, the battery applies its electromotive force (EMF) to produce current, and that current will cause a voltage over the internal resistance of the battery and thus we will read a drop in voltage across the terminals. For the particular case of the experimental setup shown in figure 3, I found that cell 1 has an internal resistance of 1.87mΩ, much less! Using the momentum of this finding, I did the same calculation on the 42S pack I examined in figure 4 and found the average resistance of a cell to be 1.38mΩ. This resistance seems more reasonable given the type of cell. Using 1.38mΩ the representative resistance, each cell will still lose about 55 watts to coulomb heating when we draw 200A from them.

Further, I calculated the average nominal capacity for the four cells shown in figure 3 to be 20.44 Ah, as shown in figure 5, above. We will use this number to be representative of each of the other cells in the pack. In context of the age of the cells this capacity makes a lot of sense and because of the very small variance in each of the four cells' capacity measurements, it should stand to reason that this is a representative subset. Further, because our cells' maximum current draw is measured in terms of capacity (rated to 10C) this capacity measurement means we can only draw 204A from each cell. Thus by placing 3 in parallel we can achieve a maximum continuous discharge rate of 612A, still enough to power our motor at full load.

During testing we unfortunately destroyed a cell by shooting 100A into it when it was fully charged. Thankfully, our temperature safety checks stopped the test and our compression mechanism prevented any damage from occurring to the lab or surrounding individuals. Nonetheless, we needed to modify our pack configuration to be 27S3P, leaving 2 cells aside and reducing our nominal voltage to 99.9V. All together, this gives us a 6.126 kWh pack, using Mihir's model of energy required to move the bike at 60mph, this should give us ~54 miles of range. Further work will be done to more accurately determine the state of charge and state of health of our cells and pack once we have assembled our 27S3P pack.

Battery Management System

As with every multi-cell battery pack, the battery management system is there to regulate and ensure the safe charging and discharging of the battery cells. Because of size, price, and availability reasons, we purchased the Orion 2 72 split-cell module. This model has a larger form factor than the regular 72 cell module but provides more protection .

Our full pack will have 81 3.7V 20.44Ah cells all held under compression. However, from the perspective of the BMS, since there are 3 cells in parallel with one another, it will treat each group of 3 as its own 3.7V 61.32Ah cell. Thus when populating the cell taps, we have room to spare. The BMS affords us 6 different cell groups. Within each cell group there can be 12 cells. Between cell groups the BMS has greater protections for voltage differences, which are crucial when wires, not busbars, connect cells in series. Our pack is made up of 3 contiguous runs of 15 20.44Ah cells and 3 contiguous runs of 12 20.44Ah cells, as shown in figure 6 below. This means, from the BMS perspective there are 5 61.32 Ah cells in the long cases and 4 61.32 Ah cells in the shorter cases.

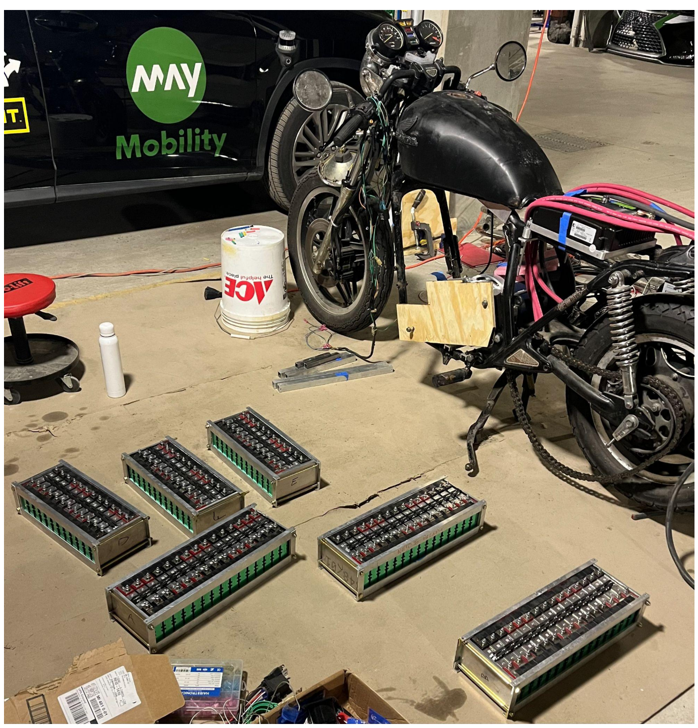


Figure 6: Batteries in their cases next to deconstructed bike

To the BMS, each cell must be wired via the cell tap in order of ascending voltage of the pack. This is because of the internal circuitry of the Orion BMS. The wires running between these cases, although short, still have resistances and must be placed between certain cell groups on the Orion BMS namely at cell 12, 24/37, 48/49, 60/72 and 96. As such, the first case of 5 61.32Ah cells occupy wiretaps 1-12 with the fifth cell connected to all the taps from 5 through 12 so as to tell the BMS there are no more cells occupying those taps. The rest of the cases are similar in configuration.

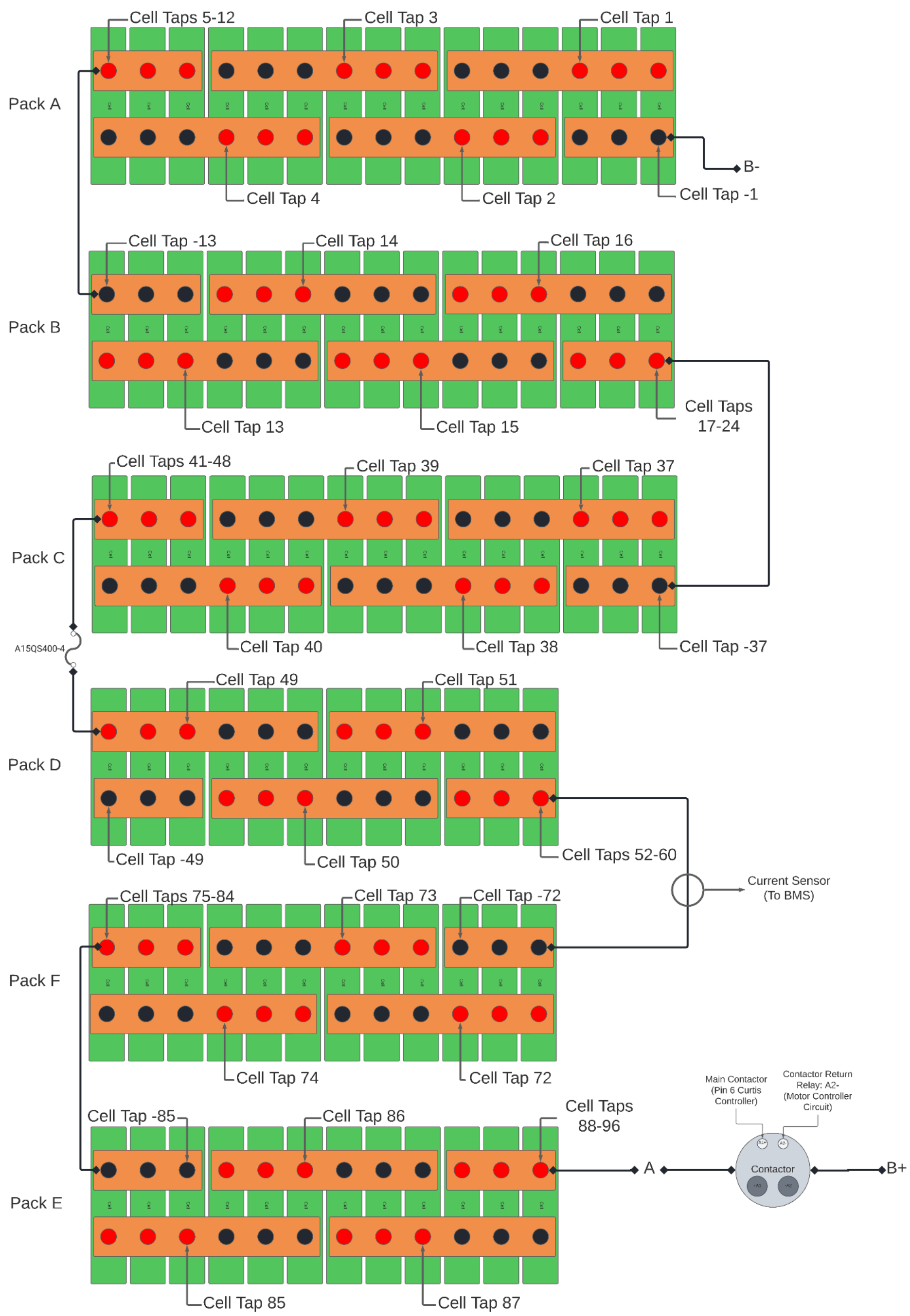


Figure 7: Battery Pack Cell and wire tap configuration w/ fuse, contactor and current sensor placements

A 48V Tyco hermetically sealed contactor is placed in between pack positive, labeled A in the figure on the preceding page, and B+, where the motor controller and the charger are connected. The BMS controls the opening and closing of this contactor via a relay switch connected to a supply pin on the motor controller. More discussion on this relay is in the Motor Controller section. When the BMS senses a fault, like if VOC for a cell is drastically different than its CCV or if a temperature sensor gets too hot, the BMS will open the contactor and disconnect the pack from the bike. When in normal operation, like charging or discharging (riding the bike) the BMS will shut the contactor by pulling its enable pin low, to allow current to flow in and out of the pack.

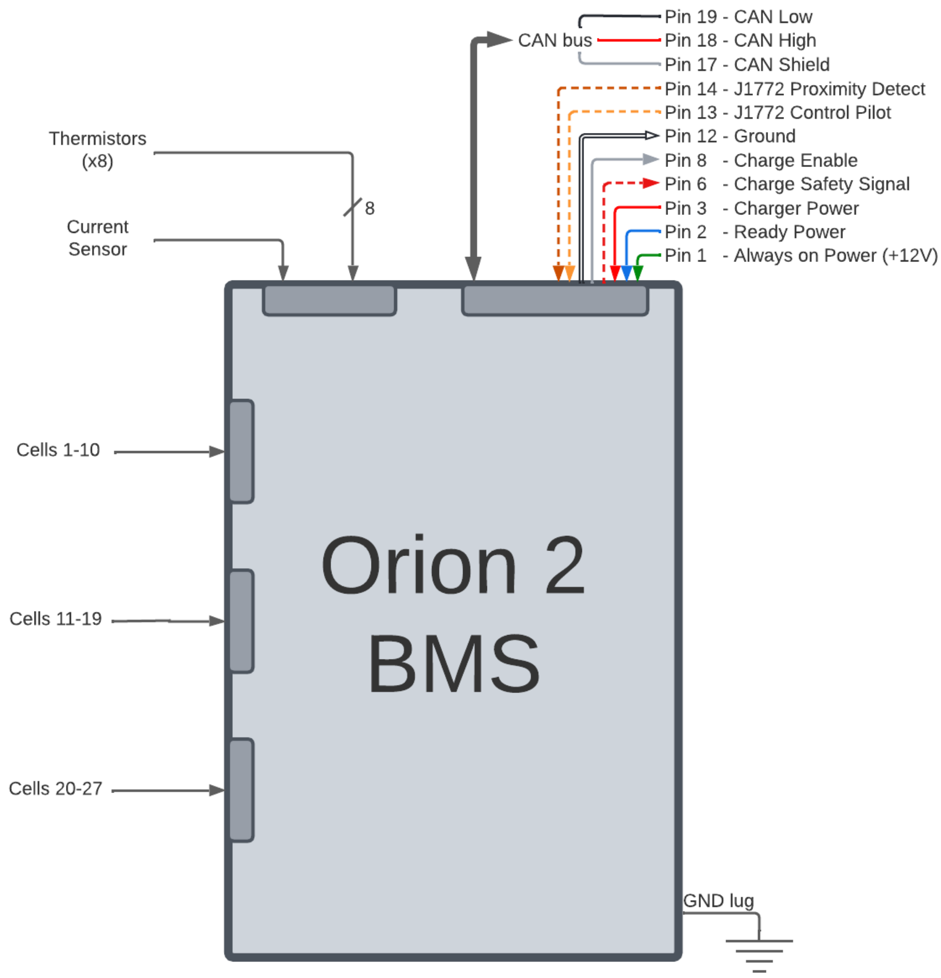


Figure 8: Battery Management Systems Circuit, pin connections

A key function of the BMS is cell balancing. Balancing only takes place during a charge cycle since all the BMS does to balance cells is burn off energy in the form of heat over internal resistors - this is called passive balancing. Cell balancing is key to getting maximal energy storage out of the pack since it brings all the cells, which may charge and discharge at different rates due to age and different rates of degradation, to the same voltage level. This keeps cells in their safe voltage range for longer, prolongs their life, and allows for more cells to be charged & discharged to completion.

The final aspect of BMS operation to which I will devote space here is the state of charge (SOC) and state of health (SOH) estimation. There are many ways to estimate SOC, such as coulomb counting, current integration, etc… whatever method the Orion 2 uses, we are able to set drift parameters to correct for the non-linear charging characteristics of our cells. As of now, I have set a few points to correct for charging a 20.44Ah pack and thus I will save further discussion of SOC and SOH for a future paper once the final battery pack is assembled.

Charging

A charger is required as a means to regulate current from the grid into the battery pack. Many chargers could do the job but one that is automotive grade and integrates rather seamlessly with the Orion 2 BMS is the Elcon 1.8 kW charger with CAN bus. This charger communicates with the BMS directly via the CAN bus and will decrease current flowing into the pack according to the derating parameters which are set in the Orion BMS. Derating is key to the longevity of a battery's life since lessening the current, lessens the overall degradation. At its peak, the charger can provide 16A which should be able to charge our pack to full in just about 4 hours. This number is derived from the Ah capacity of our pack (20.44Ah * 3 cells in parallel / 16 A charge current = 3.83 hours), to which there needs to be time added for derating when the cells reach near their maximum voltage.

J1772 Adapter

A key part of the bike is its compatibility with charging stations around the country. In the United States, the standard EV charger is the J1772, pictured below in figure 9 and 10. Both the Orion 2 BMS and the Elcon Charger support charging via the J1772 interface.

Figure 9: SAE J1772 connector, available around the United States

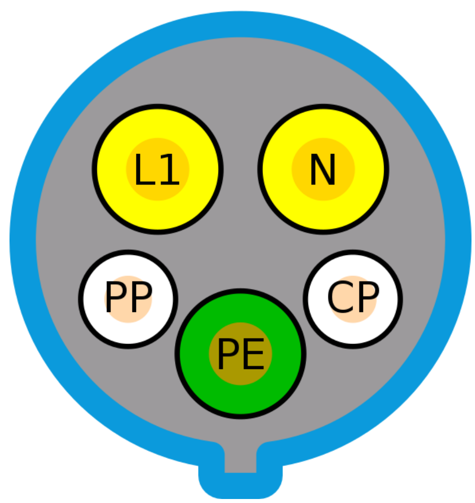


Figure 10: Pinout of SAE J1772 connector, facing into open side of a plug

Pin L1, as depicted in figure Y on the right, provides the single phase AC current used to charge a vehicle at level 1 charging stations (12A @ 120VAC). Pin N is neutral or can be used for another AC signal for level 2 charging, our bike can accept both inputs but will only ever charge at a maximum rate of 16A. The Pin labeled PP is proximity detect which is a signal that is sent to the BMS and will ensure that the user of the bike cannot drive away when it is held high (plugged in). The pin labeled CP is called the control pilot. The wall charger will receive a 1 kHz square wave from the BMS which communicates when to initiate charging and the maximum allowable current to the bike. Lastly, PE stands for protective earth and is the ground pin.

For further safety while charging, I ran both L1 and N through a relay which is controlled by the charge enable pin on the Orion 2 BMS. This pin will go high only when the BMS receives power on its charge power pin and has run through internal checks of the batteries. Once latched closed, the power can then flow through the relay and to the charger to begin charging the pack; this is an analog back up to the CANbus messages between the BMS and charger. Note, a key here was finding a switching converter that is able to accept both Level 1 and Level 2 (120-240VAC) to 12VDC efficiently.

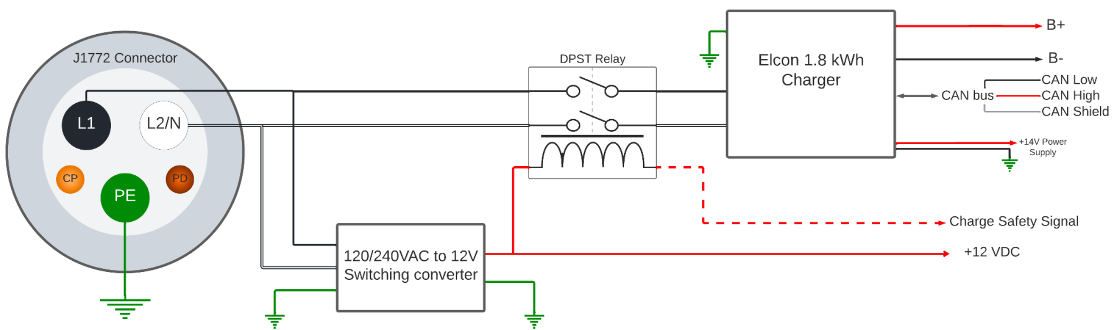


Figure 11: Charging circuit with protective relay

Motor & Motor Controller

Due to our need for speed, we selected an overly powerful HPEVS AC-35 synchronous motor which can produce a maximum of 63 Horsepower at just under 3,000rpm, will max out at 10,000 rpm, and has a whooping 128.92 Ft-Lbs of torque at startup. This motor will be controlled via the Curtis 1238-7971se motor controller. Programming parameters within the motor controller (MC), like braking intensity or acceleration rate, will be done through the programming port connections shown in the diagram on the next page. Detailed discussion of the motor controller's tunability will be saved for a later date, below I will only discuss its integration to our system.

The Curtis controller has a built-in pre-charge circuit, which is activated by the key switch ignition relay in the diagram on the next page. The pre-charge circuit allows its internal circuitry to charge capacitors prior to the application of full battery pack voltage (81-113.4V) across B+ and B-. Without a pre-charge, direct and immediate application of such a high voltage can cause damage to the system. Pre-charging capacitors in this way, controllably raises the voltage across B+ and B- which is necessary for the longevity of the device. The relay disconnects the battery pack and thus does not drain the battery over time while the bike is sitting idle.

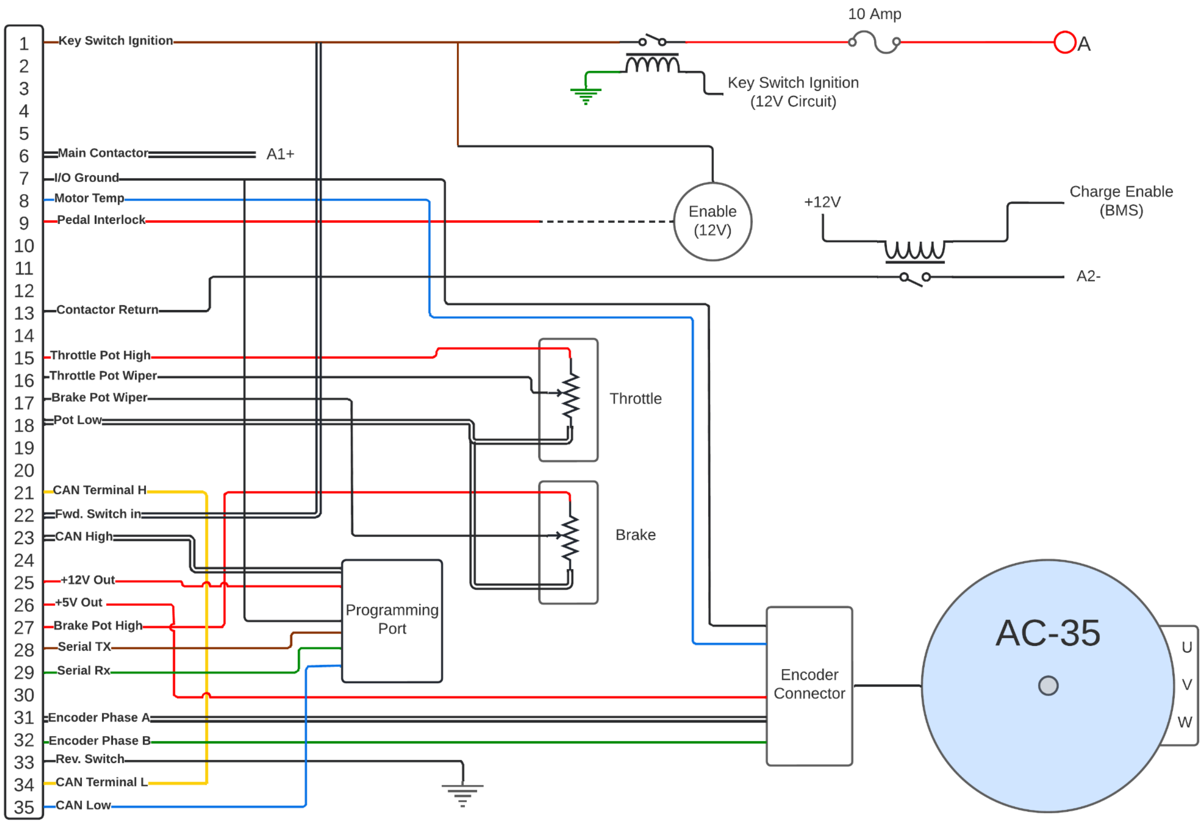


Figure 12: Motor Controller Circuitry

Both the Curtis motor controller and the BMS need to be able to open the contactor in the case of a system fault. The contactor should only be closed when both the BMS and MC say it is safe, so we designed the circuit with a relay in between A2- and the contactor return pin on the MC for the BMS to open the connection. Further, only the MC can supply the 48V needed to operate the contactor, so the A1+ connection must come from the main contactor connection on the MC.

The throttle is a potentiometer that is controlled via the original throttle cable. Further, the old clutch handle and cable will now be connected to the regenerative braking potentiometer. Regenerative braking will probably not provide our system with a significant amount of energy to meaningfully change the range of the bike, but anything is better than nothing considering the age of the cells and weight of the bike. The MC will apply braking power proportional to the amount we pull in the old clutch, now the rear-wheel left-hand brake. We do recognize that this feature could turn out to be dangerous to tune. Too much braking power may cause the wheel to spin in the opposite direction while driving which would cause the bike to skid.

Analog/OEM Circuitry Integration

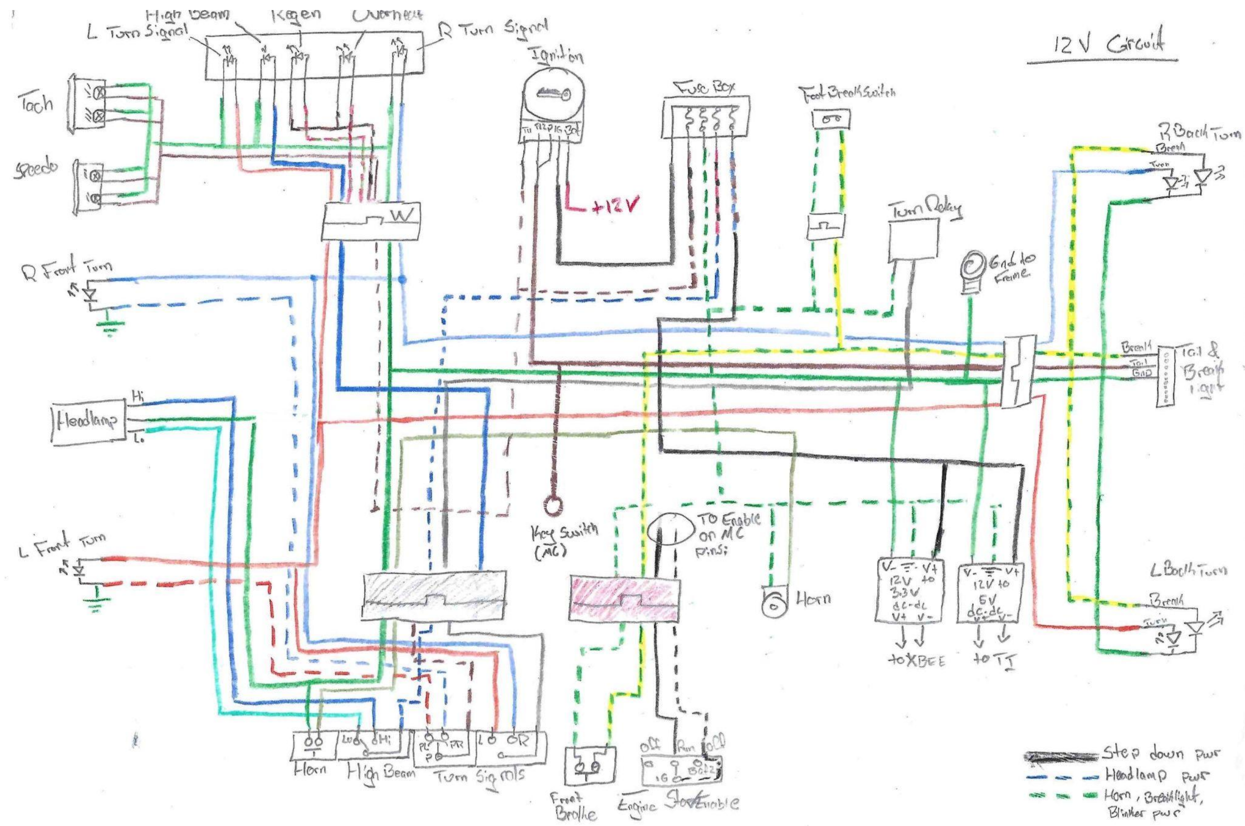


Figure 13: 12V circuit diagram

I was able to use most of Honda's original wire harness to connect the bike's indicators, brake/tail, dashboard, and headlights, however, we switched to LEDs from incandescent bulbs. This necessitated that I retool the blinker circuit since the resistance of LEDs is very small and as such, the grounding and turn relay of the original circuitry did not work. All this means is that I needed to change the turn relay, connect the negative side of the LED blinkers directly to ground and then disconnect the brown dashed wire from power in the black connector on the bottom right.

Further, it was important that we keep the aesthetic of the bike the same so I integrated the enable switch to the bike's original run/stop switch and fused all of the electronics through the original fuse housing. Note here also in the top left connector (labeled W for white) the regen and overheat lights are not connected. This is because those are signals which are going to come from the motor controller and I am unsure of how they should be connected as of yet.

A challenge with this circuit was less the design and more the parsing out what of the original circuitry was not important. Obviously, anything related to the IC engine was taken out but just the mess of untangling wires and finding what they connected to and from was a challenge. One can find the original wiring diagram for our 1981 Honda CB750C at the link in the footnote.[4]

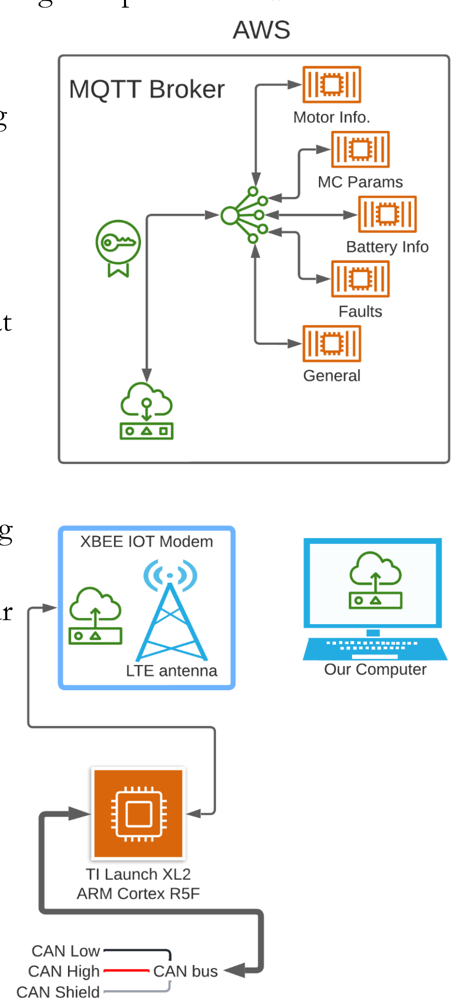
4 Wiring_Diagram_CB750C_81_82.pdf

Cloud Connectivity

Here is where we hope to separate ourselves from the pack of EV conversions by connecting the CAN bus of our vehicle to cloud. No implementation has been done here yet, below I will only outline our software /hardware connectivity plan.

In terms of hardware there are two main components, a Texas Instruments LaunchXL2 board with a real-time ARM Cortex R5F core, built for automotive grade applications. This board will connect directly to the CAN bus to snoop/send messages. The TI board also connects to an XBEE IOT LTE modem which will connect to multiple LTE networks around the country. This connection will allow us to communicate via MQTT to a broker running on an AWS. Here the hardware on the bike will send information about the bike (SOC, faults, range, etc…) so that we can connect a web dashboard to AWS and view our data.

In addition to reading data from the bike, MQTT is a bidirectional communication protocol which will allow us to send data to the bike. We may define a driving profile like eco or sport mode that we would want to upload to the bike's motor controller, and as such we can package the parameters from our laptop and send it off for the bike to read. Figure 14 gives a system level overview of how our devices are connected. Once the XBEE receives the message over LTE, it will send its contents to the TI for repackaging. The TI board will package the parameters we send it for the specific CAN bus messages required by the motor controller or BMS. It will then put those messages on CAN. For example, we may want to increase the intensity of our regenerative braking by 0.5% overall. Conventionally, this would require actually plugging into the motor controller's programming port, but with our software/hardware interface and cloud infrastructure, we won't need to plug anything in. The TI board will act as the original CAN bus adapter would have and allow us to change that parameter. Further, this design helps us change multiple parameters at once, like would be required to define an eco or sport mode. These parameters completely change how the bike drives and its energy consumption.



A main concern of ours is the vulnerability of connecting our CAN bus to the internet. All of the communications on and off the bike will be encrypted. Further, in order to connect to our AWS instance it will have to be from a known IP address, meaning only access from one of our computers can publish something for our bike to read. Second, the XBEE will statically subscribe to channels for MQTT meaning only whatever is published there will ever be read by our bike. Third, the TI will never be able to write to the CAN bus when the bike is in motion, thus reducing the harm of the worst case scenario. This work will be saved until the critical systems of the bike, all previous sections of this report, have been completed.

Progress & Next Steps

As of Monday May 2, 2022, we have settled on a mounting strategy and configuration of batteries. We now need to manufacture 2 large plates to finalize their mounting. I need to finish balancing the batteries in every pack so that we can place the busbars on them. I have yet to design covers for the cell terminals and bus bars. I also need to reconfigure the cell taps to connect each pack to the BMS. Further, I am fairly sure, I will need to tweak the SOC / SOH calculations once we place 3 cells in parallel. Once completed I can assemble the whole system and power on the motor controller for the first time - that will open its own bag of worms.

The motor controller will require that I rent the OEM CAN programmer for initial set up so I am delaying this as it is expensive. Once I have the programmer device, I plan to connect it to a logic analyzer so that I can reverse engineer the message structure using the TI board.

Acknowledgements

I would like to deeply thank Professor Jason Siegel of the University of Michigan for his continued support and guidance throughout this project. I would also like to thank Mihir Juvvadi and Luke Wing for being invaluable team members. Lastly, thank you to the 4M community for letting us use space in their garage.