* **Competition details**
  + 0 to 60mph
  + Half a mile furthest point for UMS in Tooele test track.
* **Every device**
  + 12v power input (this is not a well regulated line voltage can vary significantly)
  + low voltage is straight off a 12v coming off a battery (separate from motor voltage)
  + 12v is standard between boards(?).
* **Remote live telemetry**. Transmit status parameters from car to paddock.
  + Wireless Tx/Rx transmission and reception
    - Should connect to all of the boards
      * Likely have a central board the connects to all others and then that can split to either recording or transmitting
    - You can choose connection method
      * Ie. CAN, I2C, UART
    - Should have external antenna or be small enough to mount on the outside of car because the body of the car is electrically conductive
* **Data logging**
  + Not allowed during the race, so only during practice
  + Parameters should be recorded over a period of time (at least as long as a race)
  + Visualize the data
  + We can choose where and how we store. They can do data analysis after the fact (not planning on analyzing data while the car is running).
  + Do sampling ~10x-20x the dynamics of the system,

**Highlighted sensors to be chosen by embedded systems class.**

**Microcontroller - Teensy.** Sounds like we will use as many as we need to interface with everything.

* Multiple are planned
* 1 for accelerator
* 1 for data logging (central location for gathering and storing data). Could even be a whole raspberry pi or something.
* **Parameters**
  + Trouble Codes - Read current status of each system as described below and log/transmit the data, as well as display on dashboard. Overall section name. Everything below is its own board
    - Shutdown circuit open
      * Read the status of the shutdown circuit
      * 9x 12v high, 0v low digital inputs, must be high impedance measurements (cannot affect signal!).
      * Daughter board (footprint not yet determined)
    - Battery Management System (BMS) (AKA AMS)
      * List of available data: [orionbms\_obd2\_pids.pdf](https://www.orionbms.com/downloads/misc/orionbms_obd2_pids.pdf)
    - Motor Controller
      * Read CAN messages (controller not chosen yet will update)
    - Pedal Brake System Plausibility Device (BSPD)
      * Plausibility device detects whether signal is real/coming from a defective sensor
      * 5x 12v high, 0v low digital inputs, must be high impedance measurements (cannot affect signal!).
      * 1x Analog input for current sensor (sensor not yet chosen)
  + They were choosing sensors for the pedals, but now we are.
    - Accelerator
      * 2 potentiometers. 1 for APPS (accelerator p? plausibility sensor), and 1 for Teensy
        + BOTH pots interface with accelerator pedal. Each pot has a different DC offset per rules (Check APPS-BSE in Altium). Both outputs are sent to Teensy, then checked for agreement.
      * Teensy sends CAN messages
    - Brake
      * 1 pot (has BSPD, mentioned above) and 1 switch for if the brake goes too far (has an overtravel switch that triggers to override and stop the system)
    - APPS and BSPD are both Analog (by rules)
      * APP sensors *are* analog, the check logic is not (Teensy)
    - Pressing both pedals all the way means shutdown mode.
      * Shutdown if brakes are active *at all* and APPS is >25% travel. Latch in shutdown condition and unlatch if APPS reduces to <5% travel.
  + Vehicle Status
    - Pedal position
      * We are using CAN to communicate accelerator/brake pedal position with the motor controller, and need to log both.
    - Battery status
      * This is the non-error messages coming from the BMS (state of charge, state of health, voltage, current, temperature)
      * CAN Protocol used. See OBD II PIDs above
    - Inertial Monitoring Unit (IMU)
      * Accurately measure the acceleration/speed/position of the car
      * 6+ axes preferred
      * You can choose which sensor to use
      * They have one in their possession!
    - Global Positioning System (GPS)
      * Give rough position of the car on the track throughout the race
      * You can choose which device to use
      * One idea was to map position and data for racing analysis.
      * Maybe look into ground tracking systems instead of satellites, but that might be too expensive and just use GPS.
    - Tire Temperature and Pressure Monitoring System (TTPMS)
      * You choose a commercially available one and figure out how to interface with
    - Wheel Speed Sensor
      * You can choose the device and figure out how to interface with it
      * Hoosier 18.0”X6.0”-10” tires, 10.0” x 7.0” rim
      * Perhaps some sort of ~rotary encoder on drive shafts
    - Steering Column position
      * Measure the angular position of the steering wheel (100 degree range)
      * Rotary encoder?
    - Damper Potentiometer
      * Measure the suspension travel
      * Essentially a linear variable displacement transducer that is mounted parallel to the coilover (see photo)
      * Range of measurement will be somewhere between 35 and 85mm **\*this is tentative as the damper travel has yet to be determined\***
      * Data resolution of 1 mm would be nice, resolution of up to 2.5mm would still work
      * Accuracy of +-1 mm would be nice, up to +-2.5mm would still work
      * A fairly quick response time is going to be crucial, the car will be spending a very small amount of time in a given state, and jumps between states happen quick
      * Packaging constraints aren’t much of an issue, coilovers will most-likely be mounted on the outer body of the car
      * Linear encoder?
    - Brake Temperature sensor
      * Measure the temperature of the brake rotor
      * COTS

**Communication with Team**

* The best way to communicate with the team is via Discord
* <https://discord.gg/QBR8qBYWcj>
* **Github**: <https://github.com/FormulaU>
* **Google drive:** <https://drive.google.com/drive/folders/120djxZDYhjSLKA9M724plU-OEjL_StN4>
* **Dashboard Interface**
  + Create a digital dashboard interface that uses the parameters to give important information to the driver
  + Likely something like a small lcd screen
  + No info needed outside of the screen, we will handle the physical dashboard info
  + We can work with you all on the dashboard layout
* **Suspension Spec Sheet**

[2023-24 Evolving Suspension Spec Sheet](https://docs.google.com/spreadsheets/d/19O4KZlKmsuLw3heawv7xO9NfNTk5IIULwr3InsUSnPA/edit?usp=sharing)

- Reach out to AlexC on discord if there’s anything you need added to the spec sheet

**Priorities**

1. [Teensy 4.1 boards](https://www.pjrc.com/store/teensy41.html) for Accelerator and Brake pedals
2. Daughter Card (Jared put this here, seems like this is a task in itself, think it just means getting a standardized body that has known inputs/outputs for his circuit designers)
3. Trouble codes - see what caused the car to shut down, and store them somewhere
4. Low priority - log data that’s interesting
   1. GPS
   2. Wheel speed/pressure/temperature
   3. Steering column position
   4. Damper potentiometer
   5. Anti roll sensors

<https://www.orionbms.com/products/orion-bms-standard/>

The motor only communicates directly with the motor controller. The motor controller will provide the motor temperature, position, speed, etc. through CAN. Our current motor controller is a Cascadia Motion PM100DXR. More data on it, including the CAN protocol, can be found at

[This link](http://www.cascadiamotion.com/productlist/14-inverters/8-pm-inverters/1-pm100)

Carlos, Henry, Jared, Will, Wilson, Fredi, Dmitry, Cristian

Action Items:

12V regulator down to 5V to it’s usable for our devices

Vin = 9-18V, Iout 3A, Price: 16..44

<https://www.digikey.com/en/products/detail/mean-well-usa-inc./SD-15A-5/7706481>

Vin = 9.2-18V, Iout = 5A, Price: $27.73

<https://www.digikey.com/en/products/detail/mean-well-usa-inc/SD-25A-5/7706508>

Vin = 9-18V, Iout = 5A, Price: $17.22 → No eclosure

<https://www.digikey.com/en/products/detail/mean-well-usa-inc/PSD-30A-5/7705890>

Vin = 9-36V, Iout = 4A, Price: $49.5

<https://www.digikey.com/en/products/detail/xp-power/DTJ2024S05/10440624>

**-Wilson, Ryan**

Block diagram of system for us to go off of

* **Jared, Will, Sam**

Data storage device/ central computer device

* **Emma, Brittany, Cristian**

APPS/BSPD/BSE boards error code reading:

* **Carlos, Daniel**

Select devices for sensors

-Rotary encoder for steering column

-Temperature sensor for brake temp

-Pressure sensor for tire pressure

**-Fredi, Misael**

Need documentation on CAN protocol that the team uses to send data

-Decipher BMS CAN

**-Alex, Henry**

<https://www.orionbms.com/manuals/utility_o2/>   
 This is a list of all possible CAN messages, generating a list of which messages seem useful would be good

-Decipher Motor Controller CAN

- **Cody, Dmitry**

CAN protocol v6.2 sheet

[**https://app.box.com/s/vf9259qlaadhzxqiqrt5cco8xpsn84hk/file/27334613044**](https://app.box.com/s/vf9259qlaadhzxqiqrt5cco8xpsn84hk/file/27334613044)

**Wednesday Questions:**

How long does a trial last in minutes? Hours? How long do you spend on the track training drivers?

* Limited by battery, theoretically ~35 minutes per day (20 Minutes Min)

What throughput do we need to hit? How many messages per second max?

* 20-min to 35 minutes worth of data (in a day) Assume 1 hour of trial data

What error logs do you want to see? Know that the mission critical stuff (battery, motor controller) have OBD for you to check.

* ???

Do we need error logs or do you just want a “Check engine” light to come on?

* ????

Are the error logs going to be just as a “what happened” if an emergency shutdown occurs. Can we just buffer the last minute of error logs and save to file if the emergency shutdown signal is seen?

* ???

Does the emergency shutdown kill power to just the motor controller or the entire car? Will we have power from the 12V in an emergency shutdown?

* The emergency shutdown will shut down the entire car
* We will have 12 V in an emergency shutdown
* There are 3 emergency shutdown
  + 12 and 9 Volt lines
  + Physical headers with pins (high is on, low is off)

What tire pressure are you running? Is the stem removable? There are two types of Pressure monitors, in wheel and stem caps, do you want a commercial solution, i.e. a dashboard mounted/cell phone connected system? If not we need to discuss if we can hack the signals being sent by commercial gear.

* Can be ignored for now, since all tire sensors found so far are commercial and hard to interface from our end

How is the steering wheel set up? I remember it uses a rack and pinion for steering, so that means we’ll get multiple revolutions to move the wheels from left to right completely. Can we put a rotary encoder directly on the end or do we need to couple it to an encoder running parallel via gears?

* 100 degrees total rotation. 0 degrees is all the way left, and 100 is all the way right.

Are we programming motor controller gain from pedal and brake positions?

* ???

I propose we focus on gathering data from sensors such as tire pressure(maybe), rpm of crankshaft, battery state of charge, potentially calculating rate of consumption and battery life, and making a check engine light signal and passing these to a dashboard that we let FormUla one design, storing the error codes from BMS and Motor Controller & GPS data to an onboard device/available for download via Ethernet, and finally creating the teensy pedal controller (maybe?)

Extra notes for 11/15/23

* No need for hall effect sensor, motor controller board / motor has one built in, Can get hall data from it (RPM from CAN message)
* BSPD not connected to teensy and we will be interfacing with shutdown circuit
* Might be able to get companies to donate stuff to us
* Extra options for boards (ESP32 surface mount)
* Need to get doc listing items they already have
* Potential to use pins on PI to CAN connection \*\*\*NEED TO LOOK INTO THAT\*\*\*
* Wants us to be more focused on Data monitoring
* Plan to pull out SSD and manually pull out data and clear it\*

Link to Sensor List and BOM:

<https://docs.google.com/spreadsheets/d/1EQL0YHKri98wX61jzjHVJqKFCQWMRYFGmXEUzqYPyl0/edit#gid=1486489898>

**Action item ideas:**

Code Pi

Data logging

dashboard interface

Connect CAN Bus to Pi(grab data and buffer it. We can use the CAN transceivers )

GPS - Connects

Teensy Pedal Control (analog signals to CAN)

Get interfaces with these systems

Motor Controller Board

* Will, Jared, Sam, Dmitri

IMU - Wilson, Carlos, Daniel, Zander

BMS - Henry, Alex, Emma, Cody

* 8 byte CAN messages at 125 Hz (8ms).

Pi - Cristian, Brittany, Misael, Ryan, Fredi

Throughput on devices pi and discovery board

We can upload a RTOS on PI instead of linux

Power consumption

How much data from each group are we getting/ throughput on data storage

Montag Nov. 20: