Software Design Document

for

Baja SAE Drivetrain Optimization

Version 2.0 approved

Prepared by Krystal Bernal,

Edit Vosganians, Roi Chico,

Bowen Waugh, and Huang Xu

Cal State LA Baja SAE

3/22/2020

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Initial Submission	**11/28/19**	**First Draft**	**v 1.0**
Final Submission	**3/22/20**	**Final Draft**	**v 2.0**

1. Introduction

This SRS document is intended for the Baja SAE Drivetrain Optimization project.

1.1 Purpose

The purpose of this software is to assist the Baja SAE student-run team in optimizing the drivetrain of the competition vehicle. The software functions on two sets of RPI's, versions 3 and 4, running on Python 3.7. The goal is to be able to determine the ideal configuration of bolts and springs of the CVT's based on sensor readings and analysis on the gathered data.

1.2 Document Conventions

Bold - Section or Table Headings

Italics with Underline - Used for terms defined in the Glossary

Italics - Figure name

1.3 Intended Audience and Reading Suggestions

This document is intended for use by the Cal State LA Baja SAE team members and faculty advisors as they use the Drivetrain Optimization data acquisition system. More specifically, the Electrical and Drivetrain subsections of the team may use this document to understand and further develop an analysis methodology for optimizing the vehicle and presenting results to competition judges.

1.4 System Overview

The main function of the Drivetrain Optimization system is to provide feedback about the Cal State LA Baja SAE vehicle drivetrain configuration (Figure 1). The drivetrain for the Baja vehicle includes the engine, Continuously Variable Transmission (CVT), and the Gearbox. The part of the drivetrain our project focuses on is the CVT. The CVT consists of a Primary and Secondary, both of which are rotating mechanical parts connected by a belt. Inside of the Primary are weights and springs that can be changed to be heavier or stiffer, respectively. Inside of the Secondary is a return spring that can also be substituted for one with a different stiffness.

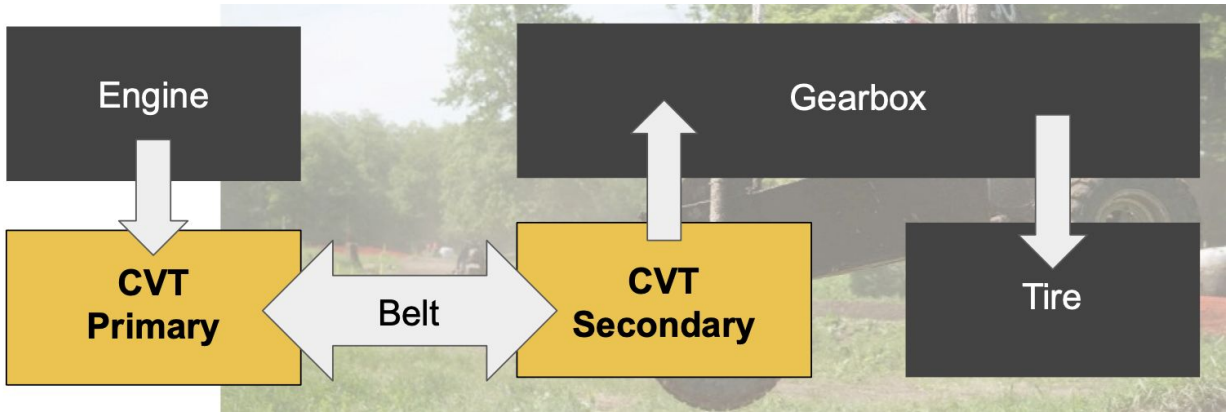


Figure 1: Components of Cal State LA Baja SAE drivetrain

The configuration of the weights and springs inside the Primary and Secondary affects the ratio of their rotational speeds throughout a course. Exploring how these ratios change from one configuration to another allows the mechanical engineers to determine whether the power output by the engine is efficiently reaching the tires after passing through the rest of the system. The configuration of the Secondary return spring affects how quickly the CVT engages, affecting vehicle performance as well. There are five potentially unique tunes we need to find that correspond to the best performance for each dynamic competition event: Hill Climb, Tractor Pull, Acceleration, Maneuverability, and Endurance.

The Drivetrain Optimization system will record the rotational speed of the CVT Primary and Secondary, the change in position over time of the Secondary, the course environment, and the vehicle position and speed during test time. At analysis time, we will calculate the CVT ratio and compare it to the course, vehicle position, and vehicle speed to determine which tune performed the best and predict which type of tunes are the best for the defined competition events.

2. Design Considerations

2.1 Assumptions and Dependencies

The software is fully dependent on Python version 3.7; it will not be fully functional on any previous version of the language. The assumption is that the software will be utilized on the two dedicated RPi's and no other device.

2.2 General Constraints

Hardware Limitation	
Component	Limitation
VL6180X Distance Sensor Range	5mm - 100mm
55140 Series Hall Effect Sensor Range	10mm - 20mm
VK162 GPS Module	No walls, cloudy skies or dead-zones.
Pi Camera V1	1024 x 768 image resolution for object recognition to work.

Software Constraints	
Sensor Name	Constraint
55140 Series Hall Effect Sensor	The system has no constraints to our knowledge.
VL6180X Distance Sensor Range	The system shall gather data in units of inches.

RPi3 Constraints
Requirement Description
The operating system and local memory is limited to the size of the micro SD card (currently 64GB).
The distance between the RPi4 and RPi3 connected by a ~6 ft Ethernet Cable.

Graphical User Interface Constraints
Requirement Description
The system shall receive data from RPi4 and RPi3.
The system shall use a screen of 5’’ 800 x 480 to display information.

2.3 Goals and Guidelines

The goal of this software is to assist the Baja SAE student-run organization in optimizing the drivetrain of their competition vehicle. Other software goals include the following:

1. The system software shall be fast and reliable
    a. The system shall record accurate data from sensors
    b. The system shall handle the storage of data to non-volatile memory
    c. The system shall automate processes to minimize user
2. The GUI shall be user-friendly
    a. The design shall be easy to view on the monitor
    b. The design shall be intuitive to use by a non-technical team member
3. The system software shall provide data analysis with correct filters
    a. The system shall filter outliers and noise in the data
    b. The system shall have troubleshooting procedures in place
    c. The system shall calculate results using reliable data

2.4 Development Methods

Our project development follows the Waterfall project management methodology. Requirements were set by the Cal State LA Baja SAE team at the beginning of the project, and the development of modules follows a sequential order. This type of project development ensures that the software agrees with the Baja team's build and testing timeline. Revisions to the software and its architecture would impede its availability for use during testing of the vehicle, so any changes may only be requested after preliminary collection and analysis of data is provided. Such changes will be left to a future software team and/or the Baja members responsible for programming and data acquisition.

3. Architectural Strategies

Python IDLE

Used as primary IDE for software and source code construction.

Gmplot Library API

The Gmplot Library API was utilized to integrate GPS data with the Google Maps web mapping service. Using this, several coordinates were plotted to show the route taken by a vehicle with the mounted GPS.

ImageAI Library

The ImageAI Library is the key library that our systems uses for object detection. It provides an easy to understand and customizable interface to integrate image recognition into our system.

ADC Library Adafruit_MCP3008 Adafruit_GPIO.SPI

This library was essential in converting the analog data that we recorded from the sensors into digital data. It is employed by our project to handle the general inputs and outputs needed to be processed.

ADC Library V16180x

In order for the distance sensor to record data properly, this library was needed. It contains the necessary dependencies in order for our system to use the distance sensor.

MATLAB

This software was essential in filtering and analyzing the data gathered from the Hall Effect sensors and the Distance Ranging sensor.

Data Acquisition Hardware

The data acquisition hardware that is included in our system is comprised of Hall Effect sensors, a Distance Ranging sensor, camera, GPS module, and an MCP3008 Analog to Digital converter.

System Hardware

The primary hardware of our system is comprised of a Raspberry Pi 3 and a Raspberry Pi 4. The RPi 3 handles the collection of data which does not require high hardware performance. The RPi 4 handles the collection of data which is heavily reliant on the performance of its operating system.

External Hardware

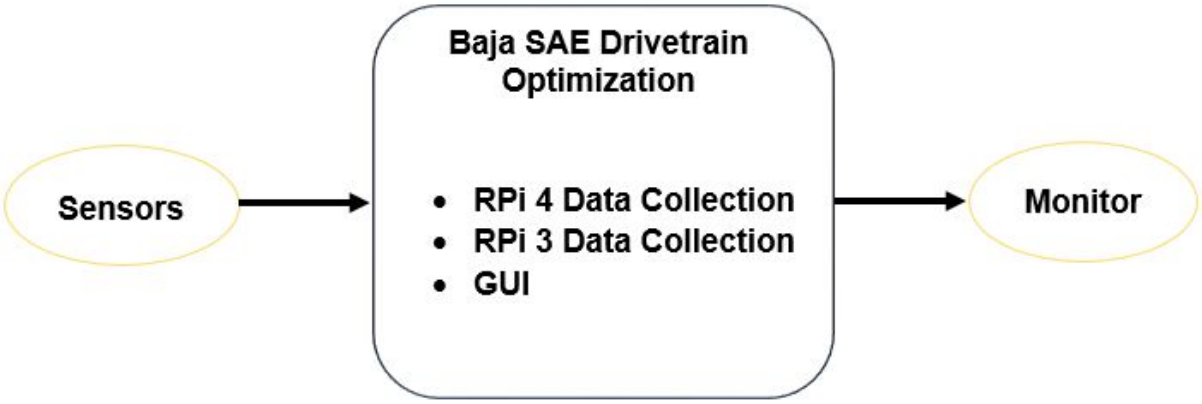
The hardware needed to interface with our system are basic computer peripherals, including a Bluetooth keyboard, a mouse with a USB cord, and a screen with an HDMI cord.

Data Interpretation

Data collected by the sensors will be interpreted by the Fast Data Collection Module and the Slow Data Collection Module.

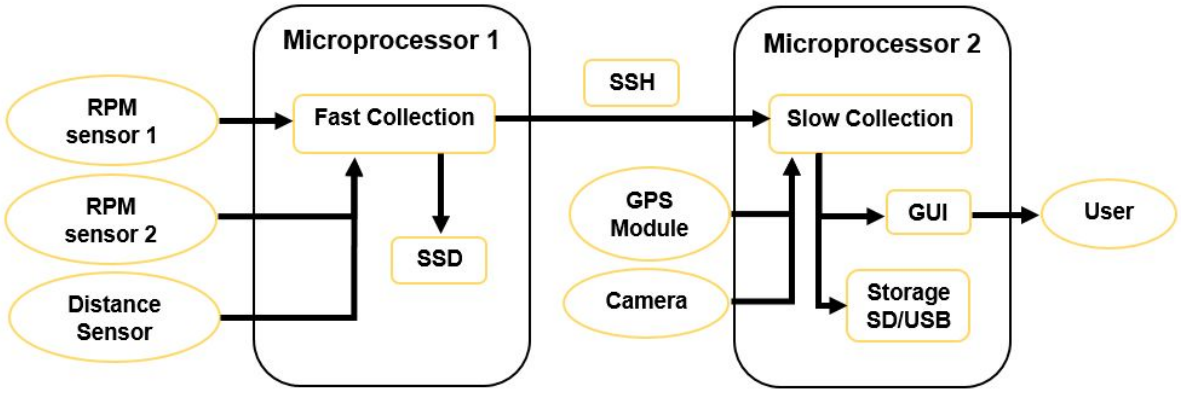
4. System Architecture

4.1 Level 0 DFD



At the top-most organizational level, our software accepts data from the sensors into two RPi's that are responsible for filtering, saving, and processing the inputs. In addition, the RPi 3 is responsible for the GUI module of the system which will display the appropriate information requested by the Baja SAE team.

4.2 Level 1 DFD



4.2.1 Microprocessor 1 - RPi 4 Data Collection

The first RPi is used for the "fast data collection" system to gather the data generated by the RPM sensors and the distance sensor. The RPi will collect the data and run it through pre-set filters in order to sift through the noise for each sensor. The information is sent to the second

RPi through the Secure Shell protocol connection for processing. This is done to ensure that the speed of the data collection is not compromised.

4.2.2 Microprocessor 2 - RPi 3 Data Collection

The second RPi is used for the “slow data collection” system to gather the data generated by the GPS module and PiCamera, process and calculate all the data gathered, and generate the GUI display consisting of the information required by the Baja SAE team. This is where all the information is going to be stored including all the .csv and .jpg files.

4.2.3 GUI

The GUI is responsible for being the only interactive portion with the user. The GUI will accept the time, date, and CVT configuration from the user and use the inputs in its display. The user shall input the type of test run, CVT primary weight, CVT primary spring, CVT secondary spring, CVT secondary clock.

5. Policies and Tactics

5.1 Choice of which specific products used

PRODUCT	DETAILS	PURPOSE
IDE		
Python 3 Language	Python3 IDLE	Data collection software
MATLAB		Data analysis
External Libraries		
Adafruit ADC Library	Adafruit_MCP3008	Interface with analog to digital converter
Google Maps Plotting Library	Gmplot	Locating and Plotting API
Image Recognition Library	ImageAI	Image processing
Adafruit Distance Sensor Library	V16180x	Interface with distance sensor

5.2 Plans for ensuring requirements traceability

Requirements are up to date and contain everything said in the following documentation. If immediate change is required, it will be evaluated and tested to ensure that it works best for the system. Upon successfully implementing an alternative approach or change, a follow-up revision will be planned.

5.3 Plans for testing software

Finished software is tested on the Baja vehicle where the gathered data is analyzed and used for the SAE International Competition on April 16th, 2020. The data will be gathered through dynamic testing procedures, simulating competition conditions and events.

5.4 Interfaces for end-users

The end-user will be able to access the interface through the display attached to RPi 3 as mentioned in section 2.2.

5.5 How to build and generate the system's deliverables

1. First, the user will press the blue "on" button for both Raspberry Pi microprocessors.
2. Once the device is powered on, the user shall plug the Hall Effect Sensors into the Raspberry Pi 4.
3. The user shall start the "GUI.py" script to run the Graphical User Interface.
4. Once the user is ready to start, press the "Start Test Run" button.
5. The user shall input all required information and press "OK"
6. Once the dynamic test is done, the user shall press the "Stop Test Run" button.
7. Repeat steps 4-6 as necessary.
8. Once all data is collected, the user should remove the SSD and USB drives and download stored data to a computer with MATLAB for analysis.
9. The drives should be replaced once they are cleared.
10. On a computer with MATLAB, the user shall run the developed data analysis script(s) to view performance information.

5.6 Plan for maintaining software

Software maintenance or upgrades will be the responsibility of the Cal State LA Baja SAE student organization or the next Computer Science team assigned to the project.

6. Detailed System Design

6.1 Fast Data Collection Module

6.1.1 Responsibilities

The Fast Data Collection Module interprets the data collected from the Hall Effect Sensors and the Distance sensor and sends them to the Slow Data Collection Module.

6.1.2 Constraints

See section 2.2.

6.1.3 Composition

The Fast Data Collection Module interacts with two Hall Effect Sensors, one Distance Sensor and one Raspberry Pi 4. An SSH connection shall be used for communication between the Raspberry Pi 3 and Raspberry Pi 4.

6.1.4 Uses/Interactions

The Fast Data Collection Module will interact with the Slow Data Collection Module. It will commence the collection of the RPM data.

6.1.5 Resources

The Fast Data Collection Module will utilize sensors to gather information.

6.2 Slow Data Collection Module

6.2.1 Responsibilities

The Slow Data Collection Module will store GLL data form the GPS module, as well as record images every three seconds of the course environment. After sending appropriate processed data to the GUI, the Slow Data Collection Module will save all data into a central location.

6.2.2 Constraints

See section 2.2

6.2.3 Composition

The Slow Data Collection Module interacts with one GPS module, one camera and one Raspberry Pi 3. A SSH connection shall be used for communication between the Raspberry Pi 3 and Raspberry Pi 4.

6.2.4 Uses/Interactions

The Slow Data Collection Module will interact with the Fast Data Collection Module and GUI.The Slow Data Collection Module will start the Hall Effect sensor script on the Fast Data Collection Module.

6.2.5 Resources

The Slow Data Collection Module will utilize the data from the Fast Data Collection Module and hardware sensors to gather information.

6.3 Graphical User Interface Module(GUI)

6.3.1 Responsibilities

The GUI will initiate the data gathering process of all sensors on both multiprocessors, accessing the RPi 4 through SSH connection.

6.3.2 Constraints

See section 2.2.

6.3.3 Composition

The GUI will include the text fields for the user to input the type of test run, CVT Primary weight, CVT Primary spring, CVT Secondary spring, CVT Secondary clock.

6.3.4 Uses/Interactions

The GUI interacts with the Slow Data Collection Module; it receives data from the Slow Data Collection Module.

6.3.5 Resources

The GUI will utilize the data obtained from the Slow Data Collection Module to gather information.

6.4 Data Analysis Module

6.4.1 Responsibilities

The Data Analysis Module will determine the best weights and springs to use and the best clocking and stiffness to use in the CVT Secondary for each respective course type. It will also track the path taken during the course runs and classify each recorded image, such as an uphill slope, turns, and rocks.

6.4.2 Constraints

See section 2.2.

6.4.3 Composition

The Data Analysis Module includes the software to determine the best configuration for the vehicle.

6.4.4 Uses/Interactions

The Data Analysis Module does not interact with other modules.

6.4.5 Resources

The Data Analysis Module will utilize the Data from the Fast Data Collection Module and the Slow Data Collection Module.

7. Detailed Lower Level Component Design

New files will be added in future revisions of the document as the project progresses.

7.1 GUI.py

7.1.1 Classification

The file will implement the methods needed for a functional GUI the user can interface with.

7.1.2 Processing Narrative (PSPEC)

It is a python file that utilizes Tkinter to create a simple GUI that allows starting the gathering of data.

7.1.3 Interface Description

The interface will consist of two simple pages. The first page allows for the starting and stopping of the data gathering. The second page contains text fields the user may use to specify the configurations used for the test run.

7.1.4 Processing Detail

This file does not require processing, it does not gather relevant data.

7.1.4.1 Restrictions/Limitations

This system has no constraints to our knowledge.

7.1.4.2 Performance Issues

This system has no performance issues, it does not gather relevant data.

7.1.4.3 Design Constraints

The GUI must be user-friendly and display text easy to read from a reasonable distance.

7.2 picamera_script.py

7.2.1 Classification

This file will implement the methods that will allow the Pi Camera to work.

7.2.2 Processing Narrative (PSPEC)

It is a python file that takes pictures every 3 seconds.

7.2.3 Interface Description

This file does not need an interface, it only takes pictures.

7.2.4 Processing Detail

The images collected will be processed to an off-site computer.

7.2.4.1 Restrictions/Limitations

The system has no constraints to our knowledge.

7.2.4.2 Performance Issues

Pi camera needs 1024 x 768 image resolution for object recognition to work.

7.2.4.3 Design Constraints

Pi camera needs to be placed in an angle where it can see the whole view of the driver for it to capture the best image.

7.3 gps_module_script.py

7.3.1 Classification

This file will implement the methods that will allow the GPS module to work.

7.3.2 Processing Narrative (PSPEC)

It is a python file that will collect data from the GPS module and when finished will be able to plot it into a map.

7.3.3 Interface Description

This file does not need an interface, it only collects data.

7.3.4 Processing Detail

The data collected will be processed within the microprocessor.

7.3.4.1 Restrictions/Limitations

The GPS module will not function if it is run within Cal State LA campus vicinity.

7.3.4.2 Performance Issues

GPS module will have trouble if the speed limit is going over 40 mph.

7.3.4.3 Design Constraints

GPS module needs to be placed in the highest elevation of the car for it to get the best accurate coordinates.

7.4 hall_effect_script.py

7.4.1 Classification

This file will implement the methods that will allow the two Hall Effect sensors and the distance sensor to work.

7.4.2 Processing Narrative (PSPEC)

It is a python file that will collect data from the Primary and Secondary CVT and outputting an RPM as well as the distance the Secondary opens.

7.4.3 Interface Description

This file does not need an interface, it only collects data.

7.4.4 Processing Detail

The data collected will be processed within the microprocessor.

7.4.4.1 Restrictions/Limitations

The system has no constraints to our knowledge.

7.4.4.2 Performance Issues

The microprocessor needs to have a fast CPU for it to collect accurate data.

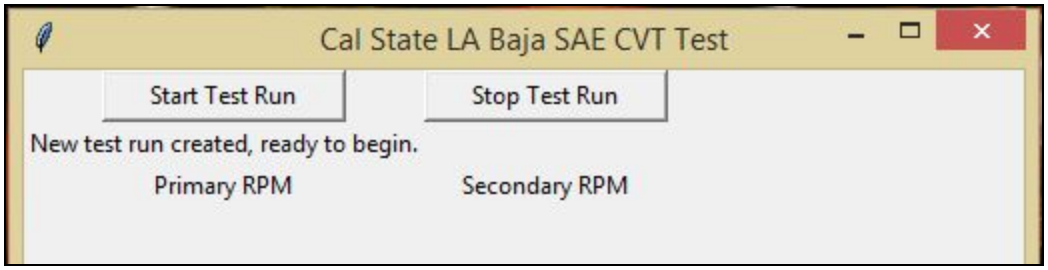
7.4.4.3 Design Constraints

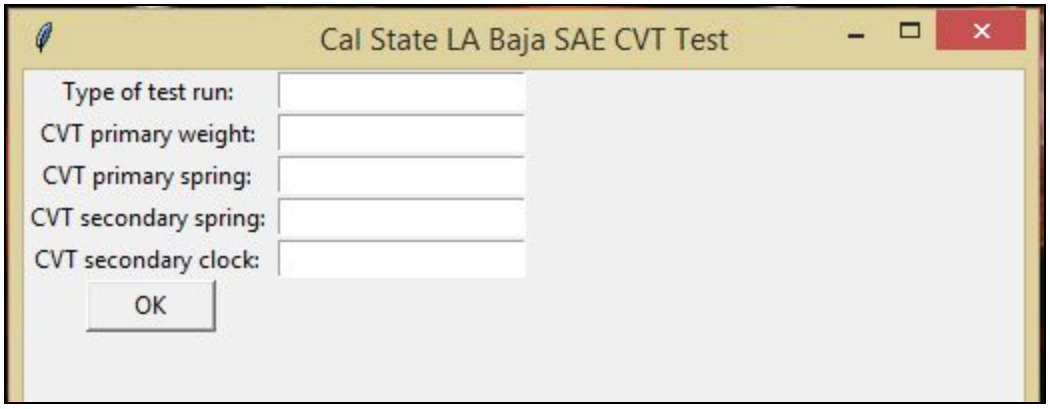
The Hall Effect sensor needs to be between 10mm - 20mm from the magnet for it to be able to collect accurate data.

8. User Interface

8.1 Overview of the User Interface

The GUI shall provide a way to input the test date, time, CVT Primary and Secondary weight and spring configurations, and course type.
The GUI shall display the CVT Secondary return rate.
The GUI shall display the ratio of RPMs of the CVT Primary and Secondary.
The GUI may be required to show more live sensor data as required by the Cal State LA Baja SAE team.
The system shall receive data from RPi4 and RPi3.
The system shall use a screen of 5’’ 800 x 480 to display information.





9. Requirements Validation and Verification

The requirements referenced in this section can be found in the project's SDD Section 4.1 Functional Requirements.

9.1 Fast Data Collection Module

Fast Data Collection Module (RPi4)		
	Requirement Description	Testing Description
4.0-1	The system shall interpret the data collected from the Hall Effect Sensors to produce RPMs of the Primary and Secondary.	Use handheld tachometer to verify RPM readings are accurate based on engine and CVT ratio specifications.
4.0-2	The system shall record the position of the Secondary.	Use a ruler to verify the distance readings are accurate.
4.0-5	The system shall save data to non-volatile memory.	The data is saved to an SSD.

Module 4.1 - Slow Data Collection (RPi3)		
	Requirement Description	Testing Description
4.1-1	The system shall store GLL data from the GPS Module.	Tested GPS module in several locations to determine the accuracy of GLL data. **Note**: Had a problem collecting data on campus.
4.1-2	The system shall record an image of the course environment every 3 seconds.	Check the specified file location for the saved pictures.
4.1-4	The system shall save data to non-volatile memory.	The data is saved to a USB drive

Module 4.2 - Graphical User Interface (GUI)		
	Requirement Description	Testing Description
4.2-1	The system shall provide a way to input the test date, time, CVT Primary and Secondary weight and spring configurations, and course type.	The second window of the GUI provides input spaces for the required information.
4.2-2	The system shall display the CVT Secondary return rate.	Unable to implement
4.2-3	The system shall display the ratio of RPMs of the CVT Primary and Secondary.	Unable to implement
4.2-3	The system may be required to show more live sensor data as required by the Cal State LA Baja SAE team.	Unable to implement

Module 4.3 - Data Analysis		
	Requirement Description	Testing Description
4.3-1	The system shall determine which weight and spring configuration primary was the best for the respective course type.	TBD
4.3-2	The system shall determine which clocking and stiffness was best for secondary for each course type.	TBD
4.3-3	The system shall determine the path the vehicle traveled throughout the course.	TBD

4.3-4	The system shall determine the speed of the vehicle at different points throughout the course.	Unable to implement
4.3-5	The system shall classify each recorded image - ex. uphill, turn left, turn right, hit a rock.	Unable to implement

Glossary

Abbreviations and Acronyms

- GPIO - General Purpose Input and Output
- RPi 3 - Raspberry Pi 3
- RPi 4 - Raspberry Pi 4
- OS - Operating System
- GUI - Graphical User Interface
- ADC - Analog to Digital Converter
- DFD - Data Flow Diagram
- NMEA - National Marine Electronics Association
- GLL - Geographic Latitude and Longitude
- SSH - Secure Shell Protocol

Definitions

- *CVT* - Continuously Variable Transmission. A type of clutch that has a Primary and Secondary component, which is connected by a belt.
- *Primary* - A rotating body connected to the engine by the engine output shaft. It drives the Secondary member by transferring power through a belt. The primary may be tuned by changing the internal weights and springs.
- *Secondary* - A rotating body connected to the Gearbox by the Gearbox input shaft. It is driven by the Primary through a belt. The Secondary may be tuned by clocking and changing the stiffness of its return spring.
- *Gearbox* - A mechanical device whose internals consists of gears that create a speed reduction from the input rotational speed to the output rotational speed. The input rotational speed comes from the Secondary and the output rotational speed goes to the vehicle wheel.

References

Documents

NMEA Reference Manual

SiRF Technology, Inc., January 2005. Revision 1.3.

Collegiate Design Series Baja SAE® Rules 2020

Baja SAE, September 8, 2019. Revision B.

Setup Guides

- **Distance Sensor Setup Guide:** https://learn.adafruit.com/adafruit-vl6180x-time-of-flight-micro-lidar-distance-sensor-breakout/python-circuitpython
- **Analogue Sensors On The Raspberry Pi Using An MCP3008:** https://www.raspberrypi-spy.co.uk/2013/10/analogue-sensors-on-the-raspberry-pi-using-an-mcp3008/