Final Report

ECE 492 - Spring 2019

Latest Revision: May 12, 2019
Prepared by: Alex Kmetz and Katie Lee

Abstract

This document summarizes the work done by the 2019 team during the Spring 2019 semester.

Table of Contents

Abbreviations

AIR	Accumulator Isolation Relay
BRB(L/R S)	Big Red Button (Left/Right Side)
CellMan	Cell Manager
GLV	Grounded Low Voltage
GLVMS	Grounded Low Voltage Master Switch
HV	High Voltage
IMD	Isolation Monitoring Device
PackMan	Pack Manager
SegMan	Segment Manager
SSOK	Safety System OK
TSAL/TSEL	Tractive System Active/Energized Light
TSI	Tractive System Interface
TSMP	Tractive System Measuring Point
TSV	Tractive System Voltage
TSVMS	Tractive System Master Switch
VSCADA	Vehicle Supervisory Control and Data Acquisition

System Design

System Overview

The goal for this semester was to have a motor spinning in the dyno room. The motor was powered by a power supply while battery packs were designed for fabrication in the Spring semester. The major subsystems are:

1. TSV - Responsible for providing the power source for the high voltage required by the motor.
2. TSI - Responsible for providing a safe interface between the TSV and the motor controller, with inputs from the GLV and pedals.
3. GLV - Responsible for providing the low voltage that operates the safety loop and other low voltage systems.
4. Motor Controller - Responsible for accepting inputs from the throttle and high voltage and providing control signals to the motor.
5. Motor
6. VSCADA - Responsible for reading data from sensors and saving that data to a file. The SCADA should also display data meaningfully.
7. Interconnect Cabling - Responsible for providing data transfer and power between each major subsystem.
8. Cooling System - Responsible for ensuring the motor controller operates at a safe temperature.

Overall Schematic

The overall DYNO wiring Diagram can be found here:
https://sites.lafayette.edu/motorsports/files/2019/05/ATP_2019_v1.1.0.pdf

Tractive System Voltage (TSV)

Overview

This year the accumulator design was completely changed from previous years in order to be rules compliant. In the future, a new battery state of charge and active balancing algorithm will be implemented. This semester we focused on developing and testing the PCBs as well as manufacturing the mechanical enclosure for the packs.

CellMan

Each battery cell has a Cell Manager (CellMan) board monitoring it. The CellMan measures individual cell temperature and voltage and communicates this information to the SegMan via a custom communication protocol. Eventually, the CellMan was also receive information from the SegMan on whether to discharge a cell or not. When discharging, the CellMan will intercept the incoming current and divert it back to the rest of the cells (refer to the high-level wiring diagram) to actively balance the cells.

SegMan

Each string of seven cells is called a segment and has a Segment Manager (SegMan) board. The SegMan collects individual cell data from the CellMan boards. The SegMan uses ISO-SPI protocol to communicate with the PacMan. The SegMan itself does not have a processor but does have digital logic through the LTC6804-1. This digital logic will allow the SegMan to store relevant information about the expected cell performance (provided by the PacMan) in memory. Then, the SegMan will use the pin S_n to transmit a signal to enable/disable individual cell discharging to actively balance a segment.

The SegMan is powered by a segment of cells (7 cells / ~24V) by directly connecting to the positive and negative terminals of a segment. This connection is fused to avoid overpowering the LTC6804-1. The SegMan has the ISO-SPI chip LTC6820 which is rated at 3.3V for communication. The segment voltage on the SegMan needs to go through a DC-DC converter to get 3.3 V.

PackMan

Each pack has fourteen cells in series (2 segments). Each pack will have a Pack Manager (PackMan) board. The firmware of the PacMan determines the thresholds of over/under voltage and over temperature as well as state of charge parameters. These parameters can be configured using the pushbuttons and the LCD screen on the packs. The PacMan utilizes an Isolated Serial Peripheral Interface (ISO-SPI) to communicate with the SegMan to actively balance the cells in each segment as well as obtain cell information. PackMan can use cell information to determine when to continue charging a cell or when to stop charging a cell. Additionally, the PacMan will be able to trip the safety loop if a cell voltage is too high or low, or if a cell temperature is too high. Furthermore, it will use a CAN interface to communicate cell information to SCADA. The firmware will be further developed next semester.

Tractive System Interface (TSI)

Hardware

The updated internal wiring diagram can be found here:
https://sites.lafayette.edu/motorsports/files/2019/05/TSI-wiring-diagram_0505-2.pdf

Board

The board was redesigned this semester to address the bugs from the Fall 2018 semester board.

Brakes

The brake system has remained unchanged from last year.

Throttle

The throttle potentiometers are simulated with turn potentiometers on the TSI testing board. There is a 5 KΩ potentiometer to control APPS1 and APPS2 at the same time as well as two 10 KΩ potentiometers to control APPS1 and APPS2 individually. The 10 KΩ pots can be adjusted to make throttle plausible/implausible.

IMD

The Bender ISOMETER IR155-3204 is used for the IMD. The IMD is hooked up directly to the safety loop with a relay so that when an IMD fault occurs the safety loop opens. The IMD is hooked up to the HV+ and HV- coming into the TSI as well as 24 V power from the GLV. The High Side Status Output

(OK_{HS}) signal is put through the safety loop relay and the PWM output is an input into the TSI microcontroller.

TSAL

The TSAL has remained unchanged from last year.

TSMP

There are tractive system measuring points on the top of the PCB. In the Throttle Plausibility section, there's APPS1_b1 and APPS2_b1 for biased APPS1 and APP2 input; APPS1_ISO for after the APPS1 input has been stepped up to 5-10 volts; and APPS_DIFF to check the voltage differential between APPS1 and APP2. In the AIRs and IMD Interface section, the only measuring point is IMD_UC1. In the Status Lights section, there is GLV_GND1 and HVPL_LV1/TSAL_LV. In the DC-to-DC section, there are measuring points for HVPL_LV1 and HV_GND1. Finally, the Pre-charge/discharge section has a measuring point for Throttle_HV.

Test Board

A test board was developed this semester to debug the TSI PCB as well as conduct all low voltage QA testing.

Firmware

Another big part of the TSI is the firmware that the microcontroller on the board will be running. This firmware is in charge of three major features of the TSI which are system measurements (voltage, current, temperature, and IMD status), CANBus communication with the VSCADA, and the drive states implemented in the car. The microcontroller in the board is the Microchip PIC32MZ2048EFH064.

The current version of the firmware can be found at:
sites.lafayette.edu/motorsports/tsi

System Measurements

Voltage

The voltage is measured with an ADC. The signal V_measure is an input to the microcontroller which converts it to a digital signal. This signal is then sent to VSCADA over CAN.

Current

The current sensor is two Hall Effect sensors that are placed on either side of an aluminum bar.

Temperature

An LM95071-Q1 Temperature Sensor interfaces with the microcontroller via SPI to provide temperature information from the TSI.

IMD Status

The Bender ISOMETER IR155-3204 outputs a PWM signal that will then be sent through a low pass filter into the microcontroller. This signal is the measured resistance between high voltage and ground and will be especially important when the safety loop opens as it will inform us if the fault was due to a signal from the IMD.

The ISOMETER outputs a PWM signal that is sent through a low-pass filter. The output of the low-pass filter is then sent into the microcontroller as an analog voltage which is converted to a digital signal by an ADC and sent to VSCADA over CAN.

The analog voltage value of the PWM signal is equal to the amplitude of the PWM multiplied by its duty cycle. Per the data sheet of the ISOMETER, a duty cycle of 90% or greater is considered an IMD fault.

This semester, the ISOMETER IR155-3203 was used because the 3204 was on backorder. However, the PCB was designed for the 3204 so this is the IMD that will be used next semester. The 3203 is a logic low whereas the 3204 is a logic high but otherwise the IMDs are the same. In order to make the 3203 work for this semester, we used a pull up resistor.

CANBus Communication

The TSI is able to send bytes of information to VSCADA. Currently it will be sending voltage, current, and temperature sensor measurements, IMD status (the PWM low-pass filtered output of the IMD), and drive state. These are all sent at a regular interval.

The TSI receives two signals from VSCADA: cooling enable/disable and configuration parameters.

Drive States

A drive state machine is implemented in the TSI firmware to control the various states the car can be in. Currently, the states that are implemented are IDLE, PRECHARGE, DRIVE_SETUP, DRIVE, and OVERCURRENT.

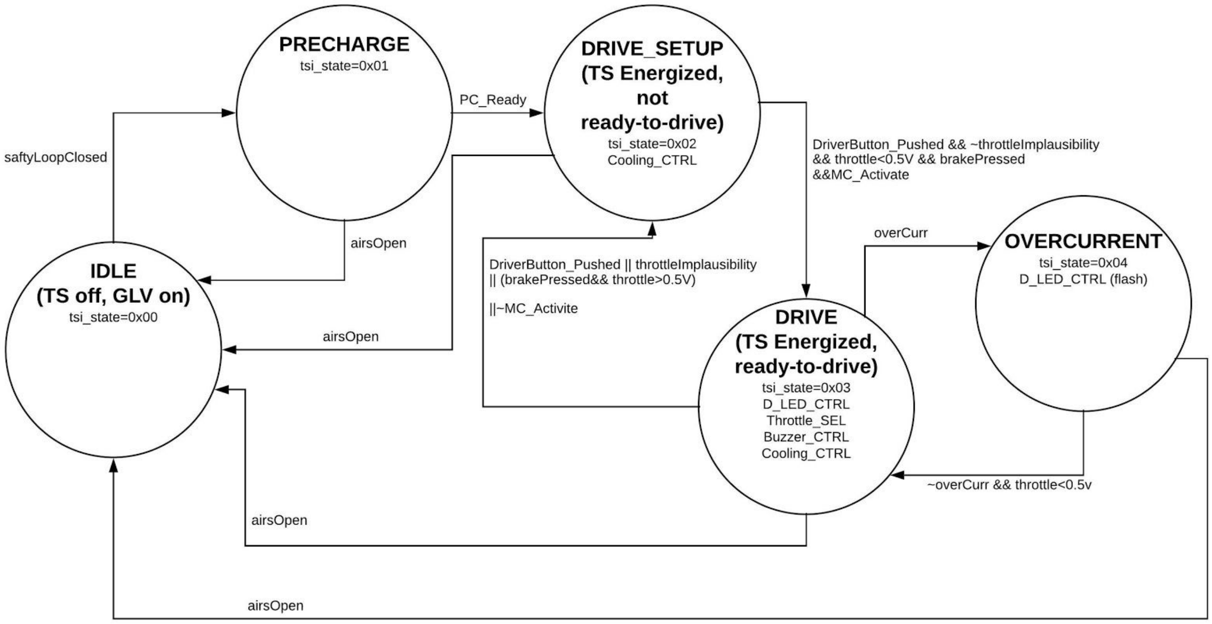

Figure 1
State Diagram of the TSI Drive States

The car starts in the IDLE state. In this state, the throttle is disabled and the car cannot move. When the safety loop is closed, the car will transition to the PRECHARGE state. The state machine will wait here until the precharge circuit has reached 90% of its charge. At this point it will automatically transition to the DRIVE_SETUP state. To move from the DRIVE_SETUP state to the DRIVE state, all of the following conditions must be met: drive button pushed, brake pressed, throttle is plausible and not pressed (less than 0.5 V), and the motor controller is fully charged. Once in the drive state, a Ready-to-Drive sound will be played as per EV9.2. The system can return to the DRIVE_SETUP state if any one of the following conditions are met: drive button is pushed, throttle is implausible, brake and throttle are both pressed, or motor controller is turned off. The system can also transition from the DRIVE state to the OVERCURRENT state if the current is too high. The OVERCURRENT threshold is set by SCADA. The system can transition back into DRIVE mode if the current returns to a safe level and the throttle is not pressed. If the system is ever in any state other than IDLE and the AIRs open, the system will return to the IDLE state.

Grounded Low Voltage (GLV)

Overview

The Grounded Low Voltage system provides 24 V power to low voltage subsystems on the vehicle, controls the safety loop, and monitors the vehicle status through the VSCADA.

Design

The design of the GLV remains largely unchanged from last year. Major components on the GLV are functional and tested with the motor. One of the major changes is from screw terminals to Molex connectors. This has made the GLV easier to wire correctly. Additionally, a Pi-to-CAN shield was designed and manufactured to free up the GPIO pins on the Raspberry Pi that were previously covered but unused by the Pi-CAN shield.

Safety Loop

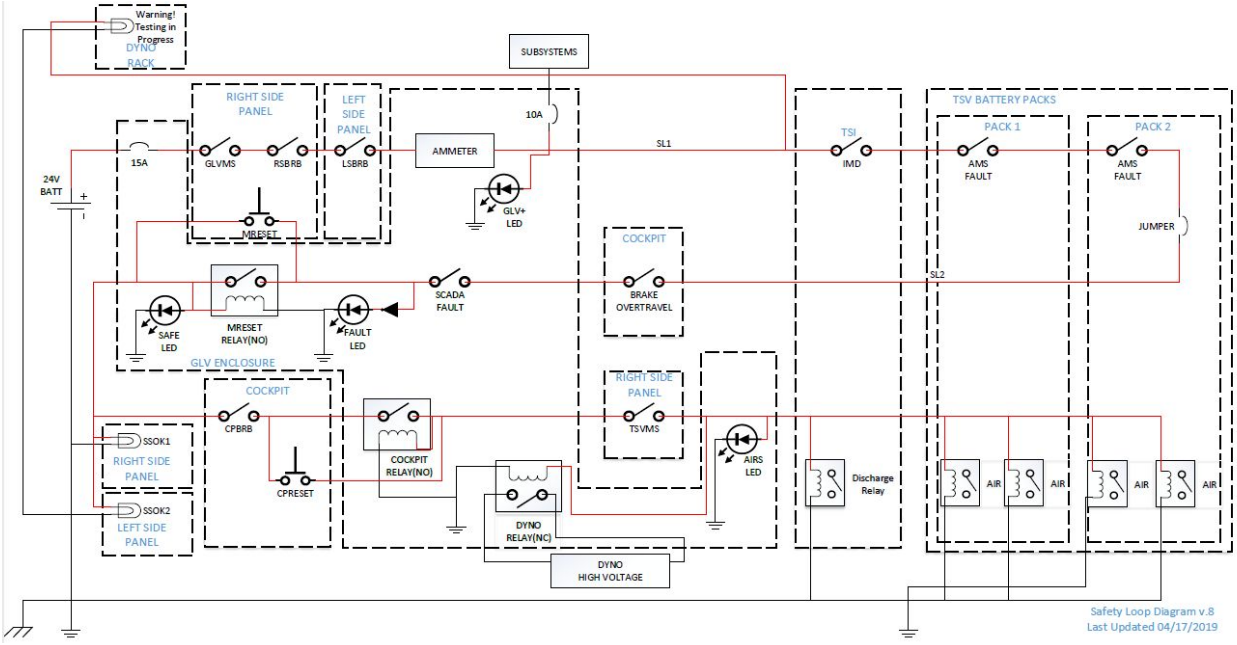

Figure 2
Safety Loop

The safety loop is a crucial part of the system that ensures the system is operating safely whenever high voltage is present.

The procedure to be followed in order to close the safety loop and allow high voltage to be permitted throughout the vehicle is as follows:

1. Turn GLVMS
    a. GLV Light Active
2. Press Master Reset Button (MReset)
    a. SSOK Lamps Active
    b. Safety Light Active
    c. Closes MRESET Relay
3. Press Cockpit Reset Button
    a. Closes Cockpit Relay

4. Turn TSVMS
    a. AIRS Light Active
    b. TSEL Active
    c. Close AIRs

Communication

I2C

The GLV communicates with the SCADA using I2C protocol. The GLV reports temperature, voltage, current and power sensor information to the SCADA.

GPIO

Raspberry Pi GPIO pins communicate with a Real Time Clock (RTC) and a CAN interface shield. The GLV communicates GLV Faults and the SCADA outputs a control signal to the SCADA safety loop relay through the GPIO pins.

CAN

The GLV does not communicate through CANBus but all CAN messages pass to SCADA through GLV.

Vehicle Supervisory Control and Data Acquisition (VSCADA)

Overview

The SCADA software is responsible for storing, monitoring, and displaying data from subsystem sensors. This data will be sent to the SCADA via CANBus. The exception to this is GLV, which communicates with SCADA using I2C protocol and the Raspberry Pi GPIO pins. The packets received from the CANBus will be parsed out into meaningful, human readable information. That data is then checked against the thresholds set in the config file. If the data is outside of the allowed thresholds it is flagged and logged in the log file. The user can also specify in the config file for the SCADA to open the safety loop through the SCADA relay if a data point is outside of the expected range. Data is stored in the database so it will be available for post processing. This year's SCADA was rewritten from the ground up in C++ and runs on a Raspberry Pi inside of the car.

CANBus Interface

The SCADA communicates with other subsystems primarily with CANBus through the Raspberry Pi GPIO pins and a Pi-to-CAN shield.

Database

The SQLite Database is used to save raw data, calibrated data and other system information. Values are recorded with an internal timestamp, sourced from a timer that's started at program boot. There's also a sensor table that contains all configured properties of the available sensors. This table is utilized during post-processing. The system info table contains general system information such as run start time and end time. The user is able to save or discard collected data and is prompted to provide a name to save under.

Displays

The Rack Display is connected to the Raspberry Pi for displaying the UI in Dyno Room. This display is especially bigger than the one that will be installed in the car, making it easier to view data collected and processed by VSCADA.

Interconnect Cabling

Overview

Interconnect cabling is responsible for providing data transfer and power between the major subsystems. Interconnect is crucial for integrating each subsystem into the Dyno room and later the car. These cables must be safe, manageable, rules complaint and durable so they will survive when the car is in motion or other forces are acting on them. This year's interconnect team focused on cable organization, rules compliance, and making wiring as fool-proof as possible. We are using the cables listed on the following interconnectivity document:
https://sites.lafayette.edu/motorsports/interconnect/

Cooling System

Overview

The cooling system is responsible for keeping the motor controller at a safe temperature. The system runs off of 24 V from the GLV.

Design

The cooling system kept a similar design to last year. The system receives 24 V from GLV. The Koolance fan from last year is still being used, but the cooling controller was eliminated from the design in order to simplify the overall system. Now, cooling is turned on whenever the motor controller is active. This is controlled by a relay which receives a control signal from SCADA over CANBus. In return, Cooling sends temperature and flow rate data to SCADA over CANBus.

Dyno Software

Overview

VSCADA is used in order to collect data from and monitor the Dyno room during testing. The software is capable of collecting sensor data from the motor, motor controller, TSI, and GLV while in the

Management

Work Breakdown Structure

The link below is to the work breakdown structure, updated as of May 6, 2019.
https://sites.lafayette.edu/motorsports/files/2019/05/Week_14_CAR_Progress.pdf

Budget

The following pie chart represents the allocation of the total budget, **$34,858.80,** among all subsystems as of May 07, 2019.

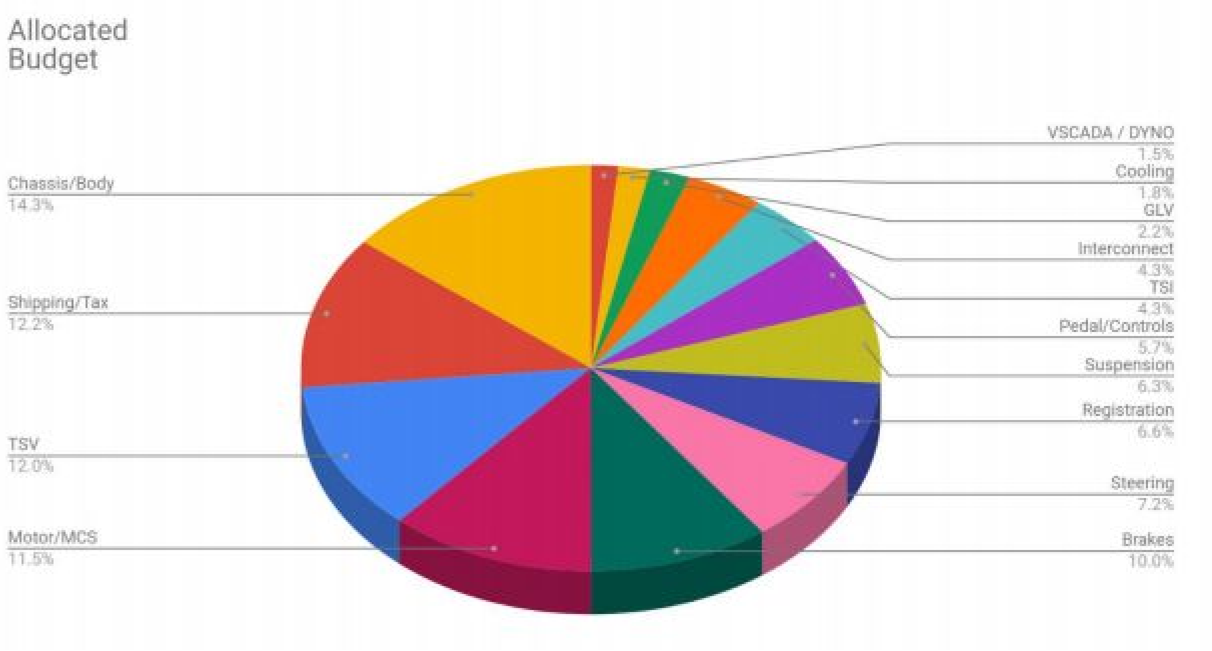

Figure 3
Final Budget Allocation as of 5/07/2019

Table 1 below indicates that we have spent **$36,653.27** which is **105.15% of** our allocated budget **$34,858.80.** We have also indicated subsystems that have overflown their allocated budget as well as subsystems who have not made any purchases as of yet. In addition, figure 4 shows the total expenses versus time in weeks.

Sub-system	Allocated Budget	Total Spent	Budget Remaining	Percentage Spent
Pedal/Controls	$2,000	$0.00	$2,000.00	0.00%
Cooling	$620	$112.59	$507.41	18.16%
Brakes	$3,500	$122.03	$3,377.97	3.49%
VSCADA / DYNO	$525	$484.02	$40.98	92.19%
Suspension	$2,200	$954.30	$1,245.70	43.38%
Steering	$2,500	$1,174.86	$1,325.14	46.99%
GLV	$780	$1,242.28	**-$462.28**	**159.27%**
Shipping/Tax	$4,247	$1,567.49	$2,679.31	36.91%

Interconnect	$1,500	$1,618.41	**-$118.41**	**107.89%**
Registration	$2,300	$2,300.00	$0.00	100.00%
TSI	$1,500	$2,961.97	**-$1,461.97**	**197.46%**
Drivetrain	$0	$3,309.05	**-$3,309.05**	
TSV	$4,187	$3,524.40	$662.60	84.17%
Motor/MCS	$4,000	$6,781.02	**-$2,781.02**	**169.53%**
Chassis/Body	$5,000	$10,500.85	**-$5,500.85**	**210.02%**

Table 1
Final Budget Status as of 5/07/2019

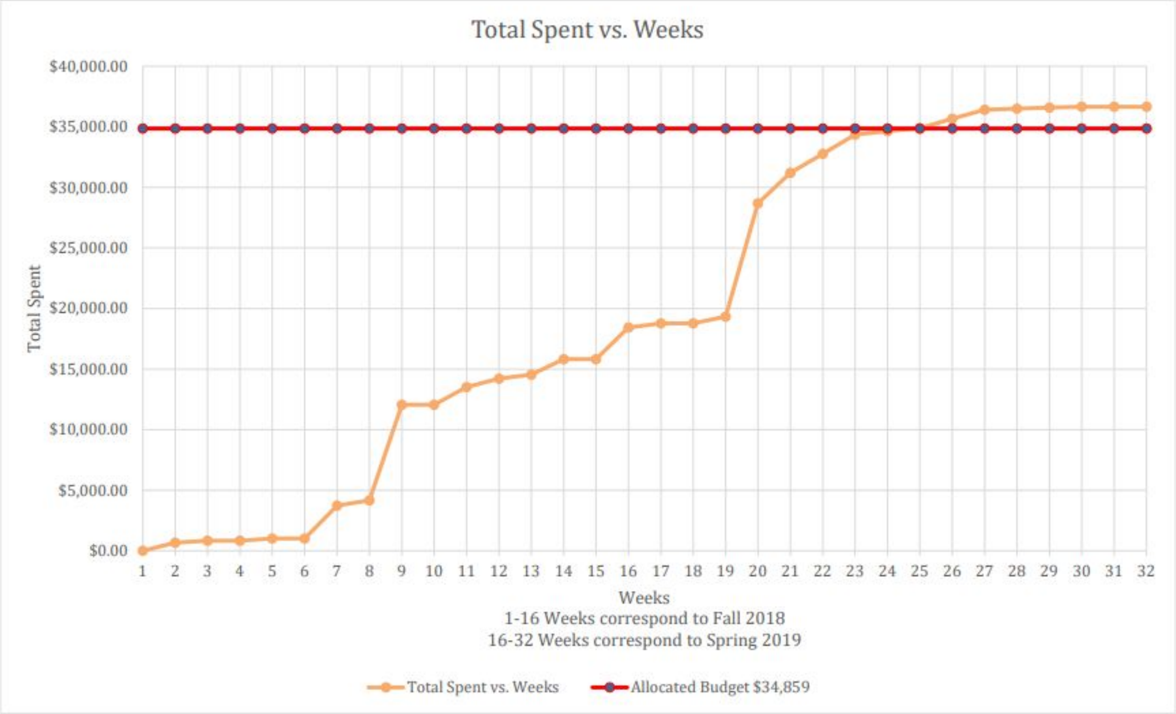

Figure 4
Total expenditure vs. Time in weeks as of 5/07/2019

We have made every attempt to ensure that the above financial documentations are complete representation of what we have spent as of 5/7/2019, however the budget allocation might be subject to change in the future. For further information, please refer to the Final Purchasing Report 5/07/2019 or visit https://sites.lafayette.edu/motorsports/finance/ .