Formula SAE Hybrid Racing–Senior Design Master Document (2019-22)

TABLE OF CONTENTS

[1. DEFINITIONS 3](#_Toc101876690)

[2. INTRODUCTION 5](#_Toc101876691)

[2.1 YEAR-TO-YEAR PROGRESS 5](#_Toc101876692)

[2.2 CONTROLS 8](#_Toc101876693)

[2.3 MONITORING 8](#_Toc101876694)

[2.4 STATUS 9](#_Toc101876695)

[2.5 PERFORMANCE 9](#_Toc101876696)

[3. SYSTEM REQUIREMENTS 10](#_Toc101876697)

[4. PROJECT DESCRIPTION 14](#_Toc101876698)

[4.1 SYSTEM THEORY OF OPERATION 14](#_Toc101876699)

[4.2 SYSTEM BLOCK DIAGRAMS 16](#_Toc101876700)

[4.3 DRIVER INPUTS 20](#_Toc101876701)

[4.3.1 SUBSYSTEM REQUIREMENTS 20](#_Toc101876702)

[4.3.2 WIRING SCHEMATICS AND DESCRIPTION 20](#_Toc101876703)

[4.3.3. SOFTWARE DESCRIPTION 22](#_Toc101876704)

[4.3.4 CHOICE OF COMPONENTS 23](#_Toc101876705)

[4.3.5 FUTURE HYDRAULIC BRAKING 23](#_Toc101876706)

[4.3.6 SUBSYSTEM TESTING 25](#_Toc101876707)

[4.4 SYSTEM STATUS INTERFACE 26](#_Toc101876708)

[4.4.1 LCD (2019-2020) 26](#_Toc101876709)

[4.4.2 NEXTION (2021-2022) 27](#_Toc101876710)

[4.4.3 RF TRANSCEIVER 29](#_Toc101876711)

[4.5 MOTOR AND GENERATOR CONTROLLERS 30](#_Toc101876712)

[4.5.1 SUBSYSTEM REQUIREMENTS 30](#_Toc101876713)

[4.5.2 WIRING SCHEMATIC 30](#_Toc101876714)

[4.5.3 DESIGN CHOICES 31](#_Toc101876715)

[4.5.4 PROGRAMMING STATE DIAGRAMS 36](#_Toc101876716)

[4.6 ENGINE FEEDBACK LOOP 37](#_Toc101876717)

[4.6.1 SUBSYSTEM REQUIREMENTS 37](#_Toc101876718)

[4.6.2 WIRING SCHEMATIC 38](#_Toc101876719)

[4.6.3 DESIGN CHOICES 38](#_Toc101876720)

[4.6.4 STATE DIAGRAM 40](#_Toc101876721)

[4.7 ACCUMULATOR MANAGEMENT SYSTEM 41](#_Toc101876722)

[4.7.1 SUBSYSTEM REQUIREMENTS 41](#_Toc101876723)

[4.7.2 WIRING DIAGRAM 45](#_Toc101876724)

[4.7.3 STATE DIAGRAM 46](#_Toc101876725)

[5 SYSTEM INTEGRATION TESTING 46](#_Toc101876726)

[5.1 SYSTEM INTEGRATION OVERVIEW 46](#_Toc101876727)

[5.2 SYSTEM REQUIREMENT CHECK 47](#_Toc101876728)

[6 USER’S MANUAL & INSTALLATION GUIDE 48](#_Toc101876729)

[6.1 INSTALLATION 48](#_Toc101876730)

[6.2 SETUP, VERIFICATION, AND TROUBLESHOOTING 48](#_Toc101876731)

[7 TO-RACE DESIGN CHANGES 53](#_Toc101876732)

[7.1 DRIVER INPUTS 53](#_Toc101876733)

[7.2 SYSTEM STATUS INTERFACE 54](#_Toc101876734)

[7.3 MOTOR CONTROLLERS/MOTORS/GENERATORS 54](#_Toc101876735)

[7.4 ENGINE FEEDBACK LOOP 55](#_Toc101876736)

[7.5 ACCUMULATOR MANAGEMENT SYSTEM 56](#_Toc101876737)

[8 CONCLUSIONS 57](#_Toc101876738)

[9 APPENDICES 58](#_Toc101876739)

[9.1 REFERENCES AND DATA SHEETS 58](#_Toc101876740)

1. DEFINITIONS

**Accumulator**:​ Energy storage for the high voltage system. Capacitor bank containing 60

capacitors rated 3000F, 2.7V each.

**ADP**: Accumulator Dedicated Processor. A PIC32MX795 PCB that is a mirror of the Motherboard with fewer soldered-on components. It takes in accumulator information and sends error messages to the Motherboard as an isoSPI slave.

**AMS**:​ Accumulator Management System. Circuitry that continuously monitors accumulator voltage levels and safely shuts down high voltage system if an issue arises.

**AIR**:​ Accumulator Isolation Relay. A relay that opens a high voltage loop. Must be normally open.

**ESF:** Electrical System Form. Documentation turned in with competition entry (not included in project scope).

**FIFO**: First In First Out. Buffer implementation where data written in first, comes out first.

**CAN or CAN Bus:** Control Area Network. An automotive data bus standard, it uses differential channel for (in this case) broadcast type data communications between the microprocessor and the motor controllers

**CRC**: Cyclic Redundancy Check. Used in CAN messages to check for errors in broadcasting.

**Discharge Circuit:** Brings the voltage of the intermediate circuit between the accumulators and the tractive system loop down below the required values within the 5 second limit.

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

**EID**: Extended Identifier. Refers to the form of ID read in to the Motherboard in CAN message processing.

**FIFO**: First In, First Out. Computing approach where messages are prioritized in order of when they were read in.

**GLVS**: Grounded Low Voltage System. 12V battery circuit that powers the Driver Display, Motherboard, ADP, and starts the motor and generator controllers

**GLVMS**: Grounded Low Voltage Master Switch. Controls power to the low voltage system.

**GUI**: Graphical User Interface.

**HVS**: High Voltage System. Activated when the accumulators provide power to the motor controllers and must be galvanically isolated from the GLVS. Includes the ICE, ultracapacitor bank and motors.

**IC Engine or ICE**: Internal Combustion Engine. 250cc engine from Kawasaki Ninja motorcycle.

**ISO SPI**: Isolated Serial Peripheral Interface. A 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. It is possible to convert back and forth between standard SPI and isoSPI using the LTC6820 on the Motherboard and ADP.

**Motherboard:** PCB of central PIC32MX795F512H microcontroller for the Hybrid Vehicle.

**Nextion**: A touchscreen display used to provide information to the driver.

**NTC**: Negative Temperature Coefficient. Refers to thermistors that decrease in resistance with increasing temperatures.

**PCB**: Printed circuit board.

**PID controller**: a proportional-integral-derivative controller is used for continuously modulated control. It calculates an error value in applies a correction in a control loop mechanism.

**Precharge Circuit:** Circuitry that protects AIR contacts by charging an intermediate circuit to 90% accumulator voltage capacity before closing the relay.

**RMAB:** Receive Message Acceptance Buffer. Stores messages the CAN module on the Motherboard will filter.

**Shutdown Circuit:** Current loop that holds the AIRs closed and disconnects the high voltage system from the accumulator when opened.

**SID**: Standard identifier. Refers to the form of ID read in by the Motherboard during CAN message processing.

**SOC**: state of charge. Refers to the charge level of the ultracapacitors which comprise the accumulator.

**TS**: Tractive System

**TSV**: Tractive System Voltage. High voltage circuitry powered by accumulators.

**SAE**: Society of Automotive Engineers.

**SMD**: Surface Mount Device. Used on PCBs rather than through-hole components.

2. INTRODUCTION

The Notre Dame Formula Hybrid team has worked for several years now on constructing a series hybrid-electric racecar that will be used in future Formula Hybrid SAE competitions. 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, while the capacitor bank serves as an energy buffer between the generator and the two electric hub motors, each of which is controlled by a Kelly motor controller. The combination of the ICE, ultracapacitors, and motors is the high voltage system, while monitoring and controlling the high voltage system is based on user inputs achieved by the low voltage system.

The vehicle includes five electric subsystems:

* Driver Inputs
* System Status Interface
* Motor Controllers/Generators/Motors
* Engine Feedback Loop
* Accumulator Management System (AMS).

The central Motherboard handles the vast majority of processing and communication between the subsystems, in addition to the ADP mandated by SAE rules.

2.1 YEAR-TO-YEAR PROGRESS

In 2019, an Electrical Engineering Senior Design Team (EESD team) worked on creating:

* An embedded electronic system to aid in the monitoring and optimization of various mechanical systems
* The system would take inputs from the driver and control the motors, engine, and driver displays electronically
* The system would continually monitor the high voltage system and shut it down if a problem arises

Details of this embedded system are as follows:

* A multifaceted embedded system that used a PIC32795 microprocessor with three main functions:
  + **Controls** – electronic control of the charging system and operation of the drive system
  + **Monitoring** – measuring the temperature and voltage levels of the system and responding to irregularities by shutting down the high voltage system
  + **Status** – giving both the driver and off-track time live updates of vehicle diagnostics from the motor and generator controllers as well as the AMS

In 2020, an EESD team worked on the following:

* Identified a need for testing procedures of the existing control system
* Identified a need for more in-depth documentation of the entire system to allow future teams to understand the underlying functions of the subsystems
* Developed ultracapacitor charging and discharging methods to change the capacitor state of charge (SOC) independently of the engine
* Documented errors in AMS – incorrect capacitor SOC and temperature readings that were in the negative range
  + System is designed to open relays on the high voltage system when values outside a specified range are read, preventing high voltage system testing
* Solved the problems shown in Table 2.1.1 below.

|  |  |  |
| --- | --- | --- |
| **Problem** | **Solution Approach** | **Implemented Solution** |
| (1) Car was stuck in “Wait” state with given error of battery charge at 142% | Engine and generator did not work, so method for independently charging/discharging capacitors was needed  Successfully diagnose error messages from the AMS PCB | Developed method for independent charge/discharge of capacitors  Created extensive documentation on AMS system |
| (2) Implement an improved user interface to display error messages from the Kelly Controllers | Baud rate was unknown for LCD display and code used incorrect settings for baud rate configuration | Baud rate was not addressed, but a GUI was designed to display off-track race data to a computer via an RF transmitter |
| (3) Implement hydraulic braking | Hybrid Club had not installed needed hardware for braking, such as wheel rotors | Developed diagrams and general understanding on implementation of hydraulic braking for future reference |
| (4) Differentiate between the left and right Kelly motor controllers | Source address of Kelly Controllers can not be physically changed, potentially implement the two CAN controller modules on the PIC32, using one controller module for each controller | Developed extensive documentation of testing procedures and overview of the CAN network protocol |
| (5) Improve voltage and temperature monitoring of the AMS subsystem by monitoring each of the 60 ultracapacitors | Thermistors had no resistance-temperature equation curve, and lack of extra pins on LTC6812-1 boards necessitates monitoring every pair of ultracapacitors | Documentation on finding the resistance-temperature curve/equation for the thermistors, outlined plan for connecting them to ultracapacitors |
| (6) Add error margins to increase reliability of the minimum and maximum acceptable values of the throttle | Currently, false errors exist such as variations in the supply voltage or electrical noise triggering false fault conditions | No progress made due to more pressing errors with respect to systems integration |

**Table 2.1.1 Progress Made by 2020 Team**

In 2021, an EESD Team worked on the following:

* Differentiation of CAN messages between left and right Kelly controllers for improved error diagnostics
* Rebuilt the system status interface to include a touchscreen and drier-centric display using a Nextion as a replacement for the original LCD
* Developed comprehensive documentation for future teams

2.2 CONTROLS

The Control System takes driver inputs and controls vehicle functions:

* **Charge system** – controlled from a switchboard next to the LCD display
  + *Function*: deliver power from the internal combustion (IC) engine to the ultracapacitor banks (accumulator)
* **Driver inputs** – several different functions
  + Switches to turn on the 12V low voltage system
  + Precharge the controllers to match the voltage of the accumulator
  + Enable the high voltage system
  + Turn on ignition
  + Start the IC engine
  + Toggle charging/not charging states
* **Drive system**  — ability to shift between forward, neutral, and reverse drive states via a switch. Controls include:
  + **Throttle pedal** – engages the hub motors in the rear wheels
  + **Brake pedal**  — engages regenerative braking in the same rear wheels (returns power in the wheels back to the accumulator)
  + **Differential steering** – when steering wheel is turned, the torque of the outside wheel is increased to improve drive performance.
  + *Additional functionality****:*** automatically shut down the high voltage system independent of any driver inputs (in case of an error detected by monitoring system)

2.3 MONITORING

Monitoring system continuously measures voltage across each capacitor cell in the accumulator to ensure even power distribution in both charging and discharging states. Additional functions:

* **Temperature monitoring** – to detect if any cell of the accumulator goes outside of the safe operating temperature
* **Kelly motor controller monitoring** – includes the embedded current and temperature sensors within each controller
  + These values and related errors are read into the central microprocessor
* *If error is detected*:
  + Due to **over voltage**, **over temperature**, or any other fault, the system will safely shut off the high voltage system from any drive state
  + Fulfills Formula Hybrid SAE rules:
    - System cannot be reset by the driver from within the vehicle – must be brought back and reset externally

2.4 STATUS

Status system includes the onboard LCD and wireless RF transceiver:

* **LCD** – displays vehicle speed, power levels, engine rpm, accumulator charge level, drive state, and any errors
* **Wireless RF transceiver** – used to transmit diagnostics to off-track computer
  + Diagnostics – include all driver display data, as well as controller current and temperature levels

2.5 PERFORMANCE

2019 EESD Team Achievements:

* Drive system works, including drive states (wait, precharge, neutral, forward, reverse) and torque vectoring
* Regenerative braking performs well
* LCD display works very well for the status interface
* Precharge, discharge, shutdown sequences, sequences work as required
* AMS shuts down the high voltage system when it should
  + Displays correct error message on LCD
* Thermistors test and confirmed to operate – but NOT installed in the accumulator

2019 EESD Team Shortcomings:

* Information from motor controllers could not be traced to a particular controller
  + Parameters taken, including speed and error messages are sometimes ambiguous
  + LCD speedometer – inaccurate speed displayed, especially when wheels are running at different RPMs
* IC engine feedback loop – electrical noise created by generator and its controller interferes with the PW signal to the throttle servo motor
  + In charging state, servo is not able to be reliably controlled
* Hardware was configured to be good enough to test – requires more work to be rules-compliant
  + Vehicle system assembled on a table, needs to be integrated once the team constructs a frame for the vehicle
  + Each input involved selecting a switch/sensor, testing with breadboard, then adding to the final PCB
  + Each output involved selecting a device where needed
    - Final design has 4 analogue inputs, 1 digital input, and 9 digital outputs from the PIC32
    - 6 SPI, 3 CAN, and 1 UART devices also communicate with PIC
    - Spare pins included on Motherboard PCB to allow additional functionality in the future

3. SYSTEM REQUIREMENTS

2019 EESD team focused on providing an embedded electronic system that controls many mechanical/electrical systems on the vehicle – safely operating and optimizing the vehicle’s performance and providing output data for the driver/team to use during and after competitions.

Design was a microprocessor-based vehicle management system that performs the following functions:

* Controls:
  + Accepts driver inputs of throttle, brake, steering, and switching
  + Engages the rear hub motors in both forward and reverse directions
  + Safely turns on and off the high voltage system
  + Starts the IC engine and toggles the charging state
  + Maintains engine RPM while charging
* Monitoring
  + Continually measures accumulator voltage and temperature
  + Shuts down the high voltage system in the event of an error
* Status
  + Displays vehicle diagnostics to the driver
  + Transmits diagnostics wirelessly to an off-track computer

Competition involves technical inspection – entry vehicles are reviewed for safety/rules compliance. Rules of inspection were used as design constraints in 2020 – refer to the 2020 Formula Hybrid SAE rulebook for a complete list of rules/regulations.

1. AMS
   1. 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
   2. The AMS must monitor all critical voltages and temperatures in the accumulator, as well as the integrity of its voltage/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.
   3. 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.
   4. 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.

**Table 3.1 – AMS Voltage Monitoring**

|  |  |
| --- | --- |
| **Chemistry** | **Maximum number of cells per voltage measurement** |
| PbAcid | 6 |
| NiMh | 6 |
| Lithium-based | 1 |

* 1. 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.

**Table 3.2 – AMS Temperature Monitoring**

|  |  |
| --- | --- |
| **Chemistry** | **Cells Monitored** |
| PbAcid | 5% |
| UltraCap | 10% |
| NiMh | 10% |
| Li-Ion | 30% |

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

1. **Team-designed AMS Board**
   1. Teams may design and built their own Accumulator Management Systems. However, microprocessor-based accumulator management systems are subject to the following restrictions:
      1. The processor must be dedicated to the AMS function only. However, it may communicate with other systems through shared peripherals or other physical links.
      2. 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.
2. **Accumulator – Isolation Relays**
   1. 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.
   2. 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).
   3. 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.
   4. 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
   5. Transient suppressors​ ​ must be added in parallel with the AIR coils. AIRs containing mercury are not permitted.
3. **Precharge**
   1. 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.
   2. 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.
   3. 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.
   4. 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.
4. **Discharge**
   1. 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.
   2. 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.
   3. 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.
5. **Motor Controllers**
   1. 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.
   2. The accelerator control must be a right-foot-operated foot pedal.
   3. 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.
   4. All acceleration control signals (between the accelerator pedal and the motor controller) must have error checking.
   5. For analog acceleration control signals, this error checking must detect open circuit, short to ground and short to sensor power
   6. For digital acceleration control signals, this error checking must detect a loss of communication.
   7. 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.
   8. The accelerator signal limit shutoff may be tested during electrical tech inspection by replicating any of the fault conditions listed in ​ EV3.5.4
   9. 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.
   10. Motor controller inputs that are galvanically isolated from the TSV may be run throughout the vehicle, but must be positively bonded to GLV ground.
   11. 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 SYSTEM THEORY OF OPERATION

Operation of the Hybrid Vehicle is reliant on a system of sensors, microprocessors, and driver inputs to function safely and effectively. The system state machine is implemented on the Motherboard, which uses a PIC32MX795 and handles the majority of the information processing. There is also an ADP for the accumulator, which is a PCB copy of the Motherboard with fewer soldered components that communicates with the Motherboard as an isoSPI slave. The system design takes in the following driver and vehicle inputs:

Inputs to Motherboard:

* Accelerator/throttle pedal position (analog)
* Brake pedal position (analog)
* Steering wheel position
* Engine RPM (digital high/low)
* Motor controller messages (CAN)
* Error messages from ADP (isoSPI)
* GLV and TSV enable, direction, ignition, starter, and charge state switches

Inputs to the ADP:

* Accumulator cell voltages (isoSPI)
* Accumulator cell temperatures (isoSPI)

Outputs:

* Driver Display (LCD 2020-2020, Nextion 2021-2022 with UART)
* Wireless RF signal (with off-track information)
* Left and right hub motor torque for front wheels
* Engine control (start engine, throttle, turn off engine)
* AIR control (ability to close relay)

All inputs (barring AMS dedicated functionality – see Section 4.7 AMS) are processed by a PIC32795, located on the Motherboard PCB designed/assembled by the 2019 team.

General vehicle system operation:

1. The low voltage system is turned on via the GLV switch, bringing up the LCD display and initializing the ​**wait**​ state.
2. The enable switch is toggled, initializing the​ **precharge**​ sequence which matches the voltage of the controller to the accumulator.
3. After 5 seconds of precharging, the system switched to the ​**neutral**​ state from which it can be placed in ​**forward**​ or ​**reverse**​. The motor and generator controllers are powered on.
   1. In ​**forward**​ mode, the throttle pedal moves the wheels forward and the brake pedal engages regenerative braking. The steering wheel controls the differential speed of the left and right wheels.
   2. In ​**reverse**​ mode, the throttle pedal moves the wheels in reverse, and the brake pedal does nothing (no regen needed in reverse).
4. While in ​**neutral**​, the engine can be started by toggling the ignition switch and then flipping the starter switch for 1 second.
5. While the engine is running, the charge enable switch can be toggled between ​**charge** and ​ **idle** states
   1. In the​ **charge**​ state, the engine charges the accumulator. It is automatically throttled to maintain a consistent RPM.
   2. In the **idle**​ state, the throttle control is not engaged, and the engine is ​**idling**​.

In addition to user operation, the system has the following functions:

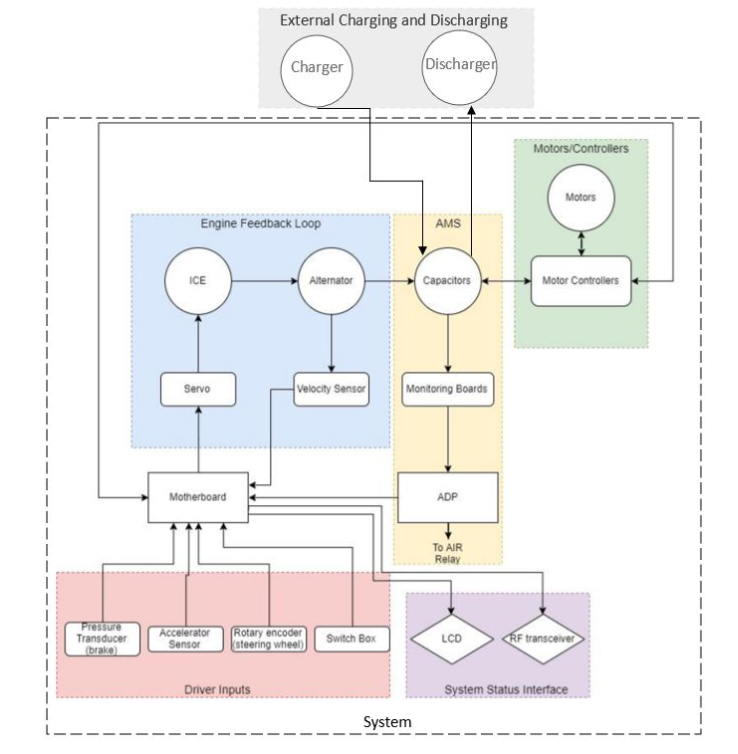
* If there is an over voltage or over temperature in the accumulators, the high voltage system is safely disconnected by a dedicated microprocessor.
* Live vehicle data containing speed, engine rpm, accumulator status, power draining/generating, steering wheel angle, and error messages are displayed on the LCD and sent wirelessly over an RF transceiver to an off-track computer.

4.2 SYSTEM BLOCK DIAGRAMS

Diagram

Description automatically generated

**Figure 4.2.1 – System Block Diagram (2019)**

****

**Figure 4.2.2 – System Block Diagram (2020)**

Diagram, schematic

Description automatically generated

**Figure 4.2.3 – Drive System State Diagram (2019)**

**Diagram

Description automatically generated**

**Figure 4.2.4 – Drive System State Diagram (2020)**

**Diagram

Description automatically generated**

**Figure 4.2.5 – Electrical System Diagram (2021)**

**Diagram

Description automatically generated**

**Figure 4.2.6 – Power Flow Diagram (2021)**

4.3 DRIVER INPUTS

4.3.1 SUBSYSTEM REQUIREMENTS

Driver inputs are used to control speed of motors and develop a torque vectoring control system for the vehicle’s drivetrain. Used to fulfill “accepts driver inputs and displays essential information” requirement.

Drive system software calculates values for the motor controllers to optimize the speed of each wheel – which improves overall handling of the vehicle.

Other component of system is the switch box which implements the process described in Section 4.1 and enables the switching between states described by Figure 4.2.3 and Figure 4.2.4 above.

Driver inputs include:

* Signals from the pedals (throttle and brake going from 0-5 V)
* Angle of the steering wheel – received by the rotary encoder
* Switch box

4.3.2 WIRING SCHEMATICS AND DESCRIPTION

Driver inputs are based on the **throttle and brake pedals** – which are 2 identical Toyota Prius Pedals.

* Pedals connected through an op amp circuit to the Motherboard – ensures the correct level of voltage signal is sent to the Motherboard
  + Op amp circuit output is received by a Motherboard analog input, which then calculates the final throttle value to send to the motor controllers
  + This interaction = the interfacing between the **driver input** and **controller** subsystems
* **Throttle pedal** has 2 inputs to the Motherboard that are averaged to calculate the motor controller throttle value
* **AMT203 absolute rotary encoder**
  + Physically attached to the steering wheel
  + Electrically connected to the Motherboard
    - Update the signals sent through the Motherboard to the motor controllers based on the current wheel position
  + Driver inputs are displayed in Figure 4.3.1 below.

Other physical setup in the driver input subsystem is the **switch box**.

* Controls the powering of the **Motherboard** and the **AMS board**, start and ignition of the **ICE**, the charging sequence, and the direction state of the vehicle
  + All these are implemented as switches on a control panel and their connections are described in Figure 4.3.2.
  + Driver has access to these switches – implementation of their use by the driver is described in Section 4.1.

Diagram, schematic

Description automatically generated

**Figure 4.3.2.1 – Driver Inputs Schematic: Encoder and Pedals**

Diagram, schematic

Description automatically generated

**Figure 4.3.2.2 – Driver Inputs Schematic: Switch Box**

4.3.3. SOFTWARE DESCRIPTION

Software used for implementation of driving and motor control incorporates the pedals, rotary encoder, and switches (mainly the direction switch).

* When system is powered up and precharge sequence completes – the system enters the neutral state based on the software.
* Direction switch in 4.3.2 above can flip between **forward, neutral,** and **reverse** (each with a different control scheme)
  + Neutral state – signals sent to the controllers are set at a minimum, therefore the controllers register no value
  + Forward state – rotary encoder is used to determine angle of steering wheel
    - Encoder is a SPI device, with SPI protocol being configured and initialized in the implementation of this software subsystem
    - Checking the wheel angle occurs with a timer interrupt of 20 μs – complies with the delay between reads noted in the encoder datasheets
* Rest of driver input software configuration involves switching between various states the vehicle could be in – allows it to implement different features from charging to driving to starting the ICE.

The control scheme is based on the position of the steering wheel – read by the rotary encoder

* Angle reading determines the turn state of the vehicle (see Figure 4.3.3)
* Throttle and brake signals are modified by an experimentally determined constant based on the current value of this angle
  + Through experimentation, these constants implement the correct motor speed in each wheel
  + Constitute the torque vectoring to improve handling in turns
* Forward state also implements braking via software
* In the reverse state, the baseline signals from the throttle and brake pedals are sent to the motor controllers

Other aspect of the driver input software implementation includes the switching of states based on the positions of the switches in the switch box.

* Implementation of state diagram is shown in Figure 4.2.2 (above)
* Driving states are a subset of states available in this code configuration
  + Vehicle waits in the **wait state** until precharging is enabled by the enable switch
  + Then, the vehicle goes to the **neutral state** and is prepared to drive (forward/reverse by flipping direction switch) if the capacitors are charged.
  + Ignition/Start switches are included in this software functionality to active the engine and enable autonomous charging.

Diagram

Description automatically generated

**Figure 4.3.3.1 – Rotary Encoder State Diagram**

4.3.4 CHOICE OF COMPONENTS

Toyota Prius pedals

* Selected as they were simple to integrate into the design & inexpensive

Rotary encoder

* Selected for its precision & its style of absolute measurement
* AMT203 reads the value of the position as an angle on a 360-degree circle
  + Allowed for a control scheme design knowing the exact position of the steering wheel and sending this measurement as a signal
  + Better than the other kind of rotary encoder that samples information about the motion/direction of the encoder through incremental measurement

Single throw single pole switches

* Used to simply implement the charging of the states required in the software configuration

4.3.5 FUTURE HYDRAULIC BRAKING

Mechanical installments still need to be made before implementation and transition into hydraulic braking is possible. The 2020 team developed a preliminary system drive diagram and overall system diagram illustrating the integration of hydraulic braking into the vehicle system. The general connections on hydraulic braking implementation are also shown below.

Diagram

Description automatically generated

**Figure 4.3.5.1 – Drive System Diagram with Hydraulic Brakes**

Diagram

Description automatically generated

**Figure 4.3.5.2 – System Block Diagram with Hydraulic Brakes**

**Diagram

Description automatically generated**

**Figure 4.3.5.3 – Hydraulic Brake Connections**

4.3.6 SUBSYSTEM TESTING

Main testing for the driver input subsystem focused on the following:

* Ensuring the state diagram implemented by the software worked properly and integrated with the pedal and encoder signals successfully

Testing was achieved by repeating the switching initialization sequence and noting the states of the relays on the Motherboard impacted by each switch flip.

* Motherboard was powered, the system enabled, and the state of the direction relay was noted when switching between forward and reverse.

Drive control system was tested after this initial testing.

* Testing focused on 2 aspects:
  + Tuning the torque vectoring implemented by the rotary encoder
  + Ensure the direction relay implemented the correct rotation and drive scheme for the different drive states

Testing yielded a complete drive system for the left and right wheel based on:

* Changing states according to switch inputs
* Having the correct signals sent to the motor based on pedal and encoder inputs.

4.4 SYSTEM STATUS INTERFACE

The System Status Interface is designed to provide live updates to the driver and off-track team with a variety of relevant information (in addition to pertinent error messages):

* Fuel level
* Ultracapacitor charge
* Vehicle speed
* Engine RPM
* Pertinent error messages

The primary component of the System Status Interface is the direct updates to the driver via a Driver Display inside the car (LCD 2019-2020, Nextion 2021-2022). The second component is the off-track transmission of these updates via an RF transmitter – which communicates to an off-track RF receiver writing data to a serial monitor. The initial implementation of the RF receiver was created in 2019, while the 2020 team created a preliminary GUI in MATLAB to display the data to an off-track team in a clear way. The 2022 team redesigned the Motherboard to implement a second UART module, to be used in the future to connect both the Nextion Driver Display and an RF transmitter to the PIC32MX795 simultaneously.

A picture containing timeline

Description automatically generated

**Figure 4.4.1 – LCD Wiring Schematic**

4.4.1 LCD (2019-2020)

The LCD screen was used to communicate information on the vehicle’s status to the driver.

* Selected this LCD because it is equipped with a controller – easy communication through SPI
* 7-inch size ideal for a lot of information displayed on one interface
* Available libraries for graphics were intuitive – easily create graphics such as gauge/progress bar
* Supports various forms of RGB interfacing – makes it flexible

LCD visuals updated the driver on various elements of vehicle status.

* Vehicle power gauge – starts in the middle
  + Moves clockwise when net power is positive (draining accumulator)
  + Moves counterclockwise when net power is negative (regenerative braking)
* Capacitor charging state – displayed in a progress bar, with percentage value
* Vehicle RPM – displayed in a progress bar, with percentage value
* Various diagnostics displayed
  + Accumulator temperature, steering wheel angle, accumulator voltage, charge state
* Errors detected by monitoring system are displayed
  + LCD program interprets the CAN bus messages received
  + Each error type allocated to a bit – bit masking identifies which errors have been set, displays corresponding text to screen

4.4.2 NEXTION (2021-2022)

The initial 7-inch LCD display used a SPI interface, and required hundreds of lines of code to incorporate the necessary graphics and suffered with issues of visibility and lack of flexibility.

* 2021 Team chose to replace the LCD with a 10.1-inch Nextion touchscreen below that communicates with the Motherboard via UART
* Advantages over LCD
  + Preload the graphics onto the display with an SD card – simpler and streamlined process for the motherboard code
  + Multiple pages can be implemented – better informational organization for driver and future team development
  + Writing data to the Nextion over UART is more straightforward than writing to the LCD with SPI
  + Many resources available online for support and help with Nextion display implementation

A picture containing text, monitor, electronics, screen

Description automatically generated

**Figure 4.4.2.1 – Nextion Display (10.1 inch)**

* In the 2021 design, the Nextion receives data from the Motherboard and displays the information on two pages: the Primary Display and the Secondary Display (Figure 4.4.2.2 and Figure 4.4.2.3 below)

**Graphical user interface

Description automatically generated**

**Figure 4.4.2.2 – Primary Display Page (2021)**

* The Primary Display is similar to a traditional dashboard display
  + Gives information n vehicle speed, drive state, capacitor bank charge, engine temperature and engine RPM
  + Swapped the RPM display from an indicator of the RPM of the ICE to the RPM of the motors due to lack of ICE functionality

Timeline

Description automatically generated with medium confidence

**Figure 4.4.2.3 – Secondary Display Page (2021)**

* The Secondary Display includes additional information that can be accessed by touching the “INFO” button on the Primary Display
  + Information includes left and right motor RPM, capacitor voltage, and error messages from the motor controllers

The Nextion communicates with the Motherboard through the UART protocol.

* Baud rate of Motherboard and Nextion are currently set to 31250
  + Can be changed to certain presets on the Nextion
* The messages are sent to the Nextion on a timer that triggers once a second
  + Process of sending messages via UART is fairly straightforward – advantage over original implementation of LCD using SPI
* Each textbox or graphic on the display is labeled in the Nextion Editor, where graphics are formatted and loaded onto the display with an SD card
  + To write to a specific box, send “boxname.val=” followed by the desired value to be sent
  + Each instruction must end by sending **0xff** three times, which signals to the Nextion that the instruction is complete
  + Example of writing the speed and RPM values to the Primary Display is shown below in Figure 4
    - “Terminate” function sends **0xff** three times
    - “Send” function sends the string character by character to the Nextion

Text

Description automatically generated

**Figure 4.4.2.4 – Nextion Write Example Code**

4.4.3 RF TRANSCEIVER

The transceiver circuit is used to transmit vehicle diagnostics to an off-track computer via RF.

* Transceiver kit was selected because it used UART
* Range on kit is greater than a kilometer – sufficient for the competition
* RF transceiver is installed close to the LCD display (next to the steering in completed vehicle)
* Off-track computer can receive updates via a serial monitor

4.5 MOTOR AND GENERATOR CONTROLLERS

The Motor and Generator Controllers subsystem deals with input/output signals for the three Kelly controllers.

* CAN message reading, analog throttle and brake signals, generator charging current signal, and forward/reverse switching

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 through CAN messaging.

4.5.1 SUBSYSTEM REQUIREMENTS

The Motor Controller subsystem must manage the controllers for the left and right hub motors and the generator.

* Hub motor controllers –
  + Need two analog signals (from 0-5 V) for throttle and regenerative braking.
  + Need two switches (12 V) to toggle forward and backwards modes
* Generator controller –
  + Needs a signal for charging current (0-5 V)

The Motherboard needs to read signals from all three controllers over a CAN bus.

* Motherboard interprets the message contents and identifies which controller sent the message
* Message contents – RPM, motor and controller temperatures, motor current, controller and switch status, error messages

4.5.2 WIRING SCHEMATIC

Diagram, schematic

Description automatically generated

**Figure 4.5.2.1 – Wiring Schematic**

4.5.3 DESIGN CHOICES

Left and Right Hub Motor Controllers

* Model KLS808I (Kelly Controls, LLC)
* Purchased by the Formula Hybrid club prior to start of 2019 project

Generator Controller

* Model KLS8080IPS (Kelly Controls, LLC)
* Purchased by the Formula Hybrid club prior to start of 2019 project

Interaction between Controllers and Motherboard

* Signals from the accelerator and brake pedals (0-5 V) could have been directly connected to Motor Controllers
* Instead, signals are interpreted by the Motherboard – signal is then sent from Motherboard to the Motor Controllers via 2 out of 3 of the Motherboard DACs, the MCP4922 (Microchip Technology Inc)
  + MCP4922 communicate over SPI, which the team was familiar with
  + MCP4922 were also the highest resolution for their price
* The 3rd DAC is used to turn on the generator controller
* The DACs are written to sequentially
  + Writing starts with the left hub motor controller, then right, then the generator
  + The SPI2 channel is configured for the DACs before writing
  + The SPI2 channel is reconfigured for other SPI2 devices after writing is completed

Forward and Reverse Switches

* These switches are wired to a three-way 12V switch located on the driver-controlled switchboard (see Driver Inputs section 4.3)
* The forward switch for the generator controller is hardwired to 12V, so it always operates in the forward state

Kelly Controller Communication

* The Kelly Controllers are programmed via USB connectors
* When running, communication is only possible via CAN bus
  + CAN – automotive standard in which devices use a single bus to broadcast information, readable by any other device
* PIC32MX795 (Motherboard microcontroller) has two CAN enabled channels
  + However, the PIC32 requires a CAN transceiver to convert the differential signal into CAN transmit and CAN receive signals
  + Team bought a MCP2562 (Microchip Technology, Inc.) transceiver as an SMD for the Motherboard PCB
  + MCP2562 was connected to the 2 Rx and Tx pins on the Motherboard’s PIC32

CAN Bus and CAN Protocol (2019)

* The CAN protocol used for the KLS8080I and KLS8080IPS controllers is based on the SAE J1939 protocol.
* Protocol runs at 250 bps
* All CAN protocols use a filtering process for message handling
  + There are two types of messages that each controllers sends
    - Each message has an extended identifier (EID) and respective data bytes
  + Filter takes messages with a type 1 EID and puts them in FIFO0
  + Filter takes messages with a type 2 EID and puts them in FIFO1
    - Data bytes for each message type are processed separately
  + A control field after the ID segment of the CAN message specifies the number of bytes of data within the message
    - The protocol checks that this control field (DLC byte) is 8 (the byte length for a correctly read CAN message) prior to processing the bytes
* Identifying which controller type is broadcasting a message was determined experimentally
  + No segment of the EID specifies which controller the message came from
  + CAN messages are of the broadcast type – no way to determine which controller’s message will be read in by the FIFO next
  + To deal with message handling, the fifth data byte was used
    - Experimentally determined to be 0x41 for the generator controller and 0x1E for the two hub motor controllers
* Speed of vehicle is determined with averaging
  + RPM values (located in message type 1) are read in for the hub motor controllers
  + Average RPM is calculated by averaging the last two RPM values read
  + Speed is computed using a wheel diameter of 20.5”
* Power consumed/generated is calculated accounting for the brake
  + Brake is determined to be pressed if the value differs from the minimum brake value
  + If brake is pressed, regenerative braking is activated
    - Hub motors are adding current to the system
    - Power generated is calculated as a negative sum of all three current values (two hub motor controllers and the generator controller)
  + If brake is not pressed, power is being consumed
    - Power consumed Is calculated as sum of the hub motor currents, minus the generator controller current
  + “Power” calculated is actually just the current, but the values are proportional
* Errors from the Kelly controllers over the CAN bus are two error bytes
  + Each bit corresponds to one of 15 error messages
  + Program decodes the error MSB and LSB and stores them in a string to be displayed to the driver
* Entire CAN message reading process occurs in a 4ms timer interrupt
  + Timer turns on the CAN, initializes it, reads a message of each type, turns CAN off, interprets the messages, and then resets the timer
  + The reading process is an interrupt, so it is not disrupted by the logic of the broader program (e.g. printing to LCD or the RF transceiver)
  + CAN message reading does not need to happen as often as other processes (e.g. rotary encoder position)

CAN Bus Overview 2020

The amount of wiring needed between I/O devices on a vehicle can be reduced with the CAN Bus protocol.

* Each I/O device is a “node” on the CAN network
  + When a node has information to share, it transmits a CAN message on the CAN bus
  + All other nodes can accept or ignore the CAN message
    - Use buffers and filtering set up in microcontroller
  + No master in CAN Bus – each message is encoded with a priority level (Ex. message about brakes takes priority over message about air conditioning)
    - Arbitration process determines which node transmits its message first
* Low-cost, centralized processing, robustness to EMI contribute to the appeal of CAN protocol for use in automotive engineering

Diagram

Description automatically generated

**Figure 4.5.3.1 – Differential CAN Voltage Signal**

Figure 4.5.3.1 above shows a CAN signal waveform, which operates on a differential voltage signal that sits idle around 2.5 V.

* Node transmits a dominant bit (logic level 0) with the high CAN wire shifting up to 5 V and the low CAN wire shifting down to 0 V
* Recessive (logic level 1) bits are when both wires sit around 2.5 V
* CAN bus is more robust to EMI errors – due to differential voltage signal and increased amplitude of dominant signal

Diagram, schematic

Description automatically generated

**Figure 4.5.3.2 Microcontroller with Two CAN Transceivers**

Since the PIC32MX795 on the Motherboard has two CAN controller modules integrated, only CAN transceivers are needed to implement the system.

* CAN transceiver converts the differential voltage signal into logic level 1s and 0s for the PIC32MX795 to process directly
* TXD pin on transceiver connects to the CAN2TX pin on the PIC32 – transceiver is an intermediary, not a device communicating with the Motherboard via UART

CAN messages consist of a 29-bit Extended Identifier (EID) that determines a CAN message’s priority and source address.

* After the EID, there is a field for data called DLC
* After the DLC, the transmitting node sends the data (usually 8 bytes)
* Finally, 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 modules decides whether to accept or ignore the message

* CAN module on Motherboard can have up to 32 acceptance filters and 4 masks
* First, receive message acceptance buffer (RMAB)
* Masks determine which bits of the message the filter will pay attention to and which bits the filter will ignore
* The filters will compare the selected bit with preset values
  + Ex. Message 1 has ID 0x0CF11E05 and Message 2 has ID 0x0CF11F05
  + Mask bits are all set to 1 – no bit of the message ID should be ignored
  + Filter bits encode the values of the message ID
    - 0b110011110000 for the standard identifier (SID)
    - 0b010001111000000101 for the extended identifier (EID)
  + After filtering, messages are stored in the CAN Message FIFOs
    - Stores a maximum of 1024 messages per module
  + Figure 4.5.3.3 below shows how accepted messages move to the system bus where the CPU of the microprocessor stores them into the FIFOs in the RAM of the microcontroller

Table

Description automatically generated

**Figure 4.5.3.3 – CAN Module FIFO Setup**

CAN Bus Progress (2021)

* Improved the CAN network and successfully implemented the differentiation
  + Reflashed the Kelly controllers and used a Saleae logic analyzer to read their corresponding extended identifier bits (EID)
    - EID bits in the CAN Bus protocol allow identification of individual devices
  + Using two filters – matching the EID bits of the right or left controller respectively, CAN messages can be read from both controllers
    - Once read, the messages are stored in four FIFO message buffers
      * FIFO0 and FIFO1 are reserved for the right controller
      * FIFO2 and FIFO3 are reserved for the left controller
  + Finally, the messages are decoded in the FIFO buffers and CAN messages are reported for display in the System Status Interface
* For future team reference, the following settings were used in the Kelly controllers user program:

Graphical user interface, application

Description automatically generated

**Figure 4.5.3.4 – Kelly Controller Settings**

4.5.4 PROGRAMMING STATE DIAGRAMS

Diagram

Description automatically generated

**Figure 4.5.4.1 – CAN Message Read State Diagram**

4.6 ENGINE FEEDBACK LOOP

The Engine Feedback Loop subsystem is the feedback loop that adjusts the output of the internal combustion engine (ICE).

* The ICE powers the generator – generator used to supply power to charge the ultracapacitors of the accumulator – ultracapacitors used to supply voltage to the vehicle’s drive system
* Feedback loop includes the ICE, a servo motor used to physically adjust the angle of the engine throttle, and a sensor to monitor the throttle value set by the servo and compare it to the desired throttle value at the time
* Loop sets the ICE RPM to the desired value to optimize the capacitor charging

4.6.1 SUBSYSTEM REQUIREMENTS

To control the speed of the engine and thus generator charging, the throttle must be modulated to deliver the correct amount of fuel for a given mechanical load.

* A servo is used to electronically open/close the throttle on demand
* A signal representing the current engine speed is necessary for throttle adjustments
* The system must compensate for changes in load so engine RPM remains stable
  + Transient response must be quick enough to avoid engine stalling
  + Excessive overshoot when changing the RPM is unacceptable –engine’s upper limit speed is 14,000 RPM
* Control algorithm must allow the engine to run at idle speed when capacitor charging is not required
* User needs to be able to bypass system to warm up the engine or perform testing

4.6.2 WIRING SCHEMATIC

Diagram, schematic

Description automatically generated

**Figure 4.6.2.1 – Wiring Schematic**

4.6.3 DESIGN CHOICES

The throttle servo was chosen to be small, but able supply the torque needed to counteract the throttle return spring.

* Servo must be able to hold open throttle without overheating/damaging its gears
  + However, throttle return spring must be able to overcome the servo (return the throttle to its fully closed position when the servo is off)
  + Hitec D485HW servo was selected to meet these requirements
* Servo must be able to run off of 12 V supply
  + The D485HW does not, so a 12 V to 6 V DC-DC converter was used
    - Converter also provides ground isolation between the ICE 12 V starting/ignition system and the microcontroller 12 V system

Monitoring the engine RPM required obtaining a signal representing the speed of a rotating shaft.

* Multiple methods are possible
  + Hall effect sensors – require a magnet on the shaft
  + Rotary encoders – require physical attachment to the shaft
  + Optical encoders – not possible on an oil-submersed crankshaft
* Selected method – using existing signal from engine’s alternator
  + Alternator provides a three-phase sinusoidal input – varies in frequency proportionally to the speed of the crankshaft
* Alternator signal is not directly compatible with microcontroller
  + Needs to be converted into a logic-level (3.3 V) square wave with the same frequency as the original signal
  + Several challenges associated with this approach
    - Switching noise is present in the sine waves
    - Alternator output has variable noise (range of 16 V to 75 V)
* Alternator signal is passed through the NCV1124 (Variable Reluctance Sensor Interface IC)
  + Designed to address similar concerns in automotive applications
    - Usually have an inductive sensor giving a periodic output that is passed to a microcontroller
  + IC has two stages
    - Active clamping circuit reduces input voltage to logic level (3.3 V)
    - Comparator circuit converts the sinusoidal input into a square wave
  + To reduce the amount of high frequency noise entering the NCV1124, an RC filter with a cutoff equal to the maximum expected frequency is placed before the input
* Alternator signal lacks a neutral lead, which posed an additional challenge
  + Alternator output is a three phase AC signal
  + NCV1124 requires a single phase referenced to ground
    - Solution: “floating ground” created by connecting three resistors across the phases in a wye configuration
    - Resistors must be accounted for when designing the RC filter

Addressing EMI on Feedback Loop

* The 2019 team encountered problems with electrical noise on the PWM signal from the Motherboard to the Servo Motor
  + 2020 team planned to address this issue, but was unable to due to COVID

Adjusting PID controller RPM setpoint

* In the future, software could be implemented that would change the RPM setpoint of the PID controller based on the voltage of the capacitors
  + Currently, the Feedback Loop is designed to operate at maximum ICE efficiency
  + Changes could be made to have the RPM setpoint increase at low accumulator charge and decrease a high accumulator charge
  + System diagram is shown below

Diagram

Description automatically generated

**Figure 4.6.3.1 – Planned ICE Control System**

4.6.4 STATE DIAGRAM

The theory of operation for the engine speed PID controller is shown below.

* Difference between desired RPM and actual RPM is calculated
  + Error is multiplied by K­op­, the proportional gain
  + Error is integrated over time and multiplied by Ki­, eliminating steady state error
  + Error’s derivative is calculated, which allows the system to smooth out oscillations and transients
* The three error calculations are combined to produce the output signal sent to the servo to make the desired adjustment to RPM

Diagram

Description automatically generated

**Figure 4.6.4.1 – State Diagram**

4.7 ACCUMULATOR MANAGEMENT SYSTEM

The Accumulator Management System (AMS) is composed of an ADP and four daisy-chained demo boards, which together handle the monitoring of the ultracapacitor cell voltages and the cell temperatures.

4.7.1 SUBSYSTEM REQUIREMENTS

* The ADP board is a mirror of the Motherboard, but the PCB has fewer components and two isoSPI translators instead of one
  + The two isoSPI translators serve different purposes
    - One of the isoSPI translators is used as a master to communicate with the demo boards
    - The second isoSPI is used as a slave to communicate with the Motherboard based on Motherboard requests
  + The AMS will shut down if the temperature is above the maximum rating or an overvoltage is detected, sending an error to the Motherboard
    - This protects the driver and board components
* The LTC6812 demo boards were selected for their capabilities in monitoring the accumulator
  + AMS needed to monitor the voltage of all cells and temperature of 10% of cells
  + Each demo board can measure 15 cells, as well as 9 temperature sensors
  + With 4 demo boards purchased, all cells could be monitored
    - Additionally, demo boards can detect over voltages, under voltages, and balance the cells
    - Demo boards are built around the LTC6812-1 multi-cell battery monitoring IC
  + AMS needed to be isolated from the rest of the system, achieved through the demo boards, ADP, and the Motherboard all communicating via isoSPI
    - Conversion between standard SPI and isoSPI is accomplished by the LTC6820
    - isoSPI signals are boosted to a higher voltage for communication between boards by the HM2100NL

Diagram

Description automatically generated

**Figure 4.7.1.1 – Signal Flow in AMS**

* The Motherboard receives information from the AMS by communicating to the ADP
  + The ADP writes over-voltage values to the demo boards, then continuously writes a command to begin reading cell voltages and checking the over-voltage flags set by the demo boards
  + Process continues until an over-voltage flag is set, at which point the AIR is opened and an error is sent to the Motherboard

Diagram

Description automatically generated

**Figure 4.7.1.2 – ADP Software Flow Diagram**

Diagram

Description automatically generated

**Figure 4.7.1.3 – AMS and Motherboard Connectivity Diagram**

Temperature Sensing

* The thermistors the Hybrid Club has are Negative Temperature Coefficient (NTC) thermistors
  + Thermistor resistance decreases with increasing temperature
  + The 2020 team searched for a resistance-temperature curve or equation for the NTC thermistors
    - Part number was found to be **TH310J39G**
    - Determined the temperature tolerance or B value that was used to calculate the resistance of the thermistor at 25 degrees Celsius

**(Equation 4.7.1.1)**

* + Where R50 was 10 kOhms and B=3933 from the Thermistor Usage document
  + Using Equation 4.7.1.1 and the list of datasheets in the resistance-temperature pdf for the Thermistor Usage document, the 2020 Team verified their resistance-temperature relation

Table

Description automatically generated

**Figure 4.7.1.4 – Thermistor Temperature Table and Equation**

* Due to COVID, the 2020 team was not able to implement temperature sensing

**Figure 4.7.1.4 – Validation of Thermistor Temperature Range and Resistance Relation**

Capacitor Charging and Discharging

* 2019 team encountered problems with inconsistent discharge patterns among individual capacitor cells
  + Led to overvoltage errors caused by individual capacitors that were at much higher-than-average voltages
  + Required passive or active balancing
    - Passive balancing was determined to be too slow by the 2020 team
    - It was decided to implement active cell balancing
  + Original 2019 plan was having a built-in system for charging and discharging, but errors in functionality drove the development of external charging/discharging
* After following 2019 startup guidelines, nonsensical voltage and temperature values appeared on the Driver Display and the AIR opened, disabling the high voltage system
  + Drove the 2020 team to create detailed documentation on the AMS subsystem, as the 2019 documentation was not adequate
* External charging/discharging was developed by the 2020 Team
  + Charges with a constant 15 mA (relatively safe) current
  + Wall-connected variable operational amplifier is connected to a transformer in series, then connected to accumulator capacitors using jumper cables
  + Multimeters were used to read current, DC voltage, and RMS voltage to maintain 15 ± 5 mA of current throughout charging
  + Full charge took 40 minutes – higher current would be faster, but not as safe
* **Note: Sparks will occur during initial connection of charge and discharge**
  + Discharging uses a 1000 W rated 47 Ohm power resistor
  + Reasonable discharging time was calculated as follows:
    - Assume total energy from 160 Volts (2.7 V \* 60 capacitors) and 50F Capacitors = CV2/2 = 640 kJ
    - By implementing a 47 Ohm resistor in a series circuit with 160 Volts, current should be 3.404 A
    - Power going into resistor given by P = IV = 3.404 \* 160 = 6444.680 Watts = 544 J/s
    - Assuming capacitor linear discharge, average discharge rate given as total joules discharged per volt = (P\*V/2)/(V) = 272 J/s
    - In theory, 640,000 [J] / 272 [J/s] = 2352 [s] = 39.22 minutes for a full discharge
  + In practice, discharging from 150 V to 55 V (95V difference) took around 23 minutes
  + A diagram of the discharging circuit with maximum accumulator charge and current is shown. Connections to the accumulator were made with jumper cables.

Diagram, schematic

Description automatically generated

**4.7.1.5 – Capacitor Discharge Circuit**

4.7.2 WIRING DIAGRAM

Diagram, schematic

Description automatically generated

**Figure 4.7.1 – Wiring Diagram**

4.7.3 STATE DIAGRAM

Diagram

Description automatically generated

5 SYSTEM INTEGRATION TESTING

5.1 SYSTEM INTEGRATION OVERVIEW

The subsystems comprising the electrical system of the hybrid car were designed and tested individually before being integrated into a full system (on both the hardware and software side).

Software integration required several considerations.

* Timing was a major concern
  + Time sensitive processes must all function without interruption
  + CAN message reading, rotary encoder, and servo functions were put in separate timers
    - Thus, all were able to operate without being bogged down by slow UART and LCD printing
* Device channel sharing was also considered
  + Four different devices use SPI 2, so it was important to make sure that the configuration bits for each SPI function were correctly set at the time at which the functions were called
  + The Saleae Logic analyzer was useful to ensure that the correct information was being sent to the correct pins

Hardware integration centered around the Motherboard PCB.

* The Motherboard PCB contains the circuitry for most of the input and output devices
* After each subsystem was wired into the integrated system, the subsystems were tested individually and then added to the final program one-by-one.

Final testing consisted of running the full process from start-to-finish. Section 6.2 describes the step-by-step process used to test the system.

5.2 SYSTEM REQUIREMENT CHECK

The following requirements were checked using the described test process in 2019 and were found to have the following results.

|  |  |
| --- | --- |
| Requirement | Test |
| Accepts driver inputs of throttle, brake, steering, and switching | Functionality verified for all switches. Brake and throttle inputs measured on Saleae logic analyzer. Steering angle measured and outputted to LCD display. |
| Engages the rear hub motors in both forward and reverse directions. | Verified throttle pedal functions in forward and reverse modes, brake pedal functions in forward mode, and neither pedal engages motors in neutral or any other states. |
| Safely turns on and off the high voltage system | Verified that the high voltage enabled switch triggers wait state for 5 seconds, then switches to neutral state. Relays were verified to be functioning. |
| Starts the IC engine and toggles the charging state | Verified that ignition and starter switches turn on engine, charge enable switch engages generator controller |
| Maintains engine rpm while charging | Engine throttle servo verified when not charging, but failed during charging state due to electrical noise |
| Continually measures accumulator voltage | Verified voltage levels in accumulator match those read by the system. Temperature measurements tested on one thermistor, but not connected to the accumulator |
| Continually measures accumulator temperature | Temperature measurements tested on one thermistor, but not connected to accumulator |
| Shuts down the high voltage system in the event of an error | Shorted accumulator during operation and verified system shutdown |
| Displays vehicle diagnostics to the driver | LCD functionality demonstrated |
| Transmits diagnostics wirelessly to an off-track computer | RF transceiver tested individually and integrated with full system |

6 USER’S MANUAL & INSTALLATION GUIDE

6.1 INSTALLATION

The final design for this project includes a functioning drive system and electrical subsystems for the Formula SAE Racing Team to install in their future overall vehicle design.

* The prototype functions as a test bench from which other software features and electrical components can be added to optimize the system
* The installment of this system into a vehicle body must consider proximity needs
  + Subsystems will need to be mounted in close proximity where electrical connections exist
  + Proximity is particularly important for the devices using serial communication protocols used for the various devices connected to the Motherboard
* Installment must consider accessibility for the human driver
  + The steering wheel and pedals will be situated in the traditional locations of a car
  + The throttle and brake must be mounted by the right foot and the wheel at arm level when the driver is sitting
  + Also, the Switch Box (implementing the state diagram controlling device start-up) must be mounted by the steering wheel
  + The LCD must be visible for viewing at all times
    - All the above devices need power, routed through the Motherboard
    - Therefore, the Motherboard and AMS board should be installed in the cockpit of the vehicle as well
* Other electrical subsystems will be installed depending on the final vehicle body manufactured by the SAE Formula Hybrid club
  + The ICE feedback loop, the motor controllers, and the motors themselves will be installed into the frame, pending a final design

6.2 SETUP, VERIFICATION, AND TROUBLESHOOTING

The system initialization process, signs of errors in the system, and troubleshooting approaches are described below.

1. Connect all components (peripheral devices and boards)
   1. Connect all devices that run off the Motherboard to their designated connector
   2. Each connector is noted by name (“Encoder”, “Throttle”, “L\_DAC”)
   3. Each connector is wired to a female connector housing to be attached to the male pins on the board
   4. Connect a 12V supply or battery to the Motherboard and AMS board using the red/black leads on the left side of the car (front is determined by front wheels of vehicle)
2. Turn on main power and upload code
   1. Flip the leftmost switch on the switch control box labelled “MTHRBRD PWR” to establish connection of the 12V source to the Motherboard and AMS Board.
   2. Upload Motherboard software using the PickIt3
      1. After uploading, the Driver Display should turn on
      2. If the Display does not turn on, cycle the power by flipping the “MTHRBRD PWR” switch off and then on again
   3. The Driver Display will display diagnostics
   4. To troubleshoot errors, note the error description
      1. Check the appropriate connection or system component described
         1. Overvoltage on a capacitor
         2. The throttle pedal being out of range
         3. The temperature of a capacitor or controller being too high
3. Turn on the Enable switch (2nd from left) and listen closely
   1. The Precharge and Discharge relays should click on
   2. After 5 seconds the main relay should click on and the car should be in the “Neutral” State
      1. See Step #2 if there are more than three clicks
   3. To test the functionality of the direction states after the enable functionality is confirmed:
      1. Turn **off** the Enable switch
      2. Turn **on** the large red switch on the ultracapacitor housing (WARNING: this activates the high voltage capabilities of the system)
      3. Turn **on** the Enable switch
      4. At this point, pressing the throttle and brake pedals should have no response from the motors
4. Once the “Neutral” state has been confirmed to be functional, test the other drive system directions
   1. Flip the Direction switch to “FWD”
      1. Pressing the throttle should spin the motors
      2. Turning the test steering wheel to a set position and then running the motors below full throttle should result in one motor turning faster than the other
      3. If no change is noted, check the connection of the rotary encoder to the Motherboard
      4. To test the regenerative braking, run motors at full throttle and then press the brake pedal
         1. The motors should come to a halt immediately, rather than slowly stopping
   2. Flip the Direction switch to “REV”
      1. Note the sound of the direction relay clicking
      2. Press the throttle pedal and then confirm that the wheels do spin in the other direction
   3. If this behavior is not observed, check the software to troubleshoot
5. Test the Start and Ignition switches:
   1. WARNING: this testing is very loud and must be tested in an outdoor open space
   2. Flip the Start and then Ignition switch to turn on the ICE
      1. Testing the functionality of the ICE can be done by monitoring the RPM
      2. The RPM PID should be setting the throttle angle based on the servo attached to the throttle
         1. Troubleshooting can be achieved by turning the servo to the correct angle and noting whether this makes sense for the RPM readout
6. Test the Charge Enable switch:
   1. Now, the ICE should be running correctly, monitored, and adjusted by the servo to optimize RPM
   2. Flip on the rightmost switch labeled “Charge EN” to enable the charging capability of the ICE to the ultracapacitors
      1. Attaching a DMM across the leads of the ultracapacitors after a short time should show the capacitors are being charged

Testing CAN Signals from the Kelly Motor Controllers (2020)

1. Apply power to the controller (detached from the vehicle system) using a12 V DC power supply (GEMTECH in Hybrid Lab) by attaching leads to the Black GND and Pink PWR (7) pins on the connector
   1. Determine the correct connector to use (out of 3 possibilities) by matching the shape of the plastic housing on the connector and the colors of the wires attached to the pins
   2. 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 “MTHRBRD PWR” switch
   3. If controllers are properly attached to the car, this should provide 12 V to the controllers.

Diagram

Description automatically generated

**Figure 6.2.1 – Wiring of Kelly Motor Controllers**

1. To verify that a signal is being transmitted, use the Saleae Logic Analyzer for decoding by attaching the CAN H or CAN L wires and configuring the CAN analyzer settings.
   1. For more extensive debugging, it is recommended to use the Keysight Infiniti Vision scope
2. To set up CAN on the oscilloscope, specify the trigger level (1.86 V) and the bit rate (250 kbps)
3. Chose the CAN\_H or CAN\_L to model – both add to the differential signal, but only one is needed when examining raw data
4. The 2020 team used the analog inputs on the Keysight scope, with a photo reference seen in Figure 6.2.2 below.
5. To analyze the data, compare the ID and ATA fields to the Kelly CAN Protocol Description.
   1. The third data entry, 0x1E, signifies the node is a motor controller, not a generator controller
   2. The generator controller has 0x41 in this position

A picture containing text, indoor, monitor

Description automatically generated

**Figure 6.2.2 – Keysight InfinitiVision Oscilloscope Reading**

To use the Kelly controller software (2020):

1. Attach the RS-232/USB converter to the rectangular connector on the controller and a PC
2. Download the KMC User App.exe
   1. Software should work when controllers are plugged in and powered on
   2. If errors occur, see the 2020 CAN Operation Guide for debugging tips
      1. Contains a user interface with real-time CAN data

Setting up Off-Track Computer (2020):

1. Confirm the BRG value in the MPLAB X code, confirm it is accurate
2. Make sure the code has been loaded to the Motherboard
3. The folder **serial\_GUI** should be downloaded to 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 (2020 default was 9600)
8. Select the port
9. Hit the Receive Button
10. If there is no data displayed, hit the receive button again

Graphical user interface, application, Word

Description automatically generated

**Figure 6.2.3 – GUI for Off-Track Computer**

Manuals for Future Teams (2021)

* Primary goal of the team was to create documentation to reduce the startup time with times and allow teams to jump into development more quickly
  + **Start-Up Guide**:
    - How to turn things on
    - Powering the low voltage system and activating 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 TO-RACE DESIGN CHANGES

7.1 DRIVER INPUTS

To improve system reliability, error margins should be included in the minimum and maximum acceptable values of Throttle 1 and Throttle 2.

* These limits are in place to detect hardware malfunctions, such as a signal shorted to power or ground
* After the system is initialized, the throttle/brake signals falling outside their nominal range will trigger faults and shut down the system
  + Therefore, small variations in supply voltage and any electrical noise are likely to trigger a false fault condition
  + Solution (for the future): add error margins to the minimum/maximum values
    - E.g. a fault condition is triggered only by a throttle value that is more than 0.2 V outside the nominal range
    - Reduce the change of a false error while still allowing the system to detect open circuits, shorts, and other hardware failures

The brake in the prototype uses a second Prius accelerator pedal for its signal.

* In the final vehicle, the brake should be a hydraulic pressure transducer
* This transducer will convert the hydraulic pressure in the mechanical braking system into a 0-5V analog signal
  + To accommodate this change, the minimum/maximum signal threshold must be adjusted

The brake signal gain factor should be added to the software and tuned according to physical testing

* The pressure transducer (at its max 5V signal) is operating at a much high pressure than needed in normal driving conditions
* To realize 100% regeneration capacity during normal driving, the gain factor must be adjusted so that full regeneration corresponds to a realistic pedal pressure and amount of mechanical braking
  + Caution: regenerative braking should not exceed mechanical braking to the point that the rear tires begin to slide before the front tires (this will cause the vehicle to spin)

Vehicle performance can be improved by implementing torque vectoring during regenerative braking

* Since heavy braking is utilized at corner entry (turning during race), torque vectoring during regenerative braking needs to increase cornering ability in the same way that torque vectoring occurs during acceleration
  + Caution: torque vectoring must be tuned to make sure that the rear tires do not lock and cause a loss of control
* Physical testing with the full vehicle will be needed to determine proper values for brake gain and torque vectoring constants
  + Torque vectoring for braking/acceleration should be tuned independently – will be different values
  + Note: since weight transfers to rear tires during acceleration, the amount of torque bias on acceleration can be much greater during braking

7.2 SYSTEM STATUS INTERFACE

(Suggestions from 2019 Team)

Utility of LCD can be increased by improving visibility for the driver

* A higher 3.3 V supply will allow increased brightness settings, as configured in the software
* A high contrast color scheme and glare reducing scheme will improve visibility

LCD can be changed to include extra data and clarify error messages

* Individual motor and controller temperatures
* Engine coolant temperatures – monitored by connecting the engine’s factory temperature to one of the spare analog inputs on the Motherboard
* Improve error format – ensure that multiple errors at the same time do not appear on top of each other/overflow off the screen
* Controller error messages improved to identify which controller sent the message

RF data system sends system status information in the form of a comma separated list

* To interpret this data, create a GUI that displays the values in real time
* Create a data log which automatically saves the information for retrieval later

7.3 MOTOR CONTROLLERS/MOTORS/GENERATORS

In order to achieve accurate information about motor speed, current, and controller errors, a method for determining the source of a CAN message needs to be designed and implemented

* Current configuration – each CAN message appears in the same format
  + Makes it difficult to determine which controller sent it
  + Without this info, vehicle speed and accumulator current cannot be accurately determined
* 2019 research indicated that the Kelly controllers do not send identifying information in either of the two possible CAN messages
  + This needs to be verified by utilizing the CAN interpreting function of the oscilloscope while changing the “CAN preferred address” parameter in the Kelly Controller GUI
* If it is determined that no addressing is available, there are other possible solutions to determine message source
  + The CAN messages include a series of bits marked “reserved” in the datasheet, which may allow the receiver to interpret which controller sent the message
  + Message 2 could be used for ID by checking the “Switch Status” bits
    - An unused switch could be set high on one controller and low on another to differentiate the left and right controllers
    - The generator controller is unique in that the Hall A, Hall B, and Hall C bits are simultaneously zero at all times
      * The drive motor controllers never have all three Hall sensors set to zero
  + These methods are dependent on the ability of the receiver to associate Message 1 (which contains speed and current data) with the corresponding Message 2 (which contains the switch and Hall states) from the same sender

To produce full output from the motors and generator, a liquid cooling system will be necessary

* System will require a water pump and radiator, as well as auxiliary fans to cool the radiator when airflow is insufficient
  + Water pump and auxiliary fans can be controlled by the spare relay outputs or digital pins in the Motherboard
* Future team members – add a liquid cooling system for the controllers and accumulator as well
  + System would require chill plates (heat exchangers which bolt to a flat surface) affixed to each device
* Temperature information for thermostatic control is available from the internal sensors in the motors, generator, and controllers – as well as thermistors in the AMS

7.4 ENGINE FEEDBACK LOOP

The engine feedback loop suffers from interference issues caused by the EMI generated by the Kelly Controllers and motors.

* When the tractive system is off, the motherboard is able to effectively regulate engine RPM by using a PID controller
* When tractive system is on, EMI causes the servo to respond incorrectly to the messages being sent – Motherboard is unable to able to accurately measure current engine RPM at times
* Proper shielding against electrical noise is needed for both the servo and engine RPM signals to solve EMI issues

To increase reliability of engine RPM measurements, a dedicated PCB should be developed for the NCV1124 chip and associated circuit

* Circuit is located away from the Motherboard to avoid introducing electrical noise from the alternator to other circuits
* Circuit is currently assembled on a prototype breadboard – would benefit from a PCB with a convenient mounting solution and protection from outside exposure

Once EMI concerns are rectified, the PID controller for the engine RPM should be optimized for performance under loading from the generator

* The team was unable to operate the RPM control and generator simultaneously – the PID was tuned to work ONLY at a no-load condition
* The PID constants will need to be adjusted for optimal performance with the generator at maximum capacity

Future team members may also consider implementing an algorithm that utilizes multiple power output levels

* Adjusts so the engine runs at a more efficient speed when the accumulator is mostly charged
* But engine increases to maximum power output when the accumulator is nearly depleted
* Potentially make use of information about the racetrack layout to optimize the charging and discharging cycle

7.5 ACCUMULATOR MANAGEMENT SYSTEM

The AMS can be made more robust by eliminating the problem of low voltage shutdowns

* Current configuration has monitor boards powered by a discrete transistor driven by the management IC – provides the 5V regulated supply they need to run
  + If voltage across all cells monitored (15 total in 2019) drops below 11V, the board shuts down and triggers an AMS error
  + Error prevents the high voltage system from operating --- cells cannot be charged by the generator
    - Simple fix: implement software that prevents the drive motors from draining the accumulator below 44V (11V x 4 boards)
    - Better solution: replace the single transistor power supply with a high voltage low dropout regulator
      * Would allow cells to be drained to just above the 5V regulated voltage

To improve the functionality of the AMS, active cell balancing could be implemented using the native functionality of the battery monitor ICs

* Each battery monitor IC is capable of driving an external transistor between each pair of cells, which can transfer charge when cells are out of balance
* Active balancing can support currents of 1-2 A
  + Allows for a more proactive balancing approach than the Maxwell balancing boards currently in place
  + Maxwell balancing boards only drain a cell if its voltage limit is exceed
  + Active balancing of the monitor ICs would allow the cells to be balanced among themselves at any voltage
    - Would reduce the risk of having one cell charged substantially more than the rest – would make it prone to exceeding its limit during high current charging

If active balancing is implemented, the current limiting resistors located between each cell junction and the sensing leads for the monitoring boards would need to be replaced with fuses

* The existing limiting resistors limit current to a safe level if one of the sense leads becomes shorted to another cell or ground
* Can only function correctly when current in the sense leads is minimal – which is true during voltage measurements
* If balance currents are added, the resistance in this circuit is no longer acceptable
* Fuses and fuse holders must be selected to meet voltage, current, and packaging requirements

8 CONCLUSIONS

The team completed and demonstrated the first functioning implementation of the monitoring and drive system for the Notre Dame Formula SAE Hybrid team in 2019.

* Team was able to meet most of the system requirements
  + Requirement of accepting driver inputs, processing these into signals for the motor controllers, and display essential information about the state of the system
  + Requirement of creating monitoring system for the motor controllers
    - Still need to add temperature monitoring with the thermistors
  + Requirement of managing the ICE and energy storage
    - Still need to handle the noise of the signals impacting the throttle control servo
* Tested system functionality in an integrated format
* Provided specifications for future areas of needed improvement

The project will need further development and improvement as integration with the overall vehicle design continues.

* Subsystems like the ICE feedback loop and the motor controllers will need to be integrated
  + Integration will occur after physical and mechanical systems are designed and constructed by the Formula Hybrid SAE club

9 APPENDICES

9.1 REFERENCES AND DATA SHEETS

**2019 References**

Formula Hybrid 2020 Rules: <https://formula-hybrid.org/students/rules-and-deadlines/>

Rotary Encoder: <https://www.cui.com/product/resource/amt20.pdf>

RF Transceiver:

<http://statics3.seeedstudio.com/assets/file/bazaar/product/HC-12_english_datasheets.pdf>

DAC: <http://ww1.microchip.com/downloads/en/devicedoc/22250a.pdf>

Monitoring Board:

<https://www.analog.com/media/en/dsp-documentation/evaluation-kit-manuals/DC2350AF.PDF>

Monitoring Board:

<https://www.analog.com/media/en/technical-documentation/data-sheets/LTC6812-1.pdf>

CAN Transceiver: <http://ww1.microchip.com/downloads/en/devicedoc/20005167c.pdf>

Dual Variable Reluctance Chip:

<https://www.onsemi.com/pub/Collateral/NCV1124%20DATA%20SHEET.PDF>

Kelly Controller:

<https://kellycontroller.com/wp-content/uploads/kls-8080i-ips/KLS8080I-IPS-Opto-isolated-Sinusoidal-BLDC-V1.10.pdf>

Kelly Controller CAN Protocol:

<https://kellycontroller.com/wp-content/uploads/kls-8080i-ips/Sinusoidal-Wave-Controller-KLS-D-8080I-8080IPS-Broadcast-CAN-Protocol.pdf>

**2020 References**

PIC32MX795F512H: [Datasheet](http://ww1.microchip.com/downloads/en/DeviceDoc/PIC32MX5XX6XX7XX_Family)Datasheet_DS60001156K.pdf)

**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](https://www.mouser.com/datasheet/2/609/dc2026cfe-1268897.pdf)

Power Resistor: [Datasheet](https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=9-1773453-2&DocType=DS&DocLang=English)

Temperature Resistance Curve: [Datasheet](https://www.mouser.com/datasheet/2/18/AAS-920-320E-Thermometrics-NTC-Diode-082317-web-1315825.pdf)

References for CAN/Motor Controllers:

Kelly Controllers (motors: KLS 14401-8080I, generator: KLS 14401-8080IPS) [Kelly KLS8080I/IPS Motor Controller User's Manual](https://kellycontroller.com/wp-content/uploads/kls-8080i-ips/KLS8080I-IPS-Opto-isolated-Sinusoidal-BLDC-V1.10.pdf)

Kelly Controller Specific CAN Protocol Description [Sinusoidal Wave Controller KLS Broadcast CAN Protocol](https://kellycontroller.com/wp-content/uploads/kls-8080i-ips/Sinusoidal-Wave-Controller-KLS-D-8080I-8080IPS-Broadcast-CAN-Protocol.pdf)

Kelly Controller Software Download (we used PC version KMC User App.exe) [Download Link](https://www.kellycontroller.com/support/)

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#](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>

**2021 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>