# Formula Hybrid: Final Report


**Parker Imlay, Musodiq Ogunlowo, Neal Smith, William Kuang, Bridget Taylor**

**Senior Design 2019-2020**

**Professor Schafer**

**5/7/2020**

# Table of Contents

# 1. Introduction

The Notre Dame Formula Hybrid Team is in the process of developing a race-ready hybrid electric car for competition in SAE events. Last year, an electrical engineering senior design team created a prototype electrical system for the car. Our project was to continue the development of this legacy project by implementing more-refined systems and increasing the robustness of the project as a whole. The team met unanticipated challenges with bringing the car back to the working state illustrated in last year's final report. Although we wanted to add new features to the car, we realized that there was a greater need for testing procedures of the existing control system and documentation for the subsystems of the car. A preliminary goal, then, was to develop a more in-depth documentation of the entire system to allow future teams and participants of ND Hybrid to easily understand the underlying function of the subsystems. In the early stages of work, this documentation was developed in addition to refinements in the system. After the cancellation of in-person classes, documentation became the primary project goal. This document will describe the Hybrid Car system as a whole as well as illustrate both the original team goals and their final outcomes.

## 1.1 Overview

The car is a series hybrid powered by a bank of series-connected ultracapacitors in series with a small internal combustion engine (ICE). A generator converts the mechanical energy from the motor to electrical energy to charge the capacitors, which in turn power the hub motors on each of the front wheels. In other words, the capacitors serve as an energy buffer between the generator and the hub motors. This combination of ICE, ultracapacitors, and motors is the high voltage system. There is also a low voltage system that monitors and controls the high voltage system based on user inputs.

In sum, the electrical system is in charge of controlling the throttle on the ICE, monitoring the ultracapacitors' voltages and temperature, taking driver inputs and converting them to motor outputs, and providing analytics.

These functions can be broken into five major subsystems:

- Driver inputs
- System status interface
- Mechanical outputs (motor controllers/generator/motors)
- Engine feedback loop
- Accumulator management system (AMS)

In addition to these subsystems, ultracapacitor charging and discharging methods were developed to change the capacitor bank state of charge (SOC) independently of the engine.

While not a subsystem, this ability is essential for independent testing of other subsystems (such as the AMS.) Therefore, it will also be included as a peripheral subsystem:

- Independent capacitor charging and discharging

The existing embedded system on the car serves three main purposes. First, it controls the charge of the accumulator ultracapacitors and the transition between states in the drive system. Second, it records the temperature and voltages of the ultracapacitors and shuts down the high voltage system when these become out of range. Third, it relays the status information from the motor and generator controllers as well as the AMS with a wireless RF transceiver into the terminal of a PC.

## 1.2 Initial Problems

Upon completion of the startup sequence outlined in the 2018-2019 senior design team's Final Report, nonsensical errors appeared on the status LCD such as incorrect capacitor SOC verified by a multimeter and temperature readings that were in the negative range. As a safety precaution, the system is designed to open relays on the high voltage system when values outside a specified range are read. Therefore, the high voltage system could not be tested until the source of the nonsensical readings was discovered.

Due to the cancellation of in-person classes, this error was never completely fixed. However, the new documentation on the AMS subsystem should allow for rapid debugging of this error. Future teams should know that the system currently will not function as described in the 2018-2019 team's Final Document until this error has been resolved. This team is not aware of any other bugs that will prevent the system from operating.

**Table 1.** Summary of Problems, Challenges and Solutions

| Problem | Challenges | Proposed Solution | Implemented Solution |
|---|---|---|---|
| 1. Car stuck in wait state with given error of battery charge at 142% | Needed a method to change the voltages on the capacitors without using the engine or generator since these would not activate | Devise a method for independent charging and discharging of the capacitors<br><br>Diagnose error messages from the AMS PCB | Developed method for independently charging and discharging capacitors<br><br>Created extensive documentation on AMS system |
| 2. Implement an improved user interface to display error messages from the Kelly Controllers | Baud rate was unknown and code used incorrect settings for baud rate configuration | Reconfigure baud rate registers for the wireless RF transceiver<br><br>Develop a GUI to display error | Designed a GUI that allowed data to be displayed to an off-track computer using the RF transceiver |

| | | | |
|---|---|---|---|
| | | messages from the Kelly Controllers | |
| 3. Implement hydraulic braking | The Hybrid team had not installed the appropriate hardware such as wheel rotors that would make implementation of a hydraulic system feasible | Develop working diagrams for how hydraulic braking would function in the hybrid car if equipment was provided | Developed diagrams and general understanding on implementation of hydraulic braking for future reference |
| 4. Differentiate between the left and right Kelly motor controllers | There was no physical way to change the source address of the Kelly Controllers | Implement the two CAN controller modules present on the PIC32MX795 chip with one controller module for each motor controller | Developed extensive documentation of testing procedures and overview of the CAN network protocol |
| 5. Improve voltage and temperature monitoring in the AMS subsystem by monitoring each of the 60 ultracapacitors (100% of all ultracapacitors) | Difficulty in finding resistance-temperature/ equation curve for thermistors, no documentation, and absence of extra pins on the LTC6812-1 boards | Addition of thermistors between every pair of ultracapacitors by either direct connection or use of 3 extra LTC6812-1 boards | Documentation on finding the resistance -temperature curve/equation for the thermistor and plans on how to connect them to the ultracapacitors |

# 2. Definitions and Useful Acronyms

**AIR** - Accumulator Isolation Relay - must be held closed by system

**CAN** - Controller Area Network - broadcast messaging protocol for data sent between the microprocessor and the generator and motor controllers

**Motherboard** - central data processing unit of the Hybrid Car using the PIC32MX795F512H

**GLVS** - Grounded low-voltage system - Powers the LCD, motherboard, and separate AMS monitoring PCB board, and starts the controllers

**High Voltage System (HVS)** - Activated when the accumulators provide power to the motor controllers and must be galvanically isolated from the GLVS

**ICE** - Internal Combustion Engine - 250cc engine from Kawasaki Ninja motorcycle

**Drive System** - enables forward, reverse, and neutral driving modes

**Charging System** - Provides power from the ICE to the AMS

# 3. Detailed System Requirements

The Notre Dame Hybrid Teams plans to use the car in SAE competitions. Therefore, the electrical system must meet the requirements laid out in the Formula Hybrid Rules. Since this is a legacy system and because the Notre Dame Formula Hybrid Team was not planning to have a race-ready car this year, the Team did not need to focus on the official requirements. However, broad design decisions were made by the 2018-2019 Team to meet those requirements. Their summary of relevant rules will be copied verbatim in section 3.1 to aid in understanding the design decisions of the legacy system. Relevant changes in the 2020 rules will also be noted.

Beyond the official SAE rules, the Notre Dame Formula Hybrid Team required a method for charging and discharging the capacitors that is not dependent on the ICE or the generator. They also required more comprehensive documentation of the existing system. The other improvements that the Team chose to address are as follows:

- Differentiation of CAN messages between the left and right motor controllers to allow for improved error diagnostics and enhancement of torque vectoring
- Updated PID controller constants to produce better control response and variable target RPM to allow for different operating regimes based on capacitor SOC
- Implementation of active or passive cell balancing (method chosen based on results of testing)
- Addition of temperature measurement capabilities to monitor all ultracapacitor temperatures
- Creation of GUI to make transmitted system diagnostic information more readable
- Rebuild of system hardware with a focus on durability for the race environment
- Possibilities of implementation of hydraulic braking

Due to the cancellation of in-person classes, the Team was not able to accomplish some of these design requirements.

## 3.1 SAE Requirements for Race-Ready Vehicle

[Copied from 2018-2019 Formula Hybrid Final Report]

1) AMS
    a) Each accumulator must be monitored by an accumulator management system (AMS) whenever the tractive system is active or the accumulator is connected to a charger.
    b) The AMS must monitor all critical voltages and temperatures in the accumulator as well the integrity of all its voltage and temperature inputs. If an out-of-range or a malfunction is detected, it must shut down the electrical systems, open the AIRs and shut down the I.C. drive system within 60 seconds.
    c) The tractive system must remain disabled until manually reset by a person other than the driver. It must not be possible for the driver to re-activate the tractive system from within the car in case of an AMS fault.
    d) The AMS must continuously measure cell voltages in order to keep those voltages inside the allowed minimum and maximums stated in the cell data sheet (See Table 10). NOTE: If individual cells are directly connected in parallel, only one voltage measurement is required for that group.

| Chemistry | Maximum Number of Cells per Voltage Measurement |
|---|---|
| Lithium based | 6 |
| NiMh | 6 |
| Pb Acid | 1 |

**Table 10.** AMS Voltage Monitoring

    e) The AMS must monitor the temperature of the minimum number of cells in the accumulator as specified in Table 11 below. The monitored cells must be equally distributed over the accumulator container(s). NOTE: It is recommended to monitor the temperature of all cells.

| Chemistry | Percent of Cells Monitored |
|---|---|
| Li-Ion | 30% |
| NiMh | 10% |
| Pb Acid | 5% |

| | |
|---|---|
| Ultracapacitor | 10% |

**Table 11.** AMS Temperature Monitoring

f) All voltage sense wires to the AMS must be protected by fuses or resistors (located as close as possible to the energy source) so that they cannot exceed their current carrying capacity in the event of a short circuit
g) Any GLV connection to the AMS must be galvanically isolated from the TSV. This isolation must be documented in the ESF.

2) Team-designed AMS Board
   a) Teams may design and build their own Accumulator Management Systems. However, microprocessor-based accumulator management systems are subject to the following restrictions:
      i) The processor must be dedicated to the AMS function only. However it may communicate with other systems through shared peripherals or other physical links. In our case, this allows isoSPI communication.
      ii) The AMS circuit board must include a watchdog timer. It is strongly recommended that teams include the ability to test the watchdog function in their designs.
3) Accumulator - Isolation Relays
   a) At least two isolation relays (AIRs) must be installed in every accumulator container, or in the accumulator section of a segmented container (See EV2.3.4 Note 2) such that no TS voltage will be present outside the accumulator or accumulator section when the TS is shut down.
   b) The accumulator isolation relays must be of a normally open (N.O.) type which are held in the closed position by the current flowing through the shutdown loop (EV7.1). When this flow of current is interrupted, the AIRs must disconnect both poles of the accumulator such that no TS voltage is present outside of the accumulator container(s).
   c) When the AIRs are opened, the voltage in the tractive system must drop to under 30 VDC (or 25 VAC RMS) in less than five seconds.
   d) The AIR contacts must be protected by Pre-Charge and Discharge circuitry, See EV2.10. If the AIR coils are not equipped with transient suppression by the manufacturer then
   e) Transient suppressors must be added in parallel with the AIR coils. AIRs containing mercury are not permitted.
4) Precharge
   a) The AIR contacts must be protected by a circuit that will pre-charge the intermediate circuit to at least 90% of the rated accumulator voltage before completing the intermediate circuit by closing the second AIR.

b) The pre-charge circuit must be disabled if the shutdown circuit is deactivated; see EV7.1. i.e. the pre-charge circuit must not be able to pre-charge the system if the shutdown circuit is open.
c) It is allowed to pre-charge the intermediate circuit for a conservatively calculated time before closing the second AIR. Monitoring the intermediate circuit voltage is not required.
d) The pre-charge circuit must operate regardless of the sequence of operations used to energize the vehicle, including, for example, restarting after being automatically shut down by a safety circuit.

5) Discharge
    a) If a discharge circuit is needed to meet the requirements of EV2.8.3, it must be designed to handle the maximum discharge current for at least 15 seconds. The calculations determining the component values must be part of the ESF.
    b) The discharge circuit must be fail-safe. i.e. wired in a way that it is always active whenever the shutdown circuit is open or de-energized.
    c) For always-on discharge circuits and other circuits that dissipate significant power for extended time periods, calculations of the maximum operating temperature of the power dissipating components (e.g., resistors) must be included in the ESF.
6) Motor Controllers
    a) The tractive system motor(s) must be connected to the accumulator through a motor controller. Bypassing the control system and connecting the tractive system accumulator directly to the motor(s) is prohibited.
    b) The accelerator control must be a right-foot-operated foot pedal.
    c) The foot pedal must return to its original, rearward position when released. The foot pedal must have positive stops at both ends of its travel, preventing its sensors from being damaged or overstressed.
    d) All acceleration control signals (between the accelerator pedal and the motor controller) must have error checking.
    e) For analog acceleration control signals, this error checking must detect open circuit, short to ground and short to sensor power
    f) For digital acceleration control signals, this error checking must detect a loss of communication.
    g) An error in the acceleration control signal must shut down the torque production in less than one (1) second when a fault is detected. NOTE: If these capabilities are built into the motor controller, then no additional error-checking circuitry is required.
    h) The accelerator signal limit shutoff may be tested during electrical tech inspection by replicating any of the fault conditions listed in EV3.5.4
    i) TS circuitry, even at low voltage levels, is not allowed in the cockpit. All motor controller inputs present in the cockpit must be galvanically isolated. This includes accelerator input, forward/reverse, on/off switches etc.

j) Motor controller inputs that are galvanically isolated from the TSV may be run throughout the vehicle, but must be positively bonded to GLV ground.
k) TS drive motors must spin freely when the TS system is in deactivated state, and when transitioned to a deactivated state.

# 4. Detailed project description

## 4.1 System theory of operation

The system comprises a variety of sensors to monitor the vehicle and take user inputs as well as outputs to control the system and display critical information. The system state machine is implemented on the motherboard, which uses a pic32795. The motherboard handles the majority of the information processing. There is also an Accumulator Dedicated Processor (ADP) that controls the accumulator system. The PCB for the ADB is an exact copy of the motherboard and communicates with the motherboard as an isoSPI slave.

The following are the inputs tied directly to the motherboard:

- Accelerator pedal position (analog)
- Brake pedal position (analog)
- Steering wheel position (SPI)
- Engine RPM (digital high/low)
- Switch states  (i.e. enable, ignition) (digital)
- Motor controller messages (CAN)
- Error messages from ADP (isoSPI)

The following are the inputs to the ADP:

- Capacitor voltages (isoSPI)
- Capacitor temperatures (isoSPI) (plan developed for implementation)

The following are the primary outputs:

- LCD display
- RF data transmission
- Hub motor torque for front wheels
- ICE throttle control through servo motor
- AIR control

## 4.2 System Block Diagram

The overall system block diagram showing how it is divided into subsystems, and the interfaces between the subsystems are shown in Figure 4.2a below. Additionally, the system drive diagram is shown in Figure 4.2b below.

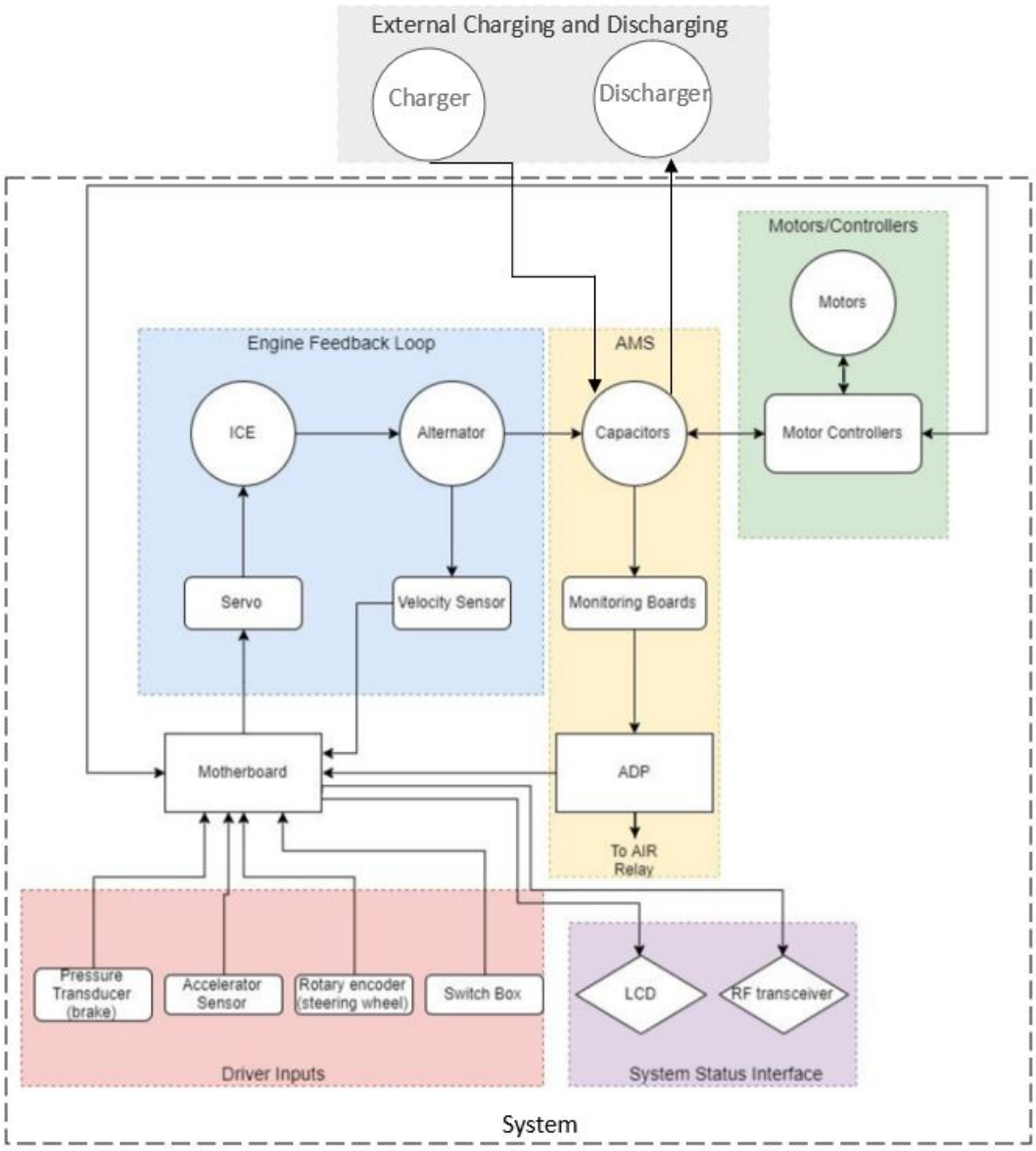



**Figure 4.2a.** Overall System Diagram

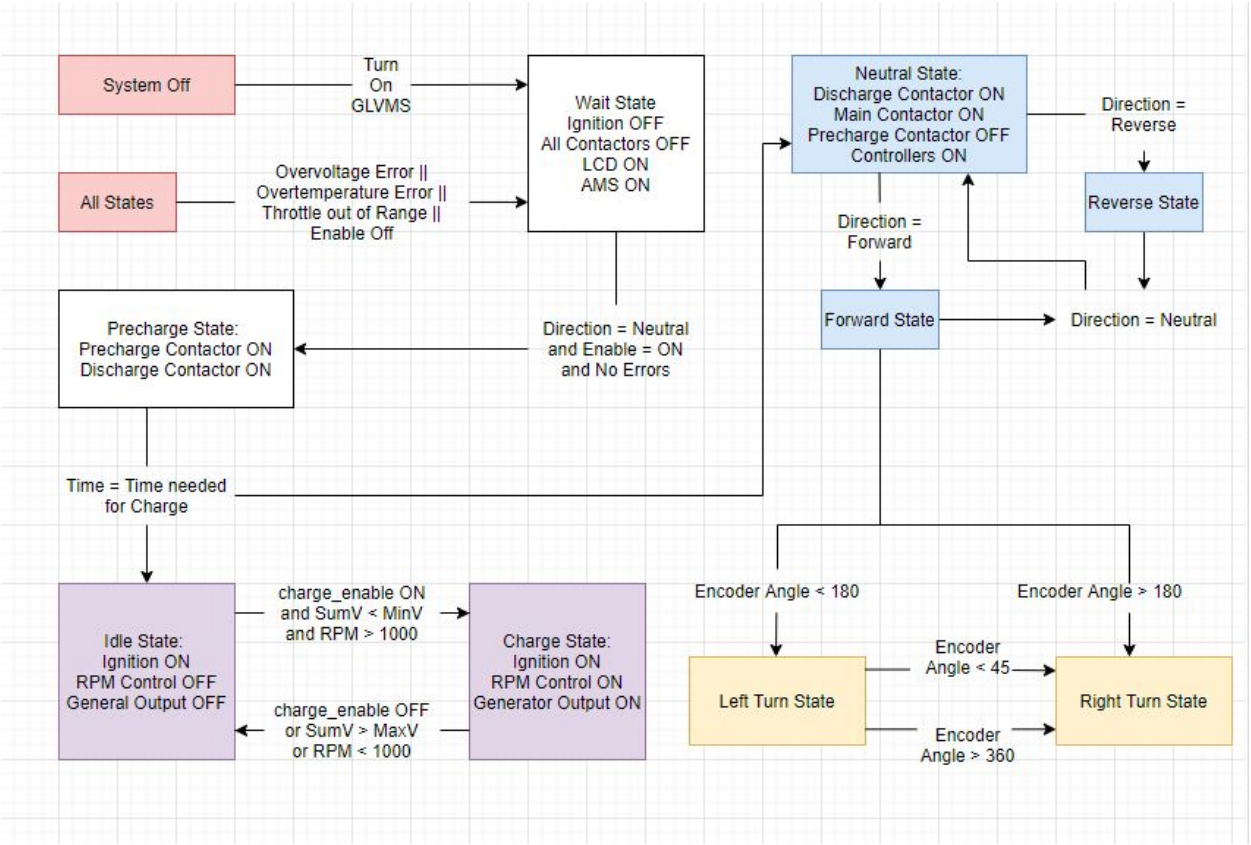



**Figure 4.2b.** System Drive Diagram

## 4.3 Driver Inputs

### 4.3.1 Overview of Subsystem

Driver inputs are the direct link between the driver and the overall system within the Hybrid Car. These inputs are the steering wheel, pedals, and switch box. These inputs connect to the system motherboard as shown in Figure 4.2, in which the switch box is mainly responsible for powering the system and switching states.

The pedals involved are a throttle pedal and a brake pedal. The pedals are connected to an op-amp and then to the motherboard so that the correct order of voltage is maintained. The brake pedal implemented by the previous team uses an accelerator pedal.

### 4.3.2 Original Goals for Subsystem

Our team originally decided to include two major improvements within driver systems.

First, error margins would be added to increase reliability of the minimum and maximum acceptable values of the throttle. Currently, a lot of false errors exist such as variations in the supply voltage or electrical noise triggering false fault conditions.

Second, a pressure transducer, along with hydraulic brakes will be implemented along with the rest of the system as a replacement to the current pedal. Assuming a seamless integration, this would increase safety such that braking is reliant on mechanical factors instead of electrical factors. For this to occur, the signal from the transducer would also need a gain factor since the pressure transducer outputs its maximum signal at a pressure far higher than normal driving conditions. This gain would be obtained from tuning according to physical testing.

### 4.3.3 Progress Made in Subsystem

For improvements involving error margins, due to a lot of more pertinent errors within the system/system integration, this goal will be postponed as a future development.

For the inclusion of a pressure transducer as well as hydraulic brakes, after discussing with the current Formula Hybrid club presidents, many mechanical installments still need to be made before implementation and transition into hydraulic braking is possible. As such, the team developed a preliminary system drive diagram (Figure 4.3a) and an overall system diagram (Figure 4.3b) for implementation of hydraulic brakes. Figure 4.3c shows general connections on how hydraulic braking implementation would occur.

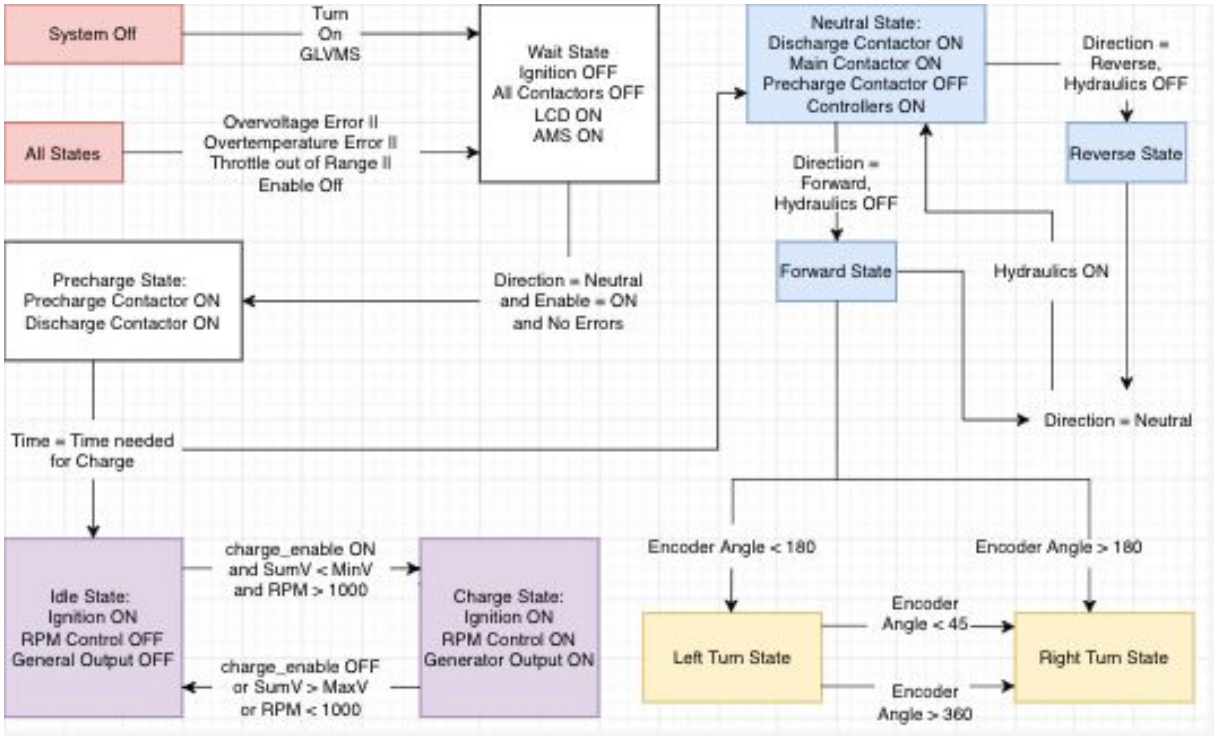



**Figure 4.3a.** Drive System Diagram with Hydraulic Brakes

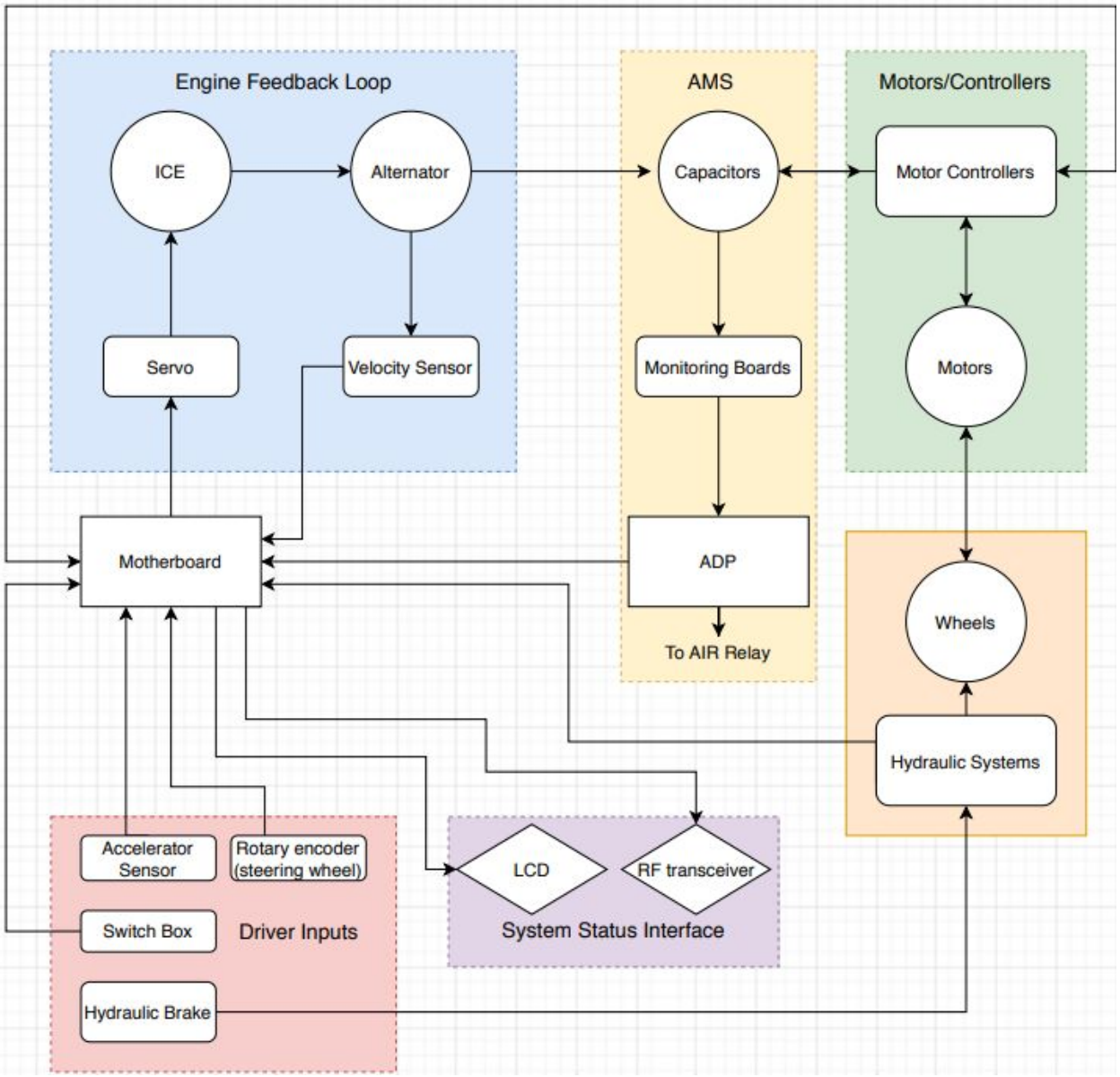



**Figure 4.3b.** Overall System Diagram with Hydraulic Brakes

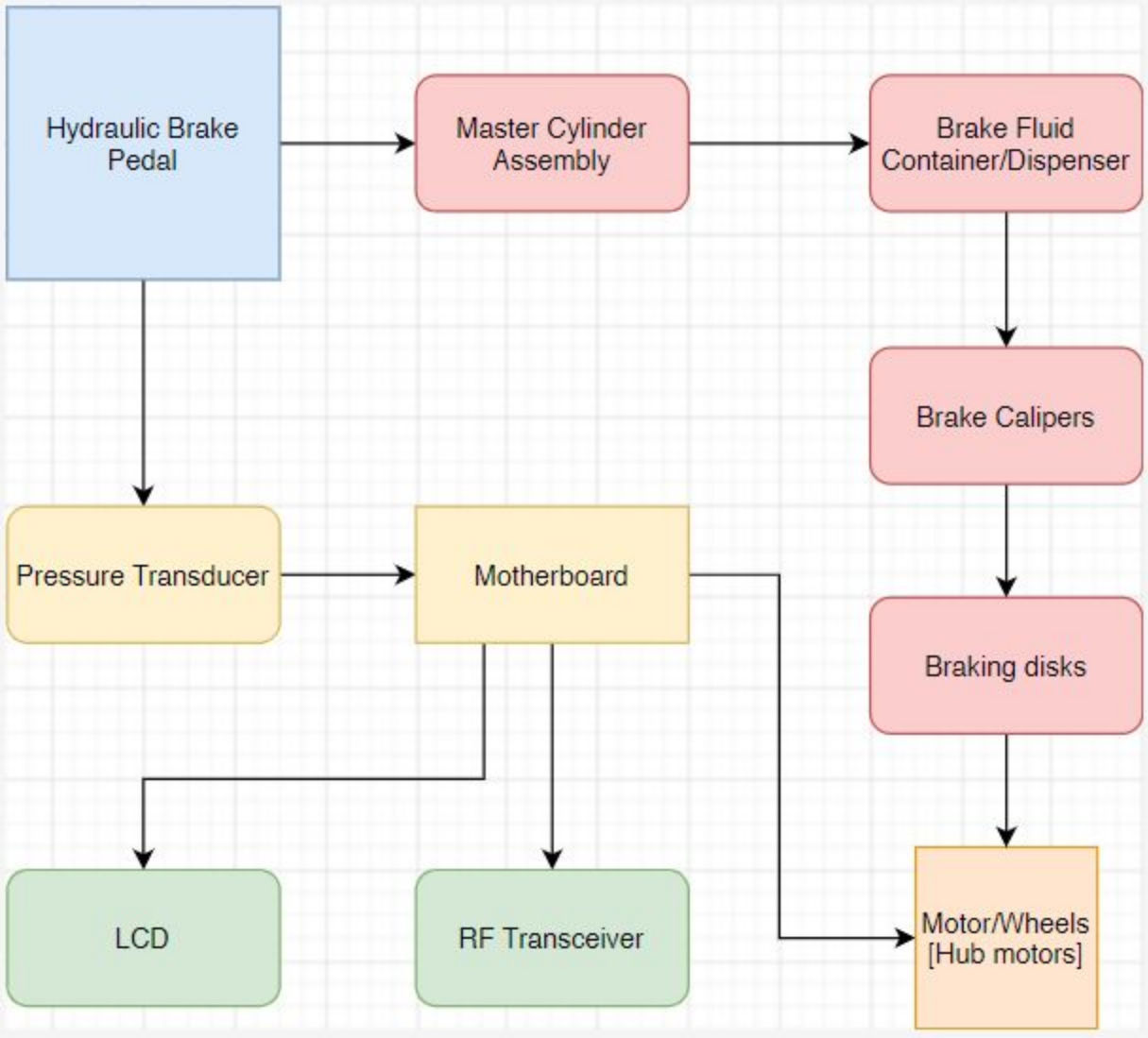



**Figure 4.3c.** Hydraulic Brake Connections

## 4.4 System Status Interface

The system status interface is primarily responsible for communicating the updates on relevant information such as fuel level, ultracapacitor charge, vehicle speed, engine RPM to the driver and also the off-track team. The updates to the driver are done using an LCD display inside the car while the off-track team receives data from an RF transmitter communicating to an RF receiver writing data using UART. A Graphical User Interface was developed on MATLAB to allow the off-track team to decipher the data in an intelligible manner.

### 4.4.1 LCD [copied from 2018-2019 Formula Hybrid Final Report]

The LCD screen is used to communicate information on the state of the vehicle to the driver on board. The team picked this specific LCD because it is equipped with a controller that makes it easy to communicate with through SPI. The 7-inch size was also ideal for the team's purposes to output a large amount of data that is easy for the driver to see. Additionally, the available libraries are intuitive, making it simple to create graphics such as gauges and progress bars. It supports various forms of RGB interfacing, which makes it flexible as well.

The display takes in constantly changing values of the vehicle speed and power and shows them in two gauges that update accordingly. The power gauge starts in the middle and moves 19 clockwise when the net power is positive (draining the acumulator) and counterclockwise when the net power is negative (regenerative braking, adding power to the accumulator). The current charging state of the capacitors is displayed in a progress bar in percentage value; the RPM is displayed the same way. There is also a list of diagnostics including the accumulator temperature, steering wheel angle, accumulator voltage, and charge state. The top right displays a 1 if the AMS is Lastly, there is a section of the screen dedicated to displaying any errors that the monitoring system detects. In order to display the errors, the LCD program interprets the CANbus messages it receives. Every type of error is allocated to a bit, therefore the program performs bit masking to identify which errors have been set and then display the corresponding text error on the screen

### 4.4.2 RF Transceiver

The transceiver circuit transmits relevant information about the state of the vehicle to an off-track computer via RF. The legacy team selected a transceiver kit because it communicates through UART, a communication protocol they were familiar with. The range is also sufficient enough for data to be monitored at more than a kilometer away. This should ideally allow the off-track team to monitor the status of the vehicle at whatever time during the competition. mmc

### 4.4.3 Original Goals for Subsystem

Our team decided to improve the system status interface by making the information presented in a more intelligible manner and also create a better way for off-track monitoring to occur while allowing for data-logging. In essence, we planned to improve the LCD display by having better color contrast, increased brightness settings and glare reducing shield to improve visibility. There was also a desire to include more data to be displayed on the LCD such as motor temperature and controller temperature. With regards to the RF system, we planned to create a GUI that allows data to be displayed in real time and also an auto-save feature to allow for information processing at later times.

### 4.4.4 Progress Made in Subsystem

The team chose to prioritise setting up the GUI for the RF system in order to allow for better off-track monitoring. This was achieved by designing a GUI on MATLAB that allows data to be received in a manner that could be well understood instead of a comma separated list. Currently, the data does not update in real time and the data logging feature does not exist yet. The changes to the LCD are still required but were considered not urgent updates.

## 4.5 Mechanical Outputs (Motor and Generator Controllers)

### 4.5.1 Overview of subsystem

In the mechanical outputs subsystem, the generator transfers power from the ICE to the ultracapacitors. The motor controllers monitor and regulate the motor rpm, current, throttle, and temperature, as well as the temperature of the controller itself. The controllers communicate with the motherboard via CAN messaging.

CAN Bus protocol reduces the amount of wiring necessary between I/O devices on a vehicle. Each I/O device is called a node on the CAN network. When a node has information to share with other nodes, it transmits and broadcasts a CAN message to the CAN bus. All other nodes have the opportunity to accept or ignore the broadcasted message. Given that there is no master in CAN protocol, each message is encoded with a priority level to enable critical messages from brakes to have priority over messages about air conditioning of the vehicle. The arbitration process will determine which node can transmit its message first. All nodes can ignore messages that do not apply to them with buffers and filtering set up in the microcontroller. Overall, the low-cost, centralized processing, and robustness to electromagnetic interference makes CAN protocol so appealing for car manufacturers.

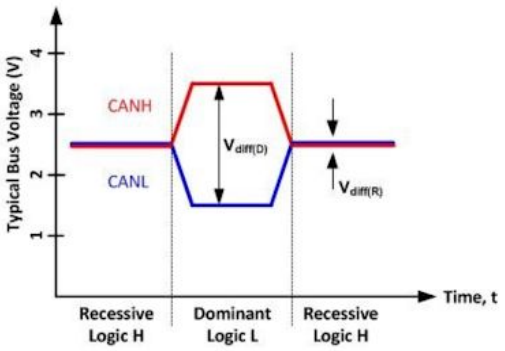



**Figure 4.5a.** Differential CAN voltage signal (Texas Instruments, 2015)

The CAN signal waveform is shown above in Fig. . 4.5a. The protocol operates on a differential voltage signal that sits idle around 2.5 V. When a node transmits a dominant bit (logic level 0),

the high CAN wire shifts up to 5 V while the low CAN wire shifts down to 0 V. Recessive bits are logic level 1. Because of the differential voltage signal and the increased amplitude of the dominant signal, the bus is more robust to errors due to electromagnetic interference.

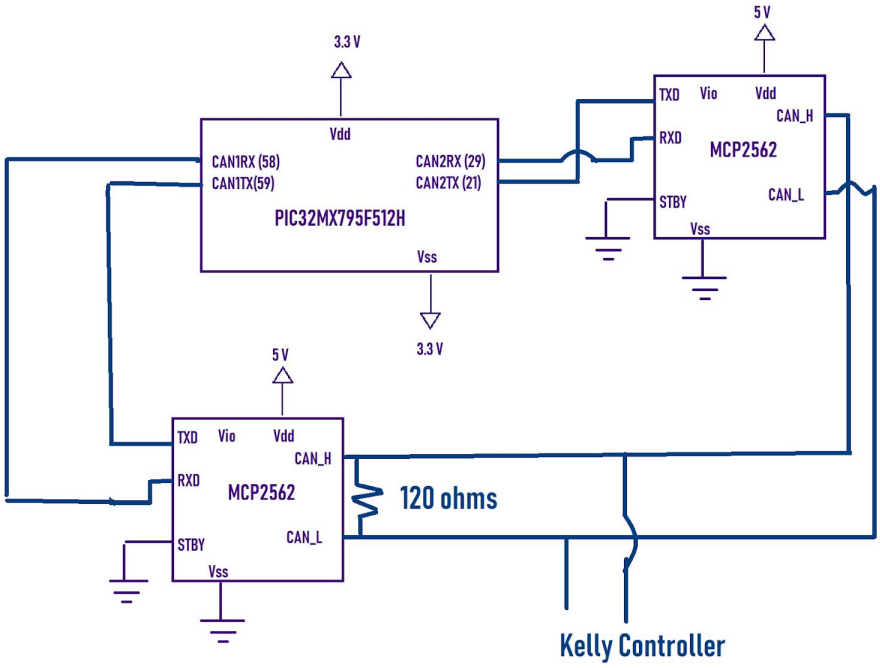



**Figure 4.5b.** Schematic diagram of microcontroller with 2 CAN nodes implemented

Since the PIC32 has two CAN controller modules integrated, only CAN transceivers are needed to implement the system. The CAN transceiver converts the differential voltage signal into logic level 1s and 0s for the microcontroller to process. Although it appears that the transceiver communicates with the microcontroller via UART with the RX and the TX pins, this is incorrect. The TXD pin on the transceiver connects to the CAN2TX pin on the PIC32.

The CAN messages consist of a 29-bit Extended Identifier (EID) that determine the message priority and source address. Following the EID, there is a field for length of data transmitted called DLC. After the DLC, the transmitting node sends the data (usually 8 bytes). Following the DLC there is a cyclic redundancy check (CRC) for errors and an acknowledgement from other nodes.

After the CAN transceiver processes the CAN message, the CAN module must decide whether to accept the message or ignore it. In the module, there is the capability to have 32 acceptance filters and 4 masks. Initially, the receive message acceptance buffer (RMAB) stores the messages that the module will filter. When filtering the message, the mask determines which bits of the message the filter will pay attention to and which bits the filter will ignore. Then, the filters will compare the selected bits with preset values. For example, Message 1 and Message 2 of the controller have different message IDs, 0x0x0CF11E05 for Message 1 and 0x0CF11F05

for Message 2. There is one mask and two filters to differentiate between the messages. The mask bits are all set to 1 to indicate that no bit of the message ID should be ignored. The filter bits encode the values of the message ID: 0b110011110000 for the standard identifier (SID) and 0b010001111000000101 for the extended identifier (EID) for Message 1. After filtering, the messages are stored in the CAN Message FIFOs which store a maximum of 1024 messages per module. The figure below shows how the accepted messages move to the system bus where the CPU of the microprocessor stores them into the FIFOs in the RAM of the microcontroller.

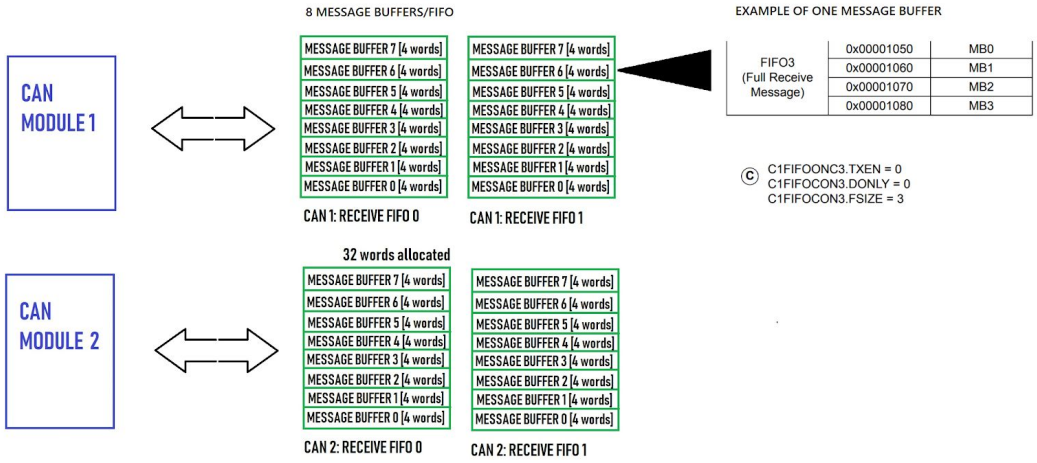



**Figure 4.5c.** CAN Module FIFO Set-Up

## 4.5.2 Original Subsystem Goals

Our team prioritized the motor controllers over the generator controller due to errors with the drive system prohibiting the function of the generator. Our initial goal was to determine if the source address of the Kelly controllers could be modified using software provided by Kelly Controls. Both the motor controllers have the same Message IDs with Source Addresses of 0x05. If we found that modification of source addressing was possible, we would have proceeded to use the distinguished source addresses to differentiate between the left and right motor controllers. If no source addressing modification was possible, we developed a plan to implement both CAN controller modules in the PIC32MX795F512H with one CAN controller per motor controller. The previous Formula Hybrid Design suggested setting one of the switch status bits in Message 2 such as the foot switch for one of the controllers. By checking this bit in the processed message, it would be simple to identify the controllers. However, there are no unused bits in Message 1 that could be user-modified, meaning that Message 1 from each controller would not be differentiated. Since Message 1 contains all of the error messages pertaining to the motor controllers, this solution would not be feasible.

### 4.5.3 Progress Towards Subsystem Goals

*Source Address Modification*

To determine if source address modification was possible on the Kelly Controllers, we first displayed one of the CAN wire signals on the logic analyzer. While we did see output on the logic analyzer that looked similar to the CAN message ID we were expecting there were many errors in the separation of the elements of the CAN message. We decided to try the new Keysight InfinitiVision MSOX3054T Oscilloscope for more precise separation of the Message ID components. After verifying we could view the CAN signal, we attempted to modify the source address with the KMC User App.exe software supplied by Kelly Controls. Unfortunately, there was no clear input field for Source Address. We tried to change the “J CAN Address” field in the software and retested the CAN signal from the controller. However, we noted that there was no change to the source address. After contacting Kelly Controls Inc for support, we concluded that source address modification was not possible. Kelly Controls, Inc essentially reiterated procedures that we had already tested, and did not provide helpful insight into the software issues.

*Implementing Two CAN modules*

To build on the existing software, we wanted to add an additional CAN module controller. In order to understand how to add the new module, we had to observe how data was stored with the single module. First, we tried to use an Arduino and standalone CAN controller (with integrated transceiver) as a transmitting CAN node with the Senior Design demo boards equipped with the PIC32MX795F512H microcontroller and a MCP2562 transceiver that we put on the breadboard. However, we learned that the acceptance filters of the CAN receiver node (the demo board) ignored the Arduino CAN messages because the Message ID differed from the Message ID of the Kelly Controllers. Next, we used the CAN signals of one of the Kelly Controllers as the transmitting node and the same receiving node of the Senior Design demo board. When we used the PicKit3 to debug the software code, we noted that the acceptance filters were still not storing the CAN data. We are not sure why this was the case, but suspect it may have been because the controllers were not fully hooked up to the Hybrid Car which could have caused errors in the CAN Message structure. Unfortunately, due to online learning measures, we were unable to finish this testing, but instead developed a comprehensive Operation Guide for the CAN Bus + Kelly Controllers linked on the website.

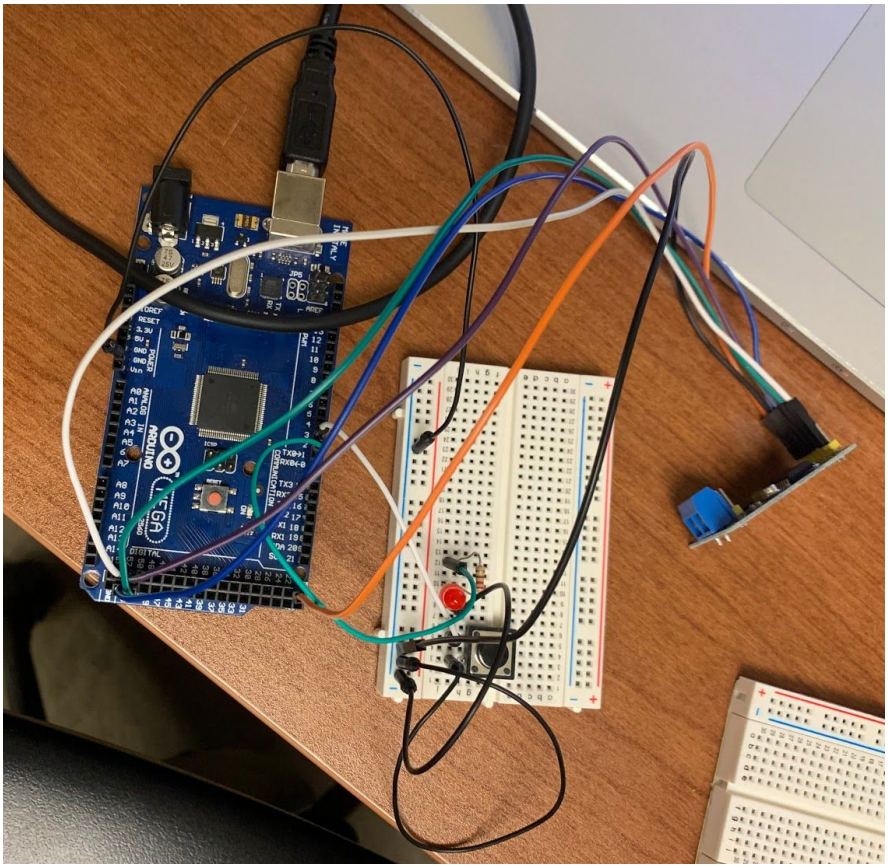


**Figure 4.5d.** Arduino MEGA 2560 and Standalone CAN Controller Testing

## 4.6 Engine Feedback Loop

### 4.6.1 Overview of subsystem

The purpose of this subsystem is to maintain ICE operation at constant RPM during charging. The ICE throttle is actuated by a servo motor that is controlled directly by the motherboard. RPM sensing is accomplished at the generator using a Variable Reluctance Sensor Interface IC, which produces a 3.3V square wave output that is interpreted by the motherboard. The motherboard contains a PID controller that holds the ICE at constant RPM through changing loads.

Design choices were made by the 2018-2019 Team and were not modified by this year's Team. The 2018-2019 team included extensive documentation of this subsystem in section 4.6 of their Final Report, which may be found on the EE Senior Design website.

### 4.6.2 Original goals for subsystem

The 2018-2019 encountered problems with electrical noise on the PWM signal from the motherboard to the servo motor that made testing under real conditions difficult. Therefore, a primary goal of this year's team was to add a noise-resistant cable between the motherboard and the servo motor. Once completed, testing was planned to find PID constants for the controller that would hold RPM steady through changing loads without stalling or excessive overshoot.

A second goal was to implement software that would change the PRM setpoint of the PID controller based on the voltage of the capacitors. Instead of only operating at maximum ICE efficiency, the new system would be able to adapt to the energy needs of the vehicle. If the capacitor voltages were measured to be below a threshold, the RPM setpoint would be increased. If they were measured above a threshold, the RPM setpoint would be decreased. The planned system diagram is shown below:

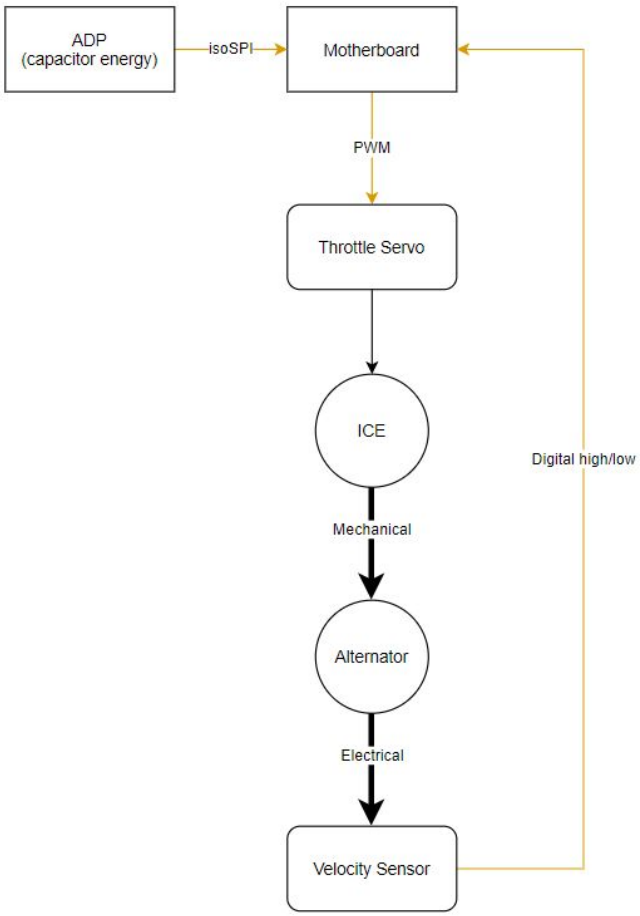



**Figure 4.6** Planned ICE control system

### 4.6.3 Progress Made in Subsystem

The goals for this subsystem can only be accomplished if major parts of the vehicle are integrated (such as the AMS.) Due to initial challenges with the AMS system and the cancellation of in-person classes, no significant work was completed on the subsystem. It is recommended that future teams postpone work on this subsystem until larger goals are accomplished and the entire system is integrated.

## 4.7 Accumulator Management System

### 4.7.1 Overview of subsystem

The purpose of this subsystem is to monitor the capacitor voltages and temperatures, communicate these values to the motherboard, shut down the high voltage system if out-of-range values are detected, and communicate the cause(s) of shutdown to the motherboard.

The system consists of four commercial monitoring boards (or “demo boards”) that directly measure the capacitor voltages and temperatures. These boards are built around the LTC6812-1 multi-cell battery monitoring IC. Four of these boards are stacked in a “daisy chain” configuration to provide complete monitoring of the capacitor stack. 4 of these were chosen since it was an SAE requirement that the voltage of all cells and the temperature of at least 10% of the ultracapacitors is monitored. Since each LTC6812 can measure voltages of 15 ultracapacitors and 9 temperature sensors for the ultracapacitors, 4 such demo boards were sufficient to meet SAE requirements. Furthermore, these demo boards are capable of detecting over-voltages, under-voltages and of balancing the cells. Communication between the monitoring boards and the ADP (and between the ADP and the motherboard) is achieved through an isoSPI interface, thus providing electrical isolation between the high voltage and low voltage systems. Conversion between standard SPI and isoSPI is accomplished by the LTC6820. The isoSPI signals are boosted to a higher voltage for communication between boards by the HM2100NL. The subsystem is summarized in the diagram below.

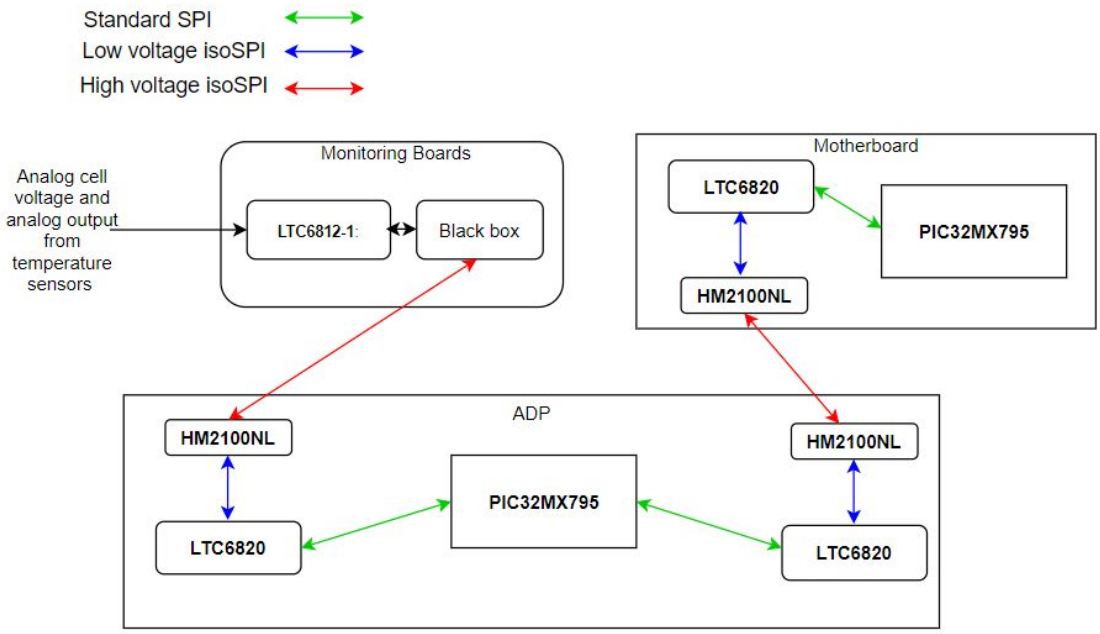



**Figure 4.7.1** Signal flow in AMS

The ADP PCB is an exact copy of the motherboard PCB. However, the ADP has a more focused task, so less hardware is included. The basic software operation of the ADP involves writing over-voltage values to the demo boards then continuously writing a command to begin reading cell voltages and checking the over-voltage flags set by the demo boards. This continues until an over-voltage flag is set, at which point the AIR is opened and an error is sent to the motherboard.

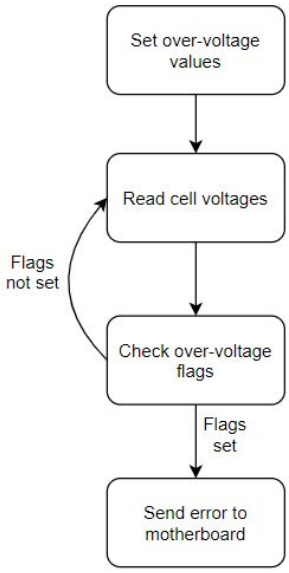


**Figure 4.7.2** Basic ADP software operation

Temperature sensors are not currently included, but can easily be integrated through use of thermistors and general purpose input/output pins on the monitoring boards or extra LTC6812-1 boards. A resistor divider is currently soldered onto the monitoring boards to simulate input from temperature sensors.

### 4.7.2 Original Goals for Subsystem

The 2018-2019 team encountered problems with inconsistent discharge patterns among individual capacitor cells. This led to overvoltage errors caused by individual capacitors that were at voltages much higher than the average capacitor voltage. To fix this problem the 2018-2019 team proposed active or passive capacitor discharging. After reviewing the capabilities of passive discharging on the demo boards, this year's team determined that discharge rates would be too slow to achieve the balancing required. Therefore, we decided to implement active cell balancing. In addition, integration of temperature sensors was planned.

### 4.7.3 Detailed Description of Progress Towards Goals

Upon following the startup guidelines in the 2018-2019 team's Final Report, nonsensical voltage and temperature values appeared on the LCD and the AIR opened, disabling the high voltage system. Upon attempting to solve these errors, the Team realized that the documentation contained in last year's Final Report is inadequate for understanding this subsystem. After cancelation of in-person classes, creating a complete guide to the subsystem became the primary goal. It is included as a Google Slides presentation on the Senior Design website.

Progress was also made in implementing temperature sensors. As the Team of 2018-2019 had recommended from the datasheet, the plan of monitoring the temperatures of at least 10% (or from last year's case of monitoring 60%) of the cells can be expanded by monitoring the temperature of all the cells. Although the team had suggested the use of a mux, further documentation was required, given that there was no documentation on this, to understand the possible use of this mux. Additionally, a plan to understand the thermistors that were tested to function in the 2018-2019 is necessary in implementing temperature sensing that may adjoin the utilization of these thermistors and the mux. By first adding documentation with regards to the thermstors, the Team can better understand its full capabilities. Thus, documentation for the thermistor and external wirings of the hybrid car were created for the purpose of both organization and understanding the plan.

Currently, the thermistors the team has are NTC thermistors, which are also called *Negative Temperature Coefficient* thermistors. In other words, the thermistor's resistance decreases with increasing temperature of the cells to which the specific thermistor is connected to. Thus, in order to use the full capability of the thermistor, this year's Team had to find the resistance-temperature curve/equation for the thermistor. Since this is a new part to be included

in the subsystem, no documentation for the data sheets were found from last year. Therefore, the Team searched for the thermistor part numbers (found to be **TH310J39G)** and calculated specific constraints specific to the thermistor and these values of temperature tolerance or B value were used to calculate the resistance of the thermistor at 25 degrees Celsius.

$$B = 3853.9 \, ln \, (\frac{R25}{R50})$$

**Equation 4.7.3** Equation used to calculate $R_{25}$ of the thermistor given the values of B = 3933 and $R_{50}$ = 10 kOhms from the datasheet in the Thermistor Usage document.

From the above equation and the list of datasheets found in the resistance-temperature pdf under the Thermistor Usage document, the Team was able to find the following resistance-temperature table and equations shown in Figure 4.7.3a below.

| Temp Range (°C) | Ratio | Beta |
|---|---|---|
| 0 to 50 | 9.08 | 3895 |
| 0 to 70 | 18.64 | 3917 |
| 25 to 50 | 2.78 | 3933 |
| 25 to 85 | 9.30 | 3969 |
| 25 to 100 | 14.64 | 3981 |
| 25 to 125 | 29.05 | 3999 |
| 37.8 to 104.4 | 9.67 | 4000 |

To calculate Rt/R25 at temperatures other than those listed in the table, use the following equation:
$Rt/R25 = \exp\{A + B/T + C/T^2 + D/T^3\}$
where T = temperature in K
where K = °C + 273.15

**Figure 4.7.3a** Resistance-temperature equations for the thermistor and the validation of matching the constraints of the thermistor with its datasheet.

Due to the cancellation of in-person classes, the implementation was not done. However, the plan of implementation is detailed in the document called Thermistor Usage. The plan consists of two possible ways to connect the 30 thermistors among the 60 ultracapacitors. No matter the plan implemented, each capacitor will be connected in series with a resistor of value such as 10 kOhms and one lead of this small connection will connect to the positive of one ultracapacitor and the other will be connected to the negative of the ultracapacitor next to the first connection. This is shown in Figure 4.7.3b below.


**Figure 4.7.3b** Connection between a pair of ultracapacitors for thermistors

The first plan is to connect this small connection directly to the ultracapacitors, as done in temperature monitoring of some Li-Ion batteries, by the use of epoxy for safety. The other option is to get 3 additional LTC6812 boards and connect the thermistors in its pins. This can satisfy the temperature measurement of the ultracapacitors since each LTC6812 board is capable of monitoring the temperature of 9 cells. Since the Team currently has 4 or 36 temperatures monitored, the additional 3 boards can sustain the monitoring of the temperatures of the rest of the 24 cells currently not monitored. A mux can also be implemented in the connection with the thermistors to switch the values monitored for each cell temperature. Note that the resistance-temperature equations will be used through the measurement of the resistance of the thermistor using a voltage divider as shown in Figure 4.7.3c below.

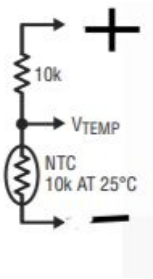



**Figure 4.7.3.c** Voltage divider for measuring resistance of a thermistor to determine its temperature.

## 4.8 Independent Capacitor Charging and Discharging

The purpose of this subsystem is to provide an independent system for charging and discharging the capacitors. Although originally, the team planned on using the built-in system for charging/discharging, many errors within system functionality occurred. One major problem was an error regarding the voltage within the capacitor, which prevented the system from switching states (as described by overvoltage error earlier). Additionally, after discussing with the Formula

Hybrid club, an external capacitor charging/discharging system would be beneficial in the future. As a result, this sub-system was created.

For charging, with help from Eduardo Mello, the current Teacher Assistant for the Electric Hybrid Vehicles class, and some equipment from the Hybrid Electric Vehicles Laboratory, a system which allows for charging with constant 15 mA (relatively safe, mild shock if touched) was designed. This design is a wall-connected variable operation amplifier connected to a transformer in series, which is then connected to the series capacitors using jumper cables. Multimeters were in place to read current, dc voltage, and rms voltage to maintain the current of 15 ± 5 mA throughout. For a full charge, this took around 40 minutes, and although charging with a higher current could reduce charging time, this was not implemented for safety reasons.

**Note: Sparks will occur during initial connection of charge and discharge**

For discharging, the circuit that was designed involves a 1000 Watt rated 47 Ohm power resistor. The design behind this circuit was developed through mathematical equations to create a reasonable discharging time. The calculations are as follows:

1) Assume that total energy from 160 Volts (2.7V*60Caps) and 50F Capacitors = CV2/2 = 640 kJ
2) By implementing a 47 Ohm resistor in a series circuit with 160 Volts, we find current to be 3.404 Amps
3) Power going into resistor given by P = IV = 3.404 * 160 = 544.680 Watts = 544 J/s
4) Assuming capacitor linear discharge, average discharge rate given as totals joules discharged/volts = (P*V/2)/(V) = 272 J/s
5) In theory, 640000 [J] /272 [J/s] = 2352 [s] = 39.22 minutes for a full discharge.

When in practice, discharging from 150V to 55 V (95V difference) took around 23 minutes.

Attached below in Figure 4.8, a diagram of the discharging circuit assuming maximum capacitor voltage and current is shown. Instead of wires, the load was connected to the capacitor series using jumper cables:

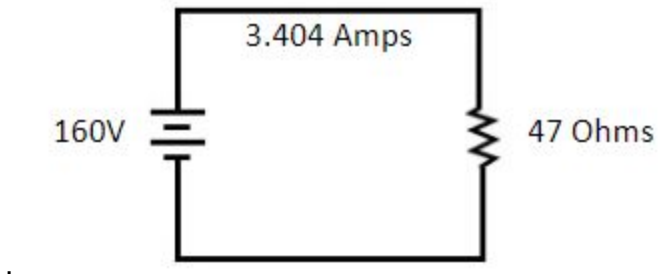



**Figure 4.8** Capacitor Discharge Circuit

# 5. System Integration Testing

Due to the circumstances of the semester, the subsystems could not be integrated. The following is a list of tasks that must be completed in order to bring the vehicle back to a working state:

- AMS: reconnect top demo board in stack to the ADP and to the second demo board using isoSPI cables
- AMS: reconnect capacitor leads to input pins in top demo board in stack
- AMS: verify that overall capacitor voltage is within operational range and charge/discharge as needed using external system
- Mechanical Outputs: verify that there is a Kelly motor controller connected between each hub motor and the motherboard. Connectors are labeled and should be easy to identify.

# 6. Users Manual

The basic operation of the system was not changed from the 2018-2019 year. The user manual from the 2018-2019 year will be paraphrased here with added clarification:

1. Complete the steps in Section 6 to re-integrate the subsystems.
2. Connect a 12V battery to the red/black leads on the left side of the car between the ICE and capacitor bank (with “left” referenced to a person facing the hub-motors from the ICE.)
3. Flip the rightmost switch (with “right” referenced as in step 2.) The switch should be labeled “ECU.”
4. Upload the proper code to both the motherboard and the ADP using a PickIt3. The code for both boards is available on the Senior Design website.
5. Check for errors on the LCD. If the LCD does not turn on or if it shows errors, cycle the system by flipping the ECU switch off and then back on.
6. Flip the “Enable” switch, which is the second from the right. Listen for a loud click. This is the precharge and discharge relays closing. THe LCD should read “Precharge.”
7. After 5 seconds there should be another loud click and the LCD should read “Neutral.”
8. Connect the high voltage system by flipping the “Enable” switch too “off,” turning the large red switch on the capacitor housing to “on,” and flipping the “Enable” switch to “on” again.
9. Test the system by changing the “Direction” switch between forward, neutral, and reverse. Depress the accelerator and brake pedals and turn the wheel to check for motor

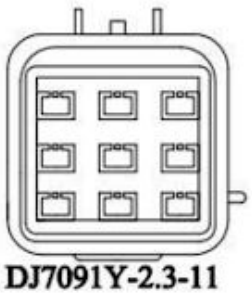
operation, torque vectoring, and recharge.

10. IF THE CAR IS IN AN OPEN AREA, flip the “Start” and then the “Ignition” switch. This will start the ICE.
11. Flip the “Charge EN” switch, which is the leftmost switch. This will allow the capacitors to begin charging.

## 6.1 To test CAN Signals from the Kelly Motor Controllers

To test CAN Signals from the Kelly Motor Controllers:

1. To apply power to the controller (detached from the hybrid car), attach a 12 V DC power supply (GEMTECH in Hybrid Lab) to the Black GND (6) and Pink PWR (7) pins on the connector. Determine which connector of the 3 possibilities by matching the shape of the plastic housing on the connector and the colors of the wires attached to the pins. Alternatively, the Low Voltage System in the car can be activated by attaching **the red and black leads** to a 12 V battery and flipping the ECU switch, LCD screen should turn on. If controllers are properly attached to the car, this should provide the 12 V to the controllers.




| | | |
|---|---|---|
| Orange<br>REV-SW<br>(14) | Black<br>GND<br>(6) | White<br>FWD<br>(12) |
| Red<br>12V<br>(11) | Yellowish<br>12V Brake<br>(25) | Blue<br>ECO<br>(22) |
| Greenish<br>CAN_H<br>(33) | Pink<br>PWR<br>(7) | Brownish<br>CAN_L<br>(34) |

Wiring of Kelly Motor Controllers

2. For quick verification that a signal is being transmitted the Saelae Logic Analyzer will decode CAN messages by attaching the CAN H or CAN L wires and configuring the CAN analyzer settings. For more extensive debugging, the Keysight Infiniti Vision scope located in the Senior Design lab is preferred.
3. To set up CAN on the oscilloscope, we specified the trigger level (1.86V) and the bit rate (250 kbps).
4. Choose the CAN_H or CAN_L to model. Although they both add to the differential signal, when examining raw data only one is necessary.
5. We used the analog inputs on the Keysight scope.
6. A photo reference of what we saw is included in Fig. 11.

7. To analyze the data, we compared the ID and the DATA fields to the Kelly CAN Protocol Description. Note that in the third data entry, the 0x1E signifies that this is one of the motor controllers and not the generator controller. It was experimentally determined by the first formula hybrid team that the generator controller will always have 0x41 in this position.


**Figure 6.1a.** Keysight InfinitiVision oscilloscope analyzing CAN signal from Kelly Controllers

To use the Kelly controller software:

1) Attach the RS-232/USB converter to the rectangular connector on the controller and to your PC.
2) Download the KMC User App.exe.

The software should work when the controllers are plugged in and powered on. If not, see the CAN Operation Guide for debugging tips. This essentially contains a user interface with all CAN data displayed real-time. Ideally, it would also allow for configuration of some of the controller settings, but we did not have any success changing the preset values on the controller.

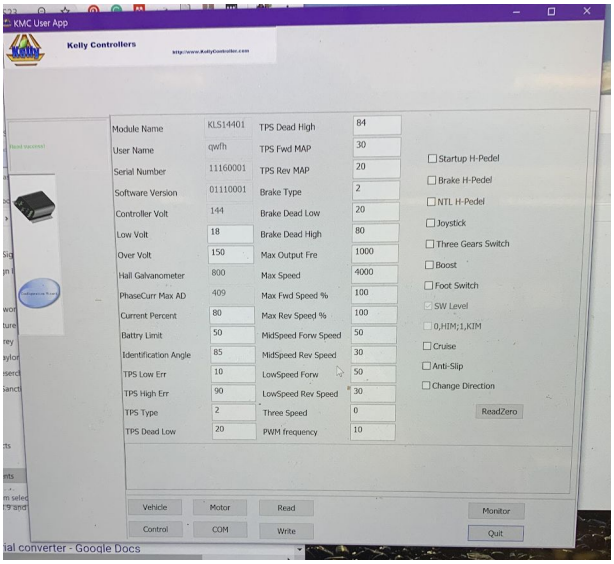



**Figure 6.1b.** KMC User Application

## 6.2 Setting Up Off-Track Computer

Much of the installation process is the same. However, in order to run the current general user interface (GUI)I, the following needs to be done:

1. Confirm the brg value in the MPLAB code and make sure that it is accurate.
2. Make sure the code has been deployed to the microchip
3. The folder serial_GUI should be downloaded in your MATLAB folder.
4. Make sure the unzipped folder is in your MATLAB folder and not just the zipped file.
5. Connect the RF receiver to your laptop using the USB
6. Run serial_GUI.m
7. Indicate the baud rate. Should be default “9600”.
8. Select the port.
9. Hit the Receive Button.
10. If there is no data displayed, hit the receive button again.
11. A photo of the GUI has been displayed in Figure 7.1.1 for reference.

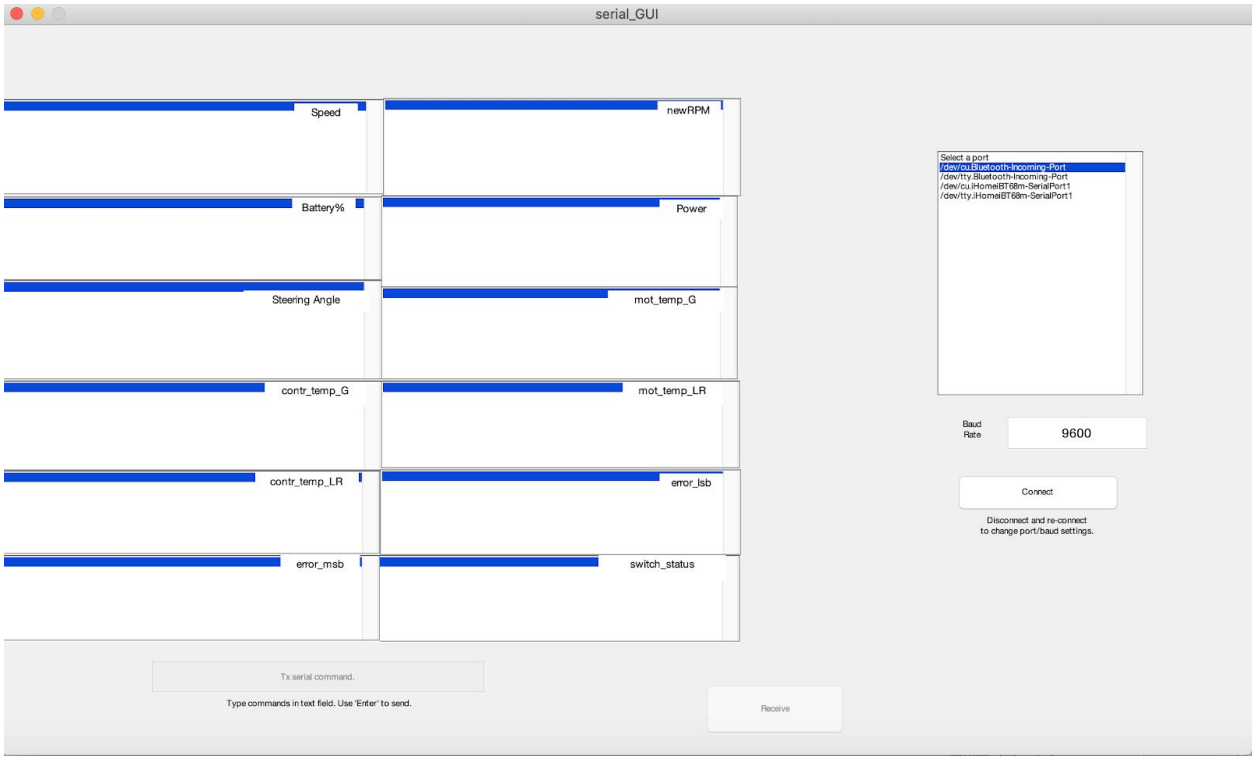



**Figure 6.2.** GUI for Off-track computer

# 7. Conclusions

Although the Team wasn't able to complete its original goals due to the unique challenges of Spring Semester 2020, it was able to make improvements in key areas and to develop more in-depth documentation to allow future teams to understand the legacy system more efficiently. This progress will prove useful to the Notre Dame Formula SAE Hybrid Team as well as to future electrical engineering senior design teams. Undertaking this project gave the Team the unique senior design experience of continuing development on an already existing system as well as working with an external club.

Further development is needed to complete the original goals of this year's team and to integrate them into the system. Once that is completed, the system will need to be rebuilt inside of the car that the Notre Dame Formula Hybrid Team builds. Special care must be taken at this final stage of development to transition the system from a prototype to a robust, race-ready product. The Team hopes that the progress made this year will allow this goal to be achieved within the next couple of years.

# 8. Appendices

### References and Data Sheets:

PIC32MX795F512H: Datasheet

**Note:** Most references remain in line with the 2018-2019 final report. However, some additional purchases were made and are listed below:

Linduino Development Board: Datasheet

Power Resistor: Datasheet

Temperature Resistance Curve: Datasheet

### References for CAN/Motor Controllers:

Kelly Controllers (motors: KLS 14401-8080I, generator: KLS 14401-8080IPS)
Kelly KLS8080I/IPS Motor Controller User's Manual

Kelly Controller Specific CAN Protocol Description
Sinusoidal Wave Controller KLS Broadcast CAN Protocol

Kelly Controller Software Download (we used PC version KMC User App.exe)
Download Link

Standalone CAN Controller with SPI (MCP2515)
http://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf

High Speed CAN Transceiver (MCP2562)
https://www.mouser.com/datasheet/2/268/20005167C-1512552.pdf

2018-2019 Formula Hybrid Senior Design Team Documentation
http://seniordesign.ee.nd.edu/2019/Design%20Teams/ecar/index_code.html#

Arduino CAN library
https://github.com/coryjfowler/MCP_CAN_lib

Arduino CAN tutorial used:
https://www.electronicshub.org/arduino-mcp2515-can-bus-tutorial/

CAN differential signal diagram from:
https://e2e.ti.com/blogs_/b/industrial_strength/archive/2015/06/04/what-do-can-bus-signals-look-like