

# Formula Hybrid Final Presentation EESD 2019-2020

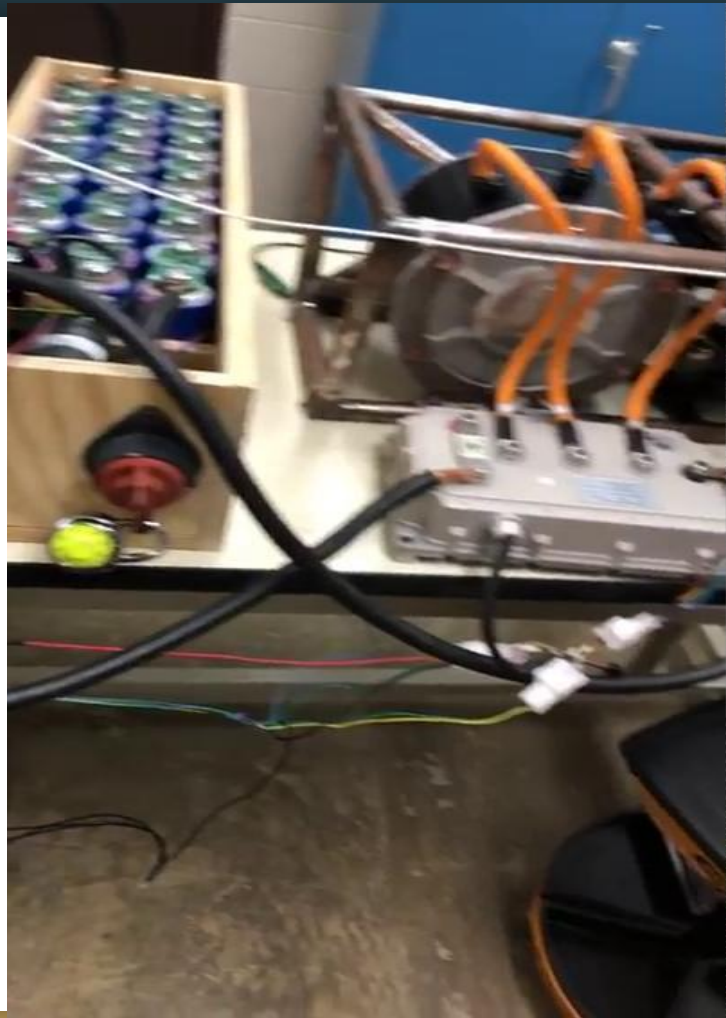
Parker Imlay, Will Kuang, Musodiq Ogunlowo, Neal Smith, Bridget Taylor



UNIVERSITY OF  
NOTRE DAME

College of Engineering





# The ND Hybrid Car

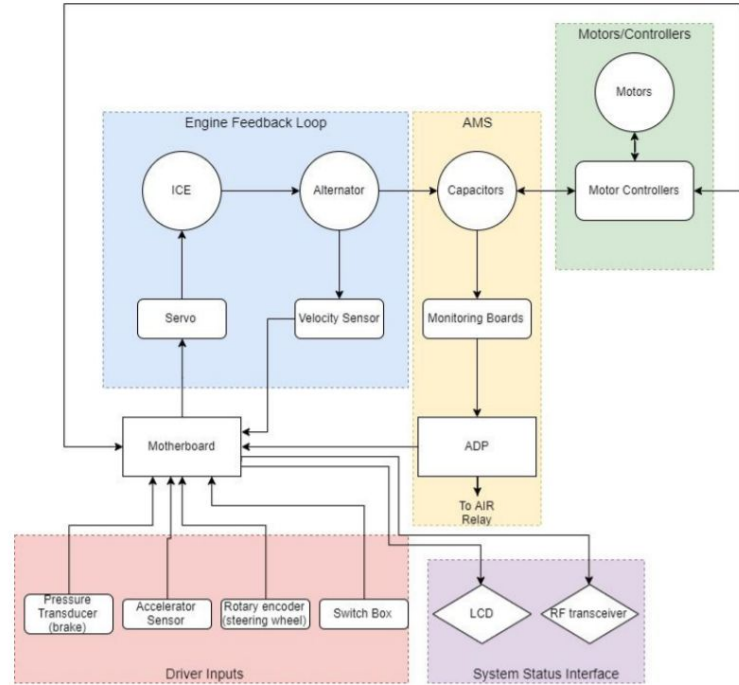
Motors/Controllers - Bridget Taylor

AMS (Monitoring Boards and ADP) - Parker Imlay

System Status Interface (GUI/RF) - Musodiq Ogunlowo

AMS (Capacitors) - William Kuang

AMS (Thermistors) - Neal Smith



# Initial Project Goals

- Determine if modification of the source address on the Kelly controllers was a possibility and implement a method to differentiate between the CAN messages of the two motor controllers
- Driver Inputs
  - Mitigation of false error signals
  - Incorporation of hydraulic pressure transducer
- Engine Feedback Loop
  - Stable RPM under changing load
  - More inputs allowing for multiple control states
- Motor/Controller/Generator
  - Torque vectoring (once addressing issue is solved)
- System Status Interface
  - LCD improvements and Transmission System GUI with data log
- Accumulator Management
  - Active cell balancing to minimize low-voltage shutdown events

# Adapted Project Goals

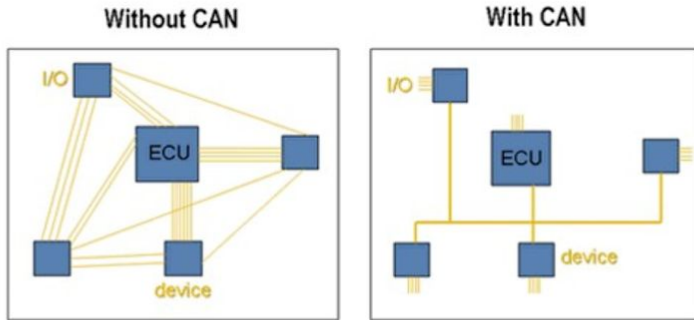
- Documentation of CAN protocol and motor controllers (Bridget)
- Documentation of AMS system (Parker)
- GUI for off-system monitoring (Musodiq)
- Independent capacitor charging and discharging (Will)
- Temperature monitoring on capacitors (Neal)

# Project Outcomes

- Finalized a method to distinguish between each motor controller and developed an extensive operation guide with documentation of testing
- Complete documentation of hardware and software for the accumulator management system
- Improvement of User Interface for Off-track monitoring computer and documentation for RF GUI
- Complete documentation of external charging and discharging process/calculations for capacitors
- Created documentation for wiring and thermistor implementation plan for ultracapacitors

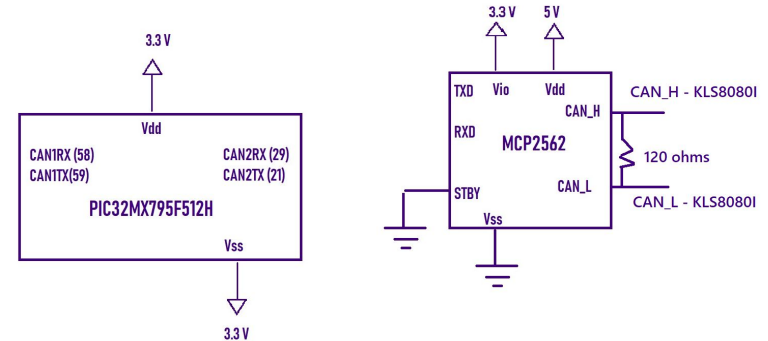
# Motor Controllers and CAN Network

## Overview of CAN



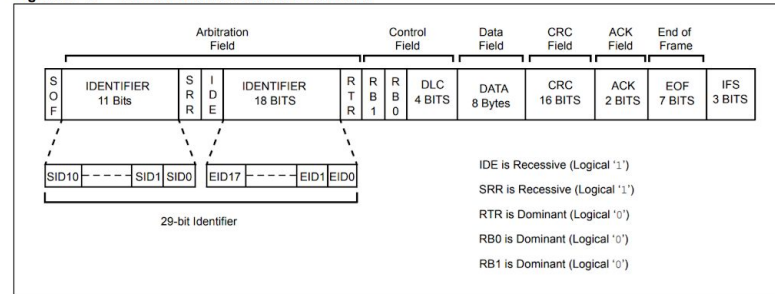
CAN network topology (National Instruments, 2019)  
<https://www.ni.com/en-us/innovations/white-papers/06/controller-area-network--can--overview.html>

Kelly Controller 14401-80801  
 Brushless DC Motor Controller  
 - 144 V, 160 A continuous current



CAN Node Set-Up on Motherboard

Figure 34-5: Format of the Extended Data Frame

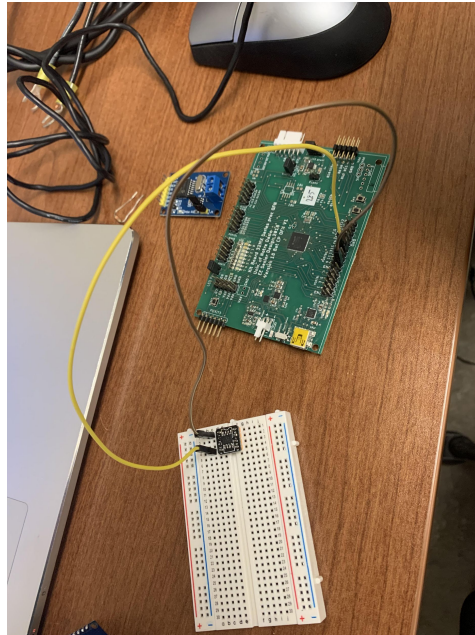


Extended Message ID Format - Section 34. Controller Area Network (CAN) - Microchip Family Reference Manual - PIC32MX795F512H

# Motor Controllers and CAN Network

## Testing Procedures

- Arduino as transmitter CAN node with standalone CAN controller + transceiver
- Senior Design demo boards with MCP2562 CAN transceiver as receiver node with motor controller as transmitter node
- Debugging with PICKit3



Demo board and MCP2562 on breadboard acting as CAN receiver node



Keysight oscilloscope showing the CAN\_L signal from the Kelly motor controller



# Motor Controllers and CAN Network

## Operation Guide and Documentation

- Introduction to CAN networks
- Decoding the CAN differential signal
- CAN Hardware Set-Up
- Testing with Arduino, demo boards, and controllers
- Debugging with logic analyzer and oscilloscope
- Guide to Kelly Controller software
- Explanation of CAN acceptance filters and masks
- Guide to CAN module data storage

### Operation Guide: CAN Communication + Kelly Controllers Formula Hybrid Senior Design Team, 5/7/2020

#### Introduction: the benefits of using CAN communication

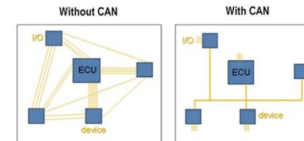
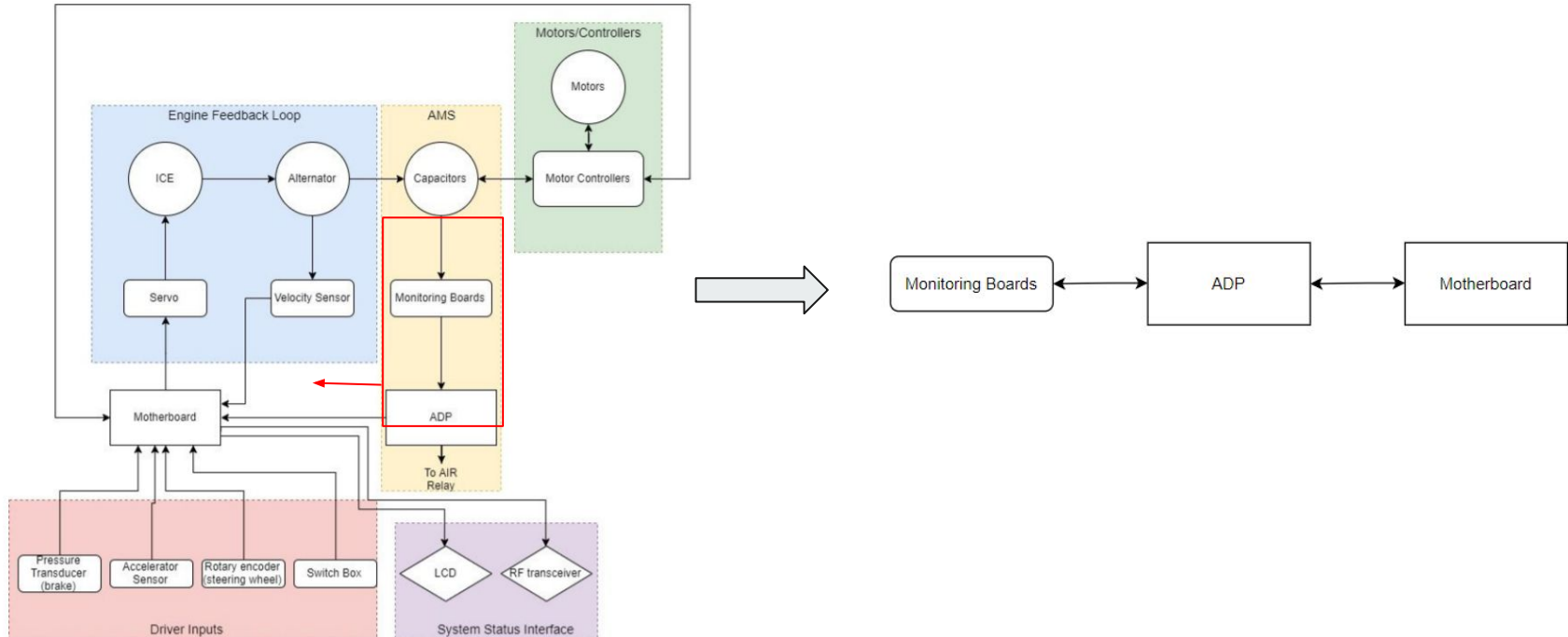


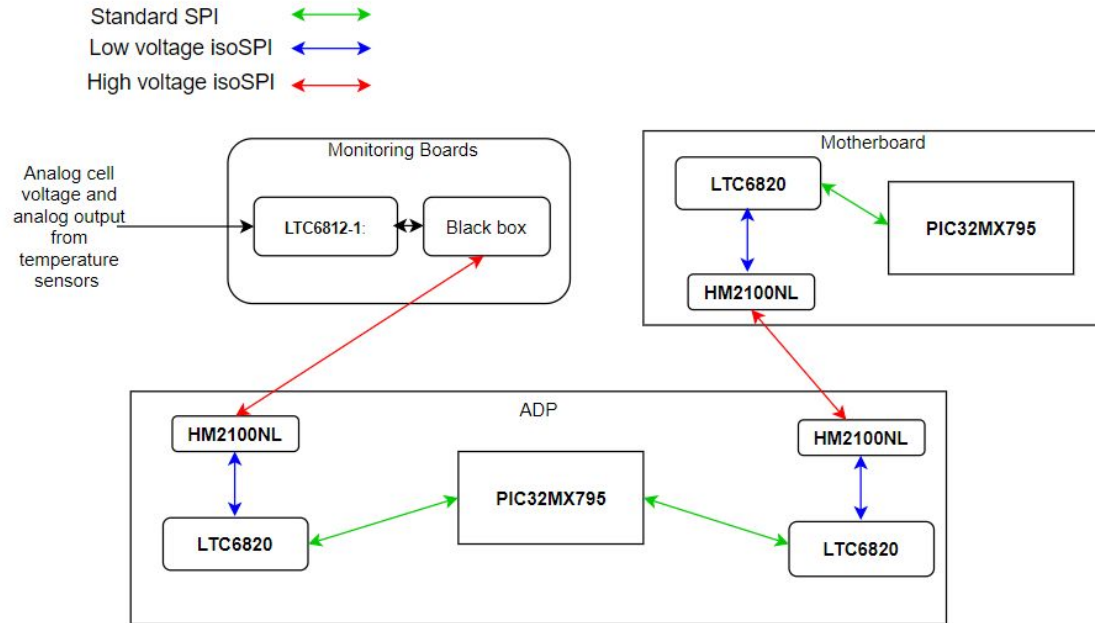
Figure 1. CAN network topology minimizes wiring (National Instruments, 2019)

CAN Bus protocol enables many IO devices/ECUs called nodes within an automotive vehicle or other system to communicate with each other with minimal wiring required (see Fig. 1). Each node broadcasts a message to all other nodes on the bus, similar to the operation of walkie-talkies at the simplest level. Some nodes, such as the engine or brake controls, relay more critical information than others, such as the air conditioning, and need priority if two nodes try to send a message at the same time. The arbitration process will determine which node can transmit its message first. All nodes can ignore messages that do not apply to them with buffers and filtering set up in the microcontroller. Overall, the low-cost, centralized processing, and robustness to electromagnetic interference makes CAN protocol so appealing for car manufacturers.

# AMS



# AMS



# AMS

- The ADP has two isoSPI modules
- It is the master in the interface between ADP and monitoring boards
- It is the slave in the interface between ADP and motherboard
- In our system, the transformer is replaced by the HM2100NL

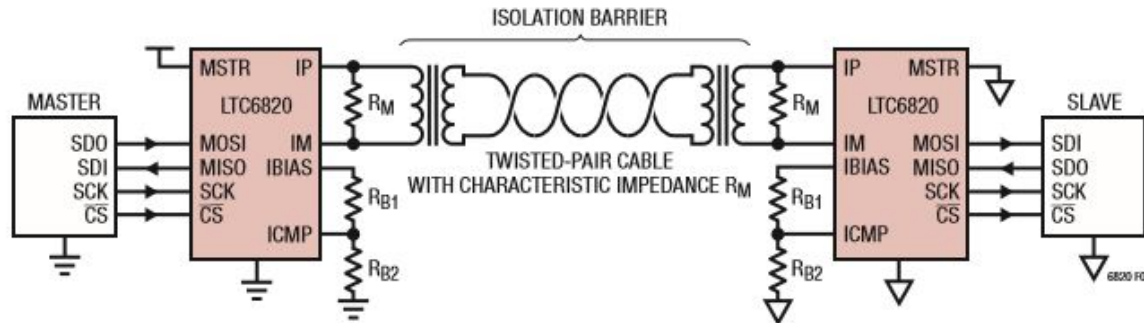
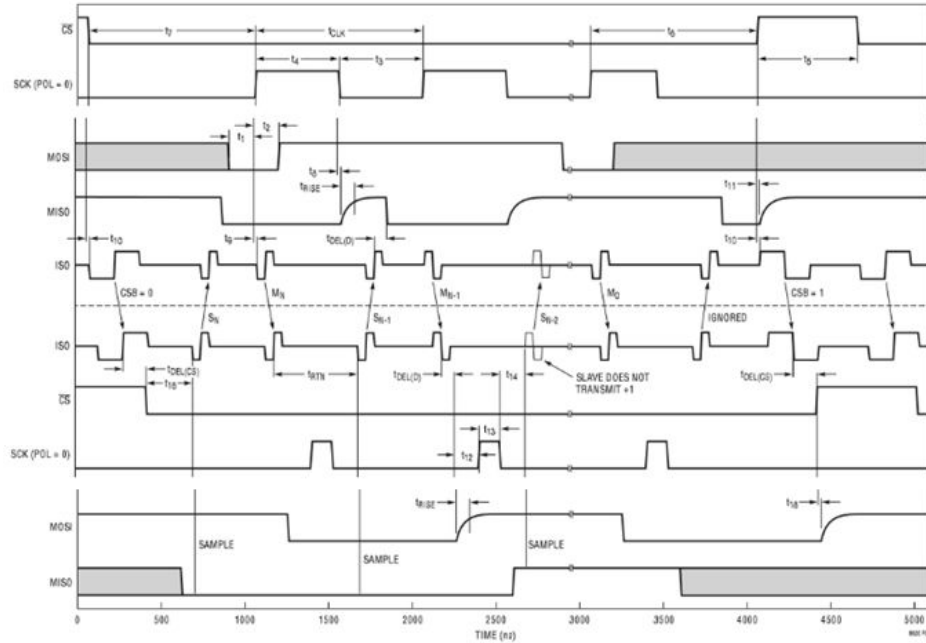
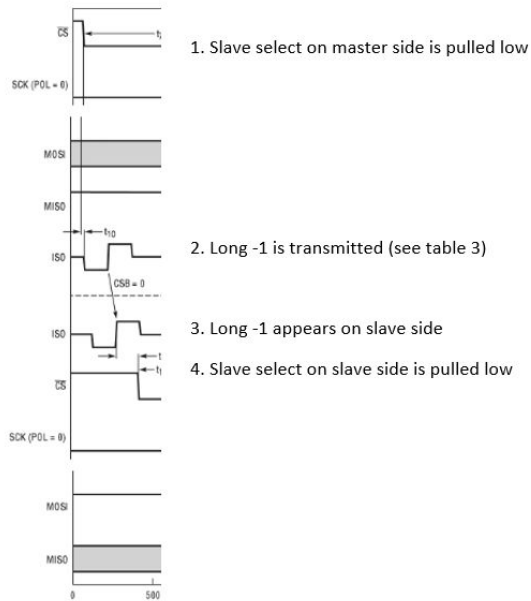


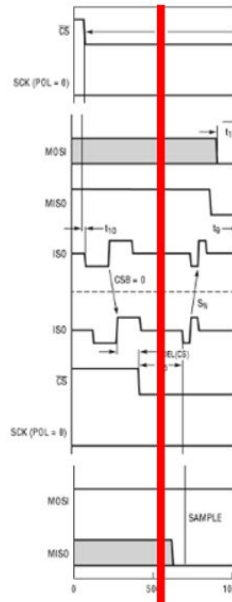
Figure from LT6820 data sheet

# isoSPI/LT6820 - timing diagram



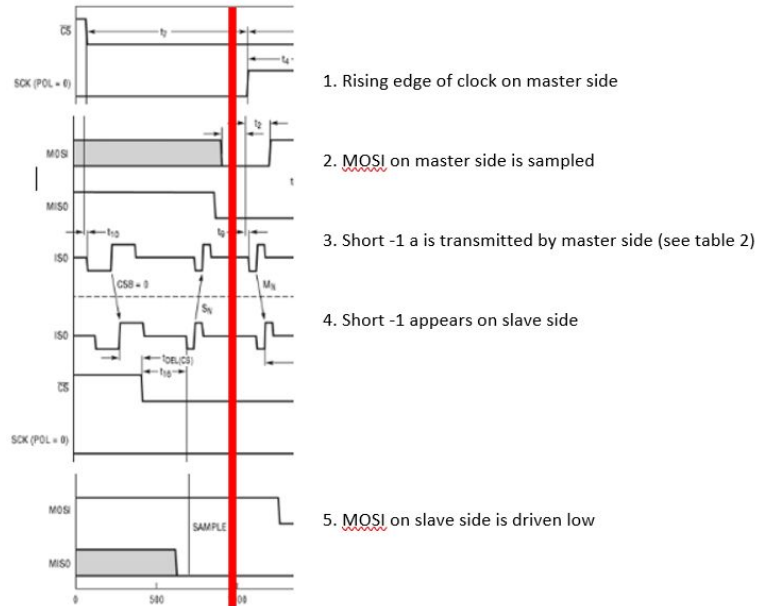


At the start of transmission (1) the slave select pin on the master side is driven low by the standard SPI master. (2) In response, the LT6820 on the master side transmits a long -1 (see table 3) through the isoSPI interface. (3) The long -1 appears on the slave side of the isoSPI interface. (4) In response, the slave select pin on the slave side is driven low by the LT6820.



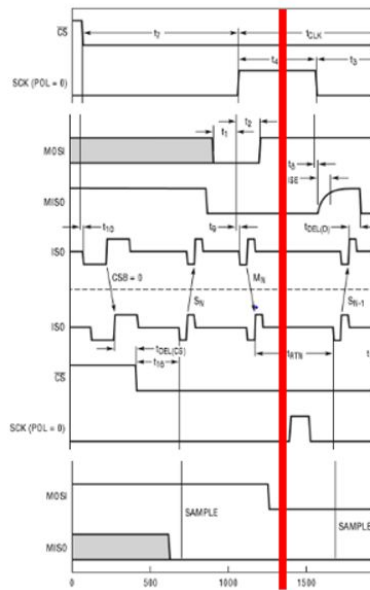
3. MISO on slave side is driven low
2. Short -1 appears on master side
2. Short -1 is transmitted by slave side (see table 3)
1. MISO on slave side is sampled low

The standard SPI slave is now operating and driving the MISO pin is low. The LT6820 on the slave side (1) samples the MISO pin on the slave side and (2), transmits the corresponding pulse through the isoSPI interface. (3) That pulse the appears on the master side. (4) In response the MISO pin on the master side is driven low by the LTC6820.



The standard SPI master has now started transmitting data by driving the MOSI pin low and (1) sending a clock pulse. (2) The LT6820 on the master side samples the MOSI pin on the rising clock edge. (3) The LT6820 on the master side sends the corresponding pulse and (4) that pulse appears on the slave side. (5) In response, the LT6820 drives the MOSI pin low on the slave side.



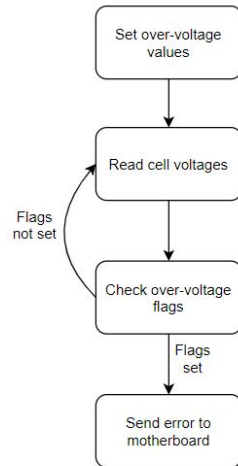


1. Falling edge of clock on master side
2. MISO on master side returns to high
6. MISO on master side is driven low
5. short -1 appears on master side
4. short -1 is transmitted by slave side
3. MISO on slave side is sampled low

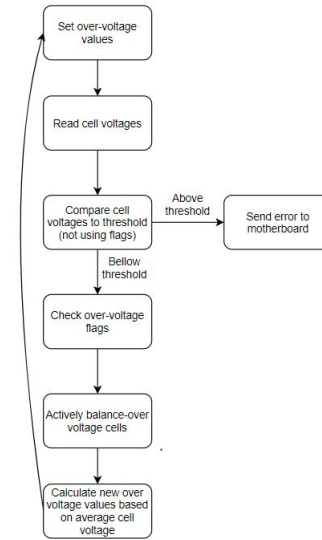
The standard SPI slave must respond, but first (1) the standard SPI master drives the clock low. (2) The LT6820 on the master side returns the MISO pin to its default state. Now the slave side responds. First, (3) the LT6820 on the slave side samples the MISO pin. (4) The LT6820 on the slave side transmits the corresponding pulse through the isoSPI interface which (5) appears on the master side. (6) The LT6820 on the master side drive the MISO pin low.

# Future Work on the AMS System

- Creating PCB for the LTC6812 (instead of using the demo boards) to allow for active cell balancing
- Re-configuring code for active cell balancing



Old code



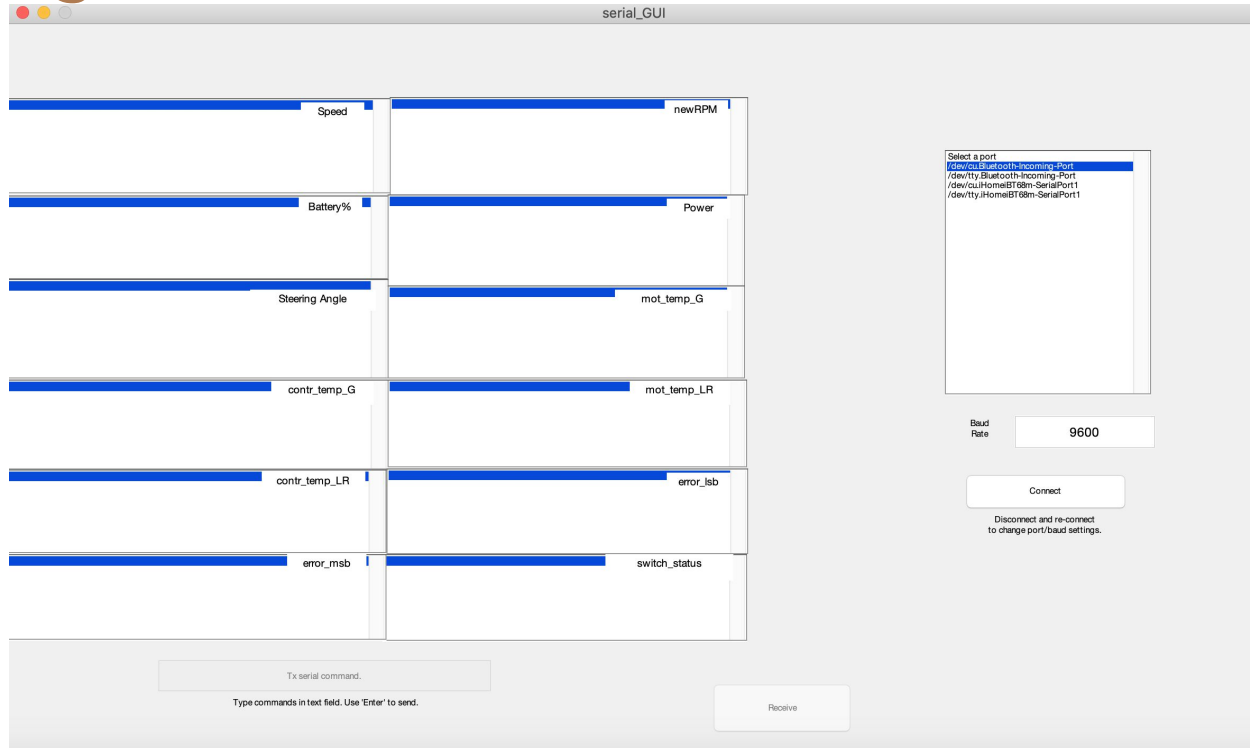
New code

# Data Log and RF GUI

```
43000,3,0,7,102
43500,223,50,557,24
44000,43,9,107,613
44500,241,54,602,451
45000,218,49,545,662
45500,151,34,377,856
46000,108,24,270,120
46500,136,30,340,433
47000,137,31,342,353
47500,205,46,512,349
48000,108,24,270,348
48500,154,35,385,702
49000,83,18,207,551
49500,245,55,612,121
50000,211,48,527,954
50500,238,54,595,150
51000,147,33,367,85
51500,135,30,337,777
52000,134,30,335,754
52500,156,35,390,400
53000,48,10,120,835
53500,95,21,237,101
54000,33,7,82,317
54500,186,42,465,30
```

- Legacy RF displayed data using only Putty to spit out CSV data received
- Designed an interactive GUI to present the data in a more meaningful manner
- Updates to the GUI would involve a killtime so that data is updated in real time but that could not be tested due to not being able to access the CAR post spring-break

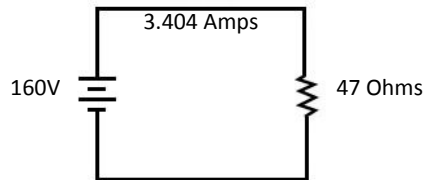
# Data Log and RF GUI



# Capacitor Charge/Discharge

## Charging

- Using a variable Op-Amp, a transformer, a few multimeters, and some clamps hooked onto the capacitor series, we were able to fully charge the system in around 35-40 minutes
- We used the wall outlet of 120V, and did calculations and varied the load so that only  $15 \pm 5$  mA would pass through the circuit
- We borrowed this equipment from the formula hybrid lab, and had help from the Formula Hybrid TA, Eduardo Mello



47 Ohm, 1000W Power Resistor  
2.362" Dia x 11.811" L

## Discharging

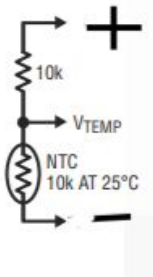
- Assuming total Power from 160 Volts ( $2.7V \times 60\text{Caps}$ ) and 50F Capacitors =  $CV^2/2 = 640$  kJ
- Using a 47 Ohm resistor in a series circuit with 160 Volts, we find current to be 3.404 Amps
- Power going into resistor given by  $P = IV = 3.404 \times 160 = 544.680$  Watts = 544 J/s
- Assuming capacitor linear discharge, average discharge rate given as totals joules discharged/volts =  $(P \times V/2)/(V) = 272$  J/s
- In theory,  $640000$  [J] /  $272$  [J/s] =  $2352$  [s] = 39.22 minutes for a full discharge
- In practice, discharge time is as slightly faster: Discharging from 150V to 55 V (95V difference) took around 23 minutes

# Thermistor Implementation

Two types (PTC and NTC). **Group has NTC**

**NTC Thermistor**- resistors with negative temperature coefficient (resistance decreases with increasing temperature).

30 can be used as temperature control for all 60 ultracapacitor using its temperature-resistance equation and graph. Also be used to detect over-voltage.



Temp Range (°C)	Ratio	Beta
0 to 50	9.08	3895
0 to 70	18.64	3917
25 to 50	2.78	3933
25 to 85	9.30	3969
25 to 100	14.64	3981
25 to 125	29.05	3999
37.8 to 104.4	9.67	4000

To calculate  $R_t/R_{25}$  at temperatures other than those listed in the table, use the following equation:  
$$R_t/R_{25} = \exp\left\{A + B/T + C/T^2 + D/T^3\right\}$$
where T = temperature in K  
where K = °C + 273.15

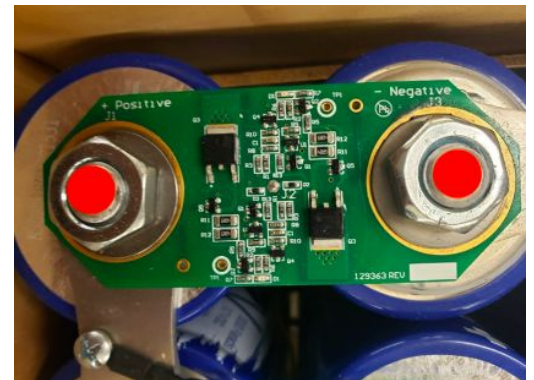
## Requirements

Isolation of ultracaps with use of the accumulator containers.

Requirement to monitor 10% of ultracaps but group wants to monitor all for safety purposes.

## Implementation

Connection of thermistors between ultracapacitors.



# Future Work on the Project

- Active cell balancing
- PCB instead of demo boards
- PID controls
- External Thermistor implementation circuitry
- One CAN module for each motor controller
- Hydraulic Braking (in the far future)

# Questions + Comments