

## ABSTRACT

Title of Thesis:

***DISTRIBUTED CONTROL FOR FORMULA SAE-  
TYPE ELECTRIC VEHICLE***

Samantha Falco, Masters of Science in Electrical  
Engineering, 2022

Thesis Directed By:

Professor, Alireza Khaligh, and Department of  
Electrical and Computer Engineering

The recent trend in transportation electrification creates an enormous increase in demand for electric vehicles (EVs). Increasingly, electric cars have novel features like autonomous driving and fault tolerance, all of which require additional hardware and computation power. Changes to the electronic control unit (ECU) structure will be needed to make these advances scalable. This thesis examines the driving economic, technical, and societal factors behind needed changes to the existing control structures. It proposes a control platform design to address issues of complexity and scalability. A generic, modular control board structure using the TMS320F2837xS digital signal processor (DSP) is described with several input/output functionalities including a wide range of analog inputs, multiple logic levels for digital pins, CAN communication, and wireless communication capabilities. A distributed control network is built by interconnecting multiple implementations of the control board, each of which has distinct responsibilities dictated by software instead of hardware. A prototype electric vehicle control structure for a Formula SAE electric vehicle was built utilizing a network of three control boards and tested to prove the viability of the proposed concept. Results of these tests and future steps for the project are discussed.

DISTRIBUTED CONTROL FOR FORMULA SAE-TYPE ELECTRIC VEHICLE

by

Samantha Falco

Thesis submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Masters of Science  
2022

Advisory Committee:  
Professor Alireza Khaligh, Chair  
Professor Sahil Shah  
Professor Kevin Daniels

© Copyright by  
Samantha Falco  
2022

## Dedication

I would like to dedicate this work to the residents of “the house” (permanent and impermanent). The people I live with had a real and positive impact on my ability to complete this project. Thank you for every laughter filled night spent distracting me from my work --- I really needed them.

## Acknowledgements

Thank you, firstly, to Professor Alireza Khaligh for supporting me throughout my time in graduate school! I would also like to thank all the members of the Maryland Power Electronics Lab, especially Dr. Rakesh Resalayyan, Arafat Hasnain, Ayodhya Somiruwana, and Chanaka Singhabahu. Time spent in the lab was formative to me, not only professionally but personally as well. I appreciate all the lab members generous sharing of time and experience and will miss hearing the jovial debates.

## Table of Contents

<i>Dedication</i> .....	<i>ii</i>
<i>Acknowledgements</i> .....	<i>iii</i>
<i>List of Tables</i> .....	<i>vi</i>
<i>List of Figures</i> .....	<i>vii</i>
<i>List of Abbreviations</i> .....	<i>viii</i>
<b>Chapter 1: Introduction</b> .....	<b>1</b>
Environment for Research .....	1
Problem Statement .....	2
Proposed Solution .....	3
Applications.....	4
<b>Chapter 2: Preliminary Research</b> .....	<b>6</b>
Electric Vehicles.....	6
Electronic Control Unit.....	8
Controller Area Network (CAN).....	11
Autonomy .....	13
Wireless Telemetry.....	14
<b>Chapter 3: Distributed Controller Hardware Design</b> .....	<b>17</b>
Design Plan.....	17
Component Selection .....	20
Schematic Design.....	21
PCB Design .....	28
Software Structure .....	29
<b>Chapter 4: Experimental Verification</b> .....	<b>32</b>
Test Plan.....	32
Peripheral Testing.....	32
System Test Scheme .....	35
Success Metrics .....	39
Results.....	40
<b>Chapter 5: Conclusions</b> .....	<b>47</b>
Conclusion .....	47
Future Work .....	48

<i>Appendices</i> .....	<b>49</b>
<b>Appendix A: Datasheets</b> .....	<b>49</b>
<b>Appendix B: 3D Layout of Control board</b> .....	<b>50</b>
<b>Appendix C: Assembled Control Board</b> .....	<b>51</b>
<i>Bibliography</i> .....	<b>53</b>

## List of Tables

Table 1: Standard CAN Message Structure .....	12
Table 2: Schematic Organization.....	21
Table 3: Layout Key .....	29
Table 4: CAN Data Byte Set-up for Inverter .....	38
Table 5: Battery BMS status CAN message format .....	39
Table 6: Vehicle speed CAN message format .....	39

## List of Figures

Figure 3: S-Curve of EV required EV adoption to meet targets [9] .....	6
Figure 4: Duck Curve.....	7
Figure 5: CAN architecture.....	12
Figure 1: Single control module block diagram .....	18
Figure 2: High level block diagram of distributed control .....	19
Figure 6: Buck converter topology .....	22
Figure 7: Power Supply Schematic.....	23
Figure 8: Analog Inputs Schematic.....	24
Figure 9: Digital Inputs Schematic .....	26
Figure 10: Digital Outputs Schematic.....	27
Figure 11: Digital and DSP Schematic .....	28
Figure 12: 2D Layout and Routing.....	29
Figure 13: Basic software flow for each control board.....	30
Figure 14: XBee Network Connection Verification .....	33
Figure 15: RSSI Test Results.....	34
Figure 16: CAN bus software verification set up .....	34
Figure 17: CAN message trace .....	35
Figure 18: (a) Block diagram of distributed control test scheme (b) Distributed control test scheme implemented with hardware .....	37
Figure 19: Permanent magnet synchronous motor speed-torque curve.....	38
Figure 20: Digital outputs from DSP after voltage scaling up to 12V.....	42
Figure 21: Digital inputs to DSP after voltage scaling of signal in Fig. 20.....	43
Figure 22: CAN message oscilloscope trace from two sources.....	44
Figure 23: CAN message reception verification and torque updates with change in accelerator position .....	45
Figure 24: Inverter power off when BMS voltage status is low .....	45
Figure 25: 3D Layout of Topside.....	50
Figure 26: 3D Layout of Bottom side .....	51
Figure 27: Assembled board, top view .....	51
Figure 28: Assembled board, bottom view .....	52

## List of Abbreviations

ADAS – Advanced driver assistance systems

BEV – Battery electric vehicle

BMS – Battery management system

CAN – Controller area network

CO<sub>2</sub> – Carbon dioxide

DSP – Digital signal processor

ECU – Electronic control unit

EV – Electric vehicle

GPS – Global positioning system

HEV – Hybrid electric vehicle

IC – Internal combustion

ICE – Internal combustion engine

LIDAR – Light detection and ranging

PHEV – Plug-in hybrid electric vehicle

RPM – Rotations per minute

RSSI – Received Signal Strength Indicator

SoC – State of Charge

ZEV – Zero emission vehicle

# Chapter 1: Introduction

## Environment for Research

It is clear to car experts and novices alike that the future of auto transportation is electric. Nearly every major automaker in the world has announced a plan to offer electric options and pledge to reach carbon neutrality sometime before 2050 [1]. In addition to industry commitments, state legislatures are also working to support the transition to electric vehicles (EVs). Transportation currently accounts for around 30% of the U.S.'s carbon dioxide (CO<sub>2</sub>) emissions, therefore, in order for states to meet their futuristic net zero goals, the combustion engines used for the majority of transportation in the U.S. must be eliminated [2].

The trend toward car electrification is clear, but as [3] and [4] describe in detail, many uncertainties exist in how the world will handle this next evolution in transportation. Factors such as government policies, infrastructure shortages, engineering scaling challenges, and the impact of urbanization on transportation are all discussed.

Demand for EV's has increased exponentially in recent years, but ownership is still limited to only the wealthiest in the United States. Active legislation in many state governments is working to make the cost of EVs more affordable for all car buyers. Additionally, the infrastructure bill passed by Congress recently included funds for EV charging infrastructure [2].

Such a strong financial and political commitment to electric vehicles from many top automakers should give researchers the confidence and enthusiasm to continue efforts in EV design, control, and technology research. Futuristic electric vehicles will require new engineering and control techniques. Meeting the increased demand for advanced electric vehicles will require new design and manufacturing techniques. The transition to EVs also promises entirely new car capabilities, most notably autonomous driving, which are still active fields of research.

To enter such a field of active research, some initial platforms are necessary upon which further research can be performed. In this work, a modular, distributed design for an EV control board is proposed. The design, a new take on traditional vehicle electronic control unit (ECU) control structures, aims to generalize and simplify existing vehicle control structures to address problems with current ECU implementations. This control structure will be capable of serving as a platform for EV research in the future.

### *Problem Statement*

For the last century, cars have become increasingly complex as consumers have come to expect more and more features in their vehicles. This will continue to be true in the coming years. The shift to full vehicle electrification, self-driving capabilities, increased fault tolerance, and more advanced onboard entertainment systems will demand more of cars' space, efficiency, and computational power. Innovative solutions to minimize the amount of hardware needed, cost, complexity, and the risk of failure will be critical [5]. Outside challenges like the world-wide semiconductor chip shortage

and associated supply chain problems increase the need for solutions. Witnessing more of the adverse effects of climate change puts more pressure on the auto industry and governments to electrify transportation.

Car drivetrains are noisy, fault filled environments. Increasing numbers of sensors needed for car electrification and automation will increase complexity. As cars become more complicated and autonomous, it becomes even more important they are not vulnerable to any single point of failure. Fault tolerance is essential. Redundancy and distributed control are important to ensure safety and reliability.

### *Proposed Solution*

All the mentioned issues must be considered when designing a new EV, and the control board designed in this thesis aims to address these problems in some capacity. This work proposes a modular, distributed design for EV control. The central concept is that a single, multipurpose control module implemented in multiple locations throughout the vehicle may create a distributed control network. Each iteration of the board is physically identical but used differently in different scenarios. External signals are encoded locally on the control board and converted to the Controller Area Network (CAN) protocol which is then put on a CAN bus.

Each module has its own EV-specific digital signal processor (DSP) capable of doing local calculations. In addition, each module will complete tasks relevant to its immediate area. At a system level, one module's DSP will be assigned as the master

and take CAN bus information from peripheral sensors and control the vehicle accordingly. For example, position sensor analog information from the accelerator pedal will be input to its local control board, encoded in CAN, pushed to the CAN bus, processed by the master control module, and then used as a factor to drive the main motors at the appropriate speed.

Such a solution would allow signals to be encoded to the more robust CAN protocol locally, thus reducing the risk of faults derived from operating in a noisy environment. Since each module has the same physical capabilities, if one board has a fault, it would be possible to reroute its load to another, thus making the design more resilient to fault events and easier to perform maintenance on.

Main contributions of the proposed design include a new distributed controller architecture which will increase fault resilience and a prototype controller module which experimentally validates the architecture.

### Applications

The initial design for this concept will be implemented in a Formula SAE EV used in student racing competitions. Rules for the design of Formula SAE competition cars are described in [6]. The distributed control design was envisioned to solve problems that past Formula SAE EVs had when using a single central ECU. Many of these problems are the same as those described in detail above.

Control for the existing Formula SAE EV is fully centralized. Centralized control creates an array of other problems. The design has a single point of failure and signals are vulnerable to noise as they travel long distances.

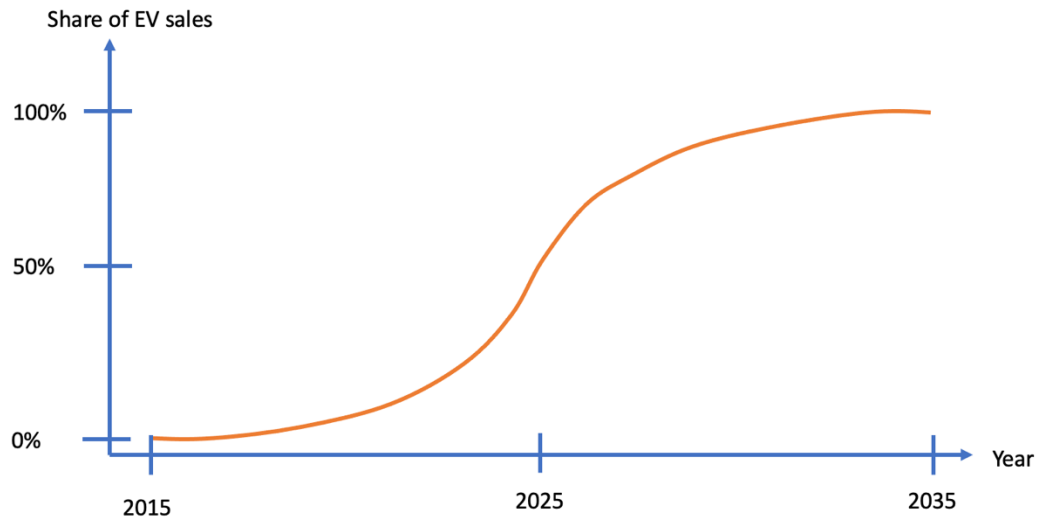
Though the distributed control was initially designed to replace the Formula SAE EV centralized ECU, it will also allow for development beyond initial uses. Future iterations of the design are set up to allow for exploration into autonomous driving, wireless communication, and even advanced fault tolerance techniques.

The rest of this work shows the process by which this design plan was implemented. Chapter 2 describes the current state-of-the-art of EV control in more detail. This specific design is introduced in Chapter 3 and the details are justified. Test procedures and results are outlined in Chapter 4. Finally, Chapter 5 discusses successes, failures, and future steps for the project.

## Chapter 2: Preliminary Research

### Electric Vehicles

To meet ambitious emission reduction pledges like the Paris Agreement, EV production will need to scale up incredibly rapidly, with 100% of vehicle sales being electric by 2035 [9],[10]. Figure 3 shows how significant the increase in EV adoption will need to be if these targets are to be met.



*Figure 1: S-Curve of EV required EV adoption to meet targets [9]*

The term electric vehicle refers to several different types of vehicle architectures including ZEV's, PHEV's, HEV's, and BEV's. It will take a combination of all these types of vehicles to transition to the zero-emission world necessary to meet CO<sub>2</sub> emission reduction pledges and combat climate change. Though examples of all four types of these vehicles exist today, much work remains to make their model feasible on a global scale. The vehicle designed in this paper is fully electric.

One such problem deals with EV integration into the grid, especially as the grid becomes largely composed of renewable energy sources. The addition of EV's creates a significant new load on existing grid infrastructure. The integration of more renewable energy sources to grid power exacerbates this problem. Discrepancies between energy demand and energy generation associated with renewable energy, or the duck curve, shown in Figure 4, will cause problems if EV owners tend to charge at inopportune times.

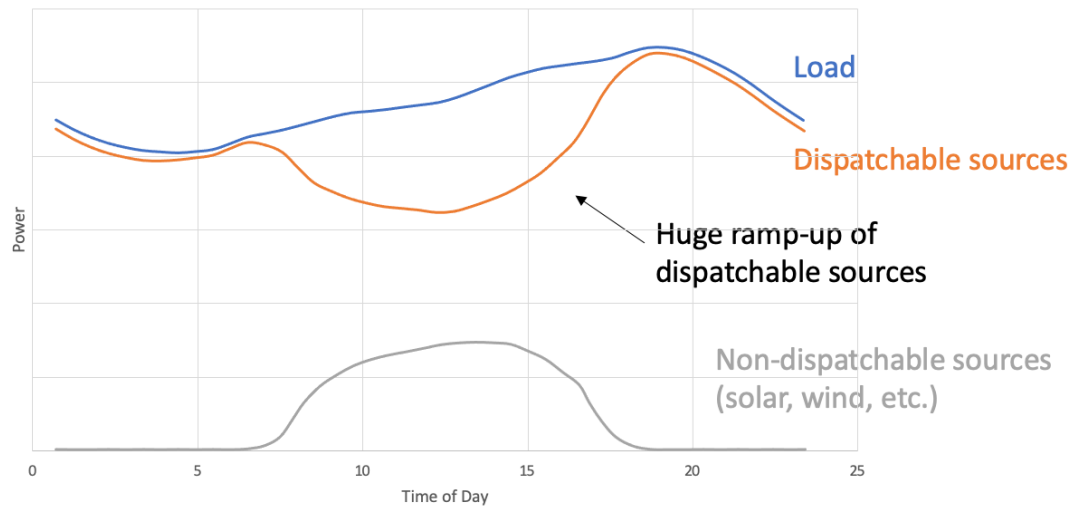


Figure 2: Duck Curve

Supportive policy must be implemented alongside robust engineering to address these issues. Article [11] predicts that a plan for “coordinated charging” of EV's when energy generation is high and implementation of tariffs for charging when generation is low is likely to be implemented. Research on the optimal coordination of EV charging has already been conducted, and such algorithms are described in [12] and [13].

A complex communication interface between the EV and the smart grid will be needed to optimize charging for the grid as well as to minimize range anxiety among EV owners. Such a communication interface will likely be achieved primarily via the charging cable while charging as well as through other wireless mechanisms, ZigBee or WIFI, between charging stations, power plants, and EV owners. Extremely accurate measurements of the EV's charging status (SoC) will need to be computed by the ECU for use in such communications.

Effective EV designs maximize driving range while minimizing cost, all without compromising safety or security. Driving range can be increased with good battery management, weight minimization, and energy efficiency. The ECU is responsible for oversight of the processes which monitor battery SoC and energy usage in the vehicle. The ECU will be required to perform accurate status measurements of all these systems.

### *Electronic Control Unit*

The ECU is the brain of a vehicle. For an EV, its main responsibilities include the following:

1. Monitoring battery state of charge level via the BMS to prevent unsafe battery conditions and maximize efficient charging and discharging
  - a. Including smart grid connection while charging
2. By-wire acceleration and braking to maximize performance and efficiency
  - a. Proper control of on-board inverter and electric motor
3. Robust monitoring of the status of all EV systems

4. Vehicle wireless communication and other driver comfort and infotainment systems
5. Physical and cyber security
6. Advanced driver assistance services (ADAS) and future-looking autonomous driving features and redundancies

Modern internal combustion cars have multiple ECUs scattered throughout the car. Each ECU usually serves an individual purpose, controlling a single closed system, like the engine, suspension, windshield wipers, etc.

Electric vehicles are only in the infancy of their evolution. EV control structure started out emulating IC cars, but their future will soon divulge. The powertrain of an EV is significantly simpler than that of an IC vehicle which simplifies control there. However, as [14] points out, EV's are increasingly "smart". The transition to EVs coincides with a push for autonomous driving, a change which demands significant increases in ECU capability.

The next generation of EVs will require more sensors, faster communication, better security, and more redundancy than ever before to make autonomy a reality. Conversely, manufacturers must find a way to decrease costs so that purchasing an EV is financially possible for people worldwide. As [15] pointed out, the limited availability of the earth metals like Lithium, Cobalt, and Nickel used to construct EV hardware and batteries will also need to be considered to make the transition to EV's feasible worldwide.

EVs of the future will further decouple user input and car response, this is called transitioning to “by-wire” actuation and means that electrical signals will be used instead of a directly mechanical linkages between, for example, the steering wheel and wheel directionality. Increasing the number of sensors and other features of a car tends to increase the number of wires needed. More wires take up more space, create more weight, and add more cost – all of which should be minimized in a good EV.

Several solutions to this problem have been identified in relevant literature. Article [16] argues that the consolidation of ECU’s will be necessary to reduce complexity and space required, even as the amount of processing power needed increases. Currently, cars are organized systemically. That is, each discrete system (environment control, brakes, etc.) has its own ECU. [5] explains that a shift may occur towards a more “zonal” architecture – where all components physically near each other are controlled by the same ECU, regardless of which system they belong to.

The distributed control board proposed in this paper provides a platform upon which such future-looking concepts can be applied. Many more ECUs will be needed in the coming years to meet global demand. Limiting hardware differences between implementations of ECUs will decrease complexity and make replacements simpler. The distributed design allows for a zonal organization which will decrease the amount of cabling necessary. Simple hardware design with accommodations for many different communication protocols and input/output types makes the board easy to reconfigure

for future needs. Allocations for wireless communication are made so that it may communicate with the smart grid as necessary.

### Controller Area Network (CAN)

Different implementations of the control board will be connected and communicate via CAN. The CAN communication protocol was first developed in the 1980's as an alternative to increasing the number of cables in automotive signal design. It allowed designers to replace heavy and expensive wire harnesses with much lighter CAN busses and made for a communication network that was significantly more robust and streamlined than what was being used before [17]. CAN is now the standard communication protocol used in all vehicles.

Basic operation of CAN as it will be used is described in [17],[17] and shown in Figure 5. Each node (control board, BMS, inverter, etc.) is connected to the CAN bus via a CAN transceiver. Each end of the bus is terminated with a 60  $\Omega$  resistor. Only one message is on the bus at a time and the next message to be pushed to the bus is determined using a priority bit at the beginning of the message as there is no master node [18]. Each message in standard CAN has 8 data bits.

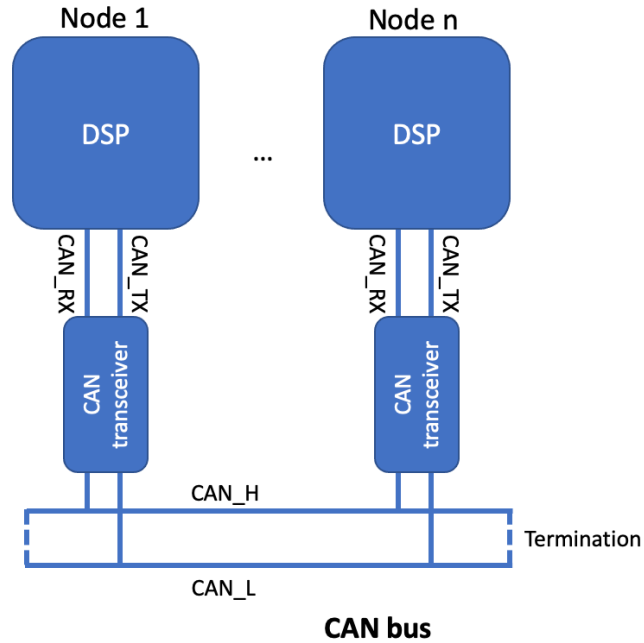


Figure 3: CAN architecture

Two CAN message structures exist: CAN 2.0A and CAN 2.0B. CAN 2.0A, or standard CAN, supports 11-bit identifiers and is used in this project. Table 1 shows the message structure for standard CAN, as described in [9]. The most important data fields are the identifier and the data bytes. These fields will be discussed in further detail in the testing chapter.

Table 1: Standard CAN Message Structure

SOF	11-bit Identifier	RTR	IDE	r0	DLC	8 Bytes of Data	CRC	ACK	EOF	IFS
-----	-------------------	-----	-----	----	-----	-----------------	-----	-----	-----	-----

CAN's reputation for robustness can be attributed to multiple features. CAN is message-based, which means it can send information to multiple recipients at once. Additionally, if a node is corrupted, CAN automatically removes it from the bus. This means a local problem will not be fatal for the vehicle. Nodes can be added to the bus during operation without problem [9]. The control board's software should be designed

so that if the failed node happens to have been assigned as the system master, responsibilities are transferred to another implementation of the board.

The original reason for CAN's invention was to minimize faults and move away from single purpose hardware [19]. Similarly, this new modular control scheme will make EV control simpler, easier to replace, and decrease hardware development time all while ensuring fault security.

### Autonomy

Autonomous driving means more than fully self-driving cars. All levels of autonomy are considered, including everything from cruise control to full autonomy. A general overview of the field of autonomous driving is provided in [20],[21],[22]. Autonomous vehicles are, most simply, a system which, given information about its location in the surrounding environment, can make decisions about its own movement. In a fully non-autonomous vehicle, all control is open loop. A fully autonomous car uses completely closed loop control. Any sensing errors must be identified and addressed quickly, as a result. This requires the closed loop system to be extremely fault-tolerant and robust. As the EV becomes more automated, more attention will need to be given to security and safety. Redundancies for every system need to be built in. The ISO 26262 standard should be followed. One advantage to using CAN as the main communicator across the vehicle is its long-standing status as the standard for vehicle communication. Years of research and development have already been completed to develop fault detection algorithms, one of which is described in [23]. Future development of this control board architecture may borrow from these existing algorithms and apply them to autonomy.

There are several sensing mechanisms the vehicle uses to gain information about its surroundings. A few types of these sensors include LIDAR, GPS, video cameras, and odometers. A software framework for autonomous driving using these inputs is described in [24]. In this framework, each external input is sensed, kept in memory, and then used in an algorithm to plan the path of the car or control its performance. Each of these sensing options requires additional sensing and control hardware to support their use.

As autonomous vehicles become more and more advanced, the amount of sensing equipment and the computational power needed to process it will only increase. The system described in [9] was implemented on a Formula SAE vehicle. To allow for the eventual implementation of autonomous driving in the control structure described in this paper, the board was described with simplicity and modularity in mind so that future students may add additional features, like ethernet capabilities, easily. The general-purpose design of the control board will allow additional features to be added to the car without having to completely redesign the hardware.

### *Wireless Telemetry*

Wireless communication is an essential feature of the control board design. The Formula SAE EV student team will need wireless communication in the board to monitor the status of the car from a ground station, up to a mile away. As shown in [25], it is important for the SAE team to be able to measure and adjust the status of

different systems in the EV in real time to ensure optimal performance. This also serves as a record of car operation which may be referenced in the event of a fault.

More generically, wireless comm capabilities are needed on an EV ECU board to facilitate communication between parties on the smart grid. Communication between the vehicle and charging stations, as described in [26] and [27], will be necessary to properly integrate coordinated charging. Additional communications mechanisms will grow between charging stations, with the driver, and to power plants as part of a “Internet of Energy”. For example, when the EV battery needs to be charged, the driver is alerted, either in the car or remotely. The EV can also communicate wirelessly with charging stations near its current locations to optimize its coordinated charging process.

The ZigBee family of communication protocols was selected to serve this purpose. Strengths and draw backs for different protocols were discussed in [4]. ZigBee PRO was selected for this board because it allows long range communication between many nodes. Wi-Fi could also have been used for this purpose as it offers longer range communication but may be limited by the number of different nodes it could communicate with at one time.

The ZigBee module communicated with each board DSP via SCI/UART asynchronous serial communication. SCI data formatting consists of a start bit, 8 data bits, and a stop bit. Serial data in the RF module is transmitted to other modules in the network with a 2.4 GHz RPSMA antenna. The modules are set up to operate in a peer-to-peer network.

This allows multiple modules to communicate with one another without any one of them being labelled as master or slave. According to the module datasheet [28], this network configuration style affords fast synchronization and start times.

## Chapter 3: Distributed Controller Hardware Design

An ECU design for a similar Formula SAE is described in [29]. General design for a Formula SAE vehicle is described in [30],[31]. The main goal for this design was to create a simple, modular design that could easily be reconfigured to meet the user's needs.

Hardware is centered around the DSP. Support hardware is needed to enable the DSP to communicate with other ECU modules in the EV, the BMS, and other sensors. Digital inputs, analog inputs, and serial pins all have input stages before entering/exiting the DSP. Power is taken from the EV's 12V battery and stepped down to voltage levels appropriate for the DSP.

### Design Plan

Design of the distributed control architecture composed of generic control modules occurred in multiple stages as follows:

1. Planning and requirement understanding
2. Component selection
3. Module schematic design
4. Module PCB layout
5. Prototype module assembly
6. Testing of individual modules and control network
7. Revisions

Each step in the process will be described in detail. A block diagram showing the composition of each control board is shown in Figure 1. Each module will have its own DSP, power supply, CAN transceiver hardware, analog and digital input processing, digital output processing, wireless communication capabilities, and personal protections. Generically including inputs and outputs to the board means it can be easily customized to local needs.

Designers may choose not to populate specific components of the board which are not needed for a particular implementation. This would decrease the cost of each board and minimize impacts due to the international chip shortage [7].

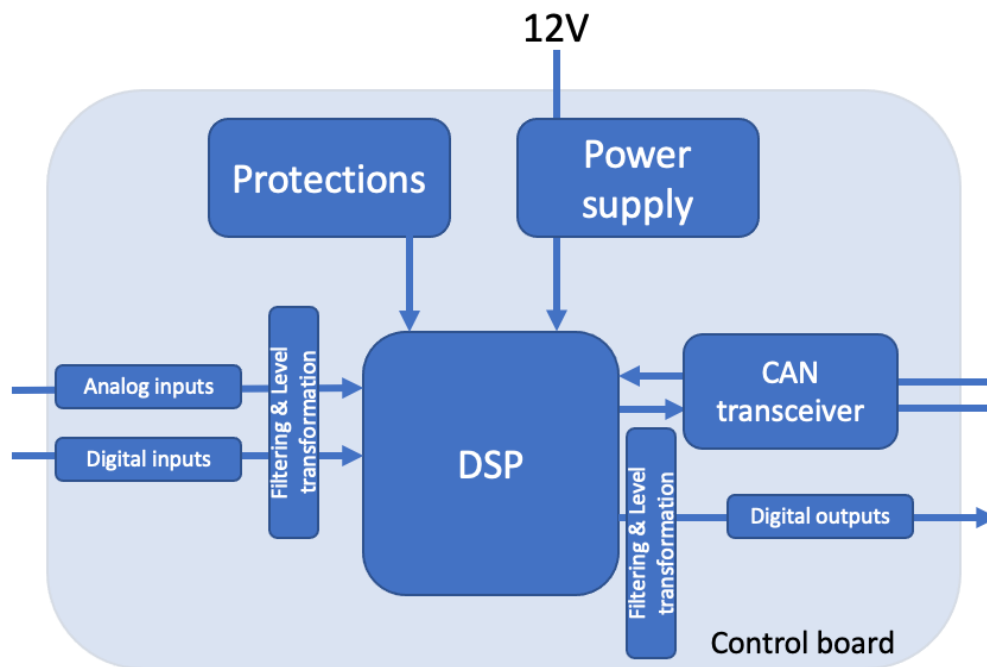


Figure 4: Single control module block diagram

Figure 2 shows one possible realization of the proposed distributed, modular control structure using three control boards. Control module 1 is located near the front of the

car and receives inputs from the accelerator. Control module 2, near the 400V traction battery stack BMS (battery management system), inputs information from the Lithium ion batteries about their status to ensure they are being operated properly. As [8] describes, the BMS is critical to ensure that the EV's battery pack operates reliably and safely. All this information is encoded on to the CAN bus and considered by the master DSP when deciding what speed to drive the inverter/motor combo at. A Zigbee wireless telemetry module is connected to one control board and used to communicate real-time car status information wirelessly within a short range.

Each board is only inputting and outputting one piece of information in this simple example. In reality, each board is physically capable of supporting 10 analog inputs, 10 digital inputs, 10 digital outputs, 2 CAN busses, and one wireless communication module.

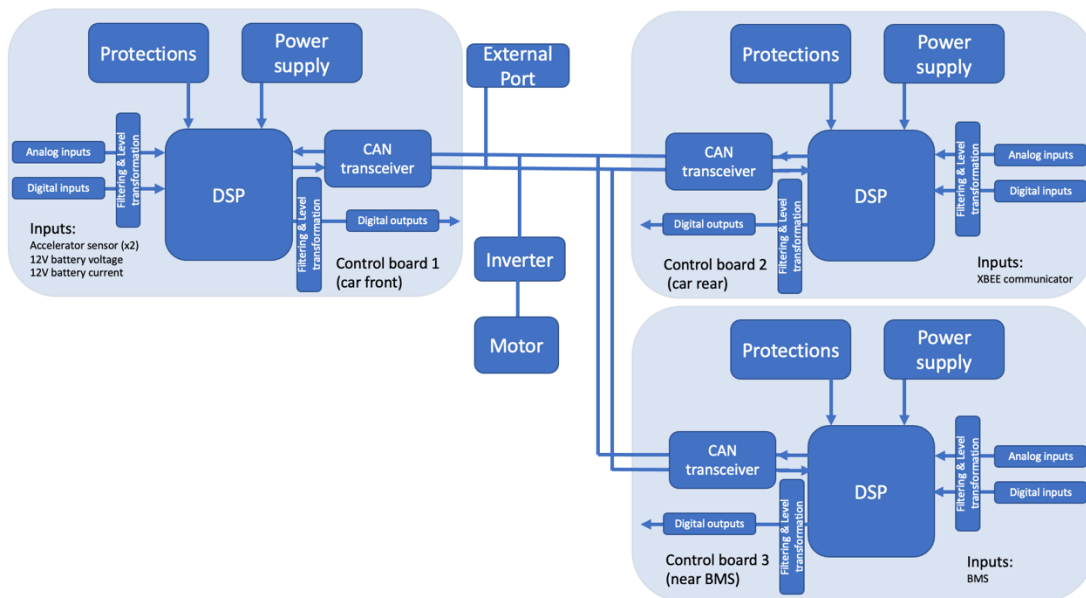


Figure 5: High level block diagram of distributed control

## Component Selection

Four of the components used on the board required special care in their selection as their proper functioning is critical to the board's success. These components are the CAN transceivers, the DSP, the power supply IC's and the ZigBee module.

### *1. Digital Signal Processor*

Texas Instrument's TMS320F28374S microcontroller, a single-core 100 pin DSP, was selected for this project. The C2000 32-bit line of microcontrollers are designed to support applications which require significant sensing and processing, especially for closed-loop systems [32],[33]. This makes it ideal for use in electric vehicles.

### *2. CAN Transceivers*

Two CAN transceivers are used per board and serve the same purpose. The transceiver converts the CAN\_HI/CAN\_LO signals from the CAN bus to the CAN\_TX/CAN\_RX signals that the DSP uses. MaxLinear's XR31233ED CAN transceiver is used [34]. This transceiver was chosen mainly because it supports 3.3V operation, so no signal voltage-level translations were needed between the DSP and the transceiver.

### *3. Voltage Regulators*

Four voltage regulators are used. The first, Infineon Technology's TLS4125D0EPV50XUMA1, steps voltage from the 12V battery down to a constant 5V output [35]. The second, Texas Instrument's LM3674MF-1.2/NOPB, uses a buck converter to step the 5V down to 1.2V [30]. The last two both use Semtech

Corporation's SC4626ZSKTRT to buck 5V down to 3.3V, one for the DSP circuit and one for the ZigBee module power supply [31]. All three regulators were chosen to meet the desired voltage levels and can support the necessary current values.

#### 4. XBee Pro Module

The Digi XBee Pro 900 HP module, XBP9B-DPST-001, was chosen to implement the ZigBee network. The Pro line of XBee modules offers exceptional data transmission range, especially with a high gain antenna. This feature is desired for the immediate uses of the module for this project.

#### Schematic Design

Design of the schematic is broken down into 5 parts, over 5 pages. The contents of each page are described in Table 2.

Table 2: Schematic Organization

	# Connectors	Description	Uses
<i>Power Supply</i>	1	Converts 12V battery input to 3.3V and 1.2V.	Powers all circuit components
<i>Analog Inputs</i>	10, 3 DAC	Processes analog input signals before input to DSP ADC	Accelerator pedal, sensor inputs, etc.
<i>Digital Inputs</i>	10	Processes digital input signals before input to DSP GPIO	Brake pedal, sensor inputs, etc.
<i>Digital Outputs</i>	10	Processes digital output signals to appropriate voltage before external output	LEDs, actuators, etc.
<i>DSP/ Processing</i>	11 GPIO, 2 CAN, 4 test points, JTAG	Includes circuits for critical functionalities like CAN, XBEE, DSP programming	CAN bus, wireless communication, programming, etc.

The power supply page has a connector which pulls 12V power from the EV battery and steps it down to 3.3V and 1.2V for use powering the DSP and other components on the board. A series of 4 buck converters are used to accomplish this, first from 12V to 5V, then 2 from 5V to 3.3V and one from 5V to 1.2V. Switching DC-DC converters are used for their efficiency relative to linear converters. Figure 6 shows the basic buck converter structure. An inductor was selected to avoid saturation and minimize current ripple, according to (1) [36].

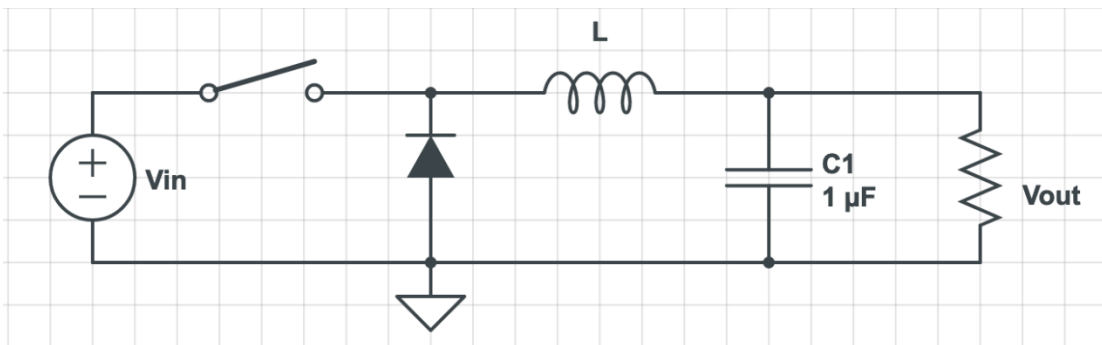


Figure 6: Buck converter topology

$$\Delta I = \frac{V_{out}(V_{in} - V_{out})}{2LV_{in}f} \quad (1)$$

$\Delta I$  = current ripple  
 $f$  = frequency in Hz  
 $L$  = minimum inductance

A TVS diode and series fuse combination is used to protect the circuit from over current and over voltage from the source. A status LED is included to show when power is

present on the board. Power decoupling capacitors are also included. Figure 7 shows the implemented power schematic.

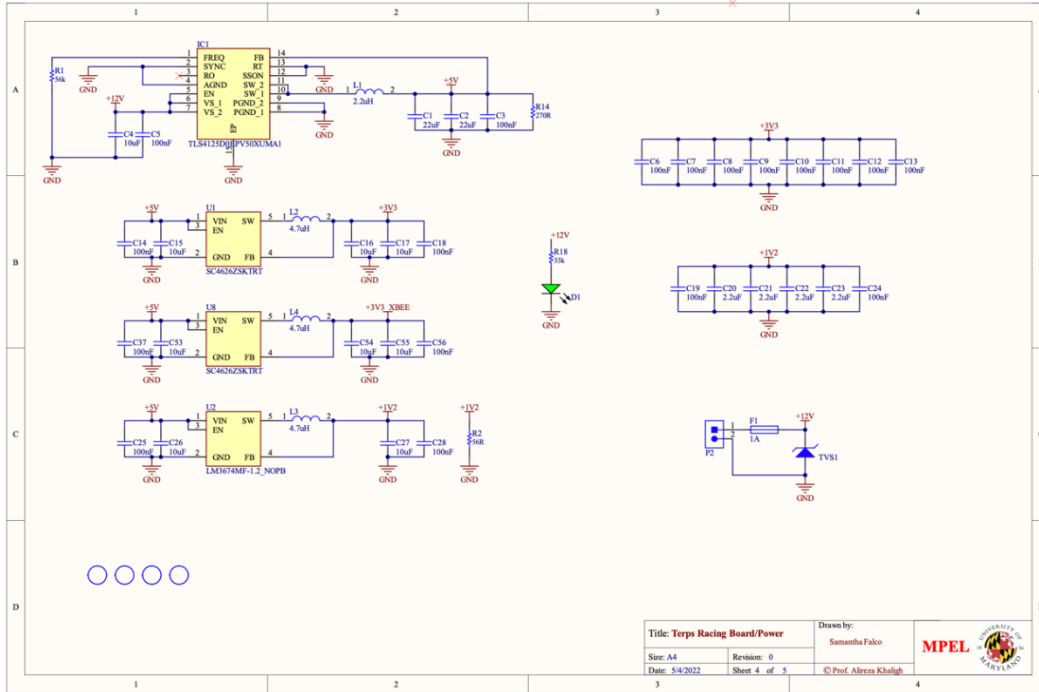


Figure 7: Power Supply Schematic

Inputs to the ECU board from the EV will be either analog or digital. The analog and digital input schematic pages in Figures 8 and 9 show how these inputs are filtered before entering the DSP.

Analog inputs, which, for example, may include the pressure sensor for the accelerator or a vehicle speed detector, range anywhere between 0 and 12V. They could also have noise which needs to be filtered out. To address this, an input stage with a voltage divider and non-inverting op-amp is added for each analog input. The voltage divider decreases the input voltage to a range which the DSP can safely accept. The active op-

amp filter eliminates noise while ensuring that the input signal is not attenuated. The value of all the resistor and capacitor components can easily be adjusted by the user based on the profile of the specific input signal being used. Equation (2) show how values may be calculated for the first stage and (3) show how values may be calculated for the second stage.

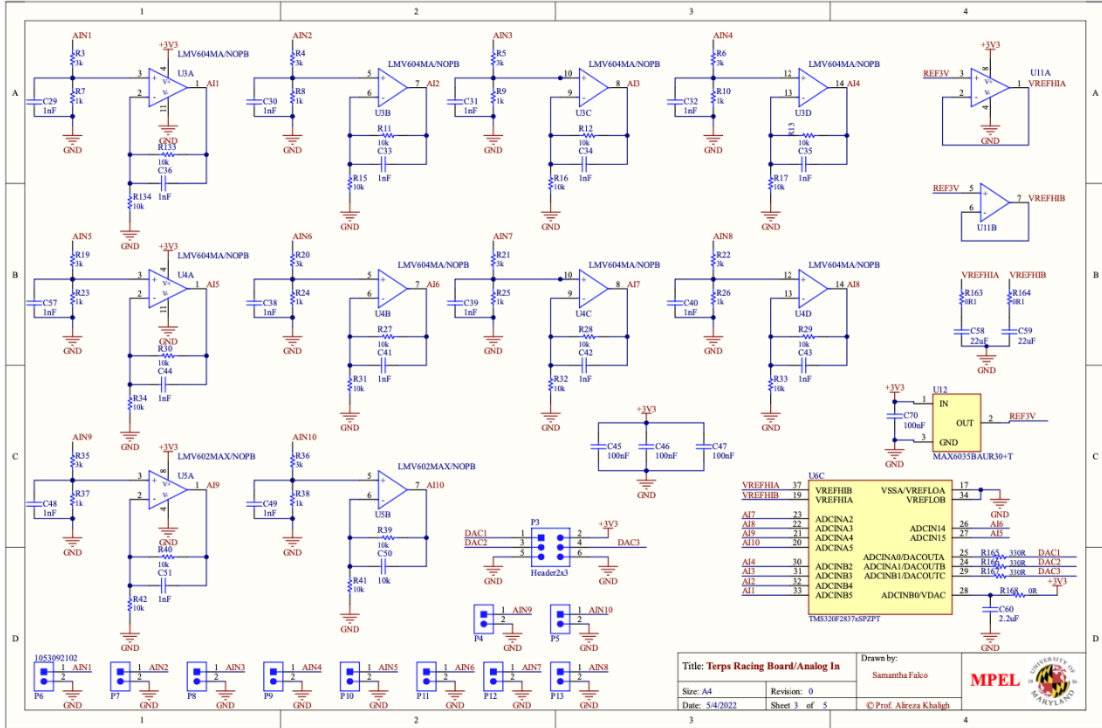


Figure 8: Analog Inputs Schematic

$$\frac{V_{out}}{V_{in}} = \frac{R_7}{R_3 + R_7} \quad (2)$$

where  $V_{out,max} = 3$   
and  $V_{in} = 0 - 12V$

$C_{29} =$  user choice, dependant on noise of sensor

$$\begin{aligned} R_{133} &= \text{user choice, small for unity gain} \\ R_{134} &= \text{user choice, large} \end{aligned} \quad (3)$$

$C_{36} = \text{user choice, dependant on noise of sensor}$

Digital inputs could include driver button pushes or other communication protocols and could range anywhere between 0 and 12V. Similar to the analog inputs, since the DSP can only handle voltages up to 3.3V, a digital inverter circuit with 3.3V collector rail voltage is implemented to scale the input voltages between 0 and 3.3V. A second digital inverter is added to the output of the first to re-invert the input voltage to its original polarity. Equation (4) show how values may be calculated for the digital inputs, stages one and two use the same component values so only stage one calculations are shown.

$$R_{45} = \frac{3.3V}{I_c}$$

where  $I_c$  is limited to minimize power loss

$$R_{49} = \frac{12V}{I_c} * \frac{\beta}{2} \quad (4)$$

where  $\beta$  is the current gain

$$R_{51} = 10 * R_{49}$$

to dissipate parasitic capacitance

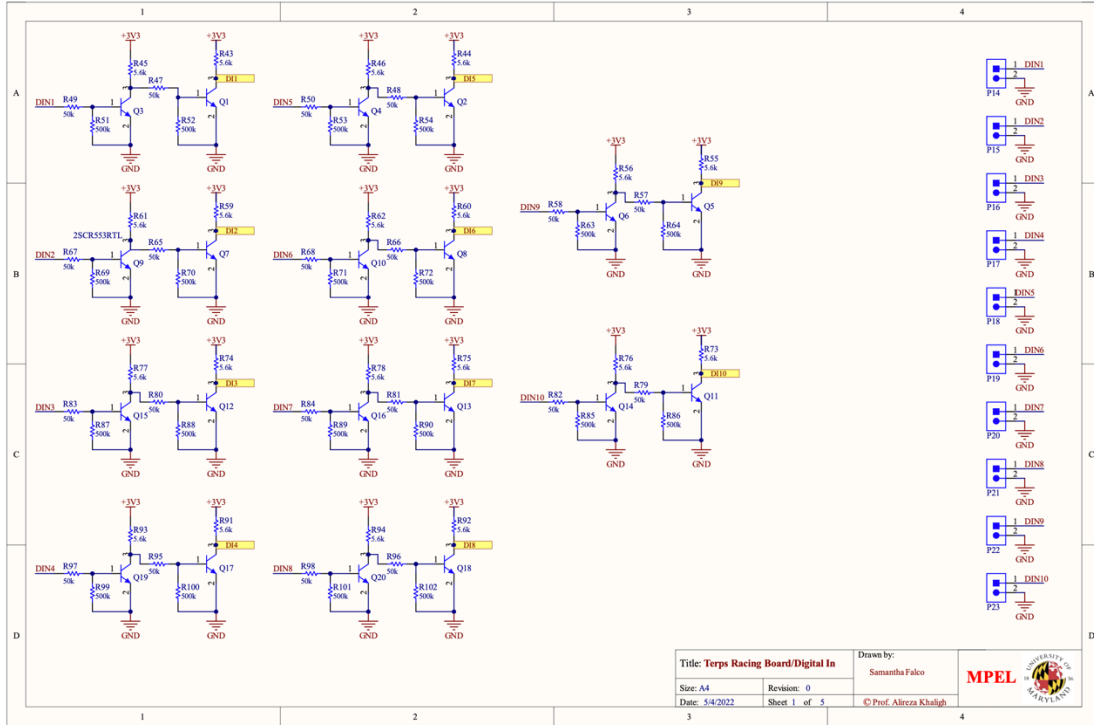


Figure 9: Digital Inputs Schematic

After processing information in the DSP, digital outputs may be used, for example, to activate user interface signals (like dashboard lights). These outputs may need to be any voltage from 0 to 12V, so an output stage is necessary to pull 0-3.3V DSP output voltages up to what is needed. This output stage is composed of two logical inverters, one NPN and one NMOS. The NPN inverter scales the signal to a 12V range and the NMOS stage re-inverts the signal. An NMOS inverter is used instead of an NPN inverter so that larger saturation currents can be driven at the output. Resistance values can be calculated using the same method as the digital inputs, as shown in (5).

$$R_{105} = \frac{12V}{I_c} \quad (5)$$

where  $I_c$  is limited to  $\sim 2\text{mA}$  to minimize power loss

$$R_{109} = \frac{3.3V}{I_c} * \frac{\beta}{2}$$

where  $\beta$  is the current gain

$$R_{111} = 10 * R_{49}$$

To dissipate parasitic capacitance

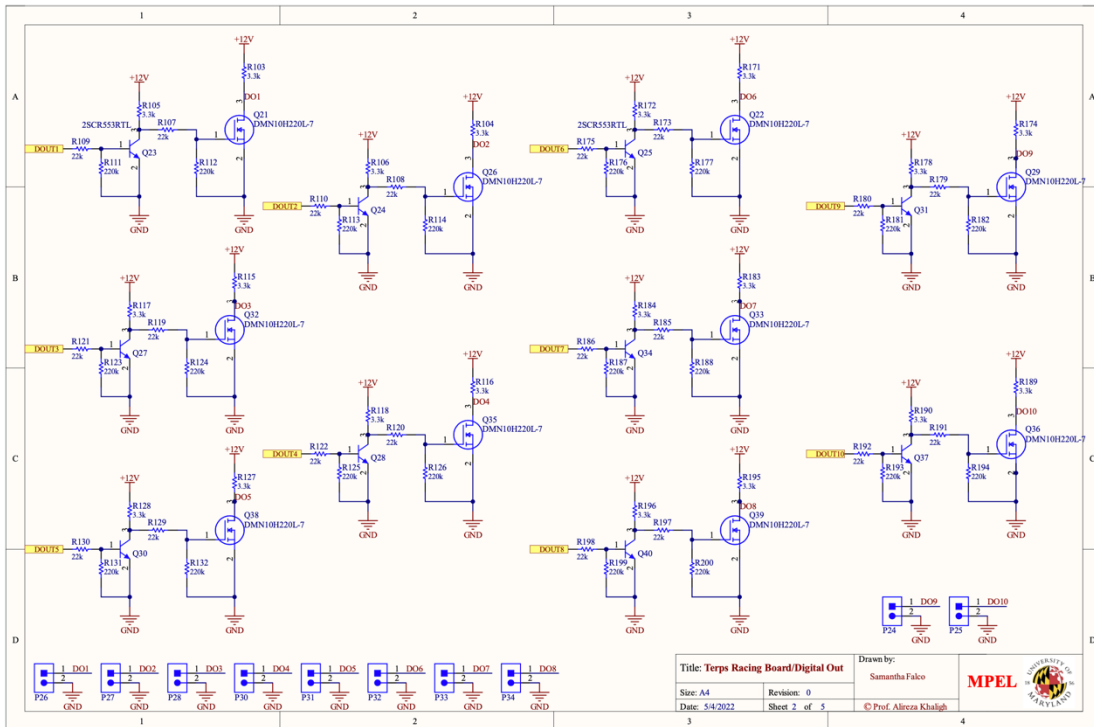


Figure 10: Digital Outputs Schematic

The final schematic is the digital and DSP schematic. It includes hardware for the following:

1. Most of the DSP pins
2. The two CAN transceivers with their terminating resistors
3. The Zigbee XBEE module
4. 4 test circuits (2 LED's, a push button, and a potentiometer)
5. The JTAG connector for debugging
6. The crystal oscillator

- 7. Additional GPIO connectors
- 8. Boot mode selector

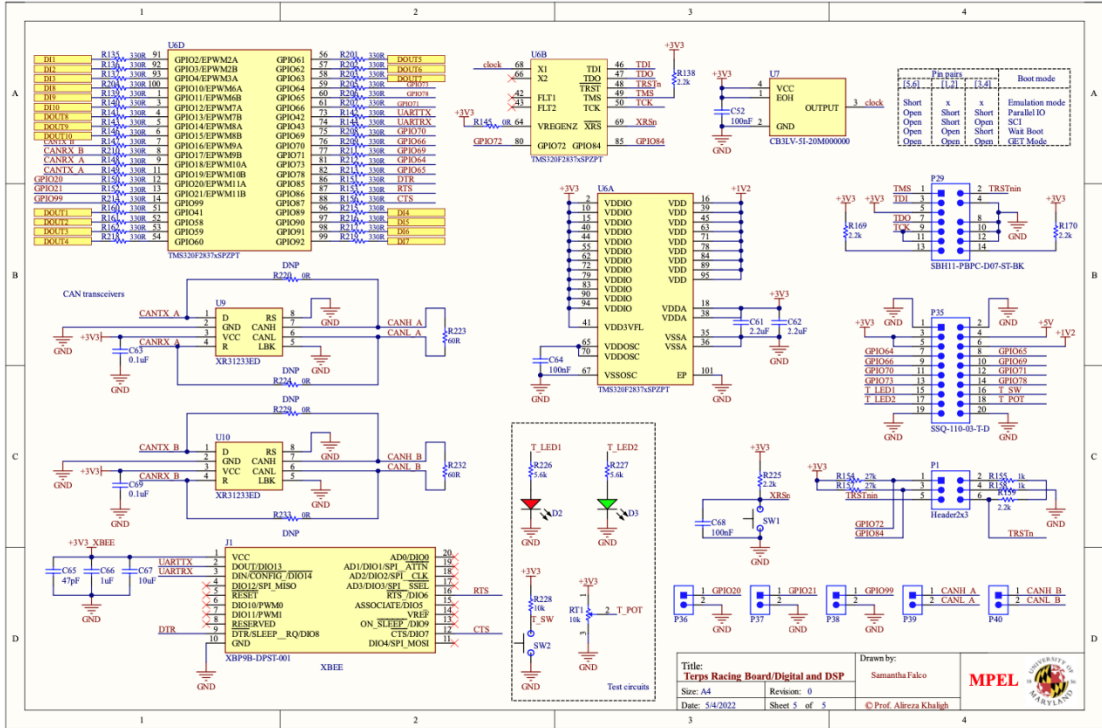


Figure 11: Digital and DSP Schematic

### PCB Design

Ease of access, readability, and reconfigurability were the main goals for the ECU board layout. All the connectors are located around the perimeter of the board and organized by type so they would be simple to use for people not intimately familiar with the layout of the board. All capacitors and resistors use the 1206 package to be easily solderable and re-solderable for prototyping and beginners. Figure 12 shows the 2D layout of the board. Table 3 is the key for the systems circled in Figure 12.

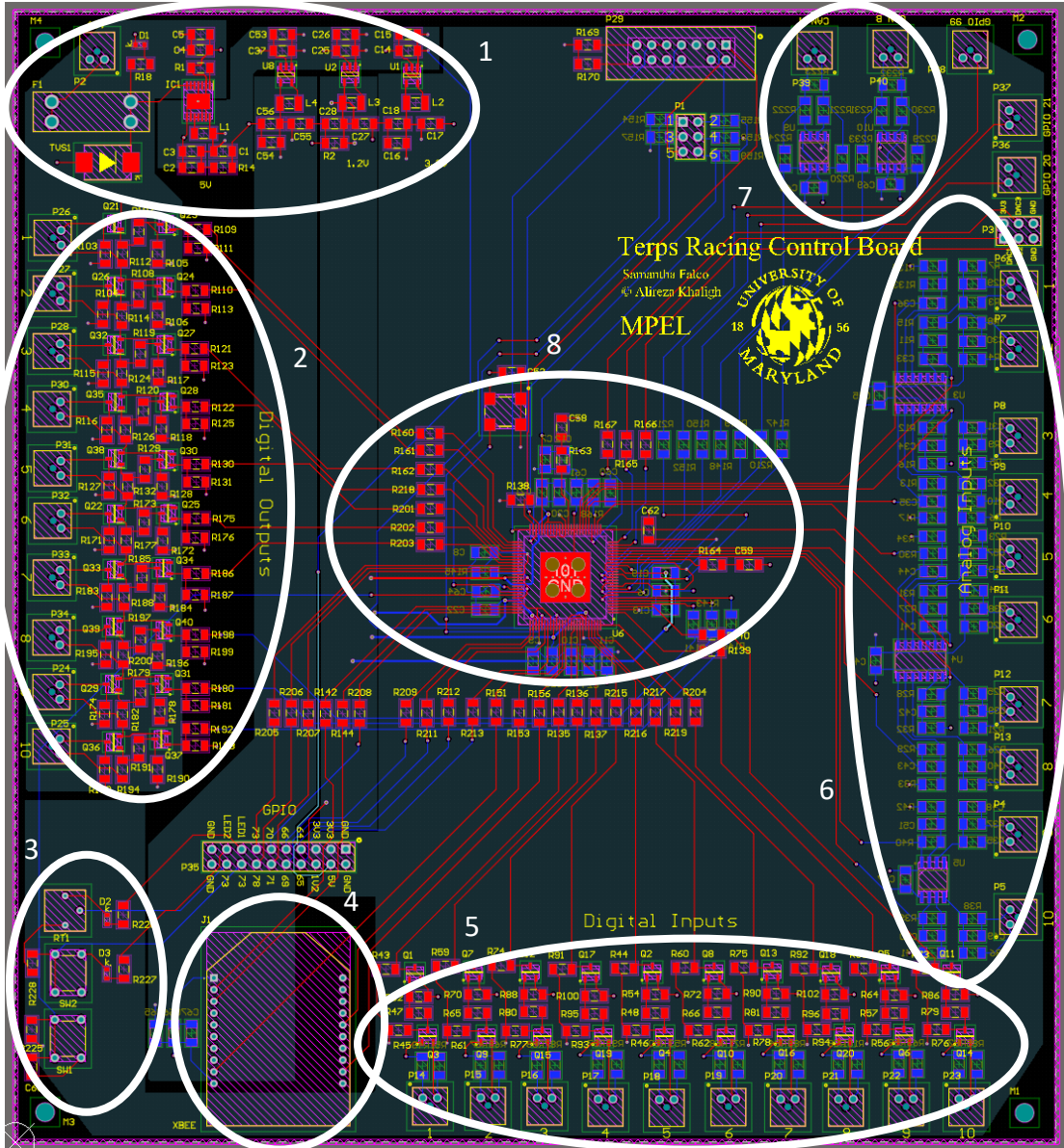


Figure 12: 2D Layout and Routing

Table 3: Layout Key

1	2	3	4	5	6	7	8
Power Supplies	Digital Outputs	Test Circuits	ZigBee Module	Digital Inputs	Analog Inputs	CAN Connection	DSP

### Software Structure

Software was developed to implement control functionalities and verify the operation of the hardware. A more detailed description of the test implementations of the software is provided in Chapter 4. There are 3 main systems that make up the software framework: CAN, ADC's, and SCI for the XBEE module. Figure 13 shows the basic software flow of a single board and Figure 18a shows how these systems work in the larger system.

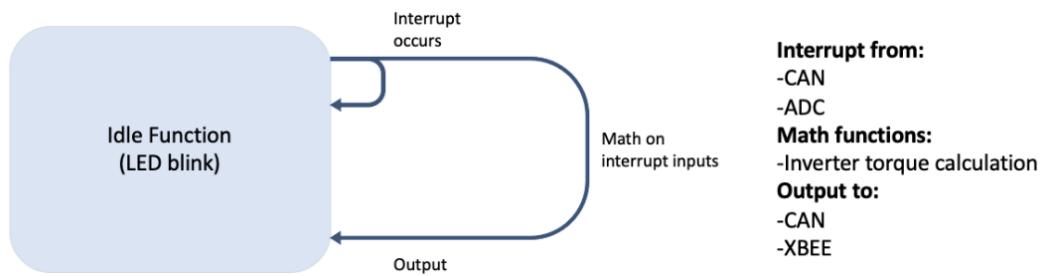


Figure 13: Basic software flow for each control board

Inputs from sensors come in an uninterrupted stream through the ADC's. An ePWM signal periodically samples this stream and saves the values in a buffer. Once all the ADC buffers are updated, a CPU interrupt is generated to apply the updated buffer values where they are needed elsewhere in the program.

Similarly, input messages are frequently being received by the CAN module. Upon receiving a new message, an interrupt is generated to use the new CAN message.

Math functions based on these inputs are then called. The results of these math functions, for example the torque value used to drive the inverter, may then be outputted from the board via the digital output pins or the CAN bus. The main means to communicate with other units within the EV's control structure is CAN. When

transmitting a CAN message is desired, the transmit data field is set and then the message is pushed to the CAN bus.

Up to this point, the XBEE module is only being used to transmit status information to a PC. Eventually the communication will go both directions. When an interrupt is generated due to one of the reasons described above, a message is transmitted to the XBEE module via the SCI hardware in the DSP. This message is then automatically wirelessly transmitted to another XBEE module connected to a PC. It is important that all the XBEE units and the SCI hardware are operating at the same baud rate. In this case, 9600 is used. Equation (6) below shows how to calculate the value to put in DSP's baud select registers (BRR).

$$BRR = \frac{LPSCLK}{(Desired\ baud * 8) - 1} \quad (6)$$

## Chapter 4: Experimental Verification

### Test Plan

Individual module functionalities as well as the capabilities of the distributed network are verified in stages.

1. Peripheral testing (CAN, XBEE)
2. Module functional tests (Analog in, digital in/out)
3. System experimental verification (Message transmission/reception, fault notification)

### Peripheral Testing

#### *XBEE Module Network*

Each peripheral capability is tested individually before combining them into the main test sequence. The left side of Figure 14 shows the set-up of two XBEE modules in DIGI's XCTU software. Each module is programmed individually before being integrated into a control board. Networks may be designed in the XCTU software. Two XBEE modules are used, each with an A24-HASM-450 antenna. The first, named "PC Receiver", is connected to a PC. Another, the "EV Transmitter", is simply powered on and placed 4 m away. The right side of Figure 14 shows this network, which was created to meet the desired test configuration. The line connecting the two modules on the right half of the image shows that they were successfully linked in the network.

XCTU boasts multiple features which test the functionality of the network. Most important of these include the radio range test, a spectrum analyzer. The radio range

test measures the RSSI, a measure of signal strength, of the device. Figure 15 shows the range test results. Better values are closer to zero. As [38] explains, a value of approximately -40dBm is excellent. Increasingly long distances between the two modules were tried. The module datasheet, [28], advertises a range of up to 90 m indoors and 1600 m (1 mile) outdoors. Eventually, remote monitoring, and even remote control of the board will be possible.

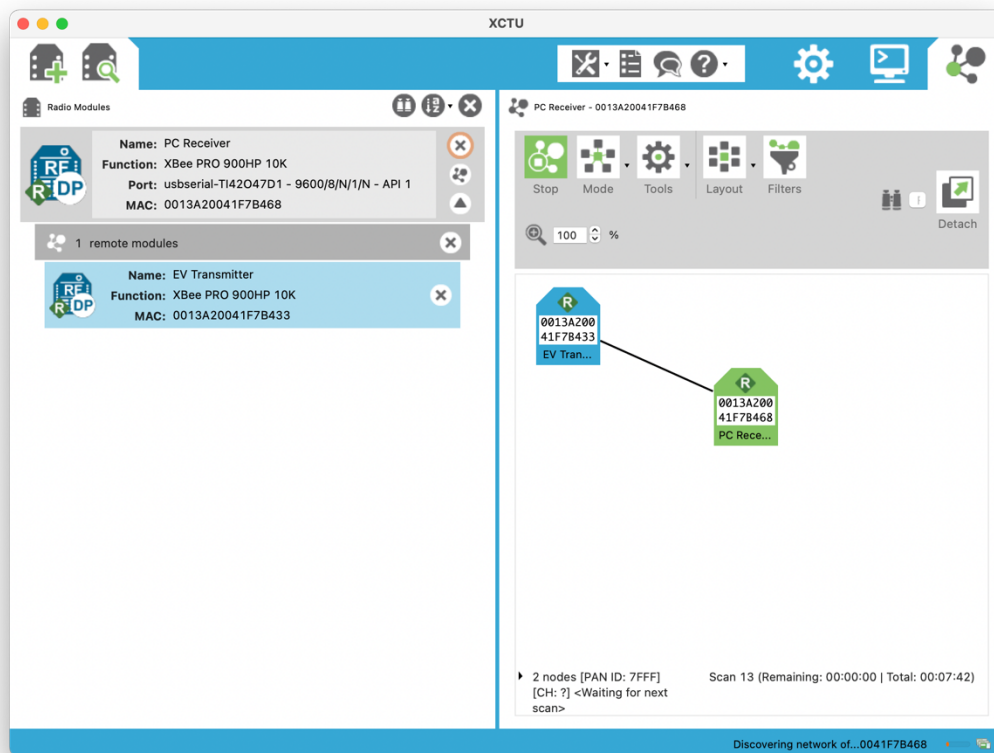


Figure 14: XBee Network Connection Verification



Figure 15: RSSI Test Results

### *CAN Bus Communication*

The control board’s CAN bus functionality is tested in two stages. First is internal loopback and then a multiple node bus. The internal loopback test ensures the hardware on the control module is working properly. Then the multiple node bus test ensures that node priorities can be effectively set and messages on the bus accepted by the proper node. Figure 16 shows the set up of the multiple node bus test. This set up includes two separate boards and two nodes. An example CAN message trace is shown in Figure 17.

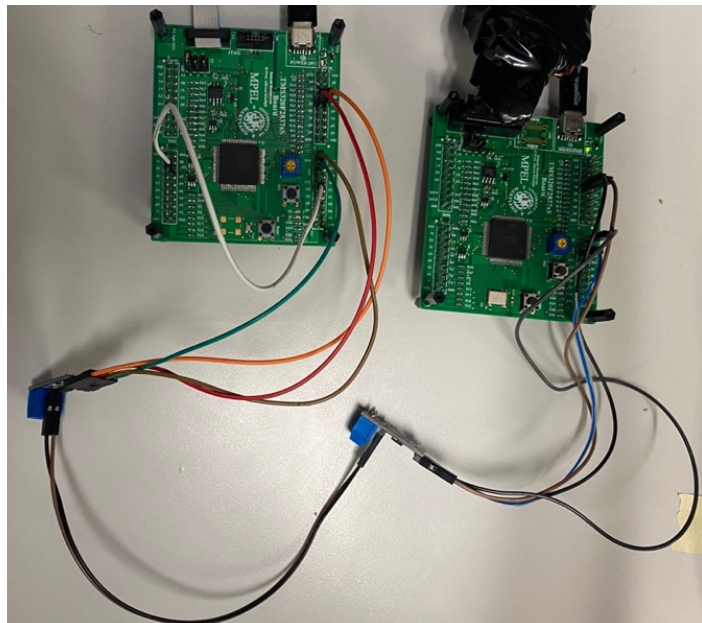


Figure 16: CAN bus software verification set up

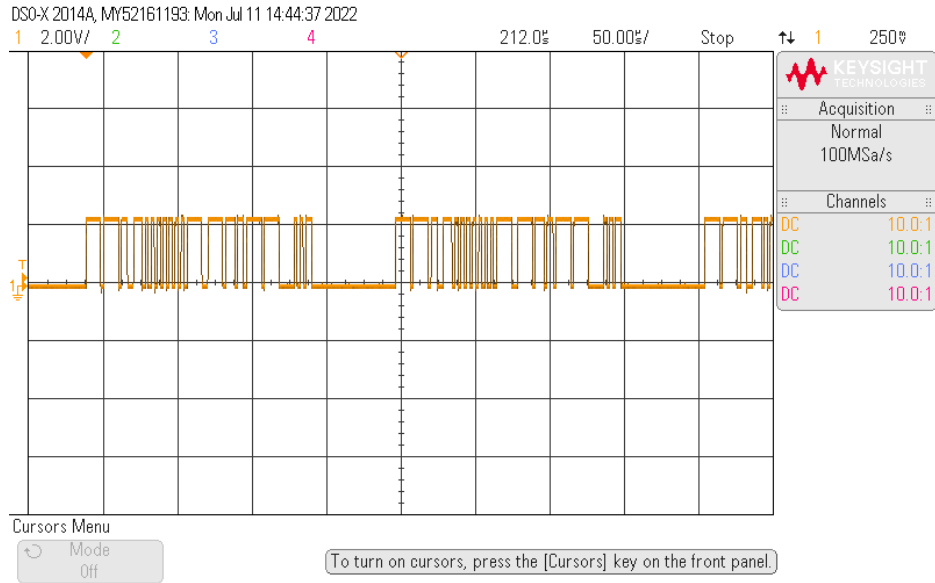
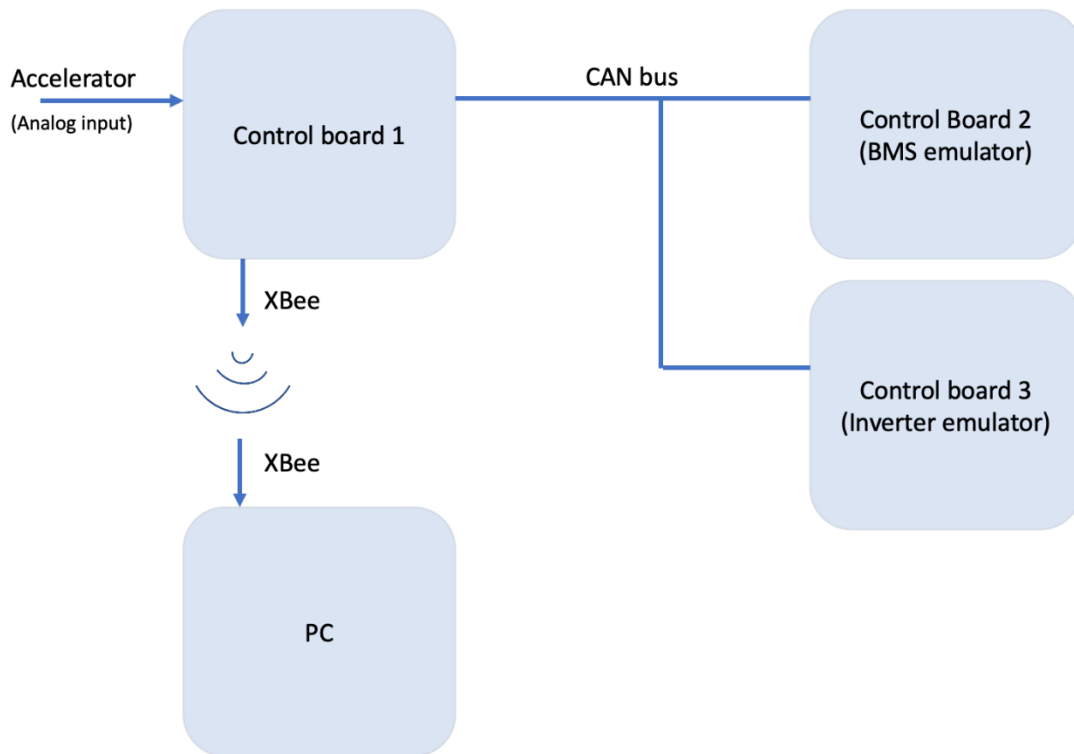


Figure 17: CAN message trace

### System Test Scheme

In addition to the basic proof-of-concept testing of all the peripherals, a test scheme was written to show how each module operates as part of a larger system. Figure 18a shows a block diagram of the functional implementation. Three control boards are used. One of the boards used is the design described above and the other two are simpler breakout boards. The breakout boards were used to decrease time before testing but eventually all three boards will be the same. The first (main) board takes in an analog input from the accelerator pedal and battery and battery status information from the CAN bus. It outputs instructions to the inverter board via CAN and transmits the system status wirelessly to a nearby PC via the XBEE module. The second board interfaces with the battery stack BMS to put the battery voltage and temperature onto the first CAN bus. The system's ability to communicate with multiple nodes on one CAN bus is tested with boards 1, 2 and 3. It acts as an emulator to the actual emulator which will be used and receives control commands from board 1 in the proper format. Figure 18b

is a photo of the test set up described above. In the photo of the test set up, two of the smaller green modules are used as placeholders until more of the blue “Terps Racing Control Boards” may be populated. These green boards have many of the same capabilities as the blue one and can work as a substitute in some scenarios. They do not, however have on-board level shifters for the input/outputs, connections for the XBEE module, or CAN transceivers. External CAN transceivers are added to these boards to communicate with the main control board.



(a)

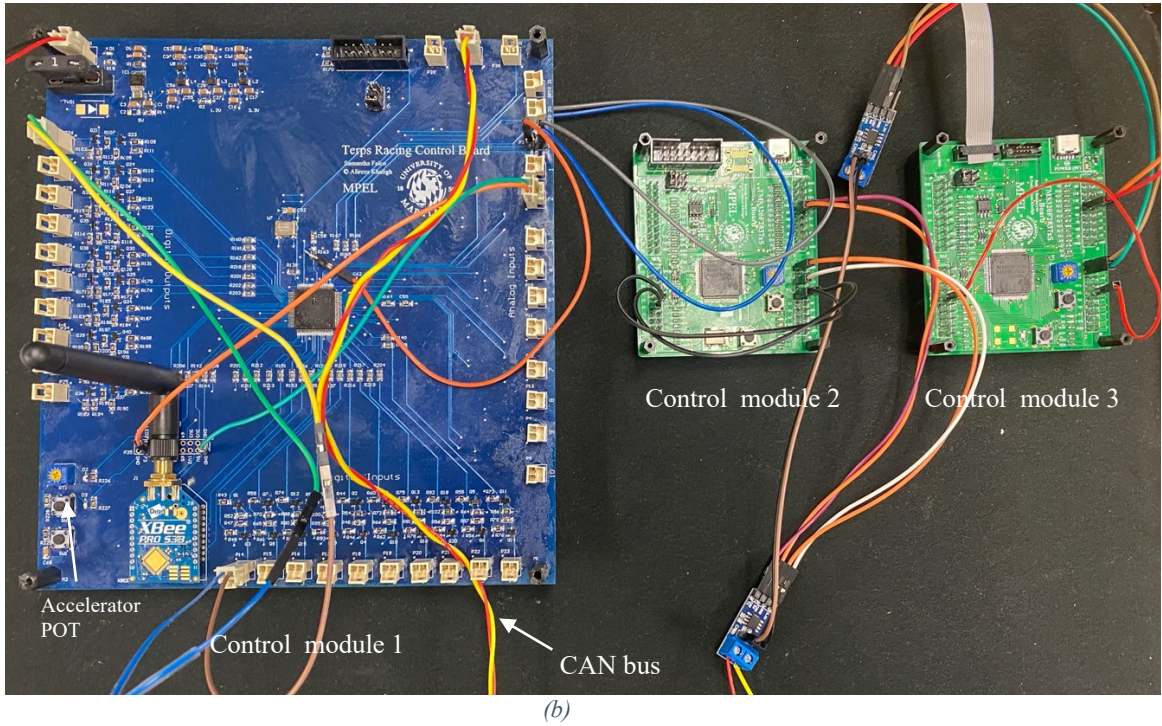


Figure 18: (a) Block diagram of distributed control test scheme (b) Distributed control test scheme implemented with hardware

The PWM waveforms generated for the inverter are determined by the voltages,  $V_a$ ,  $V_b$ ,  $V_c$ , across the switches in each phase of the three-level inverter. These voltages are determined using the equation below.

$$V = mV_{DC} \sin \omega t + \frac{V_{DC}}{2} \quad (7)$$

Here,  $V_{DC}$  is the voltage across the battery stack, as specified by the BMS. An EMRAX 228 medium voltage motor is used. The motor output is governed by its speed-torque curve, specified in [39] and shown in Figure 19. Torque is proportional to the value  $m$ , which can take a value between 0 and  $\frac{1}{2}$ . Speed (RPM) is determined by  $\omega$ , which can take a value between 0 and 120 Hz.

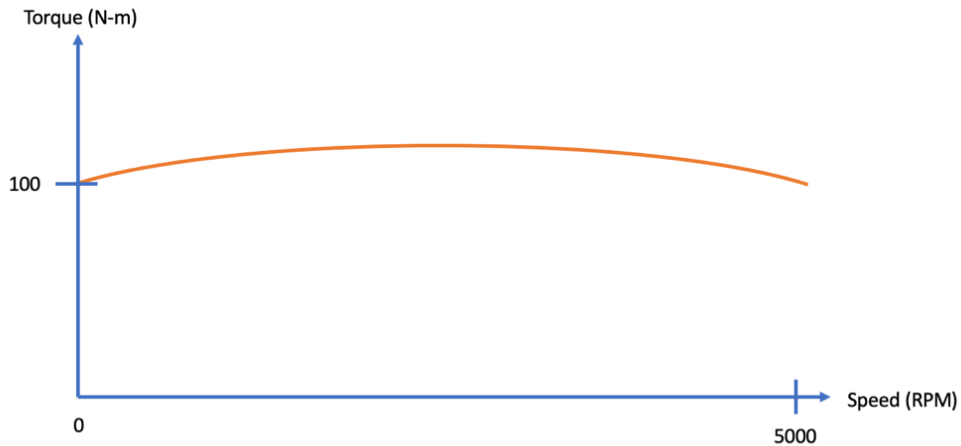


Figure 19: Permanent magnet synchronous motor speed-torque curve

The equations for  $V_a$ ,  $V_b$ ,  $V_c$  are calculated by the Cascadia PM150DX inverter based on information provided to the inverter from the main control module via the CAN bus. The inverter uses a specific data field formatting to communicate operation information, which is specified in [40]. This formatting is shown in Table 4. The inverter is set up to operate at the same baud as the other can nodes, 500kbps. The 11-bit CAN identifier, used to set priority, is set to 0xA0 (the default). CAN message priority is set using the CAN message ID. Data fields that use two bytes use the little-endian ordering.

Table 4: CAN Data Byte Set-up for Inverter

Data Byte	0	1	2	3	4	5	6	7
Command	Torque		Speed		Direction	Inverter Run	Torque Limit	

Table 4 shows how the data is broken down to be sent to the inverter. The inverter can be set up in either torque or speed mode. In torque mode, current is increased or decreased to meet the required torque. Voltage, associated with speed, changes according to the speed-torque curve for maximum efficiency. The opposite is true in

speed mode. Torque mode most closely aligns with how the traditional combustion engine vehicle operates. To make driving the EV as similar as possible to IC vehicle driving for the driver as possible, the inverter will be run in torque mode.

Calculating the torque to send to the inverter requires scaling the ADC input from the accelerator pedal. Based on the position of the accelerator pedal, the desired percentage of the maximum allowed torque is calculated. The value of the maximum torque is taken from the motor’s datasheet, [39]. The desired torque is then transmitted to the inverter using the CAN data formatting described in Table 4.

*Table 5: Battery BMS status CAN message format*

Data Byte	0	1	2	3	4	5	6	7
Command	Voltage	Temperature	-	-	-	-	-	-

*Table 6: Vehicle speed CAN message format*

Data Byte	0	1	2	3	4	5	6	7
Command	Vehicle Speed	-	-	-	-	-	-	-

### Success Metrics

The test structure described above is designed to verify the operation of the hardware, test the robustness of the software, and measure the overall performance. Several specific metrics of success were used to determine the board’s validity. These include the following:

1. Verification of CAN communication
2. Verification of wireless communication via XBEE modules
3. Demonstration of proper operation of digital input/output circuits

4. Demonstration of capacity to control inverter and motor with inputs from accelerator and BMS battery voltage information
5. Measurement of control speed. Goal time is 130ms

The above success metrics are not exhaustive in proving all the capabilities of the control board individually and as a component of the proposed distributed control system. However, they do show that the hardware is working as designed and should be able to support software enabling wireless communication, fault resilience, autonomy, and control based on external sensor inputs.

### Results

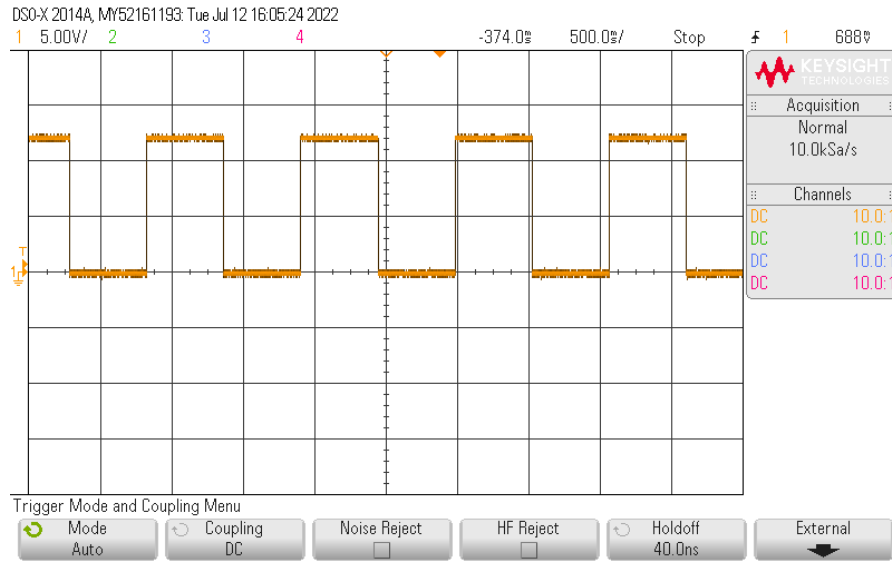
A full test plan for the control network and individual modules is described below. Images of the assembled module are provided in Appendix C. The module's physical capabilities and the control structure's potential for fault resilience were tested. Measurement of control speed, or how long it takes for the driver to feel control adjustments, will be completed once the accelerator pedal, BMS emulator, and inverter emulator are replaced by the actual hardware used by Terps Racing.

To start, one of the new, blue control boards was populated with components essential for testing. A XBEE module and antenna were connected to this board and an identical module and antenna were connected to a PC to receive Zigbee communications wirelessly. A potentiometer was connected to one of the analog input connectors to simulate analog input signals from an accelerator pedal. The control module was connected to the two other green breakout boards via CAN B to enable communication

between boards. A 12V source, analogous to that available in an EV battery, was connected to supply power to the whole board. Once all these connections were made, the code project built for this board was loaded. Main functionalities enabled by the software include CAN message reception from the BMS board, transmission to the inverter emulator board, generation of digital output signals, and monitoring of inputs from one digital input pin and the accelerator pedal analog input pin.

Set up of the two XBEE modules was completed using the guide provided in [40]. Good connection and data transmission between the two modules was verified, as shown in Figure 14 and 15. Remote datalogging of status data will be up to the user to implement as desired.

Digital output signals must be level shifted from the 3.3V output maximum of the DSP to the voltage required for the specific in-vehicle application using a two-stage level shifting circuit. The hardware was tested for 12V, the maximum output voltage. A periodic digital signal is generated by the DSP. Figure 20 shows the signal output after passing through the circuit to scale the voltage from 3.3V to 12V, as expected. This feature may be used to drive external actuators which require voltages higher than 3.3V.



*Figure 20: Digital outputs from DSP after voltage scaling up to 12V*

The module's ability to scale input voltages down to the 0-3.3V range acceptable for the DSP is tested by sending the digital output in Figure 20 into one of the digital input circuits. The digital input circuit uses two NPN inverters to scale voltages down from a maximum of 12V in amplitude and protect the DSP from over voltage. The signal, after scaling, is in the correct range but otherwise identical to the input from external sensors in the car. Figure 21 shows the signal in Figure 20 after passing through the digital input circuit. Proper operation of the digital input circuit allows high voltage signals from around the vehicle to be scaled down to a range where the DSP can read them.

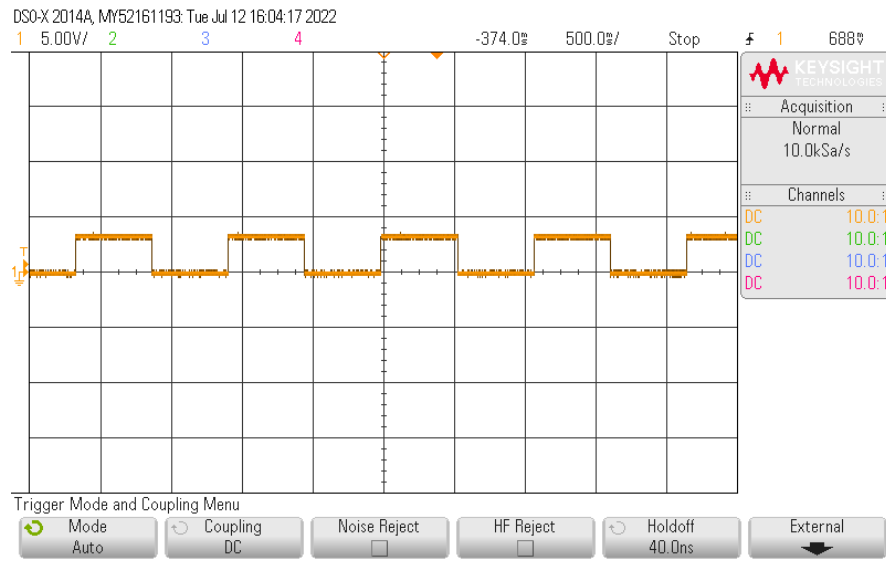


Figure 21: Digital inputs to DSP after voltage scaling of signal in Fig. 20

Next, board 2 is added to the network. Board 2 is set up to send battery and temperature status updates to board 1. It is powered by a 5V source and connected to the CAN bus. A different software project is loaded. This project internally generates battery voltage and temperature values to simulate those that will eventually come from the battery's BMS. These status values are encoded on CAN and sent to control board 1. If either of the temperature or voltage values exits the “safe” range, a warning is sent to module 1 to modify motor operation accordingly.

Finally, module 3 is added. Board 3 receives instructions from module 1 on the CAN bus about how to operate the inverter and motor. Just like module 2, it is powered by a 5V source and connected to the CAN bus. Software project three is designed to simply receive inverter operation messages from the CAN bus and display them on the console.

The addition of module 2 allows observations of the CAN bus to occur. Module 2 transmits one signal which board 1 receives, processes, and then sends out its own message to board 3. Only one message may be on the bus at any single time, but since status messages from 2 are continuously sent, two messages may try to enter the bus at any time. In this case, the message with the higher priority is put on the bus first, followed by the message with lower priority. Figure 22 shows two messages put on the bus one after another. The message on the left is from module 2 and has a smaller (shorter) data field. In response to prior messages from module 2, module 1 sent the message on the right, which has a larger (longer) data field. Module 1 has higher priority to put messages on the bus. Although only one message can be sent at a time, multiple types of messages may be put onto bus. Receiving modules must be properly configured so they only receive the information they need. Messages are currently sent and received properly, the user may decide the exact order of priority between messages on a module and between modules.

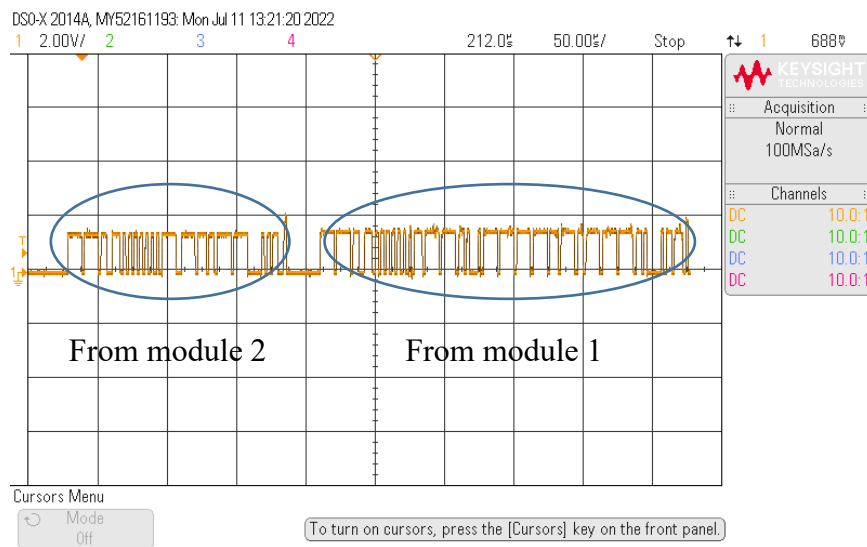
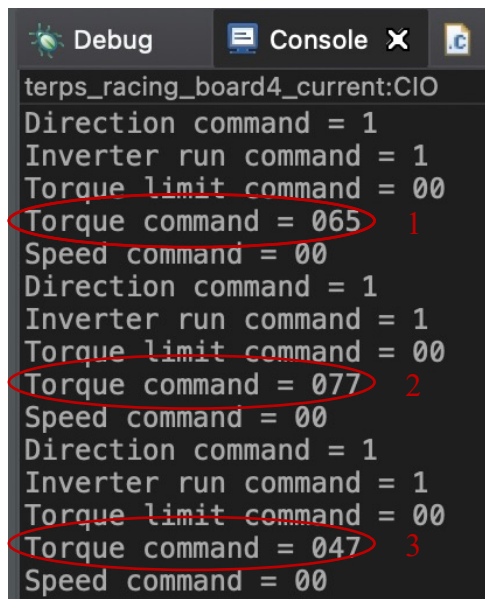


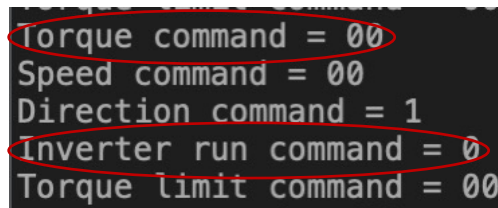
Figure 22: CAN message oscilloscope trace from two sources

Control of the inverter and motor with inputs from the accelerator pedal and BMS is possible using the current CAN priority settings. Messages received by module 3 from the CAN bus are printed to the console window sequentially. Figure 23 shows that the inverter torque command, as determined by the accelerator potentiometer on module 1, was correctly received on module 3. It also shows how the torque command updates in real time as the potentiometer position varies (red circles). Each iteration of “Torque command = xx” printed on the console is a new CAN message received. Thus, the inverter and motor can be controlled by new information from the driver and other systems across the vehicle.



```
terps_racing_board4_current:CIO
Direction command = 1
Inverter run command = 1
Torque limit command = 00
Torque command = 065 1
Speed command = 00
Direction command = 1
Inverter run command = 1
Torque limit command = 00
Torque command = 077 2
Speed command = 00
Direction command = 1
Inverter run command = 1
Torque limit command = 00
Torque command = 047 3
Speed command = 00
```

Figure 23: CAN message reception verification and torque updates with change in accelerator position



```
Torque command = 00
Speed command = 00
Direction command = 1
Inverter run command = 0
Torque limit command = 00
```

Figure 24: Inverter power off when BMS voltage status is low

Similarly, when a battery status warning is sent from module 2, module 1 determines that the inverter should turn off, and this update is reflected in board 3's console. The torque command goes from a positive integer in Figure 23 to zero in Figure 24. The inverter run command, which says whether the inverter is enabled or not, becomes zero. This shows how safety features may be built in to protect against fault propagation.

## Chapter 5: Conclusions

### Conclusion

A distributed controller architecture was proposed to replace the centralized ECU in a Formula SAE EV. The distributed control design is composed of multiple physically identical control boards whose individual functionalities are enabled with software and arranged in a network to monitor EV status and execute tasks. Functionalities of the board include CAN communication, wireless telemetry, analog inputs, digital inputs, and digital outputs.

Existing ICE and electric vehicles already use distributed control architectures. However, the generic design of the individual control modules allows for increased redundancy and reliability of the network and decreased hardware development time overall. This gives the proposed design a significant advantage over existing architectures. Future EVs will likely be shifting from a domain control approach (one controller per system) to a zonal architecture (one controller per region). Such zonal regions help reduce wiring weight and costs. The easily customizable and inherently localized design of the control module makes it well suited to a zonal architecture.

To validate the proposed distributed control architecture, a hardware prototype with three modules was designed and tested. Tests performed on the module prototype prove its functionality individually and as a component of a larger control structure. Integration of the module into the actual Formula SAE EV may begin. The module's ability to communicate with other iterations of the module via CAN was shown.

Information gathered from the BMS module as well as accelerator position on module 1 was successfully used to modify the torque command to the inverter module. Inputs from external sensors at a range of different voltages can be used as information sources for control decisions. Creating a network of physically identical control modules improves the EV's ability to eliminate failure modes. Such hardware redundancy will allow the user to dynamically shift control responsibilities based on the health status of each board with little physical change.

Other modular implementations of Formula SAE EV control hardware tend to use Arduinos for their control modules which often do not have the processing power necessary to perform all tasks. This custom design allows future students to be highly competitive in the Formula SAE competition and will become a platform for the exploration of futuristic EV features.

### *Future Work*

Next iterations will interface directly with the vehicle's inverter, BMS, and motor instead of emulator modules. Such integration will allow additional capabilities of the board, such as control time, to be measured. Edits to the schematic and PCB addressing some small flaws will be made in a second version of the module. Then the distributed control architecture will be integrated into the actual Formula SAE EV.


# Appendices

## Appendix A: Datasheets

### *XBEE Module*

**XBee®/XBee-PRO® RF Modules**

XBee®/XBee-PRO® RF Modules  
RF Module Operation  
RF Module Configuration  
Appendices



**Product Manual v1.xEx - 802.15.4 Protocol**  
For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001

**IEEE® 802.15.4 RF Modules by Digi International**

Digi International Inc.  
11001 Bren Road East  
Minnetonka, MN 55343  
877 912-3444 or 952 912-3444  
<http://www.digi.com>

9000982\_3  
2009.09.23

### *DSP*

**TEXAS INSTRUMENTS** TMS320F28375S, TMS320F28375S-Q1, TMS320F28375S-Q2, TMS320F28375S-Q3, TMS320F28375S-Q4, TMS320F28375S-Q5, TMS320F28375S-Q6, TMS320F28375S-Q7, TMS320F28375S-Q8, TMS320F28375S-Q9, TMS320F28375S-Q10, TMS320F28375S-Q11, TMS320F28375S-Q12, TMS320F28375S-Q13, TMS320F28375S-Q14, TMS320F28375S-Q15, TMS320F28375S-Q16, TMS320F28375S-Q17, TMS320F28375S-Q18, TMS320F28375S-Q19, TMS320F28375S-Q20, TMS320F28375S-Q21, TMS320F28375S-Q22, TMS320F28375S-Q23, TMS320F28375S-Q24, TMS320F28375S-Q25, TMS320F28375S-Q26, TMS320F28375S-Q27, TMS320F28375S-Q28, TMS320F28375S-Q29, TMS320F28375S-Q30, TMS320F28375S-Q31, TMS320F28375S-Q32, TMS320F28375S-Q33, TMS320F28375S-Q34, TMS320F28375S-Q35, TMS320F28375S-Q36, TMS320F28375S-Q37, TMS320F28375S-Q38, TMS320F28375S-Q39, TMS320F28375S-Q40, TMS320F28375S-Q41, TMS320F28375S-Q42, TMS320F28375S-Q43, TMS320F28375S-Q44, TMS320F28375S-Q45, TMS320F28375S-Q46, TMS320F28375S-Q47, TMS320F28375S-Q48, TMS320F28375S-Q49, TMS320F28375S-Q50, TMS320F28375S-Q51, TMS320F28375S-Q52, TMS320F28375S-Q53, TMS320F28375S-Q54, TMS320F28375S-Q55, TMS320F28375S-Q56, TMS320F28375S-Q57, TMS320F28375S-Q58, TMS320F28375S-Q59, TMS320F28375S-Q60, TMS320F28375S-Q61, TMS320F28375S-Q62, TMS320F28375S-Q63, TMS320F28375S-Q64, TMS320F28375S-Q65, TMS320F28375S-Q66, TMS320F28375S-Q67, TMS320F28375S-Q68, TMS320F28375S-Q69, TMS320F28375S-Q70, TMS320F28375S-Q71, TMS320F28375S-Q72, TMS320F28375S-Q73, TMS320F28375S-Q74, TMS320F28375S-Q75, TMS320F28375S-Q76, TMS320F28375S-Q77, TMS320F28375S-Q78, TMS320F28375S-Q79, TMS320F28375S-Q80, TMS320F28375S-Q81, TMS320F28375S-Q82, TMS320F28375S-Q83, TMS320F28375S-Q84, TMS320F28375S-Q85, TMS320F28375S-Q86, TMS320F28375S-Q87, TMS320F28375S-Q88, TMS320F28375S-Q89, TMS320F28375S-Q90, TMS320F28375S-Q91, TMS320F28375S-Q92, TMS320F28375S-Q93, TMS320F28375S-Q94, TMS320F28375S-Q95, TMS320F28375S-Q96, TMS320F28375S-Q97, TMS320F28375S-Q98, TMS320F28375S-Q99, TMS320F28375S-Q100

**TMS320F2837xS Microcontrollers**

**1 Features**

- TMS320C28x 32-bit CPU
  - 200 MHz
  - IEEE 754 single-precision Floating-Point Unit (FPU)
  - Trigonometric Math Unit (TMU)
  - Viterbi/Complex Math Unit (VCU-JH)
- Programmable Control Law Accelerator (CLA)
  - 200 MHz
  - IEEE 754 single-precision floating-point instructions
  - Executes code independently of main CPU
- On-chip memory
  - 512KB (256KW) or 1MB (512KW) of flash (ECC-protected)
  - 132KB (66KW) or 164KB (82KW) of RAM (ECC-protected or parity-protected)
  - Dual-zone security supporting third-party development
    - Unique identification number
- Clock and system control
  - Two internal zero-pin 10-MHz oscillators
  - On-chip crystal oscillator
  - Windowed watchdog timer module
  - Missing clock detection circuitry
- 1.2-V core, 3.3-V I/O design
- System peripherals
  - Two External Memory Interfaces (EMIFs) with ASRAM and SDRAM support
  - 6-channel Direct Memory Access (DMA) controller
  - Up to 169 individually programmable, multiplexed General-Purpose Input/Output (GPIO) pins with input filtering
  - Expanded Peripheral Interrupt controller (ePIE)
  - Multiple Low-Power Mode (LPM) support with external wakeup
- Communications peripherals
  - USB 2.0 (MAC + PHY)
  - Support for 12-pin 3.3 V-compatible Universal Parallel Port (UPP) interface
  - Two Controller Area Network (CAN) modules (pin-bootable)
  - Three high-speed (up to 50-MHz) SPI ports (pin-bootable)
  - Two Multichannel Buffered Serial Ports (McBSPs)
  - Four Serial Communications Interfaces (SCI/UART) (pin-bootable)
  - Two I2C interfaces (pin-bootable)
  - Analog subsystem
    - Up to four Analog-to-Digital Converters (ADCs)
      - 16-bit mode
        - 1.1 MSPS each (up to 4.4-MSPS system throughput)
        - Differential inputs
        - Up to 12 external channels
      - 12-bit mode
        - 3.5 MSPS each (up to 14-MSPS system throughput)
        - Single-ended inputs
        - Up to 24 external channels
    - Single Sample-and-Hold (S/H) on each ADC
    - Hardware-integrated post-processing of ADC conversions
      - Saturating offset calibration
      - Error from setpoint calculation
      - High, low, and zero-crossing compare, with interrupt capability
      - Trigger-to-sample delay capture
    - Eight windowed comparators with 12-bit Digital-to-Analog Converter (DAC) references
    - Three 12-bit buffered DAC outputs
  - Enhanced control peripherals
    - 24 PWM channels with enhanced features
    - 16 High-Resolution Pulse Width Modulator (HRPWM) channels
      - High resolution on both A and B channels of 8 PWM modules
      - Dead-band support (on both standard and high resolution)
    - Six Enhanced Capture (eCAP) modules
    - Three Enhanced Quadrature Encoder Pulse (eQEP) modules
    - Eight Sigma-Delta Filter Module (SDFM) input channels, 2 parallel filters per channel
      - Standard SDFM data filtering
      - Comparator filter for fast action for out of range
  - Configurable Logic Block (CLB)
    - Augments existing peripheral capability
    - Supports position manager solutions
  - Functional Safety-Compliant

▲ An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

# Inverter



User's Manual for Advanced Axial Flux Synchronous Motors and Generators

EMRAX 228 Technical Data Table (dynamometer test data)

Type	EMRAX 228 High Voltage			EMRAX 228 Medium Voltage			EMRAX 228 Low Voltage		
	AC	LC	CC	AC	LC	CC	AC	LC	CC
<b>Technical data</b>									
Air cooled = AC									
Liquid cooled = LC									
Combined cooled = Air + Liquid cooled = CC									
Ingress protection:	IP21	IP65	IP21	IP21	IP65	IP21	IP21	IP65	IP21
<b>Cooling medium specification</b> (Air Flow = AF, Inlet Water/Glycol Flow = WF, Ambient Air = AA) If Inlet WF temperature and/or AA temperature are lower, then continuous power is higher.	AF=20m³/h AA=25°C	WF=80m³/h inlet 30°C AA=25°C	WF=80m³/h inlet 30°C AA=25°C	AF=20m³/h AA=25°C	WF=80m³/h inlet 30°C AA=25°C	WF=80m³/h inlet 30°C AA=25°C	AF=20m³/h AA=25°C	WF=80m³/h inlet 30°C AA=25°C	WF=80m³/h inlet 30°C AA=25°C
Weight [kg]	12.0	12.3	12.3	12.0	12.3	12.3	12.0	12.3	12.3
Diameter ø / width [mm]	228/96								
Maximal battery voltage (Vdc) and full load/no load RPM	670 Vdc (1500/500 RPM)			675 Vdc (1570/500 RPM)			130 Vdc (4400/1200 RPM)		
Peak motor power at max RPM (flow min at cold start / few seconds at hot start) [kW]	100								
Continuous motor power (at 3000-5000 RPM depends on the motor RPM) [kW]	28-42	28-42	35-55	28-42	28-42	35-55	28-42	28-42	35-55
Maximal rotation speed [RPM]	5500 (6000 RPM peak for a few seconds)								
Maximal motor current (for 2 min if cooled as described in Annex) [A rms]	240			340			900		
Continuous motor current [A rms]	135			190			450		
Maximal motor torque (for a few seconds) [Nm]	240			340			900		
Continuous motor torque [Nm]	125			175			450		
Torque / motor current [Nm/A rms]	1.1			0.75			0.27		
Maximal temperature of the copper windings in the stator and max. temperature of the magnets [°C]	120			120			120		
Motor efficiency [%]	92-98			92-98			92-98		
Internal phase resistance at 25 °C [mΩ]	18			8.0			1.12		
Input phase wire cross-section [mm²]	10.2			15.2			38		
Wire connection	96°			96°			96°		
Inductance in mH/phase [mH]	177/183			76/79			10.3/10.6		
Controller / motor signal	Sine wave			Sine wave			Sine wave		
AC voltage between two phases [Vrms/288VAC]	0.0730			0.0678			0.0176		
Specific life speed (no load) [RPM/288VAC]	9.8			34			40		
Specific load speed (depends on the controller settings) [RPM/288VAC]	8-9.8			11-14			34-40		
Magnetic field weakening (for higher RPM at the same power and lower torque) [%]	up to 100			up to 100			up to 100		
Magnetic flux - axial [Tm]	0.0542			0.0555			0.0131		
Temperature sensor in the motor	by 81/210			by 81/210			by 81/210		
Number of pole pairs	10			10			10		
Rotor inertia [mass dia=175mm, m=5, kg] [kg·m²]	421			421			421		
Bearings (front-back) - SKF/FAG	6206-6206 (for radial forces) or 6206-7206 (for axial radial forces; for pull mode; e.g. for air propeller) or 6206-3206 (for axial radial forces; for pull-push mode, with orientation, m=20; other bearings are possible (exceptionally).								

## Appendix B: 3D Layout of Control board

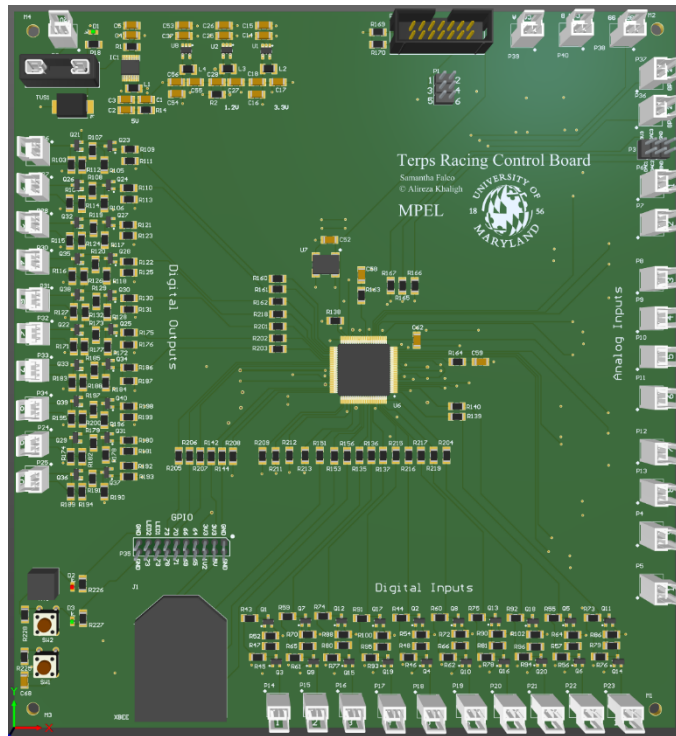


Figure 25: 3D Layout of Topside

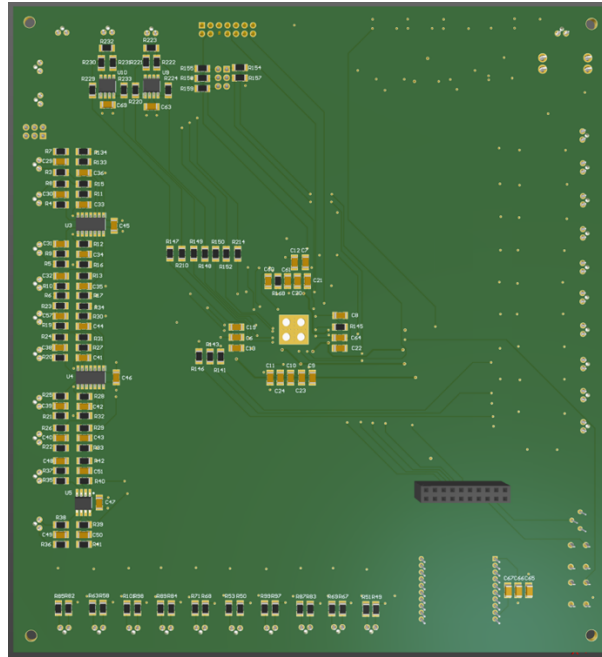


Figure 26: 3D Layout of Bottom side

Appendix C: Assembled Control Board

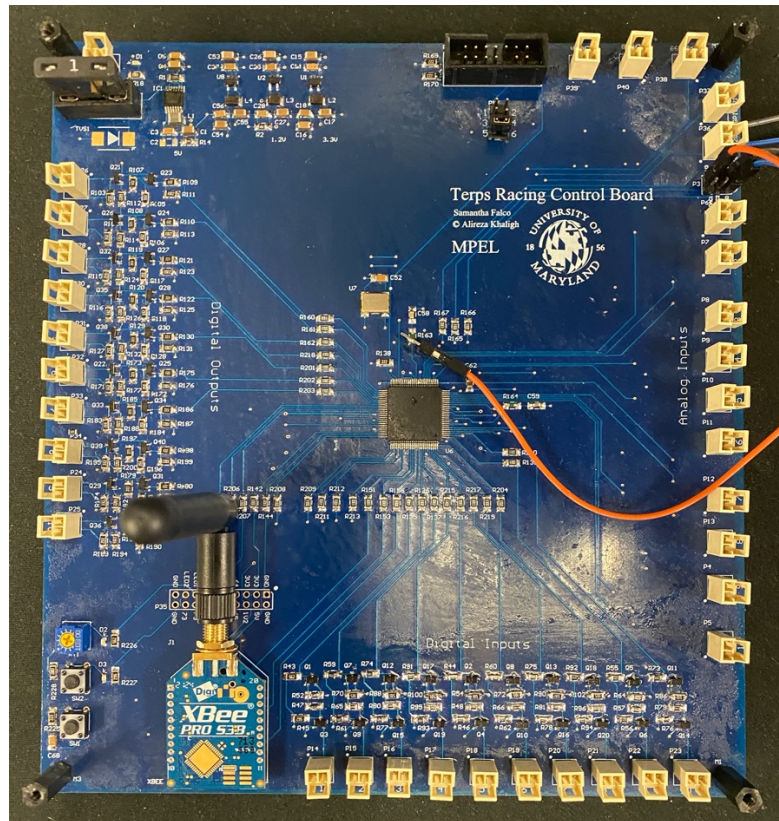


Figure 27: Assembled board, top view

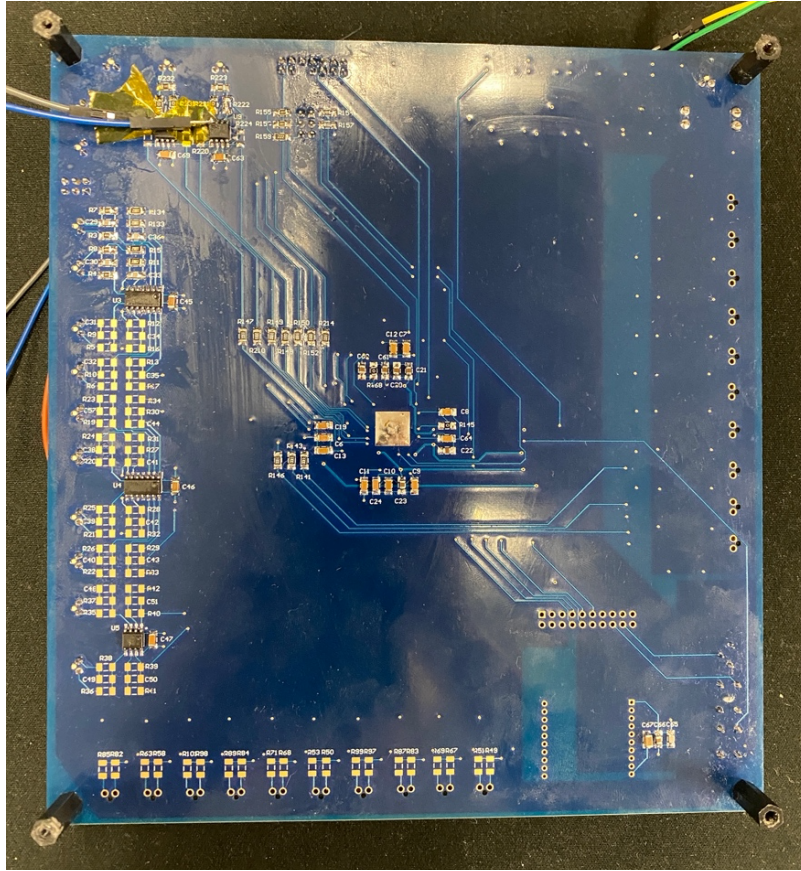


Figure 28: Assembled board, bottom view

## Bibliography

- [1] “Every Automaker’s EV Plans Through 2035 And Beyond - Forbes Wheels.” <https://www.forbes.com/wheels/news/automaker-ev-plans/> (accessed Jun. 01, 2022).
- [2] “Electric Vehicles Charge Ahead in Statehouses.” <https://pew.org/3AzSW4h> (accessed May 30, 2022).
- [3] “Insights into Future Mobility,” *MIT Energy Initiat.*, 2019, [Online]. Available: <http://energy.mit.edu/insightsintofuturemobility>
- [4] “Scaling EV infrastructure to meet net-zero targets.” Global Infrastructure Initiative, Oct. 2021.
- [5] S. Says, “How To Solve Automotive Electrical Design Challenges To Get To Market Faster,” *Semiconductor Engineering*, Jul. 02, 2019. <https://semiengineering.com/how-to-solve-automotive-electrical-design-challenges-to-get-to-market-faster/> (accessed Jun. 02, 2022).
- [6] “Formula SAE.” <https://www.fsaeonline.com/cdsweb/gen/DocumentResources.aspx> (accessed Jun. 28, 2022).
- [7] N. Pachhandara, “Council Post: Impacts Of The Global Chip Shortage And How To Prepare As The Backlog Stabilizes,” *Forbes*. <https://www.forbes.com/sites/forbestechcouncil/2022/05/19/impacts-of-the-global-chip-shortage-and-how-to-prepare-as-the-backlog-stabilizes/> (accessed Jul. 15, 2022).
- [8] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, “Battery Management System: An Overview of Its Application in the Smart Grid and Electric Vehicles,” *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 4–16, Jun. 2013, doi: 10.1109/MIE.2013.2250351.
- [9] M. Dennis, “Are We on the Brink of an Electric Vehicle Boom? Only with More Action,” Sep. 2021, Accessed: Jul. 12, 2022. [Online]. Available: <https://www.wri.org/insights/what-projected-growth-electric-vehicles-adoption>
- [10] “Paris Agreement Compatible Sectoral Benchmarks - August 2020,” p. 21.
- [11] R. Krishnamoorthy, C. Bharatiraja, and K. Krishnan, “Review of communication network interfaces and battery management for PHEV-ECU materials and components,” *Mater. Today Proc.*, vol. 45, pp. 3444–3448, Jan. 2021, doi: 10.1016/j.matpr.2020.12.933.
- [12] S. Deilami, A. S. Masoum, P. S. Moses, and M. A. S. Masoum, “Real-Time Coordination of Plug-In Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile,” *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 456–467, Sep. 2011, doi: 10.1109/TSG.2011.2159816.
- [13] E. Sortomme, M. M. Hindi, S. D. J. MacPherson, and S. S. Venkata, “Coordinated Charging of Plug-In Hybrid Electric Vehicles to Minimize Distribution System Losses,” *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 198–205, Mar. 2011, doi: 10.1109/TSG.2010.2090913.
- [14] “ECUs,” *E-Mobility Engineering*, Jun. 07, 2021. <https://www.emobility-engineering.com/focus-ecus/> (accessed Jun. 08, 2022).

- [15] “Global EV Outlook 2022 – Analysis,” *IEA*.  
<https://www.iea.org/reports/global-ev-outlook-2022> (accessed Jun. 07, 2022).
- [16] M. Lukaszewicz *et al.*, “System architecture and software design for Electric Vehicles,” in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, May 2013, pp. 1–6. doi: 10.1145/2463209.2488852.
- [17] “Controller Area Network (CAN) Tutorial.” National Instruments Corporation. [Online]. Available: ni.com
- [18] S. Corrigan, “Introduction to the Controller Area Network (CAN).” Texas Instruments, May 2016.
- [19] W. Dafang, N. Jinrui, and S. Fengchun, “The application of CAN communication in distributed control system of electric city bus,” in *2008 IEEE Vehicle Power and Propulsion Conference*, Sep. 2008, pp. 1–4. doi: 10.1109/VPPC.2008.4677533.
- [20] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016, doi: 10.1109/TIV.2016.2578706.
- [21] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving: Common Practices and Emerging Technologies,” *IEEE Access*, vol. 8, pp. 58443–58469, 2020, doi: 10.1109/ACCESS.2020.2983149.
- [22] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, “Autonomous Ground Vehicles—Concepts and a Path to the Future,” *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1831–1839, May 2012, doi: 10.1109/JPROC.2012.2189803.
- [23] X. Fan, X. Zheng, and Y. Ge, “CAN Communication Detection System Design for Engineering Vehicle Electronic Control Unit,” in *2020 15th International Conference on Computer Science Education (ICCSE)*, Aug. 2020, pp. 289–292. doi: 10.1109/ICCSE49874.2020.9201884.
- [24] K. L. Lim *et al.*, “A Modular Software Framework for Autonomous Vehicles,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 1780–1785. doi: 10.1109/IVS.2018.8500474.
- [25] Y. Yan, Y. Che, K. W. E. Cheng, and D. Wu, “Design and realization of the vehicle-mounted unit for a remote electronic monitoring and calibration system,” in *2009 3rd International Conference on Power Electronics Systems and Applications (PESA)*, May 2009, pp. 1–4.
- [26] A. Ferré, J. Fontanilles, D. Gàmez, and F. Giordano, “IWCM: Infrastructure Wireless Communication Module for vehicle communication with recharge infrastructure,” in *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, Nov. 2013, pp. 1–4. doi: 10.1109/EVS.2013.6914909.
- [27] Á. Rodríguez-Serrano, A. Torralba, E. Rodríguez-Valencia, and J. Tarifa-Galisteo, “A communication system from EV to EV Service Provider based on OCPP over a wireless network,” in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2013, pp. 5434–5438. doi: 10.1109/IECON.2013.6700020.
- [28] “XBee®/XBee-PRO® RF Modules,” Digi International Inc., 2009.
- [29] K. Meah, D. Hake, A. Smith, P. Hock, and A. Walsh, “Design and Implementation of the Software and Hardware System for a Formula SAE

- Electric Vehicle,” in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, Jun. 2020, pp. 1–5. doi: 10.1109/EEEIC/ICPSEurope49358.2020.9160530.
- [30] I. Chang, N. Kim, D. Lee, and S. W. Cha, “Designing and manufacturing of Formula SAE-Hybrid racecar for a new engineering education program,” in *2010 IEEE Vehicle Power and Propulsion Conference*, Sep. 2010, pp. 1–6. doi: 10.1109/VPPC.2010.5729097.
- [31] A. Tokosch, D. Hake, K. Meah, and J. Maier, “Design and Implementation of a Drivetrain for an FSAE Electric Vehicle,” in *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, Jun. 2019, pp. 1–4. doi: 10.1109/EEEIC.2019.8783402.
- [32] “TMS320F2837xs Microcontrollers.” Texas Instruments, Feb. 2021. [Online]. Available: <https://www.ti.com/product/TMS320F28374S>
- [33] “TMS320F2837xS Microcontrollers Technical Reference Manual.” Texas Instruments, Sep. 2019.
- [34] “XR31233 CAN Transceiver.” MaxLinear.
- [35] “OPTIREG™ switcher TLS4125D0EPV50.” Infineon, Feb. 05, 2021.
- [36] “LM3674.” Texas Instruments, Apr. 2015.
- [37] “SC4626 2.5MHz, 1A Synchronous Step Down Regulator in SOT23-5.” Semtech, Aug. 27, 2010. Accessed: Jul. 12, 2022. [Online]. Available: <https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001NqE/Nc90mUvekLadiqtxH4dvLCDesI1AmnPRXEDzwQ0eumU>
- [38] “Understanding RSSI Levels,” *MetaGeek*. <https://www.metageek.com/training/resources/understanding-rssi/> (accessed Jun. 27, 2022).
- [39] “EMRAX 228 Technical Data Table.”
- [40] “CAN Protocol Rev 5.9.” Cascadia Motion, Feb. 25, 2022.