# Formula Hybrid - Final Report

UNIVERSITY OF NOTRE DAME | COLLEGE OF ENGINEERING

Diego Contreras, Owen Degnim, Kyle Fulk, Zhuoran Han, Brendan Mahoney

Professor Mike Schafer

Senior Design - EE41430

Spring 2021

# Table of Contents

# 1. Executive Summary

We worked with the Notre Dame Formula SAE Hybrid Racing team this year to continue development on a hybrid electric race car for competition in SAE events. Our work focused on the powertrain of the car, refining the systems already in place and implementing new system features. Additionally, our team created numerous pieces of documentation that will help future teams. These documents include practical resources such as the Start-Up Guide and Capacitor Charging Guide that will offer future teams a streamlined start to their project. This final report outlines the overall design of the electrical system for the vehicle including modifications and progress we have made this year. Lastly, we include suggestions for further development on the vehicle.

**1.1 Overview**

The car is a series hybrid powered by a bank of ultracapacitors in series with an internal combustion engine (ICE). A generator motor converts mechanical energy from the ICE to electrical energy. The capacitor bank serves as an energy buffer between the generator and the two electric hub motors. Each of these motors is controlled by a Kelly motor controller. The combination of the 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 total the electrical system is responsible for controlling the throttle of the ICE, monitoring the ultracapacitors' voltage and temperature, taking driver inputs and converting them to motor outputs, and providing information to the driver.

These functions are broken down into five major subsystems:

1. Driver Inputs
2. System Status Interface
3. Mechanical Outputs and CAN bus
4. Engine Feedback Loop
5. Accumulator Management System (AMS)

In addition to these subsystems, an external system for charging and discharging the ultracapacitors was developed in order to charge/discharge the capacitor bank independent of the engine. Although this is not a subsystem, it is crucial for testing the motors and monitoring systems inside without running the engine.

The existing embedded system on the vehicle serves three main functions. First, it monitors the voltage and temperature of the ultracapacitor bank and shuts down the high voltage system if these values are out of range. Secondly, it relays status information from the motor controllers and the AMS to the driver display via UART. Lastly, it takes driver input from the throttle pedal and converts that into an output for the motors.

**1.2 Primary Accomplishments**

- Differentiation of CAN messages between the left and right Kelly controllers to allow for improved error diagnostics
- Rebuild of the system status interface to include a touchscreen, driver-centric display
- Development of comprehensive documentation for future teams

## 2. Definitions and Acronyms

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

**Accumulator** - Energy storage for the high voltage system. Capacitor bank containing 40 capacitors rated 3000F, 2.7V each.

**Motherboard** - Central data processing unit of the hybrid vehicle using the PIC32MX795F512H

**ISO SPI - Isolated Serial Peripheral Interface** - Special SPI protocol with a 2-wire differential signal that allows for reduced susceptibility to noise and longer transmission length, as well as high voltage isolation. Can be converted back and forth between standard SPI and Isolated SPI.

**GLVS - Grounded low voltage system** - Powers the Nextion display, motherboard and separate AMS monitoring PCB board

**HVS - High Voltage System** - Includes the ICE, ultracapacitor bank and motors. Activated when accumulators provide power to the motor controllers.

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

**Nextion** - A touchscreen display used to provide information to the driver

**ADP - Accumulator Dedicated Processor** - board that controls accumulator monitoring system

**PID - Proportional Integral Derivative** - control loop mechanism that calculates error value and applies a correction based on proportional, integral and derivative terms

**AIR -** Accumulator Isolation Relay

# 3. System Requirements

The Notre Dame Formula Hybrid Team plans to use this vehicle to compete in SAE sanctioned events. Therefore, the electrical system must meet the requirements laid out by the SAE Formula Hybrid Rules. Since this is a legacy project and the Notre Dame Formula Hybrid Team was not close to completion on the car's frame, body, suspension and engine, our team did not need to complete all official requirements as laid out by the Formula Hybrid Rules. Nonetheless, most design decisions were made in order to meet those requirements. The summary of relevant rules as stated by the Formula Hybrid SAE website are below, copied verbatim from the 2018 Senior Design final report.

**3.1 SAE Requirements (copied from 2018 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 1). 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 1.** AMS Voltage Monitoring

e) The AMS must monitor the temperature of the minimum number of cells in the accumulator as specified in Table 2 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 2.** 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. Project Description

### 4.1 Theory of Operation

The electrical system consists of a variety of sensors that monitor the vehicle and take user inputs as well as several outputs that control the drive system and display information. The motherboard, with a PICMX795F512H microprocessor, handles a majority of the information processing in the system. There is also an Accumulator Dedicated Processor (ADP) that handles the processing requirements of the accumulator system. The PCB for the ADP is an exact copy of the motherboard and communicates with the motherboard via isoSPI.

Inputs to the motherboard:

- Accelerator pedal (analog)
- Engine RPM (digital)
- Switches (digital)
- Motor Controller Messages (CAN)
- ADP Error Messages (isoSPI)
- Steering Wheel(Digital)

ADP inputs:

- Capacitor Voltages (isoSPI) and Temperature (isoSPI, not yet implemented)

Primary Outputs:

- Nextion Display (UART)
- RF Data Transmission (UART)
- ICE Throttle Servo
- Kelly Motor Controllers (CAN)

### 4.2 System Block Diagram

The overall system block diagram showing the included subsystems and the interface between subsystems is shown below in Figure 1.

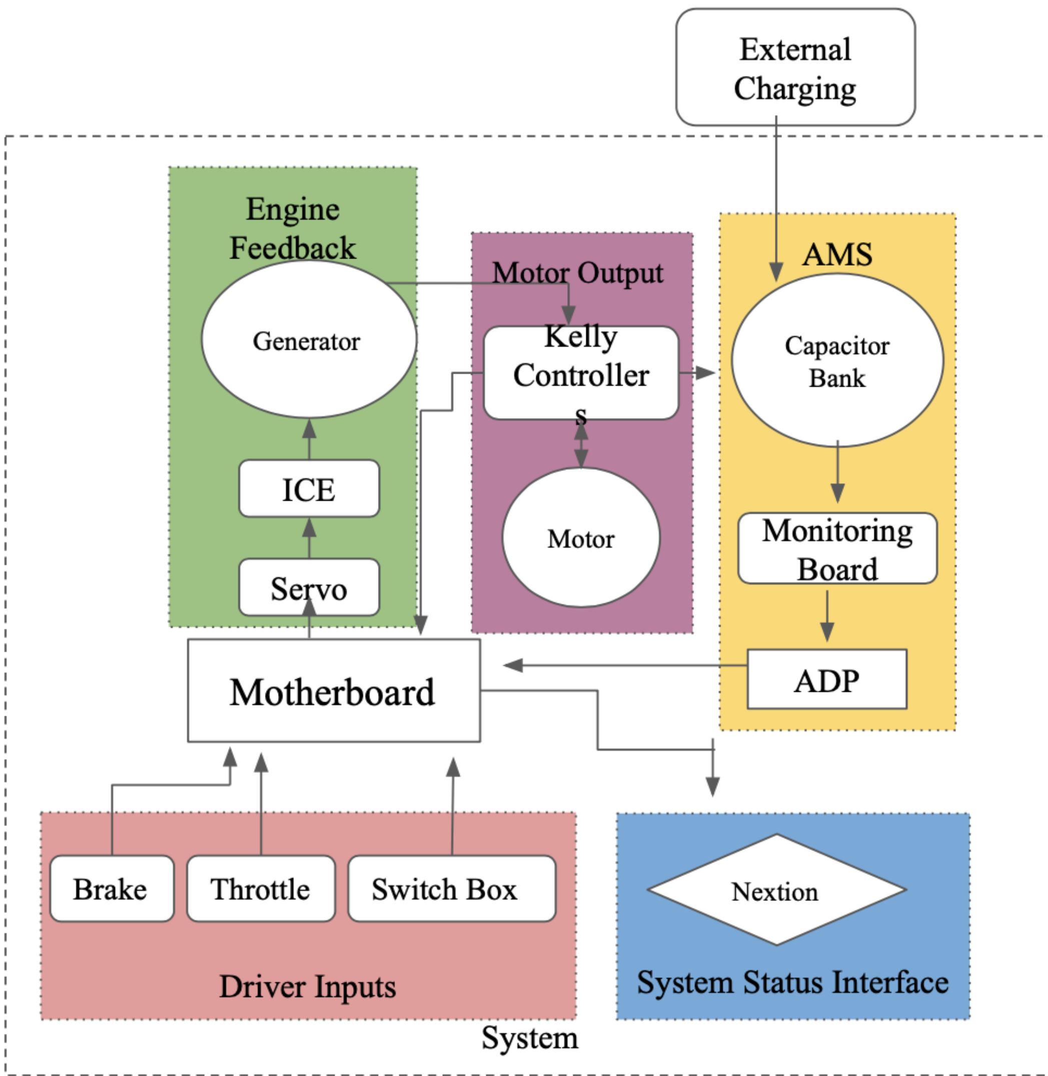


**Figure 1.** Complete System Diagram

# 5. Detailed Subsystem Description

## 5.1 Driver Inputs

5.1.1 Overview of Subsystem

The driver inputs are direct links between the driver and overall electrical system within the hybrid vehicle. These inputs are currently the pedals and the switch box. A steering wheel will need to be added in the future. These inputs connect to the motherboard as shown in Figure 1. The switch box is mainly responsible for the flow of power to the low voltage system.

The pedals consist of an accelerator pedal and brake pedal. Currently there is no mechanical braking system in place in the vehicle so the brake pedal is not currently used. The accelerator pedal is tied directly to both Kelly Controllers providing them with a 0-5V analog signal that controls the speed of the motors. In the future, it will be advantageous to process the throttle signal using the motherboard before sending to the Kelly controllers.

5.1.2 Original Goals

Our team did not put a major focus on the driver inputs, since the mechanical system of the car is not yet in place.

## 5.2 System Status Interface

5.2.1 Overview of Subsystem

The system status interface is responsible for relaying information such as fuel level, capacitor charge level, vehicle speed and engine RPM to the driver and off track team. The updates to the driver are through a display inside the vehicle. The off track team receives data via an RF transmitter that writes through UART communication. The off track system is still under development and will need to be completed by future teams.

5.2.2 Original Goals

Our team decided to improve the system status interface by displaying more information to the driver in a more clean and intelligible manner. We also planned on fixing the RF communication protocol for the off track data logging that the previous team was unable to resolve. We originally planned to improve the LCD that was being used by fixing the color and brightness contrast as well as adding new information to the display.

5.2.3 Progress Made

Our team chose to prioritize the in car display system and focused on displaying data from the motor controllers and AMS. Initially, the car was fitted with a 7-inch LCD display that communicated with the motherboard through an SPI interface. This display system required hundreds of lines of code to incorporate the necessary graphics and suffered with issues of visibility and lack of flexibility. Thus, we chose to replace the previous screen with a 10.1-inch Nextion touchscreen display, shown below in Figure 2, that communicates with the motherboard via UART communication.

**Figure 2**. Nextion Display

There were multiple reasons for choosing the Nextion display over a basic LCD screen. One advantage of the Nextion display is that it allows us to preload the graphics onto the display with an SD card, which leads to a much simpler and streamlined process for the motherboard code. The Nextion display allows for multiple pages which offers better informational organization, reducing the driver workload and it will allow for future teams to add information with ease. Additionally, the process of writing data to the Nextion over UART is much simpler than writing to the previous LCD. Lastly, there is a strong wealth of resources online for the Nextion display.

In our design, the Nextion receives data from the motherboard and displays the information on two pages: the Primary Display and Secondary Display shown in Figures 3a and 3b below.

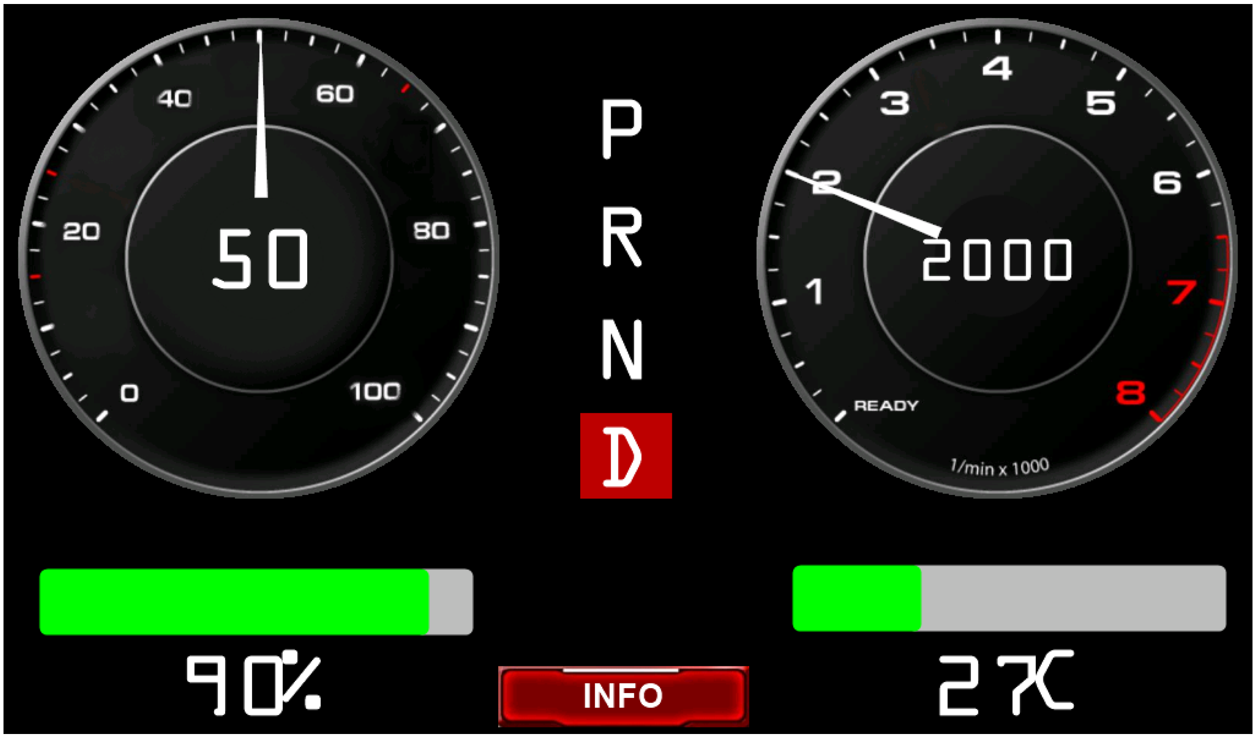


Figure 3a. Primary Display Page

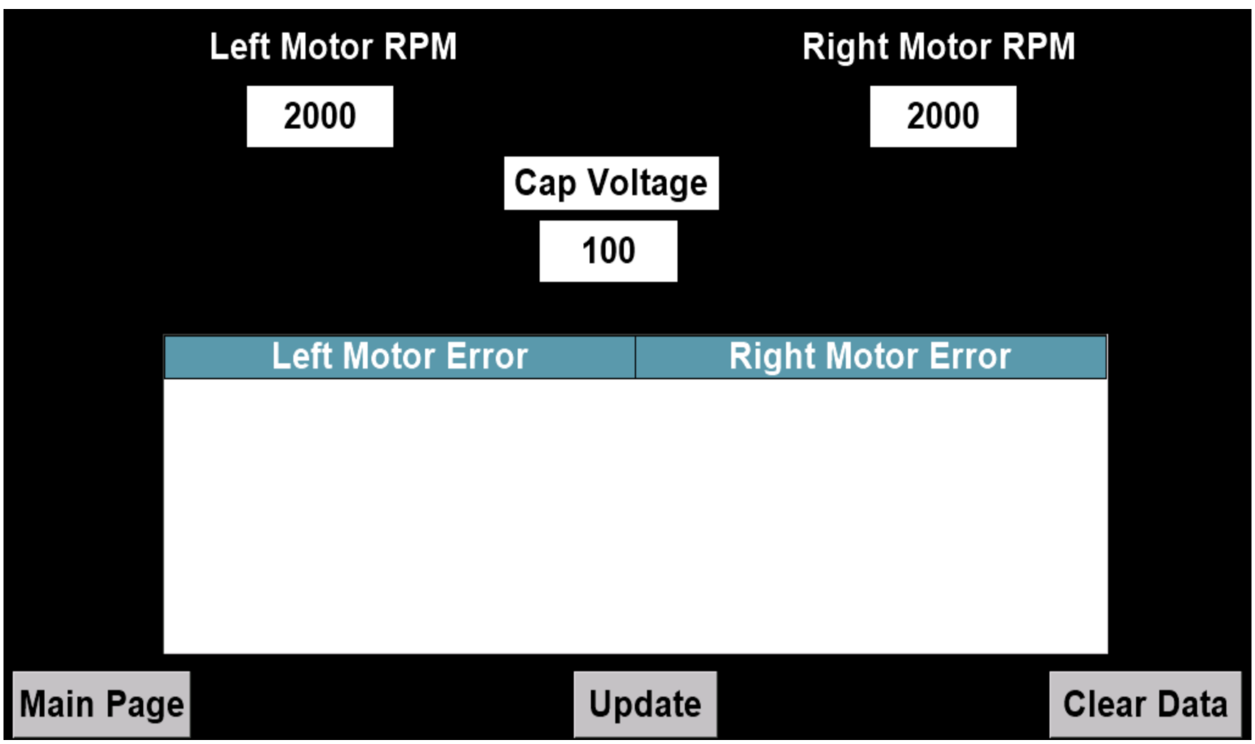


Figure 3b. Secondary Display Page

The Primary Display is similar to a traditional dashboard display and gives the driver relevant information on vehicle speed, drive state, capacitor bank charge, engine temperature and engine RPM. Since the engine is not currently working, our team has chosen to display motor RPM instead of the engine RPM on the Primary Display. The Secondary Display includes additional information that can be accessed by touching the INFO button on the Primary Display. This information includes left and right motor RPM, capacitor voltage, and error messages from the motor controllers.

The Nextion communicates with the motherboard through UART protocol. The baud rate of the motherboard and Nextion are currently set to 31250, but can be changed to certain presets on the Nextion. The messages are sent to the Nextion on a timer that triggers once a second. The process of sending a message over UART is fairly simple and another reason why we switched to the Nextion display. Each text box and graphic on the display is labeled in the Nextion Editor where the graphics are formatted and loaded onto the display with an SD card. In order to write to a specific box you must send “boxname.val=” followed by the value you want to send. Each instruction set must end by sending 0xff three times in order for the Nextion to know that the instruction is complete. An example of writing the speed and RPM values to the Primary Display is shown below in Figure 4. Where the Terminate function sends 0xff three times and the Send function sends the string character by character to the Nextion.

```
void Write_val(int rpm, int speed){
    Send("nrpm.val=");
    printf("%d",rpm);
    Terminate();
    Send("nspeed.val=");
    printf("%d",speed);
    Terminate();
}
```

**Figure 4.** Nextion Write Example

The issues with the RF transmitter still need to be addressed as our team chose to prioritize more urgent updates to the subsystem.

## 5.3 Mechanical Outputs and CAN Bus

### 5.3.1 Overview

The mechanical output subsystem involves the generator transferring power from the ICE to the ultracapacitor bank and the ultracapacitors powering the motor controllers. The motor controllers regulate the motor RPM, current, throttle and temperature. They also monitor the motor RPM, motor current and motor temperature and communicate this information to the motherboard using CAN bus 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. 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.

### 5.3.2 Original Goals

The previous year's team planned to obtain information from both motor controllers and display them separately. They also wanted to use different motor RPM readings to implement torque vectoring. However, they failed to differentiate between the left and right controller.

### 5.3.3 Progress Made

This year we improved the CAN network and successfully implemented the differentiation. We achieved this by first reflashing the Kelly controllers and using the logic analyzer to read their corresponding extended identifier bits (EID). The EID bits in CAN Bus protocol allow identification of individual devices. Using two filters, each matching the EID bits of the right and left controller, we read CAN messages from both controllers and stored the information in four FIFO message buffers. FIFO0 and FIFO1 are reserved for the right controller, and FIFO2 and FIFO3 are reserved for the left controller. Finally, we decode the messages in the FIFO buffers and report CAN messages for our system status interface.

For future teams' reference, we used the following settings in Kelly controllers' user program:

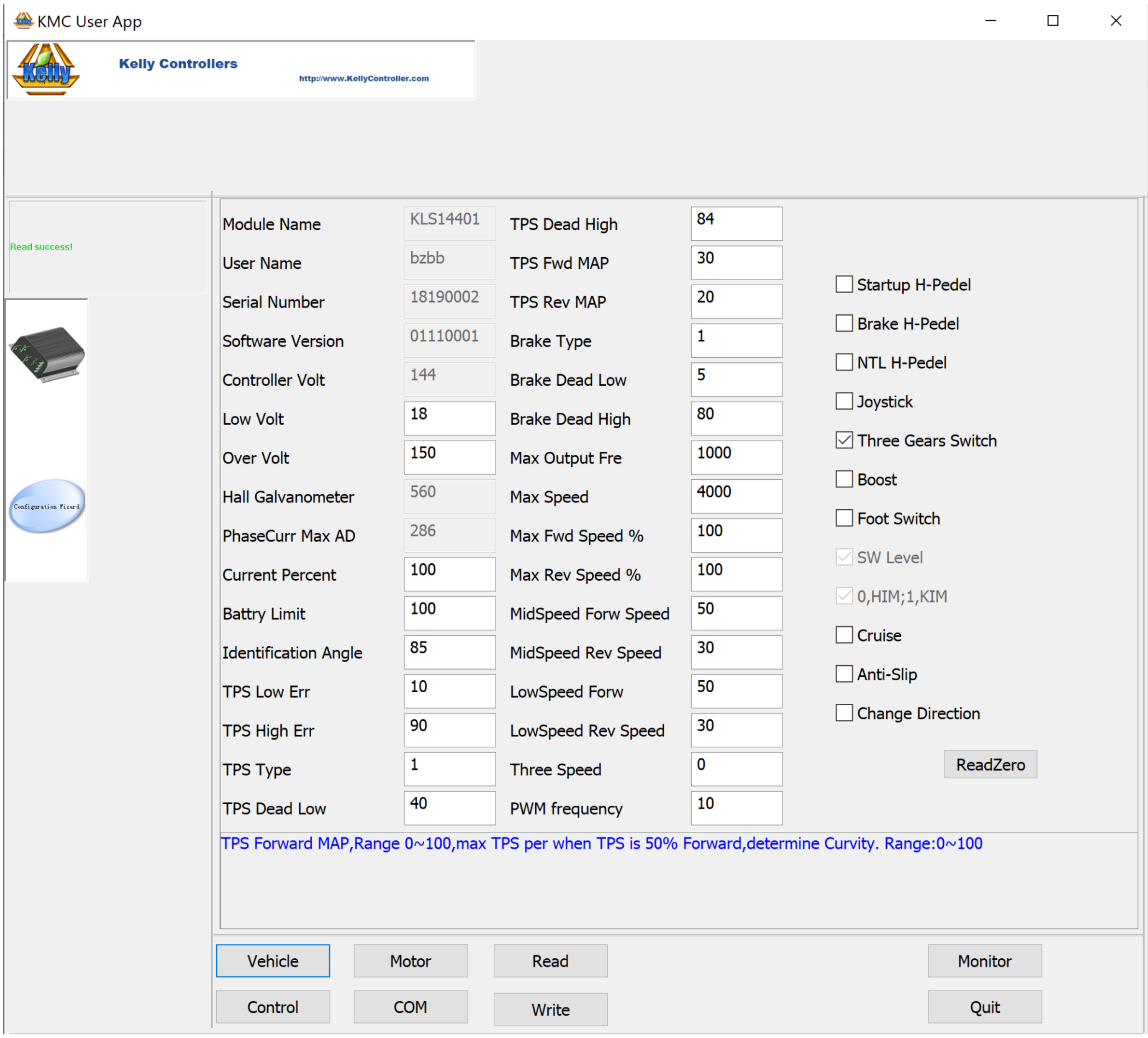


**Figure 5**. Kelly controller settings

## 5.4 Engine Feedback Loop

5.4.1 Overview

The engine feedback loop subsystem maintains the ICE at a constant RPM during capacitor charging. The ICE throttle is controlled by a servo motor that is tied directly to the motherboard. The RPM of the engine is sensed at the generator using a Variable Reluctance Sensor Interface that produces a 3.3V square wave output to be interpreted by the motherboard. The motherboard has a PID controller that keeps the ICE at a constant RPM throughout the capacitor charging state.

These initial design decisions were made by the 2018 team and have not been modified by subsequent teams. The 2018 team has extensive documentation of this subsystem that is available on the [2018 senior design website](2018 senior design website).

### 5.4.2 Original Goals

Our team planned on creating a more advanced engine control algorithm that would allow for greater efficiency in different racing environments. These would be implemented using the PID controller that is currently in the system. We planned on integrating this algorithm with the AMS in order to get information about the energy remaining in the capacitor bank. This added information would allow for the development of an algorithm that will prevent the capacitors from overcharging and prevent depletion. Below in Figure 5 is a block diagram of the planned system.

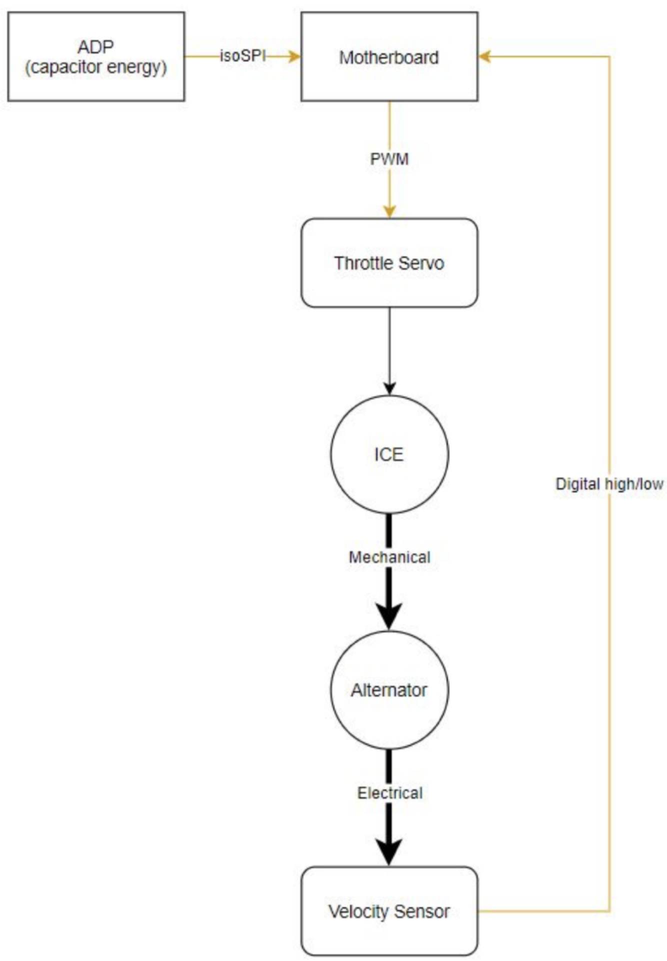


**Figure 6**. Planned Engine Feedback Loop

#### 5.4.3 Progress Made

Our team was unable to work on the engine feedback loop due to the ICE not working properly. In order to make progress on this subsystem future teams will need to first resolve the issues with running the engine. It is recommended that future teams also complete the AMS subsystem before focussing on the engine feedback loop.

### 5.5 Accumulator Management System (AMS)

#### 5.5.1 Overview

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 of the shutdown to the motherboard.

The AMS subsystem consists of four demo boards that are directly connected to the capacitors and directly measure capacitor voltage and temperature. These four boards are daisy chained together to provide monitoring of the entire capacitor stack. Four boards were chosen to monitor the capacitor bank because it is an SAE requirement that all cell voltages and at least 10% of ultracapacitor temperatures are monitored. Each board is equipped with an LTC6812 multi-cell battery monitoring IC. Each LTC6812 can measure the voltages of 15 ultracapacitors and can read from 9 temperature sensors, so four boards are sufficient to meet the SAE requirement. These boards are also able to detect over-voltages, under-voltages and the balancing of the cells.

Communication between these four boards and the ADP board is achieved through isoSPI communication. The ADP board also communicates with the motherboard through isoSPI, providing electrical isolation between the low voltage and high voltage systems. The isoSPI communication within the subsystem is shown below in Figure 6.

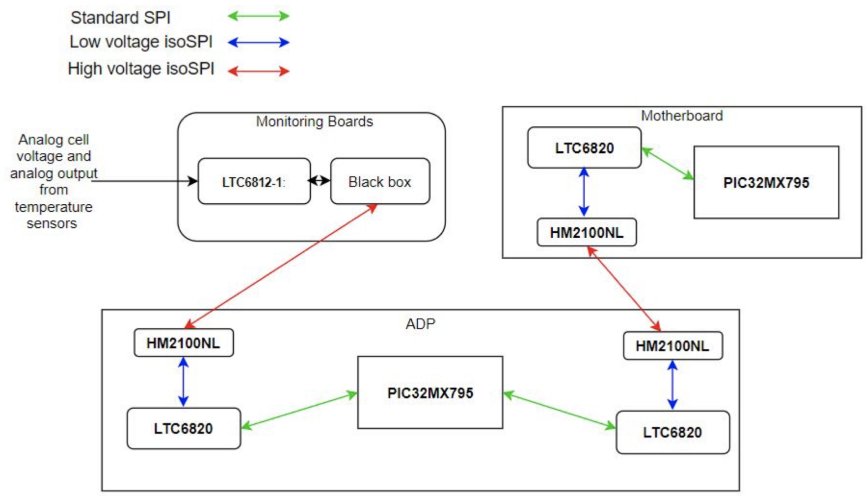


**Figure 7.** Communication in AMS

The ADP board that communicates with the four demo boards is an exact copy of the motherboard. Since the software for the ADP system was not changed by our team we will summarize the code from the 2018 team. The code for the ADP involves writing over-voltage values to the demo boards and continuously writing a command to read cell voltages and check over-voltage values. This continues unless an over-voltage value is detected, at which point the AIR is opened and the error is sent to the motherboard. Below is a software flow diagram for the ADP board created by the 2019 team.

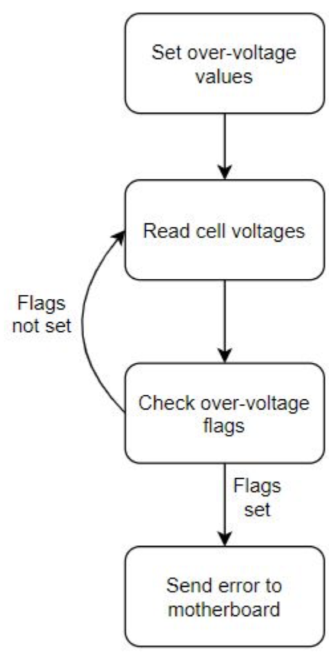


**Figure 8.** ADP Software Flow

5.5.2 Original Goals

Based on previous documentation from the 2018 and 2019 teams, we believed that this subsystem was complete. Therefore, our team chose to prioritize other subsystems. Once we began working on the entire system it was discovered that the temperature sensing had not yet been implemented as described by the 2019 team. Due to issues with other subsystems and in the interest of time we did not implement the temperature sensing in the AMS. Future teams should use the 2019 Final Report as a guide for implementing the temperature sensing.

# 6. Manuals for Future Teams

A primary goal of our work was to create documentation for future teams to ease the start-up process and allow teams to jump into development. Below is a linked list of informative documents that will be useful for future teams.

- Start-Up Guide: How to turn things on - powering the low voltage system and spinning the motors
- Capacitor Charging Guide: Step by step instructions for charging/discharging capacitors with safety information
- Faulty Capacitor Testing Guide: outlines 2 processes for testing capacitors to determine if they are faulty
- Website Guide: How to access and upload documents to website
- Kelly Controller Reflashing Guide: How to reflash the Kelly controllers
- Nextion Guide: How to add graphics and write to the display

# 7. Suggestions for Future Teams

In order to accomplish a working, race-ready vehicle future teams will need to complete certain subsystems and integrate all subsystems. The subsystems that should be prioritized are the AMS and the Engine Feedback Loop. Also the current capacitor bank consists of only 40 capacitors, as some were found faulty by our team. It is recommended that 20 new capacitors are added back into the capacitor bank. These capacitors must be the same type as the current capacitors. Dr. Bauer in the EE department is a good resource for getting capacitors.

## 7.1 AMS

The capacitor temperature sensing by the ADP and demo boards must be achieved in order to meet SAE regulations. The 2019 team describes how to implement thermistors to the monitoring boards in order to measure capacitor temperature. There is also a document on their website called ‘Capacitor Temperature Monitoring’ that outlines a plan for implementing the thermistors and calculating capacitor temperature. It is suggested that future teams use this documentation to achieve temperature monitoring by the AMS.

## 7.2 Engine Feedback Loop

Since the engine was not working our team was unable to make any progress on this subsystem. In order to complete the engine feedback loop future teams must first make sure that the engine is running properly. Also the generator that is currently connected to the vehicle requires a liquid cooling system that has not yet been implemented. If this generator is to be used in the future the liquid cooling system will need to be implemented.

## 7.3 System Status Interface

Future teams will need to implement an off track monitoring system that displays similar data to that displayed on the Nextion. The 2019 team started to develop an RF transmitter that would send data to a PC using a matlab GUI. Our team did not focus on completing this system as there were more pressing matters. It is recommended that future teams use RF transmission to achieve off track monitoring. The 2019 team matlab GUI should be used as a starting point for this system.

## 7.4 Board Design

The current motherboard being used is the same one designed by the 2018 team. Since then many changes have been made to the system and it would be beneficial to redesign the motherboard in order to be more tailored to the current system. One thing that will need to be added is a second accessible UART output. Currently the board only has one UART output, but since the Nextion and RF transmitter both use UART a second must be added. Also, the relays on the board are not currently functioning properly. The board should be redesigned so that the board's relays control the drive states of the vehicle and set the pins on the Kelly to proper 0/12 accordingly. Currently, the Nextion display is powered by a 12V-5V DC-DC converter, drawing approximately 1 amp. To simplify the system, future teams could consider adding a similar power supply on the motherboard for powering the display.

### 7.5 Recommended Purchases

Currently we are borrowing a power supply, diode bridge rectifier, and current probe from Dr. Bauer in order to charge the capacitors independently of the engine. It is recommended that future teams purchase their own power supply, rectifier, and current probe.

# 8. Conclusions

Our team made meaningful progress on this legacy project on the CAN bus, System status Interface, capacitors and developing documentation that will be useful for future teams. While we were not able to accomplish everything laid out in our project proposal, much of this was due to a lack of functioning engine and car frame/body that is necessary for taking this project to completion. Given our progress this year and communication with the team, the future is bright for the Formula 1 Hybrid team and with the right leadership and team members, they should have a race-ready car in another 1-2 years time.

# 9. References


Formula Hybrid 2021 Rules:

https://formula-hybrid.org/students/rules-and-deadlines/

2019-2020 Team Website:

http://seniordesign.ee.nd.edu/2020/Design%20Teams/formula/index.html

2018-2019 Team Website:

http://seniordesign.ee.nd.edu/2019/Design%20Teams/ecar/index.html