Alice Ma
Braeden Hamson
Jade Nguyen
Pushpesh Sharma
ECE 412 Senior Project Dev
Winter 2023

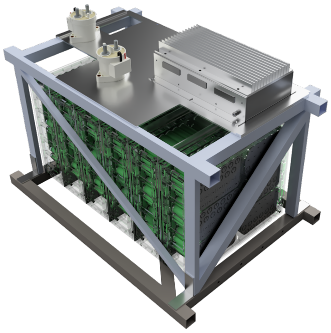
Proposal Outline and Discussion

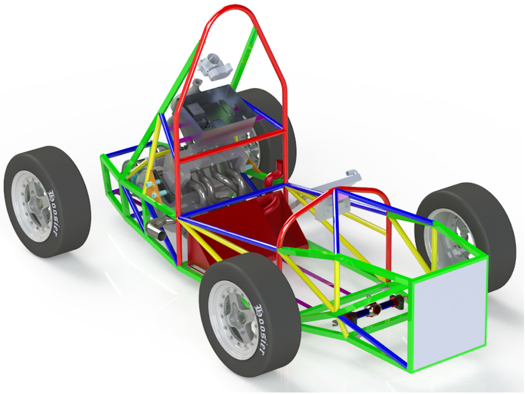
1. Executive Summary

Viking Motorsports is an EV racing team at PSU. They are a student organization and compete in the Formula SAE competition against other schools nationally. Their overall goal is to design, build and race a car every year. The electrical systems are complex, there are many systems that must work in harmony with each other. There are also many possible single point failures that can occur that will end our race.

Understanding this complex system is difficult, all of the information is locked away behind body panels and wire insulation. The motivation of this project is to tap into that data to make it easily accessible, this will enable rapid diagnostics and debugging. Additionally valuable data is encoded in CAN by various off the shelf systems within the car. Accessing this data will also be beneficial.

We want to deliver a network system that uses CAN to communicate system analytics from the car's batteries and several other measured components and relay that information to the driver's display screen. The analytics will help troubleshoot any faults and errors in the operation of the vehicle and allow for faster fixes on the race track. The system overall will consist of multiple PCBs mounted in various locations on the vehicle to allow for sensor wires to relay the data to be sent from each component to a data analyzer then get displayed on the screen.

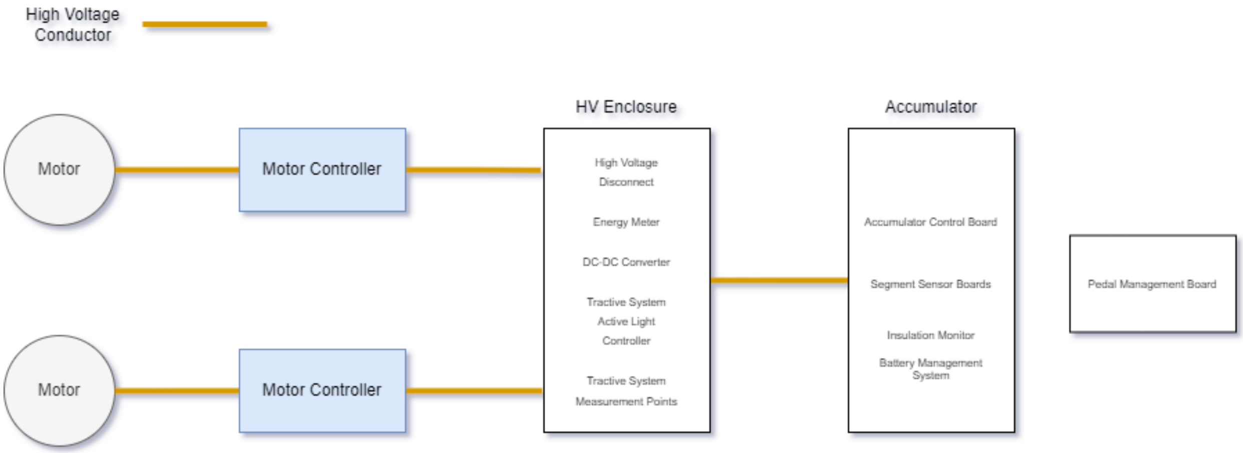




2. Background

The Viking Motorsports Team at PSU has been working on creating a fast, efficient, and innovative EV to participate in the FSAE competition. Our project involves monitoring several aspects of the EV via CAN and displaying the sensor outputs to the screen in the vehicle. Monitoring these aspects via CAN will allow for efficient diagnostics of the EV systems and make way for troubleshooting in the event of any failures.

The current electrical system is represented in the following diagram.



For simplicity, all low voltage connections have been omitted, as well as some minor systems. Currently, the only monitoring that is being done is battery temperature, and all monitoring done by off the shelf systems within their own system. None of this information is accessible without connecting a laptop to the various systems directly.

There are several different systems that need monitoring inside the EV. Primarily the HV (High Voltage) system requires monitoring of the temperature sensors for the compartments that hold the batteries. The temperature of the batteries affects the power output as well as the overall performance of the car. Other components include the accumulator isolation relays, fuse, current sensors, precharge and discharge system. The EV also requires monitoring of the LV (Low Voltage) systems as well as the status of the pedal control system.

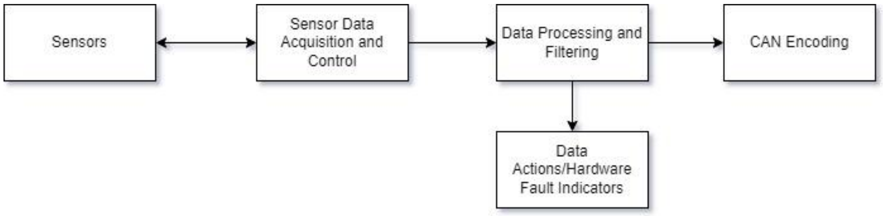
The industry standards have several requirements for CAN protocols being utilized in road vehicles. Some of them include the following from ISO 11898:

- CAN specifies the high-speed physical media attachment (HS-PMA), including HS-PMAs without and with low-power mode capability and those with selective wake-up functionality.
- CAN specifies characteristics of setting up an interchange of digital information between electronic control units of road vehicles equipped with CAN.

- CAN specifies the characteristics of setting up an interchange of digital information between modules implementing the protocol's data link layer.

Being able to monitor, process, and display these systems statuses will be extremely beneficial for troubleshooting and overlay monitoring of the EV. The Formula SAE competition has very specific requirements regarding galvanic isolation of the HV and LV systems in the EV. These requirements need to be followed when developing our CAN system.

Data Flow Diagram


Sensors
Sensor Data Acquisition and Control
Data Processing and Filtering
CAN Encoding
Data Actions/Hardware Fault Indicators

It is important to note that the VMS team has not utilized the CAN bus system for monitoring any sensor outputs in the past. However, even though the team has not implemented this system yet, there are several other off the shelf devices that utilize CAN to help monitor and also provide ability to alter the sensors and their calibration in order to get the best outcome from the overall system. Haltech offers a wide range of CAN Bus devices along with the sensors that are heavily used in the automotive industry, especially in DIY projects where factory engine management systems fail to deliver proper results. Research in companies like Haltech as well as other products like Vector used by Daimler can be beneficial for our project as well.

3. Project Overview

Our project starts with getting the STM32s to communicate with each other in order to relay information from one device to another. Each STM32 in our network will act as a CAN hub utilized to relay information from a specific component cluster on the vehicle. We then want to determine the protocols that allow data from the sensors to be received through the CAN transceivers.

We then have to design individual PCBs for each of the CAN hubs that hold the STM32s, the CAN transceivers and other peripheral components for the sensors to operate on the vehicle as well.

Using the STM32CubeIDE and C programming, we have to code each of the microcontrollers to log the data received from the CAN transceivers and relay them to the main Data Translator hub. The Data Translator further encodes the data to be read by a RaspberryPi via I^2C which is used to display the data on the screen.

We want the CAN system to be able to collect sensor outputs from different components of the EV, send them to a central control hub, which processes the sensor readouts to legible information for the user and outputs to a screen via the CAN adapter.

4. Product Design Specification

4.1 Concept of Operation / User stories

Our goal for this project is to provide several different CAN modules that can be connected to different sensors on the EV components and the placement of these modules provides flexibility and easy integration along with the rest of the parts on the EV. Since the EV has restraints for weight and countless other things, the placement of our modules is very important. These modules are connected to the sensors on the vehicle and thus take the sensor readouts when the EV is operated. The CAN module thus sends the readouts to the central control hub which is another module that we create. This central hub encodes the sensor outputs and then sends them to the LCD display on the car via a CAN Adapter.

As mentioned earlier, these modules are constructed specifically for the VMS EV and would for the most part only work for our specific application since the placement on the devices, types of sensor, variation in sensor outputs and the process to encode them would vary from one vehicle to another. Thus the VMS team and their intermediate sponsors would be our main users and stakeholders in this project.

It is important to note that the vehicles built by the VMS team in the past have not had any CAN bus system utilized for monitoring and diagnostics. All previous troubleshooting was done in the event of a system failure and the panels had to be taken apart to look at the individual components and sensors to determine the source of error. The CAN bus allows the electricians on the VMS team to simply use the LCD display and examine the readouts from the sensor in the event of the failure. This speeds the time it takes them to fix the car and get it back in the race at the FSAE competition. At the time the scope of our design concludes with relaying the data on the display screen. However, the project can be continued in the future to use the CAN Adapter to output the data logs to a laptop. There is also room in the future to be able to act upon the information received via CAN. For example, the ability to do system shutdown in the event of a component failure or out of range sensor reading can be a very effective addition to the overall safety of the vehicle.

In our case, since the sponsor is the primary user for our project, we want to follow the timeline of the rest of the EV build. Although we are not able to act upon the readouts we get from the sensor, we are merely monitoring them. However, there are products available in the market that allow for calibration and adjustment via USB connection to the CAN and sensors. This can be looked into upon future development of our project. Each of the CAN hubs will also act as data loggers as the microcontrollers will pull the data from the CAN bus, hold and store it.

As the rest of the members on the VMS team work on the other components that make the EV work, we want to work alongside their progress so as to make the integration of the CAN Bus more efficient and avoid any delays. We want the individual modules of the CAN Bus to be a plug and play system which can be routed along with the rest of the wiring on the EV. Although the design of the CAN Bus will be new for this year's EV, considering the fact that the VMS team has to build a new vehicle from scratch every year, they might be able to utilize some of the code and hardware and expand on the design of the CAN Bus for any future builds. Our goal is to create a smoothly integrated monitoring system system that is seamless with the rest of the EV build.

4.2 Stakeholders

Our project was proposed by Braedon Hamson, lead electrician on the VMS team. Him, along with several other key members on the team have been putting in countless hours in researching, developing, constructing, testing, and then improving the iterations of the EV. We are designing and constructing the CAN bus based on their requirements and working with them to create smooth integration of the CAN Bus into the EV. However we feel that there can be several other stakeholders in our project. The project team including Alice Ma , Braeden Hamson , Jade Nguyen , and Pushpesh Sharma , want to create something that leaves room for future growth and improvement. The EV inspectors that are part of the FSAE competition should also be considered solid stakeholders in this project as their judgment and criticism can be beneficial to the development of our project. But for the most part, the EV team at PSU will help integrate the CAN system, utilize it during the testing phase of the vehicle as well as on the track during the competition. Upon the success of this project, we hope that the VMS team is able to use the system for future use and add more features that helps the EV perform better in the upcoming competitions.

4.3 Requirements and Specifications

As mentioned earlier, we want to adhere to the requirements of the EV overall when designing and constructing our CAN Bus systems. We want to follow any necessary restraints set by the FSAE competition guidelines so as to avoid any delays and provide smooth integration of our systems with the rest of the vehicle. Thus most of our requirements for this project are based on the number of sensors and components that we want to read and monitor for the management of the EV. Other requirements are based on what components we utilize in our project in order to stay on budget and provide smooth integration when the time comes.

- Must be powered from 12V and/or 5V and tolerate a +/- 2V fluctuation
- When a board interacts with HV (350VDC) all HV components must be galvanically isolated from LV systems
- Must send data to a central data logging device connected to the CAN Bus
- Must use appropriate CAN Bus arbitration

- The team must cooperate with other members of the VMS team to ensure system compatibility
- Should follow accepted methods to protect devices on a CAN Bus
- Should use STM32 microcontrollers
- May monitor status of shutdown switches
- Should transmit data at a rate appropriate for the specific measurement
- Should monitor the following systems, in order of priority:
    - AIR+, AIR-, and precharge signals
    - APPS, BSPD, IMD, BMS OK, SDC status
    - Tractive system voltage, tractive system current
    - LV system voltages
    - Orion BMS 2 data
        - Battery state of charge
        - Tractive system power output
        - Battery voltage
        - Battery current
    - Cascadia Motion PM100DX data
        - Individual motor power
        - Inverter mode (running, fault, precharging etc.)
        - Relay outputs (used to detect wiring faults)

4.4 Deliverables

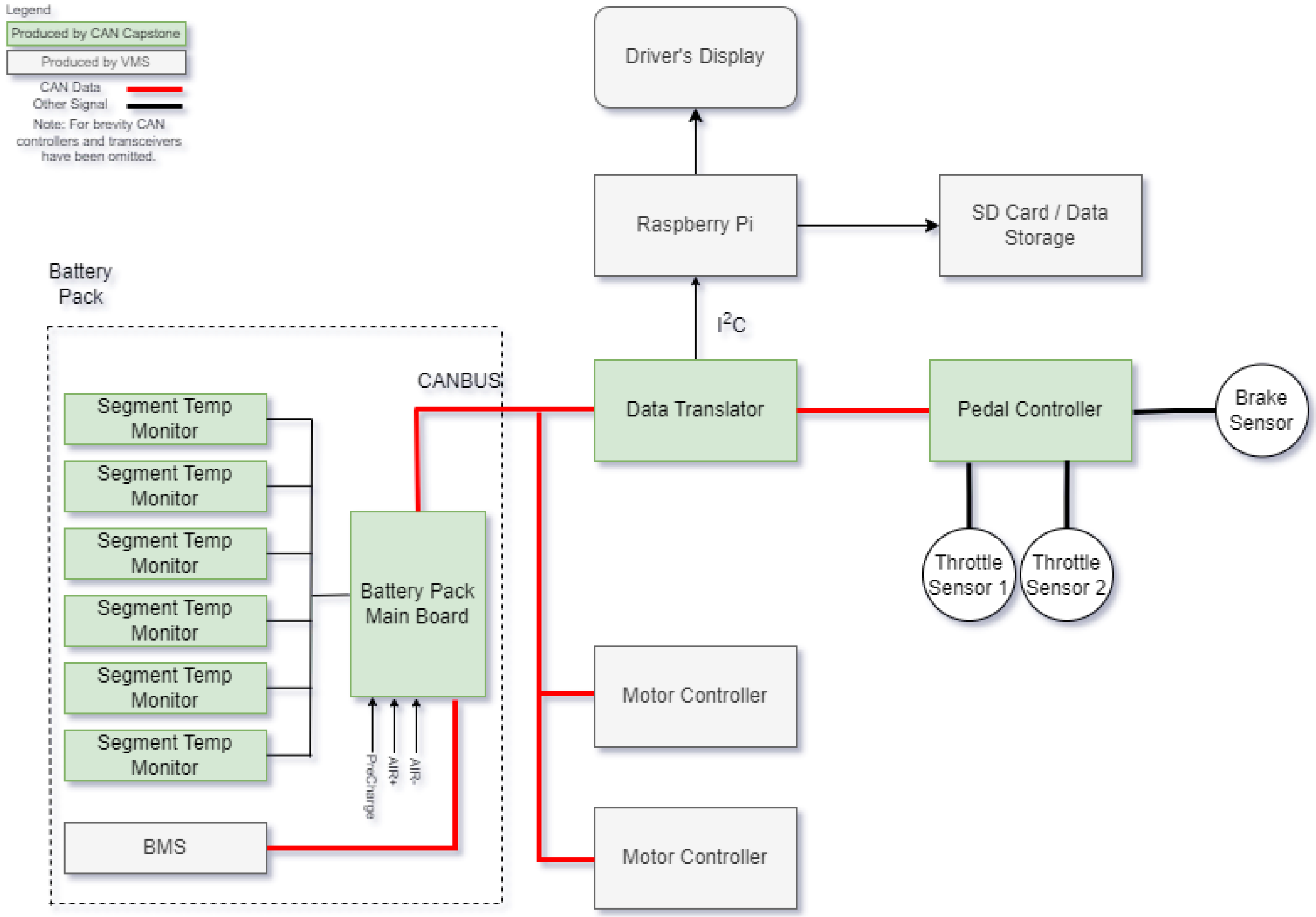
This critical section specs out what we are going to deliver to the industry sponsor by the end of your capstone. This **must** include:

- Project proposal
- Weekly Progress Reports
- Final Report
- ECE Capstone Poster Session poster
- Travel with the VMS team to Michigan to attend the FSAE competition. It is held on June 14-17 (travel and other expenses are covered)
- Detailed design documentation, explaining what the design does, how thedesign works, and why the team made the decisions it did
- Simulations, including both the simulation inputs and outputs
- Electrical CAD: Schematics and board layouts, including output files (Gerbers, PDFs, etc)
- Bill of materials and pricing
- Version control, including checked in previous revisions of your design (e.g., a Git repo).
- Individual PCBs with 3D printed housings for each of the boards.
- Functional code for STM32CubeIDE for the individual CAN hubs.

4.5 Initial product design

- Our current progress on this project consists of setting up software for the microcontroller that we will be using. We have decided to use STM32 along with the STM IDE to be able to take the sensor outputs from each component and then convert the values to send to the CAN Adapter which in the end gets displayed on the screen. We have also started working on the schematics for the PCBs that contain the STM32 microcontroller. These schematics contain the microcontroller itself along with the peripheral components that are necessary. We are also working on getting multiple STM32s to communicate with each other as the end goal would be to have several different boards placed in close proximity to the location of the sensor and have them send the sensor output to one main microcontroller board via CAN that processes the input and sends it to the CAN adapter.

- Hardware architecture



- Software architecture
    - C, STM32CubeIDE

- User interface / experience

    User Story: In the lab, the user is assembling the car, they test a certain feature and see that it does not function as intended. They can then go to the driver's steering wheel, use the on-wheel controls to enter a maintenance menu that will display all measured parameters by the CAN system. They can then use this information to find the problem. For instance, the car is powered on, but HV will not engage. The user looks at the data from the wheel and sees that the pedal control board is indicating a signal out of range of the pedal controller. Investigating the sensor connections they find the sensor unplugged. They plug the sensor in and the system indicates the sensor is in range. The car then turns on.

- Other considerations

    Electric vehicles are electrically noisy. Shielded wire will be used for long wire runs around the vehicle.

    The battery pack contains up to 350V, galvanic isolation will separate the boards that are close to HV systems from the rest of the systems.

- Back up plans

    The overall goal of this project is to create an easily debugged vehicle. If our plans totally crash and burn there's nothing wrong with creating a 1960's aesthetic by throwing a ton of blinking indicators around the car for debugging.

    A step up from that is to use Arduinos and create a similar but less complex system for debugging.

    The only aspect of this project that *must* be completed are the battery temperature monitors. Without this system the vehicle will not be allowed to race.

4.7 Verification plans

One of the biggest concerns with this project is the final integration of the CAN Bus with the rest of the components on the EV. We want to make sure that the system is able to successfully take the readouts from the sensors on the EV and encode them to be displayed on the screen. A lot of testing will be required to determine that the system is operating correctly.

The system will meet requirements if it can successfully read the temperature sensors in the battery modules and display them on the LCD. We may program the STM32s to encode the readout from the temperature sensors and determine whether it falls within the specific range for the battery to have optimal performance based on the requirements of the VMS team. Test: Does the LCD display the temperature of the specific battery module? Does it indicate when the temperature gets above a threshold range?

We are also going to test multiple STM32s to see if they can communicate with each other through the CAN network by sending messages from component A to component B, then from component B to component C, or until one goes to bus-off condition. we will determine if the system can analyze the error registers of the node as a means of measuring bus performance of a transceiver.

5. Project Management Plan

5.1 Timeline, with milestones

The timeline for our project needs to be aligned with that of the VMS team in order to finish the EV in time for the FSAE competition. Our sponsor wants to have a working CAN Bus system by the end of the ECE 412 course in order to leave ample time for system integration with the rest of the EV as well as the testing, and troubleshooting phases.

See Google Sheets below for a tentative timeline

Initial Gantt Chart for Capstone project - Google Sheets

5.2 Budget and Resources

The budget for our project is combined with that of the overall VMS budget for the EV. Our main costs are going to dependent on the following:

- Number of STM32s used and their cost
- Cost of peripheral components for the microcontrollers
- Manufacturing costs of the PCBs for the STM32s and the peripheral components
- Wiring for the integrated boards via CAN.

The VMS team has kindly provided Altium to design the PCBS for the CAN modules. Most other components will be ordered from DigiKey or may already be available in the VMS inventories as they have been working on multiple vehicles in the past. The VMS team has also provided a work area along with the rest of the team members as resources for our project.

5.3 Intellectual Property Discussion

A goal of this capstone is to provide a framework to future teams to build more advanced CAN systems. Therefore all IP generated here will be completely open source, and will be readily available to future teams. The license used for the repository and the project overall is the **GNU General Public License v3.0**

5.4 Team and Development process

- Alice Ma: Project manager and Programming lead. Working on programming STM32 for sending and receiving CAN messages.
- Braeden Hamson: VMS electrical lead. CAN capstone technical lead. Understands the vehicle systems and the interface between the CAN system and the rest of the systems of the car.
- Jade Nguyen: working with Alice on setting up a CAN bus using the STM32 microcontroller, providing simple C code to read and write to the CAN bus and send it to the transceiver.
- Pushpesh Sharma: Working on PCB schematics and using Altium to set up schematics, footprints, PCB files and manufacturing files for several boards for CAN Bus.
- Collaboration tools: Notion for team tasking and note taking, Discord for every comms and virtual meetings, Github for code and boards, Google Suite for deliverables.

Faculty Advisor Approval

Advisor Name:_Roy Kravitz_

Signature:__e-approved by Roy Kravitz (rkravitz@pdx.edu)_

Date:__27-Feb-2023__

Project Sponsor Approval

Viking Motorsports (VMS) is a student group focused on designing and building formula race cars from the ground up. Every year, VMS competes in Formula SAE, which is a worldwide collegiate competition created by SAE International.

Sponsor Name: George Riera

Signature: GR

Date: 2/20/23