

LAFAYETTE COLLEGE
DIVISION OF ELECTRICAL AND COMPUTER ENGINEERING
LFEV- Y5- 2017
PRELIMINARY DESIGN REVIEW
MONDAY, JANUARY 30TH, 2017

Table of Contents

1. Introduction	5
2. Summary of individual teams and Goals/Requirements	6
3. Work Breakdown Structure	7
4. Overall System Diagram	8
5. Individual Subsystem Contribution	9
5.1. GLV [Grounded Low Voltage]	9
5.1.1. Current State of Affairs	9
5.1.2. Goals	9
5.1.3. System Level Diagrams	10
5.1.4. Task Delegation	12
5.1.5. Requirements Checklist	14
5.1.6. Work Breakdown Structure	15
5.1.7. Individual Acceptance Test Strategy	15
5.2. JGB [John Gehrig Board (CAN bus DAQ board)]	16
5.2.1. Current State of Affairs	16
5.2.2. Goals	16
5.2.3. System Level Diagrams	17
5.2.4. Task Delegation	17
5.2.5. Work Breakdown Structure	18
5.2.6. Individual Acceptance Test Strategy	19
5.3. Interconnect, Cabling, ICD [Interconnectivity Document]	19
5.3.1. Current State of Affairs	19
5.3.2. Goals	19
5.3.3. Task Delegation	19
5.3.4. Requirements Checklist	20
5.3.5. Work Breakdown Structure	20
5.3.6. Individual Acceptance Test Strategy	21
5.4. VSCADA [Vehicle Supervisory Control and Data Acquisition]	21
5.4.1. Current State of Affairs	21
5.4.2. Goals	21
5.4.3. System Level Diagrams	22
5.4.4. Task Delegation	23
5.4.5. Requirements Checklist	23
5.4.6. Work Breakdown Structure	26

5.4.7. Individual Acceptance Test Strategy	26
5.5. Cell App	27
5.5.1. Current State of Affairs	27
5.5.2. Goals	27
5.5.3. System Level Diagrams	28
5.5.4. Task Delegation	29
5.5.5. Requirements Checklist	29
5.5.6. Work Breakdown Structure	29
5.5.7. Individual Acceptance Test Strategy	30
5.6. TSV [Tractive Systems Voltage]	30
5.6.1. Current State of Affairs	30
5.6.2. Goals	31
5.6.3. System Level Diagrams	32
5.6.4. Task Delegation	33
5.6.5. Requirements Checklist	34
5.6.6. Work Breakdown Structure	35
5.6.7. Individual Acceptance Test Strategy	36
5.7. Controller Cooling System	36
5.7.1. Current State of Affairs	36
5.7.2. Goals	36
5.7.3. System Level Diagrams	37
5.7.4. Task Delegation	38
5.7.5. Requirements Checklist	39
5.7.6. Work Breakdown Structure	40
5.7.5. Individual Acceptance Test Strategy	40
5.8. DYNO [Dynamometer]	41
5.8.1. Current State of Affairs	41
5.8.2. Goals	41
5.8.3. System Level Diagrams	41
5.8.4. Task Delegation	42
5.8.5. Requirements Checklist	42
5.8.6. Work Breakdown Structure	43
5.8.7. Individual Acceptance Test Strategy	44
5.9. Physics Modeling	44
5.9.1. Current State of Affairs	44
5.9.2. Goals	44
5.9.3. System Level Diagrams	45

5.9.4. Task Delegation	46
5.9.5 Work Breakdown Structure	47
5.10. Cruise Control	47
5.10.1. Current State of Affairs	47
5.10.2. Goals	48
5.10.3. Task Delegation	48
5.10.4. Individual Acceptance Test Strategy	49
5.10.5 Work Breakdown Structure	50
5.11. TSI [Tractive System Interface]	50
5.11.1. Current State of Affairs	50
5.11.2. Goals	50
5.11.3. System Level Diagrams	52
5.11.4. Task Delegation	54
5.11.5. Requirements Checklist	55
5.11.6. Work Breakdown Structure	56
5.11.7. Individual Acceptance Test Strategy	56
6. Overall Acceptance Test Strategy	56
7. Maintainability Plan	57
8. Cost Analysis	57

1. Introduction

Lafayette College's division of Electrical and Computer Engineering will be continuing its interdisciplinary efforts compliant with all rules and regulations of the Formula Hybrid Electric Vehicle Competition. We will be working closely with the Mechanical Engineering division as they develop, test, and implement a fully functional overall structure of this vehicle. This will be a class effort as we have 24 engineers working collaboratively to create, further existing efforts, test, and implement any and all electrical components that may be required in this vehicle.

In addition to working towards a fully functional vehicle, we will also be delivering [as per requirements D000-D014]:

- *Critical Design Review (CDR)* which will analyze, review, and update all PDR materials,
- *individual user manuals* which will catalogue the physical system, UI screens, and indicators as well as document safe operation, "getting started" information, FAQ, troubleshooting, and maintenance
- a *maintenance manual* that records all high level aspects with advanced maintenance/calibration techniques and annotated schematics
- an *Acceptance Test Plan (ATP)* that documents how the system as a whole should be tested and how the calibration/error analysis should be performed, all the while trying to prove compliance to Formula EV rules and Lafayette College deliverables
- an *Acceptance Test Report (ATR)* that documents the execution of ATP with annotated evidence of the results
- a *Project website*, which will be updated on a biweekly basis that catalogues the weekly status of the project
- a *conference paper, presentation, and video* that highlights individual team efforts.
- a *project poster*, that summarizes the project at a high level and diagrams of individual systems into a single document.

We collectively will establish a mutual performance monitoring (MPM) technique such that each team member will have an understanding of the current progress, goals, and hindrances of other teams via weekly project status letters (PSL) compliant with D014. Each team member will catalogue their individual contributions in an Individual Progress Report (IPR) compliant with D014 in a weekly fashion no later every Friday at 5pm EST. These then will be reviewed and compiled and presented to the team as a whole every Monday. Furthermore, we will be displaying updated versions of the work breakdown structure, which will be displayed in AEC 400, as we complete outcomes as well as continuously updating the website.

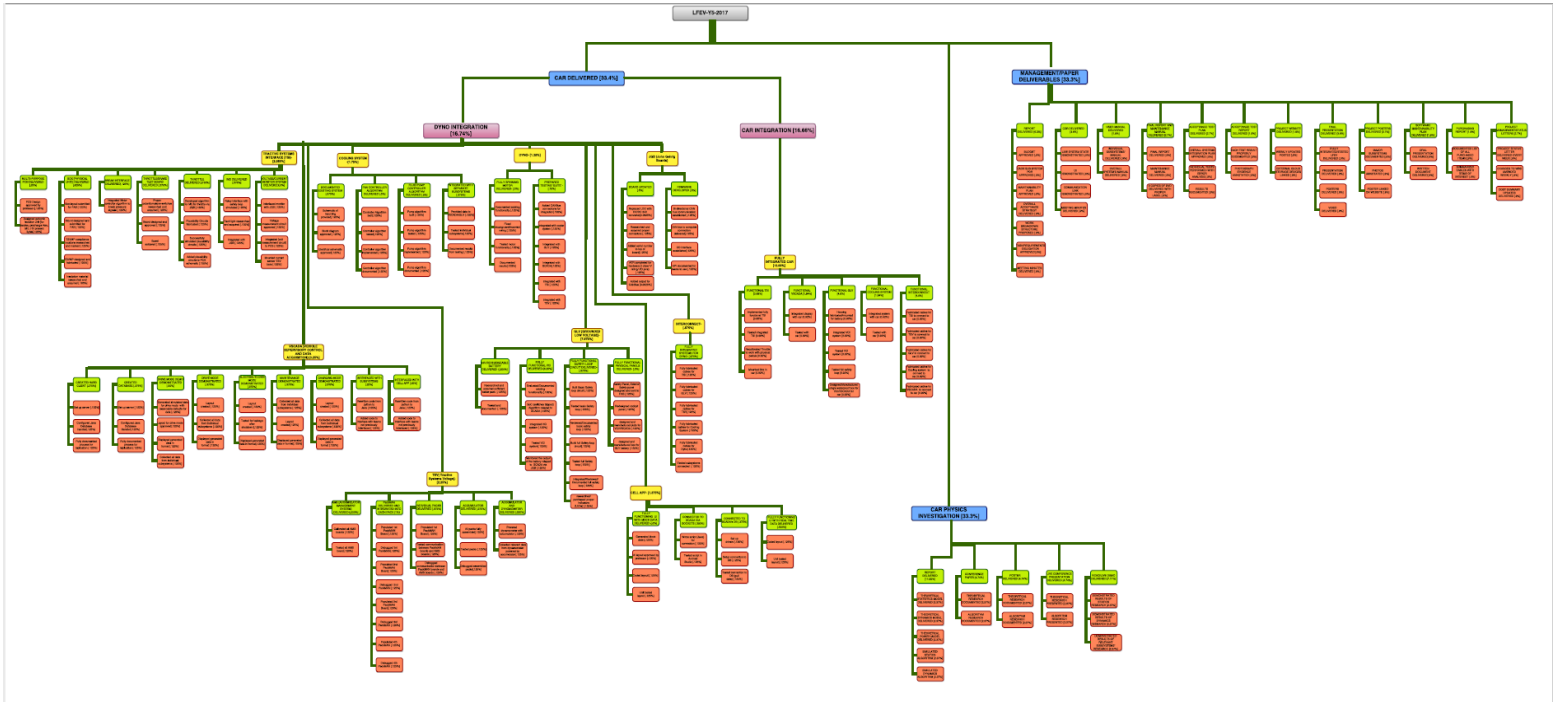
2. Summary of individual teams and Goals/Requirements

The team breakdown and overall goal are as follows:

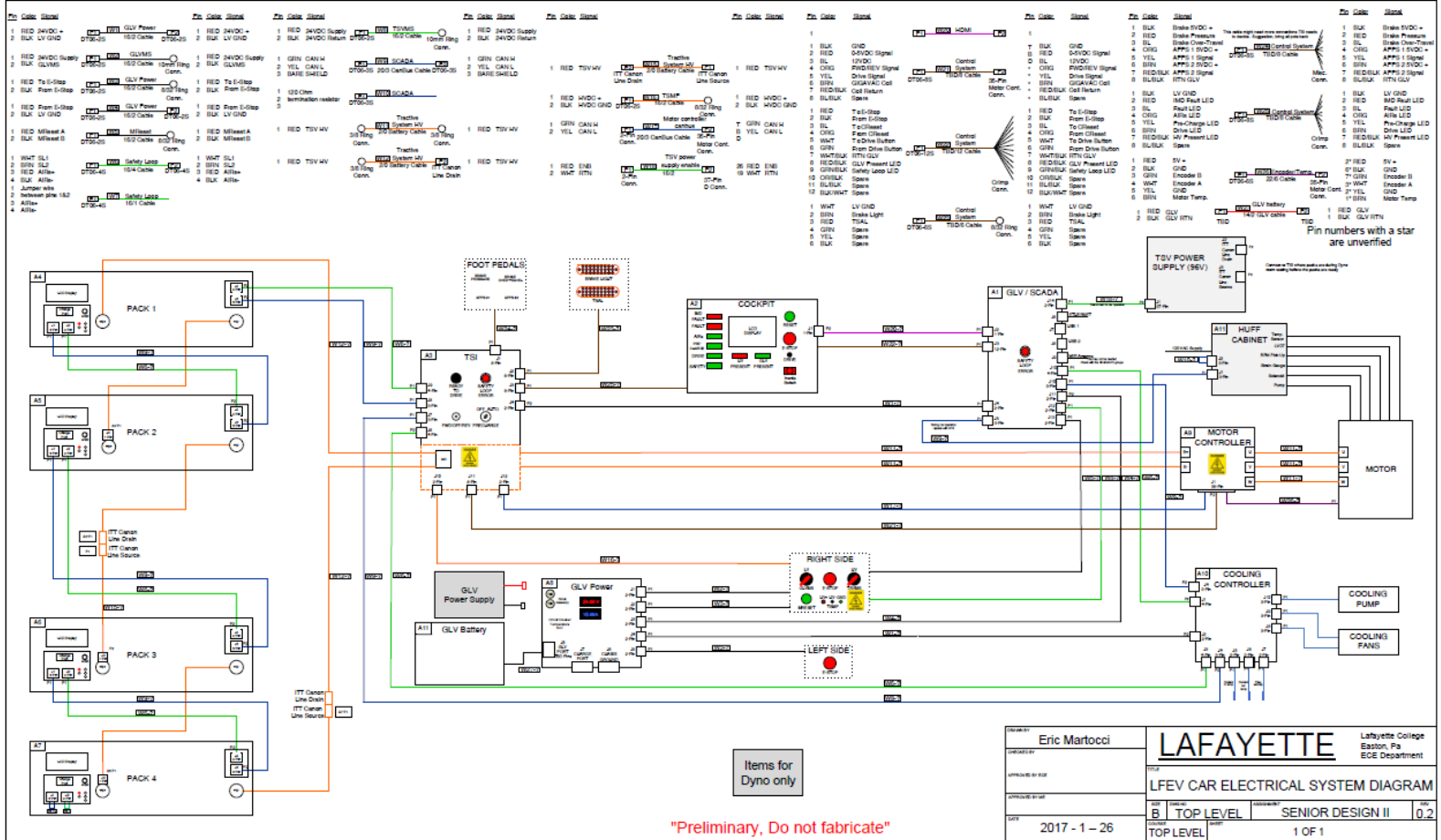
1. TSI → Jack Plumb, Adam Ness, Christer Hoeflinger
 - a. Provide connection and safety for the power supply of the motor.
2. GLV → Chris Bennett, Joe Sluke, Kyle Phillips, Marty Townley
 - a. Provide 24V to all the other systems, excluding the motor
3. VSCADA → Craig Lombardo, Austin Wiles
 - a. Provide data acquisition capability, as well as diagnostics.
4. Cell App → Kemal Dilsiz, Raji Birru
 - a. Be able to wirelessly interface with SCADA system.
5. Controller Cooling System → Xingyuan Guo, Yi Han
 - a. Using 24V, while providing constant feedback to SCADA, cool the motor.
6. Interconnect, Cabling, ICD → Larisa Chiesa, Jeremy Port
 - a. cabling: provide safe and consistent connections between subsystems.
 - b. ICD: provide documentation for both hardware and software and interfaces.
7. Dyno → Richard Diego, Eric Martocci
 - a. Build and prepare Dyno room for testing for all sub-systems
8. TSV Packs → Emilie Grybos, Meredith Guro
 - a. Provide 96V to the motor, as well as diagnostic information, and update its charging algorithms.
9. Physics Modeling, Cruise Control → Zainab Hussein
 - a. Helping TSV team understand relevant research, as well as physical modeling of the car, and developing a cruise control algorithm.
10. JGB → Gregory Flynn, Martin Townley
 - a. The JGB boards are interfaced to each subsystem that allow communication between all subsystems
11. Management → Gregory Flynn, Martin Townley, J. Mahati Hari, Stephen Beggs, and Rachel Szteinberg
 - a. Oversee all deliverables, compile all documents, assists subsystem teams, accounting, system engineering and ensures communication across all aspects of project
 - b. Marketing and Project Website → Stephan Beegs
 - c. System Engineering → Greg Flynn and Marty Townley
 - d. Project Management → J. Mahati Hari
 - e. Operations / Accounting → Rachel Szteinberg

3. Work Breakdown Structure

Below is the overall WBS for the completed project. Each individual subsystem has their WBS in section 4, where each task is able to be clearly seen.



4. Overall System Diagram



5. Individual Subsystem Contribution

5.1. GLV [Grounded Low Voltage]

5.1.1. Current State of Affairs

In the past few years, work has been put into the safety loop, VUI, and VCI design. Very limited work has been put into GLV Power.

The safety loop has well been designed and integrated; however, testing of the circuits has not yet been done. We have 4 wire DT-04 connectors that we can use for interfacing the external safety loop with the GLV box, relays, and cables we would like to reuse.

In regards to VUI, two panels have been manufactured for interfacing with the existing systems. They are both for use in the dyno-room and not on the car. The first panel, labeled *Cockpit Panel*, is used for representing the VUI inside of the cockpit, including the BRB, reset, ready-to-drive switch, indicator lights and a LCD screen to display SCADA info. The second, labeled *Safety Panel*, is a panel meant to represent the VUI on the exterior of the car and has a GLV master switch, a HV(TSV) master switch, and two BRBs.

The VCI system has been developed to relay information about the safety loop board to the VSCADA computer via the CAN bus. We do not fully understand how the existing system functions, but according to the notes from last year, not everything was finished. We have JGB boards we can use to hopefully achieve this interfacing.

5.1.2. Goals

We first need a 24V rechargeable battery that must to be researched and purchased. In conjunction with this, we need to design a way to monitor the SOC and output of the battery and possibly have this information relayed to VSCADA.

Our first priority is to build a basic safety loop circuit to turn on the 100 V power supply in the Dyno room and the AIRs on the pack. The only safety switch features will be the PackMan switches, the reset button and the e-stop button. A diagram of this basic safety loop circuit is shown in Figure 1.

After we complete building this circuit and test its functionality, we will begin to integrate the rest of the safety loop features and review the existing design, including powering up of all GLV powered systems i.e, the cooling system. An updated diagram will be created for the safety loop and for the whole GLV system showing GLV interconnects with the VSCADA system.

For VCI, we may possibly have to work on developing hardware to allow the SCADA computer to interface with the CAN bus. We may have to work with some JGB boards so that we can relay GLV data to the CAN bus/SCADA computer. For VUI, we need to make sure we provide the appropriate user interface in the car and in the dyno room for the SCADA system. Parts of this can be recycled from previous years; however, we need to consult with the SCADA team to solidify this.

Lastly, we will need to redesign and remanufactured the two existing panels for use in dyno room. We will rename the *Safety Panel* as the *Exterior Safety Panel*. In addition to this we will have manufacture corresponding panels for the actual vehicle. Lastly, we will have to design and manufacture a single, water proof box for the VCI, safety loop and VSCADA systems that can interface with either the dyno room panels or panels on the car.

5.1.3. System Level Diagrams

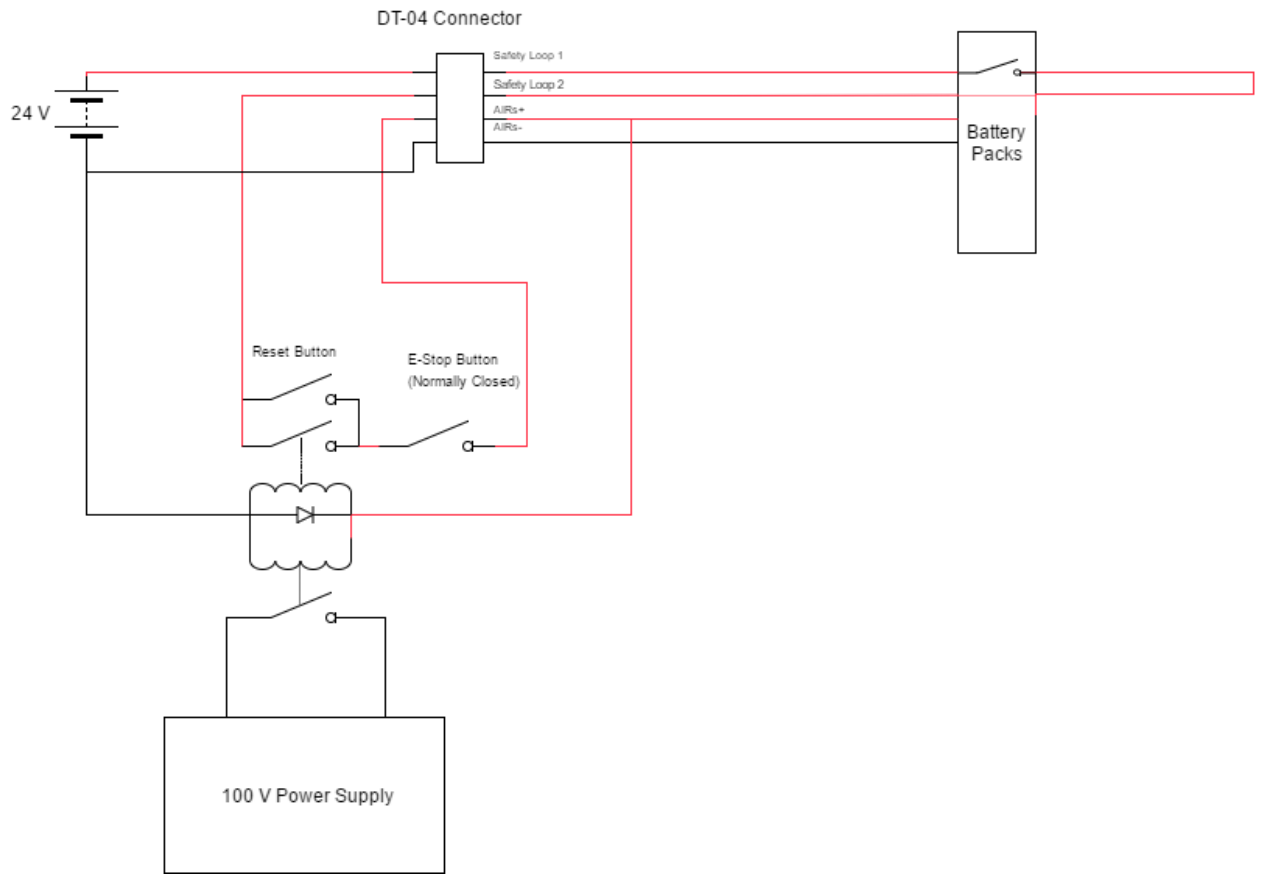


Figure 1. Diagram of Basic Safety Loop Circuit.

Safety Loop Component	Description
Fuse	A basic fuse limiting current
GLVMS	Grounded Low Voltage master switch, not accessible by driver
BRBLS	Big Red Button Left Side, not accessible by driver
BRBRS	Big Red Button Right Side, not accessible by driver
SCADA Relay	Allows SCADA computer to trigger fault
Cooling Switch	Allows cooling system to trigger fault
IMD Switch	Insulation Monitoring Device, closed if HV and GLV are galvanically isolated
PACKMAN Fault Control	Allows PACKMAN board on any pack to trigger a fault
TSVMS	Tractive System Voltage master switch, not accessible by driver
MRESET	Master Reset, resets AIRs
BRB (dash)	Big Red Button on dashboard, accessible to driver
CPR	Collision Protection Reset, inertia switch, opened by a collision. accessible to driver
TSEL	Tractive System Energized Light, lit when AIRs are energized
BOTB	Tractive system Brake over travel Button

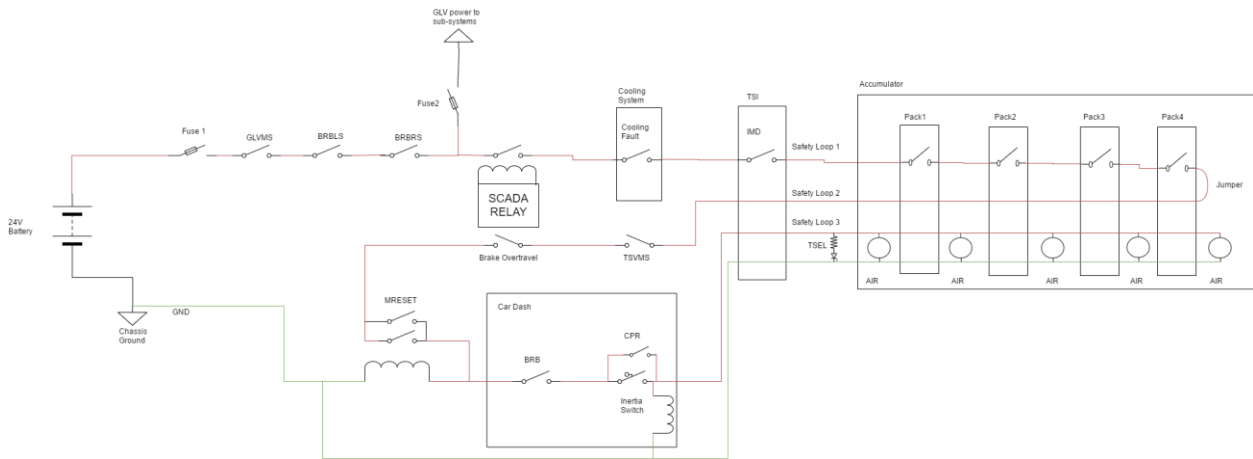


Figure 2. Full Safety Loop

		Controlled Systems			
		Engine Starter (High Current)	GLV Supply to: Instrumentation Data acquisition Computers Telemetry Etc.	I.C. Engine Ignition Fuel pumps Starter solenoid Etc.	AIRs (TS Voltage)
Shutdown Sources	TSMS				OFF
	Cockpit BRB			OFF	OFF
	AMS			OFF	OFF
	IMD			OFF	OFF
	Brake Over-travel			OFF	OFF
	Side-mounted BRBs		OFF	OFF	OFF
	GLVMS	OFF	OFF	OFF	OFF

Figure 3. Priority of Shutdown Sources

5.1.4. Task Delegation

Chris → Battery research. Purchasing, Testing, Housing

Joe → Basic Safety Loop, Full Safety Loop Design and integration

Kyle → Existing Safety Loop Review, Full Safety Loop Design and integration

Marty → Interfacing with CAN bus and VSCADA, relaying state of safety loop/battery

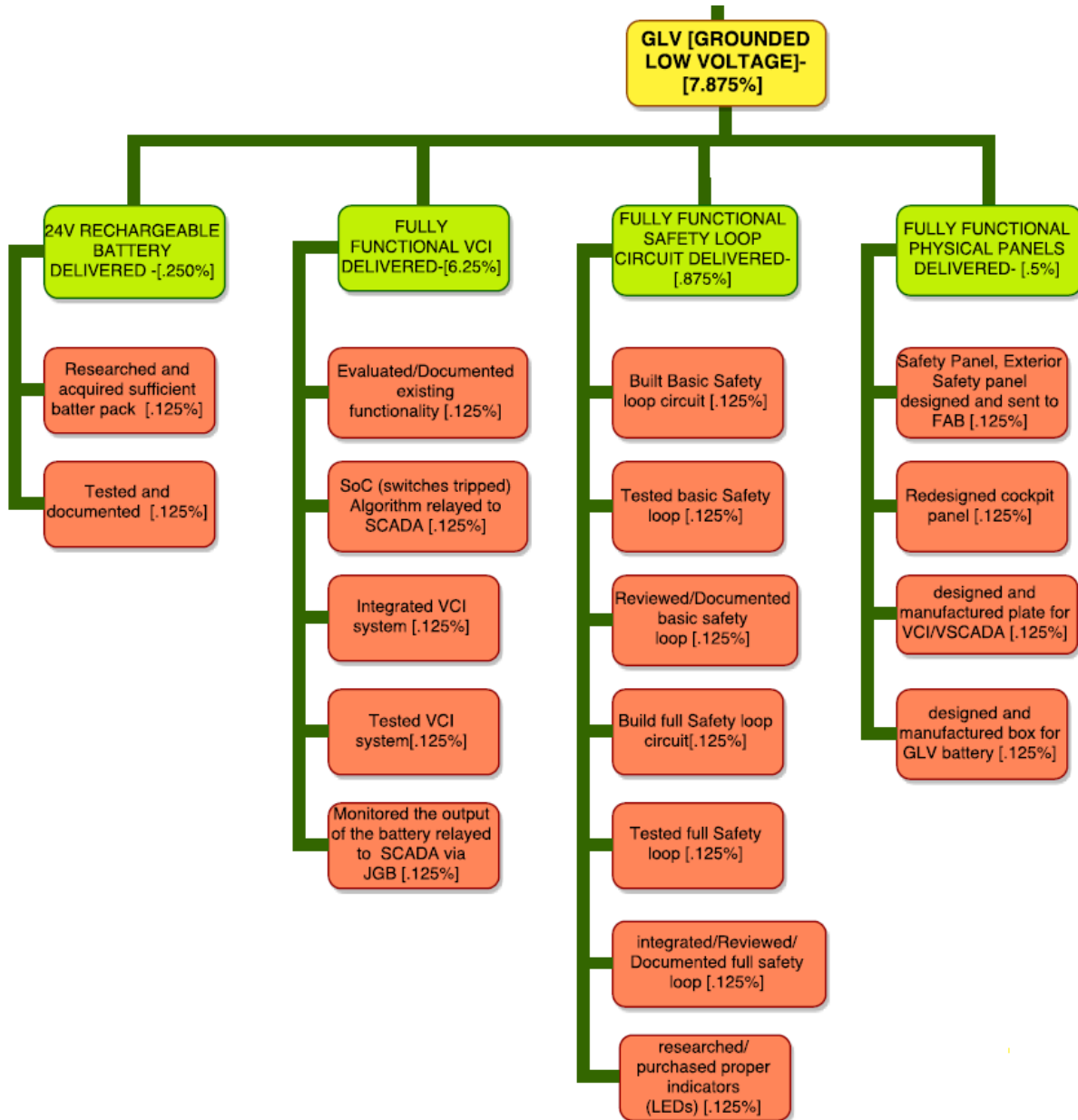
Task	Group Member	Estimated Time
Purchase 24V battery	Chris	2 days
Test and document functionality of battery.	Chris	8-10 hours
Design system to monitor battery SOC and output	Chris	18-24 hours
Build basic safety loop circuit for Dyno Room.	Joe	12-16 hours
Test basic safety loop circuit for Dyno Room.	Joe	12 hours
Purchase LEDs for safety loop.	Kyle	3 days
Review safety loop design and fully document any design changes.	Kyle	18-24 hours
Integrate/build full safety loop circuits for dyno room.	Joe and Kyle	25-30 hours

Test and document full safety loop circuit.	Joe and Kyle	20 hours
Evaluate current functionality of VCI system.	Marty	8 hours
Determine and design system to meet requirements of VCI (need to discuss with VSCADA team).	Marty	14-18 hours
Integrate VCI system with existing systems and VSCADA computer.	Marty	12-16 hours
Test VCI system functionality.	Marty	20 hours
Recover useable buttons, lights and displays from current cockpit and safety panel	Kyle	8 hours
Redesign cockpit panel in dyno room.	Joe and Kyle	12 hours
Redesign exterior (safety) panel in dyno room.	Joe and Kyle	12 hours
Create plate for safety loop, VCI, and VSCADA in dyno room.	Joe, Kyle, and Marty	2 hours
Create box for safety loop, VCI and VSCADA (water proof) for use on vehicle.	Joe, Kyle, and Marty	12-15 hours
Create box for GLV battery for use on car.	Chris	12-15 hours

5.1.5 Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R003a. GLV Power	<p>[a1]provide DC supply voltage with sufficient current to supply all the power needs</p> <p>[a2]rechargeable battery to run at least 3 hours</p> <p>[a3]be rechargeable by UL listed charging device</p> <p>[a4]capable of powering the GLV- "plug and forget"</p> <p>[a5]protected from full discharge, overcharge, overcurrent, overvoltage</p> <p>[a6]safely sit idly without fear of damage</p> <p>[a7]fully charge a discharged battery</p> <p>[a8]provide housing for battery and related electronics</p>	<p>[3a] Chris</p> <p>[a1] Chris</p> <p>[a2]Chris</p> <p>[a3]Chris</p> <p>[a4]Chris</p> <p>[a5]Chris</p> <p>[a6]Chris</p> <p>[a7]Chris</p> <p>[a8] Chris</p>
R003b. GLV Safety Loop	<p>[b1]packaged to be compliant with formula EV</p> <p>[b2]safety loop analysis document</p> <p>[b3]analysis/discussion of possible faults</p> <p>[b4]Provide housing for Safety Loop circuits.</p>	<p>[3b] Kyle/Joe</p> <p>[b1] Kyle</p> <p>[b2] Kyle</p> <p>[b3] Joe</p> <p>[b4] Joe</p>
R003c. Vehicle User Interface Panels	<p>[c1*] Dashboard controls, indicators, switches, displays for dyno room and vehicle.</p> <p>[c2*] Side panels controls, indicators, switches, displays for dyno room and vehicle.</p> <p>[c3*] Provide hardware to allow display of diagnostics to user for vehicle and dyno.</p>	<p>[3c] Joe</p> <p>[c1] Kyle</p> <p>[c2] Kyle</p> <p>[c3] Joe</p>
R003d. Vehicle Computer Interface Hardware	<p>[d1]Relay safety loop info to VSCADA via CAN bus.</p> <p>[d2]Relay SOC info to VSCADA via CAN bus.</p> <p>[d3]Provide housing for VSCADA system.</p> <p>[d4]Provide hardware for interfacing with VSCADA system (usb drives, SD card slots, etc.)</p>	<p>[3d] Kyle</p> <p>[d1] Marty</p> <p>[d2] Marty</p> <p>[d3] Kyle</p> <p>[d4] Kyle</p>

5.1.6. Work Breakdown Structure



5.1.7. Individual Acceptance Test Strategy

We have four main subsystems of the GLV that we need to test. GLV Power, the safety loop, VCI and VCU.

GLV Power: The purpose of GLV Power is to provide power to all of the low voltage systems on the car. Specifically, the system needs to provide 24V. One specific aspect that we need to test is the functionality of the battery. We need to develop some sort of testing circuit to draw the expected amount of current from the battery for a given amount of time and prove that the battery can last for long enough and handle current

spikes. In addition, we need to be able to test that the battery charges in an appropriate manner.

The GLV Power system is integrated with the safety loop so that different sub-systems receive power in the appropriate order. As different switches are tripped, different parts of the system will receive the 24V. We can make sure this aspect of GLV works once we are sure that the safety loop works as well.

Safety Loop: We are developing a preliminary/basic version of the safety loop for use in the Dyno-Room to interface with the packs and 100V power supply. This basic version will need to be tested in the Dyno-Room before we move onto developing a version of the safety loop that integrates all features/systems in the Dyno-Room. We will need to test every possible fault location to prove that it works as expected. Once we have this assured, we can move on to creating a safety loop which can be integrated into the vehicle. This will require the same testing plan as the system in the Dyno-Room.

VCI: Since the VCI functionality and requirements are loosely defined at this point, it is difficult to figure out exactly what will need to be tested. We may possibly have to work on developing hardware to allow the SCADA computer to interface with the CAN bus. In addition, we may have to work with some JGB boards so that we can relay GLV data to the CAN bus/SCADA computer.

VUI: Some of the main restrictions on the VUI are the size requirements and the functionality of the display. We need to make sure that the display available to the driver is large enough to display all the necessary information. In addition we need to develop an mock panels for the cockpit and the exterior.

5.2. JGB [John Gehrig Board (CAN bus DAQ board)]

5.2.1. Current State of Affairs

Currently there is are schematic and gerber files. The software exists but not in a maintainable way. Currently there are 2 parts of the software; there is example code from the avr site and there is a messy program that may be useful as a stepping stone. There are 2 major tasks that need to be completed.

5.2.2. Goals

First the hardware must be updated to be 24V compliant. Secondly the software should be packaged into a useable format. This would involve making an API for development with the board. The end goal is to have a stand alone JGB that anyone can develop with, even if they are not working on a car. As a result another key aspect would be detailed documentation describing all of the functions implemented in the software. Another decision we need to make is about the connectors: should we keep the screw terminals or should we move to a different connector.

5.2.3. System Level Diagrams

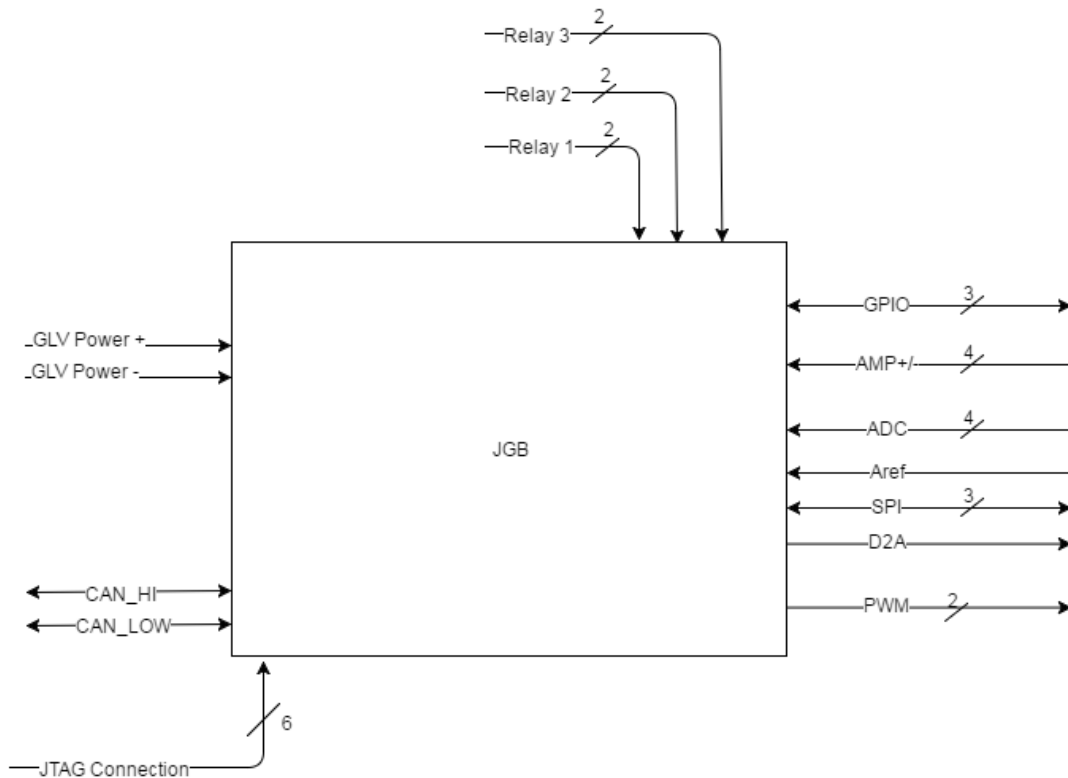


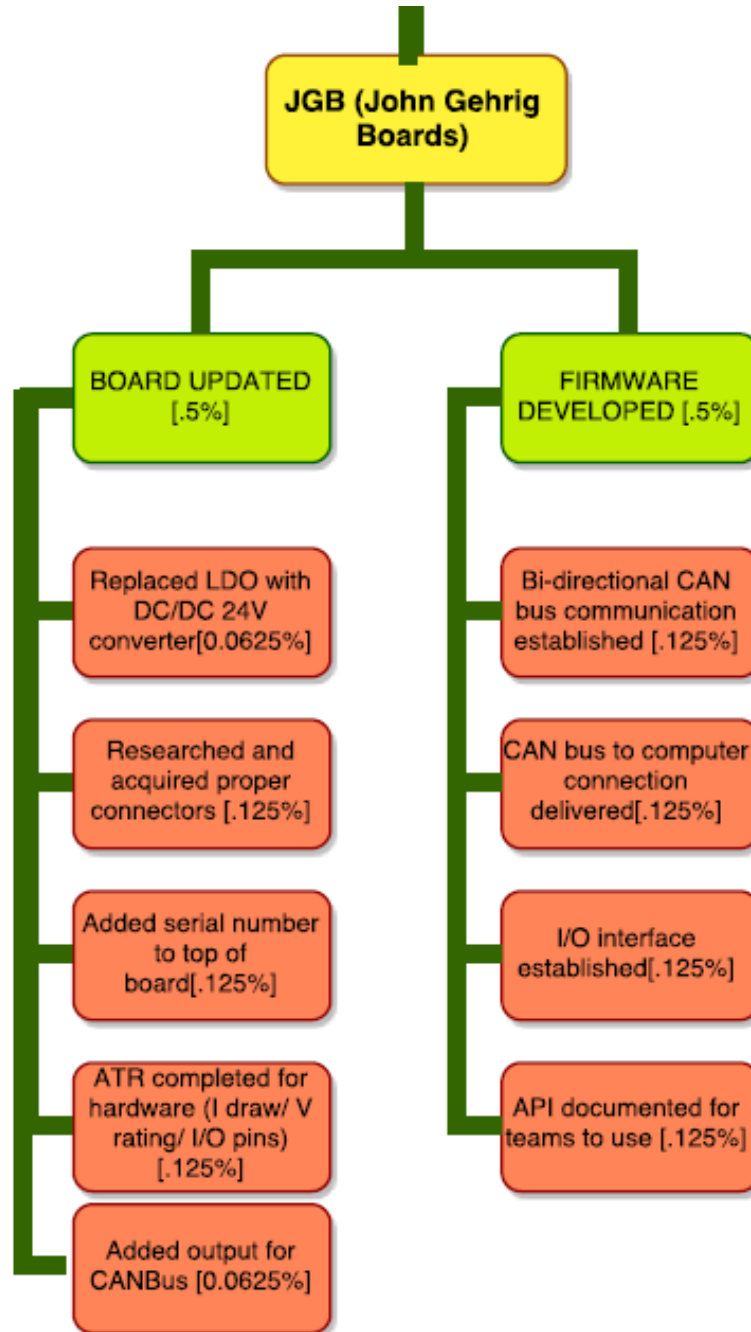
Figure 1. This system block diagram shows all of the hardware connections to the JGB. We will develop and improve an API to be able to control all of these connections.

5.2.4. Task Delegation

The person of contact for this project is Greg Flynn. The biggest pitfall will be working with the CAN bus data line. We have had no experience working with a CAN bus line before. Writing the embedded system may also be a bit of a challenge. Some of us have had experience developing in the AVR environment. The biggest budget item would be paying for all of the circuit boards to be cut. Other than that there are just the components for the board.

Task	Group Member	Estimated Time
24V compliance	Greg	1 week (+2 additional weeks for HW to get here)
Desired API completed	Marty	2 weeks
Populate boards	Greg	2 weeks
CAN bus implementation	Greg/Marty	4 weeks

5.2.5. Work Breakdown Structure



5.2.6. Individual Acceptance Test Strategy

There will be 3 main aspects to test. The first test is that the hardware functions as expected before attempting to use a CAN bus. This would involve upgrading to 24V compliant as well as testing the I/O pins. The second tests would be focused on unit testing the functions. CUnit appears to be a good program to use. Finally there would need to be testing of the hardware functionality.

5.3. Interconnect, Cabling, ICD [Interconnectivity Document]

5.3.1. Current State of Affairs

The overall system diagram currently has all known cables, labeled with wire type and connector types, but not lengths. We are using the same wire and connects in order to make our cables. A few of the wires connecting the packs are already assembled and ready. Inventory of wires, previously-made cables, and previously-bought connectors related to project has been taken, including quantities and lengths.

5.3.2. Goals

We will be reviewing all updates to the overall system diagram to determine what cables need to be created. We also have to order connectors and wires that we either don't have/haven't been able to find. Lastly, we need to replace the 6-pin connectors on the TSV packs with 3-pin connectors. This will involve closely working with TSV and the MechEs. We will also closely work with all of the teams to help deliver ICDs for all required subsystems.

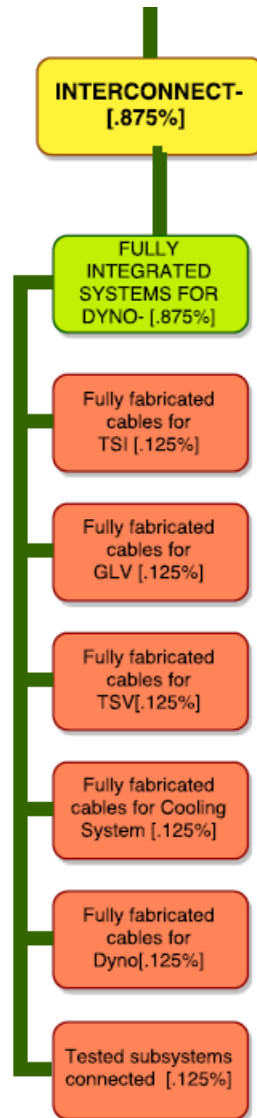
5.3.3. Task Delegation

Task	Group Member	Estimated Time
Inventory	Jeremy	5 hours
Measurements	Larisa	4 hours
Determining required materials	Jeremy	2 - 3 hours
Cable and adaptor assembly	Larisa and Jeremy	3 weeks
Individual Cable Testing	Larisa and Jeremy	3 weeks
System Assembly, Integration, and troubleshooting	Larisa and Jeremy	4 weeks
Create ICDs	Larisa and Jeremy	3 weeks

5.3.4. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R004a. Cabling	[a1]integration of TSV, GLV, motor/controller on Dynamometer [a2]integration of TSV, GLV, motor/controller on Formula car chassis	[4a] Larisa & Jeremy [a1] Larisa & Jeremy [a2] Larisa & Jeremy
R004b. Documentation	[b1]System Diagram [b2]Interface Control Documentation	[4b] Larisa & Jeremy [b1] Greg [b2] Larisa & Jeremy

5.3.5. Work Breakdown Structure



5.3.6. Individual Acceptance Test Strategy

The way we would test the cables is put them under the same conditions they will have when actually in the system. This means providing the same amperage to the cables that the car will. Including but not limited to these cables:

- All pack to pack connections
- Pack 1 and 4 to TSI connections
- TSI to foot pedals connection
- TSI to stop light and TSAL connections
- TSI to Cockpit connection
- TSI to GLV/Scada connection
- TSI to motor controller connections
- TSI to right side connection
- Cockpit to GLV/Scada connections
- GLV Power to GLV/Scada, right side, and left side connections
- GLV Power to Cooling controller connection
- Cooling to pumps and fans connections (Not Definite)
- Motor controller to Motor connections

Once they are tested individually we can integrate the physical system. Once attached, we can read the actual voltage, current, and resistance and make sure they are within the range for the particular cable.

5.4. VSCADA [Vehicle Supervisory Control and Data Acquisition]

5.4.1. Current State of Affairs

Upon analysis of the current VSCADA System, it appears that there has been substantial progress made towards creating a fully functional product. Last years team, led by Brendon Carroll, has created an operational data acquisition which can be used through firefox. The current setup includes a Raspberry Pi3, a WiFi module, and a touch screen display (which is used solely for visual display). While what has been done is valuable, it is not all practical for the current goals set forth by this years team. The code last year, which was written in Python, was 'poorly optimized' and thus in need of some rewriting.

5.4.2. Goals

It is the intent of our team that the software will be rewritten in a more maintainable and user friendly form than that of the current code. To accomplish such a goal, we will be rewriting the current code in Java and will be creating a hard client to handle the data storage rather than using a web server. Since we will not just be rewriting the current VSCADA software in another language, but also improving and expanding at the same time, prior work will serve mainly as a guideline/reference. All data shall be stored in a database; most likely MongoDB, although we have yet to make a definitive decision. One of the most important criterias for our database shall be the development for protocols to ensure data preservation should the power suddenly cut

out. We aim to make a simulation that is capable of testing various components of the overall system such as throttle or load on the Dyno to help aid in testing subsystems. We plan on implementing all criteria set forth in the SOW, excluding the requirement for onboard GPS and INS devices. With a dedicated team, we are reasonably certain that we will be able to complete the project set forth by the given final delivery date.

5.4.3. System Level Diagrams

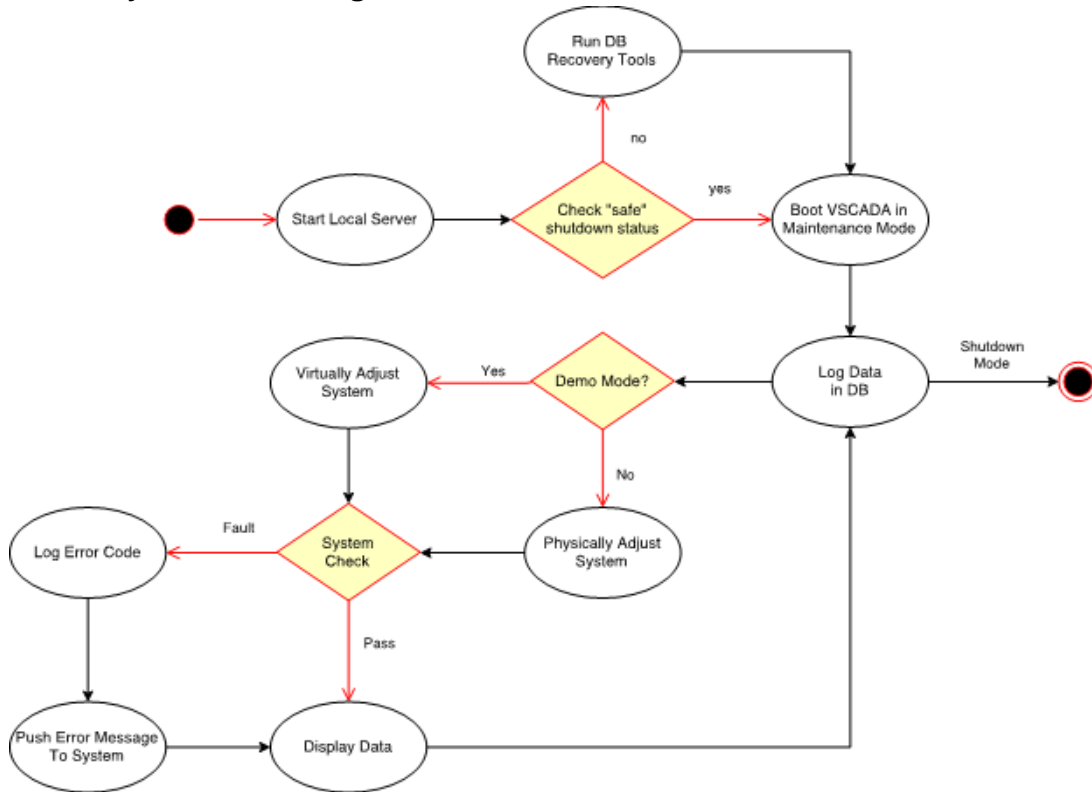


Figure 1. The proposed program control flow for the VSCADA system.

Item	DataType
pack1_SOC	FLOAT
pack2_SOC	FLOAT
⋮	⋮
cell1_V	FLOAT
cell1_A	FLOAT
⋮	⋮

Figure 2. The generic formatting for data storage. Every data point will include a timestamp.

5.4.4. Task Delegation

Task	Group Member	Estimated Time
Create a hard client	Austin	<1 Week
Create MongoDB	Craig	<1 Week
Rewrite all code from Python to Java	Craig and Austin	3 Weeks
Interfacing with subsystems	Craig and Austin	2 Weeks
UI - including different modes	Craig Austin	2 Weeks
Interfacing with Dyno	Austin	1 Week
Interfacing with Cell App	Craig	1 Week

5.4.5. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R002a. Car Dashboard Display	[a1] Successfully display speedometer, odometer, accumulator voltage and current, SOC fuel gauge, Temperatures, "check engine" light, log messages. [a2]"Soft throttle" with cruise control [a3] Integration with dynamometer including torque, motor RPM, calibrated control of torque load	[2a] Craig Lombardo [a1] Austin Wiles [a2] Zainab Hussein [a3]Austin Wiles
R002b. Safety System	[b1] controls [b2] indicators [b3] electrical interfaces	[2b] Craig Lombardo [b1] Craig Lombardo [b2] Austin Wiles [b3] Craig Lombardo
R002c. Remote Computer Interface	[c1]Complete dashboard display and control [c2]Full diagnostic display of all car and dyno data [c3]Full maintenance mode control of car and dyno parameters	[2c] Craig Lombardo [c1] Austin Wiles/Craig Lombardo [c2] Austin Wiles/Craig Lombardo [c3] Austin Wiles/Craig Lombardo
R002d. Remote Cell Phone App Interface	[d1] Data-only summary [Integration only]	[2d] Austin wiles

Required Functional Modes

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R002e. Maintenance mode	[e1]acquire and display calibrated measurands [e2]acquire and display un-calibrated measurands [e3]explicitly set control functions [e4]safety and security features	[2e] Craig Lombardo [e1] Craig Lombardo [e2] Craig Lombardo [e3] Austin Wiles [e4] Austin Wiles
R002f. Drive Mode	[f1]minimal display and interaction with extensive automation [f2]safety checking per formula EV	[2f] Craig Lombardo [f1] Craig Lombardo [f2] Austin Wiles
R002g. Drive mode demo	[g1]demo that allows all functions of driver mode exercised [g2]demo that allows all functions of driver mode tested [g3]demo that allows all functions of driver mode demonstrated	[2g] Craig Lombardo [g1] Austin Wiles/Craig Lombardo [g2] Austin Wiles/Craig Lombardo [g3] Austin Wiles/Craig Lombardo
R002h. Charging Mode	[h1]individual cell data acquired and visualized for every charging period [h2]overall cell data acquired and displayed for every charging period	[2h] Craig Lombardo [h1] Austin Wiles [h2] Austin Wiles
R002i. Shutdown Mode	[i1]long term storage without draining batteries	[2i] Craig Lombardo [i1] Craig Lombardo
R002j. Monitoring and Data Acquisition	[j1]continuously collect data for real-time forensic analysis [j2]display data from j1 on all interfaces [j3]transfer data other computers via network file transfer/removable media	[2j] Craig Lombardo [j1] Craig Lombardo [j2] Austin Wiles [j3] Austin Wiles

Data Acquisition

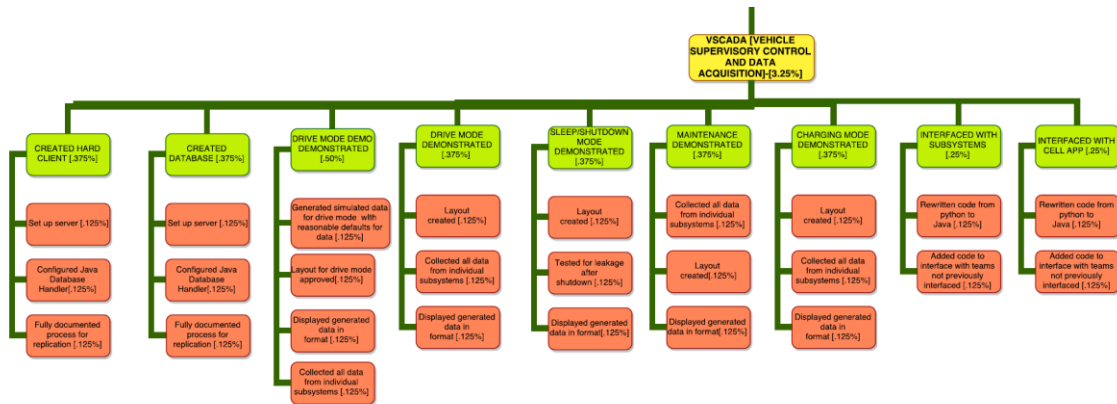
Component	Tasks/Deliverables/Desired Outcomes	Engineer
R002k.	[k1]acquire data from vehicle sensors [k2]convert data from k1 to calibrated measurands [k3]inherent support for logging sorting and updating calibration [k4]sensor data ready for review [graphical representation] [k5]log of events/exceptions/faults/ changes in operational state/ safety interface of LFEV	[2k] Craig Lombardo [k1] Craig Lombardo [k2] Craig Lombardo [k3] Craig Lombardo [k4] Austin Wiles [k5] Craig Lombardo/Austin Wiles

R002MISC	[MISC1] overall TSI voltage/current/power delivered to the motor controller [MISC2] individual cell voltages/ aggregate pack voltages and current [MISC3] sensor outputs required to assist diagnosis of failure in the pack fuse [MISC4] state of charge of the aggregate accumulator /separate packs/individual cells [MISC5] temperatures of all subsystems/individual cells/significant fuses [MISC6] data available from motor controller [MISC7] vehicle speed/distance travelled [MISC8] safety loop system rate [MISC9] temperatures of all major systems [MISC10] insulation monitor data [MISC11] data available from GPS or INS located on car [MISC12] Data available from HUFF dyno-torque/oil and coolant temp/constant flow rate/RPM [MISC13] include uncertainty parameters [MISC14] Parameterization to be sampled 60 times a minute [MISC 15] Data storage to have enough capacity for retaining data records over lifetime of system [MISC16] Back up system/re-installation recovery strategy	[2MISC] Craig Lombardo
----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------

SCADA Automation

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R002l. Control	[1]closed loop control [2]scripting of motor controller system test stand operation [3]dynamic motor RPM input and Torque [4]trapezoidal profiles simulating actual car dynamics on the competition racecourse	[2l] Craig Lombardo [1l] Craig Lombardo [2l] Craig Lombardo [3l] Craig Lombardo [4l]Craig Lombardo/Austin Wiles/Zainab Hussein
R002m.Alarm	[m1]automatically log an event/enable alarm/declare a fault to shutdown [m2]add/alter alarm rules	[2m] Craig Lombardo [m1] Craig Lombardo [m2] Craig Lombardo

5.4.6. Work Breakdown Structure



5.4.7. Individual Acceptance Test Strategy

To test our system it will be necessary to run through a few key aspects. Our first step will be to establish a new wireless link between the Raspberry Pi3 and an external computer. From there we plan on creating a mocked up database that will have fake data. This will allow both the VSCADA cockpit and cell phone teams the opportunity to interact with data as we hope to in the finalized product. This will allow us to also begin experimenting with data writing, reading, and protection in the case of various errors. This will also be a good way for our team to remain transparent and demonstrate our work as we go.

Once a server is up and running, even though it will contain fake data, our end users will have the ability to preview how the final product will interact, thus providing us with feedback as we go. The next step will be to begin revamping the code already existing. While this will require a rather large effort, having pre-existing work will allow us a framework to fall back on and also, hopefully some insight on how to debug any potential issues.

Once the system is rewritten we will test its operation with other components such as the JGBs, GLV boards etc. to confirm that we have not missed anything in the process. Once we have everything up and running we will want to check in on the progress made by the cell team to ensure integration is going smoothly. While we will be interacting throughout the process; however, this will be the first large scale test we will be able to perform. From this point we will work on the simulation software and test our abilities to send requests to the TSI team/the motor controller to help change aspects of the dyno such as torque and load. We will also want to test each of the different modes with the dyno, pretending as though we were actually testing them on the car. This will not only aid in debugging, but will also help catch our end user up to speed on how to interact with the interface while receiving feedback on how to improve our display.

5.5. Cell App

5.5.1. Current State of Affairs

There had been no progress towards a functional mobile application for the VSCADA system. Last year's system included a Wi-Fi module and a touch-screen display. These are valuable resources to learn from but the format of the touch screen display is going to be very different from the mobile application interface. However, by using the previous years code, we can get an approximate idea of how to use the data from the VSCADA for data display and what data needs to be displayed.

5.5.2. Goals

Our team's aim is to write an easily maintainable code in an open source environment for the future teams to improve upon. We are planning on using the Android Studio IDE for building the application which will be in Java language. The VSCADA Cockpit software team is also planning to use Java which will make it easier to exchange information in between the teams. Also, we have determined to use a database for storing the data simultaneously for the cases such as unexpected shutdown. MongoDB is the most likely possibility and it seems to have promising features for our design.

Wifi connection with Raspberry Pi3 used by the VSCADA Cockpit team will be through an antenna used by them. Very similar to their system we will have a connection to the database and instead of both writing and reading the data, the mobile application will only read the data. Therefore, cooperation with the Cockpit team will most definitely prove useful in later progress.

We will have the final app in the format of apk and currently we do not plan to upload it to the Google Play Store. This decision can easily be changed later if the situation requires it so. However, this would require an google account that also has the right to upload to the Google Store.

We may possibly use some third party libraries for already prepared charts and graphs for the data presentation. The final software will display these graphs and charts according to the amount of data the user wants to see, depending on the date. Also, the final product will have multiple modes, similar to the cockpit VSCADA, such as Maintenance and Drive modes. The mobile app will not require as many modes as the Cockpit app but still we would like to provide as many options as possible. Also, the final product hope to be similar to the Cockpit app in terms of interface for esthetic purposes.

5.5.3. System Level Diagrams

High Level Block Diagram for LFEV Cell Phone Application

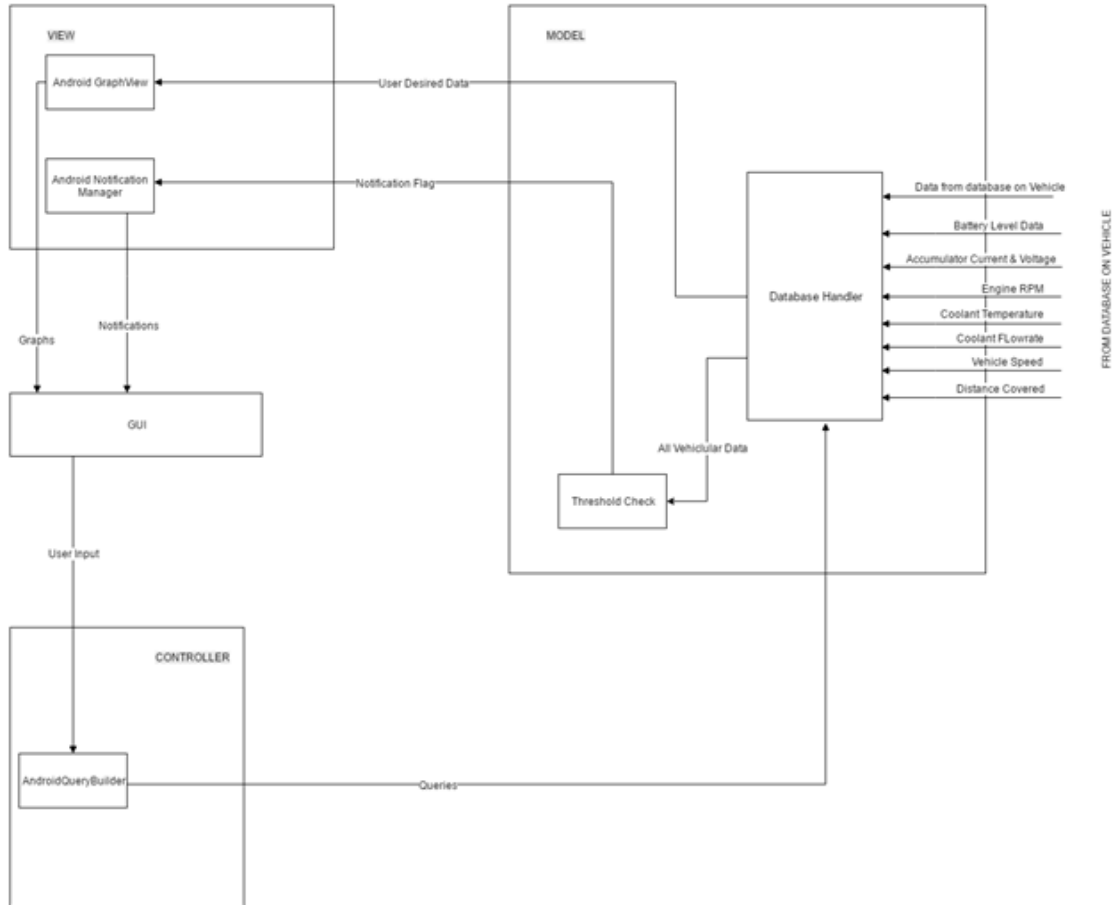


Figure 1. High Level Diagram

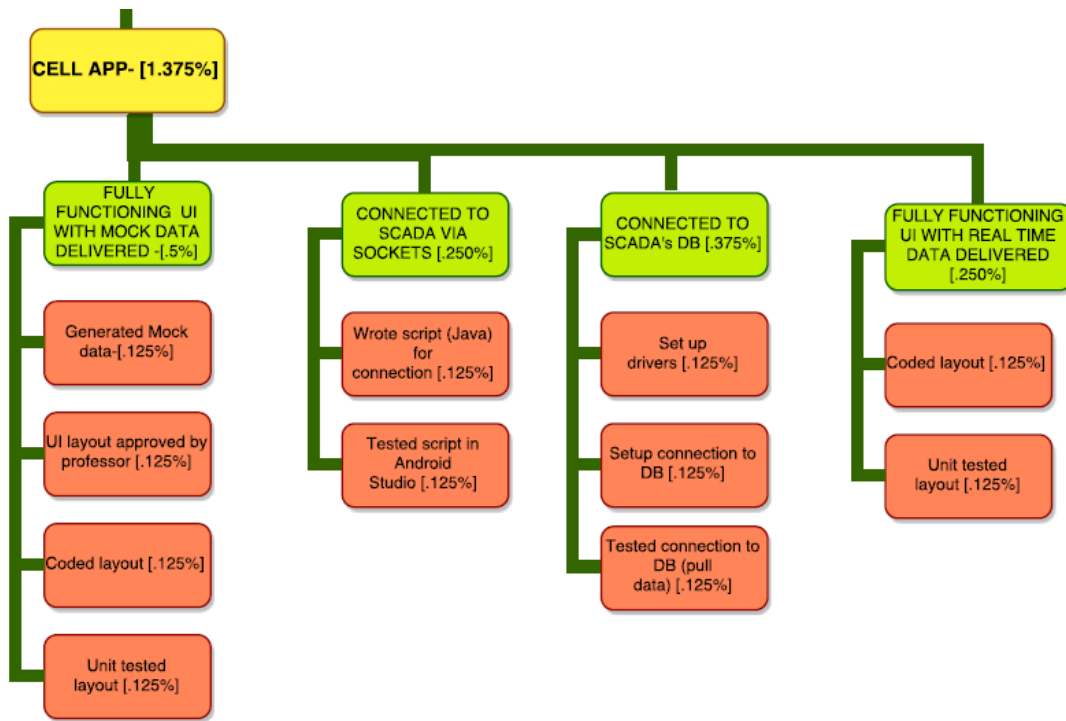
5.5.4. Task Delegation

Task	Group Member	Estimated Time
Create Layout and Professor Approval	Kemal	1 Week
Unit Testing Layout	Raji	1 Week
Database Handling (dummy data)	Kemal and Raji	1 Week
Backend code for data processing	Kemal and Raji	3 Weeks
UI - including different modes	Kemal and Raji	2 Weeks
Connecting with MongoDB	Raji	1 Week
Interfacing with Cockpit App	Kemal	1 Week

5.5.5. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R002d. Remote Cell Phone App Interface	[d1]Data-only summary	[2d] Kemal [d1] Raji

5.5.6. Work Breakdown Structure



5.5.7. Individual Acceptance Test Strategy

Testing the application will mainly follow three consecutive levels:

1) Unit Testing: At a very low level, individual components of the application like access to the database and accurate visualizations of data will be tested by specific unit tests. Android Studio has built in unit testing tools and the mobile application team will be making use of these. This will have high maintainability and can be easily updated as the new versions of Android Studio will come out.

2) Virtual Testing: Versions of the application will then be tested on a virtual simulated device. Android Studio also comes with a Android Device Emulators. We will use the API for Android 6.0 Marshmallow assuming that the team will have phones that support Android 6.0 Marshmallow.

3) Device Testing: The last level of testing is actual on device testing. For this purpose, we will use multiple different phones running on Android Marshmallow(6.x) and Nougat (7.x).

Until the VSCADA team can accurately receive data from the various other teams, dummy test data will be placed in the database for testing purposes. This dummy data will have the same structure as the database plans so that the real life data wouldn't have any problem during the implementation with the application.

Additionally, we will provide a plan for easily running tests for future teams to be able to recognize which tests are for which purposes. The tests will include comments to explain which part of the software is being tested and what the results mean. Therefore, even if the user does not fully know the code behind the application, the user will be able to trace down any bugs or problems in the application. The testing has few important sections: database connection, data handling, passing data to interface and finally building the visual interface(charts, graphs, etc). Each of these will have different files and will test for various erroneous situations. Since the mobile application will be "read-only" for the VSCADA database, any error in the mobile application will possess no threat towards the system or the data collected previously.

5.6. TSV [Tractive Systems Voltage]

5.6.1. Current State of Affairs

For the TSV, the LFEV 2016 team produced one working pack, including the PackMan and AMS boards. For our first requirement R001a, we will be utilizing all existing software and hardware to assemble the additional three packs; we will complete this without modification to the existing hardware or software as all parts have been shown to work within one pack. Once we get all four packs completed, which includes fully assembling and assembling existing documentation, BOM, and testing, we will then connect all four packs to the dynamometer to power the dynamometer. Once the packs can power the dynamometer, we will utilize the packs and dynamometer to gather relevant data. Once testing has been completed with the existing four packs, we will then move on to improvements.

R001b-R001f are unchanged from the 2016 SOW. According to the 2016 test result documents, R001b, R001c and R001f was completed according to the parameters and tests laid out in 2016's ATR (<https://sites.lafayette.edu/ece492-sp16/files/2016/05/QAR001a.pdf>, <https://sites.lafayette.edu/ece492-sp16/files/2016/05/QAR001b.pdf>, <https://sites.lafayette.edu/ece492-sp16/files/2016/05/QAR001e.pdf>). R001c will need to be updated in order to communicate to the new VSCADA system. R001d and R001e was not fully completed according to the parameters and tests laid out in 2016's ATR and, thus, will need improvement (<https://sites.lafayette.edu/ece492-sp16/files/2016/05/QAR001c.pdf>, <https://sites.lafayette.edu/ece492-sp16/files/2016/05/QAR001d.pdf>). R001g was not a requirement in 2016 and thus will need to be reviewed and implemented in 20. We do not want to explicitly omit any requirement within R001; however, we may choose to not improve upon the existing work done by previous years that satisfy existing requirements.

5.6.2. Goals

We have listed a priority list for these requirements below. Items 3-7 have equal priority. As stated above, our main focus is on goal 1, requirement R001a.

1. R001a- Integration
 - a. Documentation, Updates, Integration, Testing of each individual pack
 - b. Assemble 4 packs in series
 - c. Assemble 4 packs to Dynamometer
 - d. Data Acquisition from Dynamometer
2. R001g- Safety Shutdown
3. R001b- Cell Balancing Algorithm
4. R001c- Data Acquisition
5. R001d- Displays and Indicators
6. R001e- Pack Controls
7. R001f- Low Current Output

5.6.3. System Level Diagrams

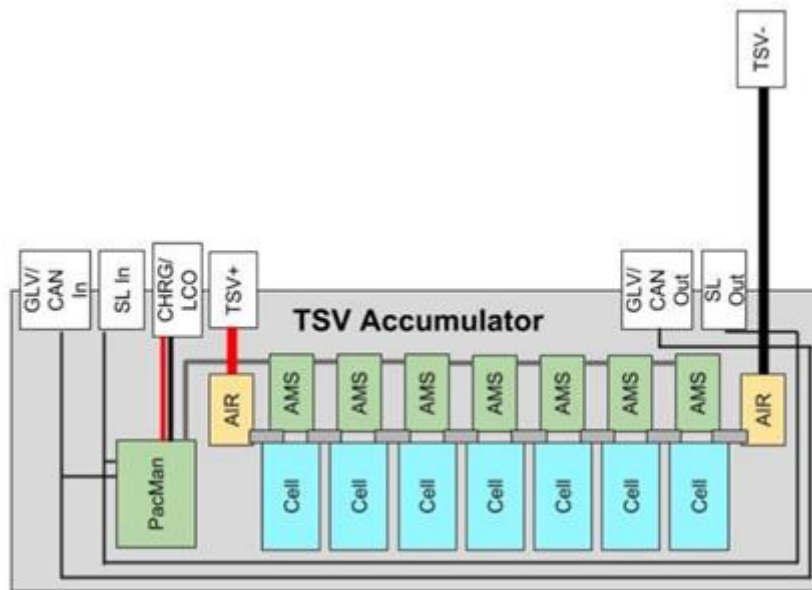


Figure 1. High Level block diagram for the single, completed accumulator; the other three accumulators will not have two AIRs. Credit: 2016 Final Presentation

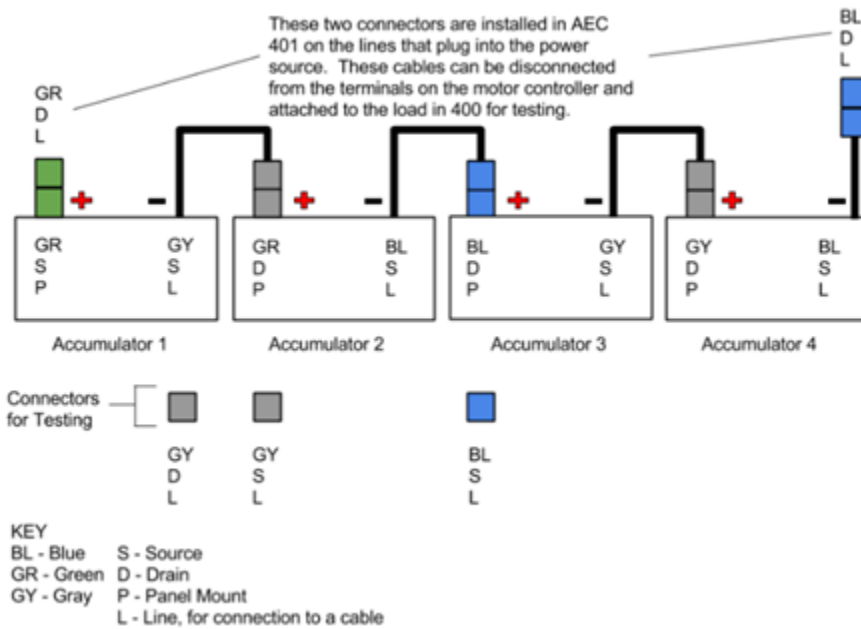


Figure 2. High Level block diagram showing the interconnections between the four accumulators. Credit: 2016 User's Manual TSV Accumulator

5.6.4. Task Delegation

Task	Group Member	Estimated Time
Populate 1 PacMAN board	Meri	4 days
Test PacMAN board with AMS boards	Emilie and Meri	0.5 days
Debugging PackMAN board (if necessary)	Meri	4 days
Debugging PackMan tester	Emilie	4 days
Test existing AMS boards	Emilie	2 days
Populate the rest of the PackMan boards.	Emilie and Meri	15 days
Test PackMAN boards with AMS boards	Emilie and Meri	0.5 days
Debug PackMAN boards	Meri	6 days
Charge each cell so that they are equally weighted	Emilie, Meri, and Rachel	6 hours/cell, for 21 cells Total of 126 hours
Find aluminum bars	Emilie and Meri Ted and Scott (MechE team)	.1 day
Install bars, AMS boards, and Packman boards into 3 packs	Emilie and Meri	3 days
Test packs individually	Meri	5 days
Debug Packs	Emilie, Meri, and Greg	2 weeks
Put 4 packs in series	Emilie, Meri, and Greg Prof Nadovich / Schmult	1 day
Put packs in dyno room, connect to dynamometer	Emilie and Meri Prof Nadovich / Schmult	1 day
Debug system	Emilie, Meri, and Greg	2 weeks
Connectivity between TSV and SCADA	Emilie, Meri, Craig, and Austin	3 days
Modify CAN connector on control	Emilie, Meri, Jeremy, and	1 day

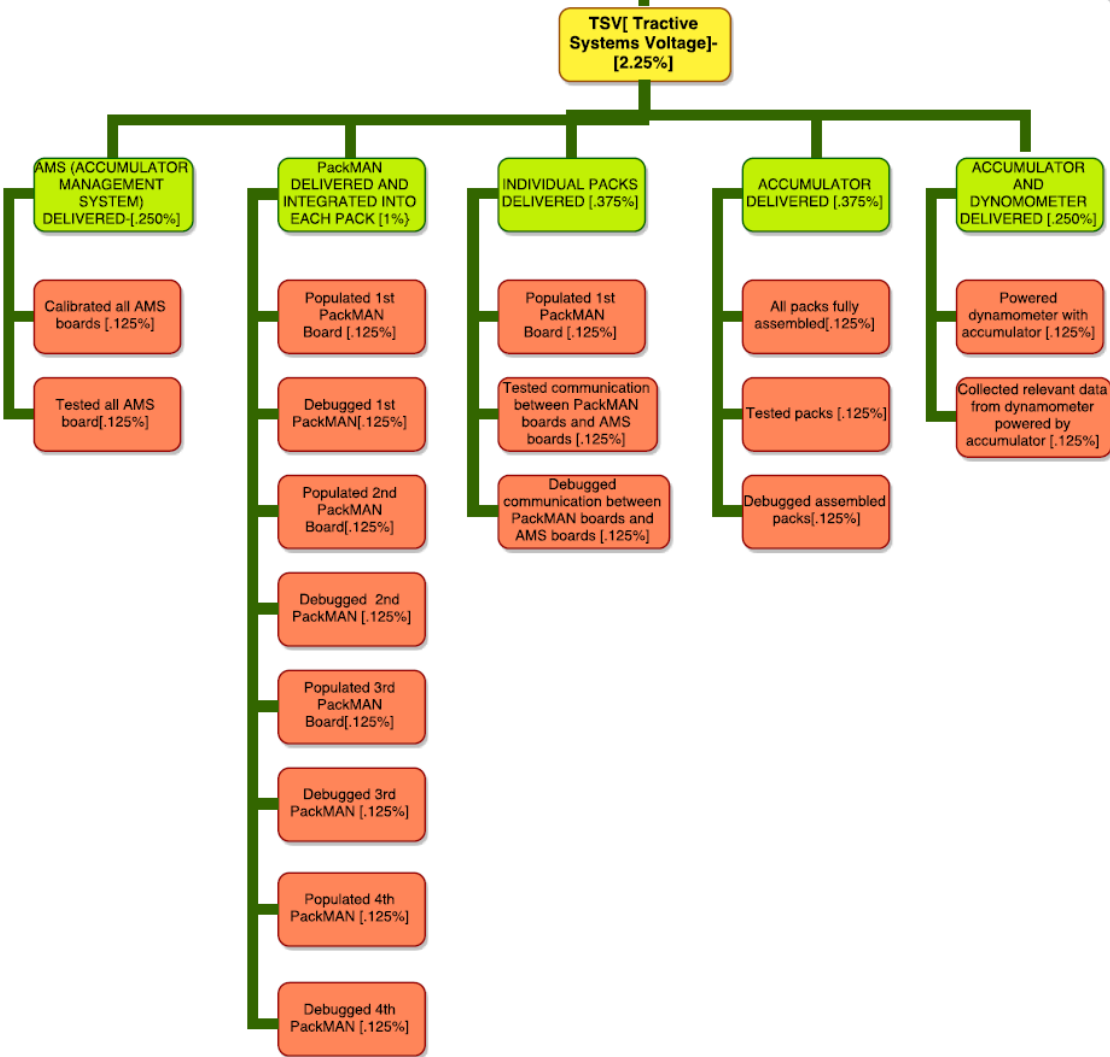
board	Larisa	
Cell balancing Algorithm	Emilie, Meri, and Zainab	Time permitting

5.6.5. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R001A. Integration	[a1] Complete assembly of accumulator. [a1.1] full documentation of assembly [a1.2] full integration of 4 working packs. [a1.3] full updated to most recent hardware and firmware revisions as of Fall 2016 [a1.4] full testing of individual packs and assembled accumulator	[a1] Meredith Guro [a1.1] Emilie Grybos [a1.2] Emilie Grybos [a1.3] Emilie Grybos [a1.4] Emilie Grybos
R001B. Cell Balancing Algorithm	[b1] Functional cell balancing algorithm in 4 packs as of current Fall 2016 firmware. [b2] Modification and extension of current cell balancing algorithm to manage all charging states and balancing of charging cells.	[1B] Emilie Grybos [b1] Meri Guro [b2] Emilie Grybos
R001C. Data Acquisition	[c1] Communicate with VSCADA team to acquire data for AMS - cell voltage, pack voltage, cell temperatures, pack internal temperatures, pack current, SOC estimates, charge algorithm state [c2] calibration and error analysis	[1C] Meri Guro [c1] Emilie Grybos [c2] Meri Guro
R001D. Displays and Indicators	[d1] fully functional display that shows [but is not limited to] pack voltage, current, SOC, cell balancing state, charging state, charging history, discharge history, cell temperatures, cell voltage, safety loop state	[1D] Emilie Grybos [d1] Meri Guro
R001E. Pack Controls	[e1] allow interacting with the data display [e2] allow setting of important modes or parameters for pack operation [e3] allow resetting to the PacMan AMS boards	[1E] Emilie Grybos [e1] Meri Guro [e2] Emilie Grybos [e3] Meri Guro
R001F. Low Current Output	[f1] allows access to the full pack voltage with 10 amp sustained current capacity.	[1F] Emilie Grybos [f1] Meri Guro

R001G. Safety Shutdown	[g1] must break safety loop during charging period [g2] must break safety loop during any unsafe period [g3] displays, indicators, interface must communicate fault condition	[1G] Emilie Grybos [g1] Meri Guro [g2] Emilie Grybos [g3] Meri Guro
------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------

5.6.6. Work Breakdown Structure



5.6.7. Individual Acceptance Test Strategy

The main units to test in TSV are: AMS board, PackMan board, assembled pack, any added firmware. We will test 21 AMS boards with the AMSVU, the 3 PackMan boards with JG test unit, assembled Packs. We will visually inspect wiring, visually inspect mechanical assembly

5.7. Controller Cooling System

5.7.1. Current State of Affairs

In the previous years, there did not exist a fully functional cooling system. A new cooling system based on 24 VDC needs to be built and new parts will be bought after budget planning and nothing of the old cooling system in Room 402 will be reused or moved (as required).

5.7.2. Goals

This year, we will on add the CAN-bus interfacing , as well as programming a fully functional system controller. We will create a cooling system that includes a new pump working on 24 V DC, 2 new fans, 1 new radiator. This new cooling controller will be able to measure the inlet, outlet fluid temperature, and fluid flow rate. Output data to SCADA via CAN bus. It will also have the functionality of automatically controlling fan and pump speed based on fluid temperature; control algorithm, as well as manual override. We will have the controller interface with the LFEV safety loop. A fail safe design shall be developed that will open the safety loop when fluid flow speed is too low, or when fluid temperature is too high.

5.7.3. System Level Diagrams

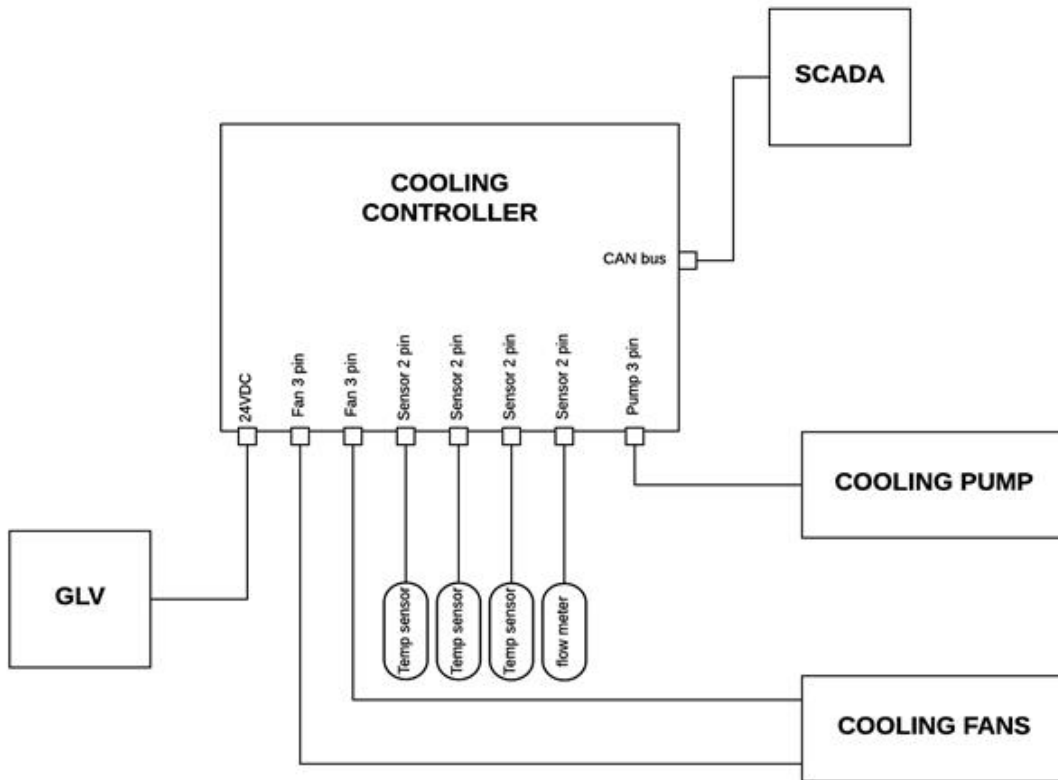


Figure 1. Overall subsystem incorporation

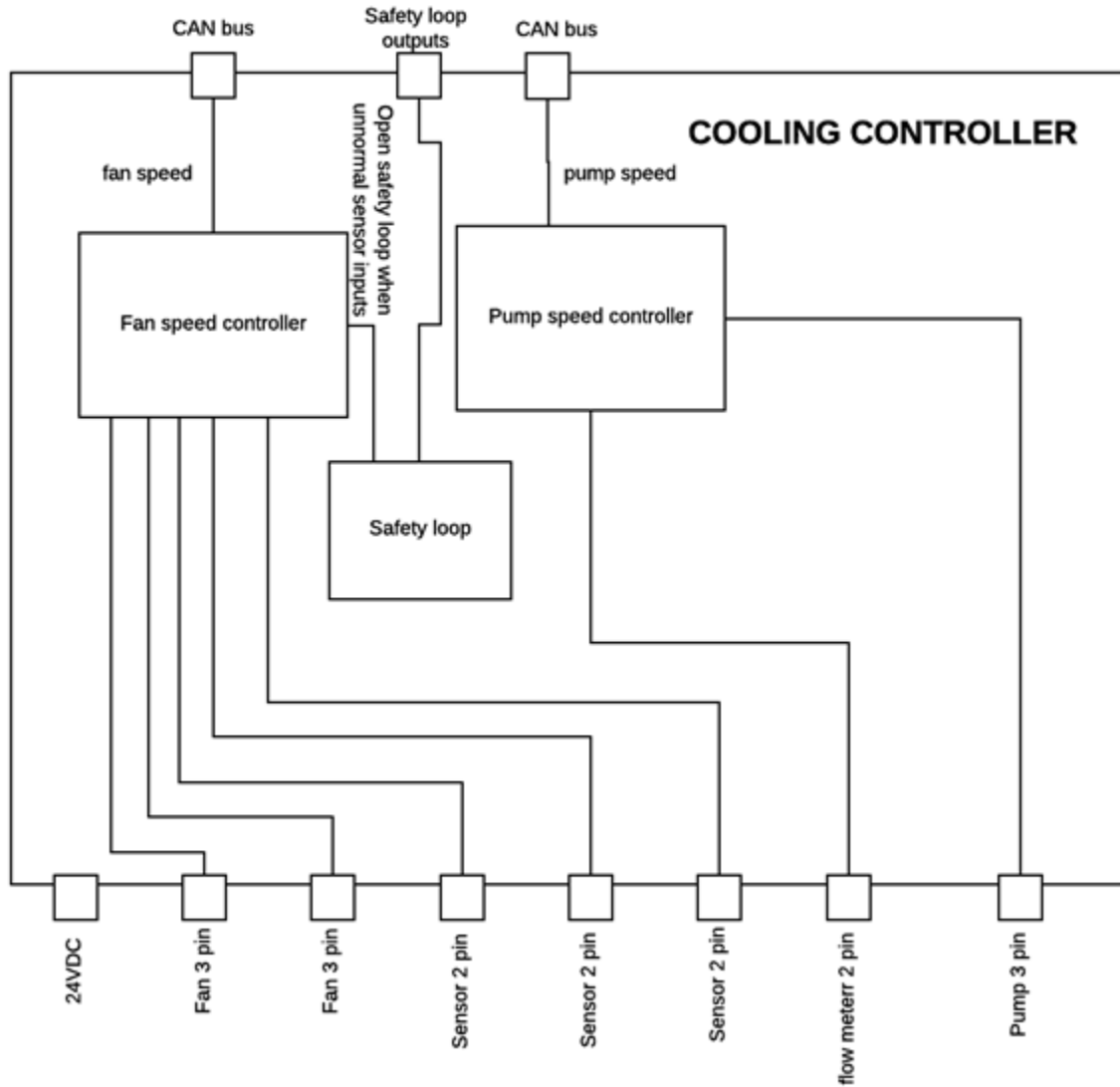


Figure 2. Detailed Cooling Controller Diagram

5.7.4. Task Delegation

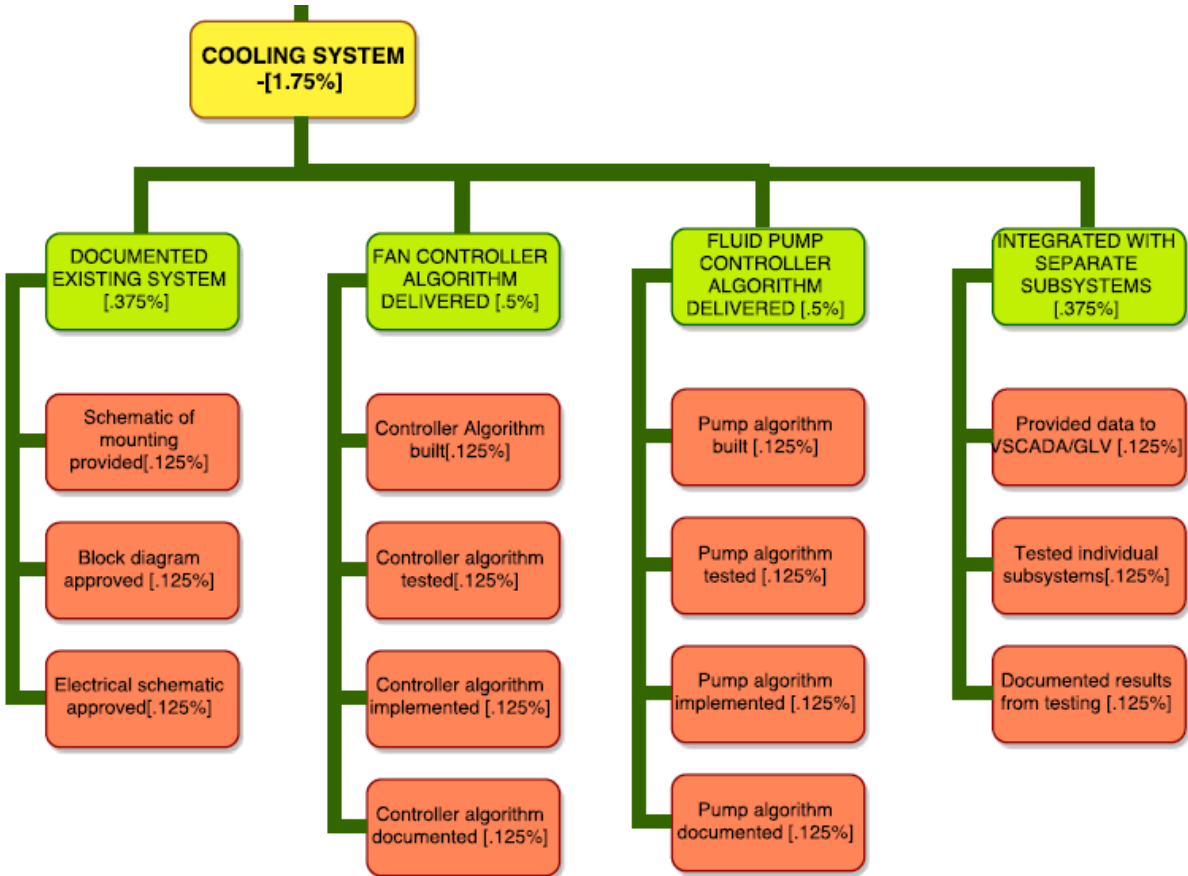
Task	Group Member	Estimated Time
Set up the purchased devices together	Xingyuan	3 hours
Control the fan speed by using Arduino controller	Yi	5 hours
Control the pump speed by using Arduino controller	Xingyuan	5 hours

Read temperature sensors' data to the Arduino controller	Yi	3 hours
Read flow meter's data to the Arduino controller	Xingyuan	3 hours
Set up the Arduino controller with CAN bus shield and test	Yi and Xingyuan	4 hours
CAN bus interfaces design	Yi and Xingyuan	15 hours
Interfacing using CAN bus	Yi and Xingyuan	5 hours
Controller interface with LFEV safety loop	Yi and Xingyuan	15 hours

5.7.5. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R007a. Operating Voltage	[a1]operates entirely on DC power	[7a] Xingyuan [a1] Xingyuan
R007b. Safety Loop Control	[b1]interface with LFEV [b2]failsafe design for when fluid flow is too low [b3]failsafe design for when temp too high	[7b] Xingyuan & Yi [b1] Xingyuan [b2] Xingyuan [b3] Yi
R007c. CAN bus interface	[c1]interface with CAN bus	[7c] Xingyuan & Yi [c1] Xingyuan&Yi
R007d. Data Acquisition	[d1]record inlet/outlet fluid temperature [d2]record fluid flow rate [d3]accessible over CAN bus	[7d] Yi [d1] Xingyuan [d2] Yi [d3] Yi
R007e. Fan and Pump Control	[e1]Fan/Speed pump algo [e2]interfaced with CAN bus	[7e] Xingyuan& Yi [e1] Xingyuan [e2] Yi
R007f. Compatibility	[f1]create controller using prespecified components	[7f] Xingyuan [f1] Xingyuan
R007g. Independence	[g1]existing cooling system is independent and functional	[7g] Yi [g1] Yi

5.7.6 Work Breakdown Structure



5.7.5. Individual Acceptance Test Strategy

The following is how we plan on testing our cooling system:

Test Unit	Observability	Testing Tool
Pump	By providing 24 VDC and speed controllability	Power supply, PWM for speed control
Sensors data	Show on Arduino display	Arduino
Fan	Fan speed controllability	PWM
PVC tubing compatibility	Attach PVC tube to system	By observation
CAN bus interfacing	Output to SCADA	SCADA
Safety loop interface	Provide specific safety loop output and determine whether the cooling system reacts correctly	Interface with LFEV safety loop

5.8. DYNO [Dynamometer]

5.8.1. Current State of Affairs

Currently what we have is an almost working dynamometer, it has worked before but has been out of use for a while and as such needs a little work before it can be used. Otherwise, the dynamometer system can be controlled from the computer in room 201 or controlled remotely. It can also receive measurements of velocity and torque created by the motor controller in the VSCADA software.

5.8.2. Goals

Currently the dynamometer system has some issues from the removal of the old physical implementations of the VSCADA and TSV interfaces and doesn't currently function from the VSCADA computer interface but within a week it should be fixed and up and running. Another issue is that the external velocity sensor is not accurate as was not mounted in the correct location and that should be another quick fix. As the VSCADA system is already integrated with the dynamometer, our goal for this semester is to integrate the dynamometer with the GLV, TSI and TSV to provide communication and power to and from those systems. The current cooling system on the dynamometer is running off a 12V power supply and that will have to be redone to 24V for competition however we can still run tests with the dynamometer with the 12V powered cooling system.

5.8.3. System Level Diagrams

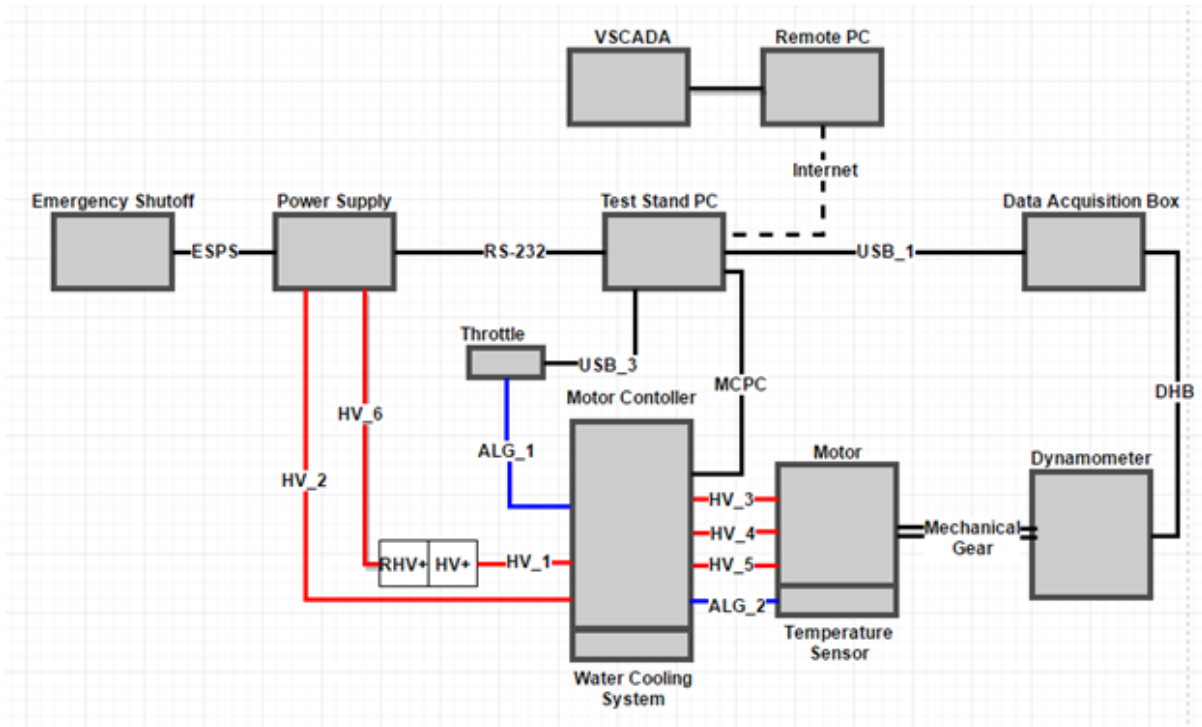


Figure 1. Overview of Dynamometer

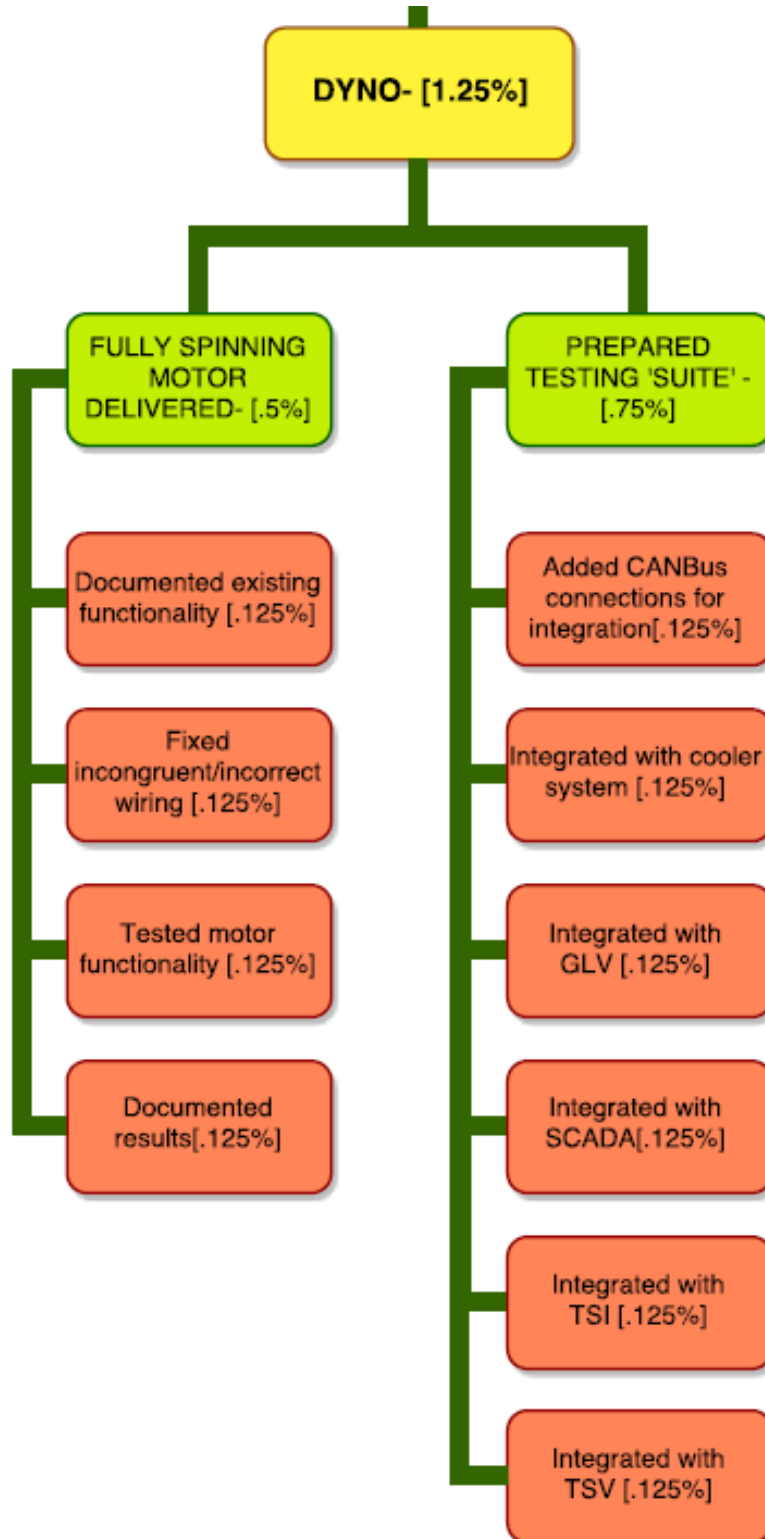
5.8.4. Task Delegation

Task	Group Member	Estimated Time
Fix external velocity sensor so that it gets the correct velocity measurement.	Richard and Eric	1 Week
Fix the Dynamometer test station so that it works again	Richard and Eric	1 Week
Integrate the TSV	Richard and Eric	2 Weeks
Integrate the TSI	Richard and Eric	2 Weeks
Integrate the GLV	Richard and Eric	2 Weeks
Integrate the new 24V Cooling System	Richard and Eric	1 Week
Integrate with the final system cabling and packaging	Richard and Eric	1 Week

5.8.5. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R006a. DYNO	[a1]measures torque *variable torque [a2]measures RPM [a3]fully integrated with TSV, VSCADA, GLV, TSI [a4]final system cabling and packaging [a5]interface with Huff system [torque measurement]/load control	[6a] Richard [a1] Richard [a2] Richard [a3] Richard and Eric [a4] Richard [a5] Richard and Eric

5.8.6. Work Breakdown Structure



5.8.7. Individual Acceptance Test Strategy

The main things to test in our subsystem is making sure that the CAN Bus can correctly communicate and integrate with other subsystems of the vehicle. Currently it seems that the best course of action would be to work on testing the Huff data acquisition board so that the requisite information can be fed through to other subsystems of the vehicle.

5.9. Physics Modeling

5.9.1. Current State of Affairs

At the moment, there exists a high level description of the model is described by the static and dynamic characteristic models. There is a Static characteristics model that takes in inputs of currents and torque to output motor speed in RPM, see figure 1. There is also a Dynamic characteristics model that takes in inputs of speed in rpm and torque to output current, see figure 2. Lastly, there is a gear ratio manipulation that to find the ratio for fastest car by analyzing fixed distance and varying time taken to realize the given distance. Gear ratio highly affects the car performance. The time unit is plugged into the following equations for proper estimate of:

1. Torque through car shaft
$$T_s(t) = GT_m(t) \quad (\text{Eq.1})$$
2. Tractive force through car's rear wheel
$$F_{tr}(t) = T_s(t)/W_r \quad (\text{Eq.2})$$
3. Car's acceleration
$$A_{car}(t) = F_{tr}(t)/M_{car} \quad (\text{Eq.3})$$

5.9.2. Goals

The work already done will be used cautiously, citing the failure to present an accurate model. On this basis, an immediate task to undertake is to widen the scope of references beyond the one book, *Automatic Control Systems*, and reports from previous years.

High level description of the model to also relate the power supplied (in the form of electrical energy) and the power delivered to the rotor (in the form of mechanical energy) to estimate power lost that should be lost as heat. See figure 3. Accurate estimation of the system loss is necessary for the cooling system team to design for that given worst case.

Calculations done in previous years that estimate speed and acceleration are modelling the motor and controller, and take to consideration just the force due to mass of car and motor torque. Additional work on the car's endurance can be focused on here where in addition to manipulating the gear ratios to increase performance, finding the optimal speed is necessary. If other consideration such as:

- Air friction (air-flux equation, whose derivation can be nested into the torque equation)

- Range of weight of the driver for estimate of worst case scenarios for the modelling
- Motor equations chosen in this permanent magnet DC (PMDC) motor from the most recent Physics memo seem to follow an ideal case that consider the capacitive and inductive effects to be of similar scale that cancel the effect of one another. Considering the torque equation is reliant on the assumptions and estimations from the motors among other physical parameters of the car, I am curious to further investigate how this assumption affects the car shaft torque, which is a product of the motor torque.

Find these actual mathematical equations that relate the input and output parameters for both the static and dynamic characteristics models. The physics memo provided shows just the resulting derivations where the coefficients already have numerical values.

There will be contact between myself, Professor Boekelheide (for Mechanics), mechanical engineering colleagues (for physical specification of the car), the SCADA team (for necessary questions on the computer algorithm) and relevant credible printed media in control systems and mechanics. It is my expectation there will be occasional interfacing with the other teams in this project who either affect or are affected by my sub-project, i.e. SCADA or TSI teams for cruise control.

5.9.3. System Level Diagrams

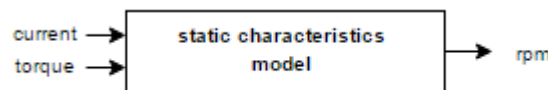


Figure 1. Static Characteristics Model

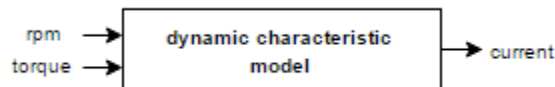


Figure 2 Dynamic Characteristics Model

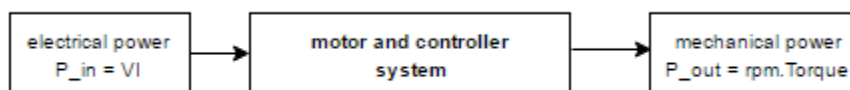
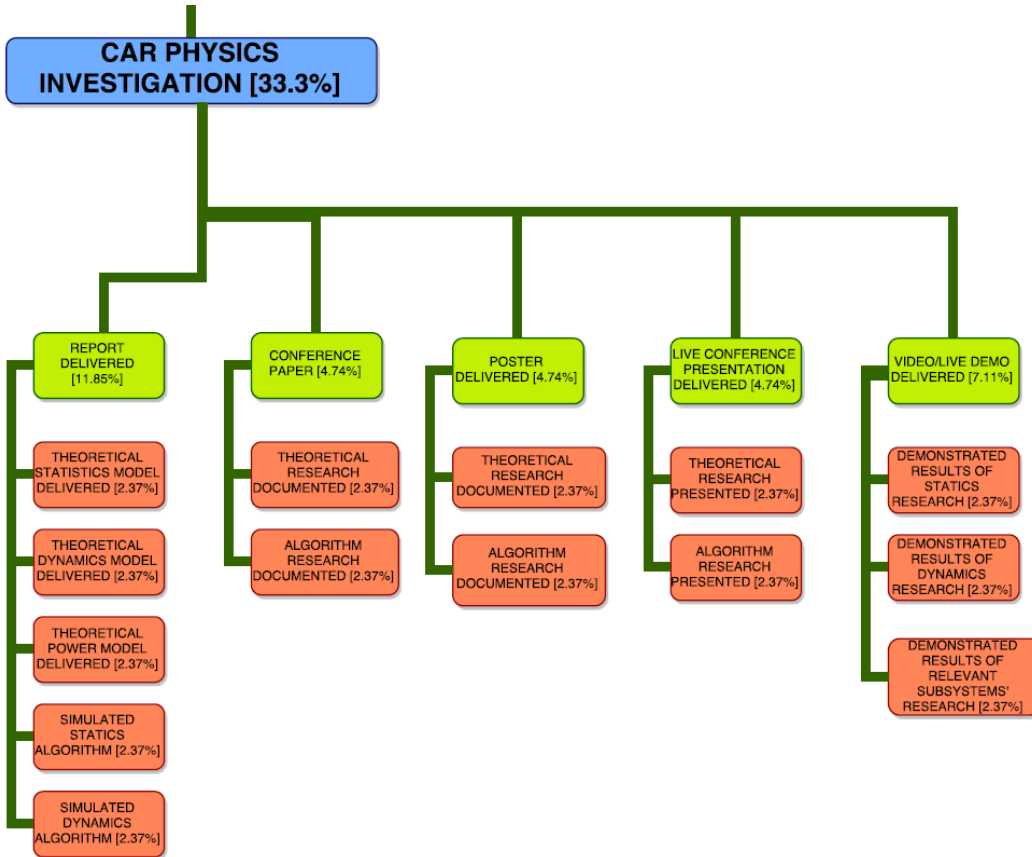


Figure 3 Power relation of motor and controller system

5.9.4. Task Delegation

Task	Group Member	Estimated Time
Understanding previous work done	Zainab	1 week
Go over and find mathematical equations relating any of the modellings needed (see equations listed in 4.9.1)	Zainab	1 week
Write computer algorithm for the modellings from task 2. Probably use Simulink extending previous work. Function based modellings could also use python.	Zainab	1 week
Simulate the written code and collect data results. Iterate at this step for reasonable results	Zainab	1 week
Demonstrate the results of the tests to the professors	Zainab	1 week
Presentation of the final research for Physics model	Zainab	1 week

5.9.5 Work Breakdown Structure



5.10. Cruise Control

5.10.1. Current State of Affairs

In Spring of 2013 there were some approximations done to the effect of elementarily answering questions on the position, speed and acceleration of the car, and the current and voltage requirements. An element of cruise control can be seen in:

- behavior of current during driving
- amount of time the car runs at continuous current, which previously found 47 minutes for 70A (SOC)
- maximum distance travelled by the car
- through examination of quantity of battery cells, the distinct power capability of each cell, the mileage could be calculated (Fuel economy)
- endurance measure through analysis of the car performance and gear ratio

There has been additional work done by the TSV teams, therefore most of the revision and additional and/or new approaches to simulate expected cruise control is necessary. All work in this second phase is contingent on proper modelling of the physics model as cruise control is reliant on both the static and dynamic characteristics of the motor and controller model.

5.10.2. Goals

All work on cruise control is contingent on proper modelling of the physics model, explained in section 4.9, as cruise control is reliant on both the static and dynamic characteristics of the motor and controller model. Some objectives of cruise control that need either new or more work on include:

- Finding the full range of speed in mph to run the car
 - At maximum rpm of motor
 - Varying of gear ratios
- Finding how much the battery lasts while running the car
 - Take to consideration battery capacity and supply current for proper discharge rate estimation
- Finding the optimum speed to run the car
- Finding the maximum acceleration of the car
- Find out to travel a mile:
 - Effect of constant torque on the amount of needed supply current
 - Effect of constant supply current on the amount of needed torque
 - What parameter is more controllable, electrical (current) or mechanical (torque)
- Estimating the fuel economy of the car

5.10.3. Task Delegation

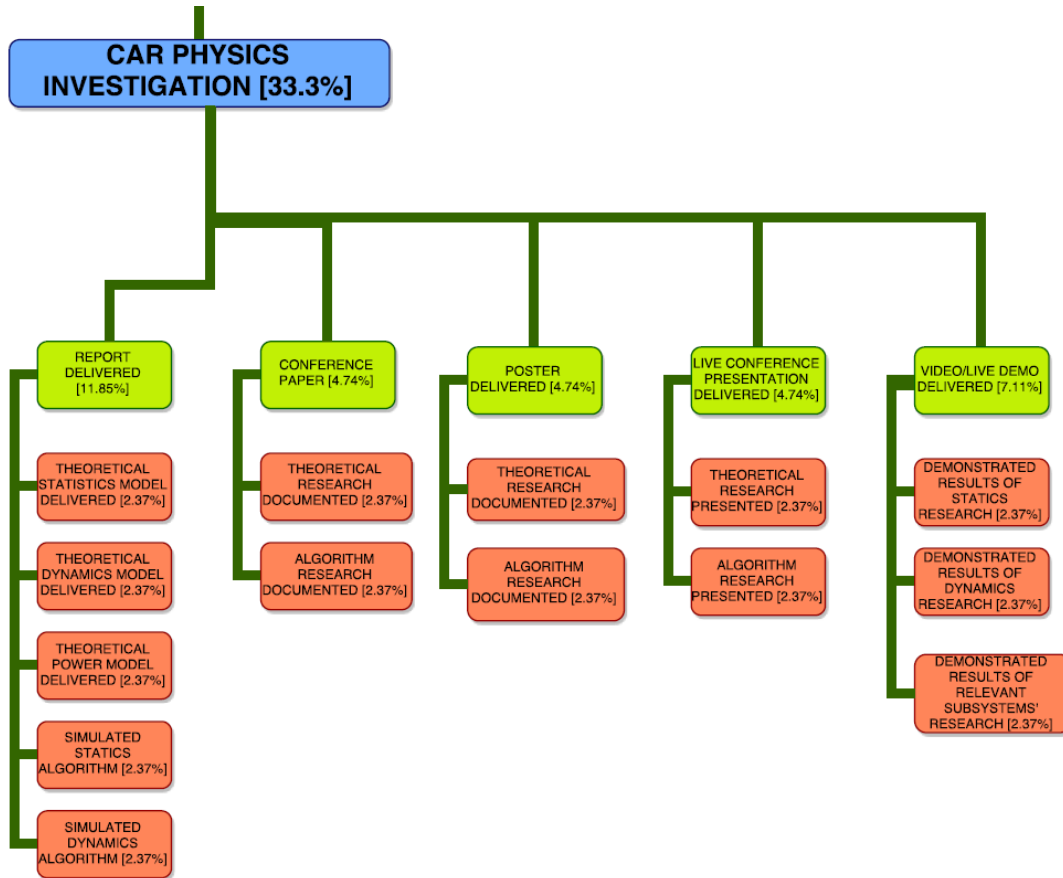
Task	Group Member	Estimated Time
Understanding previous work done	Zainab	1 week
Go over and find mathematical equations relating any of the modellings needed (listed in overview section)	Zainab	1 week
Write computer algorithm for the modellings from task 2. Probably use Simulink extending previous work. Function based modellings could also use python.	Zainab	1 week
Simulate the written code and collect data results. Iterate at this step for reasonable results	Zainab	1 week
Demonstrate the results of the tests to the professors	Zainab	1 week

Presentation of the final research for 1. Physics model and 2. Cruise control	Zainab	1 week
-------------------------------------------------------------------------------	--------	--------

5.10.4. Individual Acceptance Test Strategy

Phase	Test description	PASS/FAIL
Cruise control	<p>For a given load (throttle)</p> <ol style="list-style-type: none"> 1. Compare the optimum speed from algorithm test and the expected optimum speed are close 2. Compare the expected and resulting full range of speed in mph to run the car are close 3. Compare expected and resulting amount by which the battery lasts while running the car are close 4. Compare expected and resulting optimum speed to run the car are close 5. Compare expected and resulting maximum acceleration of the car are close <p>Test different percentages of the load reflect proportionately to the resulting speed of the car</p>	

5.10.5 Work Breakdown Structure



5.11. TSI [Tractive System Interface]

5.11.1. Current State of Affairs

At the present, the JGB will be used for brake and throttle logic, as well as relays for other functionality throughout the TSI. Tim Andrews made a PCB that functioned as both a current and voltage monitor. A CAN bus isolator has been purchased that will be used to communicate between the motor controller and our JGB (as well as SCADA, presumably). The proper IMD has already been purchased. An algorithm for the Drive button and relevant safe start conditions (throttle position, etc.) was written in a previous year.

5.11.2. Goals

Brakes are mechanical, but need to send a signal to the TSI when compressed. There will be a brake overtravel switch connected to the safety loop. In the event that the drive compresses the brake beyond normal expected travel distance (e.g. there is no brake fluid), the brake overtravel switch will open the safety loop (this is not driver

resettable). The accelerator pedal will have a two sensor plausibility check system following the rules defined in section EV2.3 of the Formula Electric rules. If an implausibility is detected, the JGB will switch the system back to the Not Ready to Drive state by the motor controller not providing voltage to the motor itself. The JGB will control most of the interface with the motor controller.

Brake Light will be controlled with a relay on the JGB, based upon information given by the brake position sensor (potentiometer or pressure sensor, dependent upon collaboration with MechE discussed later). The RTDS will be controlled by the JGB based upon information given via CAN bus from the motor controller, as well as the OFF/AUTO key. The FWD/OFF/REV switch will be an On-On-On locking key switch. This will be located out of reach of the driver, to ensure that the vehicle will not be switched from forward to reverse while moving. The FWD and REV position will place the motor controller in the respective drive position, with OFF breaking the Safety Loop. The Drive button will be interfaced with the JGB, and will be used to ensure that the vehicle does not move unexpectedly. An algorithm was written in previous years, checking throttle position when this button is pressed. If the algorithm finds that the vehicle is not in a safe state to drive, no throttle signal will be transmitted to the motor controller. The TSAL needs to be implemented. We plan on accomplishing this by using an isolating DC/DC converter that will turn on when more than 16V are present on the TSV, supplying a 5V signal used to turn on an LED for viewing in the Cockpit. This conforms to rules set forth in the Formula Hybrid (60V) as well as Formula Electric (32V) Guidelines.

The necessary IMD has been purchased (Bender21 A-ISOMETER @ iso-F1 IR155-3203) and will be interfaced with the TSV, with the one-bit output used to control a relay within the Safety Loop. Off/Auto precharge relay key switch controls if the MC precharger is connected to the MC. If it is in the Off position, the TSV can be active, but the MC will not be powered. In the auto position, when the TSV become active, the MC will automatically begin precharging. We plan to maintain galvanic isolation at all times through the use of properly-rated opto-isolators, isolating DC/DC converters, and isolated relays. Should other methods of transmission between HV and GLV be needed, team members will discuss options and confirm their validity with the course instructor. The IMD purchased is stated in the rules and is rated for use in the car. To protect from humidity and rain, any location where the TSI enclosure could be prone to failure (corners, sockets, etc.) gaskets or silicon bonding can be used to seal the components. Loctite will be used on all mechanical fasteners (nuts, bolts) to help prevent against vibrations. Galvanic isolation will be maintained to reduce the risk of shock and sufficient clearance to all high voltage parts to the enclosure will help reduce the risk of accidental contact.

Communication with SCADA will be accomplished over CAN bus, as implemented in the JGB. Additional indicators, such as lights required by the Formula Electric and Formula Hybrid rules are discussed earlier. Throttle information from SCADA will be received via CAN bus in the JGB. The JGB will be programmable for Dyno and vehicular control, where the PWM output for the throttle can be based upon input from SCADA or ADC inputs from the potentiometers.

5.11.3. System Level Diagrams

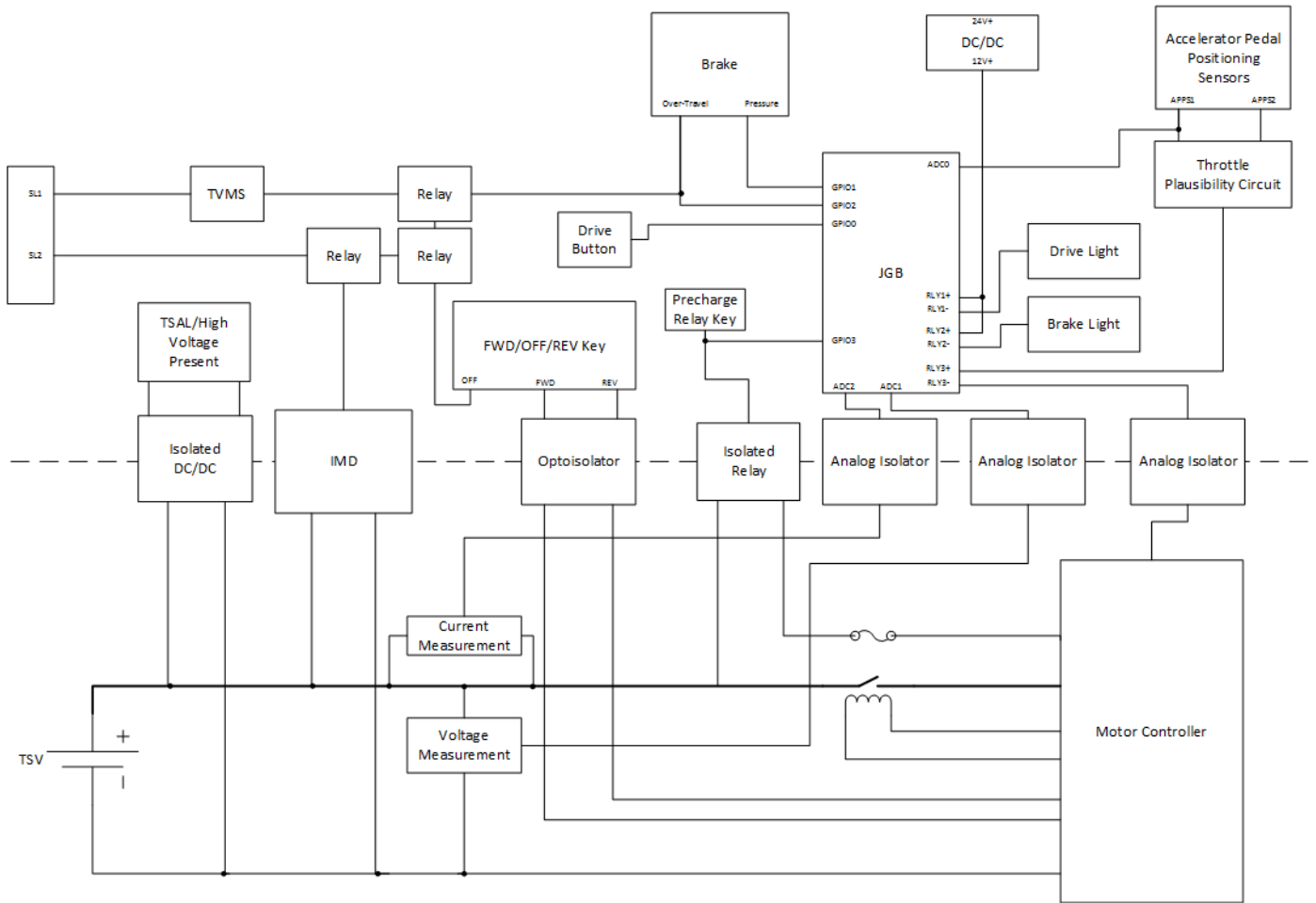


Figure 1. TSI System Overview

Shutdown System State Machine
Jack Plumb

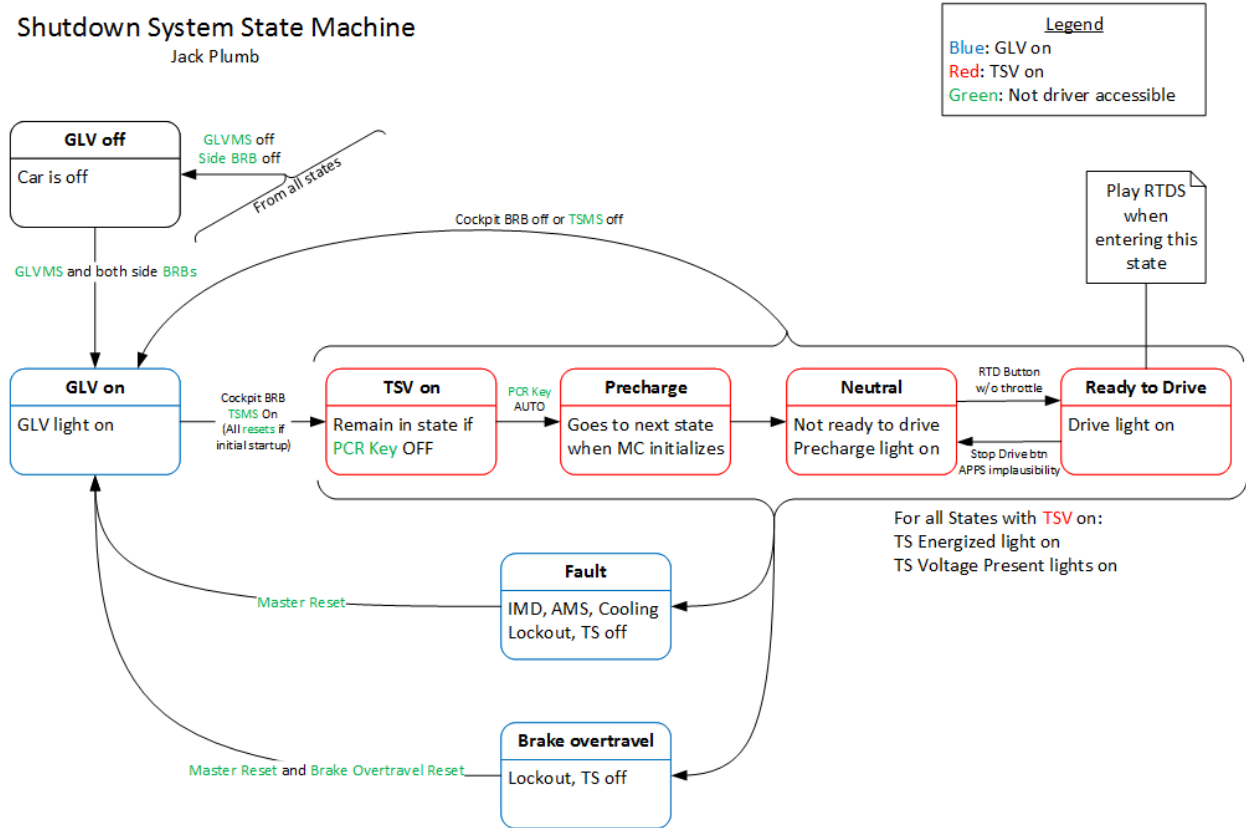


Figure 2. System Shutdown State Transition Diagram.

Some of this system falls on GLV, but here we are only explaining the part that is controlled by the TSI. This begins when the user attempts to turn on the TSV. For this to happen, the TSMS needs to be on, and the Cockpit BRB needs to be out (the master reset button may also need to be pushed). The next state is the TSV on state, where the safety loop is closed and the TSV is on, but the motor controller is still unpowered. If the Precharge Relay key is in the OFF position, the system will stay in this state. If it is in the AUTO position, the system will move to the Precharge state, which will precharge (not the precharge relay). This will power the motor controller, which will initialize and close the precharge relay. Once the MC indicates the precharge relay is closed, the system will shift to the Neutral state. In this state, the MC will be powered, but the throttle will not work. The driver then presses the Ready to Drive button to switch to the Ready to Drive state. If the system detects that the accelerator is pressed, the Ready to Drive button will not work. When the system enters the Ready to Drive state, the Ready to Drive Sound will play. In the Ready to Drive state, the throttle will be engaged and the car will be fully operational. If the driver holds the Ready to Drive button, or the systems detects APPS implausibility, the system will move back to the Neutral state. If, at any time, there is a fault (IMD, AMS, cooling, break overtravel) the TSV will be shut off and the system will need to be reset (not driver accessible).

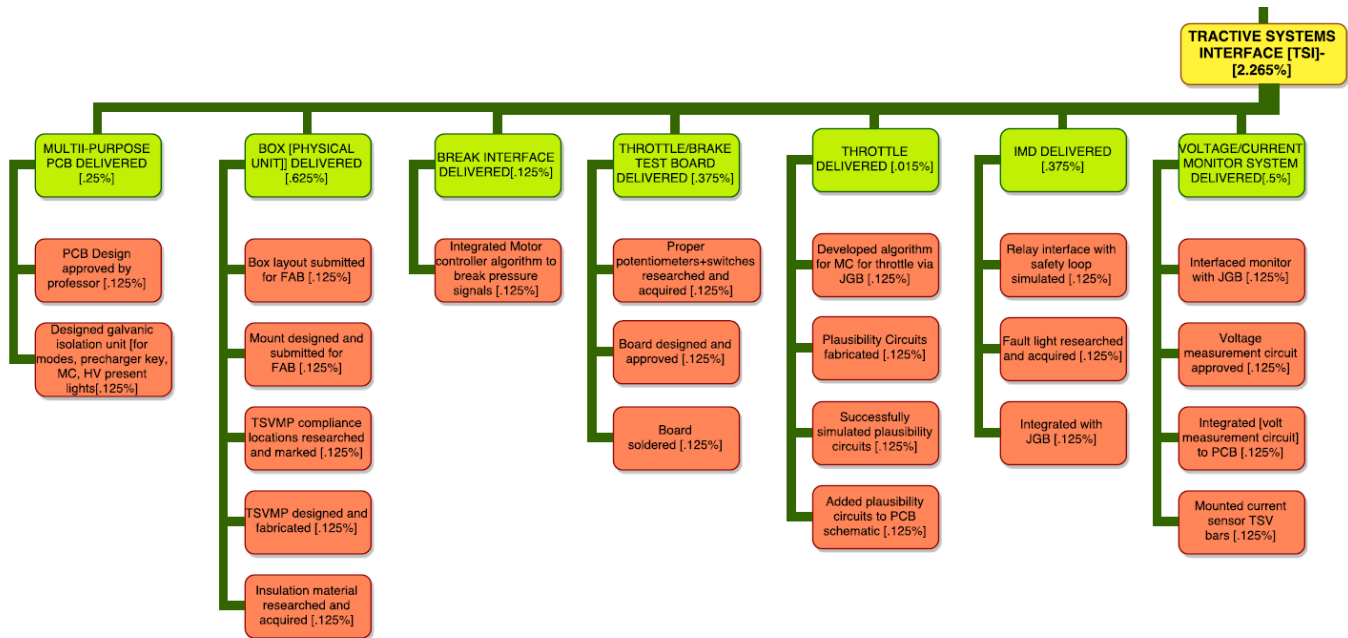
5.11.4. Task Delegation

Task	Group Member	Estimated Time
Install Brake Over Travel in Safety Loop	Jack	1 Day + MechE Install
Interface with APPS for Throttle	Adam	3 Days
Interface JGB with Motor Controller	Adam	1 Week
Interface JGB with Brake Potentiometer for Brake Light	Christer	1 Day
Connect RTDS with SCADA and JGB	Adam	2 Days
Connect Fwd/Off/Rev switch and interface with JGB	Jack	3 Days
Interface Drive button with JGB to ensure no false starts	Adam	2 Days
TSAL receives power through DC/DC Converter	Jack	2 Days + Finding Part
Connect IMD to TSI	Christer	2 Days + New PCB Design
Install Off/Auto switch for Pre-Charge Relay	Jack	1 Day
Galvanically Isolate all parts	All	Throughout Project
Keep components up to Vibration/Rain/Shock standards	Christer	Throughout Project
Connect to SCADA via CAN	Adam	1 Week
Throttle Algorithm	Christer	1 Week

5.11.5. Requirements Checklist

Component	Tasks/Deliverables/Desired Outcomes	Engineer
R005a. TSI	[a1]meets requirements from Curtis motor controller [a2]meets requirements from F EV rules [a3]packaged to be fully compliant with F EV rules must be easy to integrate w/ car [a4]GLV/tsv suitable for car or design/fabricate/test MCS dyno test stand [a5]interfaced with SCADA using CAN bus [a6]actively/dynamically keep a log of useful parameters voltage/current/state of insulation	[5a] All [a1] Adam [a2] All [a3] All [a4] GLV/TSV [a5] Adam [a6] Christer
R005b. Throttle and Brake	[b1]galvanically isolated throttled interface circuit b/w motor controller to VSCADA [b2] integrated with manual throttle, complaint with FEV rules	[5b] Adam and Jack [b1] Adam [b2] Jack
R005c. Galvanic Isolation	[c1]display resistance of insulation monitor	[5c] Christer and Adam [c1] Christer and Adam
R005d. Controls	[d1]OFF/AUTO switch [d2]FWD/NEUT/REV switches [d3]cover for d2	[5d]Jack [d1]Jack [d2]Jack [d3]MechE

5.11.6. Work Breakdown Structure



5.11.7. Individual Acceptance Test Strategy

The main units we plan on testing are the break, the throttle (safety), and lights (all). Specifically, the break will be tested to ensure that the brake pedal is depressed, and that when the Brake-Over-Travel switch actuated the Safety Loop broken. The throttle will use two potentiometers for pedal positioning and verify that they are within 10% of expected value (plausibility). The lights will be TSAL Active when 30V is present in system (as per competition rules). Brake lights will be active when brake is depressed, as well as IMD Fault Light visible in cockpit. We will be observing specified measuring points (such as banana plugs for TSMP). The TSMP will read 0V when either the Brake Overtravel or APPS systems detect a failure in brake and throttle systems respectively. The testing tools we will be using are Potentiometers to test pedals (more details in rules), resistors used to test IMD, as well as a voltmeter to ensure TSV cut when safety loop is triggered.

6. Overall Acceptance Test Strategy

The main goal of the test strategy is to prove that all systems can integrate with each other. To perform this functionality we will build the dynamometer room to match the car as closely as possible. Each team has detailed how they intend to effectively and completely test their systems. One item that needs to be enforced is that people will need to be able to justify numerical values. This will involve error and certainty analysis. To be able to test the car as a whole every single part will be integrated into the dynamometer room. This will allow teams to try to integrate as quickly as possible as soon as they have completed their relevant subsystems.

7. Maintainability Plan

We propose that our efforts this year will be maintainable due to the following reasons: any and all code will be stored in a centralized GitHub project containing a repository for each subsystem team. By using GitHub we do not have accounts tied to our Lafayette emails. This means that the repository will still exist after we graduate. More importantly, we will be able to hand control over to next year much more easily than previous years. We will also be storing any completed and approved fully-commented code and documentation on a Google Drive. Furthermore, we will be including the latest version of all documents and deliverables to the website on a weekly basis.

8. Cost Analysis

The proposed budget for the 2017 LFEV can be found in the table below. This budget was created by considering both known big budget items and items that are currently unknown but will need to be purchased later in the semester. There has been \$500 allocated to all shipping fees and taxes, with consideration for rush shipping fees for immediate need. If this is not fully depleted by purchase orders, any remaining funds can be used by teams who have spent all of their allocated budget but still need more parts.

Subsystem / Purpose	Allocated Budget
TSI	\$400
GLV	\$700
VSCADA	\$50
Cell App	\$125
Controller Cooling System	\$500
Interconnect / Cabling / ICD	\$400
DYNO	\$50
TSV	\$600
Physics & Cruise Control	\$0
JGB	\$300
Shipping / Taxes / Miscellaneous	\$500
Total	\$3,625