# WISCONSIN RACING 223E ELECTRICAL DESIGNS





#### Table of Contents

Public Release Foreword	
Overview	
Correctness, Iteration and Rules Changes	
Custom System Notes	
Firmware & Motor Controls	
2023 Competition Results	
Incompleteness/ Systems Integration	
Special Thanks	
Questions/Contact Information	
Team Picture	
Car Picture	14
Battery Management System	
Overview	
Flowchart	
Battery Main Board	
Battery Central Board	
Overview	
Features	
Circuit	
Power regulation	
TM4C launchpads	
Current sensor ADC	
Shutdown Circuit	
Layout notes	
Schematics	
Digital Signals	
Connectors	
Shutdown Circuit	21
MCUs	
Power	23
High Voltage Daughter Board	24
Overview	24
Features:	24
High Voltage	
Hotswap Controller	
TSAL	
PCB Design	

Calculations	
Schematics:	
Top Level	
High Voltage	
TSAL Driver	
Hotswap Controller	
Connectors	
ISOSPI Board	
Overview	
Features:	
Circuit	
Smart Power Distribution Unit	
Overview	
Features:	
Parallel TPS25940 Automotive eFuses	
Functional Block Diagram	
Design Features and Design Decisions	
Load Testing Data	41
Design Errors and Rework	
IC and EV Differences: UVLO	
High Level Fusing Schematic	
TM4C123GH6PM Microcontroller	
CAN, Power Regulation and Connectors	
CAN	
Power Regulation and Protection	
Vehicle Controller Main Power Relay	
Connector, ground and other mechanical components	
PCB Layout and Stackup	
Stackup	
Calculations	
Schematics:	
Top Level Schematic	
eFuse Top Level Schematic	
Parallel eFuse	51
Embedded Microcontroller	
Power Regulation and CAN	
Brake System Plausibility Device	54
Overview	54
Features:	54

	BSPD Circuitry	54
	Window comparators	55
	NOR logic	55
	Gate Driver	56
	Layout Notes	56
	Summary	56
	Schematic	57
Ba	ttery Sensor Board	58
	Overview	58
	Features:	58
	LTC6811	58
	Operation	58
	Functional Block Diagram	59
	Device Utilization	59
	5V Voltage Regulator	60
	ISOSPI Communication	60
	Temperature Monitoring	61
	Noise Issues	61
	Layout Notes	62
	Layer Stackup	62
	Schematics	63
	Top Level	63
	LTC6811	64
	Balancing	65
	Auxiliary	66
Ba	ttery Trace Board	67
	Overview	67
	Features:	67
	Multiplexing	67
	Layout Notes	67
Ba	ttery Management System (BMS) Firmware	68
	Overview	68
	Features:	68
	Breakdown by Components:	68
	Real-Time Battery Readings	69
	Voltage Readings	69
	Temperature Readings	69
	The CAN Bus	70
	Overview:	70

Task Scheduling and Execution:	70
Enhanced Debugging Capabilities:	
Conclusion:	71
Screenshots of CAN Bus at work:	
BMS Software Design:	
Introduction	72
Scheduler	72
Data Structure	73
Algorithms	74
Utilizing Timer Interrupts	
Reporting Individual Cell over CAN	77
Charging	77
Balancing	
Faulting Matrix	79
Auxiliary Microcontroller (Aux MCU)	80
Overview	
Features:	
Controller Area Network (CAN)	
Transmitting and Receiving	
Fault Monitoring	
Digital Faults	
Analog Faults	
Data Monitoring	
DCDC Monitoring	
Operation	
Inverter Motor Controller	
WR2 3-Phase 30kW Electric Motor	
Overview	
Features:	
Control Card	
Overview	
Features:	
Circuit	
Input/Output interface	
Power regulation	
Microcontroller	
Power stage	
Clock and JTAG	
Communication peripherals	

GPIO pins	
Analog pins	
Enhanced Capture Pins	
Current sense amplifier	
Motor temperature amplifier	
Layout notes	
Schematics:	
Top Level	
I/O	
Voltage Regulation	90
Microcontroller	
Current Sense	
Gate Drive	
Overview	
Features:	
Low Voltage Side of Gate Drive	
Operation	
High Voltage Side of Gate Drive	
Fault Protection	
Gate Drive	
Isolated Power Supply	
Schematics:	
Top Level	
Power Module	97
Gate Drive	
DC Bus Sense	
Infineon FS200R07A5E3_S6	
Overview	
Features:	
Double Pulse Test	
Overview	
Double Pulse Test goals	
Test procedure	
Sizing Inductor, Capacitor, and Pulse duration	
Sizing Inductor and the length of Pulse 1 and 2	
Minimum Inductance	
Target inductance	
Duration of Pulse 3	
Creating inductor	

Sizing capacitor	
Voltage and current measurement	
Differential probe	
Rogowski coils	
Oscilloscope	
Data analysis	
Waveforms at 115A load current	
Turn on and turn off times	
Switching losses calculation	
Total Power loss	
Slew rates	
Closing notes	
DC Link Capacitance Selection	
Overview	
Features	
Selection parameters	
DC voltage rating	
Inverter Firmware/ Motor Controls	
Overview	
Features	
Controls	
Sensing	
Transforms	
Clarke Transform	
Park Transform	
Two Phase Current Sense	
Adjustment	
Modulation	
Actuation	
Flux Weakening	
Fault Protection	
Precharge Board	
Overview	
Features:	
Precharge Circuit	
Operation	
Low Voltage Circuitry	
High Voltage Circuitry	
HV Indicator LED Circuit	

## WISCONSIN

Layout Notes	
Stackup	
Schematic	
PCB Printout	
Discharge Board	
Overview	
Features:	
Circuit	
Schematic	
Steering Wheel	
Overview	
Features	
Primary Power Stages	
Reverse-Voltage and ESD Protection	
5V Buck Regulator	
3.3V Supply	
CAN Interface	
I/O Expanders	
Digital GPIO Expander	
Analog-to-Digital Converter	
LCD	
Data Signals	
Backlight Driver	
Primary Connectors and Interfaces	
Vehicle I/O Connector	
Daughter Board Connector	
External USB Connector	
MicroSD Card Slot	
Debug Connectors and Interfaces	
HDMI Port	
WiFi Support	
System Control Jumpers and Indicators	
CAN Debug Port	
Main Board Schematics	
Power	
System Interfaces	
I/O	
Display	
Steering Wheel Daughter Board	

Push-Buttons	
Multi-Switch	
Potentiometers	
LED Indicator Strip	
Daughter Board Schematic	
Steering Wheel Code	
Overview	
Features:	
Operating System Configuration	
Config.txt	
Cmdline.txt	
Backend	
CAN Bus	
MCP CAN Controller	
CAN Rx and Tx	
Internet Connectivity	
GPIO	
Display	
Tkinter	
CAN Struct	
Systems Integration	
Harnessing	
Features:	
Front Harness	
Rear Harness	
Penthouse Harness	
Harness Schematics	
Front Harness	
Rear Harness	
Penthouse Harness	
Datalogging	
Low Voltage Load Characterization	
Results	
Inverter Switching	
Brake Light	
SDC	
Mechanical Characterization	
EMI	144
Motor Encoders	



Brake Plausibility Device	145
Tractive System Efficiency	146
Battery Pack Efficiency	146
Motor Efficiency	146
Inverter Efficiency	146
Combustion Car Designs	147
Overview	147
Combustion Specific Designs	147
Shared Designs - Only Found in 2023 ECar Design Release	147
Contact Us	148

## Public Release Foreword

#### Overview

This is an overview of the electrical designs for the 2023 Electric FSAE car from Wisconsin Racing (UW-Madison). This "design binder" is what we prepared for the competition's design event. Wisconsin Racing's previously-released design documentation on our website (https://www.wisconsinracing.org/e-documents/) has been a useful reference for other teams looking to design custom hardware. In that spirit, we are releasing the designs of the entire electrical system of our 2023 vehicle, including our BMS, inverter, steering wheel electronics, LV power distribution, and more. Additionally, we have a combustion car. View the end of this document for more details, or the full combustion car design binder here: https://www.wisconsinracing.org/e-documents/

#### Correctness, Iteration and Rules Changes

This is a snapshot of the state of these designs last year, and all of them are being improved this year. Therefore, the correctness and rules compliance of these designs cannot be guaranteed. Some designs call out problems found within the design, just referencing schematics may lead to missing those callouts.

#### **Custom System Notes**

Wisconsin Racing has a highly custom electronics system, and it has not been sunshine and rainbows to get to this point. It has been a 7-year team effort. If you're a new team, it is more important to have a car that moves than a car with a bunch of custom PCBs. Don't underestimate the challenge of designing reliable and rules-compliant hardware.

#### Firmware & Motor Controls

The firmware and motor controls have parts of this binder that go over their design/ overarching themes. However, the firmware is not provided. This may not give weight to the complexity of custom firmware. For each system that required firmware, the firmware took the same amount if not more time to develop, integrate and test. Keep that in mind when deciding to do a custom system. Sometimes the custom PCB design isn't that challenging, but the firmware will require major amounts of time and resources.

Event	Placement
Design	13th
Acceleration	9th
Skidpad	13th
Autocross	17th
Efficiency	6th
Endurance	DNF (17/22 laps completed)
Overall	13th

#### 2023 Competition Results

#### Incompleteness/ Systems Integration

This document is incomplete in that it is missing many details of integration, manufacturing and testing. The document is mostly focused on the design of the custom PCB systems on the car, which is not what makes a team win. Wisconsin placed 13th overall in the 2023 season after faulting out in the 17th lap of endurance due to a single broken shielding wire. Reliability and a well tested car wins competitions, not the fancy circuit boards.

#### Special Thanks

This document and the 2023 race car would not be possible without the many contributions of the engineers who worked on it and the alumni that set the stage for this car. The engineers that started H3X (<u>https://www.h3x.tech/</u>), Jason Sylvestre, Max Liben and more, set the foundation for this entire system. I would also like to thank the mentors from the 2022 season, Quinn Sabin, Kai Linsenmeyer and Evan Wildenberg. Special thanks to Alex Grotelueschen (High Voltage Lead) as he helped spearhead the creation, formatting and push for this document. Isaac Hewett was my counterpart on the mechanical side, and none of this would be possible without the car to put the electronics in (be nice to your mechanical engineers). Finally, thank you to the subsystem leads and all the team members that put in a large amount of effort to make this document and this car:

- Alex Grotelueschen, High Voltage Lead
- Calvin Geishirt, Low Voltage Co-lead
- Pratham Sarvaiya, Low Voltage Co-lead
- Shrey Patel, Controls Co-lead
- Surya Raghavendran, Controls Co-lead



- Eric Li, Firmware Co-lead
- Sean Staggs, Firmware Co-lead

Additionally, thank you to our sponsors who can all be found on the website. The sponsors take the idea of a car and help turn it into the real thing. Finally, thank you to our faculty advisor, Glenn Bower whose wisdom and assistance has kept these cars being built year after year.

#### **Questions/Contact Information**

From my personal experience, being in charge of the electrical system of a high voltage racecar is an extraordinary challenge, and documentation may not be readily available. If you have questions, feel free to email me at <u>c.schusterr@gmail.com</u> or add and message me on LinkedIn. I may not have the answer, but I will attempt to find someone who does, or provide some direction.

Thank you and good luck designing, Christian Schuster 2023 Electrical Technical Director for Wisconsin Racing

### **Team Picture**





## Car Picture



## **Battery Management System**

#### Overview

The Wisconsin Racing Battery Management System is an entirely student designed system. This system consists of seven different custom printed circuit boards and three programmable microcontrollers with student written firmware. In the accumulator live the Battery Trace Boards (BTBs) and Battery Sensor Boards (BSBs). One pair of each can monitor one segment worth of cell voltages and temperatures. With seven segments, it adds up to 14 PCBs. The other five PCBs live in the "Penthouse", the inverter control box, and are in a mother/daughter board configuration we call the Battery Main Board (BMB). Together, these boards are able to monitor all 308 battery cells, 77 cell temperatures, and make decisions based on the received information.



#### Flowchart



Battery Main Board



## **Battery Main Board**

Engineers: Christian Schuster, Pratham Sarvaiya & Ethan Bender

The Battery Main Board system consists of five custom printed circuit boards and two TM4C launchpads. The motherboard, called the Battery Central Board, sits on the bottom, and all daughter boards connect on top with board-to-board connectors. Each daughter board hosts different peripheral circuits, to decrease the complexity of the overall system, and allow ease of future revisions. Two MCU's are used, one hosts the BMS code, the other handles auxiliary faults. This system is quite large as it is the initial revision for a modular LV system, and was designed for debuggability and serviceability versus a compact form factor.



The Battery Main Board Assembly



## **Battery Central Board**

Engineers: Christian Schuster, Electrical Technical Director and Pratham Sarvaiya, LV Electronics Lead Revision: 2023.1 Dimensions: 115mm x 242mm Steady State Current Draw: 70mA

#### Overview

The Battery Central Board is the primary multi purpose board that houses several electrical systems in the car. Forming the base of the Battery Management System, this board is located in the "Penthouse" (motor controller box) next to the Inverters, and provides a centralized location for all subsystems to come together.

#### Features

- Houses SPDU, HV Daughter board, ISOSPI board, and the **BSPD**
- 2 TM4C TI launchpads
- Power regulation
- Shutdown Circuit
- Connector breakouts
- CAN interface

#### Circuit

Multiple sub circuits have been mounted onto this board. The primary circuitry includes the Shutdown Circuit, 2 TM4C launchpads, and multipurpose connectors.

#### Power regulation

Switching regulators and LDOs are used to step down from 12V to 5V to power the board. Reverse voltage protection, using a P-Channel MOSFETs, protects circuits while testing. A zener diode is used for

transient voltage regulation. Bulk capacitors are added to decouple circuits and meet transient current demands..

#### TM4C launchpads

2 launchpads from Texas instruments, each featuring a TIVA TM4C123GH6PM microcontroller, have dedicated roles as part of the central board. One is used for interacting with other circuits in the Battery Management system, and the other is used as an

Battery Main Board







Reverse voltage protection



auxiliary for multiple subsystems across the car. The following table describes the division of their functions:

BMS launchpad	Auxiliary launchpad	
BMS CAN bus	Auxiliary CAN bus	
ISOSPI communication	Shutdown circuit fault monitoring	
DC pack current sensing	Hot Swap control	
300V to 12V DC converter control	IMD monitoring	

#### **Current sensor ADC**

A 16 bit Analog to Digital converter has been used to measure the DC current sense signal. A highly accurate 5V reference generator has been chosen to provide a reference to the ADC, which communicates with the serial synchronous interfaces in the launchpads. Decoupling capacitors placed on the 5V reference and supply rail fulfill the transient current requirements of the ADC and ensure a stable reference voltage. An RC filter has been added to the current sense signal to filter any high frequency noise that might corrupt the signal.

#### Shutdown Circuit

The shutdown circuit features DPST relays to carry the shutdown circuit current. An optocoupler connected to the Reset button latches the shutdown circuit. DPST relays have low resistance and provide the high current requirements for the AIRs

#### Layout notes

The PCB layer stackup has ground and power planes to provide a low impedance supply rail to all subcircuits.

Thick traces have been used for high current paths.

Decoupling capacitors are placed close to the subcircuits.

Differential pairs have been routed with minimum gap to reduce noise.











Layer stackup













lı g

IOUT Max: 800mA (Fused at 750mA though) Input Diode is to provent reverse current when 5V is provered externally on the bench











## High Voltage Daughter Board

Engineers: Christian Schuster, Electrical Technical Director and Ethan Bender, Electrical Hardware Team Member Revision: 2023.1 Dimensions: 109.5mm x 63mm Steady State Current Draw: 100mA

#### Overview

The High Voltage Daughter Board is part of the overarching Battery Management System and is the only board with High Voltage components on the Battery Central Board. It houses a high voltage sensing transducer, the HV to LV DCDC converter and the driving circuitry for the TSAL.



#### Features:

- 300W isolated 300V to 12V DCDC converter
- High voltage bus sense transducer
- TSAL Driver circuitry
- Hotswap controller for:
  - Current and power monitoring
  - OVLO and UVLO
  - Reverse current and voltage
  - dvdt voltage ramp
  - Overcurrent protection

#### High Voltage



#### High Voltage Sensing Transducer:

The LV25-P is a voltage sensing transducer from LEM. It uses the relationship of current between the primary and secondary windings in order to create a current source. This current source in conjunction with R10, the sense resistor, creates a readable voltage. This voltage is used to set the 60V sensing threshold for the TSAL. This voltage is also used by the BMS to read the high voltage bus voltage inside of the penthouse. The calculations for solving for the resistor values, the 60V voltage threshold and the conversion from high voltage to ADC counts in the BMS microcontroller can be found in the calculations section.

#### **DCDC Converter:**

The DCDC converter is the DCM4623TD2N17C8T00 from Vicor. The converter is capable of up to 30A peak continuous current out, has a peak efficiency of 92.2% and features temperature, overcurrent, overvoltage and undervoltage protections. It was selected due to its small form factor, ease of use and high power rating, as it is imperative that the DCDC converter can produce enough current to supply the entire low voltage system. The DCDC converter is connected in parallel with the low voltage battery so it will charge the GLV battery while the car is at high voltage. The converter features a trimmer potentiometer that allows for setting the output voltage in a range from 13V to 16V. Additionally, voltage will ramp down as more current is drawn from the device. The device is enabled and disabled via an optocoupler.

The signal is from the BMS microcontroller and it enables the DCDC converter five seconds after entering high voltage and when the battery charger is not connected.

#### Thermal Considerations:

The thermals of the DCDC converter and heat sink selection are crucial to proper usage of the converter. As the DCDC converter could theoretically be connected in parallel with an over discharged GLV battery, high amount of current independent of the system's current draw could happen. A load condition of 20 A with an output voltage of 11V was considered, with the datasheet efficiency at this condition. This condition was considered assuming there was an over discharged battery, and a current limit as to not over charge the battery is set. 11V was chosen as the voltage of the device derates with current, and at 20A at the expected voltage trim, the output voltage would be 11V. The LV system is still operational at this voltage.

Ploss = V \* I \* (1-n) = 11V \* 20A \* 0.117 = 25.75W

The thermal dissipation from the case was insufficient to deal with that amount of power loss. A max temperature of 110 C was chosen along with an ambient temperature of 25 C, as the max operation temperature of the device is 125C. This allows for a thermal resistance of 85C / 25.75W or 3.3 C per watt. The device's thermal resistance to the top of the case is 1.75 C/W, and to the bottom is 6.75C/W. The thermal is taken to be 10C/W per Texas Instruments. A heatsink and thermal contact would have to be chosen in order to keep the power dissipation within specification. The following thermal circuit describes this:



Solving this finds that the thermal resistance of the adhesive plus the heatsink can be a maximum of 2.4 C/W. The best thermal adhesive that could be found had a thermal resistance of 0.3 C/W. This meant the heatsink required a thermal resistance of 2.1 C/W or less. The 628-65AB was chosen along with a fan as no convection cooled heatsink could meet the specification.

Current	Input Voltage	MOSFET Temp.	Current Sense Resistor Temp.	Inductor Temp.	Input to Output Voltage	MOSFET V <sub>drop</sub>	PCB Temp.	Ambient
10A	13.85V	35C	24C	24C	.1V	0.8V	20C	19.3C
15A	13.85V	44C	27C	26C	.115V	0.8V	29C	19.3C
20A	13.85V	57C	31C	28C	.135V	0.8V	33C	19.3C
25A	13.85V	69C	35C	40C	.175V	1V	39C	19.3C
30A	13.85V	65C	54C	43C	.16V	1.05V	64C	19.3C

The components in the system were validated thermally with the following data:

All components are well within their thermal specification as is the PCB

#### Hotswap Controller



The TPS2481 is a hotswap controller to control and monitor the DCDC converter's output. It features multiple monitoring and safety features. The controller is communicated with via I2C by the auxiliary microcontroller on the Battery Central board. In order to convert the voltage drop across the shunt resistor to a current, a register on the controller has to be written to with the resistance of the shunt resistor. The current, voltage and power out of the DCDC converter can then all be sent over I2C and monitored by the BMS.

#### Gate Control:

The hotswap controller controls the gate of an N-channel MOSFET. The MOSFET was selected for a high current rating, SMD package, not requiring extra thermal components and a low Rds(on). The MOSFET is driven by an internal charge pump to the TPS2481. This allows for a greater voltage at the gate than the system voltage, which is required switching on the N-channel MOSFET. There is then a dvdt capacitor on the gate that gets charged by the charge pump. This allows for a voltage ramp while connecting the DCDC to the LV system. This prevents inrush current and allows for a smoother connection to the system. A ramp of 7 ms was selected using the load capacitance of the system as a guide. View the attached code for in depth calculations. Additionally, there is a blanking timer capacitor to prevent over sensitivity of the circuit break features of the hotswap.

#### TSAL



The TSAL driver circuitry was redesigned this year to take advantage of the new voltage sensing transducer. This allows for all TSAL driving circuitry to be in the GLV side of the car, not the HV side. The circuitry uses a comparator that is connected to the voltage transducer. The signal then either drives the gate of the green LEDs MOSFET driver or enables a 555 timer to blink the red LEDs. The circuitry was integrated onto the HV daughter board and will drive any generic LEDs, instead of having it's own custom TSAL PCB. This allows for less risk and easier swapability of required rules circuitry.

#### **PCB** Design

#### Stackup

Due to the high current and small size of the board, it used a two layer outer copper weight. To keep costs down with a two ounce board, a two layer board was designed. The board was ordered from JLCPCB and assembled in house.

#### Layout

The board has both a LV and HV section. HV clearances were maintained with the nearest LV to HV clearance being 6mm and the nearest HV+ to HV- being 4mm. The plugs for HV+ and HV- prevent accidental swapping of connectors. The LV section of the board has three main components. The first is the high current path out of the DCDC converter as it flows through the inductor, the sense resistor and the MOSFET to the busbar. This high current path is routed on both the top and bottom of the board and stitched with vias. The other sections are the hotswap controller and the TSAL which are routed to allow for a continuous ground plane underneath them.

#### Mechanical

The daughter board is mounted to the central BMB via metal standoffs that thread into solderable PCB inserts. This allows for easy mounting and removal of hardware. This also allows for additional ground paths, which is critical given the high power nature of the board. These same standoffs are on the Power Distribution Unit and this puts the boards at the exact same height. The PDU is mounted to the output of the DCDC converter via a positive and negative busbar which allows for a rigid connection for power flow between the two. A nylon standoff is used in the HV section to prevent any HV to LV clearance issues. The bottom of the HV section is covered with Nomex for increased isolation.

#### Calculations

```
#Hotswap Controller Calcs
#Rsense- circuit breaks when there is a 50 mV difference across Rsense
Ilimmin = 32 #chosen to be higher than the max DCDC output
Vdiff = 50e-3
Rsmax = Vdiff / Ilimmin
print("rsmax: " + str(Rsmax*1000) + "mOhms")
#1.3 mOhms is what is in stock
Rs = 1.3e-3
Ilim = Vdiff / 1.3e-3
print("Ilim: " + str(Ilim))
#Power Limit resistors
Tj = 175
Ta = 30
```

```
RthCase = 83.3
RthJ = .43
Rds = .85e-3
PLim = (.7 * (Tj - RthCase * Ilim * Ilim * Rds - Ta)) / RthJ
print("Power Limit: " + str(PLim)) #Going to put a heat sink on this and
do a better thermal layout to allow for greater dissipation
#Plim max
Vref = 4
Vout = 13.5
PlimMax = Vref / (2*Rs)
print("Plim Max: " + str(PlimMax) + "W")
Pout = Vout * Ilim
print("Nominal Power Out: " + str(Pout) + "W")
Vprog = 2 * Pout * Rs
print("Vprog: " + str(Vprog))
Rprog2 = 20e3 #Choosing a value of 20k
Rprog1 = Rprog2 * ((Vref/Vprog) - 1)
print("Rprog1: " + str(Rprog1*.001) + "kOhms")
#Timer Capacitor
#Load capacitance of 2000uF, as that is the bulk capacitance on PDU
C = 20000e-6
Ton = (C * Pout) / (2 * Ilim * Ilim) + (C*Vout*Vout)/(2*Pout)
print("ton: " + str(Ton))
IsMax = 34e-6 #controller current source Value
Vtime = 4.1 #controller timer voltage threshold value
tolerance = 1.3
Ctime = Ton * tolerance * (IsMax/Vtime)
print("Timer Capacitor: " + str(Ctime*pow(10,6)) + "uF")
#selecting timer capacitor of .01uF
#Gate resistor
#35uA max in datasheet
Igmax = 35e-6
VgMax = 16
Rgatemin = VgMax / Igmax
print("Gate Resistor minimum: " + str(Rgatemin * pow(10,-3)) + " kOhm")
#Inrush Capacitor
Igate = 22e-6
Inom = 30
```



Cgate = 12900e-12 ton = (C \* Vout) / Inom Igate = 22e-6 print(ton) Cdvdt = (ton \* Igate)/Vout - Cgate print("Inrush Cap: " + str(Cdvdt \* pow(10,9)) + "nF")

#### Code Output:

rsmax: 1.5625mOhms Ilim: 38.46153846153847 Power Limit: 65.53770469244526 Plim Max: 1538.4615384615386W Nominal Power Out: 519.2307692307693W Vprog: 1.35 Rprog1: 39.25925925925926kOhms ton: 0.0070199999999999 Timer Capacitor: 0.07567902439024389uF Gate Resistor minimum: 457.14285714285717 kOhm 0.0090000000000001 Inrush Cap: 1.7666666666666693nF











#### Hotswap Controller

WISCONSIN




# **ISOSPI** Board

Engineer: Christian Schuster, Electrical Technical Director Revision: 2023.1 Board Dimensions: 45mm x 15mm Current Draw: 7mA

### Overview

The purpose of the ISOSPI daughter board is to allow for future proofing, debugging and modularity. ISOSPI is a proprietary isolated communication protocol by Analog Devices. Therefore, if the team ever goes to a different Battery Management IC, a new protocol would be required. The ISOSPI daughter board hosts all peripheral circuitry for that ISOSPI protocol. If a new IC was chosen, such as a TI IC, this small board could be swapped instead of a full overhaul of the Battery Main Board. Additionally, this small board allows portability of this communication protocol so that the full Battery Main Board is not needed to debug and work with the Battery Management system

#### Features:

• SPI to ISOSPI conversion circuitry





ISOSPI Daughter Board Schematic

### Circuit

The SPI to ISOSPI conversion is done using the LTC6820 IC. The IC is configured as recommended by the datasheet. For more reliable communication, the maximum drive strength is used. Split termination and a common mode choke are used to reject common noise. A center tapped transformer was also chosen for this purpose. The chosen connectors were a T2M connector from Samtec and standard headers, both of which have proven reliable to the team and are widely used.



# **Smart Power Distribution Unit**

Engineer: Christian Schuster, Electrical Technical Director Revision: 2023.1 Dimensions: 100mm x 110mm Steady State Current Draw: 100mA

### Overview

The Smart Power Distribution Unit is the low voltage fusing, circuit protection and current monitoring PCB. It is designed to be compatible with both the team's EV and IC race cars. It was designed as a daughterboard to be easily swapped between systems. It was redesigned away from high side switches in previous designs to eFuses. This allows for dvdt startup to prevent inrush current, which has been a recurring issue for the team. To make this switch, it required using two TPS25940 eFuses in parallel in order to get the current limit of each fuse output to be high enough.

#### Features:

- 8 fuse outputs using two parallel TPS25940 eFuses per output
- Multistage vehicle power on
- 8 A continuous current out of each fuse output
- Overvoltage lockout
- Reverse Voltage Protection
- Overcurrent protection
- Short circuit protection
- Current monitoring
- Voltage ramp using dvdt to prevent inrush current
- Auto resettable fuses
- Power good and fault monitoring
- Integrated TM4C microcontroller
- CAN Transceiver and related circuitry
- Multiplexer to allow for high number of analog inputs
- 5V and 3V3 Power regulation for sensors on the car and LV system





### Parallel TPS25940 Automotive eFuses



The eFuses wide variety of features, namely inrush current protection, were ideal for the design and an improvement over high side switches. The main issue with eFuses was finding an automotive rated eFuse with a high enough current rating. To work around this issue, parallel eFuses were implemented. The drawback to this is design complexity, but beyond that, due to potentially different RDSon values inside the eFuse, current may not be shared equally. Different RDSon values mean one fuse could have 3A of current while the other has 5A. This means that for both fuses at a 5A current limit, the paralleled fuses are only rated for 8 A together. The parallel operation of the eFuses was derived using this application note from Texas Instruments (<u>https://www.ti.com/lit/an/slva836/slva836.pdf</u>). The fuses features are included above.



### Functional Block Diagram



### **Design Features and Design Decisions**

Specific equations for values can be found in the appendix of this project and are done in Python. They can also be found in the datasheet for this eFuse.

- Devsleep is disabled, no reason to sleep eFuses in our operation
- Added a button to reset fuse as the auto retry fuses were out of stock and the reset button would be useful for the latching version. Also useful for testing
- Only broke out one PGOOD line as PGOOD is dependent on the same criteria for each fuse
- Tied faults together so that the fault monitoring is per output, not per fuse. (8 faults instead of 16)
- Device enable, overvoltage lockout and undervoltage lockout resistor divider are shared as the mechanism for setting those values is simple voltage feedback
- Voltage Ramp
  - dvdt capacitor is shared as the voltage ramp of the fuse outputs needs to be in sync, otherwise load sharing during start will be off
  - Current sources sum during dvdt capacitor charge, so capacitor needs to be double the value calculated for specific ramp time in datasheet
  - Ramp time was decided based on max power dissipation during the ramping period to prevent the fuse from overheating

- Ramp can be adjusted for high capacitive loads. Note that a high resistive load and high capacitive load can cause startup failure as a long ramp time with a high steady state current can cause temperature lockout. Test values for this ramp value are shown.
- Each fuse has its own separate IMON output that outputs (52uA \* drain\_current). This coupled with a resistor creates a readable voltage.
- Input capacitors are denoted as optional for noise suppression in the datasheet. Given EMI issues in the past and noise injected from either the rectifier on the IC vehicle or the DCDC converter on the EV, two input capacitors were chosen of varying values to allow better filtering and decoupling.
- Load sharing and overall fuse output current rating is found using the max and min RDSon values of the eFuses and assuming the worst case. This means that at a current of 8A, worst case one eFuse could have 3A and the other 5A. This means the output current rating of the system is 8A, not 10A per fuse.
- Fast trip threshold (short circuit protection) is 1.5 \* ILIM + 0.375A. That means for two eFuses set to 5A, fast trip worst case is 15.375A.

### Load Testing Data

Testing was done using an electronic load set to a constant current draw and a parallel capacitor. Higher performance was also seen when using a resistive load versus constant current

Resistive Load	Capacitive load	Status
1A	4400uF	Pass
2A	4400uF	Pass
3A	4400uF	Pass
4A	2200uF	Pass
5A	2200uF	Pass
6A	2200uF	Pass
7A	1000uF	Pass
8A	1000uF	Pass
9A	0(Is likely higher, did not have through hole caps to test, also already out of the "worst case" rated value)	Pass
10A	0	Fail

### **Design Errors and Rework**

The implementation of the eFuses was done according to the TI application note for parallel eFuses. This document was written using the TPS259241 eFuse, not the TPS25940. The



**TPS25941 ILIM** 

overall concepts matched, with the exception of how the current limiting mechanism works between the two eFuses. The functional blocks of each of the fuses are shown below.





TPS25940 ILIM (Device that is on the car)

The TPS25941 uses a current source on the input of the non inverting terminal of its op amp. For this implementation, a single resistor is tied to the ILIM pin of all parallel eFuses. The current's sum together and create a voltage across that resistor to create feedback. A single resistor was used initially in the design, tied to both ILIM pins. This did not work. The TPS25940 (device on the car) ILIM mechanism is a voltage controlled current source. By tying both pins to a single resistor, the two voltage controlled sources interfere with one another as one would fulfill the feedback for the other. This meant that only one of the fuses would have the desired current limit and the other fuse had the current limit of if there were no ILIM resistor. This meant one fuse had a limit of 5A and the other had a limit of 0.5A.

This issue was solved by cutting the trace between the two eFuses and soldering a resistor between the ILIM pin on one of the eFuses and ground and using the already existing resistor for the other.

### IC and EV Differences: UVLO

The team has had issues in the past with over discharging low voltage batteries and this has led to an accumulation of poor performing batteries. For this reason, undervoltage lockout was a key feature for this design. UVLO's importance is especially true for the IC vehicle as the IC vehicle has a starter motor that draws significant current when cranking the engine. In the past driver's would try to start off dead batteries and over discharge them.

The cranking scenario also requires slightly different operations on the IC versus EV. Cranking causes significant voltage droop across the system. On the combustion vehicle a Under Voltage Lockout of 8.5V was set instead of the 10.8V on EV. This value was chosen using a 20% charged LV battery and cranking to measure a baseline value. A higher value could have been chosen for greater battery longevity, but in a racing scenario all that matters is starting the car. Therefore, a value had to be chosen that would only cut power when the car would be unable to start.



## High Level Fusing Schematic



Low pass filters were added to all of the IMON outputs. Additionally, to deal with the high number of analog outputs, a multiplexer was added in order to mux between outputs. A dual 4:1 mux was selected so that both eFuses for a single output could be sampled at the same time. The low pass filter on the muxes was also adjusted to allow for settling time in the signal during muxing. This was verified using an oscilloscope to ensure that the sampled values matched the peak value post mux.





### TM4C123GH6PM Microcontroller

The TM4C microcontroller is used to monitor the fault pins of the eFuses as well as monitor the output currents. It transmits these values over the vehicle's CAN bus. It also controls the multiplexer. The firmware for the TM4C is written in house by our firmware team using the Tivaware embedded C libraries from Texas Instruments. The overall schematic design was done following datasheet recommendations. UART was broken out to allow for faster communication during debugging for transient events that the vehicles CAN bus is not fast enough to trace. A ferrite bead was added to the analog power input to filter noise and decouple the analog power from overall MCU power.





### CAN, Power Regulation and Connectors

### CAN

The team uses the TCAN family of CAN transceivers with integrated level shifters. The CAN circuit is used across both cars and features LEDs to show when CAN communication is occurring, a common mode choke for noise rejection and ESD diodes. This circuit is common on all PCBs that use CAN on the car.

### Power Regulation and Protection

Power regulators for APPS, some sensors and the microcontroller are on the Power Distribution Unit. Linear regulators were selected as APPS and TM4C are both analog devices so a clean power supply is critical for signal integrity. Diodes are added to the inputs of the linear regulators so that the 5V circuitry can be powered without reverse current flowing back through the regulators. This allows for better freedom in debugging. The Power Distribution Unit also features reverse and overvoltage protection for the vehicle. The P Channel MOSFET is specced to have the lowest RDSon possible to limit losses on the input. A 2.2 mF capacitor is used to decouple power from the LV battery.

### Vehicle Controller Main Power Relay

The car starts in two stages. The first stage is always hot where critical systems are given power upon boot, including the vehicle controller. Once the vehicle controller receives a signal from the keyswitch, if it deems that a full boot is safe, power out of the rest of the car is allowed to flow. This is done using the Main Power Relay. A switch was added this year for quality of life as there are many times where the vehicle controller may not be in the car for bench tests and the multistage startup will want to be bypassed.

### Connector, ground and other mechanical components

Two high current blade connectors were selected. The high current was needed for the fused outputs, and it was decided to use the same connector for signals as well. This meant less unique parts and a more reliable stackup. The connectors are also quite sturdy and help add to the structure of the board. The board also features soldered threaded inserts in each of the four corners and standoffs to allow rigid mounting to whatever carrier board it is on. These soldered inserts are grounded and also intended as a way to ensure proper low impedance grounding given the high current in the board.

### PCB Layout and Stackup

The layout's main constraint was high current traces. For space constraints and due to eFuses being on both sides of the board power traces were laid on both sides with stitching vias between them. For heat dissipation the high current traces were on the outside of the board Digital signals were given as much space from analog signals as possible. Filters were placed near where measurement happens as were pull up resistors. For proper grounding, one full layer was devoted to a continuous ground plane with no other nets on that layer. Each decoupling capacitor was given its own via to that ground plane for a low impedance path to ground for the greatest decoupling and noise rejection.

### Stackup

A four layer stackup was used due to desiring a continuous ground plane and an inability to route signals with power polygons on top and bottom. A 2 oz outer copper weight was used to make high current trace size manageable.

	Name	Material	Туре	Weight	Thickness		
			Overlay				
		Solder Resist	Solder Mask		0.01016mm		
1		E	Signal				
	Dielectric 2		Prepreg		0.07112mm	4.1	
2		CF-004	Signal				
			Dielectric			4.8	
3	Signals	CF-004	Signal	1oz	0.035mm		
			Prepreg			4.1	
4		•	Signal				
	Bottom Solder	Solder Resist	Solder Mask		0.01016mm		
			Overlay				

Manufactured by Multicircuits



### Calculations

```
#efuse calculations
N = 2
# Max Current through one fuse if one has min Ron and one has max Ron
# Should try to limit load current to 8A then to prevent overshooting the current limit on
one
Rdsmax = 85e-3
Rdsmin = 55e-3
Iload = 8 #Adjust this value to see what the max load current through one fuse could be
worst case
Imaxonlfuse = Iload * Rdsmax / (Rdsmax + Rdsmin)
print("Max load Current: " + str(Iload) + " A")
print("Imax single fuse: " + str(Imaxon1fuse) + " A\n\n")
# Current Monitor resistor
Vsensemax = 3.3
Ilim = 5
factor = 52e-6
Rimon = Vsensemax / (Ilim * factor)
print("Rmon: " + str(Rimon * pow(10,-3)) + " kOhms")
print("Rmon real world: 12 kOhms")
Rimon = 12000
Vsensemax = Rimon * Ilim * factor
print("Vsense @ real world: " + str(Vsensemax) + " V\n\n")
#Rilim equivalent resistor
Ilim = 5
factor = 89e3
Rilimeq = factor / (Ilim)
#Choosing value of 17k, ensure with tolerance that don't exceed max current limit
RlimReal = 17000
Ilim = N*factor/(RlimReal - RlimReal * .01)
print("Rilimeq: " + str(Rilimeq * pow(10,-3)) + " kOhm")
print("Current limit at max tolerance: " + str(Ilim))
print("Real world Rilimeq: " + str(RlimReal * pow(10,-3)) + " kOhm\n\n")
#Equivalent dvdt capacitor #based on very conservative estimations of a load of 3 ohms on
each fuse,
# And a capacitance of
Idv = 1e-6
Gaindv = 11.9 #gain of the dvdt capacitor
Tdv = 5e-3 #Ramp time choosing 5mS
Cint = 70e-12 #pin capacitance #this is an estimaion
Vin = 13.5
Cdvdt = (N * Idv * Gaindv * Tdv) / Vin - (N * Cint)
print("Cdvdt: " + str(Cdvdt * pow(10,9)) + " nF")
print("Real world Cdvdt: 10nF \nGives a Tdv of around 5.7ms\n\n")
#Find max ramp time before thermal shutdown, pretty much worst case analysis
f = 1/6
Vin = 13.5
tdvdt = 50e-3
Cout = 5000e-6
R1 = 3 #deciding on an arbitrary 3 ohm load
Pdload = f * (Vin*Vin) / Rl
#Assume that we are going to current limit at inrush
Irush = 5
Pdinrush = .5 * Vin * Irush
Pstart = Pdinrush + Pdload
print(Pstart)
# Undervoltage and Overvoltage Lockout
# Datasheets example calculation actually is the values we want anyways, so just using those
R1 = 475e3
```

```
R2 = 16.7e3
R3 = 31.2e3
print("Values for OV and UV resistor divider, shared")
print("R1: " + str(R1 * pow(10,-3)) + " kOhms")
print("R2: " + str(R2 * pow(10,-3)) + " kOhms")
print("R3: " + str(R3 * pow(10, -3)) + " kOhms n n")
#these values give a OV of 16.5 and UV of 10.8V
#Find ramp time before thermal shutdown, pretty much worst case analysis
\dot{\tau} = 0
Rl = 6 #deciding on an arbitrary 3 ohm load
Pdload = f * (Vin*Vin) / Rl
print("Pdload: " + str(Pdload))
for i in range(10,200,10):
    print("\nIteration " + str(j))
    j += 1
    f = 1/6
    Vin = 13.5
    tdvdt = i * pow(10, -3) #50e-3
    Cout = 1000e-6
    Ilim = 5
    tdvlim = Cout * Vin / Ilim
    Iinrush = Cout * Vin/tdvdt
    print("Inrush Current: " + str(Iinrush))
    Pdinrush = .5 * Vin * Iinrush
    print("Inrush power: " + str(Pdinrush))
    Pstart = Pdinrush + Pdload
    print("Start power: " + str(Pstart))
    print(str(tdvdt) + "ms rise time")
# Conclusion from thermal calculations is that we cannot start up the devices under both a
high load capacitance
# and a high resistive load, it appears that a a rise time of 5 to 6 ms strikes the best
balance between high capacitive
# and high resistive loads. For both the highest resistive and highest capacitive loads,
they may require tuning
# note there is a graph in the datasheet that needs to be referenced for these calculations
to make sense based on time before
# Thermal shutdown at various power dissipations
```

#### Code Output:

Max load Current: 8 A Imax single fuse: 4.857142857142857 A Rmon: 12.692307692307693 kOhms Rmon real world: 12 kOhms Vsense @ real world: 3.1199999999999997 V Rilimeq: 17.8 kOhm Current limit at max tolerance: 10.57635175282234 Real world Rilimeg: 17.0 kOhm Cdvdt: 8.674814814814814 nF Real world Cdvdt: 10nF Gives a Tdv of around 5.7ms 43.875 Values for OV and UV resistor divider, shared R1: 475.0 kOhms R2: 16.7 kOhms R3: 31.2 kOhms

Revision 223.1 FERMINE REVISIONS
-------------------------------------













# Brake System Plausibility Device

Engineer: Pratham Sarvaiya, LV Electronics Lead Revision: 2023.1 Dimensions: 32mm x 27mm Steady State Current Draw: 10mA

### Overview

The Brake System Plausibility Device is a non programmable circuit used to monitor accumulator current and brake pressure. This board directly interfaces with a current sensor placed on HV and a brake fluid pressure sensor. It is located on top of the Battery Central Board and interfaces with a relay on the shutdown circuit to open the AIRs.



### Features:

- 4 channel Window comparators to ensure sensor signal is within range
- Pull down resistors to fast fault shutdown circuit if loss of signal from either sensor
- RC delay to check for simultaneous hard braking and >10% throttle
- MOSFET driver to create a robust fault signal that controls the shutdown circuit

## **BSPD** Circuitry

This board controls the Shutdown Circuit by comparing the signals from the current sensor and the brake fluid pressure sensor. When the board loses signal from either sensor, or it detects hard braking at the same time as more than 10% throttle, it opens the shutdown circuit.



#### Window comparators

4 individual comparators are used to monitor signals directly from the 2 sensors(2 comparators for each sensor).



U5B and U5C check if the signals are lower than the 0.45V short circuit reference. If the signal from either sensor is lost, pulldown resistors default the sensor signal to ground. The comparators can then immediately ground their open drain outputs which are pulled up to 5V through a single pull up resistor. This output directly controls the Shutdown Circuit. U5E and U5D monitor the upper threshold of the sensors by comparing them against a voltage set by 2 on board potentiometers. If either sensor goes higher than the set range, those comparators ground their own separate outputs that go to a NOR logic gate.

### NOR logic

The NOR logic gate shown on the left accepts signals from U53 and U5D comparators. Only if both sensors go higher than the set threshold at the same time does the NOR gate output high.



If the output of the NOR gate is high, it charges the RC circuit shown on the right. The RC has a time constant of 0.47 seconds using a 4.7k resistor and 100uF capacitor. The delay reference is set to 3.12V with a resistor divider, and once the capacitor reaches this value, the U8 comparator changes its output from high to low and faults the BSPD output signal. The propagation delay of various components and RC component tolerances was taken into account to ensure the total delay is around 0.5 seconds.

### Gate Driver

The comparator's signals from U8, U5E and U5D are open drain and are all connected to 5V through one Pull up resistor R16, effectively creating a logical OR operation. The gate driver on the right monitors this common signal. A logic high on the input causes a logic high on the output, and a logic low input(ground) causes a logic low(ground) output.

Therefore, if either the low threshold checking comparators faults (U5E or U5D) or the delay reference comparator

faults(U8), the gate driver opens the shutdown circuit by grounding the nBPD fault signal. This signal then goes to the Shutdown Circuit located on the Carrier board.

### Layout Notes

With all surface mount components, this 2-layer BSPD has a small form factor(32mm x 17mm) so it can be easily integrated with other systems. As shown on the right, it uses 2 standard 2.54mm pitch headers to connect to the Battery Central Board from which it receives power and signals from the sensors, and sends the fault signal to control the Shutdown Circuit.

The small size and easily replaceable package(SSOP-14) of the 4 channel comparator, located at the center of the board, provides ease of routing traces to and from the headers.

2 vertical potentiometers set the upper comparison threshold for each sensor. The potentiometer's small size and easily adjustable knob allows for quick calibrating of the board with the sensors.

Decoupling capacitors and pull down

resistors are placed physically close to the comparators to ensure the comparators measure an accurate signal.

### Summary

Overall, this compact design allows us to accurately monitor the current sensor and open the shutdown circuit in the case that any fault occurs from the brake and/or current sensors.











# **Battery Sensor Board**

Engineer: Alex Grotelueschen, HV Electronics Lead Revision: 2023.1 Dimensions: 40mm x 105mm

### Overview

The Battery Sensor Boards feature a powerful battery monitoring integrated circuit from Analog Devices, the LTC6811. The LTC6811 communicates with our Battery Main Board using the ISOSPI protocol, a differential pair galvanically isolated signal. Seven BSBs are daisy chained together to monitor the entire battery pack. The ISOSPI lines are galvanically isolated between each BSB, and again on the Battery Main Board.

#### Features:

- LTC6811 Battery Monitoring Integrated Circuit
- Two 8:1 Multiplexers that send cell temperatures to the LTC6811
- Cell Balancing capability with the LTC6811
- LTC6811 has I2C outputs that are used to control temperature multiplexing and BTB multiplexing
- Galvanically isolated ISOSPI communication
- Almost entirely in the Tractive System
- Max voltage of 47V

Battery Sensor Board PCB Layout

## LTC6811

The LTC6811 is a 12-Cell Battery Monitoring IC from Analog Devices. This powerful IC has 16-bit ADCs for voltage monitoring. It also has 5 GPIO outputs that can be configured for SPI, I2C, Digital, or Analog inputs. There are two LDOs (5.7V and 3V) that can be used to power external circuits. There are 12 internal FETs that can be switched on to activate cell balancing.

### Operation

When no messages are received, the LTC6811 goes into idle mode to save power. Once LV is turned on in the car, the BMS sends a message to the LTC6811 telling it to start a voltage conversion. Next, the BMS sends a message requesting the read voltages and the LTC6811 sends the voltages to the BMS. A similar process is done for the temperatures. This process is repeated while the BMS is active.



### Functional Block Diagram



#### **Device Utilization**

Given our battery architecture of 4p11s segments, each BSB only monitors 11 cells at a time. Each cell tap input has a RC Filter circuit with 10nF and 40 ohms on board. The multiplexers on the Battery Trace Boards have 60 ohms on internal resistance, adding up to 100 ohms of series resistance on each cell input. Zener diodes were placed between series cells to help with ESD issues from hot plugging that could cause chip failures in past revisions. The balancing pins are connected to the cell tap before the hardware filter to help reduce the impedance during balancing. 3 of 5 GPIO outputs are utilized, two for I2C for external multiplexers, and one for reading an analog value from the temperature sensors.





#### 5V Voltage Regulator

The LTC6811 has an on chip LDO that can supply 5.7V (DRIVE), but can only source 1mA. In addition to being powered off of the battery stack, 5V needs to be fed back into the device which is used as the primary power supply for the device. A 5V Regulator is created with an NPN BJT, using the intrinsic 0.7V drop of the BJT. The Drive pin is low when the device is in idle mode, and is high when the device is active. A status LED is placed on the output of the BJT to monitor the status of the device.



#### **ISOSPI** Communication

ISOSPI is a communication protocol unique to certain Analog Devices ICs. It is a differential pair signal that uses current pulses to encode data. The LTC6811-1 variant has the ability to organize multiple LTC6811s in a daisy chain configuration. This configuration works well for our application with the ability to have each segment's BSB daisy chained to the next. Each end of the ISOSPI lines are galvanically isolated with center tap transformers to isolate the signal from high voltage before leaving the accumulator. Common mode chokes are also on the lines to reduce transients, similar to CAN. The ISOSPI Daughter board hosts the LTC6811 generates the pulses for the second LTC6811, and so on.





### **Temperature Monitoring**

Two 8:1 I2C controlled multiplexers are used to monitor 11 thermistors on each segment. VREF2 is 3V when a reading is in progress, but is grounded when the IC is doing other tasks to minimize power consumption. A 10k Ohm NTC thermistor creates a voltage divider with the 10K resistors on the board, and the voltage drop over the thermistor is measured to calculate the temperature. A voltage follower op-amp is used to reduce the impedance of the measurement. The same I2C bus that controls these two multiplexers is sent to the Battery Trace Board to mux through the four parallel strings of voltages.



#### Noise Issues

One unforeseen issue that was discovered during testing was the amount of noise the voltage and temperature wires would pick up in the battery. During testing early on, we noticed repeatedly that various temperatures were shooting up to unobtainable temperatures for 1 sample. We tested various solutions such as increasing the low pass filter on the op amp output and increasing the sampling frequency that didn't have much success until we tried shielding. We first tested by shielding



one segment's voltage and temperature sense wires coming from below the penthouse. The shielding on that segment was grounded to that segment's ground. The improvement was immediately clear, so the shielding was added to all segments to solve our issues.



### Layout Notes

A couple considerations had to be made while laying out this board.

- Due to the sensitivity of ISOSPI, the ground plane cannot be routed underneath any ISOSPI circuitry.
- Voltages are routed in sequential order as much as possible on the board to reduce the voltage difference between adjacent traces.
- The size of the board was kept as close as possible to the last revision to act as a drop in replacement.
- The additional Zener diodes and increased transformer size increased the complexity of routing.
- Large test points were added for ease of probing with high voltage rated probes.

### Layer Stackup

This board has a 4 Layer, 1oz stackup. A continuous ground plane exists on the third layer, and the 2nd layer was used for power polygons for the different supply voltages. Signals are routed primarily on layers 1 and 4, with an additional ground plane on layer 4.

These boards were purchased from JLC PCB, and all passive components were PCBAed from JLC PCB as well to reduce the amount of assembly time needed when they arrived.

O J3 BSB ID:	0
000000000000000000000000000000000000000	D D
	R3X CEL
GPI02	

#	Name	Material	Туре	Weight	Thickness	Dk	Df
	Top Overlay		Overlay				
	Top Solder	Solder Resist 🛛 🔤	Solder Mask		0.4mil	3.5	
1	Top Layer		Signal	1oz	1.4mil		
	Dielectric 2	PP-006	Prepreg		2.8mil	4.1	0.02
2	Layer 1	CF-004	Signal	1oz	1.378mil		
	Dielectric 1	FR-4	Dielectric		12.6mil	4.8	
3	Layer 2	CF-004	Signal	1oz	1.378mil		
	Dielectric 3	PP-006	Prepreg		2.8mil	4.1	0.02
4	Bottom Layer	-	Signal	1oz	1.4mil		
	Bottom Solder	Solder Resist 🛛 🔤	Solder Mask		0.4mil	3.5	
	Bottom Overlay		Overlay				

	TANT CONTRINT	A LOUINOLIN	
	Title <b>Top Level</b>	Designer Alex Grotelueschen 223.1	ate: 6/13/2023 ile: BSB TopLevel.SchDoc
Laga Ark Laga A			









# **Battery Trace Board**

Engineers: Quinn Sabin, 2022 Electrical Technical Director and Kai Linsenmeyer, 2022 HV Electronics Lead Revision: 2022.2 Dimensions: 50mm x 537mm

### Overview

Designed in 2022, the Battery Trace Board is our solution to measuring all 308 cells. Each board is equipped with 5, dual channel 4:1 multiplexers that are used to switch through each of the parallel strings of cells. The outputs of the multiplexers are passed up to a Battery Sensor Board that lives above the isolation sheet via a cable assembly. One thermistor value per set of four cells is also passed to the BSB.

#### Features:

- 5 dual channel 4:1 multiplexers
- 1 I2C GPIO Expander
- Cell taps to every cell in the segment
- Thermistor Inputs
- Entirely in the Tractive System
- Max voltage of 47V



Multiplexing Circuitry, Repeated 5 Times

### Multiplexing

Each dual channel multiplexer muxes through two sets of cells at a time. All 5 muxes are connected to the same two GPIO outputs from a I2C GPIO expander so that all multiplexers switch to the same string of cells at the same time. The BMS continuously switches through each string of cells so that all cells will be monitored regularly.

### Layout Notes

In collaboration with our mechanical battery team, the board outline was created so that the board would fit into place easily on the top of our battery segments. Mounting holes were added with tolerance to keep the board from moving during driving.





# Battery Management System (BMS) Firmware

Principal Engineers: Eric Bocheng Li, Gabriel Gozum, Michael Siem Contributing Engineers: Alex Quilliam, Noah DeGroot, Jakob Landgraf, Jerry Xu, Sean Staggs, Christian Schuster

### Overview

The Battery Management System (BMS) is the electronic system aimed at managing and maintaining the car's battery pack. Its functionality includes measuring the battery pack's temperatures and voltages, battery charging and balancing, and CAN communications. The BMS runs on a TM4C Launchpad that's plugged into the Battery Main Board (BMB), connecting to LTC6811 IC for measurements.

#### Features:

- Prevents the car from running when certain fault conditions hit
- Temperature and voltage monitoring with ADC value conversions
- Dynamic power limits set from battery monitoring and sent to the vehicle controller
- Cells connected in series with multiplexers
- Customized CAN database and message sending
- Student-written Custom MCU Driver Library and Scheduler
- It's a cool BMS and it works!

### Breakdown by Components:

On the page below you will find a high level overview diagram of the BMS. They can be divided into three major parts, which will be explained in detail on the following pages. They work together to ensure the features mentioned above are met and executed correctly from the beginning of signal inputs, to data processing, to outgoing communications.

- 1. Voltage and Temperature Reading (input)
- 2. CAN Communication Bus (output)
- 3. BMS Algorithm (software)





### **Real-Time Battery Readings**

The electric car's accumulator contains a 7x11 grid of Li-Po battery modules with four cells connected in series for each module, so 308 battery cells are monitored by the battery management system. The battery stack monitor is the Analog Devices LTC6811 which is located on the Battery Sensor Board. There are seven Battery Sensor Boards which are each connected to a Battery Trace Board that are attached to a segment of eleven battery modules. Each module has four voltage taps for each cell and a singular temperature sensor. The voltage taps are multiplexed by the Battery Trace Board throughout the voltage measurements. The BMS communicates with the battery sensor board through isoSPI. The BMS is able to take temperature and voltage readings efficiently by interfacing with the battery sensor board and multiplexing through the cell taps.



High-level BMS-Battery monitoring schematic of one of seven battery segments

#### Voltage Readings

Being able to read the voltage of the cells is important for determining the state of charge, performing balancing of batteries, determining current, and fault monitoring of the pack. The BMS measures the voltage of each cell, checks for faults, and performs battery-related calculations based on voltage.

The voltage measurement starts with a command targeted for the battery stack monitors to perform the voltage readings. 3.5 milliseconds is then allotted to the monitor to take the readings. After waiting, the data from the monitor is then transferred to the closed cell voltage readings member of the BMS struct. By using the most recent battery pack current reading, resistance is then calculated. Finally, open cell voltage is calculated immediately after.

After the voltage readings have been updated to the BMS struct, general statistics of the pack are determined by updating the maximum, minimum, and average values of the cell discharge resistance, cell charge resistance, closed cell voltage, and open cell voltage. Additionally, closed and open cell voltage faults are checked for each cell by comparing the recently updated voltage readings with a set limit. If the read voltages are above or below the limits, a fault flag is set. The multiplexer between the four cells is then incremented before the next voltage reading.

#### **Temperature Readings**

Monitoring the temperature of the batteries is important for avoiding dangerous battery defects from causing damage. The BMS is capable of reading the batteries' temperatures and faults when temperatures fall out of range. Even further, the temperature readings are used to



determine limits on charge and discharge current of the battery pack. This allows for dynamic power scaling as cell temperature increases.

The temperature of a battery module is read by the battery stack monitor on one of its GPIO. The input on the GPIO is essentially a multiplexed temperature sensor value of one of the eleven cell modules monitored by the sensor board.

The temperature reading process starts with a call to the battery stack monitor to read the input on the configured GPIO. The read temperature information is then communicated to the BMS and temporarily stored. The BMS then decodes the temperature data and makes the temperature level global for other tasks to use for calculations. Lastly, the BMS increments the temperature multiplexer to the temperature level of the next cell module to be read by the next execution of the temperature measurement task.

### The CAN Bus

#### **Overview:**

The Battery Management System (BMS) relies on the seamless integration with the Controller Area Network (CAN) bus to achieve efficient battery monitoring and control. The CAN bus serves as a vital communication channel, facilitating the



exchange of crucial battery status information between the BMS and other vehicle systems. The CAN bus acts as a reliable and robust medium for real-time data sharing between the BMS and the rest of the car's control systems. It enables the BMS to provide up-to-date information about the battery's status, including voltage and temperature measurements, to the vehicle's overall control system. Through this integration, the BMS ensures that all parts of the car have access to the most accurate and current battery data, enabling optimal decision-making for the vehicle's operation.

#### Task Scheduling and Execution:

The BMS incorporates a line by line, hand-written, round-robin scheduler. This scheduler efficiently manages the execution of various tasks listed in a scheduler table. Tasks related to voltage and temperature measurements, as well as calculations for critical values like cell state of charge and charge and discharge resistance, are meticulously performed by the BMS in a timely manner. The CAN bus facilitates the orderly execution of these tasks, ensuring that the battery status is continuously monitored and evaluated.

#### Enhanced Debugging Capabilities:

The CAN bus plays a pivotal role in providing enhanced debugging capabilities for the BMS. It serves as a conduit for relaying real-time signals that need to be measured, allowing us to effectively diagnose and troubleshoot any potential issues in the battery system. By utilizing the CAN bus for debugging purposes, we can promptly identify and resolve anomalies, thereby

ensuring the optimal performance and reliability of the car. The ability to measure critical signals in real-time through the CAN bus greatly streamline the debugging process, minimizing downtime and enhancing overall system efficiency.

### Conclusion:

The integration of the CAN bus within the BMS is indispensable for achieving efficient battery management. Through the CAN bus, the BMS can seamlessly share real-time battery status information and provide enhanced debugging capabilities. This integration empowers us with accurate and up-to-date battery data, enabling us to make informed decisions regarding the vehicle's operation. Ultimately, the CAN bus integration significantly contributes to the overall performance, reliability, and safety of the car's battery system, ensuring a smooth and enjoyable driving experience.

### Screenshots of CAN Bus at work:

(Charging Cart Configuration)

			-	*	····· •			A MI Duce	aartiaa	*
erator C	Con	Generator T.		Raw Value	Raw Step	Phys Value	Phys Step	Unit	Start Bit	Length
	aff.	None	$\sim$	0	1	0	1	Flag	60	4
	di	None	$\sim$	F	1	15	1	Flag	56	4
	197	None	$\sim$	B8	CD	9.20	10.05	Α	20	12
	11	None	$\sim$	77C	CD	383.2	40.2	V	8	12
	af	None	$\sim$	0	1	0	1	Flag	4	2
	197	None	~	0	1	0	1	Flag	2	2
	diff.	None	$\sim$	1	1	1	1	Flag	0	2
			None None None None None None None	None       None	Image: Second	None         0         1           None         F         1           None         B8         CD           None         77C         CD           None         0         1           None         1         1	Image: None         > 0         1         0           Image: None         > F         1         15           Image: None         > B8         CD         9,20           Image: None         > 77C         CD         383.2           Image: None         > 0         1         0           Image: None         > 0         1         0           Image: None         > 0         1         0           Image: None         > 0         1         0	Image: None         Vol         1         0         1           Image: None         V         F         1         15         1           Image: None         V         F         1         15         1           Image: None         V         B8         CD         9,20         10.05           Image: None         V         77C         CD         383.2         40.2           Image: None         V         0         1         0         1           Image: None         V         0         1         0         1           Image: None         V         1         1         1         1	Image: system         Image: s	Image: None         V         0         1         0         1         Frage         60           Image: None         V         F         1         15         1         Flage         56           Image: None         V         B8         CD         9.20         10.05         A         20           Image: None         V         77C         CD         383.2         40.2         V         8           Image: None         V         0         1         0         1         Flage         4           Image: None         V         0         1         0         1         Flage         2           Image: None         V         0         1         0         1         Flage         2

#### (BMS/Inverter Data Tracing)



(Reporting BMS data through CAN)





### BMS Software Design:

#### Introduction

For the BMS to function properly, a software system has to be running behind the scenes. On the software side, our BMS incorporates many programming designs together to ensure that each component works as we intended. Here is some detailed information about the software system.

#### Scheduler

To run things in a software system, a scheduler is needed. Our BMS uses a student-written bare metal scheduler that utilizes the timer interrupt in our TM4C driver library so that whenever the system ticks once, the scheduler's tick increments by 1 too. The scheduling algorithm it uses is round-robin, giving us the advantage of choosing time intervals that each task runs at, while also giving some tasks always-ran priority (by giving it a 0 tick interval). (Scheduler run function call)

![](_page_71_Figure_7.jpeg)

Because the scheduler determines what methods are called when the system runs, and sometimes we want some methods to be called and sometimes not, we have a scheduler table of available tasks, which can be organized into three groups based on their functionalities:

- 1. Voltage temperature reading (explained above)
- 2. CAN data handling (explained above)


- 3. BMS algorithms (explained here)
- 4. Charging/Balancing (explained here)

### Data Structure

With a large number of cells in our battery pack, a huge amount of data fields need to be processed in the BMS code. To address the problem, there is a dedicated *bms\_t struct* that stores the overall data of all BMS fields. Some most commonly used fields are **pack voltage**, **discharge/charge current limit, balancing information, pack cell voltage, and charging resistances**. To account for every cell, there is also a dedicated *cell\_t struct* that stores information about each individual cell, such as voltage, current, temperature, and resistance. (Individual cell and BMS data struct)

```
typedef struct
```

{

```
uint16_t closed_cell_voltage; //!< [0.1 mV]
uint16_t open_cell_voltage; //!< [0.1 mV]
uint8_t ch_i_limit; //!< [Amps]
uint16_t dch_i_limit; //!< [Amps]</pre>
```

```
uint8_t charge_res; //!< [0.1 mOhm]
uint8_t discharge_res; //!< [0.1 mOhm]
int8_t temperature; //!< [deg C]</pre>
```

```
int32_t capacity; //!< [A*ms]
int32_t energy; //!< [Watt*sec]
float soc; //!< [Fraction]
float soe; //!< [Fraction]</pre>
```

```
bool balancing_active;
} cell_t;
```



pedef struct

# float i\_pack[5]; //!< Pack Current [1 Amps] Discharge Current is Positive float i\_pack\_offset; //!< Drift Offset for Current [1 Amp] float i\_pack\_voltage; //!< [Amps]</pre> uint16\_t discharge\_current\_limit; //!< Discharge Current Limit [1 Amp] uint16\_t charge\_current\_limit; //!< Charge Current Limit [1 Amp]</pre> uint32 t airs ticks: // Statistics stats\_t discharge\_res\_stats; //!< Cell Discharge Resistance Stats stats\_t charge\_res\_stats; //!< Cell Charge Resistance Stats stats\_t open\_cell\_voltage\_stats; //!< Open Cell Voltage Stats</pre> stats\_t closed\_cell\_voltage\_stats; //!< Closed Cell Voltage Stats stats\_t cell\_temperature\_stats; //!< Cell temperature stats</pre> stats\_t cell\_soc\_stats; //!< State of charge stats stats\_t cell\_cop\_acity\_stats; //!< Cell capacity stats stats\_t cell\_ch\_i\_limit\_stats; //!< Cell charge current limit stats stats\_t cell\_dch\_i\_limit\_stats; //!< Cell discharge current limit stats</pre> // CAN Rx Values bool airs\_closed; // received on CAN // EEPROM information //!< Statistics that are tracked in eeprom //!< Counter for number of time writing to eeprom. eeprom\_stats stored\_stats; // Balancing Info uint8\_tbalancing\_cell\_id; //!< Cell ID of the cell that's actively balancing uint8\_t num\_cells\_balancing; //!< Number of cells with DCC switches enabled. uint16\_t balancing\_vltg\_limit; //!< Voltage limit to discharge to for cell balancing</pre>

cell\_t cells[TOTAL\_CELLS]; //!< Cell information</pre> uint16\_t pack\_ccv; //!< Pack closed cell voltage [0.01 Volts] uint16\_t pack\_ocv; //!< Pack open cell voltage [0.01 Volts]</pre> uint16\_t pack\_charge\_res; //!< Pack charge resistance [m0hms] uint16\_t pack\_discharge\_res; //!< PAck discharge resistance [m0hms]</pre>

#### Algorithms

In simple terms, while processing incoming data, first a value is read from a sensor's register (which corresponds to one cell's reading), then depending on which type of data is read, the value is converted into the actual value for the system to use. It does this by using techniques such as look-up tables and interpolation. As readings come and go, the system then calculates statistics such as current state of charge, limits on discharge, charge, voltage, current... on both the pack and individual cell level.

(Calculating current limit from cell temperature readings)





#### **Utilizing Timer Interrupts**

Round robin scheduler has its own limitations. Sometimes one task can take up all the processing power or another task might not get to run at all. But while the car is running, we do want to make sure we're constantly taking new readings of voltage and temperature. To address this problem, we implemented method calls to read voltage and temperature inside our MCU, TM4C Launchpad's time interrupts, so that no matter what is going on scheduler side, the methods to read new temperature and voltage are always called.

(Setting read function calls as interrupts)

IntDefaultHandler,	// ADC Sequence 0
IntDefaultHandler,	// ADC Sequence 1
IntDefaultHandler,	// ADC Sequence 2
IntDefaultHandler,	// ADC Sequence 3
IntDefaultHandler,	// Watchdog timer
<pre>main_read_measure_voltage_all,</pre>	// Timer 0 subtimer A
<pre>main_read_measure_voltage_all,</pre>	// Timer 0 subtimer B
<pre>main_read_temperature_channel,</pre>	// Timer 1 subtimer A
<pre>main_read_temperature_channel,</pre>	// Timer 1 subtimer B
IntDefaultHandler,	// Timer 2 subtimer A
IntDefaultHandler,	// Timer 2 subtimer B
IntDefaultHandler,	// Analog Comparator 0
IntDefaultHandler,	// Analog Comparator 1
IntDefaultHandler,	// Analog Comparator 2

#### (Current measurement method call)

```
float current_get_measurement(void)
{
    float voltage = ltc1864_get_measurement();
    bms.i_pack_voltage = voltage;
    float i_measured = 0;
    if (voltage < I_SENSE_OORL_VOLTAGE)
    {
        fault_set_cCurrentSensorOORL(voltage);
    }
    else if (voltage > I_SENSE_OORH_VOLTAGE)
    {
        fault_set_cCurrentSensorOORH(voltage);
    }
    else
    {
        // Voltage is valid. Convert to I value. Apply drift.
        i_measured = (voltage - 2.5) * 1000.0 / I_SENSE_GAIN;
        i_measured = i_measured - bms_get_i_pack_offset();
        // clear fault if one existed
        fault_clear_cCurrentSensorOORH();
        fault_clear_cCurrentSensorOORH();
    }
    return i_measured;
}
```

(Initialization of timer interrupt)



{

}

# void timer0\_init()

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);
while(!SysCtlPeripheralReady(SYSCTL\_PERIPH\_TIMER0)) {};

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_ONE\_SHOT);

//Enable interrupts
IntEnable(INT\_TIMER0A);

//IntPrioritySet(TIMER0\_BASE, 1);

TimerIntEnable(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);



## Reporting Individual Cell over CAN

(Individual cell signals in CAN)

	BMS_CellVoltage59 [V]
	BMS_CellVoltage60 [V]
	BMS_CellVoltage61 [V]
	BMS_CellVoltage62 [V]
	BMS_CellVoltage63 [V]
	BMS_CellVoltage64 [V]
	BMS_CellVoltage65 [V]
	BMS_CellVoltage66 [V]
	BMS_CellVoltage67 [V]
	BMS_CellVoltage68 [V]
	BMS_CellVoltage69 [V]
	BMS_CellVoltage70 [V]
	BMS_CellVoltage71 [V]
	BMS_CellVoltage72 [V]
	BMS_CellVoltage73 [V]
	BMS_CellVoltage74 [V]
	BMS_CellVoltage75 [V]
Image: Contract of the second seco	BMS_CellVoltage76 [V]

(CAN sending method iterating TOTAL\_CELLS)

```
void can_send_cell_broadcast_1(void)
{
    static uint8_t cell_i = 0;
    uint8_t data[CAN_CELL_BROADCAST_1_LEN];
    data[0] = cell_i;
    data[1] = (uint8_t)bms_get_closed_cell_voltage(cell_i);
    data[2] = (uint8_t)(bms_get_closed_cell_voltage(cell_i) >> 8);
    data[3] = (uint8_t)bms_get_open_cell_voltage(cell_i);
    data[4] = (uint8_t)(bms_get_open_cell_voltage(cell_i) >> 8);
    data[5] = bms_get_cell_ch_i(cell_i);
    data[6] = (uint8_t) bms_get_cell_dch_i(cell_i);
    data[7] = (uint8_t) (bms_get_cell_dch_i(cell_i) >> 8);
    can_send_message(CAN_CELL_BROADCAST_1_OBJ_ID, data);
    cell_i = (cell_i + 1) % TOTAL_CELLS;
}
```

#### Charging

The BMS supports charging the pack through a charging cart, with CAN signals to indicate whether the car is in an active charging session or not. Furthermore, we are able to input charging commands such as current and voltage limit to the charger to prevent dangerous charging activities.



#### (CAN sending method to begin charging)

#### Balancing

The LTC6811 we have chosen allows us to actively balance the battery cells due to their 4 in series connection. To enable balancing, an external CAN signal needs to be input into the system.

(Vector CANoe command to enable cell balancing)

									_		IDE							
4	Manual	~	bms::E	MSrx_C	onfig		2	41	CAN 1		Data	× 8	~ ▶	L.	lame		bms::BMSrx_0	Config
5	Manual	~	bms::0	CHARGEF	Rrx_Cmd	i	1	8FF4AEFx	CAN 1	Exten	ded CAN Data	~ 8	$\sim$	Trigger			Manual	
													i.	1	уре	CAN Data		
	CAN FD Properties												\$					
														E	RS			
														CA	N Properties			\$
														F	TR			
<b>C</b> 1																		
Signal	Signals Raw Data																	
Name		0	Generato	or Contro	ol	Generator Type	1	Raw Value	Ra	w Step	Phys Value		Phys Step		Unit	Start Bit	: Length	
BMSr	_Enable_Cell_Balancing	1	►		1	None	$\sim$	0	1		0		1		Flag	7	1	
BMSrx	_Charger_Voltage_Limi	t	►		1	None	$\sim$	0	CC	D	0.0		655.4		v	24	16	
BMSr	_Charger_Current_Limi	t	•		1	None	$\sim$	0	co	D	0.00		163.85		Amps	40	16	

(Cell balancing method call to enable balancing on hardware)





### **Faulting Matrix**

The BMS needs to know when the battery pack has entered an unsafe state. We are able to create a faulting matrix system that detects bad readings or dangerous data in the BMS and send fault signals through CAN to indicate the car needs to stop, enter a fault condition or to derate power output.

(Fault for when cell temperature is above 60C degrees)



# Auxiliary Microcontroller (Aux MCU)

Engineers: Jakob Landgraf, Shrey Modi, 2023

## Overview

The Aux MCU is a Texas Instruments TM4C123GH6PM Launchpad located on the Battery Central Board which monitors various systems on the electric car. By reading and decoding analog and digital signals from the output of the devices the controller is connected to, the Aux MCU reports its respective system's faults and data to CAN in real time.

## Features:

- Fault monitoring for the following systems:
  - Shutdown Circuit (SDC)
  - Insulation Monitoring Device (IMD)
  - Brake System Plausibility Device (BSPD)
  - Powertrain Control Module (PCM)
  - Accumulator Isolation Relays (AIRs)
- Data monitoring for the following systems:
  - Brake pressure
  - 12V Battery voltage
  - Electrical Isolation status
  - Fault data
- Controls the hotswap controller for the HV to GLV DCDC converter
  - Monitors voltage in
  - Monitors voltage out
  - Monitors current and power out
  - Protects against overcurrent and reverse voltage
  - Enables DCDC five seconds after the positive AIR is closed
- Real time reporting of faults and data to CAN

# Controller Area Network (CAN)

## Transmitting and Receiving

The Aux MCU is equipped with CAN capabilities which enables the controller's data and fault readings to be broadcast to the rest of the car's electronic control units. The CAN baud rate can be configured in the code to match the bus rate, which is adjustable up to 1 MBaud. The fault and data transmission is handled by a function which executes every 50 microseconds. The execution is controlled by a periodic timer which interrupts and runs the CAN transmission at the timer timeout. When the CAN transmission function runs, the analog and digital inputs are read and decoded. If a change in the SDC return fault status has been

Aux MCU seen on bottom left of Battery Central Board



read, the digital values are immediately broadcasted. Otherwise, all digital and analog values as well as the fault states are broadcasted every 0.2 seconds.

The Aux MCU also receives CAN messages. When the controller receives a message indicating incoming new program data, it executes the bootloader. The remaining CAN messages then contain program data targeted for the Aux MCU. This allows easy programming of the controller's flash memory through CAN whenever an update to firmware is needed.

## Fault Monitoring

## **Digital Faults**

The Aux MCU has two digital inputs: PCM and BSPD faults. The digital faults are read when the CAN transmission function is called, and the fault status is then set according to the digital input. If a fault has been detected upon reading the digital input, a CAN message is sent indicating the fault.

## **Analog Faults**

The rest of the devices in which the Aux MCU monitors output analog data. The analog output devices include the IMD, BSPD, AIR control signals, SDC supply, and SDC return. The analog input for the IMD, BSPD, and AIR is handled when the CAN transmission function is called. Utilizing the analog to digital converter the MCU is equipped with, the inputs are decoded. If a voltage exceeding a cutoff is read from one of the devices, a fault is detected, and a CAN message is transmitted. The analog to digital converter for the SDC supply and return is handled separately within a function which executes at a rate of 40 kHz. If the voltage read from SDC supply and return exceeds a cutoff, a fault is detected, and a CAN message is transmitted.

# Data Monitoring

## **DCDC** Monitoring

The hotswap controller for the high voltage to low voltage DCDC converter has the ability to monitor voltage, current, power and enable control. All of these are handled by the Aux MCU, which are done via I2C. The controller is configured in order to output the current measurements based on the current sense resistor.

## Operation

In addition to fault monitoring and reporting, the data which corresponds to each fault is read and broadcasted. The digital input from the PCM holds no other data than its digital value. The voltage levels for all analog-driven faults are stored upon analog to digital conversion and then transmitted during the CAN transmission handler. Along with the BPD digital input, the BPD also outputs an analog voltage which is reported. The AIR and SDC supply and return only have its analog voltage values reported. The IMD data is output as a PWM wave which is decoded to identify the device's status. Other data the Aux MCU reads and reports is brake pressure and low voltage battery voltage.



# Inverter Motor Controller

For a highly in depth inverter design guide, check out our previous Quad Inverter design write up, found on our website

(https://www.wisconsinracing.org/wp-content/uploads/2020/10/WR\_Quad\_Inverter.pdf).

Wisconsin Racing's electric drivetrain is fully custom, from the motor, all the way to the three phase inverter. The inverter is designed using a six pack IGBT module with two custom PCBs mounted on the top. The first PCB is the control card, which is a purely low voltage board that houses the peripheral sensors and amplifiers for the inverter. It also houses a C2000 processor which has the team's custom motor controls. The second board is the gate drive, which drives the six gates of the IGBTs. The switching performance of the IGBTs were validated via double pulse testing, and motor parameters were gained via characterization as well.







# WR2 3-Phase 30kW Electric Motor

(This design writeup is provided for context, not to be robust documentation on the WR2 Motor) Engineer: Max Liben, 2019 Technical Director Revision: 2

## Overview

The WR2 Electric Motor is a 3-Phase, 12 slot, 10 pole Internal Permanent Magnet Synchronous Machine (IPMSM). The WR2 boasts a 37 kW peak power rating and is capable of achieving 27 Nm of torque. The WR2 is designed as an AWD in-hub motor, and each motor is driven by a custom Inverter Motor Controller.



### Features:

- Fractional slot concentrated windings
- 10 kW/kg power density
- VACOFLUX 48 laminations
- VACOMAX 262 SmCo magnets
- DMLS Cooling Jacket



Exploded View of Motor



Efficiency Map from Dyno Testing



# **Control Card**

Engineer: Quinn Sabin, 2022 Electrical Technical Director Revision: 2023.1 (2022.2) Dimensions: 61.9mm x 64.7mm Steady State Current Draw: 900mA (switching), 400mA (not switching)

## Overview

The Control Card is one of the 2 custom circuit boards that make up the Motor Controller. This dedicated low voltage board is secured on top of the gate drive and is the primary interface between the gate drive and all other systems on the car. Each control card is responsible for controlling 1 motor, making a total of 4 control cards located in the Penthouse.

### Features:

- F280049PZQ microcontroller with C2000 processor
- JTAG support
- Differential SPI transceiver for encoders
- Current sense and motor temperature amplifiers
- CAN transceiver
- Buck converter and LDOs for power regulation
- Connectors to interface with LV and HV systems
- 32 Kb EEPROM

## Circuit

This board contains hardware for system communication and firmware for controlling the motors. By interfacing with the car's ECU, motor encoders, and phase current sensors, the microcontroller generates 6 PWM signals that are sent to the gate drive. Different parts of the schematic are explained below and the PCB layout is detailed later.

#### Input/Output interface

Multiple T2M connectors from Samtec have been used to interact with various systems across the car. For example, receiving battery voltage, interfacing with CAN, interfacing with the gate drive, the encoders and the current sensors.

#### **Power regulation**

Switching regulators and LDOs have been used for regulating the input 12V to 5V and 3.3V. A reverse voltage protection circuit using a P-channel MOSFET protects the circuits while testing.





Bulk capacitors and decoupling capacitors have been added for all subcircuits. A dual output reference generator has been used to provide the ADCs and current sensors with

### Microcontroller

a stable voltage.

To describe the hardware associated with the microcontroller, its schematic symbol has been divided into various sections, grouped according to functionality.

For the power stage, regulators mentioned earlier are used to supply the microcontroller with appropriate voltage rails.

#### Power stage

Digital and Analog power pins are supplied with the required 5V and 3.3V. A ferrite bead on the 3.3V provides high frequency noise filtering. 0603 and 0805 packaged decoupling capacitors are added to fulfill the transient current demands of the microcontroller.

#### Clock and JTAG

An external LVCMOS MEMS oscillator , with 20MHz frequency is used. This AEQ-100 qualified external clock is chosen to provide an accurate and stable clock frequency to the microcontroller, with low susceptibility to vibration and high temperature that might be experienced by the Motor controllers.

A small SPST keypad switch is added to allow easy resetting of the microcontroller while programming and debugging

#### Communication peripherals

To interface with its CAN controller and connect with the CAN bus across the car, a CAN transceiver with an inbuilt 3.3V to 5V level shifter has been selected. Split termination using 62 ohm resistors and 1nF capacitor has been chosen to match typical characteristic impedance of the CAN bus. The RC network also forms a low pass filter and helps in preventing noise from affecting the bus. A common choke L200 has been

added to reject common mode noise and improve signal quality. ESD diodes D1 placed on CAN H and CAN L protect the transceiver from ESD events and transient surges due to the common mode choke.

SPI The microcontroller has 2 channels of communication interface. The first is used to communicate with the encoder on the motor. A full duplex RS-485 transceiver has been selected to generate the differential clock and data pairs that interact with the encoder.. The differential pairs are

terminated with 120 ohms and a 10nF capacitor based on the encoder datasheet recommendation. The termination prevents signal reflection by matching the characteristic impedance of the PCB traces and harness used in the encoders.











#### GPIO pins

Several GPIO pins on the microcontroller have been initialized to continuously monitor and communicate with the gate drive. Fault signals and reset lines have been routed between the gate drive and control card interface. Pull up resistors for these signals are located on the gate drive.

#### Analog pins

These pins are used for monitoring the phase current sense and motor temperature signals, as explained below.

#### **Enhanced Capture Pins**

Wisconsin

The gate drive IC sends both the IGBT temperature and the HV bus voltage over an altering duty cycle PWM. This duty cycle is read by the microcontroller using an Enhanced Capture Peripheral and then coverterted back to an analog voltage, then to the actual parameter.

#### Current sense amplifier

2 phase current sensors are selected to monitor the current flowing out of the IGBT module. A unity gain buffer has been employed to prevent loading the current sensors and providing a low output impedance to the ADC of the microcontroller. With a 1.65V reference, the sensor can measure +- 350A, and outputs a signal between 0.7904V to 2.5096V. An RC filter low pass filter with a cutoff frequency of 38kHz has been added to the output of the op amp to filter any high frequency noise. The operational amplifier

chosen has a low offset( $\pm 5 \mu$ V), low offset drift ( $\pm 0.2 \mu$ V/°C), and 10-MHz bandwidth. A high slew rate(20V/us) and short settling time(1us) are some of the other required features of this op amp to reduce delay in reading the signal. This hardware buffer has been repeated for both the phase current sensors, and the output filtered signal has been routed to the ADCs of the microcontroller.

#### Motor temperature amplifier

To monitor motor temperature, the control card includes an operational amplifier to buffer the thermistor signal before it is routed to the ADC. R205 provides the pull down resistor for the thermistor to monitor a maximum temperature of 200C. A 470 ohm resistor and 1uF capacitor low pass filter, with a cutoff frequency of 338 Hz, has been added to the output of the amplifier to prevent noise interference. Schottky diode D202 has been added to quickly discharge the C207 capacitor when the control card power supply is turned off, thereby safeguarding the microcontroller's ADCs.



GND

(1 \* 0.002456) + 1.65 = 1







#### Layout notes.

The control card is a 4 layer board and precise care has been taken with the PCB layout. Below is an overview of the primary layers in the board.

ø	Name	Material	Туре	Weight	Thickness	Dk	Df	
	Top Overlay		Overlay					
	Top Solder	Solder Resist 🛛 📟	Solder Mask		0.01016mm	3.5		
1	Top Layer	CF-004						
		PP-006				4.29		
2	GND	CF-004						
		FR-4 🚥	Dielectric		1.2mm	4.8		
3		CF-004						
		PP-006				4.29		
4		CF-004						
	Bottom Solder	Solder Resist	Solder Mask		0.01016mm	3.5		
	Bottom Overlay		Overlay					

PCB layer stackup

Some features of the layout are:

- Multiple ground planes and polygons have been spread across the board to ensure a stable noise free ground reference to all the subcircuits.
- Power polygons are used to handle the current capacity of subcircuits
- 3.3V polygon is placed between ground planes for decoupling
- Stitching vias are added to tie ground planes
- Decoupling capacitors are placed close to subcircuits
- ESD diodes placed next to connectors
- Thermal vias added for LDO
- Differential pairs are routed with minimum gap and trace width for impedance matching















# Gate Drive

Engineer: Kai Linsenmeyer, 2022 HV Electronics Lead Revision: 2022.2 Dimensions: 80mm x 83mm

## Overview

The Gate Drive is the galvanically isolated interface between the motor control microcontroller and the six pack IGBT module. The gate drive is built around the UCC21750QDWQ1 gate drive IC from Texas Instruments. The design was optimized for isolation, fast switching and robust fault protections.

#### Features:

- Six isolated gate drive circuits
- Isolated analog signal sensing
- 10 amp gate drive capabilities
- DESAT protection
- EMI rejection
- Fault and power good (ready) signals



## Low Voltage Side of Gate Drive



## Operation

The low voltage side of the gate drive features the interface between the gate drive and the motor controller. The ready pin allows sensing of if the gate drive is receiving proper power on both the Low Voltage and High Voltage sides of the circuitry. The fault pin signals if there is a



desaturation fault or other error. Both of these fault signals are built into the fault monitoring of the motor controller. A low pass filter was placed on the PWM input into the gate drive to eliminate noise, while being slight enough to not cause a significant phase shift in the data output. The UCC21750 features a PWM output that corresponds to an analog input on the high voltage side of the circuit. The motor controller monitors two of these outputs in order to read the DC bus at the IGBT as well as the internal thermistor of the IGBT. It does this using an enhanced capture peripheral.



## High Voltage Side of Gate Drive

### **Fault Protection**

The gate drive circuitry has desaturation, miller clamping short circuit protection, ESD protection and not shown in the schematic, overcurrent protections implemented by the motor controller.

The DESAT pin charges the blanking capacitor via a current source at VDD. While the device is off or inactive, an internal pull down prevents false faults. During the normal operation, the diodes are forward biased, keeping the voltage at the DESAT pin clamped to their forward voltage. During shoot through or a short circuit event, the capacitor will charge up to VDD and trip a DESAT fault at 9V.

During a short circuit event, the miller capacitance of the IGBT will become charged, raising the voltage seen at the gate drive and CLMPI pins to be higher than 15V. This triggers a fault that stops switching and this is seen by the motor controller which stops switching on all gates.

## Gate Drive

The gate drive can source a peak of 10A which allows for fast switching speeds without requiring the use of external amplification. A gate resistance of one ohm was chosen because the max potential difference at the gate is 30V, the internal gate resistance of the IGBT is minimum two ohms, which equates to 10A of current.

To prevent false turn on and faster discharge -15V is used for the low side voltage.

See the IGBT page for matlab code pertaining to the switching losses for the gate drive IC.

## Isolated Power Supply



The TEC 3-1223 was selected as it met the design parameters for isolation, dual voltage outputs and current rating for switching. The input power to the supply is fed through a common mode choke to prevent back coupling from the high voltage side of the system into the power of the low voltage side.





**H** 709654000

₹C

DC bus sense

DC bus sense

HV+ HV-

HV+











# Infineon FS200R07A5E3\_S6

## Overview

The Infineon FS200R07A5E3\_S6 is a 6 Pack IGBT Module that is used in our custom motor controllers. This IGBT module tightly packages 3 discrete IGBT half bridges all in one compact design.

#### Features:

- 6 IGBTs in 1 module with easy bolted connections
- 705V Rated
  - It is best practice to derate by 50% so it is perfect for our 323V peak system.
- 200A Rated, optimized for 20kW loads
  - Each motor will be at or below 30kW for most driving scenarios so we will be operating near peak efficiency.
- Internal Thermistor
  - $\circ$   $\;$  Allows for easy thermal monitoring and management  $\;$
- Integrated heat spreader and small form factor
  - Allows us to easily package our motor controllers side by side and cool all 4 with a single cold plate.
- Low Switching Losses, Kelvin connection reduces parasitic inductance in gate drive





Typical appearance





# **Double Pulse Test**

Engineers: Pratham Sarvaiya, Alex Quilliam, Kai Linsenmeyer Contributions : WEMPEC (Wisconsin Electric Machines and Power Electronics Consortium)

## Overview

Switching IGBTs at high frequencies to drive AC motors results in substantial switching losses due to the finite interval it takes for IGBTs to transition between on and off states. To reduce switching losses, the gate drive has been optimized by providing high source/sink strength to the gates, and reducing gate to emitter loop parastics.

### Double Pulse Test goals

Double pulse testing is a way to validate and tune the designs, the goals are as follows:

- Characterize switching losses(Turn on losses, Turn off losses, Dead time losses, Diode reverse recovery losses)
- Evaluate effect of switching frequency on efficiency
- Evaluate gate drive parastics
- Tune gate drive variables for future designs
- Tune dead time to reduce losses

### Test procedure

The test is conducted on a half bridge from the IGBT module. The MCU on the control card has been used to generate pulses that control isolated Gate Drive ICs on the gate drive board which in turn control the IGBTs. Double pulse tests are typically conducted on a half bridge( low side and high side switch).





Testing equipment



Half bridge

This test requires an inductor, capacitor and a half bridge.

The double pulse test can be segmented into 3 periods as indicated in the diagram.

1. Period 1 - Turn on low side IGBT with a gate pulse. Current flows from inductor to low side IGBT depicted by 1 in the diagram. This is done to reach test current through the inductor. After a specified period, turn off low side IGBT to measure turn off losses.

- 2. Period 2 Measure dead time losses as inductor current flows through body diode of high side IGBT as indicated by 2 in diagram
- 3. Period 3 Turn on low side IGBT to measure turn on losses and reverse recovery losses. Current flow indicated by 3. Turn off IGBT after a period, wait for inductor current to fall to zero as it demagnetizes through the body diode of high side IGBT.

Voltage and current measurements are made during these 3 periods.

## Sizing Inductor, Capacitor, and Pulse duration

To closely represent our system, the test has been conducted at 300V, from low currents(10A) to high currents(100A)

Sizing Inductor and the length of Pulse 1 and 2

The inductance determines how long it takes for current to build up in the low side

IGBT. A lower value of inductor is beneficial because a higher  $\frac{di}{dt}$  limits the temperature

rise of the IGBT. However, a higher inductance helps maintain a constant current during the off period so that turn on and turn off losses can be measured at similar test currents. Hence, an optimum operating point was calculated and chosen.

#### Minimum Inductance

A minimum inductance has been chosen to limit drop in current between the pulses. **The break time was chosen to be 1us** as that is the dead time used in the motor controls code. With this dead time where neither switch is on, the dead time losses can be characterized. Based on the estimated values, **Minimum inductance = 3.5 uH.** 



# Target inductance

Target inductance can be calculated using the following equation (based on the equation for voltage across an inductor) :

$$L_{load} = t_1 \times \frac{V_{dc} - V_{ce(sat)}}{I_{test}}$$

Vce(sat) = 1.45V (from datasheet)

The inductance should be chosen such that at maximum test current, the pulse duration t1 is not higher than 100uS. This puts a limit on a **maximum inductance of 192 uH.** Calculating a few different values to ensure t1 is higher than 1us(typical turn on time for IGBT) but lower than 100us, **Target inductance is chosen to be = 70 uH** 

Based on the target inductance, the following pulse durations estimates were used to build up to the test current.

#### Duration of Pulse 3

Since period 3 is only required to calculate turn on losses, its pulse duration should not be long to prevent overheating the IGBTs. Typical turn on time for each IGBT from the datasheet is 0.17us(turn on delay + rise time). Adding in a safety factor and taking into account a possibly slower rise time due to differences in test conditions and actual gate drive parasitics, t3 is chosen to be 2 uS.

#### Creating inductor

8 gauge wire coiled around a spool has been used as an air core inductor. 70uH target inductance was achieved with the help of an LCR meter. Air core inductor has been chosen because the low inductance value and no risk of magnetic saturation at high frequencies results in highly linear performance.

#### Sizing capacitor

During the first pulse, the high current is supplied by the DC bus link capacitor. The required capacitance can be calculated by using the energy balance equation between an inductor and capacitor.

To reduce excessive bus voltage drop, 2 150uF capacitors in parallel, with a total capacitance of 300 $\mu$ F, have been used. This limits  $\Delta V dc$  to 5V at around 100A load current.

#### Required capacitance: 2 150uF in parallel

#### Voltage and current measurement **Differential probe**

To measure the IGBT voltage waveforms, an isolated high voltage differential probe from Tektronix has been used. The 50 MHz high bandwidth of this probe is sufficient for accurately measuring rise and fall times. The probe's wires have been twisted to reduce common mode interference.

#### Rogowski coils

To facilitate isolated current measurement, a 30 MHz 1200A PEM Rogowski coil has been used. Compared to traditional current transducers, Rogowski coils provide high accuracy for AC currents, and since they do not contain a magnetic core, saturation is not a concern at higher currents.

#### Oscilloscope

A Tektronix 4 series, high bandwidth oscilloscope has been chosen to capture the voltage and current waveforms.

Differential probes and Rogowski coils both contain electronics to accurately measure signals, which means that they both have different propagation delays in sending signals to the oscilloscope. Based on the datasheet values, the probes have been deskewed prior to testing. Irregularities in deskew can lead to erroneous measurements and can over/under estimate switching losses and timing measurements.







Oscilloscope





DC link capacitor



#### Data analysis

Matlab has been used to process and analyze the data. As an example, data analysis for one measurement has been shown below. Waveforms at 115A load current

The oscilloscope waveform has been displayed on the right as a reference. The pulse duration was increased to achieve the desired 115A test current. As can be seen in the graph, the IGBTs take a finite amount of time to turn on and turn off. The rise/fall in VCE coincides with a fall/rise in emitter current IE. This intersection leads to switching losses.

To calculate the turn on and turn off energy loss, Voltage and Current have been multiplied in Matlab to give power, the results of which have been integrated over the time period to give energy loss.

A moving average filter on the power plot helps improve the calculation by removing random noise.

This process has been followed for all the waveforms and the data has been presented below.

#### Turn on and turn off times

As per Infineon standards for measuring timing data, T(on) has been measured from 10% of VGE rise to 10% of IE rise, and T(off) from 90% of VGE fall to 90% of IE fall. This helps remove uncertainties in the starting and ending 10% of the waveforms. As expected, an increase in load current leads to decrease in turn on time. The higher load current charges the output capacitance of the IGBT quicker, resulting in a faster turn off. As for turn on times, a minor yet general rise in turn on times has been measured. A higher load current

provides negative feedback to the gate-emitter loop, increasing the time it takes for the IGBT to fully turn on.

#### Switching losses calculation

The following calculations have been made to depict the switching losses plot on the right:

- Turn on and turn off losses across load current have been calculated as shown earlier.
- Dead time losses have been calculated based on a 1us deadtime, and respective load current.
- IGBT shoot through losses due to diode reverse recovery are a part of turn on losses
- Diode reverse recovery losses have been calculated based on current overshoot and reverse voltage(300V) applied to the diode.













Switching losses



- Turn on, turn off, dead time, Diode reverse recovery, all switching losses have been added.
- Load current is considered the peak of the AC waveforms, and so all load currents have been converted to their RMS values
- Energy loss x Switching frequency gives Power loss
- Test has been carried out at 25°C
- Switching losses for all 6 IGBTs have been summed
- Measured energy loss data is within reasonable range of IGBT datasheet values, hence verifying the results of this test

#### Total Power loss

Conduction losses have been calculated based on load current, VCE(saturation) from the IGBT datasheet, and Vf(body diode forward voltage drop) from the datasheet as well. Summing switching losses and conduction losses gives total Inverter IGBT losses. A pie chart depicts the division of total Inverter losses at 80A RMS.





Slew rate data in Inverters can offer several insights into Inverter and Motor performance. Increasing slew rates reduces switching losses but increases bearing currents in the motor. Based on voltage and current rise/fall times, slew rates have been depicted below. These plots follow expected trends.

For turn off slew rates, increasing load current increases di/dt because the turn off time decreases. Dv/dt increases with current because while dv stays the same(300V to 0V), dt decreases.

For turn on slew rates, increasing load current increases di/dt because even though dt increases with current, di increase has a larger effect. Dv/dt decreases because while dv stays the same(300V - 0V), turn on time increases.



Loss distribution at 80A RMS, 20kHz 25°C





Turn off slew rates

#### **Closing notes**

Switching data provides valuable insight into Inverter performance. While these tests were conducted at 25°C, higher die temperature will have an overall effect of increasing total losses. However, this test still validates the gate drive design and provides a way to tune various parameters.

# DC Link Capacitance Selection

Engineer: Jason Sylvestre

# Overview

DC Link capacitors are required in traction inverters to provide a stable DC voltage and supply the transient current demands of the motors. Driving motors with an AC waveform introduces higher

frequency harmonics onto the DC bus. To reduce the AC ripple voltage and current, bulk capacitors are added. This inverter uses UL32 Film capacitors from Electronic Concepts for their low ESR and ESL,

### Features

- 150 uF
- Rated for 500VDC
- 12nH ESL
- 0.47 mOhms ESR

## Selection parameters

To define the input parameters, maximum bus voltage is 320V and Inverter switching frequency is 20kHz.

The primary role of the capacitor is to provide the high frequency AC current that cannot be supplied by the battery pack. The motors may draw varying amounts of RMS current during its operation. To support upwards of 100A RMS current draw, the capacitors need to provide high frequency AC currents. The UL32 film capacitors can provide up to 110A(RMS) at 25C, and have a frequency rating of 82.2kHz.

The second role of the capacitor is to reduce the ripple voltage of the DC bus. Based on the RMS current flowing through the capacitor, the voltage ripple can be estimated using the following equation:

$$C_{min} = \frac{I_{out} \times dc \times (1 - dc) \times 1000}{f_{SW} \times V_{P(max)}}$$

Increasing the capacitance limits the peak to peak voltage, however, increasing capacitance after a point leads to diminishing returns. Plotting the peak to peak voltage as a function of capacitance shows that 150uF is a good value to limit the ripple.

### DC voltage rating

The UL32 capacitors are rated for 500V but can withstand 1.3X Rated Voltage for 1 minute. Peak voltage on the DC bus results from the back EMF of the motor at highest speed. Following guidelines and recommendations on rating the capacitors, 500V DC is high enough for operating conditions.



DC link capacitors - UL32 - Electronic Concepts

# Inverter Firmware/ Motor Controls

Engineers: Will Sixel, Evan Wildenberg, Christian Schuster, Alex Quilliam

## Overview

The inverters run custom motor controls firmware using field-oriented controls with integrated fault protection. Custom controls allow a greater deal of freedom and flexibility to change things to fit the needs of the car.

### Features

- Field-oriented controls
- Fault protection
- Flux weakening

# Controls

The inverters run a standard field-oriented controls algorithm that can be broken down into five stages:

- 1. Sensing
- 2. Transforms
- 3. Adjustment
- 4. Modulation
- 5. Actuation

All five of these stages are run in the main controls loop at 20khz.

### Sensing

The controls algorithm relies on input from several sensors:

- Two Hall-effect sensors, which measure the phase current of two of the phases
- An absolute rotary encoder, which measures the position of the rotor
- DC bus voltage
- Motor winding and IGBT thermistors

These sensors are read in a variety of different ways. The hall effect sensors and motor winding thermistors are analog inputs. The motor winding uses a lookup table to go from sensed voltage to temperature. The DC Bus voltage and IGBT thermistor is a PWM that is read using enhanced capture and the corresponding duty cycle is mapped to the analog voltage sensed by the gate drive IC's analog input pin. The encoder is communicated via SPI that goes to a differential pair transceiver and has 13 bits of resolution.





### Transforms

After the phase currents and rotor position have been sensed, they are passed through several transforms, which will convert the input currents from a stationary to a rotating reference frame.

#### **Clarke Transform**

The Clarke transform converts three sinusoidal input currents into sinusoidal flux-producing and torque-producing output currents. The flux-producing (d-axis) current produces a magnetic field, but it does not contribute to torque. The torque-producing (q-axis) current is offset 90° from the flux-producing current, and is what creates torque. Since we only sense two phase



Three phase inverter output

Torque- and flux producing currents (from a stationary reference frame)





Torque- and flux producing currents

(from a stationary reference frame)

Torque- and flux producing currents (from a rotating reference frame)



currents, we use a modified Clarke transform that takes two input currents.

#### Park Transform

The Park transform converts the stationary dq reference frame of the Clarke transform into the rotating reference frame using the angle of rotor.

#### Two Phase Current Sense

To allow for tight packaging constraints and fewer components, the controls are all done using two current sensors instead of three. This is possible using the relation of phases A + B + C = 0 in a wye configuration motor. This allows the Clarke transform to use this equation to do a substitution for one of the three phases.

#### Adjustment

Two PI controllers are used to adjust the measured current to match the reference current. One PI controller is for d-axis current, and its reference is always zero, except during flux weakening. The other PI controller is for the q-axis current, and its reference is set by a second-order polynomial based on the motor torque constant, which converts a torque request into a q-axis current reference. The PI controllers initial gains are based on the first order transfer function of the motor, 1 / (R + Ls). These parameters were measured directly on each motor and allowed for initial tuning with manufacturing differences. The gains and the bandwidth of the PI controllers that were tuned based on drive data to minimize current oscillations.
## Modulation

The corrected currents are converted back into the stationary dq reference frame through the inverse Park transform. Using a direct-digital implementation of PWM generation, the turn-on times of each switch are calculated using the sinusoidal d-axis current and the switching frequency. A switching frequency of 20khz was chosen as an optimal tradeoff between being computationally efficient and minimizing motor harmonics.

# Actuation

The turn-on times calculated during the modulation step are passed to the PWM hardware, and dead time is added. Dead time ensures that two switches on the same phase are not closed at the same time, which would cause shoot-through. The dead time was calculated based on the turn-on and turn-off times of the IGBTs, with an additional safety factor giving a dead time of 1us. This deadtime was also validated in double pulse testing.



## Flux Weakening

Flux-weakening allows the motors to spin faster than their rated speed, although with reduced torque. Flux-weakening is achieved by setting a negative d-axis current reference, which generates a magnetic field that opposes the field of the permanent magnets. Flux weakening is not currently in use on the vehicle as due to a raised pack voltage the max speed of the motor is at or near the maximum speed reached during an acceleration run. Due to the inherent risk with flux weakening and the loss of efficiency during the entry condition, it was decided to not run flux weakening with the current drivetrain configuration.

# **Fault Protection**

The inverter performs a number of fault checks to ensure the safe operation of the inverters and motors. Phase overcurrent is checked in hardware, triggering an interrupt that will immediately stop the inverter. Additional fault checks include gate power, software overcurrent, DC bus under and overvoltage, encoder overspeed, and motor and IGBT over temperature.



# **Precharge Board**

Engineer: Alex Grotelueschen, HV Electronics Lead Revision: 2023.1 Dimensions: 82mm x 65mm

# Overview

The Precharge circuit board is a completely non-programmable circuit board that houses our Precharge circuitry and HV Indicator LED circuitry. The Precharge makes use of a dual channel comparator to monitor the HV bus voltage output to control both circuits. The Precharge board is sectioned into two sides, the left side holds all the Low Voltage circuitry, and the right side holds all the high voltage circuitry. The two sides are separated by a visible air gap, and only cross with an optocoupler and the precharge relay.

#### Features:

- Dual channel comparator
- Uses a feedback circuit to determine when the HV output has been charged to 90% to end precharging
- 3, HV to 12V Linear Dropout Regulators to power comparator, and HV Indicator LED
- 12V controlled compact Precharge Relay from TE Connectivity
- 3 footprints available to adjust the AIR control signal capacitance



Precharge PCB Layout

## Precharge Circuit

The Precharge circuit uses voltage sensing resistor dividers, and a comparator to monitor the HV bus, and stop Precharging when the output voltage is measured to be 90% of the input. *Rules Reference: EV.6.6* 

## Operation

When the shutdown circuit enters the accumulator, it is used to close the HV- AIR, and sent to the Precharge circuit to trigger the start of precharging. The Precharge relay is closed using the SDC in, and HV+ Pre-AIR is connected to off-board precharge resistors. At the same time, HV- is connected to the LDOs and resistor dividers, starting the comparator measurement. When the two resistor dividers are equal, it signals that the tractive system is at 90% of the accumulator voltage. The output of the comparator is grounded, closing an optocoupler which will switch another relay on the low voltage side. When this relay is closed, the precharge relay will open, and the shutdown circuit is sent to the HV+ AIR, finishing precharge.



#### Low Voltage Circuitry

The shutdown circuit into the accumulator (SDC RTN) is connected to the coil of the Precharge relay, which connects HV+ Pre-AIR (HV+ IN) to the offboard precharge resistors. There are three, 3.3k Ohm Resistors connected in parallel to have an equivalent 1.1k Ohm series resistance during precharge. There are 4, 150uF capacitors in parallel, one for each motor controller, on the bus. This gives us a precharge time of ~1.5 seconds.



Low Voltage Precharge Circuitry

$$\tau = -\ln(0.1) RC = -\ln(0.1) * 1100 * 0.0006 = 1.52 s$$

When precharge is measured to be done, the CHARGED signal is pulled high, and the second relay is tripped. When this happens, pins 12 and 9 are opened, and pins 5 and 8 are shorted together. This removes Ground from the precharge relay, opening that relay, and AIRCOIL+ is shorted to SDC RTN, sending 12V to the HV+ AIR, closing it.

Both relays have flyback voltage protection diodes. C101, C102, and C103 provide footprints for large electrolytic capacitors. Our model of AIR requires 4 Amps of inrush current to close, so extra capacitance is helpful to decouple the GLV bus and Shutdown Circuit Supply power during these events.

## **High Voltage Circuitry**

To measure the high voltage bus at a safe level, voltage dividers are used to step down the voltage. We measure 0.99% of the output bus voltage (COMP OUT), and 90% of 1% of the input voltage (COMP IN). At our max

and 90% of 1% of the input voltage (COMP IN). At our max battery pack voltage (323V), COMP IN will be 2.9V. This sets the 90% Voltage the comparator will trip at.



$$COMP IN_{MAX} = 323 * (2k/(100k + 100k + 2k)) * ((100k + 100k)/(100k + 100k + 20k)) = 2.9V$$

Since COMP OUT is set at 0.99% of our pack voltage, at max pack voltage the comparator will trip when HV+ OUT is 293V

$$HV +_{OUT} = COMPOUT * ((100k + 100k + 2k)/2k) = 2.9 * ((100k + 100k + 2k)/2k) = 293.6V$$



The comparator used has an open drain output, meaning the output will be grounded when  $IN- \ge IN+$ . The output of the comparator is placed on the low side of the optocoupler, so that current can flow through the optocoupler to the comparator when COMP IN = COMP OUT, signaling the end of precharge.

# HV Indicator LED Circuit

The HV Indicator circuit utilizes the same voltage divider, comparator, and supply voltage that the precharge circuit uses. The comparator compares a set voltage to the stepped down bus voltage, and when it hits that threshold, the LED is turned on.

#### Rules Reference: EV.6.7

60V HV Comparator ISO 12V ISO 12V When HV+ is at 60V, COMP+ is at 0.594V Indicator LED is currently part no. 19080255 rated for 17mA DC R110 R112 787 0.1% U100B COMP OUT R117 60V INDIC 60V COM 0.01 1% 0494 250NRHE LM2903AVODRO Å R111 866 0.1% HV. HV-

COMP OUT will equal 0.59V when HV+ OUT is equal to 60V, so that is fed to the negative input of the comparator.

 $COMP OUT_{60V} = 60V * (2k/(2k + 100k + 100k)) = 0.59V$ 

This voltage is created with a voltage divider with the isolated 12V generated from the LDOs. 60V COMP - = 12V \* (866/(866 + 16.7k)) = 0.59V

When COMP OUT < 0.59V, the output is grounded and voltage is dropped across R112. When COMP OUT > 0.59V, the comparator output is open, and current flows out to the 60V indicator LED. A 250mA fuse is in series with the output voltage to protect the accumulator in the case of a short circuit.



Isolated 12V is generated from three 12V Linear dropout regulators in parallel. Each LDO can supply a maximum of 10mA, and the Indicator LED is rated for a maximum of 17mA. With R112 in series, the LED can draw a maximum of 15mA. R113, R114, and R115 are placed on the outputs of the LDOs so that they can balance their output voltages across them.



# Layout Notes

The precharge uses an interesting board cutout to separate the HV and LV circuits on the board. This provides a very visual separation between the two sides on top of physically separating them. On the high voltage side, the HV+ traces are kept on the outside of the board, and the voltage steps down as you get closer to the center towards the comparator. On the LV side, thick traces are used for the AIRCOIL+ signal to help with the inrush current experienced from the AIR closing.



2D View of the Precharge PCB

#### Stackup

The Precharge is a standard 2-layer, 1 oz copper PCB. It was manufactured from JLC PCB, and assembled by the team when it arrived.

	Name	Material	Туре	Weight	Thickness	Dk
	Top Overlay		Overlay			
	Top Solder	Solder Resist 🛛 📟	Solder Mask		0.01016mm	3.5
1	Top Layer		Signal		0.03556mm	
	Dielectric 1	FR-4 🚥	Dielectric		0.32004mm	4.8
2	Bottom Layer		Signal	1oz	0.03556mm	
	Bottom Solder	Solder Resist 🛛 🥅	Solder Mask		0.01016mm	3.5
	Bottom Overlay		Overlay			

Layer Stackup of the Precharge PCB





# PCB Printout





# **Discharge Board**

Engineer: Panithan (Com) Boonkert, Electrical Hardware Team Member Revision: 2023.1 Dimensions: 75mm x 67mm

## Overview

The Discharge board is a non-programmable board housing the discharge circuit. This board's purpose is to allow the high voltage capacitor to discharge to the power resistor, which is controlled by a relay.

#### Features:

- Power resistor integrated onto the PCB so that the entire circuit is one piece
- Relay to control the high voltage from the capacitor to the power capacitor
- Reverse voltage protection for the relay



# Circuit

The discharge circuit consists of a normally

closed relay that connects a power resistor between HV+ and HV-. The shutdown circuit goes high when the car is getting ready to drive, and that signal opens the relay on the board, disconnecting the power resistor. New to this year, the power resistor is mounted onto the board so that the entire circuit is one package. Diodes are placed in locations to protect against coil flyback and reverse voltage on the GLV side.

Rules Reference: EV.6.6.3

## Schematic





# **Steering Wheel**

Engineer: Will Kramer, Electrical Hardware Team Member Revision 2023.1 Dimensions: 157x136mm (Main Board), 199x120mm (Daughter Board) Steady State Current Draw: 920mA

# Overview

The Steering Wheel acts as the main method of exchanging information between the driver and the vehicle: it gives critical drive information through an LCD and enables the driver to quickly tune vehicle parameters on-the-fly. The central processor is the Raspberry Pi Compute Module 4 (**CM4**), shown below in green; this single-board computer runs Linux for rapid firmware development, supports WiFi, USB, and HDMI for system debug, directly drives the LCD image, and facilitates most driver-vehicle communication. The Pi interfaces with other ICs to expand its capabilities, including a GPIO expander and ADC for driver I/O interfacing and a CAN controller/transceiver pair for vehicle communication. The Steering Wheel main board includes an onboard 5V/3.3V power supply for low-voltage electronics and an LCD backlight driver circuit, enabling 12V input power; it also features HDMI, USB, CAN, and MicroSD interfaces for use in system bring-up, debug, and functionality extension. A separate daughter board holds the driver I/O devices, including push-buttons, potentiometers, and an LED indicator strip, reducing the need for internal harnessing and allowing for future modifications to the I/O layout. This design aims to combine our previous vehicle-specific steering wheels into one which is powerful, flexible, and robust enough to be used for several years.

## Features

- Three-board design: Main Board (shown right in blue) holds peripheral circuitry and Raspberry Pi SBC (shown in green), Daughter Board connects to driver I/O
- Raspberry Pi Compute Module
  4 (CM4) SBC runs Linux, manages high-level I/O
- 4.3" LCD shows critical drive information, picture and backlight drivers onboard
- Programmable LED strip indicates ideal shift timing or other high-priority information



Steering Wheel Main Board with CM4 SBC

- Flexible LCD, WiFi, and externally-accessible USB interfaces enhance vehicle diagnostics access
- Integrated multi-rail power system with ESD, reverse-voltage, over-temperature, and overcurrent protections, rated for future expansions to power needs
- Robust debug support allows rapid system bring-up: HDMI, USB, CAN, and WiFi interfaces, system control jumpers, indicator LEDs, test points, and functional block labels
- CAN controller/transceiver pair enables communication with on-vehicle peripherals
- Digital GPIO expander and ADC pair power driver I/O, unplanned future features
- MicroSD interface enables CM4 booting without onboard eMMC

# **Primary Power Stages**

The primary power stage of the Main Board is a 12V to 5V buck regulator circuit with internal overcurrent and over-temperature protections and discrete ESD and reverse-voltage protection. The 5V rail powers all other circuitry on the Daughter and Main Boards: LDO linear voltage regulators onboard the CM4 step 5V down to 3.3V and 1.8V rails (the 1.8V rail is not used), and the LCD backlight driver (discussed later) steps 5V up to a higher driving voltage. The 5V power rail is designed to provide up to 4A to downstream devices.

## **Reverse-Voltage and ESD Protection**

A P-channel MOSFET (Q1) provides reverse-voltage protection from the 12V system input. When 12V is applied with correct polarity, the MOSFET's body diode allows current to flow from drain to source, with a forward voltage drop of  $V_F \leq 1.2V$ ; the source

node voltage can then rise to up to a minimum of  $V_{IN} - V_F = 12.0V - 1.2V = 10.8V$ . Because the gate node is tied to ground,

 $V_{GS} = V_G - V_S = 0.0V - 10.8V = -10.8V$ . Looking at the MOSFET's datasheet, that  $V_{GS}$  corresponds to a drain current  $I_D$  of more negative than -20A, significantly higher than the maximum buck regulator current draw of 2.36A (discussed below); the MOSFET then enters its saturation region, allowing current to flow efficiently to the buck

regulator. When 12V is applied with reversed



Main Board ESD and Reverse-Voltage Protection

polarity, the circuit 12V input node then has a voltage of -12V with respect to circuit ground; the highest voltage in the circuit is then 0V, so  $V_{GS}$  cannot become negative and the MOSFET stays in its cut-off region.

A 24V bidirectional TVS (transient voltage suppression) diode is added to the system's 12V input to keep the buck converter input within its maximum recommended voltage of 36V while also allowing for the use of higher input voltages, up to 18V. A 3A fuse is selected to limit the buck regulator current draw (discussed below, maximum draw is 2.36A) in case of unexpected faults.

# 5V Buck Regulator



Main Board 5V Buck Regulator

The Texas Instruments TPS5450 5A buck regulator IC (*U1*) was selected as the 5V rail power supply IC for its high current rating and efficiency, and for its integrated over-current and over-temperature protections. The CM4 draws up to 2.0A from the 5V rail, the LED indicator strip (discussed later) draws up to 0.5A, and all other electronics would draw up to a combined 0.6A, for a total 5V rail current draw of up to 3.1A; typical current draw is lower, at 1.4A for the CM4, 0.4A for the LED strip, and 0.2A for other electronics, for a total of 2.0A. Based on these figures, the 5V power stage was designed with a capacity of up to 4A in order to enable future upgrades to system functionality. The IC can be switched off via a jumper (*VCC\_OFF*) for debug purposes; test points are also provided for use in system debug.

The TPS5450 uses a voltage sense pin to control its output voltage; the sense voltage is compared to an internal reference voltage of 1.221V, and any difference between the sense and reference voltages is amplified to increase or decrease the output voltage. R2 and R3 form a voltage divider to drive the sense pin: R2 is selected as 10k, per the datasheet's recommendation that all designs use this value; R3 is selected as 3.16k, giving an output

# WISCONSIN

voltage of 1.221V \*  $\frac{10k+3.16k}{3.16k}$  = 5.085V. This accounts for parasitic voltage drops in the PCB power traces and ensures an acceptable voltage level at device power inputs.

The minimum size of the inductor *L1* is calculated based on Equation 5 of the TPS5450 datasheet, shown below.  $V_{OUT(MAX)}$  is 5.19V, using the same output voltage calculation as above but assuming worst-case tolerances on *R2*, *R3*, and the TPS5450 internal reference voltage;  $V_{IN(MAX)}$  is 18V by design;  $K_{IND}$  is given as 0.25 per the TPS5450 datasheet;  $I_{OUT}$  is selected at the typical expected system current draw of 2.0A; and  $F_{SW(MIN)}$  is the minimum IC switching frequency, given in the datasheet as 400kHz.

$$L_{MIN} = \frac{V_{OUT(MAX)} * (V_{IN(MAX)} - V_{OUT})}{V_{IN(MAX)} * K_{IND} * I_{OUT} * F_{SW(MIN)}} \qquad L_{MIN} = \frac{5.19V^{*}(18V - 5.085V)}{18V^{*}0.25^{*}2.0A^{*}400kHz} = 18.6uH$$

Equation 5 of the TPS5450 Datasheet

Based on this calculation, an 18.6uH or larger inductor is determined to be required; a 22uH inductor with a rated current of at least 4.26A is selected, per Equation 7 of the TPS5450 datasheet (the actual inductor selected is rated for 10A due to supply shortages at the time of design):

$$I_{L(PK)} = I_{OUT(MAX)} + \frac{V_{OUT} \times (V_{IN(MAX)} - V_{OUT})}{1.6 \times V_{IN(MAX)} \times L_{OUT} \times F_{SW(MIN)}} \qquad I_{L(PK)} = 4.0A \times \frac{5.085V^{*}(18V - 5.085V)}{1.6^{*}18V^{*}22uH^{*}400kHz} = 4.26A$$

Equation 7 of the TPS5450 Datasheet

Per the TPS5450 datasheet, the catch diode (*D2*) must have a reverse voltage rating of  $V_{IN(MAX)} + 0.5V = 18V + 0.5V = 18.5V$ , a peak current of at least  $I_{L(PK)} = 4.26A$ , and a low forward voltage drop for high efficiency; a Schottky diode with a reverse voltage rating of 45V, a peak forward current of 7A, and a typical forward voltage of 440mV is selected. 120uF of input capacitance and 30uF of low-ESR output capacitance are also provided per the datasheet's design recommendations.

At an input voltage of 12V and a typical load of 2.0A, the TPS5450's performance curves (see figure 8-2, right) suggest an efficiency of roughly 92%; this equates to a 12V input current of

 $I_{IN} = I_{OUT} * \frac{V_{OUT}}{V_{IN} * Efficiency} = 2.0A * \frac{5.085V}{12V * 0.92} = 0.92A.$ 



Figure 8-2 of the TPS5450 Datasheet



121

In the worst-case scenario of a 10V input voltage and 4.0A load, an efficiency of roughly 88% is expected; this gives a maximum input current draw of 4.  $0A * \frac{5.19V}{10V^{*}0.88} = 2.36A$ .

#### 3.3V Supply

3.3V is supplied to low-power devices via a 400mA LDO linear voltage regulator which is onboard the CM4; the maximum current consumption of all 3.3V devices sums to 280mA.

## **CAN** Interface

The CM4 does not support CAN directly in hardware, so a discrete controller/transceiver pair is used to provide CAN support. The MCP2515T-I/SO CAN controller (U5) connects to the CM4 over SPI at 3.3V signaling levels. A 16MHz oscillator was chosen to support both 1Mb/s and 500kb/s CAN bus operation through internal clock dividers; a single-ended oscillator is preferred for its design simplicity and reliability. The CAN TX/RX logic signals then route to the TCAN1042(H)VDQ1 CAN transceiver (U4). The transceiver acts as the physical interface layer and features an onboard logic level shifter which enables interpreting of the 3.3V-level signals from the CAN controller. Indicator LEDs are included to show the activity of these 3.3V intermediate signals for debug purposes. The external-facing CAN signals of the transceiver then pass through a common-mode choke for noise suppression; a bidirectional TVS diode protects against ESD events, and 62-ohm terminating resistors (R4, R5) are included



Main Board CAN Controller, Transceiver, and Protection

per the CAN specification and may be fitted.

# I/O Expanders

## Digital GPIO Expander

In order to account for the large number of digital GPIO pins used up by the LCD RGB parallel interface on the CM4, a 16-channel digital GPIO expander is included onboard. 14 channels are routed to the Daughter Board, where 11 are currently used for push-buttons; the remaining unused inputs are routed to the Vehicle I/O Connector for general-purpose future use. The digital GPIO expander communicates with the CM4 via I<sup>2</sup>C.

## Analog-to-Digital Converter

An 8-channel ADC is included for analog inputs, such as the two driver IO potentiometers. 6 ADC inputs are routed to the daughter board connector, where 2 are currently used; the 2 remaining inputs are routed to the Vehicle I/O connector for future use. 10k series resistors and 0.1uF ADC-side capacitors to ground form a debounce circuit for each ADC input, with a time constant of  $\tau = R * C = 10kohm * 0.1uF = 1ms$ ; this gives a filter corner frequency of  $f_{corner} = 1/(2\pi * \tau) = 1/(2\pi * 1ms) = 159Hz$ , acting as a debounce for the potentiometers and filtering out any high-frequency interference noise. Like the digital GPIO expander, the ADC also communicates with the CM4 via I<sup>2</sup>C.

# LCD

The Steering Wheel supports an LCD interface as its primary method for communicating

information to the driver. The LCD control signals are driven directly and generically by the CM4; a 60Hz, 4.3-inch, 480x272 pixel display is currently used, but a wide variety of other displays may also be substituted in the future. An onboard constant-current backlight driver provides display illumination for direct-sunlight visibility.

## Data Signals

The LCD is driven using an RGB parallel interface: 24 lines of data are provided in parallel for each pixel (8 each for red, green, and blue), and clock and sync inputs communicate to the LCD the position of the pixel intending to be



Selection of Main Board LCD Control Signals

# WISCONSIN

drawn at any given instant. The original intention was to use an HDMI LCD for its ease of use, but there are very few HDMI displays on the market which are bright enough to be viewed in direct sunlight, and even fewer which are small enough for our application. Driver chips exist which can convert HDMI signals into RGB parallel, but they require higher clock frequencies than our RGB parallel LCD could support and had very fine-pitch contacts that were outside our manufacturing capabilities. So instead of using HDMI, the CM4 is configured to drive the RGB parallel signals directly using its high-speed GPIO pins. The CM4 has 30 GPIO pins available, and a full RGB parallel interface would utilize 28 of those; for this reason, a 5-6-5 color depth which uses only 20 GPIO pins was used. The 5-6-5 color depth refers to the use of 5 bits to represent the red channel, 6 for green, and 5 for blue: this reduces the number of display signals used at the expense of increased quantization artifacts in the LCD picture. For each color channel, unused LSBs which are not driven directly are tied to driven MSBs in-order, as shown in the above image; this was found to produce the lowest average quantization error when reducing the color depth from 8-bit and still allows full black and full white to be displayed.

# **Backlight Driver**



## LCD Backlight Driver Circuitry

The LCD backlight illuminates the display; it is constructed from multiple series strings of LEDs and requires a constant-current driver for proper functionality. The LCD panel chosen for the current Steering Wheel design has a typical forward drive voltage of 25.6V at the recommended 40mA drive current, so a boost converter (U2) is selected that can increase the main 5V supply voltage to a level suitable for the backlight. *R67* is used to set the system drive

current using a feedback pin on the backlight driver IC: the feedback pin creates a feedback loop to control the switching output such that the feedback input matches an internal voltage reference of 204mV. A 5.1 ohm resistor gives a drive current of  $I = \frac{V}{R} = \frac{204mV}{5.1ohm} = 40mA$ .

The switching inductor *L*3 is chosen at 4.7uH per the boost converter IC's datasheet recommendation; the inductor is rated at 2.2A, which is significantly larger than the 40mA drive current or the inductor peak-peak ripple current of 0.71A, as shown below. The rectifying Schottky diode *D*5 is selected to carry the DC current of 40mA, withstand a maximum reverse voltage of 40V (the maximum that the driver IC could ever produce), and carry a peak current of 0.40A (40mA DC plus half the peak-peak ripple current of 0.71A calculated below).

$$\Delta I_{L(P-P)} = \frac{1}{L \times (\frac{1}{V_{OUT} - V_{IN}} + \frac{1}{V_{IN}}) \times F_S}$$

Equation 4 of the Backlight Driver IC's Datasheet  $\Delta I_{L(P-P)} = \frac{1}{4.7uH^*(\frac{1}{25.6V-5V} + \frac{1}{5V})^* 1.2MHz} = 0.71A$ 

A PWM input from the CM4 can reduce the brightness of the LCD for indoor operation, if desired. The CM4 pin

which is connected to this input has a pull-down resistor, but the driver IC's PWM input is active high, so an N-channel MOSFET is used as a signal inverter to ensure that the LCD backlight turns on immediately after system startup, before the CM4 can configure that pin as a HIGH output.

# Primary Connectors and Interfaces

# Vehicle I/O Connector

This 16-pin connector is the primary interface between the steering wheel and the rest of the vehicle. It carries 12V power, CAN signals, four digital I/O channels, two analog input channels, and the signals needed for a USB device interface. T2M connectors are selected for their high reliability in past Wisconsin Racing designs and for their automotive qualification option. T2M connectors are rated for 3.8A, greater than the maximum Steering Wheel input current of 2.36A; however, extra 12V and GND pins are included for easier harnessing and redundancy in case of connector failure.



Main Board Vehicle I/O Connector

# Daughter Board Connector

This 28-pin connector interfaces between the Steering Wheel Daughter Board and Main Board in order to relay all driver I/O signals besides those displayed on the LCD. Fused 5V and 3.3V power, two high-speed GPIO pins from the CM4, 14 digital GPIO channels, and 6 analog inputs

# WISCONSIN

are provided to the Daughter Board. All inputs are also pulled up to 3.3V to prevent them from floating. In order to have the minimum possible spacing between the main and daughter boards, a low-profile connector was selected: this connector does not have a keyed shroud, so pins 6 and 21 are reserved as key pins to prevent misalignment during assembly. The Daughter Board connector is intended to provide a generic interface between the two Steering Wheel circuit boards such that the Daughter Board can be changed at some point in the future without needing to create a new Main Board; this enables future expansions or modifications to the driver I/O layout while reducing the amount of



Main Board to Daughter Board Connector

harnessing needed on the interior of the Steering Wheel enclosure, as previous experience has shown harnessing to be prone to mechanical fatigue or failure over time.

#### External USB Connector

The CM4 has the ability to act as a USB device, allowing for data transfer between itself and an external computer while the Steering Wheel is powered. This in turn enables access to diagnostic information or rapid updates to the Steering Wheel firmware without requiring a WiFi connection. The USB interface is broken out to a USB Micro-B connector directly on the Main Board, and it is also routed to a waterproof 4-pin connector on the backside of the Steering Wheel enclosure. 6.1V TVS diodes prevent ESD damage on the data lines, and the USB connector's



Main Board USB Mini-B Connector

VCC line is used for host presence detection per another official design by the Raspberry Pi foundation, designers and manufacturers of the CM4.

#### MicroSD Card Slot

Due to supply shortages for the CM4, we were unsure if we would be able to source one with onboard eMMC storage; this would then require that an external SD card be used to hold the Steering Wheel firmware. Mechanical considerations made an external SD card a generally



poor candidate for use in high-vibration environments such as our vehicles, but a MicroSD card slot was added just in case we were not able to source an ideal CM4: thankfully, we have not needed to use this slot while driving. A small driver IC (U8) limits the current which the MicroSD card uses to 200mA in case of a fault, per the CM4's design recommendations.



Main Board MicroSD Card Slot and Circuitry

## **Debug Connectors and Interfaces**

#### HDMI Port

HDMI0, the primary HDMI port of the CM4, is broken out to a full-size HDMI connector for use in firmware development. This port was used to verify expected functionality of the CM4 during early board bring-up, and it may be used to display more detailed diagnostic information that would fit on the LCD when used in a test environment.



Main Board HDMI Connector and Circuitry



#### WiFi Support

On select models, the CM4 has onboard WiFi with a small built-in antenna. In order to ensure adequate signal integrity, a keep-out zone underneath this antenna was required on the Main Board PCB. An external antenna may also be connected to the CM4 in the future for greater signal range and strength.



Main Board WiFi Copper Cutout, Under CM4

## System Control Jumpers and Indicators

The CM4 supported several status indicators and control flags which relay or determine its functionality; a selection of these interfaces was chosen for use in board bring-up and debug. The SYS\_PWR LED indicated whether the primary 5V power stage is active; the CM\_ACT LED indicates that the CM4 is actively processing, or it flashes error codes is the CM4 detects a fault or is unable to boot; and the remaining jumpers can be used to manipulate the CM4's boot sequence or functionality, either as part of debug or as a



System Control Jumpers and Status Flags

semi-permanent disabling of the WiFi, Bluetooth, or ROM interfaces to save on system power.

## **CAN Debug Port**

This 4-pin connector gives access to the CAN bus via a team-standard interface; we have several peripheral boards and adapters which use this pinout, so it was included in the design for convenience in debug.



Main Board CAN Debug Port















W/FCONTCINI	V LUCCINCIAN	0	
23 - Display	Revision 1.2		
Title Steering Wheel 2.	Designer Will Kramer	Date: 5/14/2023	File: Display.SchDoc



# Steering Wheel Daughter Board

The Daughter Board, as previously mentioned, is intended to be flexible and open to future changes. The current design holds 6 push-button switches, two potentiometers, a 5-way multi-switch for menu navigation, and 12 individually-addressable LEDs that form an indicator strip for high-priority information. These elements are all attached directly to the daughter board, which acts as a harness to the main board via the Daughter Board connector and a carrier for all surface-mount components present.

#### **Push-Buttons**

6 push-buttons connect to the Daughter Board for arbitrary driver functions; their functionality is determined in firmware and may be modified in the future. A 10k resistor and 1uF capacitor form a debounce circuit with a time constant of  $\tau = R * C = 10kohm * 1uF = 10ms$ , which is equal to the 10ms contact bounce time specified for the push-buttons used.

#### Multi-Switch

A 5-way switch (up, down, left, right, center) is mounted at the center of the Steering Wheel front plate; this multi-switch is intended to be used for menu navigation via the LCD in order to view diagnostic information or to change infrequently-accessed system parameters.

#### Potentiometers

Two potentiometers are provided for inputs with several possible states. These pass to the Main Board ADC, where

onboard RC filtering smooths out sliding noise as the potentiometers rotate.

#### LED Indicator Strip

A strip of 12 individuallyaddressable LEDs sits at the top of the Daughter Board, above the LCD. These are driven by a single control signal line from the CM4 which is stepped up from a 3.3V logic-level up to 5V by *U1* and then daisy-chained through the full run of 12 LEDs. On our internal combustion vehicle, this LED



LED RGB Data Logic-Level Shifter and First Two LEDs in Indicator Strip Series

indicator strip is used as a shift indicator, telling the driver when to shift for maximum power or efficiency; the indicator can also be used to display any other high-priority information so the driver can access it at a glance or in their peripheral vision.



Push-Button Debounce



# Steering Wheel Code

Principal Engineer: Sean Staggs, Firmware Lead

Contributing Engineers: Amitai Debiche, Jai Sinha, Max Wu, Sanjay Menon, Summit Koegel Revision 2023.1

# Overview

The steering wheel code is the collection of all the digital files necessary to make the 2023 steering wheel generational upgrade work. This primarily consists of the operating system and its configuration, CAN bus code, code for controlling all of the I2C devices, code for the display, and a backend to connect all of the processes.

## Features:

- Operating system that allows for leveraging a scheduler and higher abstraction object-oriented programming languages for graphics
- Multi-process architecture for better efficiency and speed
- Custom CAN Controller driver for the MCP2515
- Non-blocking IO control scheme
- Driver focused display

The necessity of all of the above features is the result of a hardware switch to a Raspberry Pi Compute Module 4

(CM4) for the core functionality of the system. The most important part of this was that it was a switch from a microprocessor based system to a fully-fledged computer which provided the processing power for using a python code base that greatly sped up development.

# **Operating System Configuration**

The first part of making the system work was the choice and configuration of an operating system. To minimize the amount of load on memory and the processor and maximize compatibility with the hardware, the team used Raspberry Pi OS which is an Unix fork.

# Config.txt

Most of the settings for the operating system and its hardware reside in the config.txt. This file dictates which peripherals are enabled, links them with their drivers, and sets core functionalities like the baud rates for all of the peripheral busses.

## Cmdline.txt

In addition to the core functionality built in Config.txt, some of the unix boot settings reside in cmdline.txt. Changes were made to help expedite the boot process for the system.





Backend



The architecture of the steering wheel's code was built to leverage the multiple processors present within the CM4 by forking off different processes to handle the different functionalities of the code base. This consisted of building separate processes for CAN, I/O, and Display. This was done to leave one processor to handle any other processes that the operating system needed to run (like ssh). The original plan was to leverage a multi-threaded approach to each sub-process, but during hardware integration testing it became apparent that each process could only run a single thread at a time, forcing a transition to a series of non-blocking processes that would be run from a control loop.

The separate subprocesses are constantly communicating information between each other to ensure the full functionality of the wheel. To facilitate this, as the processes are forked off, they are provided with queue objects that allow them to enqueue the most recent data that they possess whenever there is something to update. The result of this is that each control loop is constantly checking for new data to be transferred in. The one danger with this implementation is that the data transfer of data can easily start hogging the vast majority of a processor's time compared to the core functionality of a given process.

# CAN Bus

Given the automotive nature of this application, much of the information sent and received by the wheel interacts with the CAN bus. This presented a significant challenge because the CM4 that the wheel uses has no native capabilities for receiving or transmitting CAN. As such, a MCP2515 CAN Controller is used as an intermediary between the TCAN transceiver and the CM4 using SPI.

## MCP CAN Controller

The MCP libraries built for use with Python all faced significant config and functionality challenges that proved impossible to resolve. As such, the team developed its own MCP driver to enable the use of CAN on the wheel. The MCP driver uses the Adafruit Circuit Python SPI library to run the interaction between the CM4 and the MCP. The driver is broken into 3 main commands with a couple of auxiliary commands that are mainly for internal use by the driver itself.

The first command provided by the driver is the init command that comes when a driver object is created. This command sets the three CNF registers on the MCP that provide bit-synchronization and baud rate settings. It then sets the MCP to normal mode, allowing for it to start functioning. Finally, the driver sets the RX\_CTRL registers to not use any filters and to take all messages it receives, including error frames.

The driver also provides a receive function that takes advantage of the MCP's RX\_STATUS registers to determine if a message is waiting to be read. If a message is waiting, it gets read, processed and returned with values given for id, length, and payload. This allows for an added abstraction layer so that a separate function can take the respective DBC files and turn the payload into meaningful values.

The last functionality of the driver is the ability to transmit CAN messages. This takes the ID and payload and reformats it for the MCP and then decides which transmit buffer to place it in. This leverages the TX\_STATUS register to determine if there is space for a message to be

transmitted. Much like receive, the main point of this is to provide an abstraction layer that a DBC parser would be able to interact with to pack and transmit CAN.

# CAN Rx and Tx

Using a DBC Parser, the CAN Rx and CAN Tx functions handle the packing/unpacking of messages to interface with the MCP Driver. The future of these functions is an auto-parser that can create the same high quality packing/unpacking without wasting the time of a DBC fluent firmware member.

## **Internet Connectivity**

The current internet functionality of the CM4 is limited to the use of UDP messages to forward the CAN data to and from a server. This means that the CAN process checks for UDP messages on its port in a non-blocking manner and then proceeds to parse them and send them off for transmission. If there is internet connectivity, the CM4 also transmits a UDP message for each received CAN packet.

# GPIO

The wheel comes with a series of dials, buttons, and a joystick that allow for the driver to interact with the wheel. The long term use of this is to provide the ability to interact with a series of debugging displays as well as to set event modes and even trigger data playback.

# Display

The display subprocess serves to receive CAN and GPIO stimuli and turn it into a series of displays that help the driver and team understand what the vehicle is doing. In its current form, there is a drive display for each car that provides key information that allows the driver to push the car to its extremes and know if there are any critical issues facing the car.

## Tkinter

The TKinter displays are set up with a frame for each "display" and a series of subframes for each piece of displayed information. Each frame has a label and a value with fonts and placement built to allow for quick reading by an experienced user while maintaining usability for novices. This architecture has the benefit of being able to quickly add and remove displays to those available on the wheel.

## **CAN Struct**

The current implementation currently keeps a full copy of the CAN struct maintained by the CAN process to allow for the display of any and all information that could be of use to the displays. The problem with this is that the entire object has to be copied and replaced every time one is dequeued. In the future, it would likely be wise to create a cascading set of objects to allow for the transmission and copying of much smaller objects that can then find their place within a larger object. This will create minimal load on the front-end developers while greatly increasing the efficiency of the data transfer process.



# Systems Integration

Engineers: Alex Grotelueschen, HV Electronics Lead Pratham Sarvaiya, LV Electronics Lead

# Harnessing

Wiring harness design and manufacturing is a cross sub-team effort between the electrical hardware team and the controls team. The team starts by listing every system component in a centralized excel spreadsheet. Components are then added to a RapidHarness schematic, and signals are connected. Lengths can then be added between nodes, and a wiring table is generated for manufacturing. The team then manufactures the entire harness in house.



#### Features:

- Main harness is split into 2 sections for ease of manufacturing and maintenance
- Field serviceable Deutsch DTM connectors used as much as possible
- RapidHarness schematics that greatly aid in manufacturing process

#### **Front Harness**

The front harness contains everything in front of the firewall. This includes dash buttons, APPS potentiometers, brake pressure sensors, the steering wheel, and more. The front harness connects to the rear harness with a "bulkhead" connector, a high pin density Deutsch connector.

#### **Rear Harness**

The rear harness contains everything behind the firewall. This provides the communication between the vehicle controller, the BMS, and the motor controllers. Various other important sensors and switches live in the rear harness as well. Both the "Penthouse", the motor controller enclosure, and the accumulator connect to the rear harness using high density Deutsch Autosport series connectors. New to this year, the Solidworks model of our vehicle was used to get measurements prior to the car being manufactured in order to start harnessing earlier.

#### Penthouse Harness

The penthouse harness connects the PCBs internal to the penthouse to the main car harness through a bulkhead connector. The Battery Main Board and 4 Motor Controllers live in the penthouse and need to be harnessed.









# Datalogging

Datalogging our car has been one of the most important tools to diagnose issues and tune our car to maximum performance. On our electric car, we use a Vector GL1010 Logger in conjunction with Vector's CANape software to analyze our data. The GL1010 has the capability to monitor and record 2 CAN buses with the capacity of about ~1 hour of the car being on.

# Low Voltage Load Characterization

The Smart Power Distribution unit's embedded microcontroller tracks current load on each fuse and transmits that information over CAN every second. This information helps us optimize our fuse panel for future iterations.

The loads on each fuse are as follows:

AH 1	AH 2	Fuse 1	Fuse 2	Fuse 3	Fuse 4	Fuse 5	Fuse 6
ECU AH	Inverter GLV x4	Steering Wheel	Rear Sensors	Left Pump	Right Pump	Shutdown Circuit	Front Sensors
BMS GLV							ECU GLV

Using data pulled from our Data Logger, we are able to obtain the amount of current we are pulling on our Low Voltage bus at different states of the car. Below is a snapshot from a day of testing. We can divide the states of the car into 3 areas, LV on, HV on, and Inverter switching.



#### Results

Extrapolating from the data obtained above, we can determine how certain systems behave during different states of the car. Included below are certain traces that are particularly interesting.

-				
	Fuse	LV On	HV On	Driving
	AH1	0.65A	0.65A	0.65A
	AH2	1.3A	1.3A	3.69A
	Fuse 1	0.35A	0.35A	0.35A
	Fuse 2 (Brakes on)	1.7A	1.7A	1.7A
	Fuse 3 (Pump on)	Peak 1A	Peak 1A	Peak 1A
	Fuse 4 (Pump on)	Peak 1A	Peak 1A	Peak 1A
	Fuse 5	0.08A	0.85A	0.85A
	Fuse 6	Peak 0.35A	Peak 0.35A	Peak 0.35A
-				

#### Inverter Switching

The second always hot fuse powers all four motor controllers. At steady state, they pull a collective 1.3A, which is about 400 mA per inverter. However, when the inverter is switching, the gate drive starts to pull more low voltage current, increasing the combined current to 3.7A, or about 900mA per inverter.



#### Brake Light

From a first glance, fuse 2 looks like it is randomly oscillating. But tracing the fuse current with the brake light signal shows that the current increases when the brake light is on. This massive increase makes sense because of the new brake lights we are running this year.





#### SDC

From testing in the past, we know that our AIRs take 4 Amps of current to close. Over CAN, it is hard to capture this inrush current at the current timescale of messages. However, every once in a while it is captured. Included is one such instance, where the transient current peaks at ~3.2 A. When bandwidth is increased, we would likely see it spike to 4A



## Mechanical Characterization

Datalogging is also used to support the growth of our mechanical design. Various sensors are integrated onto the car to capture valuable information that validates the designs from the mechanical team. Some of the sensors we have logged include:

- Shock Pots
- Coolant Temperature Sensors
- Coolant Flow Sensors
- Strain Gauges
- Tire Temperature Sensors

## EMI

Electromagnetic interference (EMI) was taken into account during the design phase and sensitive signals were given various protections, including filters, common mode chokes, voltage buffers, and diff pair signals. However, EMI still showed up in some places on the car. The solutions to these issues are below.

## Motor Encoders

The motor encoders were especially susceptible to EMI noise due to proximity to the motor, the inverter and the length of the wires. Additionally, the high clock rate and small margin for error exacerbated this issue. To mitigate this EMI, the team added shielding to all encoder wires in and outside of the penthouse as well as ferrite beads. The motors cooling jackets were all ensured to be fully grounded with low resistance paths to chassis grounds. Problems arose when a ground wire broke, and it was decided to transmit ground to the encoder sensor through the shielding. This greatly degraded the performance of the shielding as it was now carrying the sensor's return current.


#### Brake Plausibility Device

When the pumps would turn on and off the BPD would sometimes fault. This happened due to the brake pressure sense line picking up EMI. This was fixed by shielding the pump wires as well as adding a low pass filter to the input.

## **Tractive System Efficiency**

Tractive system efficiency has been evaluated at continuous operating conditions, the summary is as follows:

Sub system	Efficiency @ peak power out	Efficiency @ nominal power (40kW)	Test
Battery Pack	93%	96%	Drive Testing data
Motors	90%	95%	Dyne Data
Inverters	95%	97%	Double Pulse
Motor + Inverter Efficiency	86%	92.2%	
Overall TS Electrical Efficiency	80%	88.4%	

#### **Battery Pack Efficiency**

Battery Pack Efficiency was calculated by using drive testing data. The BMS, pack current sensor and high voltage daughter board was utilized in order to determine the voltage drop of the battery pack at various load currents while driving. This allows a calculation of open cell and closed cell voltages at different loads and efficiency can be calculated from there.

#### **Motor Efficiency**

The motor efficiency was able to be derived from doing dynamometer testing at UW-Madison's power electronics labs. From there, sweeps of speed and torque while monitoring input electrical power and output mechanical power allowed solving for efficiency.

#### **Inverter Efficiency**

The inverter's efficiency was derived using the double pulse testing method. Details of this test can be found above in this report.



## **Combustion Car Designs**

### Overview

Wisconsin Racing also builds a car for the combustion competition. The unique electrical designs for the combustion car can be found here: https://www.wisconsinracing.org/e-documents/.

### **Combustion Specific Designs**

- Controls Signals Board
  - Converts a 7 channel digital gear position sensor to a single analog output
  - Converts a frequency from a wheel speed encoder to an analog voltage to enable the reading of the wheel speed sensor
- Dyno Power Distribution Unit
  - A simplified and robust version of the Smart Power Distribution Unit for dyno testing
- ETC Switch Panel
  - A switch panel that fits in the harness in order to speed along the ETC test
- CCar Carrier Board
  - Carries the Power Distribution Unit, the Brake System Plausibility Device and houses the latching circuitry and relays for the combustion cars shutdown circuit.
- Tire Temperature Sensor
  - An ESP32 MCU, CAN transceiver and IR temperature sensor for custom tire temperature sensing
- Eddy Current Dynomometer
  - A Labview implementation for a dynamometer for the combustion engine
- Systems Integration

### Shared Designs - Only Found in 2023 ECar Design Release

- Smart Power Distribution Unit
- Steering Wheel
- Brake Systems Plausibility Device



# Contact Us

To contact the team, email: <u>team@go.uwracing.com</u> To contact the 2023 Electrical Technical Director with questions related to this document and its designs, email: c.schusterr@gmail.com

#### Thank you for reading! Good luck from Wisconsin!



