TrackPack: Intelligent Vehicle Racing Monitor and Recording System

University of Central Florida

Senior Design

Group 6

Kevin Singh	Computer Engineering
Anjali Jodharam	Computer Engineering
George Gruse	Photonic Science and Engineering
Myles Musanti	Electrical Engineering

Reviewers

Mark Llewellyn	Computer Science
C. Kyle Renshaw	Optics and Photonics
Sonali Das	Electrical Engineering

July 25th, 2023

Table of Contents

List of Figures

List of Tables

1 Executive Summary

Vehicle racers around the world are constantly searching for ways to improve both their vehicles and their skills as drivers. Historically, to test vehicle performance data, it would be almost impossible to do so without going to your local closed course raceway. It's common that people don't live within a reasonable distance to a closed course raceway and testing your vehicle on a closed course raceway can be costly. TrackPack seeks to bring the vehicle performance data tracking from the closed course raceway directly to the hands of the user.

TrackPack seeks to bring vehicle performance data that traditionally would be generated on a raceway using multiple sensors along the track to a small compact, hand-held, user-friendly device. To accomplish this challenge, TrackPack incorporates a design that allows the user to mount TrackPack directly to the windshield of the vehicle, this design includes multiple sensors and Bluetooth OBD-II reading from the vehicles computer to display all the vehicle's metrics directly to the user. This document was developed to describe the entirety of the design and implementation process for TrackPack. We begin by discussing the plans and goals for the design and implementation of TrackPack, then we delve into comparable technologies that are like TrackPack that already exist in today's market. Subsequently, we compare different technologies that exist, and which would be the best fit for implementing all TrackPack's functionalities. Following this, we carefully select potential hardware candidates that can be used by performing a full-depth analysis of each hardware item and comparing them against other potential selections. Then, we examine the various standards that exist for the components that we selected in the previous section and develop a set of constraints for TrackPack that we maintained while moving into the design process. The design process accurately represents all our hardware schematics and testing, as well as our software design. Following the design process is how we plan on integrating our components and implementing our final design. Lastly, we have administrative content consisting of our bill of materials, milestones, work distributions, etc. The final pages of the document includes our conclusions and sources.

2 Project Description

2.1 Project Introduction

With thousands of vehicle racetracks and millions of car enthusiasts in the United States alone, people are consistently seeking ways to measure their vehicle's performance, as well as their own performance as drivers. Vehicle racing has evolved greatly since inception in the late 1800s, and cars today are becoming extremely fast and precise machines with immense capabilities.

With the competition in mind, TrackPack was born. Looking for ways to measure, record, and share accurate vehicle statistics both on and off the track has been a challenge for years. With groundbreaking advancements in technology, we are now able to allocate advanced features in a compact design.

At the root of TrackPack is a small microcomputer with extensive processing power. In addition to the microcomputer, GPS and an accelerometer allow the user to accurately track location, speed, and acceleration. Using these parameters alone, we can accurately determine a vehicle's 0-60mph time, 0-100mph time, ⅛ mile time, ¼ mile time, and other key racing measurements such as braking distances, g-force, lap times, etc.

TrackPack doesn't stop there, however, using an onboard camera TrackPack allows you to record and save footage from the vehicle to be played back later and compared with the measured parameters. Furthermore, whether you're a casual driver, occasional spirited driver, or competitive racer, TrackPack can help you measure your vehicle's health. Utilizing the Onboard Diagnostics port in the vehicle, TrackPack allows users to measure and manage their vehicles health by gaining access to a slew of parameters directly from the vehicle's computers.

2.2 Project Motivation

Similar products to TrackPack currently exist on the market with certain limitations in functionality. These limitations require consumers to purchase additional accessories to read all the parameters that TrackPack reads. We set out to create an all-in-one device that consumers can use to track all this data in a single, concise, portable, and easy to use design. The market for consumer electronics dealing with vehicle performance is substantial, racing enthusiasts are always looking for a way to enhance their vehicles performance. Of course, enhancing vehicle performance yields a more spirited driving experience, but TrackPack wants to enable users to take their driving experience to the next level. By tracking the user's vehicle performance and live data from the vehicle, we can ensure that the consumer has the most accurate performance data on their vehicle.

2.3 Project Goals

Basic goals:

- Design a module that monitors a vehicle's performance in real-time.
- Provide accurate readings of the vehicle's acceleration, braking, and handling.
- Help drivers understand their car's performance to optimize driving strategy and improve lap times.
- Assist in identifying issues with the vehicle and point the driver in the right direction for repairs or upgrades.
- Design a lens system to record drivers' perspective for review.

Advanced goals:

- Allow racers and car enthusiasts to make quick adjustments to their driving style or vehicle setup based on real-time data feedback.
- Provide comparative analysis of different vehicles or setups to help users make informed decisions about improving performance.
- Develop an optical design and video collection system that records a perspective closely matching what the driver sees during races and driving.
- Enable users to cross-reference any issues found in the data collected to any terrain encountered.

Overall, the basic goals focus on designing a module that monitors vehicle performance, provides real-time data feedback, and assists in identifying issues with the vehicle. The advanced goals build upon these basic goals by enabling racers and car enthusiasts to make informed decisions based on comparative analysis, and by providing a more detailed and accurate perspective through the optical design and video collection system.

2.4 Project Objectives

The TrackPack embodies a compact, light-weight design that is battery powered. This module is OBD-II compatible where it readr vehicle performance parameters and store results on an SD card. Significant specifications such as location, are accurately tracked using onboard GPS. An accelerometer has been implemented to measure proper acceleration to determine g-force. As a bonus, the TrackPack includes a video recorder which serves as a dash cam and/or a way to share your experience with friends and family. The recordings allow the viewer to see everything that the driver saw and more. The videos also allow the driver useful feedback, having the data collected during the drive presented alongside the recording. Its objective is to provide accurate and reliable performance data for car enthusiasts and professionals who want to improve their driving skills and enhance their vehicle's performance. The data collected by TrackPack can be used to fine-tune a vehicle's performance and make modifications to improve its performance, speed, and handling.

#	Objectives
2.4.1	Plug-and-Play functionality
2.4.2	Lightweight & Portable Design
2.4.3	Low-Latency parameter tracking
2.4.4	Video Footage
2.4.5	High quality display
2.4.6	Wide angle footage
2.4.7	External data storage
2.4.8	User-friendly interface

Table 1: TrackPack Objectives

2.5 Function of Project

Our device takes the input from the OBD II port as well as use this port to supply power to the device on the vehicle to then read back the values of the emissions, fuel efficiency, etc. We transmitted this data to our microcontroller and add the data from the accelerometer and GPS. The microcontroller has to determine when to begin reading the detailed statistics. Once the measurement has been calculated the measured value is then dislayed while continuing to collect the speed from the accelerometer. The statistics are read out to the user on a display and the footage that is taken from the camera module also displays these statistics back to the user on an LCD display with the current statistics that the user has set to scan for. Once the data is recorded from the device the data is then transmitted to the display with the aim of a 3ms time delay, to give the data as quickly as possible to the user. All the data taken from the OBD II port is read by the microcontroller present on the PCB along with the additional modules. To implement the image processing done by our dash cam we may need additional processing power to successfully present the entirety of the data.

2.6 Marketing Analysis

Devices exist that can monitor individual aspects such as race time, vehicle health, acceleration times, position, and driver POV but few solutions exist that can achieve all the above. The products that are on the market can cost upwards of $1000, which acts as a barrier to entry level racers. There are devices that support connecting external monitoring systems, but these systems require more space within the vehicle. The goal of the TrackPack is to provide a low-cost all-in-one solution to all levels of drivers, which would fill a need market space and encourage other companies to provide a greater scope of measurements in a single device.

2.7 Project Requirement Specifications

2.7.1 Project Hardware Specifications

Specification Number	Hardware Parameters	Measurements
2.7.1.1	PCB board size	10cm x 10cm
2.7.1.2	Dash Cam (FOV)	>80° to capture an even wider angle than what the driver sees
2.7.1.3	Pixel Resolution	1080p video for high-definition recordings
2.7.1.4	Video Frame Rate	Due to fast paced nature of driving a frame rate of 30 fps will limit motion blur during quick accelerations
2.7.1.5	Optical Resolution	The spot size of the on axis and off axis rays will be smaller 250 microns in radius
2.7.1.6	Optical Aperture	The optical system will be designed to achieve a f-number between f/1.8 and f/2.8 to balance light input and the depth of field
2.7.1.7	Complete device in housing size	The TrackPack will be compact and portable. 4in x 3in x 2 in
2.7.1.8	Trackable speed	0 mph to 999 mph
2.7.1.9	Power Supply	The TrackPack will be able to obtain power from the OBD II port to have no need of a separate battery support
2.7.1.10	OBD II compatible	The TrackPack will include OBD II compatibility to collect vehicle parameters such as engine pressures, engine temperatures, emissions, etc.
2.7.1.11	Weight	The TrackPack will be lightweight to support vehicle weight reduction. <= 1 lbs.
2.7.1.12	Accelerometer	i2c and SPI interface Scales of 2g to 16g
2.7.1.13	Gyroscope	i2c and SPI interface Measurement range 125 to 2000 dps Sensitivity 4.375, 8.75, 17.50, 35, 70 mdps
2.7.1.14	GPS Module	Altitude of 50,000m Max update rate 10 Hz Horizontal position accuracy <2.5m CEP Acquisition sensitivity -148dBm Tracking sensitivity -167dBm

Table 2: Project Hardware Requirement Specifications

2.7.2 Project Software Specifications

Specification Number	Software Specifications
2.7.2.1	Ability to connect to Bluetooth OBD-II adapter to read vehicle computer
2.7.2.2	Ability to correctly and accurately grab user location and velocity
2.7.2.3	Ability to capture and store measured parameters
2.7.2.4	Ability to capture and store videos
2.7.2.5	Ability to measure vehicle acceleration metrics

Table 3: Project Software Requirement Specifications

2.8 House of Quality

The House of Quality matrix is an important tool in defining customer needs and correlating these needs with the fundamentals of development. To develop a great product, it's important to identify the wants and needs of the customer and the engineering requirements. By utilizing a House of Quality matrix, we can determine how the wants and needs of the customer coincide with the engineering requirements and what level of precedence certain features hold.

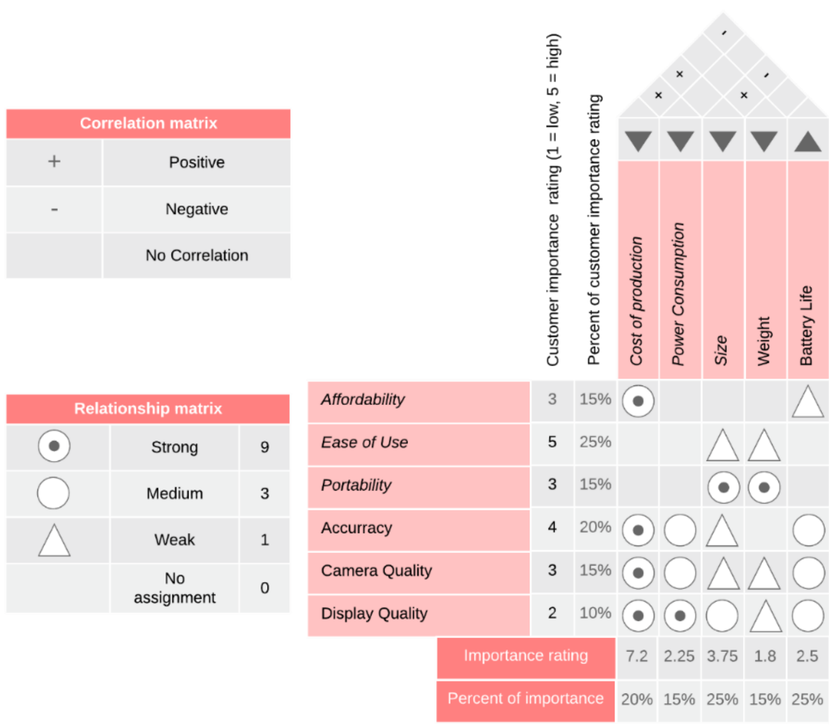


Figure 1: House of Quality

2.9 Block Diagram in Hardware

Figure 2: Block Diagram in Hardware

2.10 Flowchart in Software

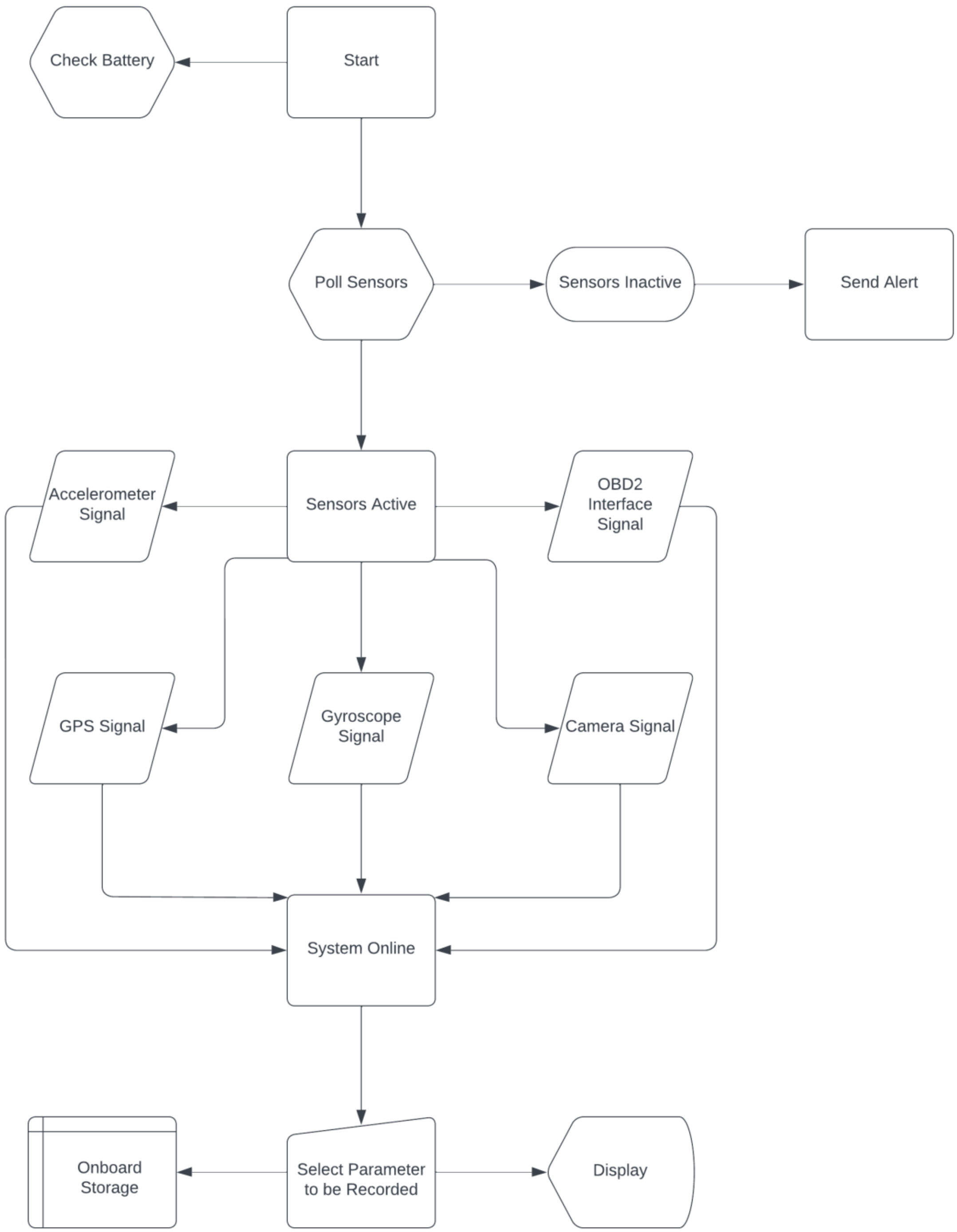


Figure 3: Flowchart in Software

3 Research and Part Selection

Part selection research is crucial in various fields such as engineering, manufacturing, and construction. It involves identifying and choosing the appropriate components, materials, and parts to use in the design and production of a particular product or system. The selection process involves considering multiple factors such as cost, availability, functionality, durability, and compatibility with other parts. Our part selections are meticulously researched and compared when it comes to ensuring the high building quality that consumers will see on the TrackPack. Here we will discuss both the research done during the selection of the parts, as well as relevant technologies which make up the part selection process.

The main constraints we are considering when selecting the components for our system are cost and size, as we need the system to be relatively compact and affordable. Each individual component must also be entirely compatible with one another. When creating our schematic this constraint we would have to ensure that if were to pick a component to ensure the output type to see what parts we would be limited to and to look for future parts in the design process to make sure that they are also compatible.

3.1 Existing Products Comparison

Similar products to TrackPack existing on the market have their own unique features and advantages, and the best choice depends on the user's specific needs and preferences. Depending on if the consumer is more focused on accuracy, ease of use, or cost, this can dictate which product they choose. The following relevant products are beneficial to our research, it allows us to intricately analyze the advantages, disadvantages, features, etc. of each design. With extensive comparison between existing products on the market, we can use the research to support the final design and build of TrackPack.

3.1.1 Garmin Catalyst Driving Performance Optimizer

Figure 4: Garmin Catalyst

Garmin is a company that specializes in the design of global positioning system (GPS) enabled products. Among the many items that Garmin designs and produces, they primarily make GPS navigational tools, smartwatches, fitness trackers, action cameras, and radar systems. Individuals and professionals in fields like aviation and marine navigation, search and rescue missions, and law enforcement utilize the advanced technology that Garmin offers.

The Garmin Catalyst Driving Performance Optimizer is an advanced performance data and coaching tool created especially for track drivers. It is a device that mounts to the dashboard or windshield of a vehicle and employs a mix of sensors, cameras, and GPS technology to collect real-time data on a driver's performance. This includes speed, lap times, and driving line. The device is compatible with mobile applications on Android devices and iOS devices.

The Garmin Catalyst Driving Performance Optimizer has a compact, 7.84" W x 4.79" H x 0.93" D, design that includes a heavy-duty mount for the cockpit. It connects to a power source and the vehicle's OBD II port. The device features a touch screen display size of 6.0" W x 3.5" H with a display resolution of 1024 x 600 pixels. The display makes it easy to see the driver's performance information and coaching feedback. The display includes a predictive lap timer, speed, lap times, and G-forces among other data elements. The data fields that are shown on the screen can be altered by the driver to fit their preferences. All the information gathered while driving is logged onto the device and may subsequently be accessed on the Catalyst companion app. As a result, the driver can examine their performance statistics in greater detail and pinpoint areas where they can improve over time. Equipped with a high-quality camera, the Garmin Catalyst Driving Performance Optimizer records at 1080p resolution and 30 fps. The camera includes a wide-angle lens that is designed to capture a 140° field of view of a driver's performance/trip on the track.

Capabilities of the Garmin Catalyst Driving Performance Optimizer include 10 Hz multi-GNSS positioning, image processing, and built-in accelerometers to generate the driver’s racing line on the track. The device also provides an on-track driving coach where the device provides audio cues through either Bluetooth or the vehicle’s stereo. With the help of this ground-breaking technology, user's best times for each track section are combined to provide their ideal driving time based on lines users have driven. The Garmin Catalyst Driving Performance Optimizer displays the driver’s apex performance, showing how the timing of apex decisions impacts the overall performance and speed on the track. In addition to the collected data, the device also keeps track of your best lap time, adaptive delta time, number of laps and total session time.

3.1.2 Dragy

Dragy Motorsports is a company that provides performance measurement devices for consumers interested in monitoring and tracking the performance of their vehicle. The Dragy device is encased in a portable design of a 1” x 3” box weighing in at 2 lbs. which is a size that works well with being set on the dashboard of your vehicle or even stored in your glovebox when not in used.

Figure 5: Dragy

Dragy is an independent device that works in conjunction with a smartphone equipped with Bluetooth control to the device that holds up to 10 hours of battery life. The device is connected to an application that is consistently being updated with new features, this application is accessible to both iOS devices

Figure 5: Dragy

and Android devices. The smartphone app allows users to view and share their performance data in real-time. The app also provides various features, such as performance leaderboards, video overlays, and social media sharing options.

Dragy offers a video synchronization feature which allows users to combine their video footage with the performance data collected by Dragy. Dragy synchronizes video footage with performance data using its internal clock, so users can view both simultaneously. Users simply start recording video on their smartphone or camera at the same time as they begin a Dragy run.

Dragy aims to provide race precision timing so that car enthusiasts can make modifications to their vehicle accordingly to hit their peak performance. This device allows for the monitoring of a driver’s 0-60mph, 60-130mph, 100-200kmh, 1/4-mile, 1/2-mile performance, etc. Dragy gathers a vehicle’s performance parameters by using high speed GPS satellites. Some features include measuring parameters such as: acceleration, braking time, G-force measurement, and lap timer.

With the support of accurate performance readings, Dragy encourages the idea of a cost-effective lifestyle where they support consumers to spend wisely on the modifications to their vehicle instead of a costly performance measurement tool.

3.1.3 VBOX Video HD2

Figure 6: VBOX Video HD2

VBOX Video HD2 is a performance device that relays the parameters of a user’s vehicle in real-time. The comprehensive collection of data allows users to analyze the performance of their vehicle and their driving, allowing drivers to make the appropriate adjustments to their driving. The VBOX Video HD2 utilizes intelligent data logging technology, various sensors, and real-time streaming to form an analysis.

The VBOX Video HD2 holds up to a six-hour battery life with a weight of 130g, which is noted to be an ideal weight. The VBOX Video HD2 is advertised to be compatible with any type of vehicle such as a car, motorcycle, bicycle, jet-ski, powerboat etc. In addition to offering compatibility with Apple dev ices like the iPhone, iPod Touch, and iPad, the VBOX Sport also comes with a Bluetooth connection, allowing for the use of the device with Bluetooth to enhance GPS reception on iOS devices, or to add GPS functionality if the devices are not equipped with GPS. The system includes an internal power backup to prevent lost or corrupted data.

The VBOX Video HD2 is equipped with a dual camera setup, with HDMI video output and 1080p 30 fps HD video. Video footage and data from the data logger are recorded along with a synchronized video from the VBOX Video HD2, allowing users to capture high-quality race footage. Some components to the video feature include multiple camera inputs and switching options, live streaming, synchronized data overlay on video footage, and customizable video overlays and graphics.

The VBOX Video HD2 is used to record GPS data such as speed, acceleration, distance, and time. Users can easily analyze the performance data that has been captured by the VBOX Video HD2 directly using the analysis software - Circuit Tools - and users can store these results directly into the SD card that is provided. Some notable features of the VBOX Video HD2 include: 10Hz GPS engine, internal GPS antenna, socket for external GPS antenna, USB charging, SD card logging, waterproof camera, and free data analysis software. The VBOX Video HD2 has advanced features such as predictive lap timing, a built-in display, and the ability to measure lateral G-forces.

3.1.4 SoloStorm

SoloStorm is a smartphone-based performance tracking app that records performance data from various sensors, such as accelerometers, GPS, and OBD-II. SoloStorm offers a camera recording option. A compatible camera must be connected to SoloStorm and configured as a video source to use the camera recording feature. After this is set up, users can start and stop the recording manually or automatically, and the footage is stored with all the user's other session(s) data.

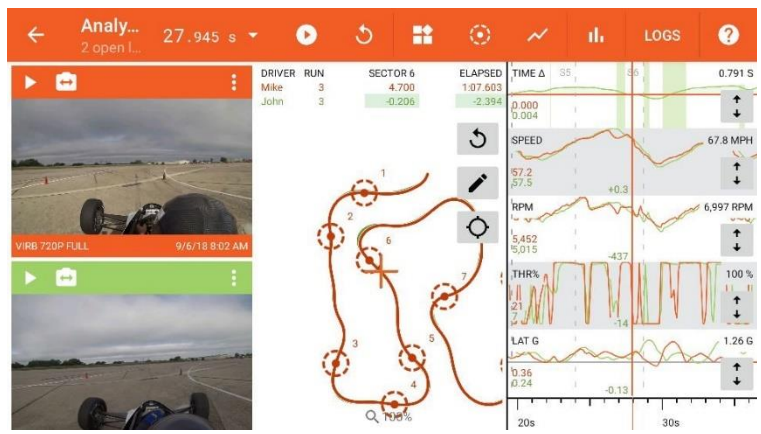


Figure 7: SoloStorm

The SoloStorm Autocross Data Logger is only compatible with Android and the SoloStorm software must be purchased with a Bluetooth GPS receiver (SoloStorm GPS Package). At an additional cost for OBD II compliant vehicles, consumers can purchase the Bluetooth OBD II reader. With the Bluetooth OBD II reader users can log throttle position and RPM. Another addition for CAN OBD II compliant vehicles, consumers can purchase the SoloStorm RaceCapture/TrackPackage which will allow consumers to log OBD and other CAN bus channels at higher sample rates. Some vehicles that are compatible with this feature may allow consumers to log brake pressure and steering angle. For vehicles that are not OBD II compliant, SoloStorm offers the SoloStorm Race Capture/Pro Package where SoloStorm can be connected via a USB or wireless connection to standalone data acquisition systems that can connect directly to your vehicle's sensors. These systems typically include their own sensors and data acquisition hardware. This package provides consumers with features where they can run brake pressure, steering, or custom sensors with this data logger and it includes analog inputs, CAN bus connectivity, accelerometers, and gyroscopes, among other features.

Product Comparison				
Product	Garmin	Dragy	VBOX Video HD2	SoloStorm
Battery Life/Power Supply	up to 2 hours	Up to 10 hours	12 V auxiliary or cigar lighter socket	N/A
Connection Type	Bluetooth	Bluetooth	Bluetooth	Bluetooth OBD II/Standalone data acquisition system
iOS and Android Compatible	Yes	Yes	Yes	Android only
Dimensions	7.84" W x 4.79" H x 0.93" D	1” x 3”	178 x 143 x 35.5 mm	N/A
Built-in Display	Yes	No	No	No
Weight	~15.4 oz	~2 lbs	~870 g	N/A
Camera	Yes	No	Yes	No
Cost	$999.00	$189.00	$ 3,895.00	$304.00

Table 4: Existing Product Comparison

3.1.5 Dashboard Vehicle Camera Comparisons

To create a starting point for the desired imaging specifications, we decided to research existing dashboard camera technology. This was decided due to the wide availability of cameras on Amazon and their commonplace status within many vehicles. The table below shows the specification determined from the Amazon listings.

	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5	Cam 6
Resolution	1080p	1440p	1080p	1080p	1080p	1080p
FOV	150°	170°	170	150	165	170
F/#	1.8	1.8	1.8	N/A	1.8	N/A
Frame Rate	60 fps	30 fps	30 fps	30 fps	60 fps	N/A

Sensor Size	1/2.8”	N/A	N/A	N/A	N/A	N/A
# of lenses	6	6	N/A	6	N/A	7

Table 5: Amazon Dashboard Camera Comparison

The Amazon listings were surveyed based on Amazon’s top listings when “Car Dash Cam” was searched for assuming that the higher listings were from reputable sellers. Some listings did not include each specification we were looking for and some also seemed to have unreliable specifications according to some reviews. From the chart we can see that the minimum resolution of these cameras is 1920 pixels by 1080 pixels. This is to provide a clear image of the road without pixelation. Also, these listings favor smaller F/#’s, all items that had the F/# listed were 1.8. This allows those systems to collect as much light as possible and enables the device to have a faster shutter speed, or frame rate in video. Another detail gathered from these products also contain six or more lenses in a small form factor. From this research we have created goals and constraints that we used to design the optical system included in our project.

3.2 Technology Comparison

3.2.1 UART

UART works by using two wires, one for transmitting data and another for receiving data. The data is transmitted in a series of bits, with each bit representing a 1 or a 0. The bits are sent one after another, with a start bit and a stop bit framing each byte of data. The start bit is always a logic 0, and the stop bit is always a logic 1. The data bits can be any combination of 1s and 0s.

One of the main advantages of UART is that it is asynchronous, which means that the transmitting device and the receiving device do not have to be synchronized with each other. This makes it easier to implement and more flexible than synchronous communication protocols, which require the two devices to be synchronized. Another advantage of UART is that it is relatively simple to implement. It only requires a few hardware components, such as a shift register and a baud rate generator, which can be easily integrated into a microcontroller or other embedded system. The baud rate is the rate at which the data is transmitted over the UART connection, and it determines the speed of the communication. The baud rate is usually set by the transmitting device, and the receiving device must be configured to match the same baud rate to receive the data correctly. The baud rate is usually expressed in bits per second (bps), and common baud rates include 9600, 19200, and 115200 bps. To utilize UART, the transmitting device sends the data serially one byte at a time. The data is sent using the start bit, followed by the data bits, and then the stop bit. The receiving device detects the start bit, and then samples the data bits at the appropriate time to receive the data. Once the stop bit is detected, the receiving device knows that the byte is complete and can be processed.

3.2.2 SPI

SPI works by using a master/slave architecture, where one device (the master) controls the communication and initiates the data transfer, and one or more devices (the slaves) respond to the master's commands and send data back to the master. The master device generates a clock signal that is used to synchronize the communication between the devices, and data is transmitted and received simultaneously over separate wires. SPI uses four wires for communication: a clock signal (SCK), a master output slave input (MOSI) signal, a master input slave output (MISO) signal, and a chip select (CS) signal.

The clock signal is generated by the master and is used to synchronize the communication between the master and the slave devices. The MOSI signal is used by the master to send data to the slave, and the MISO signal is used by the slave to send data back to the master. The CS signal is used to select the slave device that the master wants to communicate with. SPI data is transmitted in packets, with each packet consisting of a set number of bits. The master initiates the data transfer by sending a packet of bits to the slave, and the slave responds by sending a packet of bits back to the master. The packets can be any length, and the master and slave must agree on the packet length before the communication begins. One of the advantages of SPI is its speed. SPI can operate at high speeds, up to several megabits per second, which makes it ideal for applications that require fast data transfer rates. Another advantage of SPI is its simplicity. The protocol is relatively easy to implement and requires only a few hardware components, making it a popular choice for low-cost embedded systems.

3.2.3 I^2C

I^2C is a synchronous, multi-master/multi-slave (controller/target), packet switched, single-ended, serial communication bus. It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication. A particular strength of I^2C is the capability of a microcontroller to control a network of device chips with just two general-purpose I/O pins and software. Many other bus technologies used in similar applications, such as Serial Peripheral Interface Bus (SPI), require more pins and signals to connect multiple devices. I2C uses only two bidirectional open-collector or open-drain lines: serial data line (SDA) and serial clock line (SCL), pulled up with resistors. Typical voltages used are +5V or +3.3V, although systems with other voltages are permitted. I^2C has several speed modes, and we used the highest that our MCU can process to send to the user. The High-speed mode (Hs) has a maximum speed of 3.4Mbit/s.

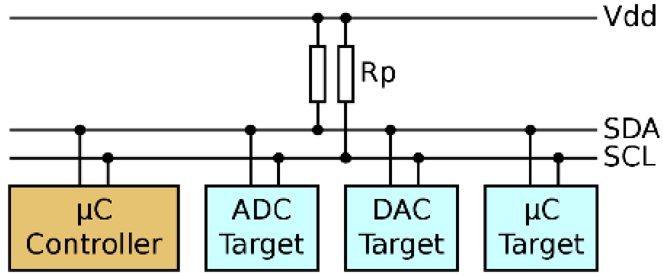


Figure 8: Example of Microcontroller Using I2C

3.2.4 Communication Protocol Choice

Protocol	UART	I2C	SPI
Complexity	Simplest	Easy to chain multiple devices	Complexity increases as device count increases
Speed	Slowest	Faster than UART	Fastest
Number of Devices	Up to 2 devices	Up to 127	As many as needed
Number of Wires	1	2	4
Duplex	Full Duplex	Half Duplex	Full Duplex
Master to Slave ratio	Single to Single	Multiple slaves and masters	1 master, multiple slaves

Table 6: Communication Protocol Comparison

Given the above criteria that each communication protocol met, we decided to use the I^2C. The speeds are fast enough for design requirements and mainly we are going to have several connections between our corresponding devices that are tracking the data in operation, so having a system that is easy to run multiple devices at once was the most important feature when choosing the protocol. This fact will be considered when selecting the auxiliary components of our design as we want them to be I^2C compatible. The selection of the I^2C will greatly reduce the number of pins we need to connect to when creating the schematic and eventually PCB of our project.

3.2.5 PCB Design

Autodesk Eagle, also known as simply Eagle, is the software program for designing printed circuit boards (PCBs) and schematics that we used for our project. For our system we implemented our design using the parts available in the component library as well as importing in the parts we need and adding the required footprints. With the completed file we were able to create our unique PCB design by taking the Gerber file from Eagle and uploading it directly to the manufacturer to be produced. PCB way also provides the ability to assemble the components onto the PCB. The company offers both surface-mount technology (SMT) and through-hole technology (THT) assembly services. For our design we would like to have multiple layers to make a stronger board and try to keep the total footprint of the board to a minimum to reduce space, while keeping all the necessary components to have ample power. The disadvantages to this are a higher cost, and longer lead times, so as soon as a design is agreed upon with all auxiliary components, ordered the board immediately to ensure proper time to debug and test before the due date.

3.2.6 Voltage Regulation

The OBD II port provides several different voltages, depending on the specific pin and function being used. Pin 16 of the OBD II connector provides battery voltage (approximately 12V) directly from the vehicle's battery, which can be used to power external devices that are connected to the OBD II port. However, other pins in the OBD II connector provide different voltages and signals, depending on their specific function. For example, Pin 2 (J1850 Bus +) and Pin 10 (J1850 Bus -) provide a differential voltage signal for communication with certain vehicle modules. Pin 4 (Chassis Ground) and Pin 5 (Signal Ground) provide ground connections for the various signals and voltages. We formed a DC-to-DC conversion to function the PCB and system in order to draw power from the OBD II port and eliminate an external power supply. The other option is to use the 12V accessory outlet to get power and have an adapter to USB to then connect to power the PCB and camera modules. This alternative will let us have an easier connection and provide easier access to the user to plug in the device as well as disassembly to move between vehicles. Another benefit of moving the power supply to the 12V accessory outlet is the removal of any cables below the steering column that could impede the driver in reaching the pedals. One flaw with this input method for the power supply is that the voltage level could be elevated to the 13.5-15V range while the engine is running, so any DC-to-DC converter must be rated to operate at this higher voltage level.

This voltage level will have to be stepped down to operate our devices without damaging them on average between 3Vand 5V values. Linear and switching DC voltage regulators are two commonly used methods for regulating voltage in electronic devices. While both types of regulators serve the same purpose of maintaining a stable output voltage, there are significant differences between them in terms of efficiency, size, cost, and performance. A linear voltage regulator operates by continuously dissipating excess voltage as heat, while maintaining a constant voltage output. This method of voltage regulation is simple and effective, but it can be inefficient, particularly when the input voltage is much higher than the output voltage. Linear voltage regulators are typically smaller and cheaper than switching voltage regulators, but they can also be less precise and generate more heat. On the other hand, switching voltage regulators use a more complex method of voltage regulation that involves rapidly switching the input voltage on and off to maintain a stable output voltage. This method is more efficient than linear voltage regulation, as the excess voltage is not dissipated as heat, but rather stored and reused. Switching voltage regulators are typically larger and more expensive than linear voltage regulators, but they are also more precise and generate less heat. One advantage of switching voltage regulators is their ability to regulate a wide range of input voltages, making them ideal for use in battery-powered devices that have fluctuating input voltages. Switching voltage regulators are also able to handle higher power levels than linear voltage regulators, making them suitable for use in devices that require high levels of power. Linear voltage regulators are simpler and more straightforward to design and use, making them a popular choice for applications that do not require high efficiency or precision. They are also less prone to noise and other issues that can affect the performance of switching voltage regulators. In summary, the choice between linear and switching DC voltage regulators depends on the specific requirements of the application, including the input voltage range, the required output voltage stability, the power level, and the cost and size constraints. While linear voltage regulators are simple, compact, and inexpensive, switching voltage regulators offer higher efficiency, precision, and flexibility in handling varying input voltages and power levels. Most of the components in our system will require an operating voltage level somewhere between the 3V and 5V values. So, we must regulate the voltage from the 12V of the OBD II port and then have a 3.3V and 5V DC-DC converter within the system. In previous courses to regulate voltage we have used the 7805 and 7815 voltage regulators to regulate to 3 and 5 Volts with an input of 5 volts, so we can implement a similar circuit in order to bring our voltage down using a similar schematic and have a voltage closer to the 3-5V range that will power the PCB and then be used by the DC-to-DC converter.

A 3.3V to 5V DCDC converter is a type of DC-to-DC converter that is used to step up a 3.3V input voltage to a 5V output voltage. This process is commonly used in electronic devices that require a higher voltage than the input voltage to operate. The DCDC converter operates by using an inductor and a switching transistor to convert the input voltage into a series of pulses. These pulses are then filtered and regulated to produce a stable output voltage. There are several different types of 3.3V to 5V DCDC converters, including buck-boost, flyback, and boost converters. The choice of converter depends on the specific requirements of the application, such as input voltage range, output voltage, and efficiency.

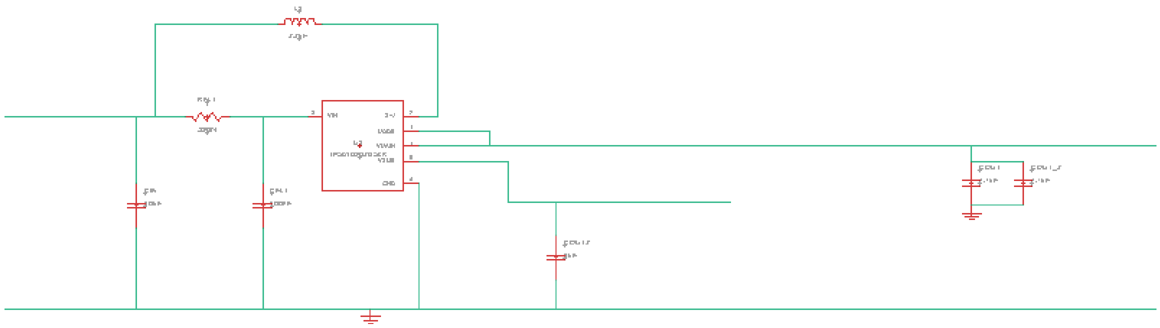


Figure 9: 3.3V VDCDC Converter

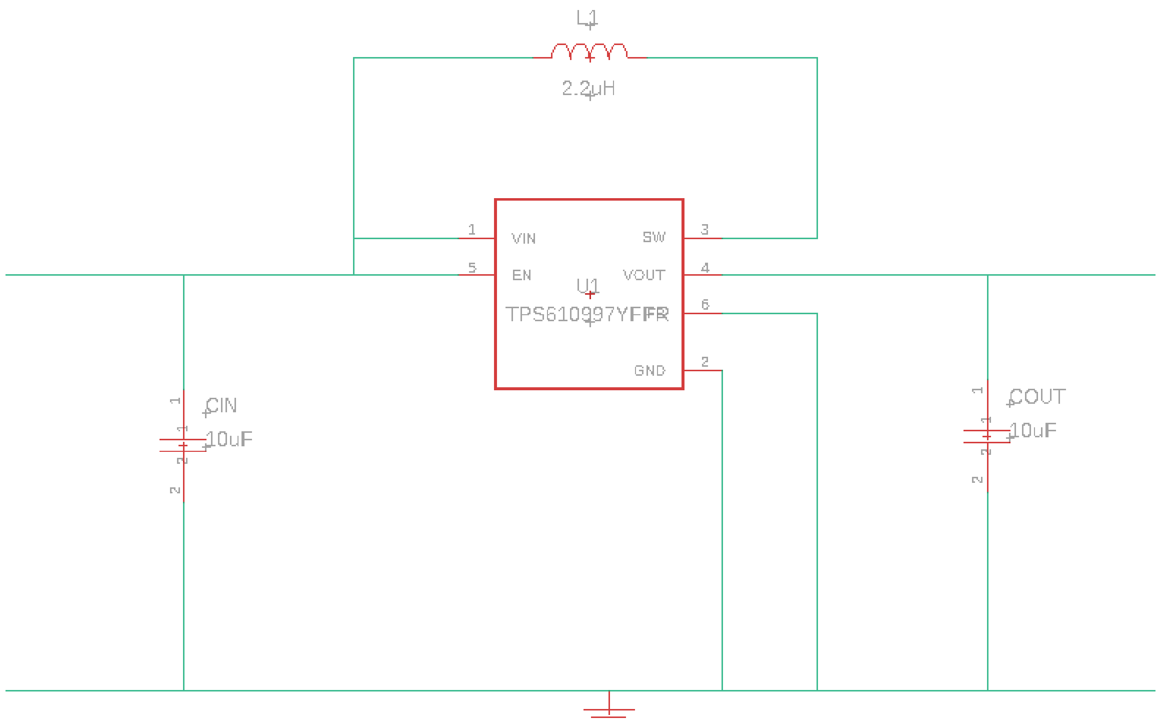


Figure 10: 5V VDCDC Converter

One common example of a 3.3V to 5V DCDC converter is the LM2675 from Texas Instruments. This is a simple switch-mode power supply that provides a fixed 5V output voltage from a 3.3V input voltage. It has a maximum output current of 1A and an efficiency of up to 90%. Another example is the MP2315 from Monolithic Power Systems. This is a synchronous buck converter that provides a variable output voltage up to 5V from a 2.7V to 5.5V input voltage. It has a maximum output current of 1.5A and an efficiency of up to 95%.3.3V to 5V DCDC converters are widely used in a variety of electronic devices, including microcontrollers, sensors, and other low-power applications.

They are typically small and efficient, making them ideal for use in portable and battery-powered devices.

3.2.7 Power Supply

Our main goal after discussing the project, was the possibility of removing an external power supply in the form of a battery and getting the power straight from the vehicle. After researching into the two options of the OBD II port, and the 12V car accessory outlet (cigarette lighter), we chose the 12V accessory due to its ease of access to the user and easier connection port. The 12V cigarette lighter socket in a car provides more than enough power to supply both the PCB and the camera module. To convert the 12V DC power supply from the cigarette lighter socket to a 5V DC power supply suitable for USB devices, a converter circuit is needed. The rectified DC output is then regulated to provide a stable 5V DC output. Switching voltage regulator technology is chosen to reduce unnecessary power consumption wherever possible. In addition to low power consumption, the switching regulator produces far less heat than the linear regulator, which is a critical consideration as the device will be sitting on top of the dash and will be exposed to external heat through the windshield of the vehicle.

To see what possible DC-to-DC converters were available we used the Webench power designer tool available from Texas Instruments. This tool allows us to select an appropriate value for our input voltage, in this case around 12V, and select the ranges we want for each output (3V or 5V) The tool simplifies the design process by automating the selection of key components such as inductors, capacitors, and switching transistors, while considering various design constraints such as size, efficiency, and cost. The user can then select a specific topology and configure it further by specifying additional design constraints such as maximum component size, output ripple voltage, and thermal constraints. Once the design is configured, the tool will generate a complete bill of materials (BOM) that includes all the necessary components, as well as a schematic and layout diagram. Using this system, we wanted to prioritize a small footprint as our highest criteria with at least an efficiency rating above 90% for both the 3V and 5V converters, as past this value there would be little change in the performance and a much larger BOM leading to higher cost and footprint of the circuit. Both schematics can then be imported into Autodesk Eagle with a schematic layout, as well as the footprint layout, if they are in the Eagle library already. Components that are not will have to be added manually by using the footprint and package numbers given from Webench.

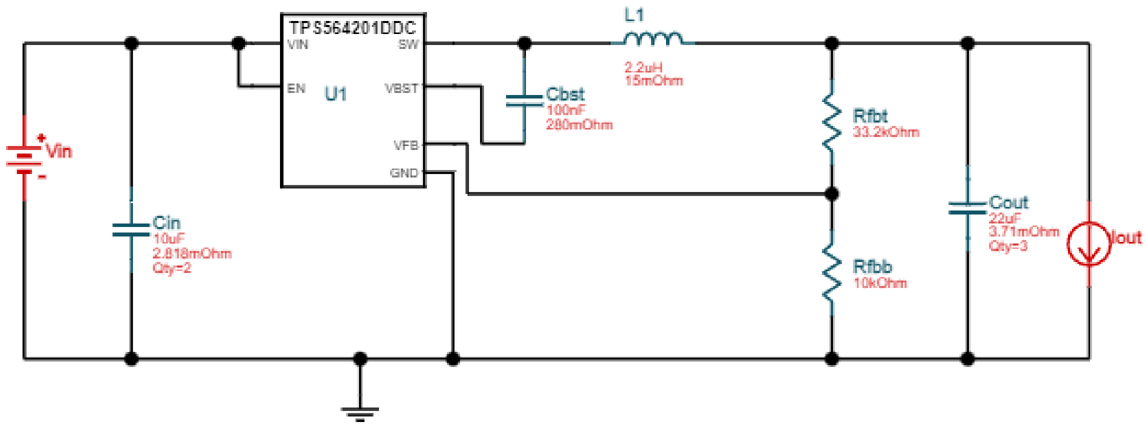


Figure 11: TPS564201 Topology

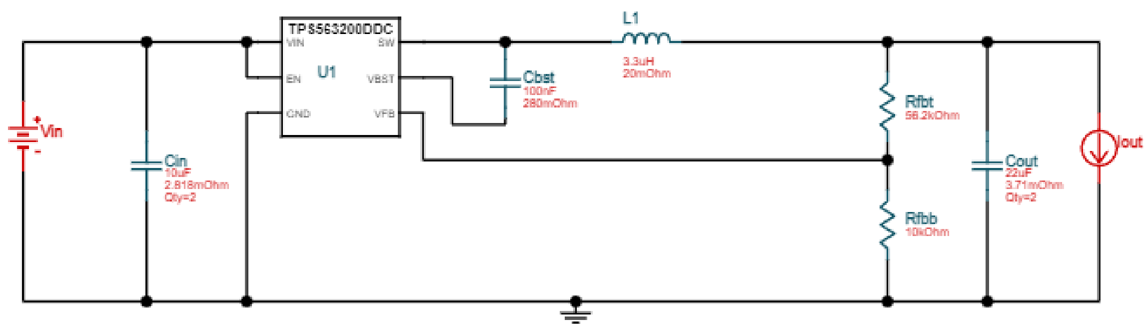


Figure 12: TPS563200 Topology

Power Supply	3V DC to DC TPS564201	5V DC to DC TPS563200
Efficiency	91.7%	93.6%
BOM Cost	$1.08	$0.95
Footprint	163 mm²	158 mm²
BOM Count	10	9
Topology	Buck	Buck
Frequency	478.16kHz	791.16 kHz
IC Cost	$0.35 | 1ku	$0.30 | 1ku

Table 7: Power Supply Comparison

3.2.8 Li-ion vs NiMH Power Supply

One stretch goal is the ability to have a portable device that removes the need of the 12V accessory outlet. Also, to present our design later in SDII having a mobile power supply would allow us to demonstrate the TrackPack's functionality in person and not through a video format. When it comes to rechargeable batteries, there are several options available, each with its own set of advantages and disadvantages. Two of the most popular types of rechargeable batteries are Lithium-ion (Li-ion) and Nickel-metal Hydride (NiMH). We will compare the advantages and disadvantages of these two types of batteries.

Lithium-ion batteries have a high energy density, which means they can store a lot of energy in a small space. This makes them ideal for use in portable electronic devices like smartphones, laptops, and tablets. Lithium-ion batteries have a low self-discharge rate, which means they can retain their charge for a longer time. This makes them ideal for use in devices that are not used frequently, like emergency backup systems. Fast Lithium-ion batteries can be charged quickly, usually in a few hours. They also have a long cycle life, which means they can be recharged and used multiple times before they need to be replaced. This makes them cost-effective in the long run. However, they are prone to overheating and can catch fire or explode if not handled properly. This makes them potentially dangerous and requires careful handling. Over time after use, they can lose their capacity, even if they are not used frequently. This means they may need to be replaced more frequently than other types of batteries. Lithium-ion batteries are difficult to recycle and can be harmful to the environment if not disposed of properly.

Nickel-metal Hydride batteries come in similar sizes and models but have their own advantages and disadvantages as well. Nickel-metal Hydride batteries have a higher capacity than Li-ion batteries, which means they can provide more power and last longer. They are less expensive to manufacture than Li-ion batteries, making them a more cost-effective option at sale value, but Nickel-metal Hydride batteries can develop a memory effect, which means they can lose their capacity if they are not fully discharged before recharging, meaning they are more susceptible to wearing out faster than their competitor. Nickel-metal Hydride batteries have a higher self-discharge rate than Li-ion batteries, which means they can lose their charge over time even when not in use. Nickel-metal Hydride batteries take longer to charge than Li-ion batteries, which can be a disadvantage in certain applications.

Feature	Nickel Metal Hydride	Lithium Ion
Charge/discharge life cycle	Up to 2000 cycles	Up to 1200cycles
Memory Effect	Reduces % of total capacity	n/a
Conditioning	Recommended when new and between long storage	n/a
Power Density(MW/m^3)	1.5-4	0.4-2
Nominal Voltage	1.2V	3.6V
Efficiency	66%-92%	80%-90%
Self-discharge	30%loss/month	2.5%loss/month

Table 8: Battery Statistics Comparison

Given the required criteria of our design, and with the possibility using a battery power source to show our design demonstration during SDII, we chose the Lithium-ion battery pack. For the purpose of using it in the presentation we need a battery that can last around 4 hours and not have to worry about it failing during a demo. The hindrance of the NiMH battery having the memory effect that could prevent us from having a power supply for the entire presentation, as well as its sufficiently longer charge time to get it back up and running. The device would also not be used for a substantial period, so we would have to worry about the self-discharge that would additionally take away from its maximum charge. As a product that can be reproduced and manufactured, the TrackPack would get used only upon visiting a course and we would not want the customer’s device to have the degradation between use in the power supply, considering track visits are likely not a daily event. The main focal point of the external power supply for the user is to just grab the device and place it for use, so limiting the work the driver has to do is the most important design criteria.

3.2.9 Image Processing

For our PCB we required an MCU to control and operate our system, the recording of images from the cam is too power intensive to also have it run off the PCB’s microcontroller. To process the images will have a separate MCU to run the dash cam. This will be in the form of a Raspberry Pi. The Raspberry Pi is also known for its GPIO (General Purpose Input/Output) pins, which allow users to connect various sensors, actuators, and other electronic components to the board. This makes the Raspberry Pi an excellent choice for projects. Moreover, the Raspberry Pi supports various programming languages, including Python, C/C++, and Java, which makes it easy to develop and control projects. The boards are powered by an ARM-based CPU, with various models featuring different speeds, RAM, and connectivity options. Due to its versatility, we can connect our camera and lens module to the Raspberry pi which will be strong enough to process the video. Another reason for the need for the additional MCU is the larger memory space to collect all the video taken that will be a much larger size than any of the other data transferred. This data can be stored for later viewing. One of the most common ways to use the Raspberry Pi for image processing is to connect a camera module to the board and use it to capture images or video. The Raspberry Pi camera module is a small camera that can be attached to the board using a ribbon cable. It can capture images with a resolution of up to 8 megapixels and video at up to 1080p resolution. The camera module can be controlled using the Raspberry Pi's GPIO pins, and images and video can be saved to the board's SD card. Once images or video have been captured, the Raspberry Pi can be used to process the data. This can involve using software libraries and tools such as OpenCV, which is a popular open-source computer vision library. OpenCV provides a range of functions for image processing, including image filtering, edge detection, and object detection. The Raspberry Pi's processing power may be limited compared to a dedicated computer, but it is still capable of performing many basic image processing tasks, mainly the ability for us to save the data locally on an SD card.

3.2.10 Image Processing Software

PyTorch, TensorFlow, and OpenCV are all popular frameworks used for image processing tasks. While they have similarities in terms of their functionality, each framework has its own unique strengths and weaknesses. Image Input and Output: All three frameworks support reading and writing images from and to various file formats such as JPEG and PNG. However, PyTorch relies heavily on the PIL library for image input and output, while TensorFlow has its own built-in image I/O functions. OpenCV, on the other hand, provides a range of functions for image input and output, including support for video streams. All three frameworks provide a range of functions for image preprocessing. However, PyTorch has a more extensive set of functions for image augmentation, including random transformations such as rotation, translation, and flipping. TensorFlow also provides a range of image preprocessing functions, but its support for image augmentation is not as extensive as PyTorch's. OpenCV provides a wide range of image preprocessing functions, including filtering, image segmentation, and morphological operations. All three frameworks provide functions for analyzing images and extracting features such as edges, corners, and blobs. PyTorch and TensorFlow both have high-level APIs for building and training convolutional neural networks (CNNs) that are commonly used for image processing tasks. OpenCV, on the other hand, has a range of functions for detecting objects in images using various object detection algorithms such as YOLO and Faster R-CNN Performance is a crucial factor when it comes to image processing. PyTorch and TensorFlow are both optimized for running on GPUs, which can greatly speed up image processing tasks. OpenCV is also optimized for running on CPUs and GPUs, but its performance may not be as good as PyTorch and TensorFlow for deep learning tasks. The ease of use of a framework can greatly affect its popularity and adoption. PyTorch and TensorFlow both have a steep learning curve, especially for beginners. However, PyTorch has a more pythonic syntax and is generally considered to be more user-friendly. TensorFlow has a more complex syntax but has more extensive documentation and tutorials available. OpenCV has a C++ API, which may be more challenging for beginners to use. However, it also has a pythonic API, which is more user-friendly. Community support is another crucial factor when it comes to selecting a framework. PyTorch and TensorFlow both have large and active communities, with a wealth of resources available, including tutorials, documentation, and user forums. OpenCV also has a large community, but its focus is more on computer vision in general rather than deep learning specifically. Each framework is better suited to certain types of image processing tasks. PyTorch and TensorFlow are both commonly used for deep learning tasks such as image classification, object detection, and segmentation. OpenCV is more commonly used for traditional computer vision tasks such as image filtering, feature extraction, and object tracking.

In summary, PyTorch, TensorFlow, and OpenCV all have their own unique strengths and weaknesses when it comes to image processing. PyTorch and TensorFlow are both optimized for deep learning tasks and have extensive support for building and training CNNs. OpenCV, on the other hand, is more focused on traditional computer vision tasks but also provides support for deep learning. The choice of framework will depend on the specific requirements of the image processing task and the expertise and preferences of the developer or researcher.

	OpenCv	TensorFlow	PyTorch
API Level	High	High and Low	Low
Architecture	Simple, readable	Complex	Complex
Datasets	Small datasets	Larger Datasets (high performance)	Larger Datasets (high performance)
Debugging	Simple network	Difficult to debug	User friendly debugging capabilities
Speed	Slower in C++ faster in Python	Fast high performance	Fast high Performance
Language	C++, Python	C++, CUDA, Python	Lua

Table 9: Imaging Software

3.2.11 Human Visual System

The human eye is an incredibly complex and sophisticated optical system, capable of perceiving a vast range of visual information. The eye consists of several different components, each with its own function and purpose. The most well-known part of the eye is the retina, which is responsible for converting light into neural signals that can be interpreted by the brain.

The fovea is a small, central area of the retina that is responsible for our most detailed and precise vision. It is only a few millimeters in diameter and contains a high density of cone cells, which are responsible for color vision and fine detail. The fovea is crucial for tasks that require high visual acuity, such as reading or recognizing faces.

One of the most important aspects of the fovea is its viewing angle. The viewing angle is the maximum angle that an object can be seen by the fovea without moving the eyes. In other words, it is the angle of the visual field that is subtended by the fovea. The viewing angle is determined by the size of the fovea and its distance from the object being viewed.

The viewing angle of the fovea is approximately 1 degree. This means that an object must be within a few degrees of the center of the visual field to be seen in high detail. Objects that are outside of this range will appear blurry and indistinct. This is why we need to move our eyes to scan a scene to see everything in detail.

The viewing angle of the fovea has important implications for a number of fields, including optics, and visual perception. For example, understanding the limits of the foveal viewing angle is crucial for designing high-resolution camera systems meant to replicate the human vision, such as the design needed for our project.

Understanding how the human visual system collects information about the environment is key to understanding what we should attempt to achieve with our design. Since the fovea of the human eye has an FOV of approximately two degrees but has the advantage of rapid eye movement to observe a scene we had to achieve a larger field angle with our lens design to collect video that resembles the driver general perspective.

3.2.12 Image Ray Tracing

We would like to have a lens system that is designed to minimize the need for expensive specialty optics. To do this the lens design will be limited to spherical singlet lenses. Lenses with spherical curvature are easier to produce and cheaper than other alternatives such as parabolic lenses, which reduce spherical aberration at an increased cost. To start the lens design, the system will be designed as a single thin lens. This calculation method assumes the lens to be a two-dimensional plane, ignoring the propagation of rays that occur between the two surfaces of a lens. To calculate the focal length of the single lens needed to achieve the desired FOV used the equation below where d is the diagonal size of the sensor.

$$FOV = \tan^{-1}\left(\frac{d}{2f}\right)$$

Once the required focal length of a single thin lens is determined, the lens makers equation below will be used to calculate the necessary surface curvatures and thickness of the lens so that it can be input into Zemax.

$$P = \frac{1}{f} = (n-1)\left(\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)d}{nR_1R_2}\right)$$

The next design step will be to split the single lens power across two or more lenses. With more than a single lens, paraxial ray tracing simplifies the process of tracing rays through multiple surfaces and can be utilized in spreadsheets to quickly determine how rays propagate as variables are adjusted. The equations for paraxial ray equation for a spherical surface and paraxial ray propagation in free space are below.

$$n_1u_1 = n_2u_2 - \frac{(n_2 - n_1)}{R}y$$

$$y' = y + u'z$$

The lens shapes that will be available for the design are bi-convex, plano-convex, bi-concave, and meniscus. Each shape bends incoming light rays in different manners and can be combined with varying lens power to capture light from the desired FOV and focus it to the correct dimensions on the imaging sensors plane. The radii of both faces determine whether the lens will converge or diverge incoming rays. An ideal converging lens focuses light from all angles to a single focal point forming a real image. An ideal diverging lens spreads light away from the optical axis and as a result forms a virtual image. In actuality, lenses have a focal depth, a range of distance that rays are focused on. Spherical aberration causes rays farthest from the optical axis to focus closer to the back surface of the lens than rays that are nearest to the optical axis. Another form of distortion that must be accounted for is field curvature. This distortion is an inherent property of spherical lenses and causes the focal plane to curve in image space. Because the sensor we used a flat surface, optical design will need to be used to minimize this effect. Combinations of converging and diverging will enable the system to focus rays from the maximum incident angle to a plane as close to that of the on-axis rays. Below are examples of types of converging and diverging lenses.

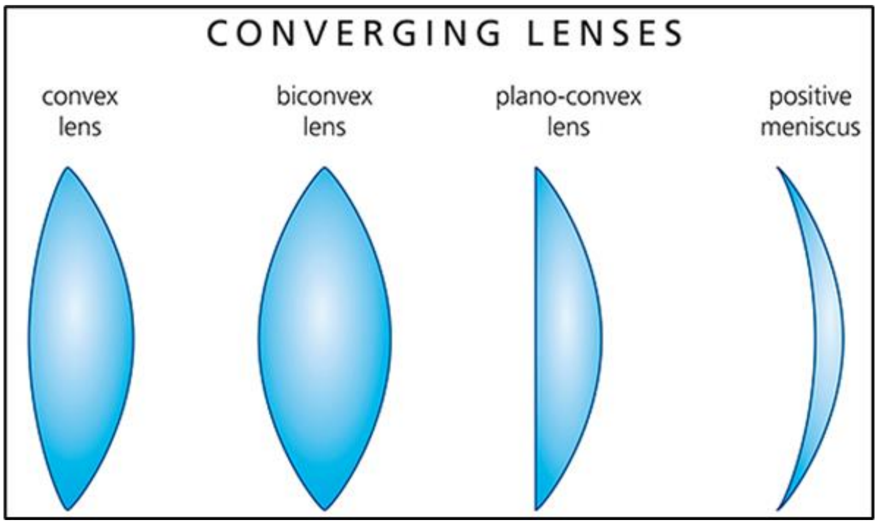


Figure 13: Example of Converging Lenses

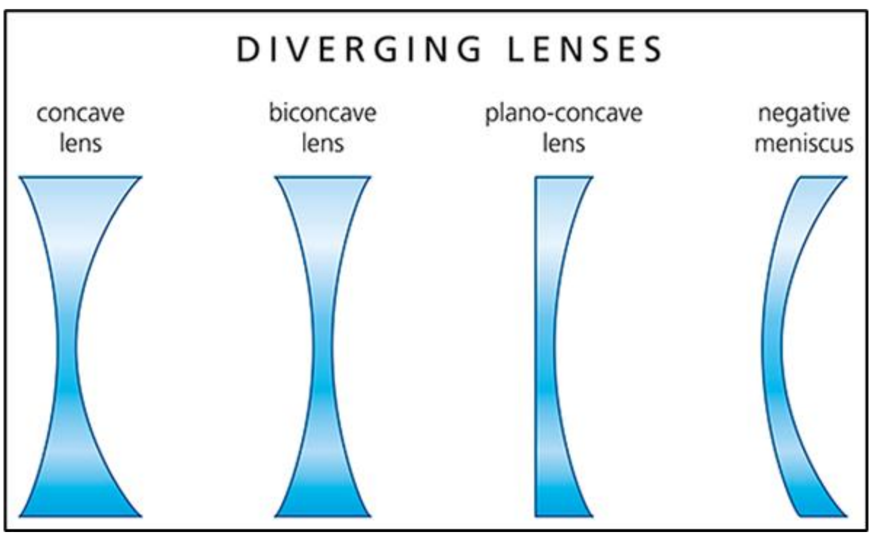


Figure 14: Example of Diverging Lenses

Another degree of freedom in the design is the material choice which will change the index of refraction that each wavelength experiences. Thorlabs Inc. offers lens in four different materials, N- BK7, UV Fused Silica, N-SF11, and CaF2. These materials refract light in the visible spectrum, but also outside of this range in the infrared and ultraviolet. A bandpass filter for 400 nm to 700 nm is also offered by Thorlabs Inc. and could be utilized. Finally, to reduce glare from light reflecting off the road, a linearly polarized filter can be added.

Parameter	Value
Field of View (FOV)	>110°
Image Height	4.45 mm
Resolution	2028 by 1080 pixels
Frame Rate	50 frames per second
Lens System Diameter	25.4 mm
Number of lenses	3 lenses
F-Number Range	F/1.8 - F/2.8

Table 10: Ideal Optical System

The table above shows the parameters that we are aiming for with the lens design. After researching various designs that balance the use of common optical elements, price, and desired Optical specifications we have decided to design a Cooke Triplet style lens. A Cooke Triplet system consists of three lenses, two of positive power and one of negative power equal to the sum of the two positive lenses. A Cooke Triplet is special because it can correct Seidel Aberrations more efficiently than if the system were designed without these guidelines. Another key benefit of this style is that it can correct for field curvature, which at high FOV will be the main source of distortions in our system. To maximize the light collecting on the Raspberry Pi High Quality Sensor the image height will be half the diagonal length of the CMOS sensor which is 4.45 millimeters. The f-number determines two major factors in an optical system, the intensity of the collected light and the focal depth of the image. The intensity of the light is significant for our design because the shutter speed of the sensor would need to be reduced if enough light is not being collected which would lower the maximum frames per second that we could record at. Also, the focal depth of our system is important because if it is too shallow only a very small range of distance will be in focus when recording. For these reasons, a f-number between F/1.8 and F/2.8 is the goal of the optical system. A higher f-number indicates a smaller entrance pupil size which reduced the intensity of incoming light but increases the depth of field of the image. The range listed above will provide us with the proper balance of light collection while also prioritizing a larger depth of field in our system so that no autofocusing elements will be needed.

3.2.13 Zemax OpticStudio

Once the starting point is calculated for the desired specifications, the lens calculations will be transferred into Zemax OpticStudio to further optimize the ray paths through the system. Zemax is a powerful ray tracing software that quickly computes the propagation of light through optics. An important quantity that can be measured through Zemax is the spot size of incoming rays at various angles. The larger the spot size of the incoming rays, the more blur is added to the collected image. Using this software feature we can optimize the system to reduce the spot size to an area equivalent to the area of each pixel on the sensor. Zemax also has features that allow individual variables to be changed to maximize or achieve user defined characteristics such effective focal length, chromatic aberration, and spherical aberration. The ability to quickly optimize portions of the setup to achieve these specifications will ideally enable the system to use commonly available lens.

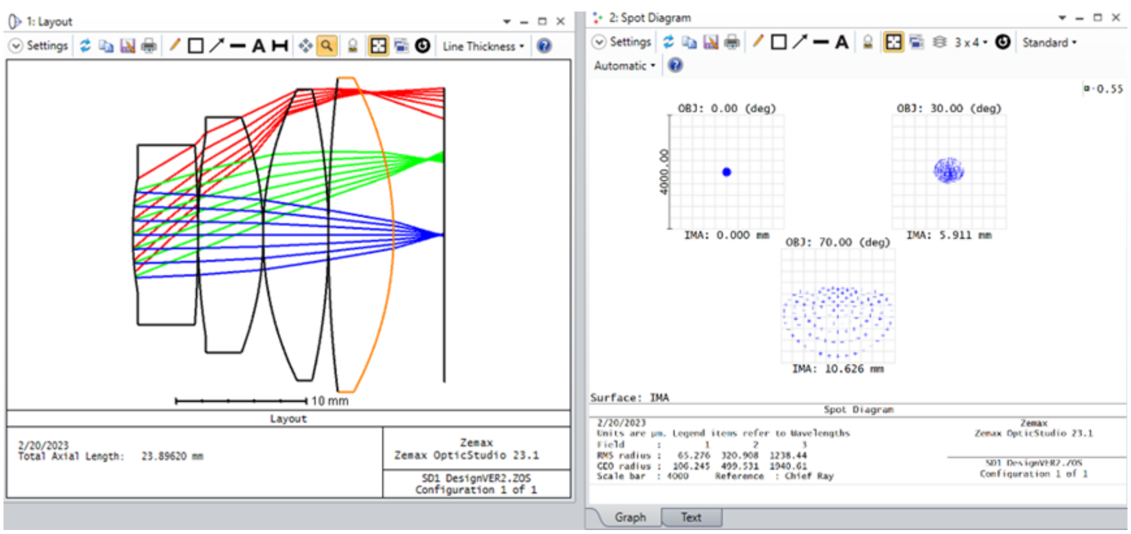
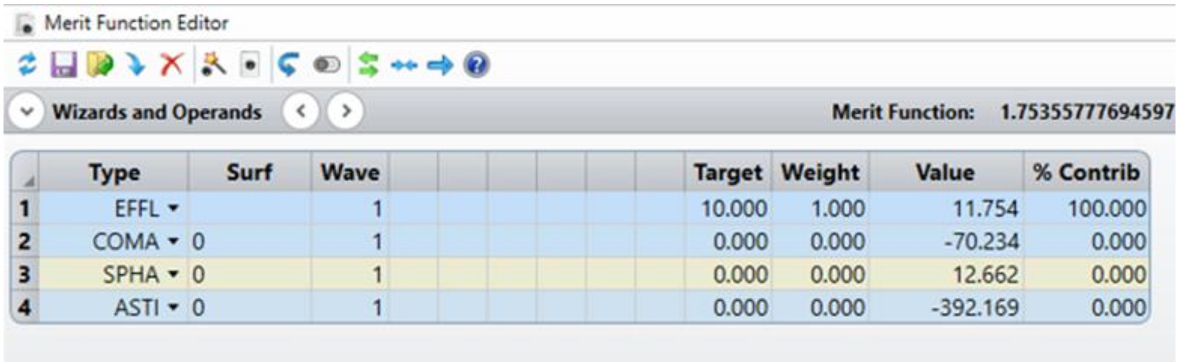


Figure 15: Preliminary Optical Design in Zemax

The window to the left shows the layout of the lens as well as the ray traces at each angle. In this image the rays are spaced at 70 degrees, 30 degrees, and 0 degrees. The window on the left calculated the RMS spot size of the focused light. The RMS spot size is a measurement of how tightly focused collimated rays of light are when they are at the back focal plane. The aim of the design is to get the RMS spot size of the on-axis ray to equal the pixel size so having this type of quantitative measurement will validate the achieved resolution of the system.


Merit Function Editor

Wizards and Operands | Merit Function: 1.75355777694597

	Type	Surf	Wave						Target	Weight	Value	% Contrib
1	EFFL		1						10.000	1.000	11.754	100.000
2	COMA	0	1						0.000	0.000	-70.234	0.000
3	SPHA	0	1						0.000	0.000	12.662	0.000
4	ASTI	0	1						0.000	0.000	-392.169	0.000

Figure 16: Zemax Merit Function Editor

Figure 16 shows the merit function editor within Zemax. Each row is a different quantitative measure of the system, such as EFFL which is the effective focal length. Other variables here represent other aberrations such as coma, spherical aberration, and astigmatism and Zemax has a wide variety of other functions. Merit functions can be given target values and adjust system values such as lens thickness, radius, and distances until the target values are achieved. This will also provide us with a detailed and quantitative analysis of the final lens design that can then be compared to other technologies.

3.3 Hardware Part Selection

3.3.1 Single Board Computer (SBC)

To determine which single board computer should be used for TrackPack, we need to specifically identify the demands of the hardware and match it with a suitable prospect from a list of potential candidates. It’s important that we utilize a single board computer that is capable of simultaneously powering the display, sensors, image processing, and storing this information all with a low latency. Ideally, we are looking for a single board computer that has sufficient processing power and memory, while also retaining a small form factor and uses relatively low power. Using a product that is easily expandable is a significant factor also, a product with a large aftermarket hardware availability will make integrating the additional hardware substantially easier. Furthermore, a more widely used single board computer assists with the availability of resources. Should any issues arise, we hope to have a substantial number of online communities, libraries, or resources to assist with debugging.

The first consideration when choosing our single-board computer will be the processing power. While the essential sensors that TrackPack will utilize (GPS, accelerometer, and gyroscope) don’t require a great amount of processing power, and can typically be powered by a microcontroller, we’ll require the additional processing power to implement the display, image processing, and OBD2 integration. The amount of processing power demanded by TrackPack is the simple reason why a typical microcontroller cannot be utilized.

The second consideration when choosing our single-board computer will be the form factor and power consumption. It’s important that TrackPack maintains the size constraints of 8in x 5in x 3in, this size constraint is important due to the portable and mountable nature of TrackPack. It’s ideal that TrackPack can be dismounted, moved, and remounted between vehicles as easily and quickly as possible. Our board selection must also remain relatively low power to extend the battery life. Typically, single board computers with similar processing power and features maintain similar power consumptions.

The third and final consideration when choosing our single board computer will be the cost, expandability, and support. It’s essential that we choose a board that is cost effective. The obvious choice between two boards that boast similar specifications, but different price points is the cheaper board, however, if the cheaper board comes at the expense of lacking aftermarket expandability options and support, then the more expensive board will be our choice.

3.3.1.1 Nvidia Jetson Nano Developer Kit

Figure 17: Nvidia Jetson Nano

The Jetson Nano is priced at $149.00 on Amazon and boasts a Quad-Core ARM Cortex-A57 CPU and a NVIDIA Maxwell GPU. The Jetson Nano comes standard with 4GB of LPDDR4 memory, and a plethora of interfaces such 4x USB 3.0, 2x MIPI CSI-2, HDMI and DisplayPort, Gigabit Ethernet, M.2 Key E, MicroSD, 3x I2C, 2x SPI, UART, and I2S. The entire assembled Jetson Nano Developer Kit comes in at 3.94in x 3.15in x 1.14in. The Jetson Nano is also capable of dual power modes where the first power mode can run with as little at 5w, while the second power mode utilizes 10w. Since the Jetson Nano was designed around AI, it also features powerful image encoding, up to 4Kp30, and powerful image decoding, up to 4Kp60. NVIDIA provides several resources and documentation on the Jetson Nano, and there are plenty of online resources, articles, and videos on using the Jetson Nano.

3.3.1.2 Asus Tinker Board S R2.0

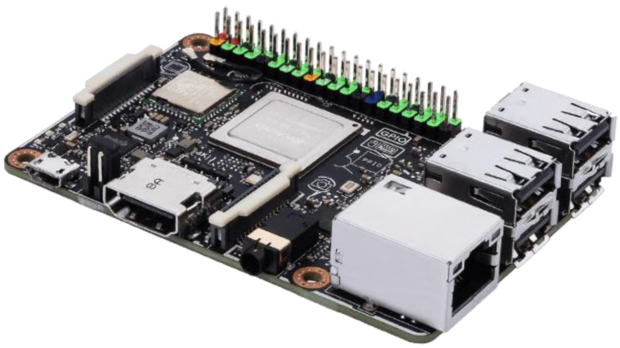


Figure 18: Asus Tinker Board S R2.0

The Asus Tinker Board S R2.0 is the next generation of the original Asus Tinker Board S. The Tinker board S R2.0 is priced at $142.99 on Amazon and features a Rockchip Quad-Core RK3288 CPU and an ARM Mali-T764 GPU. The Tinker Board S R2.0 comes with 2GB of Dual Channel DDR3. The Tinker Board S R2.0 contains an abundance of interfaces such as 4x USB 2.0, MIPI CSI, MIPI DSI, HDMI, Gigabit Ethernet, eMMC, MicroSD, 2x I2C, 2x SPI, 4x UART, and I2S. The Tinker Board S R2.0 also contains onboard 802.11b/g/n and Bluetooth 4.2. The Tinker Board S R2.0 comes in at 3.37in x 2.13in. Asus provides multiple resources and documentation on the Tinker Board S R2.0.

3.3.1.3 Raspberry Pi 4 Model B 4GB

The Raspberry Pi 4 is priced at $55.00 from any approved reseller and features a Quad-Core ARM Cortex-A72 CPU which is integrated on the Broadcom BCM2711 SoC along with the Broadcom Videocore VI GPU. The selected model of the Raspberry Pi 4 contains 4GB of LPDDR4 memory, and their standard number of interfaces including: 2x USB 3.0, 2x USB 2.0, MIPI CSI, MIPI DSI, 2x micro-HDMI, Gigabit Ethernet, MicroSD, 6x I2C, 6x SPI, and 6x UART. The Raspberry Pi 4 also contains onboard 802.11ac and Bluetooth 5.0. The Raspberry Pi 4 comes in at 3.35in x 2.20in. There are a substantial number of resources and documentation available both directly from Raspberry Pi, and other online mediums, and aftermarket part availability is considerable as well.

Figure 19: Raspberry Pi 4

3.3.1.4 HardKernel ODROID-N2+ 4GB

Figure 20: HardKernel ODROID-N2+

The ODROID-N2+ is priced at $83.00 directly from HardKernel and features the ARM big.LITTLE architecture CPU with a Quad-Core ARM Cortex-A73 and a Dual-Core ARM Cortex-A53 as well as a Mali-G52 GPU. The selected version of the ODROID-N2+ contains 4GB of DDR4 memory. The ODROID N2+ interfaces include: 4x USB 3.0, USB 2.0, HDMI, Gigabit Ethernet, eMMC, MicroSD, 2x I2C, SPI, and UART. The ODROID N2+ comes in at 3.54 in x 3.54in x 0.67in. HardKernel provides a considerable number of resources and documentation, as well as additional parts for the ODROID N2+.

3.3.1.5 Single Board Computer Comparison

In section 3.2.1 we explained the predominant features that we're seeking out of the single board computer that we plan to utilize. After establishing the key features, we then delved into the specifications of each single board computer that was on our radar. In this section, we're going to directly compare each of the single-board computers that we mentioned previously to determine which is the best fit for TrackPack.

3.3.1.5.1 Processing Power

As aforementioned, while TrackPack doesn't require a substantial amount of processing power to efficiently run the sensors, we do need additional processing power in comparison to what typical microcontrollers can provide to incorporate the display, OBD2 integration, GPS, and image processing simultaneously. Each of the CPUs and GPUs from our four single board computer choices will be evaluated.

Single Board Computer	CPU Clock Frequency	GPU Clock Frequency
NVIDIA Jetson Nano	1.43 GHz	640MHz
Asus Tinker Board S R2.0	1.8GHz	600MHz
Raspberry Pi 4 Model B	1.5GHz	500MHz
HardKernel ODROID N2+	2.4Ghz (Cortex-A73) 2.0Ghz (Cortex-A53)	800MHz

Table 11: Single Board Computer Processing Power Comparison

By evaluating the chart above, the clear winner is the HardKernel ODROID N2+. While clock frequency generally means the processor is faster and capable of executing more cycles per second, other factors can play a role in CPU and GPU speed.

3.3.1.5.2 Form Factor and Power Consumption

TrackPack strives to be as low power as possible to achieve the maximum battery life. In this section, we will evaluate the power requirements of each single board computer, as we'll as their power consumptions.

Single Board Computer	Recommended Voltage (V)	Recommended Amperage (A)	Power Consumption (W)	Form Factor
NVIDIA Jetson Nano	5V	3A	5W – 10W	3.94in x 3.15in
Asus Tinker Board S R2.0	5V	3A	3.5W – 5W	3.37in x 2.13in
Raspberry Pi 4 Model B	5V	3A	2.7W – 6.4W	3.35in x 2.20in
HardKernel ODROID N2+	12V	2A	2.2W – 6.2W	3.54 in x 3.54in

Table 12: Single Board Computer Form Factor and Power Consumption Comparison

By evaluating the chart above, while the HardKernel ODROID N2+ has the lowest power consumption, it also has the greatest voltage demand from the power supply. The NVIDIA Jetson Nano, Asus Tinker Board S R2.0, and Raspberry Pi 4, all utilize similar power supplies, however, the Raspberry Pi 4 has the lowest power consumption. In terms of form factor, the Asus Tinker Board S R2.0 consumes the least area, with the Raspberry Pi 4 following closely behind.

3.3.1.5.3 Cost and Support

In addition to being as low power as possible, TrackPack also strives to be as cost effective as possible. However, cost effectiveness does not directly correlate to choosing the cheaper option regardless of specification. In this section, support will not only be evaluated on how many resources can be found from the manufacturer or other resources solely, but also by the extent of the aftermarket support for each single board computer. The support evaluation will be placed on a point scale in relation to the other single board computers and used to determine the winning board in the next section.

Single Board Computer	Price	Support
NVIDIA Jetson Nano	$149.00	3
Asus Tinker Board S R2.0	$142.99	2
Raspberry Pi 4 Model B	$55.00	4
HardKernel ODROID N2+	$83.00	1

Table 13: Single Board Computer Cost and Support Comparison

3.3.1.5.4 Final Verdict

In this section we'll grade each of the single board computers based on their placements in the four categories (processing power, form factor, power consumption, cost, and support). Depending on where each single-board computer ranked in each category respectively, they will be assigned a value between one and four to be totaled.

Single Board Computer	Cost	Support	Processing Power	Form Factor	Power Consumption/Supply	Total Points
NVIDIA Jetson Nano	+1	+3	+2	+1	+2	9
Asus Tinker Board S R2.0	+2	+2	+3	+4	+3	14
Raspberry Pi 4 Model B	+4	+4	+1	+3	+4	16
HardKernel ODROID N2+	+3	+1	+4	+2	+1	11

Table 14: Single Board Computer Final Verdict

After carefully reviewing each single-board computer and assessing them by cross-referencing with the demands required by TrackPack, we've decided to utilize a Raspberry Pi 4 Model B 4GB. We feel that while the Raspberry Pi 4 may fall slightly short in processing power, it makes up for it substantially in the remaining categories.

3.3.2 Global Navigation Satellite System (GNSS)

GNSS Implementation is the most crucial aspect of TrackPack. It's important that TrackPack measures GNSS with a high level of precision and relatively low latency so that we can effectively find ideal values such as the distance travelled along with the speed travelled. We can use this data to extrapolate the remaining parameters we intend to measure, such as ¼ mile time, ⅛ mile time, 0 – 60 mph time, 60 – 130 mph time, lap time, etc. In this section, we'll discuss the different GNSS technologies and dive into different GNSS modules to decide which of these would be the best fit for TrackPack.

Signals Bands

It's important to understand that when people refer to GPS, they are just referring to the GNSS signal band that is typically used in North America.

There are seven major GNSS signal bands which include:

- GPS
- GLONAS
- Galileo
- BeiDou

- NAVIC
- SBAS
- QZSS

It's crucial that the amount of different GNSS signal bands be evaluated because many of the potential GNSS modules that will be evaluated in this section are capable of concurrent reception from multiple GNSS signal bands, the use of concurrent reception improves position accuracy, position availability, and reliability.

Update Rate

Update rate in GNSS modules refers to how often the module calculates and reports its position. The update rate is measured in Hertz (Hz) and the standard rate for most devices is 1Hz, which means that the device calculates and reports its positions every second.

Position Accuracy

While we discussed accuracy briefly when introducing signal bands, each GNSS module has specifications on the level of precision their module can provide. Typically, the position accuracy is measured in CEP or Circular Error Probable. Essentially the GNSS module manufacturer will specify the amount of meters CEP, what this means it that within the specified meter diameter, there is a 50% probability that your position is within the circumference.

Acquisition

When it comes to GNSS, each module has a specific acquisition time whether it is performing a cold start or performing a reacquisition if the signal is lost for some reason. While substantially low acquisition times aren't crucial due to the nature of TrackPack, reducing the acquisition time in general provides a more well-round, easy-to-use product.

Sensitivity

There are three primary types of sensitivities with most GNSS modules including: Acquisition, Tracking, and Navigation. Sensitivity is measured in dBm or decibel-milliwatts which is the power ratio in decibels in relation to one milliwatt. In relation to GNSS sensitivity, the closer the sensitivity is to zero, the stronger the signal strength is. Acquisition sensitivity refers to the minimum power level at which a GNSS module can simply get a position, commonly in a cold start situation where the module is required to initially search for all the satellites in view and connect. Tracking sensitivity refers to the minimum power level at which a GNSS module can maintain a connection to one or more satellites. Navigation sensitivity refers to the minimum power level at which a GNSS module can maintain a connection to one or more satellites to provide an accurate location while navigating. Since TrackPack will not be utilizing the GNSS module to perform and navigation but rather just calculating metrics, the navigation sensitivity will be omitted from the GNSS module evaluation.

3.3.2.1 U-blox ZED-F9P

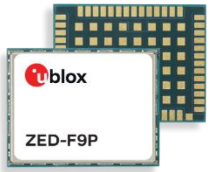


Figure 21: Ublox ZED-F9P

The U-blox Zed-F9P is a high precision GNSS module capable of concurrent reception from four GNSS bands including: GPS + QZSS / SBAS, GLONASS, Galileo, and BeiDou. The U-blox Zed-F9P comes in at $199.00 from Digikey and combines multi-band GNSS with Real Time Kinematics (RTK) for extreme precision. The U-blox Zed-F9P has an update rate of up to 20Hz and a position accuracy of 0.01m + 1 ppm CEP. This reference to position accuracy correlates to the distance travelled, implying that this GNSS module is accurate to 10mm CEP with an additional 1mm CEP accuracy variation per kilometer travelled. The U-blox Zed-F9P has a cold start acquisition time of 24s and a reacquisition time of 2s. The sensitivities of the U-blox Zed-F9P are -148dBm for acquisition and -167dBm for tracking. Finally, the U-blox Zed-F9P requires between 2.7V and 3.6V and consumes 68mA at 3V.

3.3.2.2 U-blox CAM-M8C

Figure 22: Ublox CAM-M8C

The U-blox CAM-M8C is a standard precision GNSS module capable of concurrent reception from three GNSS bands including: GPS / QZSS, GLONASS, Galileo, and BeiDou. The U-blox CAM-M8C comes in at $27.00 from Digikey and has an update rate of up to 18Hz when only connected to a single GNSS band and 10Hz when utilizing 2 concurrent bands. The U-blox CAM-M8C has a position accuracy of 2.5m CEP and has a cold start acquisition time of 26s and a reacquisition time of 1s. The sensitivities of the U-blox CAM-M8C are -148dBm for acquisition and -164dBm for tracking. Finally, the U-blox CAM-M8C requires between 1.6V and 3.6V and consumes 28mA at 3V.

3.3.2.3 Quectel L26-T

Figure 23: Quectel L26-T

The Quectel L26-T is another standard precision GNSS module capable of concurrent reception from three GNSS bands including: GPS / QZSS, GLONASS, Galileo, and BeiDou. The Quectel L26-T comes in at $39.68 from Digikey and has an update rate of up to 5Hz. The Quectel L26-T has a position accuracy of 1.5m CEP and has a cold start acquisition time of 35s and a reacquisition time of 2s. The sensitivities of the Quectel L26-T are -145dBm for acquisition and -162dBm for tracking. Finally, the Quectel L26-T requires between 3.0V and 3.6V and consumes 75 – 80mA at 3.3V.

3.3.2.4 GNSS Comparison

In section 3.2.2 we explained the predominant features that we're evaluating out of the GNSS module that we plan to utilize. After establishing the key features, we then delved into the specifications of each GNSS module that was on our radar. In this section, we're going to directly compare each of the single-board computers that we mentioned previously to determine which is the best fit for TrackPack.

3.3.2.4.1 Signal Bands

We've come to the consensus that access to more GNSS signal bands increases position accuracy and availability, as well as overall reliability.

GNSS Module	Number of Concurrent Bands	GPS	GLONASS	Galileo	BeiDou	NAVIC	SBAS	QZSS
U-blox ZED-F9P	4	✓	✓	✓	✓		✓	✓
U-blox CAM-M8C	3	✓	✓	✓	✓			✓
Quectel L26-T	3	✓	✓	✓	✓			✓

Table 15: GNSS Signal Bands Comparison

Evaluation of the chart above shows that not only does the U-blox ZED-F9P have access to an additional signal band in comparison to the U-blox CAM-M8C and the Quectel L26-T, but it also is able to concurrently connect to an extra band as well.

3.3.2.4.2 Update Rate and Position Accuracy

The update rate and position accuracy are some of the more important GNSS factors that we'll be evaluating. Typically, a substantially higher update rate isn't necessary unless you need to track high speed metrics where the location can change drastically in the fraction of a second. While our location won't be changing at an extreme rate, it's important that we keep the update rate on the higher end to increase precision in TrackPack's analytics. It's also important that TrackPack can track your position accurately to ensure that the measurements are as correct as possible so we can extrapolate the data we need with a high level of precision.

GNSS Module	Update Rate	Position Accuracy
U-blox ZED-F9P	20Hz	0.01m + 1 ppm CEP
U-blox CAM-M8C	10Hz	2.5m CEP

Quectel L26-T	5Hz	1.5m CEP

Table 16: GNSS Update Rate and Position Accuracy Comparison

The chart above shows that the U-blox ZED-F9P has a substantially higher update rate, which is two to four times greater than its competition. The chart also reveals that its position accuracy is substantially greater than that of the U-blox CAM-M8C and the Quectel L26-T.

3.3.2.4.3 Acquisition Time

Acquisition time is one of the factors we're evaluating that isn't a core factor to TrackPack, however, faster acquisition times will lead to a more enjoyable, portable, and ready to use product.

GNSS Module	Acquisition Time	Reacquisition Time
U-blox ZED-F9P	24s	2s
U-blox CAM-M8C	26s	1s
Quectel L26-T	35s	2s

Table 17: GNSS Acquisition Time Comparison

We can see from the chart above, that acquisition times between the U-blox ZED-F9P and U-blox CAM-M8C are relatively similar, while the acquisition time of the Quectel L26-T falls behind a bit.

3.3.2.4.4 Sensitivity

Similarly, sensitivity is another factor that is not a core factor when deciding on a GNSS module for TrackPack.

GNSS Module	Acquisition Sensitivity	Tracking Sensitivity
U-blox ZED-F9P	-148dBm	-167dBm
U-blox CAM-M8C	-148dBm	-164dBm
Quectel L26-T	-145dBm	-162dBm

Table 18: GNSS Sensitivity Comparison

The chart above shows that the Acquisition and Tracking Sensitivities are similar across all three GNSS modules, upon further research it appears that this sensitivity range is about industry standard for most GNSS modules. For this reason, sensitivity will not play a role in deciding which GNSS module will be used for TrackPack.

3.3.2.4.5 Supply Voltage, Power Consumption, and Price

The last and final category evaluates the supply voltage, power consumption, and price. Power consumption and price are what we're predominantly considering in this category based on the design constraints of TrackPack.

GNSS Module	Supply Voltage	Power Consumption	Price
U-blox ZED-F9P	2.7V – 3.6V	68mA at 3V	$199.00
U-blox CAM-M8C	1.6V – 3.6V	28mA at 3V	$27.00
Quectel L26-T	3.0V – 3.6V	75 – 80mA at 3.3V	$39.68

Table 19: GNSS Supply Voltage, Power Consumption, and Price Comparison

3.3.2.4.6 Final Verdict

From the chart above, we can see that the supply voltages for all three GNSS modules are similar, with the U-blox CAM-M8C being capable of operating on lower voltage. Power consumption between the U-blox ZED-F9P and the Quectel L26-T are similar, however, the U-blox CAM-M8C really sets itself forward here by being more than half as power efficient than its competition. The U-blox ZED-F9P came in between five and eight times expensive as the Quectel L26-T and the U-blox CAM-M8C respectively. After comparing each aspect of the three GNSS modules, it's clear that the U-blox ZED-F9P is a great GNSS module and has extreme accuracy and precision and relatively low power consumption, however, due to the design requirement and restraints of TrackPack, the U-blox ZED-F9P is slightly overkill in both performance and price. For this reason, we decided to choose the U-blox CAM-M8C.

3.3.3 Display

After deciding on which single board computer to use for TrackPack, it was important to select an accommodating display. During the initial inception of TrackPack, we decided that we wanted TrackPack to have as small a footprint as possible, like other products on the market. Ideally, TrackPack would contain no display and deliver all its information remotely and directly to a phone application. TrackPack would be easy to use and control with a few buttons on the device itself, and the rest of the controls and functionality to be performed through the app.

In this section, we will compare different displays to use for TrackPack. We carefully analyzed various aspects such as dimensions, resolution, weight, cost, and compatibility.

3.3.3.1 Miuzei 4in Touchscreen Display

The Miuzei Raspberry Pi 4 Touchscreen display is a compact and convenient display screen with Raspberry Pi 4 Model B compatibility. The Miuzei is a 4-inch display, and it is a compact and affordable display. The display features a 4-inch screen with a refresh rate of 60 FPS and a resolution of 800 x 480 pixels, which makes it ideal for a variety of applications, including embedded systems, IoT devices, and other projects. This display has small dimensions of 3.86 x 2.28 inches.

The touch function is only supported by Kali, Raspbian, Ubuntu, Octopie. The 4-inch display monitor with touch control function works upon touch drive installation. The Miuzei is equipped with a resistive touch screen. This is a slight drawback since this screen has a layer of conductive material on top of the screen which can reduce the clarity of the display, it’s not as accurate as a capacitive screen, and this resistive screen relies on pressure to register a touch. This display can be used as standard HDMI output device for computer display with no touch function.
The Miuzei is equipped only with HDMI display connectivity to the Raspberry Pi 4 and supports multiple operating systems, including Raspbian, Kali, Octopi, Linux, and Ubuntu. A potential downfall to the HDMI connectivity is that the display will be using majority of the GPIO pins. With this type of connectivity, this limits users from implementing other units such as an IMU.

From a technical standpoint, the Miuzei Raspberry Pi 4 Touchscreen 4-inch display is easy to set up and use. The display connects directly to the Raspberry Pi 4 via a HDMI port and is powered by the Raspberry Pi. The Miuzei is a lightweight display weighing in at 7.4 ounces. One potential drawback of the Miuzei Raspberry Pi 4 Touchscreen 4-inch display is its relatively small display dimensions. While the display is suitable for many projects, its 4-inch screen may be too small for some applications, especially those that require more detailed graphics or a larger viewing area. Additionally, the 800 x 480-pixel resolution is not as high as some other displays on the market, which may impact the clarity of text and graphics.

Consumers searching for a small and inexpensive display for your Raspberry Pi projects, the Miuzei Raspberry Pi 4 Touchscreen with its 4-inch display is a good option to consider, especially with at a low cost of $34.99. It might not be perfect for every situation, but its touch screen feature works well and it's easy to use, which is why it has become quite popular among users.

3.3.3.2 Elecrow 5in Touchscreen Display

The Elecrow display, like the Miuzei, is Raspberry Pi compatible. The Elecrow display has a 5-inch display with a 800 x 480 pixel resolution. The Elecrow is a lightweight display weighing in at 3.99 ounces. With compatibility to the Raspberry Pi, the touch screen feature allows for easy navigation and interaction with the Raspberry Pi device.

The Elecrow display comes equipped with a plug and play functionality which adds to the ease of use of this display. There is a direct connectivity between the Raspberry Pi and the display through HDMI connectivity. With the HDMI display connectivity, this limits the user from utilizing the GPIO pins for any additional uses. Consumers may find this limitation a determining factor of whether this is the right display for them.

With the small display of 5-inches, some consumers may find this size to bear limitations for their use. The display has a refresh rate of 60 FPS and it is equipped with an LCD display. This display has dimensions of 4.72 x 2.76 x 0.35 inches. The Elecrow display is compact and portable, making it ideal for users who need a display that they can take on the go. However, the small size of the display may be a limiting factor for some users, particularly those who need a larger display for their projects. Additionally, the resistive touch display may not be as durable as other types of touch displays, and it does not support multi-touch gestures and decreased durability. Compared to many displays on the market, the Elecrow is relatively affordable with a cost of $43.99. Consumers may find this display preferable due to its cost, portability, and compatibility.

3.3.3.3 Raspberry Pi 7in Touchscreen Display

At its core, the Raspberry Pi display is a 7-inch LCD display with a resolution of 800 x 480 pixels. Display can provide a clear and detailed view of your Raspberry Pi's output, whether you're running a graphical user interface or a command-line interface. The display is also capacitive, so this may be preferable to users since the screen can easily detect touch input from your fingers, it is more durable than a resistive screen, and it has a faster response time.

A primary advantage of the Raspberry Pi display is that that display is connected through a display ribbon, therefore utilizing minimal GPIO pins. For consumers with a more sophisticated intent for their project may find this to be very beneficial. This is important if you want to use those pins for other things, like controlling motors or sensors. So if you're working on a project that requires a lot of pin usage, the Raspberry Pi display can be a really useful addition. This display is about 3.94 x 2.99 x 0.79 inches and weighs in about 13.4 ounces. Given that this display has a greater screen size than the Miuzei and the Elecrow, the Raspberry Pi will be heavier, but the bigger screen size can be a real advantage if you need to see a lot of information at once.

Consumers may find factors such as compatibility and the large display with capacitive touch very much preferred over other displays. For these enhanced features, the cost of $60.00 may be worth it to consumers.

3.3.3.4 Display Comparison

3.3.3.4.1 Build and Cost Comparison

Below is a chart comparing the overall build and cost of each display. As we compare these displays, we aim to determine which display is most ideal. We hope to find a display that aligns with our cost-effective and compact build of TrackPack.

Display Build and Cost Comparison			
Display	Miuzei	Elecrow	Raspberry Pi
Display Size	4 inches	5 inches	7 inches
Weight	7.4 ounces	3.99 ounces	13.4 ounces
Dimensions	3.86 x 2.28 inches.	4.72 x 2.76 x 0.35 inches	3.94 x 2.99 x 0.79 inches
Cost	$34.99	$43.99	$60.00

Table 20: Display Form Factor and Cost Comparison

3.3.3.4.2 Connection Type and Display Type Comparison

Below is a chart comparing the connectivity and display types of each display. Ease of use is an important factor we would like to consider for our consumers. We would also like to factor in the most effective set up as part of our decision.

Display Build and Cost Comparison			
Display	Miuzei	Elecrow	Raspberry Pi
Connectivity	HDMI	HDMI	DSI
Touchscreen Type	Resistive	Resistive	Capacitive

Table 21: Display Connection Type and Display Type Comparison

3.3.3.4.3 Final Verdict

In this section we will grade each of the displays based on their placement in the 3 categories (cost, connectivity, and touchscreen type.). Depending on where each display is ranked in each category respectively, they will be assigned a value between one and four to be totaled.

Display	Cost	Connectivity	Touchscreen Type	Total Points
Miuzei	+3	+1	+1	5
Elecrow	+2	+1	+1	4

Raspberry Pi	+1	+3	+3	7

Table 22: Display Final Verdict

We decided that a small 2x16 display would assist the user to flip through menus and perform more functionality directly on the TrackPack itself as opposed to the phone application, furthermore, this would reduce the amount of development need in the mobile app. Well, as the research began and TrackPack started to come to life, the ideas grew larger. TrackPack didn't aspire to be as good as its competitors, it aspired to be greater. To alleviate the development of a mobile app to control TrackPack, we decided to incorporate a larger touchscreen digital display, all TrackPack's parameters can be accessed as easily as possible.

We decided to utilize the official Raspberry Pi 7in touchscreen LCD display. This display is easily integrable with the Raspberry Pi 4 using only two connections: power is taken from the GPIO port, and a ribbon cable connects to the DSI port. The display comes with the necessary adapter board that handles the power and signal conversion and has a display resolution of 800 x 480.

Concerns based on the size of the display have arisen. TrackPack was intended to be mounted directly on the windshield, this way the camera has a clear field of view through the windshield, and the user has a clear field of view of the display. While the large display would make parameters easy to see, concerns circled around whether the large display would hinder the driver's field of view of the road. We won't know with certainty if the driver's field of view is hindered until TrackPack is assembled and mounted in a vehicle, however, to address this concern should it arise, we came up with a potential solution. Our solution is to simply mount TrackPack somewhere else, either on the side of the windshield and out of focus, or even somewhere on the dashboard of the vehicle. Mounting the TrackPack somewhere out of focus creates a new issue, the camera would not have a clear view through the windshield. This issue can easily be mitigated with the use of an external camera that can be mounted on the windshield. We're hoping that these design concerns don't arise because they would conflict with TrackPack's portability aspect.

3.3.4 OBD-II

The On-Board Diagnostics II, otherwise known as the OBD II, is a vehicle diagnosis system typically found in vehicles of the year 1996 or newer. The OBD II monitors specific parameters of a vehicle such as:

- Real-time parameters: RPM, speed, pedal position, spark advance, airflow rate, coolant temperature, etc.
- Status of "Check Engine" light
- Emission readiness status
- Freeze frame: a "snapshot" of parameters at the time a trouble event has occurred.

- Diagnostic trouble codes (DTCs).
- Oxygen sensor test results
- Vehicle Identification Number (VIN)
- Number of ignition cycles
- Number of miles driven with MIL on

The OBD II port is a standardized interface that makes it easier for the Electronic Control Unit (ECU) of a vehicle to communicate with an outside diagnostic instrument. The OBD-II port is a vital diagnostic component that enables in-car real-time system monitoring. It is often found underneath the dashboard on the driver's side of the car. In essence, the OBD II connector provides access to a vehicle's internal computer systems. A modern car's ECU handles a variety of tasks, including keeping track of the engine's performance, pollution levels, fuel economy, and several other systems. The OBD II port provides access to a substantial amount of information from the onboard computer from the vehicle. Using a OBD II scanner versus an OBD II reader can extend the capabilities we have. With a reader, this is a less expensive option where we can only retrieve diagnostic trouble codes from the engine control unit, depending on the type of OBD II reader you have, we may or may not have the ability to clear the trouble code(s). The less expensive OBD II tool has its limitations with the type of data we can access and the lack of information about specific manufacturer codes. With a scanner, the cost does increase so do the capabilities. The capabilities with a scanner are extended with greater access to a vehicle’s diagnostic data, system, and features. A scanner provides the ability to clear diagnostic trouble codes, it displays of live data from various sensors, facilitate advanced troubleshooting options, and they can read manufacturer specific codes. An OBD II scanner is best suited for the functionality of TrackPack because of the capability of reading and displaying real time data. With the extensive parameters provided by a scanner, this supports the overall goal of TrackPack: to provide detailed feedback of a driver’s vehicle performance so that the driver can decide on potential modifications to their vehicle or determine if there needs to be adjustments in the driver’s performance on the track. As we researched more on OBD II scanners, we discovered that there are various types of scanners that support different types of connections such as Bluetooth, serial connections, and wi-fi. This realization prompted us to evaluate and carefully select the most suitable OBD II scanner for our TrackPack project.

The OBD II gathers information from several sensors and monitoring devices installed in the engine and other components of a vehicle. A standard pinout, or wiring diagram, for the OBD II port enables the diagnostic tool to connect with the car's onboard computer and gather DTCs.

Each OBD II port pin's (16 pins) function is specified by the OBD II pinout. Each pin on the OBD II port provides a different purpose. Two different types of OBD II port connectors include a wired connector and a Bluetooth adapter. When it comes to determining if a wired or wireless option is best, in the case of the TrackPack, the wireless option is preferred. The wireless connection eliminates the unnecessary restraint a cable connection would have. Consumers can move freely within and around their vehicle without being confined to the TrackPack in the vehicle.

The OBD II is important to the design of TrackPack because this is where the TrackPack will collect performance parameters from a user's vehicle. The data collected will be passed on to the user so that they can be informed of how their vehicle is performing. This will help users determine which modifications, if any, are needed to acquire the desired performance.

A wireless communication module, memory, and a microprocessor are all parts of a Bluetooth OBD II scanner. The device's microcontroller oversees processing data obtained from the engine control unit of the car and transmits the data to the user's device, i.e., computer or mobile device. The wireless communication module transmits data wirelessly that is needed to operate the scanner, while the memory stores the software and transfers the data wirelessly to the external device.

3.3.4.1 ELM327 Bluetooth Adapter

Figure 25: ELM327 Bluetooth Adapter

The ELM327 Bluetooth adapter is compatible with smartphone devices and tablets. The ELM327 has a wired option depending on what the consumer would prefer. The wired option for the ELM327 adapter typically involves a USB connection that allows it to be plugged into a computer or other device. Yet, because of its practicality and use, the Bluetooth Adapter is typically more widely used. The ELM327 Bluetooth adapter has been advertised as having a primary benefit of its ease of use. The ELM327 Bluetooth Adapter is designed with extended compatibility with various mobile applications and software. Some mobile applications include Torque Pro, Car Scanner ELM OBD 2, and OBD Fusion. With its compatibility, users can read, interpret, and analyze the data gathered from the vehicle's engine control unit.

The ELM327 Bluetooth Adapter has advanced scan capabilities compared to other Bluetooth OBD II adapters. This module specifically has comprehensive diagnostic capabilities, including the ability to reset the Check Engine Light and monitor emissions systems, as well as read and clear diagnostic trouble codes (DTCs), see real-time data, including engine speed, throttle position, and coolant temperature. Since the ELM327 Bluetooth adapter is compatible with various third-party software applications, this allows it to perform even more advanced functions such as logging data, performing custom tuning, and even displaying data on a dashboard in real-time.

The ELM327 Bluetooth Adapter is a small compact device constructed of plastic. The device has indicators such as LED lights that let users know when the device is turned on and in communication with their mobile device. Also, the adapter is made to be plug-and-play, so the mobile device does not need to have any additional software or drivers loaded to use it. The ELM327 Bluetooth adapter is best known and preferred because of its low cost and dependability.

3.3.4.2 Panlong

The Panlong OBD II scanner works similarly to the ELM327 Bluetooth Adapter. While the Panlong OBD II scanner is connected to the OBD II port, it is powered directly, so this eliminates the need for an external power source. The Bluetooth adapter is convenient to use since there are no strict restrictions on being tethered to the vehicle. Like many others, the Panlong OBD II scanner is a compact and lightweight module that can easily be stored. The module is compatible across many platforms that include Windows, Android, and iOS. Some modules on the market are restricted to either iOS platforms/Android platforms or just PC platforms.

Figure 26: Panlong Bluetooth Adapter

The Panlong adapter has reading capabilities for basic information which can be useful for diagnosing simple issues with the vehicle. The Panlong adapter is a more basic OBD II scanner that is designed primarily for reading and clearing diagnostic trouble codes (DTCs It can also display some real-time data such as vehicle speed, RPM, and engine load, but it does not have the same advanced diagnostic capabilities as the ELM327 Bluetooth adapter. The Panlong scanner does not have the advanced diagnostic capabilities such as resetting the Check Engine Light, monitoring emissions systems, or performing custom tuning. This limits its usefulness for more advanced diagnostics and repairs.

The Panlong adapter is a more basic tool that is primarily designed for reading and clearing DTCs. This module is an ideal choice for consumers that need simple and straight forward readings from the scanner. Users that have a minimal requirement of obtaining basic diagnostics on a vehicle may choose this option, especially when factoring in the affordability of this simple device.

3.3.4.3 OBDLink MX+

Figure 27: OBDLink MX+ Bluetooth Adapter

The OBDLink MX+ is one of the more advanced Bluetooth OBD II scanners on the market. The Bluetooth scanner carries capabilities such as enhanced trouble codes and enhanced parameters (PIDS). The scanner allows for PIDs on the supporting mobile applications to be graphed, added to the dashboard, and viewed on the data grid page. The OBDLink MX + is supported on platforms such as iOS, Android, and Windows. With secure 128-bit data encryption, the Bluetooth OBD II scanner is advertised to be hacker-proof. advanced security technology. on the OBD II scanner sets it apart from many others. Many Bluetooth OBD II scanners require a pairing pin on initial setup, the pin is typically advertised on the website, and it is as basic as 1234 or 0000. The OBDLink MX+ employs an innovative multi-layered link security mechanism that eliminates the possibility of security breaches from unapproved users.

The OBDLink MX + allows users to turn off the check engine light, and erase stored diagnostic information, read and erase stored and pending trouble codes (both generic and manufacturer-specific), access freeze frame information, display, graph, and log 90+ real-time parameters, create custom digital dashboards, measure, and display fuel economy, etc. The OBDLink MX+ is one of the more expensive scanners on the market due to the sophisticated capabilities and features it has. Unlike other scanners that are low-cost, the OBDLink MX+ can read and clear trouble codes such as ABS, airbag, transmission, body control. The scanner provides access to many other parameters and sensors that the typical OBD II doesn't offer. The mobile application that the OBDLink MX+ is supplied with the following features as advertised on their website: provides advanced diagnostics, trip logging, multi-parameter graphing, customizable gauges, 0-60 times, 1/4-mile performance, Freeze Frame, SMOG readiness, over-voltage protection to prevent electrical fires, firmware updates, and Dropbox support. The Bluetooth scanner supplies a power saving sleep mode, BatterySaver, so that users may leave their OBD II scanner plugged into the OBD II port without worrying about depleting the vehicle's battery.

3.3.4.4 Veepeak

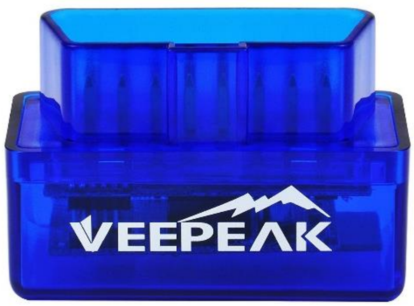


Figure 28: Veepeak Bluetooth Adapter

Veepeak is a company that specializes in automotive tools and accessories. The OBDCheck VP11 is Veepeak's Bluetooth OBD II scanner. With this Bluetooth module, users can view car performance data, but there are some limitations that apply. OBDCheck VP11 is only compatible with Android devices and Windows PC only. They do offer an iOS compatible product which is the Wi-Fi version OBDCheck BLE. Accompanying the OBDCheck VP11, users need a third-party OBD II application such as OBD Fusion, Car Scanner ELM OBD2, Dr. Prius and DashCommand. Upon installing this mobile application, users then have access to the features of the OBDCheck VP11.

The OBDCheck VP11 can read, and clear fault codes related to the engine, transmission, and emission systems, and display live sensor data such as engine RPM, coolant temperature, fuel system status, oxygen sensor readings, throttle, boost, speed, fuel trim, and more. It can also perform smog tests by checking the readiness of the vehicle's emission control system. Some limitations apply such as the module may not be able to scan certain proprietary systems or modules that are not part of the OBD II standard. The OBDCheck VP11 may not be able to read ABS (anti-lock brake system) codes or airbag codes on some vehicles. Additionally, some advanced features, such as bi-directional control or programming, may not be supported.

3.3.4.5 OBD-II Bluetooth Adapter Comparison

3.3.4.5.1 OBD-II Scanner Protocol Compatibility Comparison

We would like TrackPack to be versatile and increase compatibility possibilities. In this section, we compare each Bluetooth OBD II scanner protocol compatibility to determine which device is best.

Bluetooth OBD II Scanner Protocol Compatibility Comparison				
	ELM327	Panlong	OBDLink MX+	Veepeak
SAE J1850-PWM	✓	✓	✓	✓
SAE J1850-VPW	✓	✓	✓	✓
ISO 9141-2	✓	✓	✓	✓

ISO 14230-4 (slow)	✓	✓	✓	
ISO 14230-4 (fast)	✓	✓	✓	
ISO 15765-4 (CAN)	✓	✓	✓	✓
SAE J2411 (SWCAN)	✓		✓	
SAE J1939	✓			

Table 20: OBD-II Bluetooth Adapter Protocol Compatibility Comparison

Based upon the chart, it demonstrates the ELM327 Bluetooth OBD II scanner possesses the most protocol compatibility in comparison to the rest of the devices.

3.3.4.5.2 Build and Cost Comparison

Below is a chart comparing the overall build and cost of each Bluetooth scanner. We aim to develop a cost-effective device while also considering the support and specifications of each product.

Bluetooth OBD II Scanner Build and Cost Comparison				
OBD II Scanner	ELM327	Panlong	OBDLink MX+	Veepeak
Weight	0.81 ounces	0.64 ounces	1.2 ounces	1.12 ounces
Dimensions	1.97 x 1.18 x 0.59 inches	1.89 x 0.98 x 1.26 inches	1.97 x 1.77 x 0.91 inches	1.89 x 1.26 x 0.98 inches
Part Number	OTR-000	PL-B02	MX201	VP11
Cost	$13.99	$12.99	$139.95	$13.99

Table 21: OBD-II Bluetooth Adapter Form Factor and Cost Comparison

Overall, each Bluetooth scanner is quite similar in size and weight. The purpose of this comparison chart is to determine the most compact, lightweight, and cost-effective part. With this consideration, we must factor in the specifications and compatibility of each device also.

3.3.4.5.3 Final Verdict

In this section we will grade each of the Bluetooth OBD II scanners based on their placements in the three categories (cost, protocol compatibility, and overall build). Depending on where each Bluetooth OBD II scanner ranked in each category respectively, they will be assigned a value between one and four to be totaled.

Bluetooth OBD II Scanner	Cost	Support	Build	Total Points
ELM327	+3	+4	+4	10
Panlong	+4	+2	+3	9
OBDLink MX+	+1	+3	+1	5
Veepeak	+3	+1	+2	6

Table 22: OBD-II Bluetooth Adapter Final Verdict

TrackPack is meant to be a seamless, lightweight, and easy to use device. We have chosen to research Bluetooth module options for OBD II scanners to achieve ease of use for our consumers. The TrackPack requires more intricate readings than just the basics, therefore the Panlong Adapter is not the optimal choice for this build. The TrackPack requires in-depth analysis to be provided for consumers to get a well-rounded summary of their vehicle's performance. In terms of comparison with the ELM327 Bluetooth adapter and the OBDCheck VP11, they are quite similar. The performance and features of each device are comparable. The main difference is that the OBDCheck VP11 is specifically designed for use with smartphones and tablets and may be more convenient for users who prefer to work with those devices. The ELM327 Bluetooth adapter is a more general-purpose tool that can work with a wider range of devices and may be more versatile unlike the Panlong that is limited to on the OBD II protocols. With a more advanced OBD II scanner like OBDLink MX+, the enhanced features come with a significantly higher cost. The features provided with the OBDLink MX+ are not necessary to the design of the TrackPack. The OBDLink MX+ appears to gear more towards mechanics for vehicle repairment. TrackPack is made to enhance performance on vehicles, not necessarily to repair vehicles.

After careful consideration, we finally arrived at the decision to incorporate a Bluetooth scanner into our design, with the aim of adding to the seamlessness and convenience of the TrackPack user experience. After researching different types of OBD II Bluetooth scanners, we came across many types of scanners, and we decided to use the ELM327 connector which comes with a serial connection option and a Bluetooth option. We chose the ELM327 Bluetooth OBD II scanner since it was the most cost-effective option that as a wider range of protocol compatibilities.

3.3.5 Inertial Measurement Unit (IMU)

Aside from the GPS, an IMU is how we plan to capture the remaining accelerometer and gyroscope metrics. To further understand the metrics the IMU will be capturing, it's important to establish the fundamentals of each.

Accelerometer

An accelerometer is beneficial to TrackPack because we can measure the acceleration forces acting on the vehicle. Accelerometers measure the gravitational forces, g-forces, that act on the vehicle when accelerating, braking, and cornering. Measurement of these forces is important to determine the impact on the stability, speed, and maneuverability of the vehicle. Drivers can use these parameters to improve their driving and vehicle performance. Specifically with the measurement of the lateral G-forces during cornering, readings as such can assist the driver in determining the optimal speed and trajectory through a corner. Understanding the accelerometer readings can help vehicle builders to perform the necessary modifications needed for their vehicle to handle more predictably.

Gyroscope

Like accelerometers, gyroscopes are beneficial to racers, however, the gyroscope poses greater value to off-road vehicle enthusiasts in comparison to on-road vehicle enthusiasts. Nonetheless, road racing vehicles can benefit from gyroscope information to accurately help them track stability, and maneuverability through corners. Furthermore, off-road users benefit from a gyroscope to accurately track the vehicles' angle.

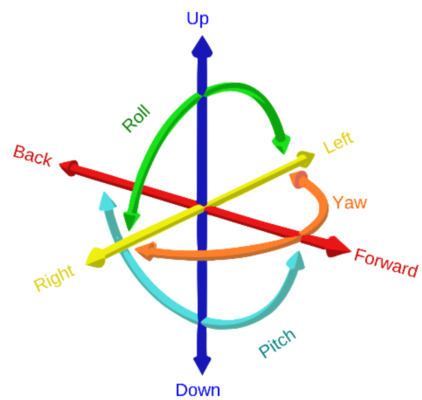


Figure 29: Example of Pitch, Yaw, and Roll

It's important to understand that an object in a three-dimensional space has 6 degrees of freedom (DOF). The object can have a translation movement or rotation movement. Translation movements are up/down, left/right, forwards/backwards. While rotation movements are pitch, yaw, and roll. These 6 degrees of movements are typically tracked using an accelerometer and either a gyroscope or magnetometer. While the three sensors do have some overlap in the parameters they can capture, this overlap helps add accuracy to the data. We know that an object in a three-dimensional space can only move 6DOF. As mentioned previously, the measured parameters from these three sensors overlap with one-another, so if each of the three sensors can determine 3DOF individually, ignoring the fact that each sensor may be calculating a degree of freedom that was already calculated by another sensor, we come to get a total of 9DOF. 10DOF IMU's simply add a barometer. So, a 6DOF IMU captures essentially the same parameters as a 9DOF or 10DOF IMU. Sensor fusion allows us to utilize more sensors in conjunction with one-another, we can mix the data from the different sensors to increase the quality and accuracy of the final measurement, which simply means that 9DOF and 10DOF IMU's are more precise.

3.3.5.1 Adafruit 9-DOF IMU

Figure 30: Adafruit 9-DOF IMU

The Adafruit 9DOF IMU integrates the Bosch BNO055 9DOF sensor that features 9-axis sensor fusion by utilizing a 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer The integrated Bosch BNO055 operates on a voltage range from 2.4V to 3.6V and features digital interfacing through I2C and UART. The accelerometer incorporates programmable ranges ±2g, ±4g, ±8g, and ±16g. The gyroscope incorporates programmable ranges ±250dps, ±500dps, ±1000dps, and ±2000dps. The Magnetometer incorporates a typical measurement range of s ±2500µT. The accelerometer and gyroscope feature selectable low pass filters. The Bosch BNO055 features motion triggered interrupts and multiple modes of operation that give full control of each individual sensor.

3.3.5.2 SparkFun 9-DOF IMU

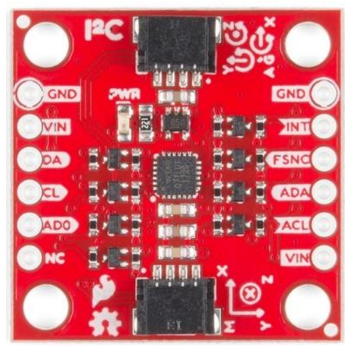


Figure 31: SparkFun 9-DOF IMU

The SparkFun 9DoF IMU integrates the TDK ICM-20948 9DOF sensor that features 9-axis sensor fusion by utilizing a 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer The integrated TDK ICM-20948 operates on a voltage range from 1.71V to 3.6V and features digital interfacing through I2C and SPI. The accelerometer incorporates programmable ranges ±2g, ±4g, ±8g, and ±16g. The gyroscope incorporates programmable ranges ±250dps, ±500dps, ±1000dps, and ±2000dps. The Magnetometer incorporates a typical measurement range of s ±4900µT. The accelerometer and gyroscope feature selectable low pass filters. The TDK ICM-20948 features motion triggered interrupts and multiple modes of operation that give full control of each individual sensor.

3.3.5.3 BerryGPS-IMU GPS and 10DOF

Figure 32: BerryGPS-IMU GPS and 10DOF

The BerryGPS-IMU GPS and 10DOF integrates the ST LSM6DSL sensor that features a 3-axis gyroscope, 3-axis accelerometer, the ST LIS3MDL sensor that features a 3-axis magnetometer, and the Bosch BMP388 barometer. The integrated ST LSM6DSL operates on a voltage range from 1.71V to 3.6V and features digital interfacing through I2C and SPI. The integrated ST LIS3MDL operates on a voltage range from 1.9V to 3.6V and features digital interfacing through I2C and SPI. The accelerometer incorporates programmable ranges ±2g, ±4g, ±8g, and ±16g. The gyroscope incorporates programmable ranges ±250dps, ±500dps, ±1000dps, and ±2000dps. The Magnetometer incorporates a typical measurement range of s ±1600µT. The accelerometer and gyroscope feature selectable low pass filters. The ST LSM6DSL and ST LIS3MDL features motion triggered interrupts and multiple modes of operation that give full control of each individual sensor.

3.3.5.4 IMU Comparison

In section 3.2.5 we described the sensors that TrackPack will be utilizing and why the measurements and level of accuracy from these sensors are important to the functionality of TrackPack. We also introduced the specifications of each IMU. In this section, we'll directly compare each IMU to determine which would be a suitable fit for TrackPack.

IMU	Adafruit 9-DOF IMU	SparkFun 9DoF IMU	BerryGPS-IMU GPS and 10DOF
Accelerometer Programmable Ranges	±2g, ±4g, ±8g, and ±16g	±2g, ±4g, ±8g, and ±16g	±2g, ±4g, ±8g, and ±16g
Gyroscope Programmable Ranges	±250dps, ±500dps, ±1000dps, and ±2000dps	±250dps, ±500dps, ±1000dps, and ±2000dps	±250dps, ±500dps, ±1000dps, and ±2000dps
Magnetometer Measurement Range	±2500µT	±4900µT	±1600µT
Voltage Range	2.4V – 3.6V	1.71V – 3.6V	1.71V – 3.6V 1.9V – 3.6V
Interfacing	I2C and UART	I2C and SPI	I2C and SPI
Low-Pass Filtering	✓	✓	✓
Motion Triggered Interrupts	✓	✓	✓
Selectable Power Modes	✓	✓	✓

Table 23: IMU Comparison

Evaluation of the chart above shows that all three IMU's feature an accelerometer and gyroscope with the same programmable ranges. The SparkFun 9DoF IMU can read greater levels of magnetic fields, making it slightly more precise. The Adafruit 9-DOF IMU interfaces with I2C and UART as opposed to the competitors I2C and SPI interfacing, this breakout board also has a slightly higher minimum voltage. The primary difference is that the BerryGPS-IMU GPS and 10DOF come with an additional barometric sensor integrated, as well as the GNSS module integrated already. If we required a GNSS module with a higher level of precision, the best idea for TrackPack would be to utilize the either of the previously mentioned 9DOF boards from Adafruit or Sparkfun, as their specifications are almost identical, along with the separate high precision GNSS module. However, seeing as we decided to go with the more affordable standard precision level GNSS module, the BerryGPS-IMU GPS and 10DOF are perfectly suitable for TrackPack and we can potentially pull data from the additional barometric sensor it provides. We would like to point out that the most cost-effective route when choosing the IMU is to purchase the IMU and GPS separately, however, we decided that the cost is worth the benefits to maintain a small form-factor and ease of use.

3.3.6 Raspberry Pi High Quality Camera

For the image collection and processing, we used a Raspberry Pi 4 along with the Raspberry Pi High Quality Camera. The sensor used in the camera module is a Sony IMX477 sensor. The IMX477 is a 12.3-megapixel resolution CMOS sensor and measures 7.9 millimeters diagonally. When recording video this sensor can record 2028p x 1080p resolution at 50 frames per second and 1332p x 990p resolution at 120 frames per second. The ability to record at frame rates as high as 120 frames per second will reduce motion blur in the video that would have affected the smoothness and quality of the video. The 1080p resolution will provide more value when the camera is not recording high speed races, for instance in traffic incidents the pixels per inch are more important than the speed at which images are collected. The Raspberry Pi will collect the data from the camera module and combine the data collected from the entire system with the video. The Raspberry Pi is also where the digital image filtering for field curvature distortion could be optimized as an advanced goal.

The sensor includes a C-mount and a CS-mount threaded adapter to connect a lens or lens tube to the sensor. The C-mount connector has a focal distance of 17.525 millimeters and the CS-mount has a focal distance of 12.525 millimeters. Ideally, we would design the system to interface with the CS-mounting hardware, but no commercial optics company sells lens tubes with that thread type. As a result, we chose to design an optical system to connect to the sensor with the CS-mounting standards. To accommodate such a tight focal distance the mount has an adjustable focusing knob with a bandwidth of 10 millimeters. This will allow us a small window of acceptable focal lengths instead of requiring an exact value. The Raspberry Pi camera sensor can output the signal as RAW12/10/8 or COMP8 to the Raspberry Pi 4 computer over the included ribbon cable.

Since we utilized a Raspberry Pi 4 to process the videos and data collected, we have chosen to use an official Raspberry Pi camera module. The goal of choosing an official Raspberry Pi sensor is to guarantee compatibility with the Raspberry Pi in the hopes that it makes integrating the video collection a smooth process. Raspberry Pi newest two varieties of sensors are the High-Quality camera sensor and the Global Shutter sensor. There are two main differences between these sensors, and each has specifications suited to different applications. The High-Quality camera has a sensor resolution of 12 megapixels while the Global Shutter camera has a sensor resolution of 1.2 megapixels. The higher pixel count on the High-Quality camera will be more advantageous to our design because our goal is to collect video with a resolution of 1080p and the 12-megapixel sensor will give greater flexibility if the spot size of the light collected is larger than the pixel size. The second difference between the two sensors is the way they sample each pixel. The High-Quality sensor scans each pixel in each row one at a time to collect the final image frame. The Global shutter sensor is able to capture the data from every pixel at the same time which reduces distortion commonly found on standard sensors. The global shutter is useful for machine vision and high shutter speed photography, but after researching each sensor we have chosen to use the standard High-Quality camera sensors as the high resolution will be more beneficial to our project than the global shutter.

The Raspberry Pi High Quality camera module is available with three different mounting options for our lens array. The three types are the C-Mount, CS-Mount, and M12 mount styles, each of which have slight differences. The biggest difference between all three is the flange focal distance, which simply is the distance between the deepest point of the threading and the sensor’s surface. This variable plays a crucial role in our project because FOV is inversely related to the back focal length. Therefore, to achieve the highest FOV, the back focal distance should be as small as possible so that the rays with the highest incident angle can be focused within the sensors nine-millimeter diagonal size. C-Mount lenses have the longest flange focal dist nmance of 17.526 millimeters. To accommodate this mounting style in our design, the lens array will need to have the largest magnification to focus rays from that far away to the sensor’s less than one centimeter size. The next mounting style available is the CS-Mount, with a flange focal distance of 12.526 millimeters. This shorter focal distance will allow our design to use fewer optics to focus the image on to the sensor. Finally, the M12- mount has no standard flange focal distance, but with the Raspberry Pi High-Quality sensor the flange focal distance can be as low as four millimeters. This would be the most ideal mounting option if our budget was unrestricted due to the inverse relation of focal length and FOV, however it would require a more specialized design. We have chosen to utilize the CS-mounting style as it provides the best balance between the highest achievable FOV and our available budget. Another factor that affected our choice is how we mounted our lens array to the sensor. Our goal is to 3D-Print the mounts to secure each lens, and due to the CS-Mounts larger diameter it will give up greater tolerance in our design to accommodate any printing or design errors.

3.3.7 Storage

The TrackPack will be designed to collect vehicle parameters and record footage so choosing the best storage option is important. Upon selecting the Raspberry Pi 4 Model B to be the optimal choice for the TrackPack design, the unit does not provide internal storage. The single board computer offers a microSD slot and USB 3.0 port. In this section, we will discuss the two options for external storage, microSD card storage and USB drive storage.

3.3.7.1 microSD Card

The microSD card is a simple, lightweight device that is commonly used among devices such as smartphones and cameras. Since the microSD card is small, they are inherently portable and minimalistic for storing data and transferring data between devices. Considering these advantages, it's especially a favorable option for consumers since they are cost-effective. The microSD card is supported among an extensive variety of applications and requires minimal power.

Even though the microSD card seems an ideal option for consumers, there are other factors to consider like the durability of the microSD card. The storage card can be susceptible to electronic corruption, and it can easily be broken. The life span of the microSD card is finite since the technology behind the storage card is flash memory where there are limited read/write cycles, even though they are designed to last for a theoretical limit of 30 years. Consumers may view these disadvantages as miniscule, especially since the lifespan of the storage card is extensive, never mind the cycle limitations. The microSD card is not the fastest on the market when it comes to data transfer speed, but consumers typically don't notice the difference between the speed of microSD card and that of a USB drive.

3.3.7.2 USB Drive

The USB drive is like a microSD card in terms of ease of use and portability. The USB drive can be easily stored for travel, and they are a simple and effective way to save and transmit data. The USB drive may offer more durability compared to the microSD card. The USB driver is encased with either a plastic or metal housing which protects the internals of the USB drive. There are marginal differences between the USB drive's read and write speeds. The USB drive does perform at a fast rate of up to about 40 GB/s depending on which model USB drive consumers use. The USB drives are more resistant to damage, and they are resistant to corruption and data loss. Like any other technology, the USB drive can be more prone to failure over time. USB drives use flash memory technology, so since there are limits on their rea/write cycles, the external storage unit can wear out over time.

3.3.7.3 Storage Comparison

The microSD cards and USB drive are popular storage devices, with notable differences. Both external storage devices differ in size, the microSD card being the smaller device. Some consumers may prefer the smaller storage device because it may be more convenient to use with portable devices. Other consumers may prefer the USB drive since the transfer speed of data is higher than the microSD card and the USB drive is more durable. Depending on the application, consumers would have to decide which external storage device is best for them. In the case of TrackPack, we chose to utilize the microSD card as our external storage simply due to the seamlessness of the device. We see the difference in transfer speeds between the devices to be negligible since this is a device that will be in an active environment. The Raspberry Pi we used has a built in microSD slot and we would like consumers to have a sleek and compact design rather than a USB drive hanging out the side. Since we chose a storage device that gets stored internally, this eliminates the possibility of possible breakage from the USB drive, since it would be open and more susceptible to breaking off. Since the use of TrackPack will be in an extremely active environment, the USB drive would not be ideal.

Interface	Storage Type	Data Transfer Speed
Default	SD, SDHC, SDXC, SDUC	12.5 MB/s
High	SD, SDHC, SDXC, SDUC	25 MB/s
UHS-I	SD, SDXC, SDUC	50 MB/s
UHS-II	SD, SDHC, SDXC, SDUC	312 MB/s
UHS-III	SD, SDHC, SDXC, SDUC	624 MB/s
SD Express	SD, SDHC, SDXC, SDUC	985 MB/s
	USB 2.0	480 MB/s
	USB 3.0	5 GB/s
	USB 3.1	10 GB/s
	USB 3.2	20 GB/s
	USB 4	40 GB/s

Table 24: Storage Specifications

MicroSD Card and USB Drive Comparison				
	SanDisk 64GB Extreme PRO MicroSD	SanDisk 256GB Ultra Luxe USB	SanDisk 256GB Extreme PRO MicroSD	SanDisk 256GB Ultra Luxe USB
Cost	$13.69	$10.23	$33.90	$21.99

Dimensions	0.04 x 0.59 x 0.43 inches	1.57 x 0.62 x 0.23 inches	0.04 x 0.59 x 0.43 inches	1.57 x 0.62 x 0.23 inches
Weight	0.176 ounces	0.279 ounces	0.176 ounces	0.279 ounces
Item Number	SDSQXCU-064G-GN6MA	SDCZ74-256G-G46	SDSQXCD-256G-GN6MA	SDCZ74-256G-G46

Table 25: Storage Comparison

3.3.8 Microcontrollers

When researching into available MCUs, components we were looking for devices that would have sufficient memory, speed, and processing power to be able to run data retrieval components in the Trackpack. Another feature that we would prefer to have in our microcontroller is I²C compatibility. Two options that met our criteria were an MCU from Texas Instruments MSP430 series and the ATMEGA328.

3.3.8.1 MSP 430

The MSP430 series of microcontrollers (MCUs) is a popular family of 16-bit MCUs developed by Texas Instruments. The series is designed to provide low-power performance for a wide range of applications, including industrial automation and remote sensing. One of the key features of the MSP430 series is its low power consumption. The series includes several low-power modes, including standby and shutdown modes, which help to extend battery life in battery-operated devices. The MSP430 also features an ultra-low power consumption mode that can be used to prolong battery life even further.

3.3.8.2 ATMEGA328

ATmega328 is an 8-bit, 28-Pin AVR Microcontroller, manufactured by Microchip, follows RISC Architecture, and has a flash-type program memory of 32KB. Atmega328 is the microcontroller, used in basic Arduino boards i.e., Arduino UNO, Arduino Pro Mini and Arduino Nano. The Atmega microcontroller also has a rich set of development tools and programming software, and has a range of integrated development environments (IDEs) such as Atmel Studio, AVR Studio, and Arduino IDE.

3.3.8.3 Microcontroller Comparison and Selection

We decided to choose the MSP430F168IPM as though it has less speed, it is still significant to relay the information back to the user fast enough for our product. It also has more I/O ports and more memory size for all the different data statistics it will read, and requiring a lower voltage supply with the addition of low power modes to conserve power when not in use and requires a lower Voltage supply than the ATMEGA. From previous courses, we also have experience using the development environment for the MSP430 series, and the language that the MSP430 uses.

Microcontroller	MSP430F168IPM	ATMEGA328P-PU DIP28

Core Size	16-Bit	8-Bit
Speed	8MHz	20MHz
Connectivity	I²C, SPI, UART/USART	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	48	23
Program Memory Size	48KB (48K x 8 + 256B)	32KB (16K x 16)
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V	1.8V ~ 5.5V

Table 26: Microcontroller Comparison

4 Design Constraints and Standards

TrackPack aims to adhere to a specific set of standards for both hardware and software design, which will act as a framework for the project's development. By establishing these standards, we can ensure that the construction and evaluation of TrackPack are carried out in a consistent manner. This will ensure that the project remains focused and on track. TrackPack has set forth to follow a strict set of standards and design constraints relating to both the hardware and the software.

4.1 Standards

TrackPack is a project that aims to provide analytical data about a driver's vehicle, which requires compliance with multiple project standards and regulations. The project must be able to retrieve and provide analytical data about a driver’s vehicle. These standards and regulations are put in place to ensure that the project meets safety and performance requirements. Shown below is a table containing the standards to be implemented in this project.

Section	Standard
4.1.1	PCB
4.1.2	Soldering
4.1.3	12V Car Outlet Power
4.1.4	USB Communication
4.1.5	Chargeable Battery Supply
4.1.6	Optical Mounting Hardware
4.1.7	CSI

4.1.8	HDMI
4.1.9	SD Card
4.1.10	Privacy
4.1.11	Accuracy

Table 27: List of Standards

4.1.1 PCB Standard

For our PCB design we followed the IPC-2221 standard. This standard covers acceptable circuit board design, interconnections and how to correctly mount components. The most significant topics covered in this standard include how to properly space conductors and how large the traces on the board should be. When placing conductors on the PCB the distance between the two components would need to be spaced a certain distance. The way these components are spaced is based on two measurements, clearance and creepage, which can be seen on the figures below. Ideally the space between the conductors will be as much as possible without becoming redundant. These are defined in international standards IEC 950 and EN 60950.

4.1.2 Soldering Standard

J-STD-001 is a standard issued by IPC for soldered electrical and electronic assemblies. The standard specifies material specifications, process requirements, and acceptability criteria. Joint industry-standard(J-STD-001) is the industrial specification for electronics and electrical assemblies that are grouped according to the product classes. Electronic products are classified into three groups according to manufacturability, performance requirements, process control regulations, and verification testing.

Class 1: general electronic products

Class 2: service electronic products

Class 3: High-performance electronic products

The latest version of this document is J-STD-001 H. These standards outline materials, methods, and verification criteria for making high-quality soldered interconnections (lead and lead-free). This certification includes a thorough explanation of the following elements:

- Material, component, and equipment
- Soldering and assembly requirements
- Terminal and wire connection
- Through-hole mounting
- Surface mounting of components

- Cleaning and residue requirements
- Coating, encapsulation, and adhesives

4.1.3 12V Car Accessory Outlet Standard

Standard for 12 Volt Cigarette Lighters, Power Outlets, and Accessory Plugs J563_200902. This standard is intended to cover cigar or cigarette lighters as well as power outlets based on the form and dimensions of the cigar lighter, and accessory plugs for use in these devices. Components covered herein are designed to work in nominal 12 VDC systems. This standard is a full performance specification. It includes dimensional and operational parameters as well as performance characteristics which must be met when submitting a cigar lighter assembly, power outlet assembly, or plug for production approval. This standard constitutes an acceptance specification for these devices.

This standard covers the operational, reliability, durability, acceptance, and testing requirements for a cigar lighter (also referred to as just “lighter”) for installation in the passenger compartment of production vehicles. This standard covers power outlets that are based on the form and dimensions of the lighter receptacle intended for installation in the passenger compartment of production vehicles. This standard also covers plugs designed for insertion into the power outlet. Associated components supplied as part of, or with the lighter or outlet are also covered. Additional requirements may be added for these devices when mounted outside the passenger compartment of production vehicles.

Testing shall be done on part families (i.e., lighter receptacles and related knob-elements), as opposed to separate piece-parts, as directed by the appropriate purchasing agreement. Lighter knob-elements and lighter receptacles are not intended to be interchangeable when manufactured by different suppliers.

4.1.4 USB Standard

IEEE 1394, like USB 2, is a high-speed serial I/O (Input/Output) technology that can be found on many peripheral devices. Currently, the newest USB version, or specification, is USB 4.0, and it is contained within USB Type-C cables. It replaces USB 3.2 and 3.0 and enables data transmission speeds of either 40 Gbps or 20 Gbps. The Thunderbolt standard 3 and 4 use a USB-C connector.

4.1.5 Battery Standard

guidance for an objective evaluation of lithium-based energy storage technologies by a potential user for any stationary application is provided in this document. IEEE Std 1679-2010, IEEE Recommended Practice for the Characterization and Evaluation of Emerging Energy Storage Technologies in Stationary Applications is to be used in conjunction with this document. Secondary (rechargeable) electro-chemistries with lithium ions as the active species exchanged between the electrodes during charging and discharging are included in the category of lithium-based batteries for the purposes of this document.

Lithium-ion, lithium-ion polymer, lithium-metal polymer, and lithium-sulfur batteries are examples of secondary lithium-based batteries. Primary (non-rechargeable) lithium batteries are beyond the scope of this document. A technology description, information on aging and failure modes, a discussion on safety issues, evaluation techniques, and regulatory issues are provided in this document. Sizing, installation, maintenance, and testing techniques are not covered, except insofar as they may influence the evaluation of a lithium-based battery for its intended application. Lithium-ion batteries are known for their potential for combustion when mishandled, especially when dealing with charging and discharging. To fully protect our product and our consumers, we must thus strictly adhere to some very common standards. IEC 61960 and IEC 62133-2:2017 detail a lot of battery specifications as well, including the physical dimensions (which applies to manufacturers), along with electrical tests which range from charge and discharge performance to endurance in cycles and electrostatic discharge, tolerances, and markings/designations. Further, the latter is more specifically geared towards Safety and is more in-line with our use, detailing the charge and discharge procedure, tolerances, terminal contacts, wiring, current, temperature, and voltage management, venting, and more while providing specification of testing procedures. Both provide a lot of information to battery manufacturers, but they are still necessary in our case due to the possibility of misuse/mishandling of the batteries, these standards apply to secondary cells but may be helpful in guiding us in our primary system. IEC 62368-1 supersedes IEC 60065 and IEC 60950-1, and represents a shift in engineering principles, it is a standard for the safety of electrical and electronic equipment that classifies energy sources, prescribes safeguards, and gives guidance regarding the applications of and requirements for those safeguards. IEC 62133 further defines requirements and tests for secondary cells and batteries, specifically those which are sealed and portable. This standard mostly provides information on tests which should be run and a variety of requirements. UL 1642 is a general safety standard for lithium batteries, rechargeable or not, and includes testing for short-circuiting, heating, temperature cycling, forced-discharge, and altitude simulation along with fire-exposure, flaming particles, projectiles, and explosions. This standard is further substituted by application specific ones.

4.1.6 Optical Mounting Hardware Standard

The Raspberry Pi camera module is equipped with a C-mount threaded connection. To install and align each optical element we C-mounted threaded lens tubes. Using Edmund Optics as a reference, the necessary lens tube comes in a range of lengths from 5.6 millimeters up to 24.2 millimeters and can be threaded to make longer lengths of tube. The diameter of compatible lenses also ranges from 3 millimeters to 25.4 millimeters which will be a limitation on the maximum lens diameter. Spacer rings and retainer rings will secure the elements in the desired locations.

4.1.7 Camera Serial Interface (CSI) Standard

Camera Serial Interface (CSI) is the standard for communication between sensor modules and its managing computer. CSI has three sub-standards being CSI-1, CSI-2 and CSI-3.

CSI-1 was introduced as the first standard communication language with a minimal feature set. CSI-2 expanded on the features available in CSI-1 by including support for RAW-16 and RAW-20 color depth and as of September 2019 introduced support for RAW-24 color depth. Other features that are currently available in CSI-2 version 3 are Smart Region of Interest, End-of-Transmission Short Packet and Unified Serial Link. Unified Serial Link reduced the number of transmission lines needed for CSI communication which increased the maximum speed of data transmission. CSI-3 is the most recent version of CSI and is designed to facilitate communication between multiple cameras and computers, while also increasing the speed of video and image transmission. The computer controlling the image sensor uses the Camera Command Set (CSS) to send instructions to the sensor. All the above standards were created by the MIPI Alliance to facilitate the communication of sensors and computers in mobile computing. CSI-2 will be utilized when sending and receiving signals from the Raspberry Pi 4 Model B to the Raspberry Pi High Quality Camera Sensor (Sony IMX477R).

4.1.8 High-Definition Multimedia Interface (HDMI) Standard

High-Definition Multimedia Interface (HDMI) was first introduced in 2002 with the goal of decreasing the connector dimensions and implementing a signal path for audio to be transmitted. The HDMI standard has a bit rate of up to 48 gigabits per second which will provide more than enough bandwidth for us to output video signal from the Raspberry Pi to the display. HDMI 1.0 is capable of outputting 1080p video at 60 frames per second. HDMI has five standard connector variants, the three most important for our project being Type A, Type C (HDMI mini), and Type D (HDMI Micro). All three types contain 19 pins for data transmission with the major difference being the form factor. Type A and Type C have the same pin assignment while Type D has a separate pin arrangement. The HDMI mini connector is used on smaller displays which potentially will be used to connect to the Raspberry Pi’s HDMI Type A connector.

4.1.9 SD Card Standard

The Secure Digital (SD) cards are a type of external storage that can be used to store music, videos, photos, documents, etc. It was introduced in 1999 through the collaboration between SanDisk, Panasonic, and Toshiba. The goal was to provide a memory storage device that could improve customer experience. Considering the size, portability, and capacity for data storage of an SD card, this made the SD card a preferred and popular choice for many devices such as cameras and smartphones. SD cards and microSD cards share the same standards of: SD, SDHC, SDXC, and SDUC and microSD, microSDHC, microSDXC, and microSDUC. The SDHC and SDXC are the more popular stands for both the SD card and microSD card. With the help from an adapter, the microSD card can be used in devices that support only SD cards. The storage capacity can range from 2GB (SD) to 32GB (SDHC) to 2TB (SDXC). SD cards use flash memory to provide nonvolatile storage, this allows the retention of data even without a connected power source. The flash memory technology in SD cards allows fast data transfer rates, low power consumption, and enhanced security.

4.1.10 Privacy Standard

An important component of TrackPack is that we must comply with data privacy regulations. Since TrackPack will be dealing with sensitive information about a driver's vehicle, this data must be protected from unauthorized access. We must ensure that the data collection and storage methods used are secure and that only authorized personnel have access to the data. To comply with data privacy regulations, the project team must ensure that the data collection and storage methods used are secure. Ensuring we follow the appropriate security precautions; we have made sure to utilize an OBD II scanner provided with a security pin to gain access to its data.

4.1.11 Accuracy Standard

Accuracy and reliability are critical standards that TrackPack must comply with, as the data collected from the driver's vehicle must be precise and reliable to provide valuable insights. The data collected and analyzed from the vehicle will be used to provide insights into the vehicle's condition, driving behavior, and overall performance. Therefore, it is essential to ensure that the methods used to collect and analyze data are reliable and provide accurate results consistently.

To ensure accuracy and reliability, the project team must carefully select the sensors and data collection methods used. The sensors must be high-quality and calibrated correctly to ensure that they provide accurate data. The data collection methods must also be reliable and provide accurate results consistently. This includes ensuring that the transmission of data has minimal latency. Overall, complying with the accuracy and reliability standards is crucial for the success of the TrackPack project. By ensuring that the data collected is precise and reliable, the system can provide valuable insights to the driver.

4.2 Project Constraints

During the development of TrackPack, there will be several limiting factors that will impact the functionality and design of the system. To ensure the success of the TrackPack project, it is crucial to consider these limiting factors. By understanding these constraints, the project team can determine how the system will operate and design it around the constraints mentioned. This will ensure that the system remains functional, effective, and efficient while meeting the project goals. Shown below is a table containing the constraints we may encounter throughout this project.

Section	Constraint
4.2.1	Safety

4.2.2	Economic
4.2.3	Ethical
4.2.4	Time
4.2.5	Processing
4.2.6	Equipment
4.2.7	Environmental
4.2.8	Optical Design
4.2.9	Presentation
4.2.10	Hardware Availability

Table 28: List of Constraints

4.2.1 Safety Constraints

As our user will be behind the wheel of a vehicle, the safety of the driver is the most important thing. Two key areas we want to avoid are the driver's access to the pedals, and the driver's view. To keep TrackPack and least invasive and as portable as possible, we utilized a Bluetooth OBD-II dongle to eliminate the safety hazard that can be caused by operating a motor vehicle at high rates of speed with wires running across the driver. By using 12V accessory outlet, commonly found near the center console, we want the connection to the device to be as thin as possible to avoid getting in the way of the driver's ability to reach any input they may have in their vehicle on that center dash, such as the radio or air conditioning system. The second main safety feature we want to consider is the driver's visibility. TrackPack intends to be mounted to the dashboard of the vehicle. For this reason, we want to make the system (camera and display included) as small as possible so as not to take away from the driver's view of the road. Taking this into consideration, the display screen should be small enough to not block the driver's field of view, but still large enough so that the driver can look and quickly read their statistics from the driver's seat and immediately return their view to the road. The system must also be robust as our project goal is for our final design to see actual track use. Therefore, the system must be able to handle the quick acceleration and turns a user might take. The housing must also remain still and securely mounted to the windshield as dislodging in motion could break the system or distract the driver and cause an accident. As important as it is that the device is compact, it also needs to be neatly put together and connections between the PCB, Raspberry Pi (camera module), and the power supply need to be encased entirely in the housing and show none of the interior components, for protection of the device, as well as the appeal to the user.

4.2.2 Economic Constraints

As this is not a sponsored or endorsed project, we would like to keep our total cost as low as possible. On major components we emphasized the budget towards those hardware components that ensure quality, reliability, and success of implementation with minimal problems debugging, if there is not a substantial price difference. If any of these components also require a much earlier arrival date for the team's success, (i.e., the PCB) shipping costs will be allowed to ensure an earlier arrival date. On smaller components we looked to use any assets available to us already or the option to buy in bulk if multiples are required. Many of the parts that we've selected for TrackPack are in high demand, since we are currently recovering from a chip shortage, many of these parts of out of stock through the normal vendors. We would like to manufacture TrackPack and keep the total cost under $600 to mitigate the amount that each member must contribute monetarily and keep TrackPack as an affordable replacement to other similar products on the market.

4.2.3 Ethical Constraints

A notable ethical constraint also factors in a safety constraint of TrackPack. TrackPack is intended to be a device for off-road use only. The device can be used to analyze simple parameters while driving on the public road, but TrackPack is designed with advanced features beyond that of everyday driving. TrackPack can detect many other parameters of a vehicle which require generating high-speed metrics on the public road. Considering the legality and safety concerns of this, these tests should only be conducted on a closed course. Recognizing the importance of our safety while testing, we did not test the functionality of the device on the public road. Accompanying the driver, we also had a passenger present to analyze the device so the driver can focus on the road while reaching the required statistics we would like to measure and record. Since TrackPack has a video feature, we have also considered only having videos saved and to be watched later to avoid the driver looking at video while driving and causing unsafe conditions.

4.2.4 Time Constraints

Throughout the design and build of TrackPack, there are time restrictions we should consider and be aware of. Since the development of TrackPack is marked by a deadline at the end of the Senior Design II semester, there are personal oriented and class-oriented deadlines we need to follow. As time being one of the constraints on completion, time can also impact the quality of our model. We aim to complete the design of TrackPack to follow closely among the devices existing on the market. At this point, this is where team dynamics and careful planning play a vital role when developing TrackPack. The time restrictions imposed on this project determine the outcome and design of the project. It is critical to examine time restrictions realistically. If one approach may provide a higher quality design but it exceeds the time limits, another approach should be chosen to fulfill the deadlines instead. Depending on the parts we selected to assist with the project design, ensuring and verifying stock availability and delivery time of the parts can impact our deadlines. In the case of unavailability of parts or extremely delayed delivery, we may need to devise an alternative. Delay in parts arrival can reduce the time we have to assemble TrackPack and debug any issues that may arise during the testing and build of the device. To avoid this hurdle, ordering the parts earlier than we need may be ideal. This is to mitigate the issue of parts becoming out of stock and/or the delay in parts' arrival.

4.2.5 Processing Constraints

Since we utilized a Raspberry Pi for image processing and to compile all the data into one location, we must understand the capabilities of this computer. The Raspberry Pi, while a high performing computer for its size, is still limited to a clock speed of 1.5 GHz and a total RAM memory size of 4 gigabytes. Due to this, we had to closely monitor how we are using the computers' resources so as not to overload the Raspberry Pi without proper cooling. Alternatively, we could include a cooling system for the Raspberry Pi, but this will increase the total cost and does not provide us with unlimited processing power. Also, because Raspberry Pi computers run on a Linux based operating system we are limited to software that can operate on the operating system we chose. Linux software can be poorly optimized which will lead to excessive processing power being used compared to software that has greater support for optimizing image processing programs.

4.2.6 Equipment Constraints

Considering the amount of hardware that TrackPack features, there are a few equipment constraints. The design of the housing will need to protect the components from excessive temperatures that it may potentially experience while the device is not in use. For example, if the car is parked in the sun while TrackPack is mounted but not in use, the processors within the unit cannot operate above a certain temperature threshold and may even be permanently damaged if immediately powered on without cooling. This can also be considered a safety constraint as well due to the potential fire hazard electronics pose in hot environments. Furthermore, finding an ideal OBD-II Bluetooth connector that meets both newer and older standards while remaining cost effective is a restraint. To accomplish this, we much choose an OBD-II Bluetooth connector that supports protocols from 1996 to 2023. A major drawback with a Bluetooth adapter that can scan all major years, makes, and models, is that if we want to remain cost effective on the Bluetooth connector, we'll have to sacrifice some of the OBD-II functions that come with the higher priced connectors. Scanning additional computer modules such as SRS and ABS are a luxury that is accompanied by the more costly OBD-II connectors. Luckily, scanning these additional modules does not provide any additional useful information for TrackPack.

4.2.7 Environmental Constraints

Environmental constraints are subset constraints that can tie together with ethical constraints. Our primary environmental constraint is that to test TrackPack effectively, it requires a relatively large amount of driving to ensure all the hardware is working correctly and tracking the data appropriately. The downfall to the amount of driving required is the increased amount of vehicular pollution. It's common that performance vehicles already generate and higher amount of pollution in comparison to the typical economic vehicle due to the high output engines, and commonly the removal of the vehicle's catalytic converter. It's difficult to test TrackPack functionality while not physically in a vehicle, and TrackPack's primary operation will require us to adjust the recording of the data to account for any latency issues to ensure that the data displayed to the user is as accurate and correct as possible. With this constraint in mind, we mitigated it by keeping the physical vehicular testing time of TrackPack to a minimum and dialing in proper sensor functionalities as much as we can without utilizing a vehicle.

4.2.8 Optical Constraints

A design constraint on the optical system is the types of lenses available for us to use. When designing the lens system in Zemax, we must carefully select the parts since we cannot manufacture our own custom optics due to price limitations. This leaves us with using parts sold by vendors such as Thorlabs, Edmund Optics, and Newport. These sellers manufacture lenses, and most of their lenses with identical radii of curvature on each face. Similarly, they manufacture lenses in discrete standard diameters. This factor will play a less significant role in the overall design process, but it will require us to select lenses that have a diameter greater than or equal to the calculated size in Zemax. Other lenses used in the design such as the meniscus lenses are sold with a maximum diameter of one inch which will be a constraint on what can be included in the design.

4.2.9 Presentation Constraints

Presenting the functionality of TrackPack creates a constraint since TrackPack is intended for off-road or closed course use only. To promote safe driving on public roads, we don't anticipate testing the limits of TrackPack on public roads, however, this creates a constraint for the presentation aspect of TrackPack.

For our complete presentation, we would like to clearly demonstrate the full functionality of TrackPack, however, this complete demonstration in compliance with the law would require us to travel to our local raceway and with permission of the owners conduct some small tests and recordings.

4.2.10 Hardware Availability Constraints

Hardware availability is a substantial constraint since we are still recovering from a recent chip shortage. Important hardware to TrackPack such as the Raspberry Pi 4 is completely sold out and backordered for months. The lack of availability for certain hardware components ties together with our time constraints.

Many components have also increased significantly in price due to lost profits, so overspending might be necessary to receive the needed components in a timely manner to complete our design by the deadline.

5 System Design

TrackPack is a carefully planned and intricately interconnected network of various components, each with its own unique functions and roles. To ensure the seamless operation of TrackPack, it is crucial to analyze and understand each of these components thoroughly. Our analysis will encompass a wide range of factors, including the technology used in each component, its design, its interface, and its compatibility with other components. In this section, we will cover an in-depth analysis of the various components in the subsystem of TrackPack. For each component selected for the build of TrackPack, we will discuss their individual functionalities and how they are integrated with each other.

In the following sections, we will provide a comprehensive overview of the purpose of each component and the role each component plays in the overall system and functionality of TrackPack. This will give us a clear understanding of how all the different pieces fit together, allowing us to optimize the system's performance and ensure its success. The various parts elected for this build includes:

- Microcontroller
- PCB
- Display
- CMOS Sensor
- Design Lens
- Accelerometer
- Gyroscope
- GPS module
- OBD II scanner

The following sections will illustrate the function and integration of each part.

5.1 Hardware Design

In this section, the TrackPack hardware design configuration will be demonstrated. To demonstrate the overall design overview of the TrackPack hardware configuration, schematic designs will be included along with the integration process.

5.1.1 Schematics

Using Autodesk Eagle all of the schematics below were created to show the layouts of our devices. All RLC components were put in using the US standard footprint and schematics available in the Eagle Library. Footprints for the components of our design were downloaded using UltraLibrarian’s catalog to find the devices we need and export them the Eagle6. To incorporate these new devices into Eagle we executed the script given from UltraLibrarian, and created a new library that would contain all of the components on our design.

5.1.2 Sensors

Accelerometer

Further into our research, we were searching for the ideal accelerometer to utilize for TrackPack. We discovered the BerryGPS-IMU and 10DOF. It quickly became clear that this IMU was the perfect fit for our intended design, due to its impressive capabilities and features. With the accelerometer on board, we'll be able to accurately measure the acceleration forces impacting the vehicle, as well as the gravitational forces that come into play during acceleration, braking, and cornering. We incorporated this IMU into our PCB, enabling us to capture these critical parameters and deliver a more robust and reliable system overall.

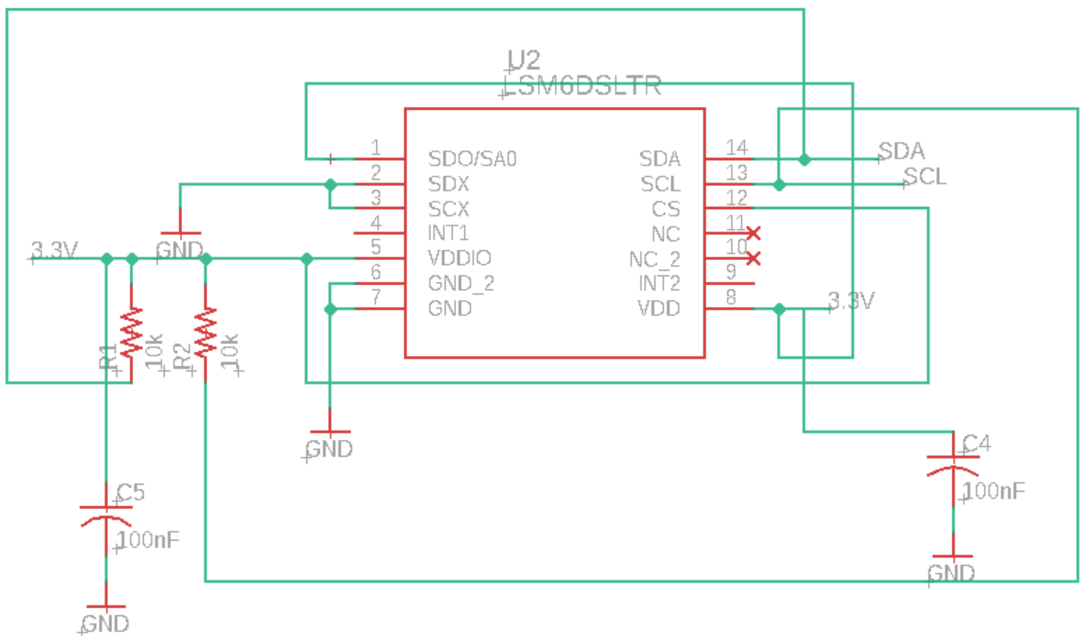


Figure 33: Accelerometer LSM6DS3

All modes will be following the same timing pattern. The primary difference lies within how fast the data transfers. Data starts the transmission process with the starting condition that is signaled by the SDA pulled low as SCL remains high. When the SCL is low, the SDA sets the first bit of data and keeps the SCL low. When the SCL rises, the data is obtained. Validation of the bit is done when the SDA does not change within the rising and falling edge of the SCL. This process continues until there is a clock pulse that pulls the SDA low for the stop condition. With the microcontroller as the master for our design the slaves (accelerometer and GPS) will receive the clock signal from the MSP430 on the SCL line, and the data will be transferred between the two on the SDA line.

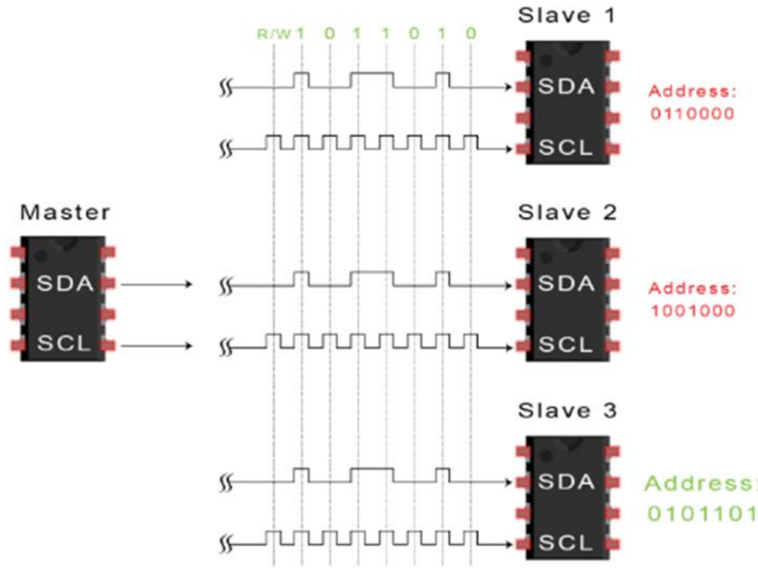


Figure 34: I2C read/write communication

Gyroscope

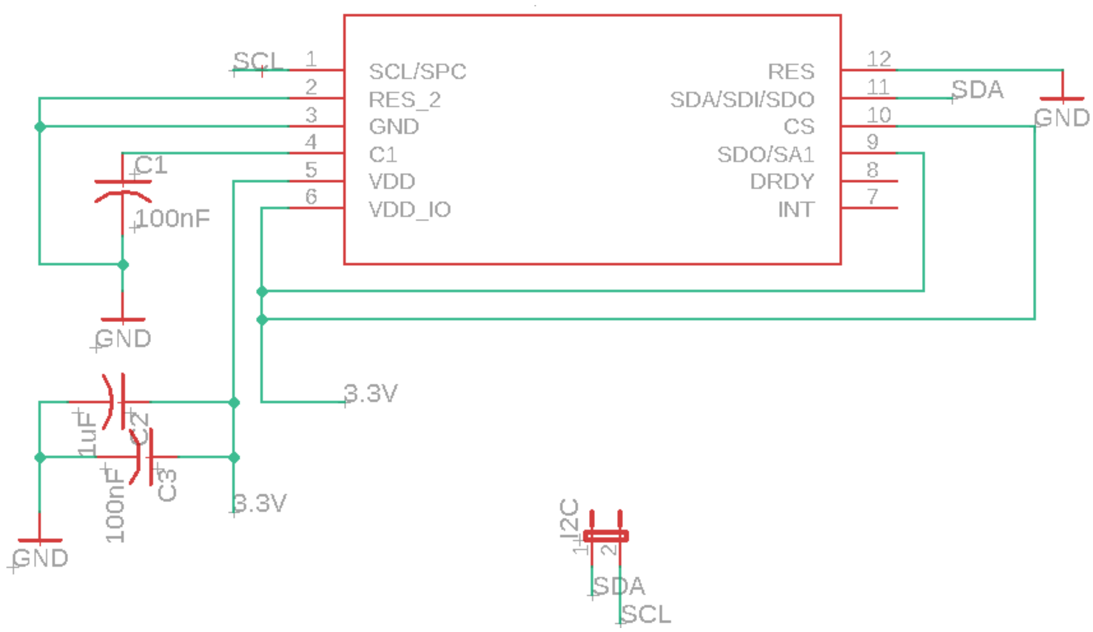


Figure 35: Schematic of LIS3MDL

The schematic on each module of the IMUv4 are similar in that they all will connect to the SDA_SDI pins and the RLC values are provided by the manufacturer for voltage regulation for operation, or bypass components to reduce noise in transmission.

GNSS

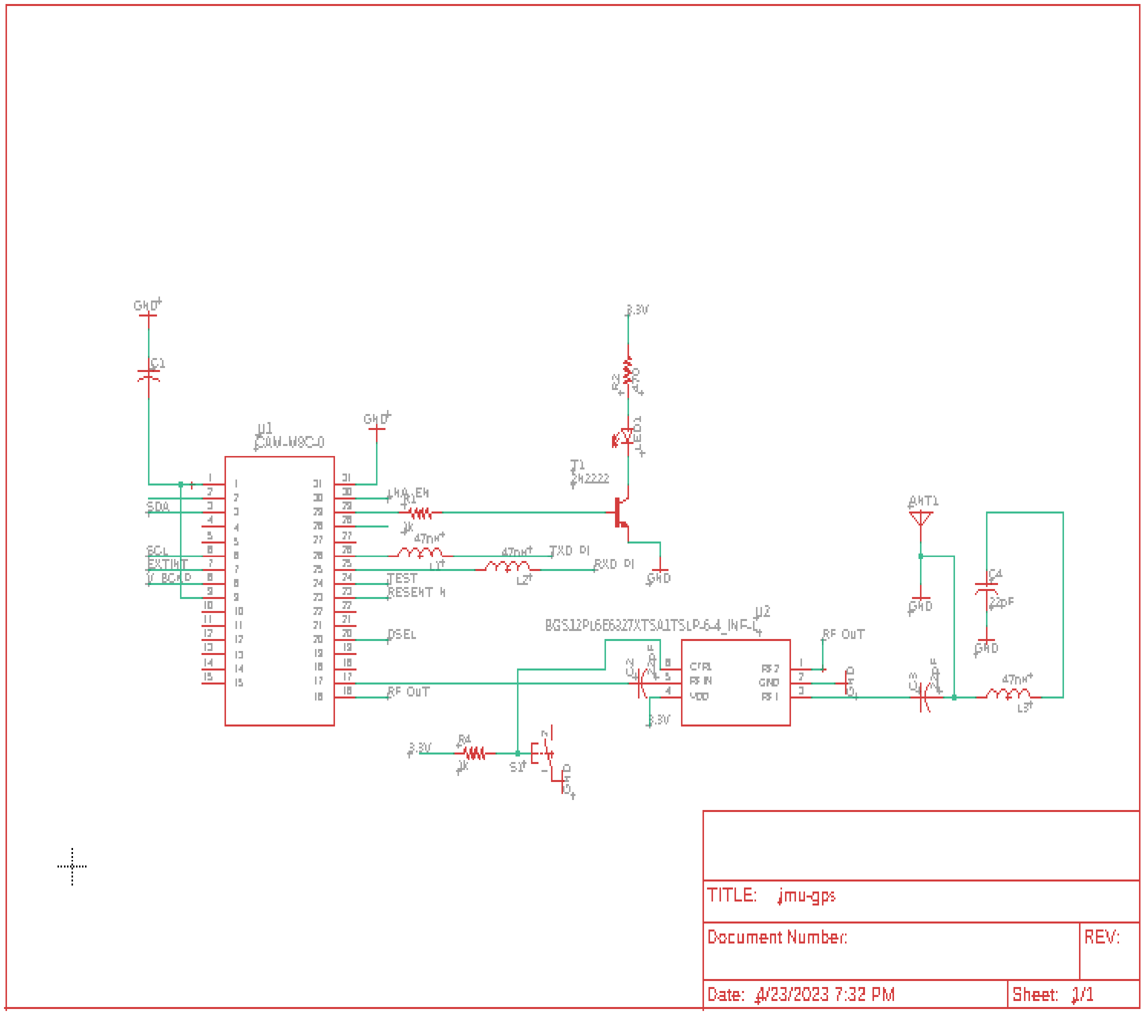


Figure 36: Schematic of CAM-M8C-0 and BGS12PL6

Breaking down the schematic of how the GPS module on the IMUv4 works, the CAM-M8C is the receiver of the module connected to the local 3.3V source. The connected passive components are for voltage regulation into the device. The BGS12 acts as the switch to the CAM-M8C to power it on and off, with an attachment for an external antenna.

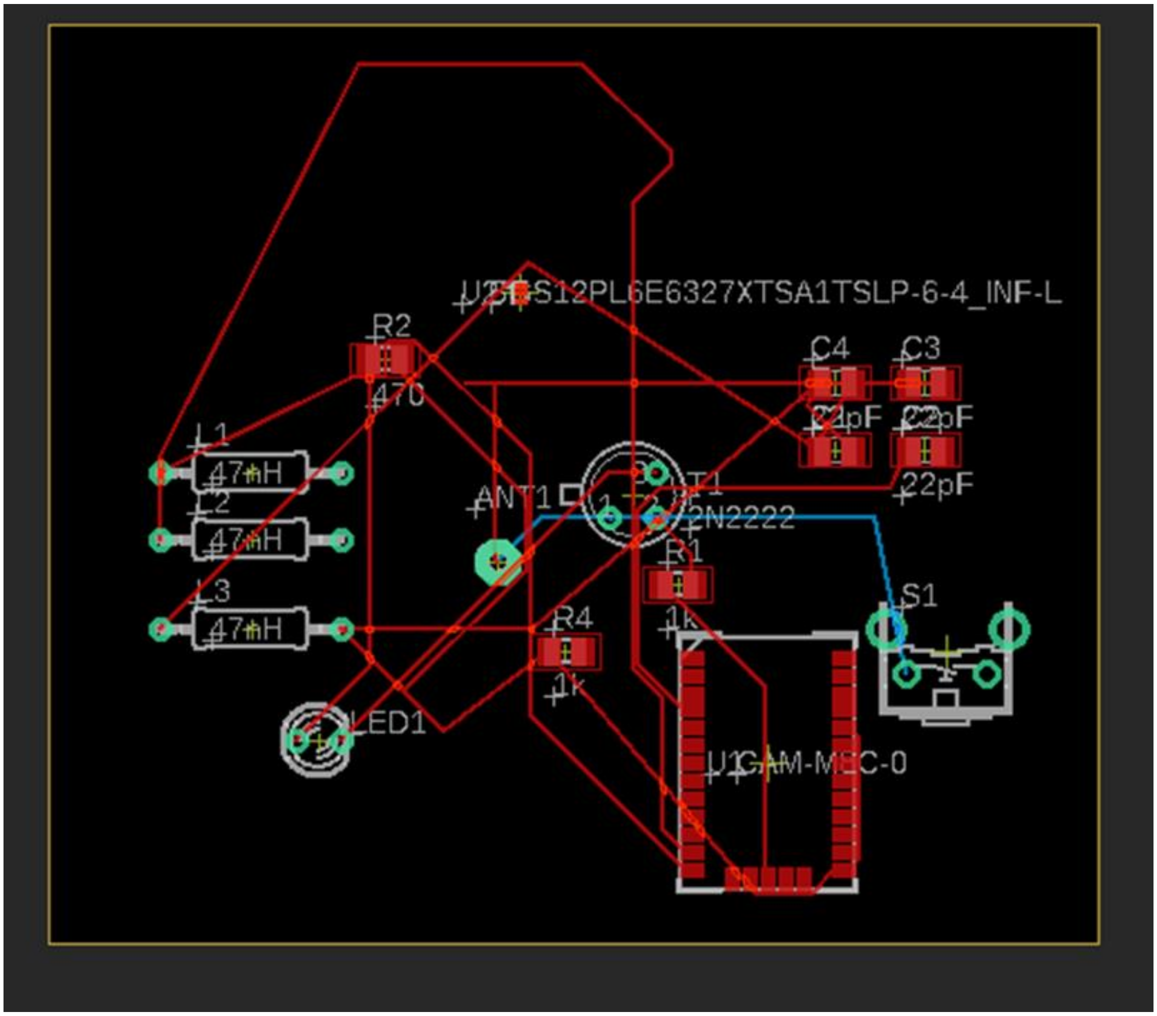


Figure 37: PCB view of GPS module

CMOS

To realize the designed optic lens and image processing we are using the camera module on the Raspberry Pi 4 HQ camera board. Below is the detailed schematic of their camera module provided by Raspberry Pi. This schematic shows the connections for each type of communication protocol available to the Raspberry Pi 4 but we still intend to exclusively use I^2C. As this is an external device it will not show on our unique PCB but knowing the connections of the device will help us if we have any issues debugging to see how the devices read and write connections are wired.

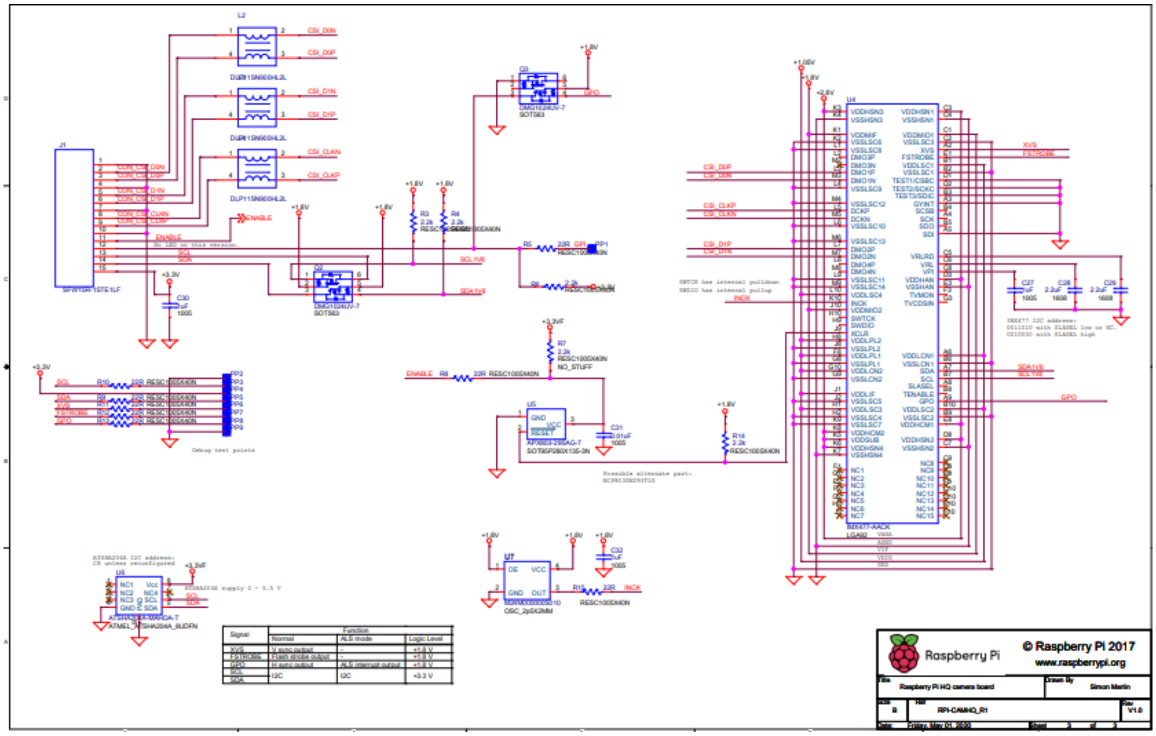


Figure 38: Raspberry Pi 4 Camera Schematic

5.1.3 5V and 3.3V Rails

These are the realized schematics from the voltage regulators provided by Webench. The RLC values were chosen, given the information in the datasheet to achieve the desired output and then given with their respective efficiency curves per input voltage. With a 12V input these two regulators are to have their highest efficiency at load currents near the load of our systems.

For the 5V regulator the output voltage is set with a resistor divider from the output node to the FB pin. TI recommends using 1% tolerance or better divider resistors. Start by using V_{out}=0.804 x (1 + R_{FBT}/R_{FBB}) to calculate V_{out}. To improve efficiency at very light loads consider using larger value resistors, too high of resistance will be more susceptible to noise and voltage errors from the FB input current will be more noticeable.

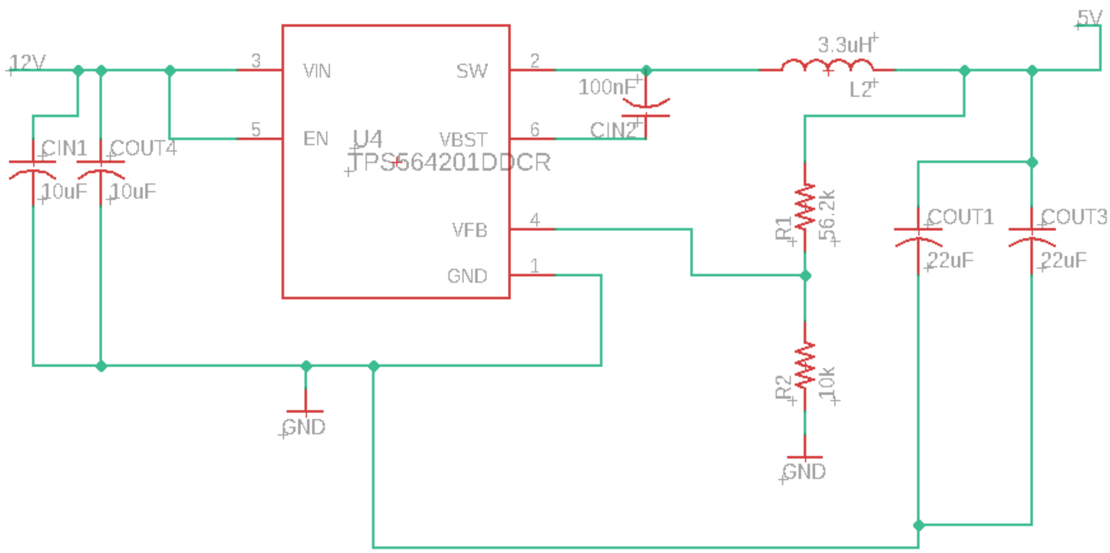


Figure 39: Schematic of 5VDC converter

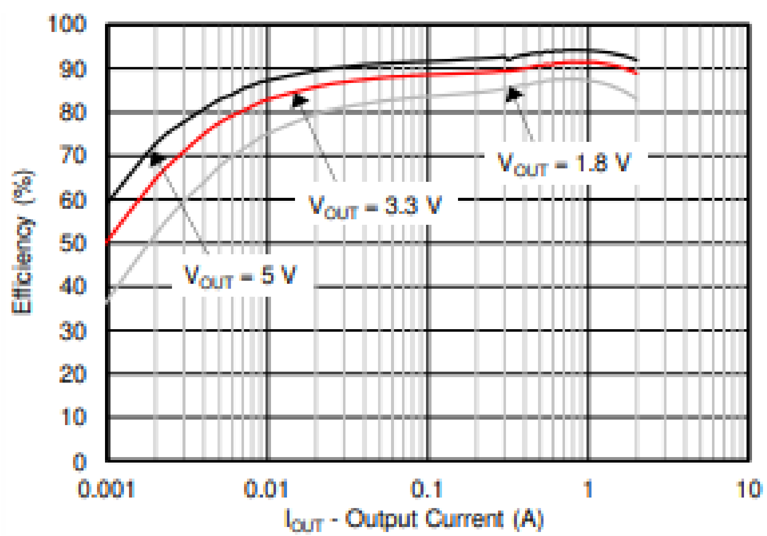


Figure 40: Effciency of TPS564201 vs Load Current

To understand the location of the RLC components of the layouts Webench produces of the DC voltage converters we researched in the datasheets to see for any information for the reasoning behind the location of the part. In the datasheet provided by TI, along with the layout example they give the layout guidelines which prove significantly more useful in the development of the board as most of them pertain to the placement of the devices on the PCB to maximize efficiency, heat dissipation, and to remove impedance.

- VIN and GND traces should be as wide as possible to reduce trace impedance. The wide areas are also of advantage from the viewpoint of heat dissipation.
- The input capacitor and output capacitor should be placed as close to the device as possible to minimize trace impedance.

- Provide sufficient vias for the input capacitor and output capacitor.
- Keep the SW trace as physically short and wide as practical to minimize radiated emissions.
- Do not allow switching current to flow under the device.
- A separate VOUT path should be connected to the upper feedback resistor.

A precision 0.8-V reference voltage (VREF) is used to maintain a tightly regulated output voltage over the entire operating temperature range. The output voltage is set by a resistor divider from VOUT to the FB pin. $R_{FBT}=\frac{V_{OUT}-V_{REF}}{V_{REF}}xR_{FBB}$ It is recommended to use 1% tolerance resistors with a low temperature coefficient for the FB divider. Select the bottom-side resistor, RFBB, for the desired divider current and use to calculate the top-side resistor, RFBT. The recommended range for RFBT is 10 kΩ to 100 kΩ.

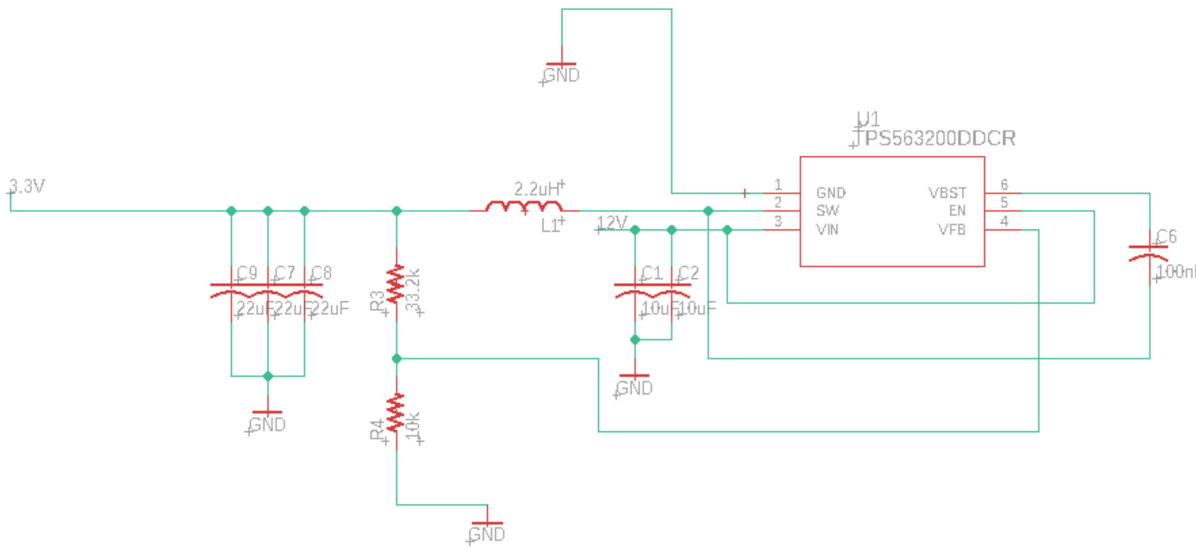


Figure 41: Schematic of 3VDC converter

When laying out the schematic for the 3.3V converter we adhered to the location requirements given by Texas Instruments

- The input bypass capacitor CIN must be placed as close as possible to the VIN and GND pins. Grounding for both the input and output capacitors should consist of localized top side planes that connect to the GND pin.
- Minimize trace length to the FB pin net. Both feedback resistors, RFBT and RFBB, must be located close to the FB pin. If VOUT accuracy at the load is important, make sure VOUT sense is made at the load. Route VOUT sense path away from noisy nodes and preferably through a layer on the other side of a shielded layer.
- Use ground plane in one of the middle layers as noise shielding and heat dissipation path if possible.

- Make VIN, VOUT, and ground bus connections as wide as possible. This reduces any voltage drops on the input or output paths of the converter and maximizes efficiency.

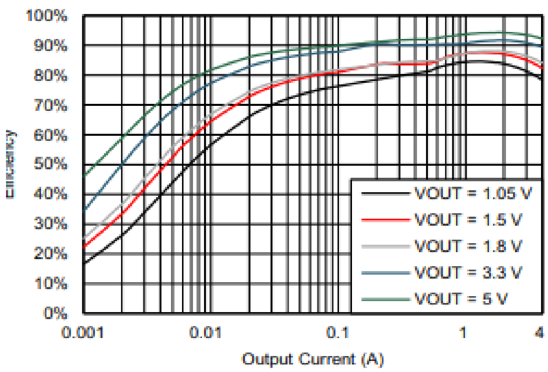


Figure 42: Efficiency of 3.3V converter vs Load Current

Once the load current reaches closer to the operating levels of the components, we approach a 90% efficiency rating. When considering the layout of the two converters into the PCB size is not an issue as the device footprints are both small and don't require many RLC elements as well. In the completed design these two systems will be kept central to allow easy jumping to the input voltages of the other components.

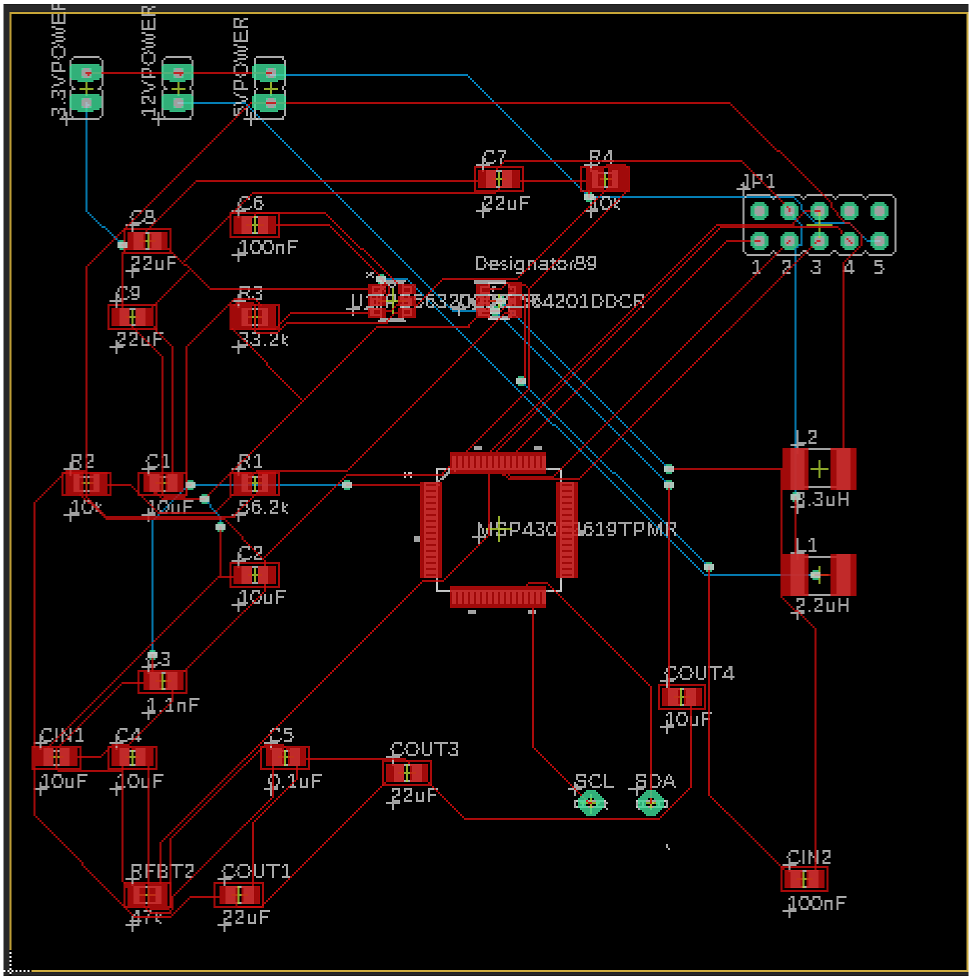


Figure 43: PCB view of voltage converter board

5.1.4 Microcontroller

This schematic below of the MSP430 includes only the connections for the power as the several other connections to the auxiliary components will be better shown in the integrated schematic. The RLC values were derived from the datasheet for typical use at our rated power. Pins 3.1,3.2, and 3.3 are the pins we used to incorporate the I^{2}C data to the other components on our PCB.

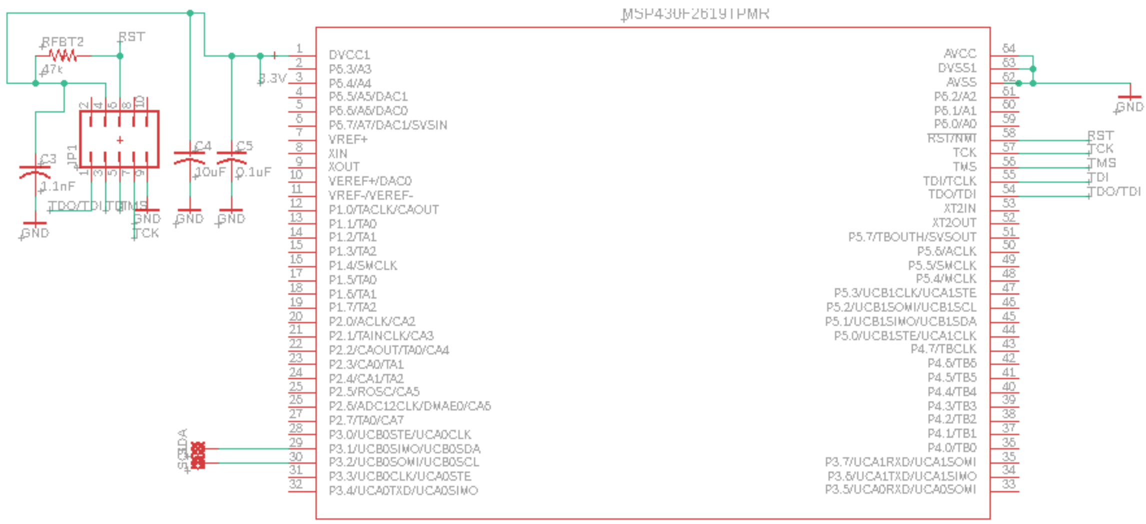


Figure 44: Schematic of the MSP430f168ipm

With I2C interfacing, only two pins are required from each device. The serial data (SDA) pin will be the line that will enable data transmission from the GPS, accelerometer, and the gyroscope to the MSP430. With data flowing in one direction, the I2C interface can utilize ultra-fast mode. This mode can provide a maximum transmission speed of 5 Mbit/s. In this mode, data can only be written. Thus, data related to the speed of the card will be written to the project controller's connected local storage unit. However, if errors arise in the movement of data due to the fast transmission of data, other modes such as high-speed, fast, and standard mode can be used. Since we would like to have multiple devices running off the I^{2}C language protocol as it would reduce the number of pins required and thus less wires and traces required, we can drop our speed down at still a fast enough rate to avoid loss of data and transfer of data bilaterally. The MSP-FET (MSP Flash Emulation Tool) by Texas Instruments is a powerful and essential device tailored for programming and debugging the MSP430 and MSP432 microcontrollers. As a leading semiconductor company, Texas Instruments designed the MSP-FET to provide engineers and developers with a reliable and efficient solution for firmware development and testing. With its versatile features, the MSP-FET enables seamless connection to target devices via JTAG or Spy-Bi-Wire interfaces. This allows for real-time code execution, as well as programming and erasing of flash memory, which is vital during the development and deployment stages of embedded systems.

The transaction on the bus is started through a START (ST) signal. A START condition is defined as a HIGH to LOW transition on the data line while the SCL line is held

HIGH. After this has been transmitted by the master, the bus is considered busy. The next byte of data transmitted after the start condition contains the address of the slave in the first 7 bits and the eighth bit tells whether the master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a start condition with its address. If they match, the device considers itself addressed by the master. The Slave ADdress (SAD) associated to the LSM6DSL is 110101xb. The SDO/SA0 pin can be used to modify the less significant bit of the device address. If the SDO/SA0 pin is connected to the supply voltage, LSb is '1' (address 1101011b); else if the SDO/SA0 pin is connected to ground, the LSb value is '0' (address 1101010b). This solution permits connecting and addressing two different inertial modules to the same I2C bus. Data transfer with acknowledge is mandatory. The transmitter must release the SDA line during the acknowledge pulse. The receiver must then pull the data line LOW so that it remains stable low during the HIGH period of the acknowledged clock pulse. The receiver which has been addressed is obliged to generate an acknowledgement after each byte of data received. The I2C embedded inside the LSM6DSL behaves like a slave device and the following protocol must be adhered to

Mode	Data Speed	Capacitance Rating	Drive	Direction
Standard	100kbit/s	400pF	Open Drain	Full-Duplex
Fast	400kbit/s-1Mbit/s	400-500pF	Open Drain	Full-Duplex
High-Speed	1.7 Mbit/s	400-100pF	Open Drain	Full-Duplex
Ultra-Fast	5Mbit/s	N/A	Push-Pull	Simplex

Table 29: Speed modes of I^2C communication

5.2 Optical Design

In this section, the optical design steps performed to create the optical subsystem will be detailed. To describe the design process and setup, multiple simulations and calculation steps will be demonstrated.

To begin, we conducted investigations into existing generic design methods for optical systems. Specifically, we explored three main design methods: Cooke Triplets, reverse telephoto lenses, and short-focus wide angle lenses. Each lens system has its own pros and cons that must be carefully considered. The Cooke Triplet design can produce a sharp image, correct for field curvature, and has a simple design. However, this comes at the expense of a smaller field of view. The reverse telephoto wide angle lens can achieve a wider field of view, but the design process is more complex and requires more optics or expensive custom optics. Finally, the short-focus symmetric wide-angle lens can achieve a similar field of view as the reverse telephoto lens, but with a symmetric design that can reduce the cost of the system. However, short-focus wide-angle lenses cannot correct for image distortions such as field curvature and vignetting as well as the Cooke Triplet can.

In summary, each lens design style has its own set of benefits and drawbacks that must be carefully weighed when designing a lens system for our project. Ultimately, we have decided to base our optical setup on the Cooke Triplet design, as it can achieve sharp images with a large aperture and lens shapes that are within our budget. Although this design will limit the maximum field of view we can capture, it will be able to correct for field curvature aberrations, which will ideally result in a recording that closely resembles the driver's view.

5.2.1 Optical Design Overview

To begin the lens design, a Matlab script was written to calculate a thin lens approximation of a Cooke Triplet. This was a crucial step in the design process because the e Triplet is a generic lens design format that was created to reduce field curvature that occurs when imaging large field angles. It consists of 3 lenses, typically a bi-convex lens, followed by a bi-concave lens, and finally the b-convex lens again except this time the lens is mirrored across the vertical axis. The center lens often is made of a different material with a higher index than the other two lenses to correct for both field curvature and chromatic aberration.

The general prescription derived from Matlab was then transferred into Zemax to perform precise ray tracing calculations. Using Zemax also allowed for several different variations of the design to be tested simultaneously, changing lens thickness, and spacing, as well as including more lens than just the Triplet to increase the desired metrics. This was an important step because the ability to test different designs simultaneously allowed for a faster design process, which was especially important given the time constraints of the project.

The most beneficial modification to the design was including primary lenses in the system before the Triplet to gradually focus the off-axis rays thus reducing the RMS spot size of the rays. This modification allowed for even sharper images to be captured, which was important for achieving the desired image quality for the project. Another benefit from using Zemax was the ability to quickly substitute lenses available from the three major lens manufacturers to create a final design that can be made from readily available lenses. This ensured that the design could be produced with a minimum of custom components, making it more affordable and easier to manufacture.

From the Zemax simulations we can measure the exact back focal distance of the system, and with that we can design the lens mounts and connector to the CS-Mount. This was an important step because precise spacing between each component was crucial to the overall performance of the system. To design the mounts, we used the CAD software Autodesk Fusion360. Since the design will require precise spacing between each component, each lens will have its own 3D-Printed holder which will be placed on threaded rods running parallel to the optical axis. This approach ensured that each lens was held securely in place, and that precise alignment was maintained even under challenging conditions.

Each lens mount will have a set screw hole to secure the lens within the mount without the need for permanent or semi-permanent glues. Then locking nuts will be used to secure each lens at the proper distance. This approach allowed for quick and easy lens replacement or re-alignment if necessary. Also, this system provides a modular approach to lens alignment and replacement which will be needed when attempting to reach stretch goals and advanced goals. Compared to typical lens tube mounting hardware, we will be able to adjust the position of every lens at once, rather than having to remove each prior lens to adjust any covered lenses.

Finally, this 3D design will have a sheath around its entire length to block any stray light from impinging on the sensor and obfuscating the video signal. This was an important step because stray light can significantly degrade image quality and could lead to poor performance in low-light conditions. By blocking stray light, we ensured that the image quality remained high, even under challenging conditions.

5.2.2 Explicit Design Process

The figure below shows the Matlab code written to calculate the surface curvatures of the thin lens approximation of the Cooke Triplet. The thin lens approximation assumes that the lenses are infinitely thin and that light rays pass through the lens without any bending. However, lenses have a definite thickness and the light rays bend when passing through them. Therefore, the thin lens approximation provides only an estimate of the optical performance of the lens system.

To achieve an 80-degree field of view (FOV) for our project, the back focal distance was determined to be 6.5 millimeters. The back focal distance is the distance from the last lens element of the lens system to the image plane or sensor. It is an important parameter that determines the working distance of the lens system and the size of the image circle that the lens system can cover.

To simplify the design process, the Matlab script calculated the curvatures needed for the first three surfaces of the Cooke Triplet. These surfaces are typically a bi-convex lens, followed by a bi-concave lens, and finally the bi-convex lens again except this time the lens is mirrored across the vertical axis. The first three surfaces were mirrored across the central plane of the middle lens to form the triplet.

```
f = 6.5; %mm
h = 4.5; %mm
FOV = 4 * asind(h/(4*f));
phi_tot = 1/f; %mm^-1

%glass 1  N-bk7
n_d1 = 1.5168;
V1 = 64.17;

%glass 2 N-SF2
n_d2 = 1.647690;
V2 = 33.820209;

phi1 = 0.5 * phi_tot;
phi2 = phi1;

syms phi11 phi12;
eqn1 = phi11 + phi12 == phi1;
eqn2 = (phi11/V1) + (phi12/V2) == 0;
[A, B] = equationsToMatrix([eqn1, eqn2],[phi11, phi12]);

%solve matrix
X = linsolve(A, B);

%convert to double
phi11 = double(X(1));
phi12 = double(X(2));

%surface curvature
C2 = phi12/(n_d2 - 1);
C1 = (phi11/(n_d1 - 1)) + C2;

R1 = 1/C1;
R2 = 1/C2;
C3 = C2;
R3 = 1/C3;
C4 = 0;
R4 = inf;
```

Figure 45: Optical Design Matlab Code

After running this code, the radii of curvature needed for all six surfaces were calculated to be:

Surface	Curvature, mm
1	5.4835
2	-7.556
3	-7.556
4	-7.566

5	7.566
6	5.4835

Table 30: Radii Curvature Calculations

The Matlab script was useful for providing a starting point for the design, but to perform more precise ray tracing calculations, the design was transferred into Zemax. Zemax allowed for several different variations of the design to be tested simultaneously, changing lens thickness, and spacing, as well as including more lenses than just the Triplet to increase the desired metrics.

These lenses were then brought into Zemax to determine the viability of the design. The object space f/# was defined as 2.5 to allow for a larger FOV with a relatively short back focal length, and the field angles for each set of rays were set to 0°, 20°, and 40°. The lenses were all given the same thickness of 2 millimeters and identical spacing of 0.5 millimeters. These values were chosen to allow all of the off-axis rays to pass through the stop. After generating the cross-sectional view of the ray trace the field angles had to be adjusted due to 0°, 15°, and 40°. This was because the higher angle rays were not fully passing through the aperture stop of the system. The figures below show the cross-sectional ray trace and the RMS spot size diagrams of on axis rays, 15° off axis, and 30° off axis.

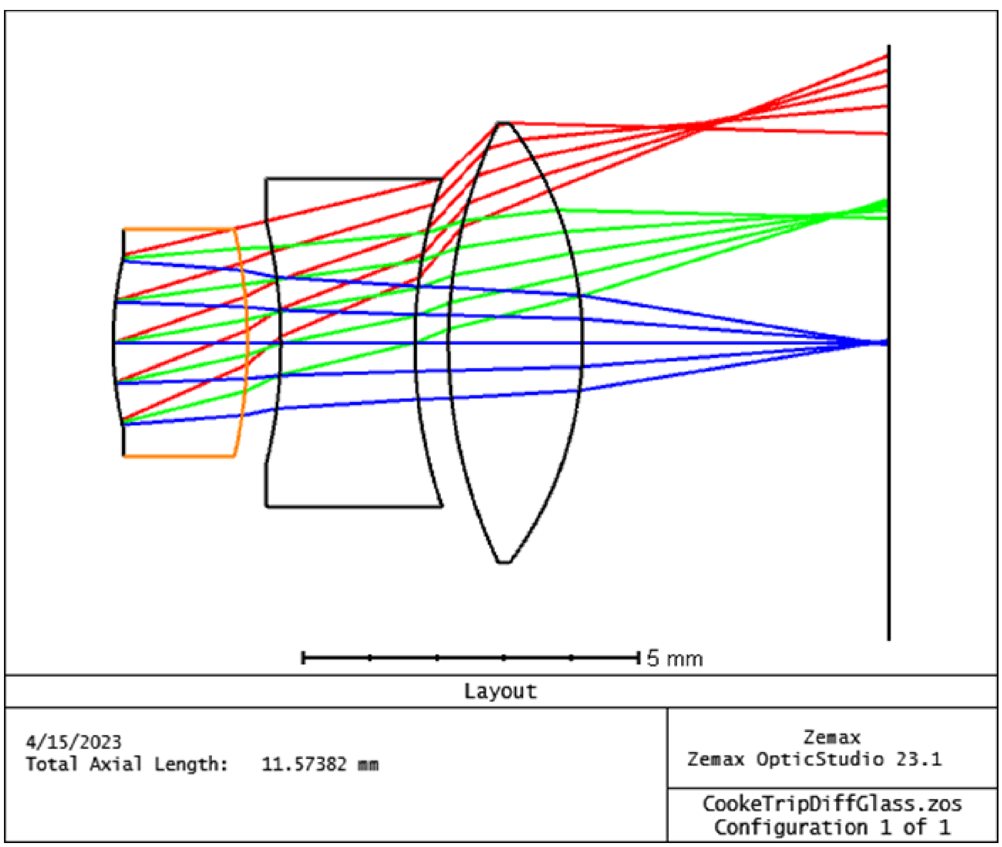


Figure 46: Zemax ray trace of Cooke Triplet

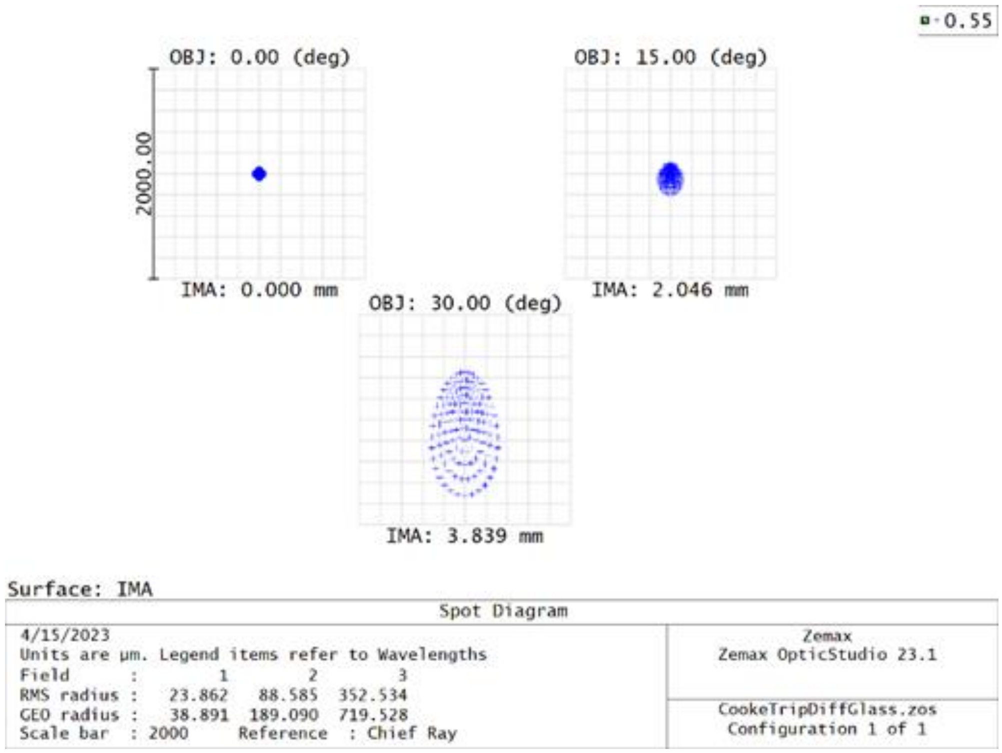


Figure 47: RMS spot size diagram of Cooke Triplet

5.2.3 Optical Design

The previous Cooke Triplet design alone is not sufficient to achieve the large field angle required for our goals. One of the issues is that the system uses optics that are between two millimeters and six millimeters, which are not readily available from lens distributors such as Thorlabs, Edmund Optics, and Newport. This can be a major problem as it can significantly increase the cost of the system, as sourcing the required custom lenses will be a challenge.

Moreover, the design has poor focusing of rays beyond a 30-degree field of view, which will not be acceptable for achieving our required wide angle. To overcome these limitations, we included two negative plano-concave lenses in our design. These lenses have a large diameter and negative focusing power, which causes off-axis rays to disperse greater than on-axis rays before encountering the triplet.

The increased dispersion of these rays has two effects on the system. Firstly, it delays the focusing of far off-axis rays through the triplet system so that the focal distance is closer to the on-axis rays. Secondly, it allows the use of larger diameter optics in the triplet. This is important as the larger diameter optics provide better off-axis performance and reduce the need for custom small optics.

The figure below shows the final design created after including the two plano-concave lenses. This design provides better performance than the Cooke triplet design above, and it uses readily available lenses from the three major lens manufacturers. The figure below shows the final design created after including the two plano-concave lenses.

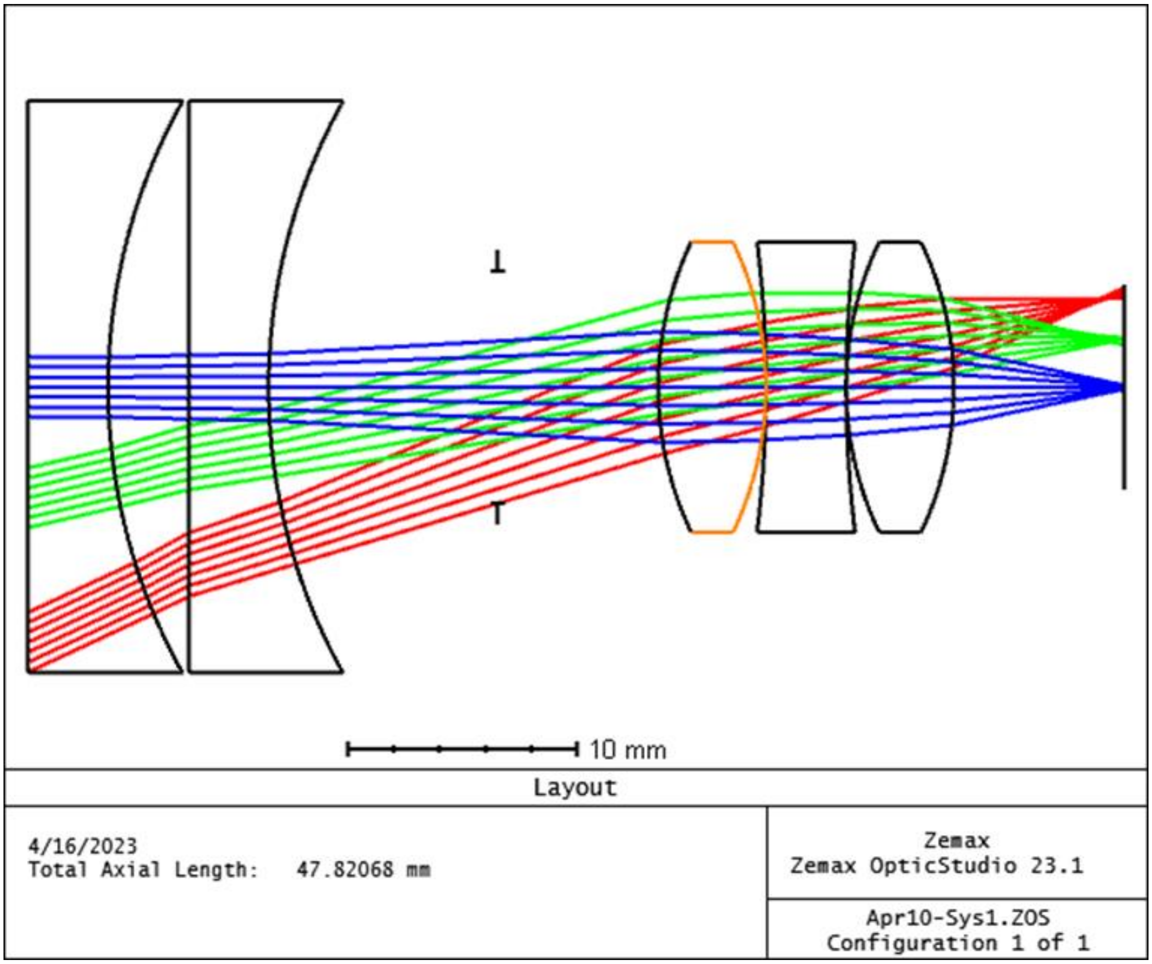


Figure 48: Zemax ray trace of final system

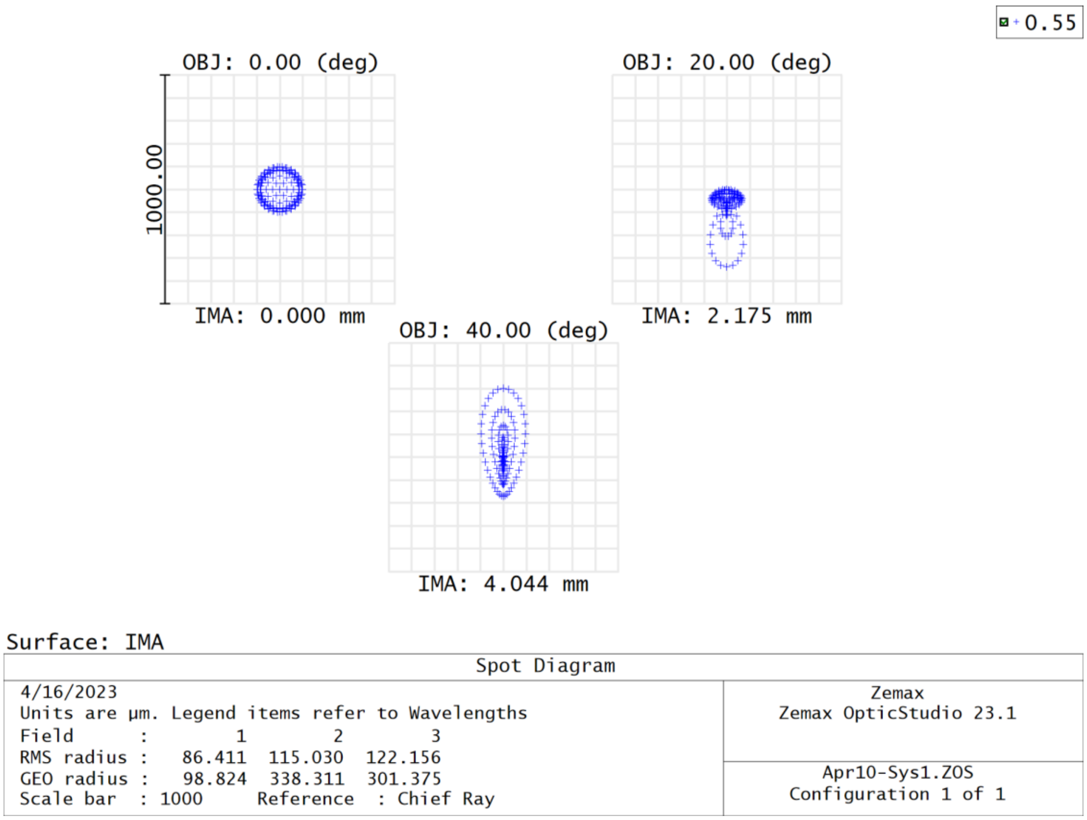


Figure 49: RMS spot size of final system

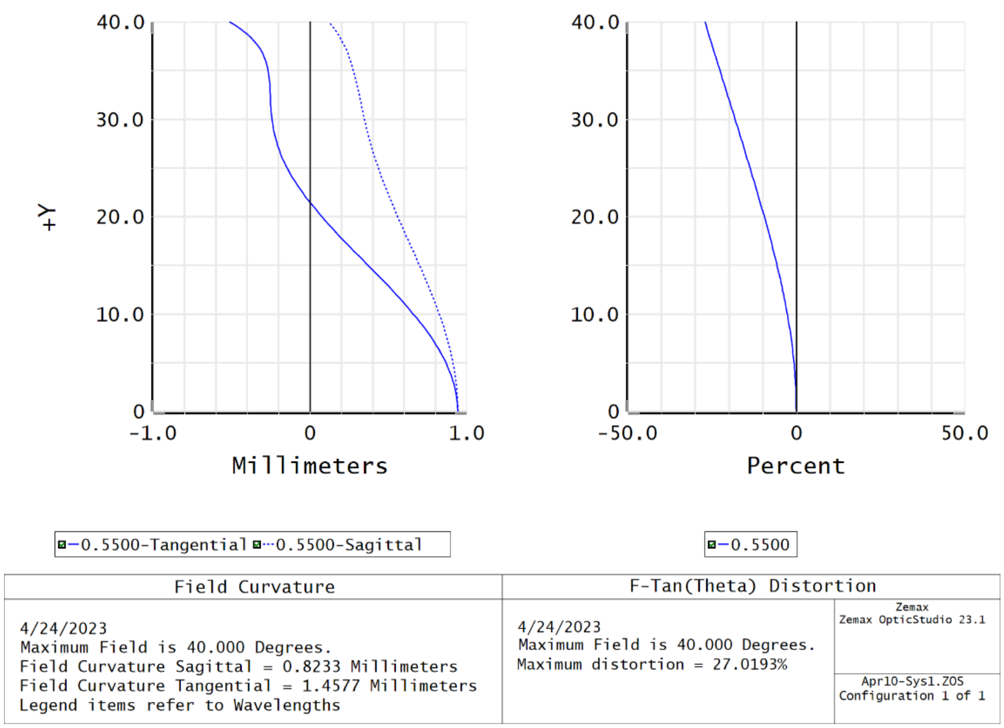


Figure 50: Field curvature versus lateral offset and angle

The numerical values of the system are included in the table below.

Type	Radius, mm	Thickness, mm	Semi Diameter, mm
Surface 1 (Lens 1)	∞	3.5	12.5
Surface 2	25.7	3.5	12.5
Surface 3 (Lens 2)	∞	3.5	12.5
Surface 4	25.7	10	12.5
Stop	∞	7	5.081
Surface 5 (Lens 3)	14.7	4.7	6.35
Surface 6	-14.7	0	6.35
Surface 7 (Lens 4)	-52.1	3.5	6.35
Surface 8	52.1	0	6.35
Surface 9 (Lens 5)	14.7	4.7	6.35
Surface 10	-14.7	7.421	6.35

Table 31: Lens Calculations

The lenses were chosen based on the availability of shapes from Thorlabs. Simulations were conducted to determine the best spacing between each of the elements to focus the rays on the sensor's face. The distance from the back surface of lens 5 to the imaging plane with the best focus is 7.421 millimeters.

5.2.4 Optical-Mechanical Design

With the optical lens system complete, we are now able to design the mounting system to attach the lens to the sensor and to properly align them. We have decided to use Autodesk Fusion360 to design the mounting system as opposed to using prefabricated lens mounting hardware. We opted to design a custom solution for several reasons. Using prefabricated mounts is significantly more expensive and would greatly increase the time to align each lens at the proper distance due to having to remove each successive lens. A benefit of designing our own mounts is the ability to create a system that allows us to adjust the spacing of each lens simultaneously. The lenses used have diameters of one inch and one-half inch, which was a variable our design was created around.

After completing the optical lens system design, the next step is to design a mounting system to attach the lens to the sensor and ensure proper alignment. To achieve this, we have chosen to use Autodesk Fusion360 to design the mounting system rather than using prefabricated lens mounting hardware. There are several reasons for this decision, with one being prefabricated mounts are significantly more expensive. Using prefabricated mounts can also drastically increase the time required to align each lens at the proper distance due to the need to remove each successive lens. This increase in steps would only get worse as the system reaches completion. Modelling our own mounts will hopefully circumvent these issues.

Designing a custom mounting system provides us with several benefits. Firstly, it allows us to create a system that enables us to adjust the spacing of each lens simultaneously, thus simplifying the alignment process. This also provides us with greater control over the final design, allowing us to optimize the mounting system to suit our specific needs. The lenses we will use have diameters of one inch and one-half inch, which we will design around. By designing the mounting system ourselves, we can ensure that the system is both cost-effective and tailored to our specific requirements.

5.2.4.1 Optical-Mechanical Design Process

Several versions of the opto-mechanical hardware were considered during the initial design phase of the project. More than 10 unique CAD designs were tested throughout, each with their own pros and cons. Many of the early designs focused on allowing the most freedom when aligning the optical components. The earliest design utilized lens mounts positioned on threaded rods and secured using nuts. This design would have theoretically allowed us to achieve the most accurate system, but in testing it remained difficult to properly monitor the alignment. Other designs were tested that secured the triplet lens configuration together but allowed it to move freely with respect to the primary lenses, however this design also proved to be more cumbersome and less effect at alignment. The final lens design was created around the CAD models provided by Thorlabs, the distributor whom we purchased the lenses from. The initial idea was to create a perfect mold to ‘sandwich’ together perfectly encasing the lenses.

The CAD models were imported into Fusion360 and the spacing between each element was adjusted to the exact distances specified in the Zemax design. A cylindrical “blank” was created around the optics extending approximately 47 millimeters. The CAD lenses were then used to remove the material within the cylinder at their precise locations. Holes were generated along the optical path, with a diameter of 11.8 millimeters and 24.5 millimeters along the half inch diameter and inch diameter optics respectively. The purpose for openings smaller than the lenses was to ensure that they would be secured and not shift locations or orientation. Threading was modeled on the back end of the casing with dimensions of one inch in diameter with 20 threads per inch and extended eight millimeters along the casing. Designing the threading to extend eight millimeters enabled the casing to be thread completely against the Raspberry Pi sensor and use the included focusing ring to make fine tune adjustments.

With the general casing designed, the model was then split into two halves, with holes passing through each half with diameters of 2.3 millimeters. These holes were created so that two m2.5 screws could be used to join the two halves of the case together once the lenses were in place. The purpose of creating holes smaller than the screws was so that the screws could thread into the plastic casing without the need to model such small features. After this, small aesthetic modifications were made to reduce the total material needed to 3D print the casing.

5.2.4.2 3-Dimensional Lens Mount Design

The figures below show the finalized casing. Figure 51 shows the whole casing modelled in Fusion360. Figure 52 shows one half of the casing and how the lenses fit in place in the design.

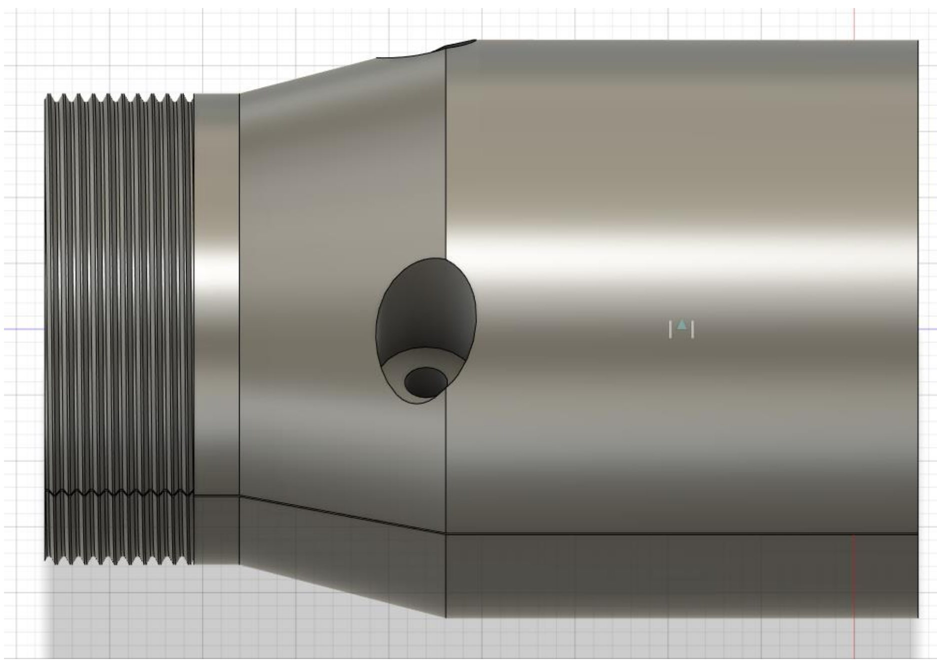


Figure 51: Lens Casing External View

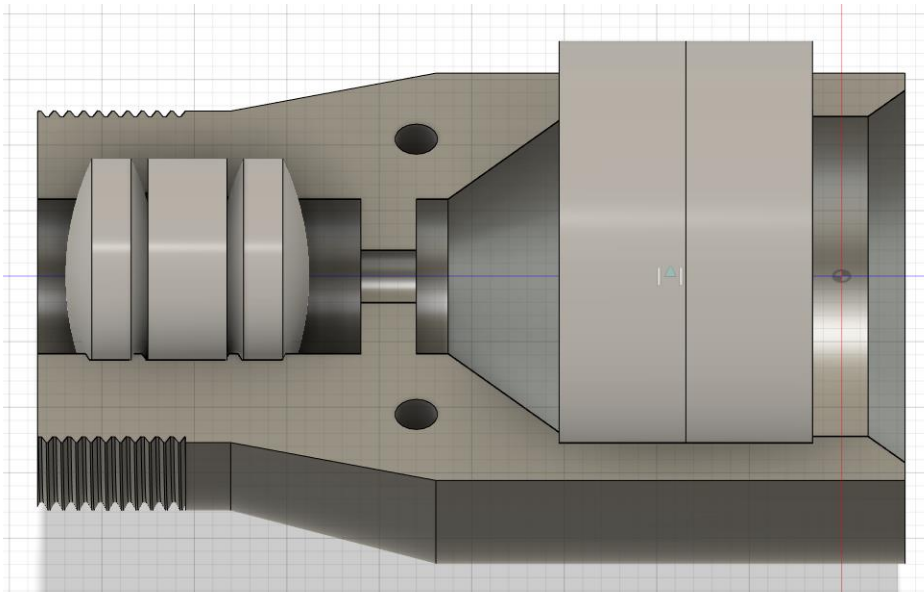


Figure 52: Lens Casing Cross Sectional View

5.2.4.3 3-Dimensional Sensor Mount Design

To mount the sensor and optical system to the Raspberry Pi casing, we chose to design our own mounting hardware. Initially we had planned to use the cutout already located on the case we chose, but through testing we found that it caused clearance issues with the whole system, so we opted to design our own. The image below shows the casing designed to protect the Raspberry Pi High Quality Sensor.

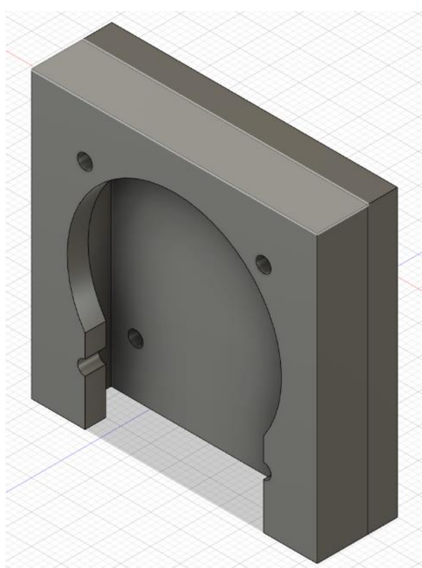


Figure 53: Raspberry Pi High Quality Camera Sensor Casing

It consisted of two halves, a front and rear plate that were secured together using two m2.5 screws. We then created a mount that utilized the ¼" – 20 tripod mount adapter that was included on the sensor to connect it to our project. The figure below shows the finalized design to connect the entire optical system. The prongs were placed between the vent grating on the display casing and then secured with a secondary piece, which was then secured using a m2.5 nut and bolt.

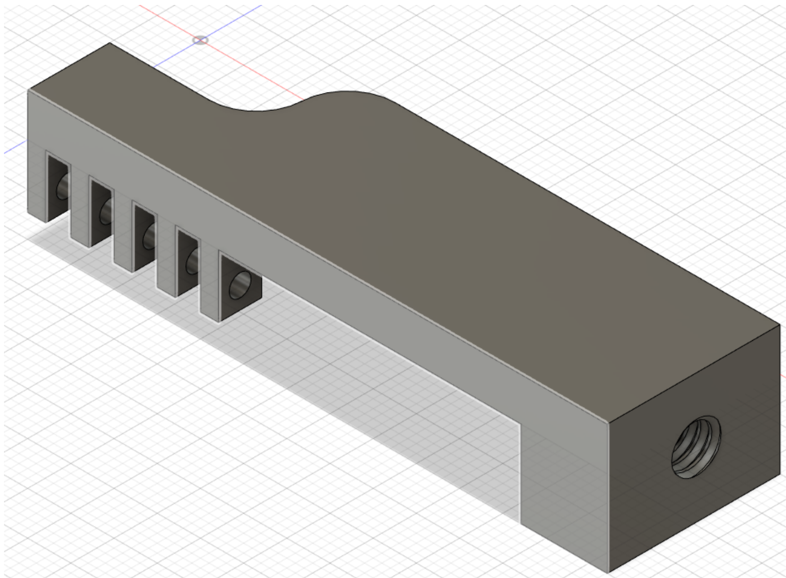


Figure 54: Mounting Hardware to Display Casing

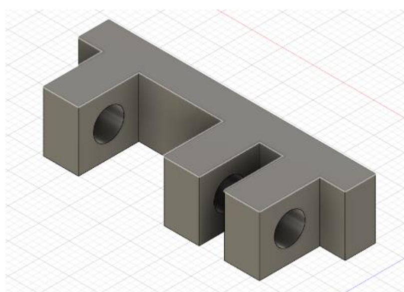


Figure 54: Securing Piece for Display Mounting

5.2.5 Prototyping

The use of 3D modeling for these mounts allows for rapid adjustments and changes once assembling the system. Issues that could potentially arise are tolerance issues causing misalignment of the optical path length and unintendedly blocking the highest angle rays from passing fully through the system. Overall, the issues faced during the prototyping phase will likely be able to be corrected by small changes in the 3-D printed models.

Rapid prototyping was made possible by utilizing a 3D printer. On average each piece on this system took less than an hour to print. This allowed us great flexibility when making small adjustments for tolerance issues and any of the issues encountered when implementing the design in the real world.

5.2.6 Final Optical Design Specification

In this project, one of the primary objectives was to design an optical system that met a set of ideal values. These values were established based on the specific requirements of the design project, and included parameters such as the field of view, image height, resolution, frame rate, maximum diameter, and F-number. The table below compares the ideal values to the current values of our system. The results of this comparison show that while some parameters fell short of the ideal values, other values were met with a considerable margin.

Parameter	Goal	Current
Field of View (FOV)	>110°	80°
Image Height	4.45 mm	4.45 mm
Resolution	2028 by 1080 pixels	2028 by 1080 pixels
Frame Rate	50 frames per second	50 frames per second
Lens System Diameter	25.4 mm	25.4 mm
Number of lenses	3 lenses	5 lenses
F-Number Range	F/1.8 - F/2.8	F/2.5

Table 32: Optical System Comparison

One parameter that fell short of the ideal value was the field of view, which decreased to 80 degrees. This reduction in field of view could have potential effects on the system's performance since a wider field of view is more desirable. However, despite this decrease in field of view, other key parameters such as the image height, resolution, frame rate, and maximum diameter remained unchanged. These parameters are crucial to the system's overall performance. Additionally, it is worth noting that the F-number of the system was defined at F/2.5. This is a significant factor in determining the system's light-gathering ability, and an F/2.5 aperture value is quite fast. This is an advantage for the system, as it allows for better low-light performance and potentially allows us to achieve the fastest shutter speeds with our sensor. Finally, it is important to mention that the number of lenses in the system increased to five lenses total. While this increase may have added complexity to the system, it also provides additional opportunities for optical design optimization. It was necessary to enable the design to be completed with lenses that are readily available as determined in the above section that a three-lens system would not have been practical for our purposes. Overall, while some parameters fell short of the ideal values, the achieved values for the optical system demonstrate a successful design that meets the application's key requirements. The areas for improvement highlighted by the comparison can serve as potential variables for future optimization in Senior Design 2.

5.3 Physical Design

Developing the physical design for TrackPack involved creating 3D CAD models to represent the enclosure that will house the 7in Raspberry Pi display, along with the board for the display, the Raspberry Pi 4, the IMU, the PCB board, and the necessary connection cables. Modeling this enclosure is intricate because it’s important that we stay within the limits of our size constraints. For the models to be an accurate representation of the final design, we must follow the size specifications provided by the manufacturers, and any potential mounting points must also be replicated accurately.

The first stage of developing the physical design is to precisely model the display, since the display will be the maximum length and height for TrackPack, while the remaining hardware will be smaller and just add to the width.

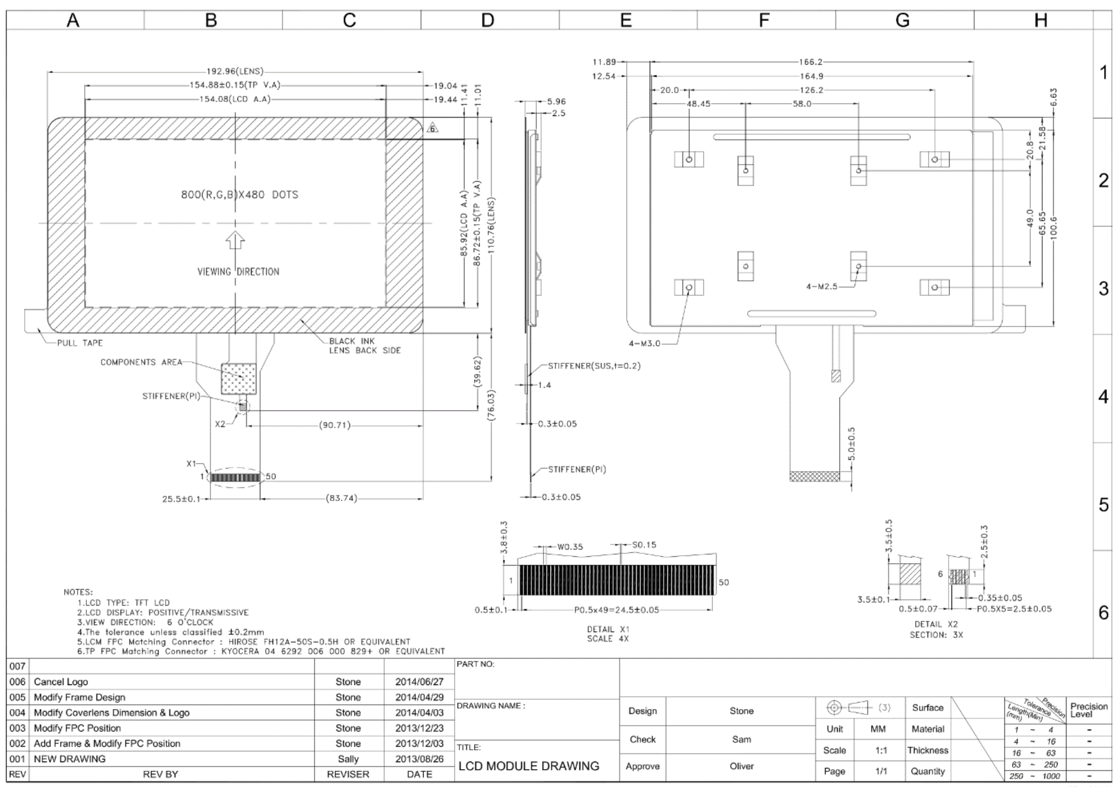


Figure 56: Raspberry Pi Touch Display Mechanical Specifications

Figure 42 is provided by Raspberry Pi and provides exact dimensions, tolerances, spacing, mounting points, and screw types.

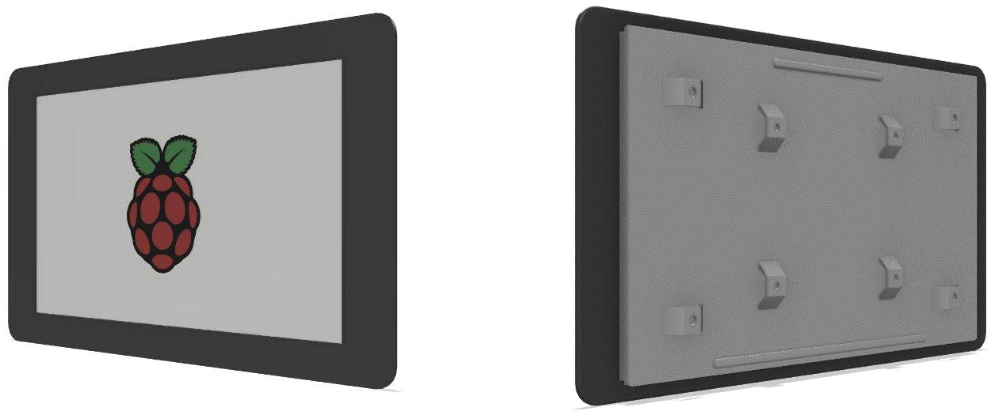


Figure 57: Raspberry Pi Touch Display CAD Model

Figure 43 models the Raspberry Pi Touch Display and was imported from GRABCAD and cross referenced against the mechanical drawing to ensure correctness.

With the display accurately modeled, we can begin modeling the anticipated housing around the length and height of the display. The width of the housing must maintain the size constraint with an applicable amount of room for the remaining hardware and sufficient cooling inside the enclosure.

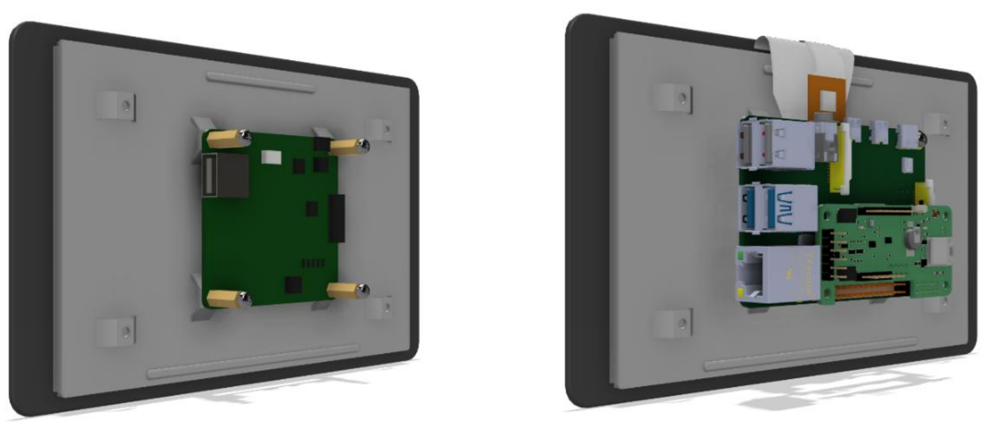


Figure 58: Raspberry Pi Touch Display with adapter board(left) and remaining hardware(right)

Figure 44 shows the final display model of the Raspberry Pi Touch Display with the adapter board, Raspberry Pi 4, and IMU mounted to the rear of the display. Using this model, we can generate accurate measurements for the width of the enclosure.

Figure 59: TrackPack Hardware Enclosure

Figure 45 shows the enclosure that will house the display, and all the hardware. This model was imported from GRABCAD and modified to work with additional hardware such as the IMU and the PCB board. We also added mounting points on the rear of the enclosure to enable us to add a windshield suction cup.

With the hardware enclosure model complete, the last and final stage of designing the enclosure is to model the windshield mount. We must note that the windshield mount is completely detachable and to be referenced as a separate entity that should not violate our size constraints.

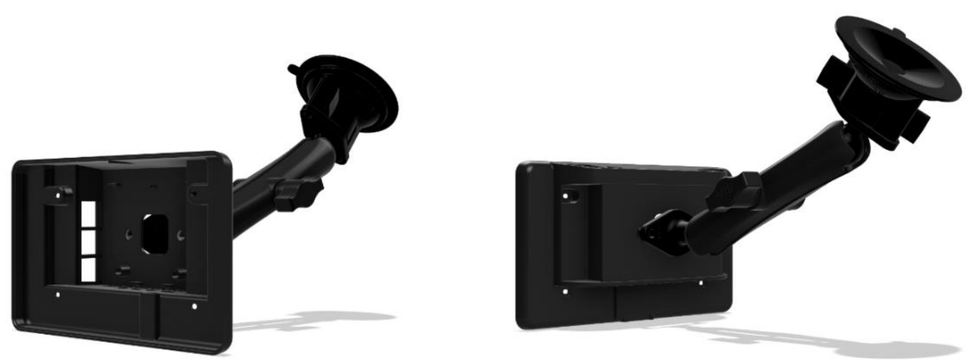


Figure 60: TrackPack Completed Enclosure with Windshield Mount

Figure 57 shows TrackPack's completed enclosure with the windshield suction cup mount. The mount is designed using models of RAM Mounts components including the RAM Diamond Ball Base, the RAM Long Double Socket Arm, and the RAM Twist-Lock Suction Cup Base. The rear of the enclosure is a square hole used for a small fan for ventilation and cooling of the hardware components. To effectively maintain proper ventilation and cooling with the windshield mount attached to the rear of the enclosure, we implemented small plastic/nylon spacers between the mount the enclosure and the mount.

5.4 Software Design

When starting TrackPack, the initial screen will prompt the user to select their mode of operation. To increase performance, we won't be pulling all the data that will be available to the user on initial startup, instead we will pull the data to be displayed based on which mode of operation the user has selected. By reading and displaying only the necessary parameters we can increase performance, reduce power consumption, and reduce the overall load on the hardware.

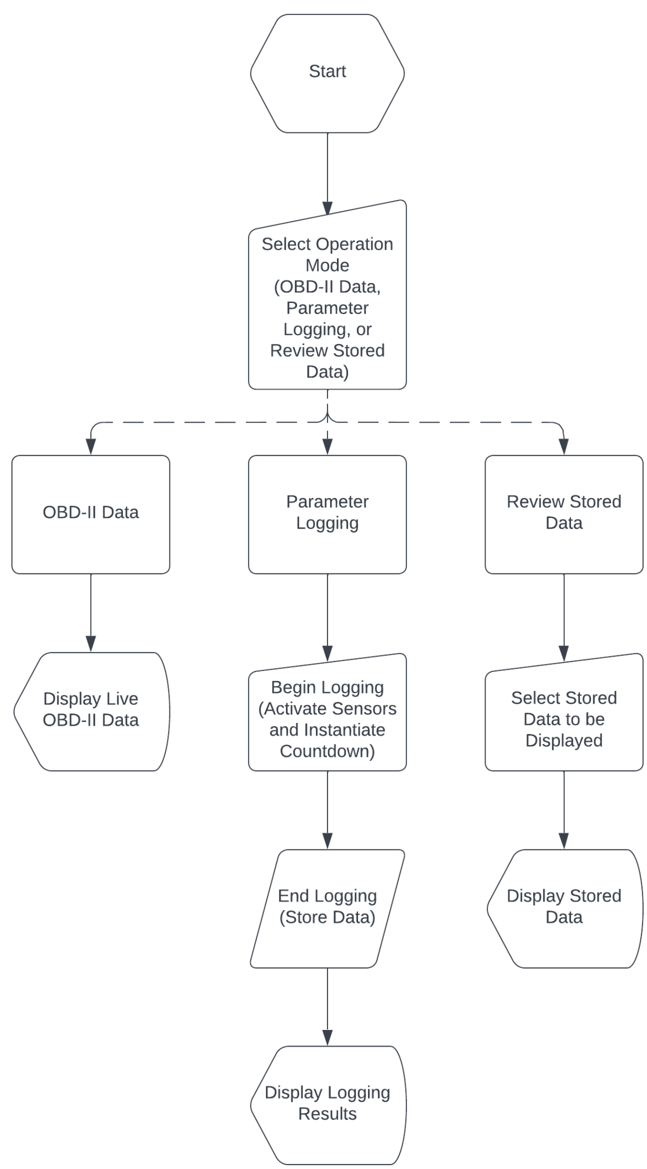


Figure 61: Flow of Software

If the user selects OBD-II Data mode, the connected OBD-II Bluetooth adapter will begin reading and displaying important vehicle parameters and diagnostic information. If the user selects Parameter Logging mode, TrackPack will initialize the GPS, accelerometer, gyroscope, and Camera in preparation to begin datalogging the measurements from these sensors. In Parameter Logging mode, once the sensors are initialized, the user will be presented with the option to begin recording. Once the recording begins, TrackPack will display a countdown timer for the user to start, once the test is complete TrackPack will automatically store the data to be reviewed later. If the user selects Review Stored Data, the various sets of stored data will display allowing the user to specifically select which set of data they want to review and then display that set of data.

5.4.1 Software Interface

The design behind the software interface is to keep it as minimalistic as possible to promote ease of use and to streamline the user experience. The Mode Selection interface will be the initial interface when loading the TrackPack GUI. On this interface, the user can select which mode of operation they would like. This interface along with the following interfaces were designed with large buttons and clear heading titles to promote usability on the Raspberry Pi 7in Touch Screen Display. The entirety of the GUI application will be written in Python3 using the available Python libraries. The following prototype software interfaces were designed using Figma.

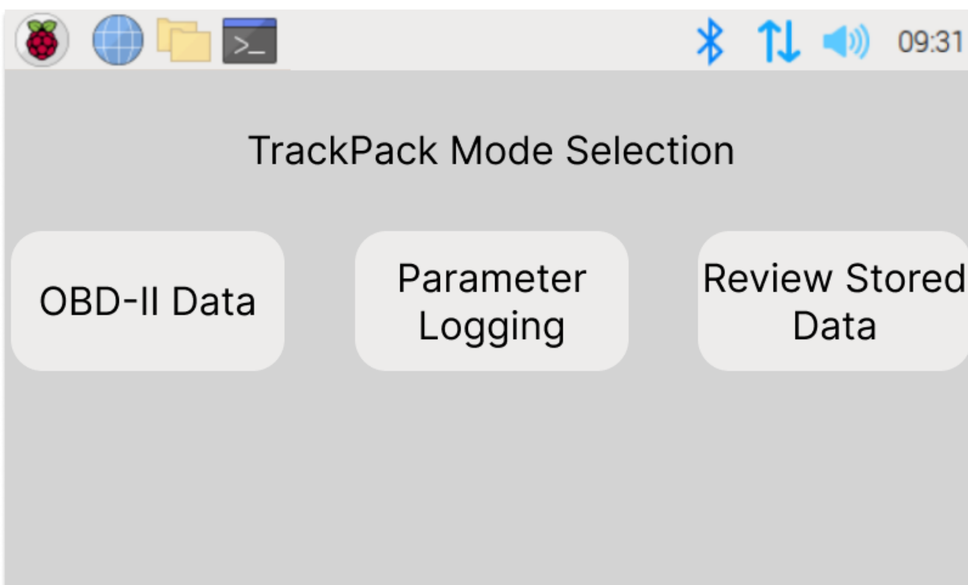


Figure 62: TrackPack Mode Selection/Initial Interface

The OBD-II Data button will reference the OBD-II live data interface. When switching to the OBD-II live data interface, TrackPack will check to ensure that the Bluetooth module built into the Raspberry Pi has established a connection with Bluetooth OBD-II adapter. When the Bluetooth connection is established and confirmed, we will begin displaying the primary OBD-II live metrics to the user. Below these live metrics are options to use the remaining functionality of the OBD-II adapter which are to view more data parameters and view the diagnostic information/fault codes.

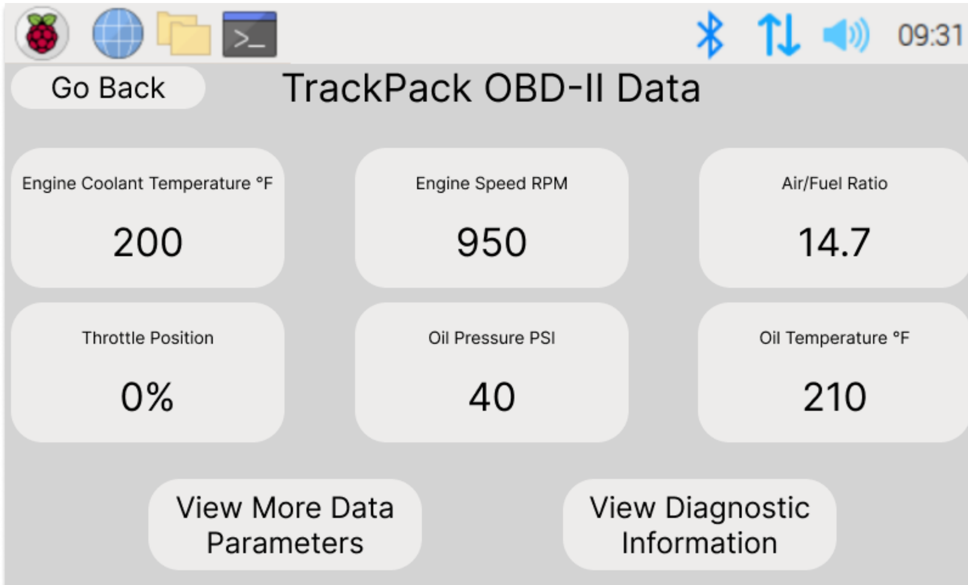


Figure 63: TrackPack OBD-II Live Data Interface

The View More Data Parameters button will reference the OBD-II More Data interface. This interface is designed to display other less common OBD-II live metrics.

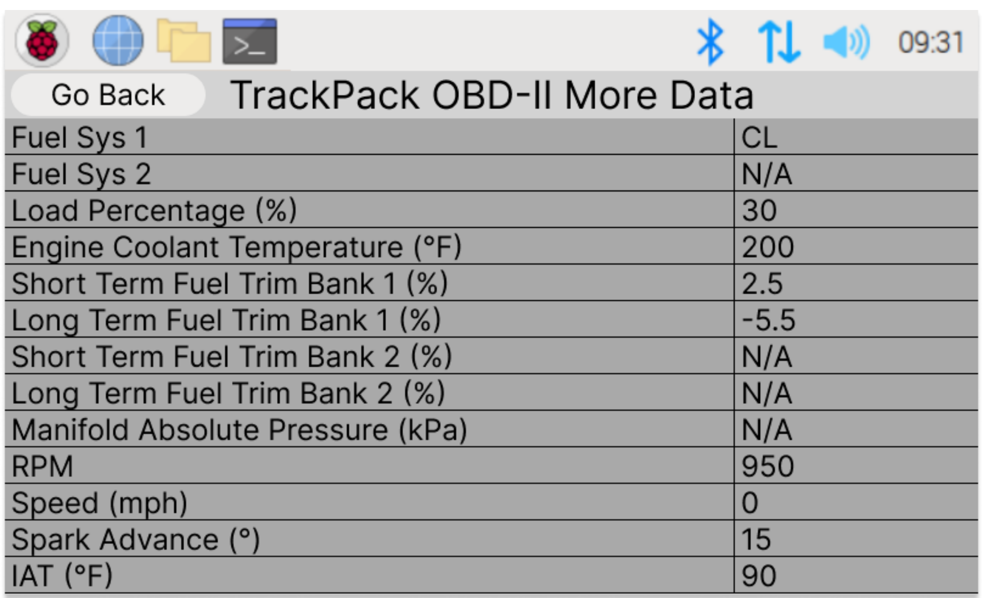


Figure 64: TrackPack OBD-II More Live Data Interface

The View Diagnostic Information button will reference the OBD-II Diagnostic Info interface. This interface is designed to display any diagnostic information/fault codes that are stored in the vehicles ECU. If the vehicle has no current stored codes, then this interface will notify the user that there are no stored fault codes in the vehicle’s computer.

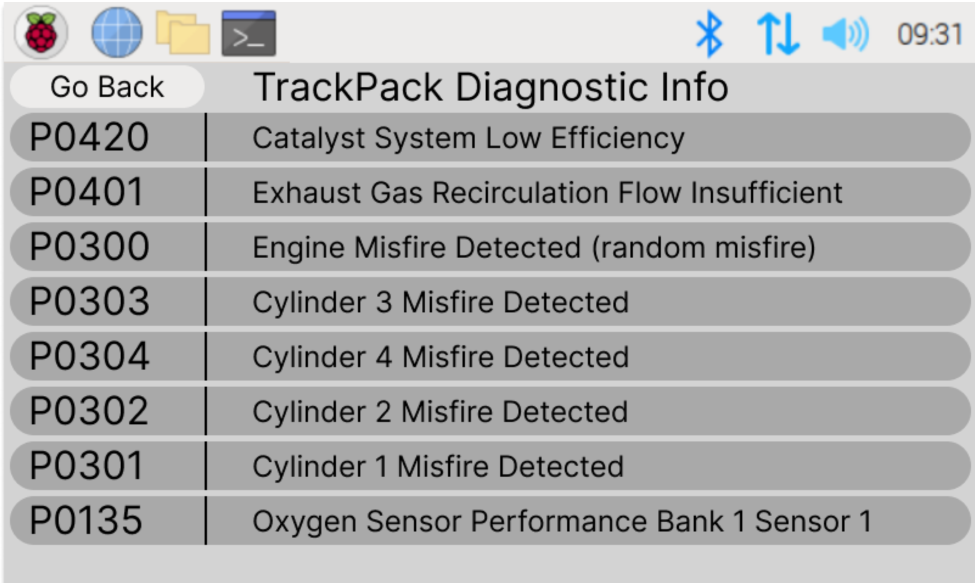


Figure 65: TrackPack OBD-II Diagnostic Info/Fault Code Interface

The Parameter Logging button back on the mode selection interface will present the user with a button that allows them to begin logging their parameters. Upon pressing the ‘begin logging’ button, a countdown timer will begin to notify the user when to start accelerating their vehicle. The goal of this interface is to use the data from the sensors to determine when the vehicle begins decelerating to notify TrackPack to end logging, however, we may have to implement a settings menu that allows the user to set a predetermined amount of time before the parameter logging automatically ends.

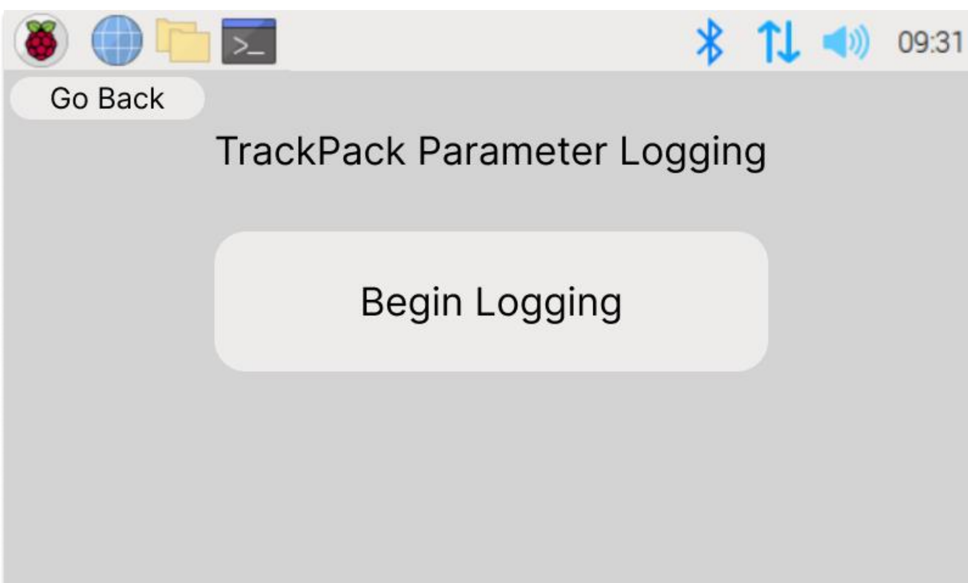


Figure 66: TrackPack Initial Parameter Logging Interface

When the countdown completes after the user selects the ‘begin logging’ button, TrackPack will switch to the Parameter Logging interface to display the live logging data to the user. When the logging is complete, this data will remain on the display and automatically be stored to the device using the date the log was completed and an index number if there were multiple logs on the specified date. The Parameter Logging interface will display the live timer for the current log, with the metrics from the sensors below. Current speed in mph can be found at the bottom left, and the acceleration in g-force can be found at the bottom right.

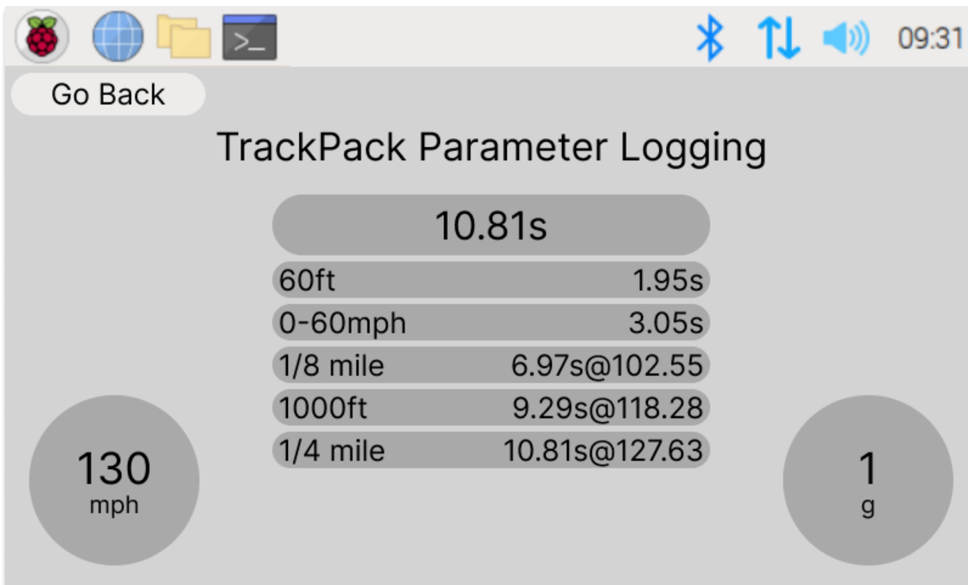


Figure 67: TrackPack Parameter Logging Interface

When the countdown completes after the user selects the ‘begin logging’ button, TrackPack will switch to the Parameter Logging interface to display the live logging data to the user. When the logging is complete, this data will remain on the display and automatically be stored to the device using the date the log was completed and an index number if there were multiple logs on the specified date. The Parameter Logging interface will display the live timer for the current log, with the metrics from the sensors below. Current speed in mph can be found at the bottom left, and the acceleration in g-force can be found at the bottom right.

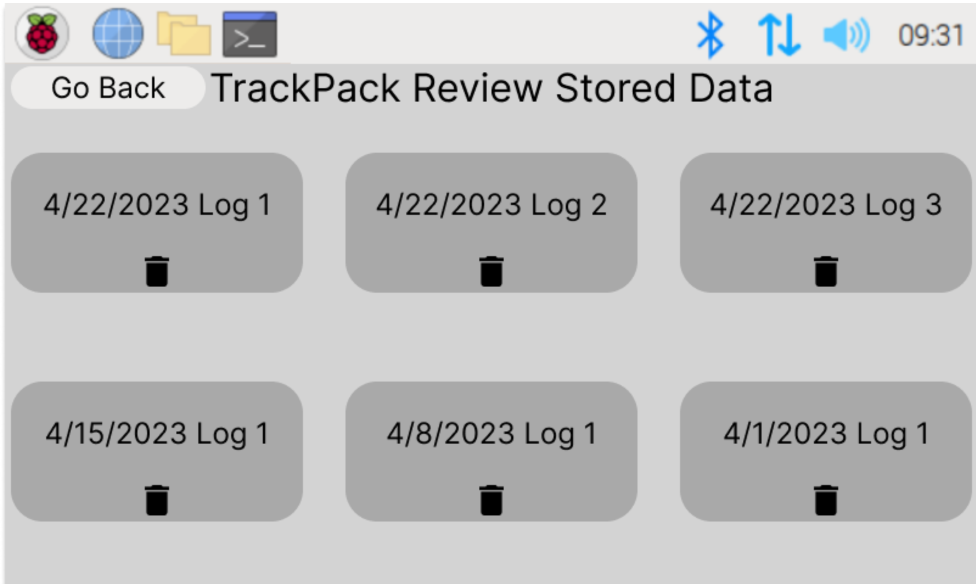


Figure 68: TrackPack View Stored Data Interface

6 Overall Integration and Testing

Accompanying the MSP430F168IPM MCU and the Raspberry Pi 4, we will have a BerryGPS-IMU GPS and 10DOF. IMUs are composed of a combination of accelerometers, gyroscopes, and magnetometers. These three sensors work together seamlessly to provide notable information about an object's movement and position.

In our system, we have carefully integrated the MSP430F168IPM and Raspberry Pi 4 with our selected IMU that also includes a built-in GPS module. The BerryGPS-IMU is an advanced IMU that contains an onboard GPS module with excellent accuracy, low power consumption, and a compact form factor. The 10DOF aspect of the IMU on the other hand is a sensor that can measure ten degrees of freedom, including three-axis acceleration, three-axis angular velocity, three-axis magnetic field strength, and barometric pressure.

Our research project will benefit from the synergy of these components. The combination of the MSP430F168IPM and Raspberry Pi 4 provides the computational power needed to process the data generated by the IMU subsystem. The BerryGPS-IMU, with its high accuracy, is perfect for tracking the location and movements of our test subject, while the 10DOF sensor will provide us with valuable data about the surrounding environment. A prefabricated IMU is ideal instead of building one on our own for the purpose of direct readings and processing power preservation. Since we chose an off the shelf IMU, we can avoid hurdles such as implementing the sensors in a way where we would have to ensure that we can process the data into our MCU to produce the correct readings. With a prebuilt IMU, this process of data transmission is taken care of.

In the following section, we will discuss the different functions and integrations of each sensor incorporated in the MCU and IMU subsystem.

6.1 Integration

In this section, we'll introduce the integration methodology utilizing all the previous parts and design schematics/criteria that we developed previously. The goal of this section is to create a fundamentally functional design for TrackPack regarding the respective hardware, electrical, optical, and software designs.

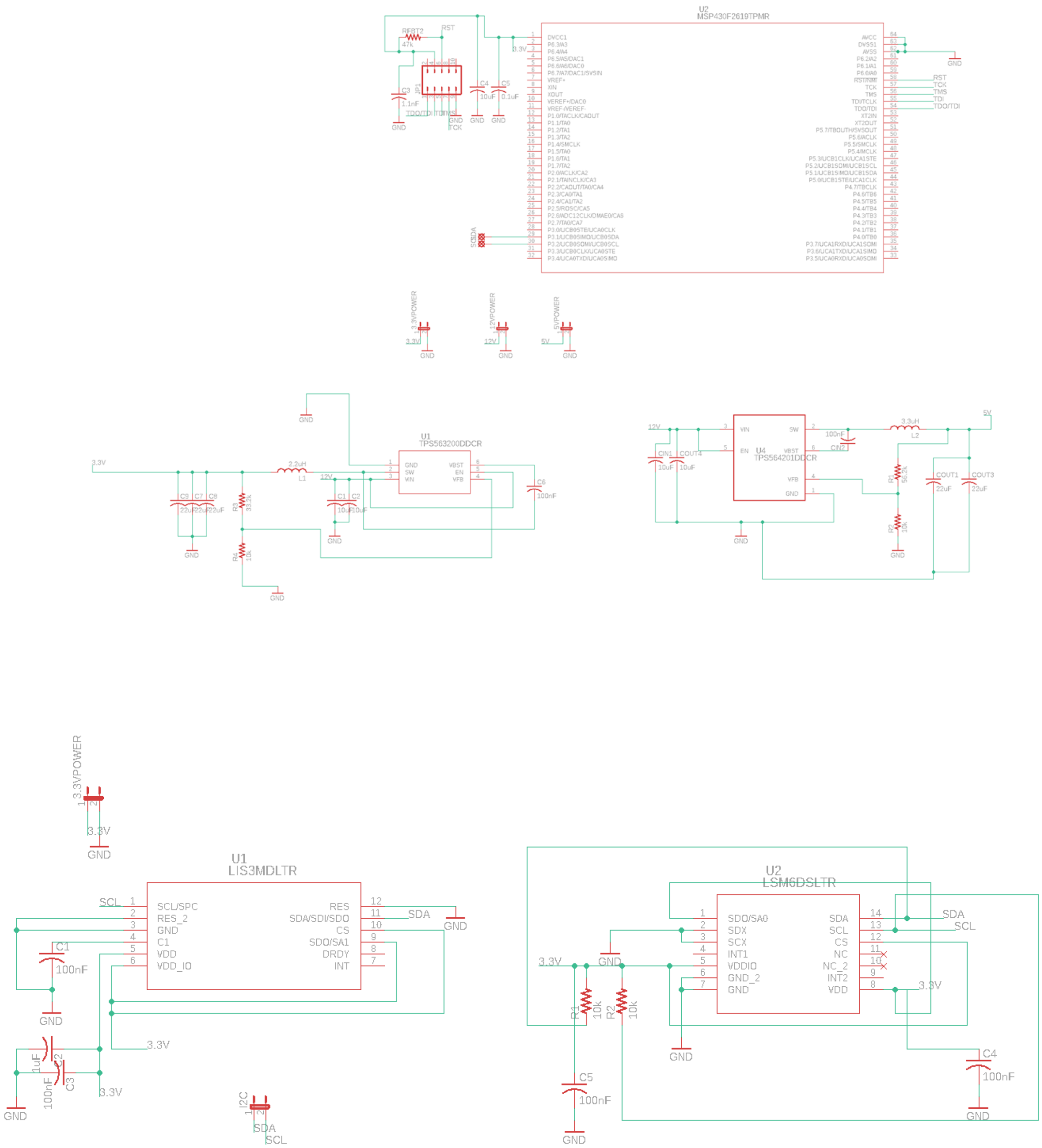


Figure 69: Integrated System Schematic

The design of the overall integrated schematic focused on keeping the voltage regulators in the center as all the components would tap into them for the power input. The SDA and SCL pins of the devices are connected appropriately to make use of the I2C communication and limit the number of pins that we would need to incorporate each device. The Raspberry Pi 4 camera module that we used to attack our lens to is not pictured here as it will not be attached to the PCB but will take the 5V power from the board as an output to power it and the display.

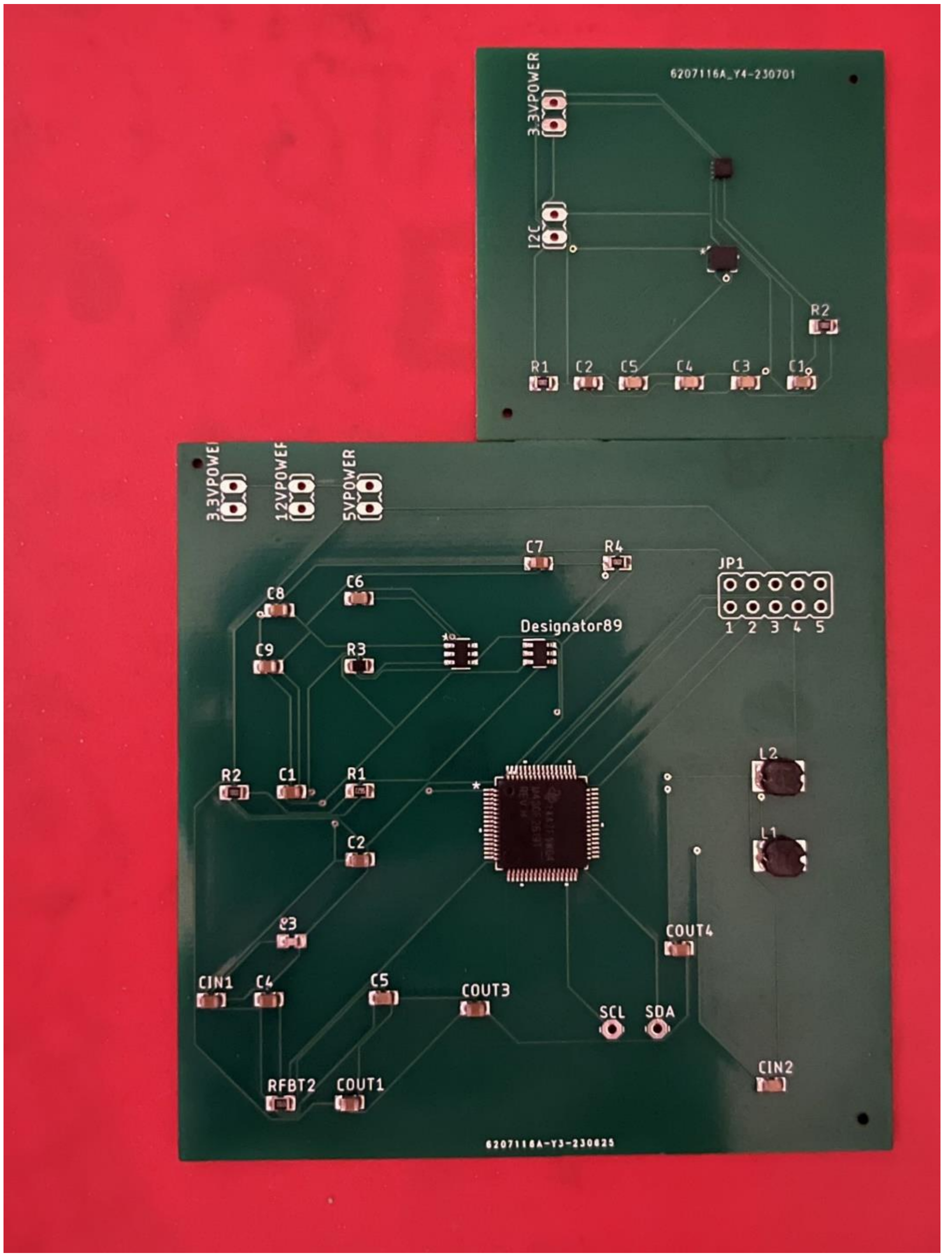


Figure 70: PCB design

When making the final Board layout, keeping a tight area was important was we need the board to fit into the housing and as stated in the constraints be small enough to not impede the driver/s view of the road. Most of the components were pushed to the center while keeping space to avoid width and clearance issues between the larger GPS and accelerometer modules. Components were also kept away from the edge to avoid contact with the housing except the volage regulator inputs which were placed in the corner of the board to give easy access to the power supply. Once the parts were placed, we used the autorouter function to generate the traces between the components that would give us a close to 100% value while limiting the number of vias. Any remaining air wires, or clearance issues were routed or placed manually and then ran the autorouter process again to reach the final layout.

6.1.1 PCB Integration

The manufacturing of our PCB is a straightforward process, after completing the PCB design on Autodesk EAGLE we can generate and export a Gerber file. The selected PCB manufacturer uses the Gerber file to accurately replicate and create the board. The manufacture time on the PCB can be anywhere between one-day and multiple weeks, so it's vital to TrackPack's timeline that the PCB design is completed early to mitigate any potential delays involving manufacturing times. There are several PCB manufacturers such as PCBWay, JLCPCB, and Advanced Circuits. While it's important that the PCB is manufactured in a timely manner, it is also important that we select a manufacturer that is cost-effective and reliable.

6.1.2 Software Integration

There are various aspects of the software design that we had to implement to integrate each component of the TrackPack. This section focuses on the software design of the IMU and camera with the MSP430 and Raspberry Pi microcontroller, respectively. In this analysis, we will cover how we collected and stored various parameters such as acceleration, orientation, and visual footage. The collected data can then be analyzed to improve vehicle performance and driving skills.

TrackPack will require multiple sensors that will operate independently and produce the measurement to be displayed. Other than sensor fusion, which will be done separately by the IMU to produce accurate and precise measurements, each sensor will produce its own measurements and will not have any reliance on another sensor.

6.1.2.1 Software Sensor Configuration

The collection of data from the IMU begins with connecting the IMU to the MSP430 microcontroller, where we transfered the data. The connection of the IMU to the MSP430 is created using I2C communication protocol. The MSP430 will be considered as the master device and the IMU will serve as the slave device. We utilized the two-wire interface with a data line, SDA, and a clock line, SCL. Configuration of the IMU is required before we can begin reading the parameters from the IMU. Provided with the BERRYGPS-IMU V4 is a datasheet with the corresponding register addresses and values needed to configure the IMU. These register addresses are needed to correctly program the MSP430 using the C programming language so that the MSP430 can read and store/send the correct data to the correct location. The configuration has respective settings created for each sensor which includes the range of motion, sampling rate, and filter settings. To transfer the register addresses and values to the IMU we will send the data through an I2C bus.

After the preliminary configuration is complete, we can begin reading data from each sensor. We will use the I2C protocol which will send a read request to the IMU for the sensors' data. Upon the data request, the IMU will transmit the unrefined data which will be stored in the memory of the MSP430. After receiving the unrefined sensor data, it is necessary to convert and process those values using calibration parameters before they can be used.

We plan on storing the data from the IMU and the video footage on the same external storage card located on the Raspberry Pi. Since the video footage will be stored directly onto the microSD on the Raspberry Pi, it is necessary to transmit the information from the MSP430 to the Raspberry Pi. The data transmission is achieved by utilizing the I2C serial communication protocol where the MSP430 is the transmitter, and the Raspberry Pi is the receiver. Synchronization between the MSP430 and the Raspberry Pi is vital to ensure that the data is correctly transmitted and received. A simple communication protocol such as adding start and stop bytes to each message to ensure that the data is properly framed and can be easily parsed can help mitigate the probability of improper data transmission.

We be used the Raspberry Pi High Quality Camera which features the Sony IMX477 CMOS sensor. The CMOS sensor integrated into the CSI port embedded into the Raspberry Pi. On initial setup, we had to install the appropriate software on the Raspberry Pi which will enable a valid interaction with the camera. Once the camera software module is initiated, we moved forward with activating the camera interface on the Raspberry Pi. Implementing the camera interface allows access to special footage settings such as frame rate, image size, object detection etc. The Raspberry Pi will then be capable of processing image/video data. There are popular open-source computer vision libraries such OpenCV which can assist with processing data.

6.1.2.2 Raspberry Pi Software Configuration

Once we have all the peripherals configured and the microcontroller software implemented to read data from the BERRYGPS-IMU V4, we must configure the Raspberry Pi to take in the data and store/display the data for the user. We're going to be utilizing the Raspberry Pi official supported operating system, Raspberry Pi OS (formerly known as Raspbian). The Raspberry Pi foundation has made installing an OS on the Raspberry Pi quick and easy using their Raspberry Pi Imager tool. Once we use the Raspberry Pi Imager tool to install the OS on the SD card, we're ready to begin integrating the sensors and image processing on the Raspberry Pi.

The Raspberry Pi OS allows the seamless and real-time integration of data between the BERRYGPS-IMU V4 and the Raspberry Pi. By interpreting the data on the Raspberry Pi, the user can obtain valuable insights for various applications, such as tracking movements and navigating.

6.1.2.3 Raspberry Pi GUI

Before developing the graphical user interface, we'll be using Python to create scripts that will pull in the data that's being read from the sensors and the GPS module. This process involves using various libraries and modules in Python to interface with the sensors and GPS module and extract the necessary data. After testing the scripts in the terminal to ensure that the readings are being correctly read and displayed, we can then begin creating an all-in-one interface that will allow us to display the data easily and conveniently on the display. This is an important step as it allows us to identify and fix any issues with the data readings before they are integrated into the GUI. We can combine the Python scripts that we developed previously and using the Python library Tkinter, we can create a streamlined interface to display all our data to the display. This interface can include visual elements such as graphs and charts to make it easier to interpret and analyze the data. Overall, using Python to create scripts and Tkinter to develop a GUI provides an efficient and effective way to monitor and display sensor and GPS data.

6.1.2.4 Software Integration

Once the wireframe designs of the software are brought to life using the Python3 Tkinter library, we can then begin integrating the sensors, OBD-II adapter, and camera. The sensors on the IMU will be processed by the MSP430 and sent through an I2C bus to the Raspberry Pi. Once the Raspberry Pi receives the information from the MSP430, a Python script on the backend of the GUI will process all the information and display it to the user. Using the software to integrate the Raspberry Pi High Quality is slightly more straightforward, primarily because the camera is designed to easily integrate with the Raspberry Pi 4. Once the Raspberry Pi camera is connected, we can utilize the Python library Picamera2 to access the camera system. Picamera2 is built on the open source libcamera software stack, however, Picamera2 is specifically designed for integrating the Raspberry Pi 4 and the Raspberry Pi lineup of cameras. Picamera2 will simplify the software integration of the camera, allowing us to develop our application more easily with the camera's functionalities. To integrate the OBD-II Bluetooth adapter to the Raspberry Pi 4 on the software, we'll be using pyOBD which is an open source OBD-II compliant vehicle diagnostic program, completely written in Python. The downside the pyOBD is that it was developed over 9 years ago in Python2, so we'll need to rework the project to make necessary modifications to work with Python3 and the Raspberry Pi 4.

6.1.3 OBD II Integration

Selecting a versatile OBD II Bluetooth scanner is important for the flexibility and diversity of TrackPack. TrackPack's intended design is to be used across many vehicles, the idea is to have minimum constraints on vehicle compatibility type. For the sake of achieving versatility, we chose to use the ELM327 Bluetooth OBD II connector since this scanner was shown to be the most cost-effective choice with the advantages of wide range capability. The table below shows the different types of vehicles protocols and the corresponding protocols they are compatible with.

OBD II Protocols	Compatibility
ISO 15765 (CAN bus)	Mandatory in US cars since 2008 and is today used in many cars
ISO14230-4 (KWP2000)	The Keyword Protocol 2000 was a common protocol for 2003+ cars in e.g., Asia
ISO9141-2	Used in EU, Chrysler & Asian cars in 2000-04
SAE J1850 (VPW)	Used mostly in older GM cars
SAE J1850 (PWM)	Used mostly in older Ford cars

Table 33: Types of Vehicle Protocols

The test subjects we used were all compatible with the following OBD II protocols: ISO 15765 (CAN bus), ISO14230-4 (KWP2000), and ISO9141-2. OBD II scanners can connect to these ports on your vehicles and identify the trouble code from any manufacturer that uses one of the OBD-II protocols.

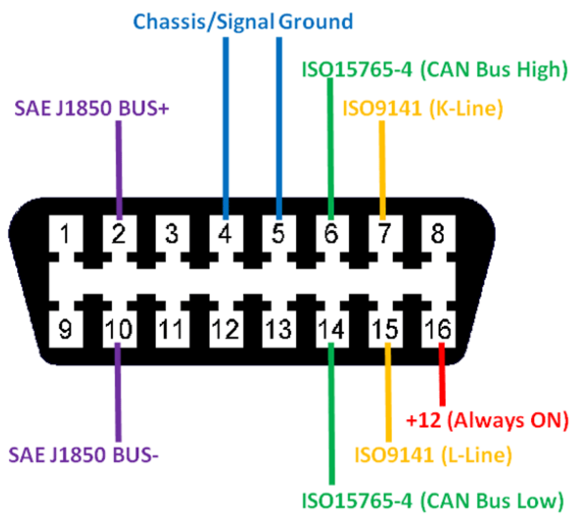


Figure 71: OBD II Pinout

Pin	Description	Pin	Description
1	Vendor Option	9	Vendor Option
2	J1850 Bus +	10	J1850 BUS
3	Vendor Option	11	Vendor Option

4	Chassis Ground	12	Vendor Option
5	Signal Gorund	13	Vendor Option
6	CAN J-2234 High	14	CAN J-2234 Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 Low
8	Vendor Option	16	Battery Power

Table 34: OBD II Pinout Labeled

The diagram and table above display what a general OBD II port looks like and the labeling and purpose of each pin, respectively.

The ELM327 provides the functionality needed for the intended design of TrackPack. Considering the microcontroller unit, we are using is equipped with Bluetooth capability, we were more inclined to include the ELM327 scanner in our design. As we continue to work on the TrackPack project, we have discovered that the ELM327 Bluetooth OBD II connector will be an essential component of our system. Our plan is to connect the OBD II scanner with the Raspberry Pi 4 Model B to provide the system with real-time access to the vehicle parameters. This way, the TrackPack system can retrieve relevant vehicle data and use it to form a comprehensive analysis of the vehicle's performance. By gaining authorized access to a vehicle's parameters, the scanner can transmit this data to the Raspberry Pi, which will then use it to determine the current state of the vehicle. This will be immensely beneficial for the TrackPack system, as it will be able to keep track of the vehicle's performance and make necessary adjustments as needed.

The Bluetooth scanner will be used to retrieve the performance parameters of the vehicle and transmit the data to a built-in display on the TrackPack. By displaying the vehicle's performance parameters in real-time, the TrackPack can help the driver take corrective action to prevent any potential problems that may arise. This will be particularly useful in the case of possible error codes that would induce a check engine light or any other possible problems that may occur. The scanner will aid in narrowing down potential issues with the vehicle in real-time, thereby enabling the driver to address them promptly.

P0100	Air flow Circuit Malfunction
P0104	Malfunction of the air mass meter
P0105	Intake Manifold Absolute Pressure Sensor Circuit / Intake Manifold Barometric Pressure Malfunction
P0109	Intake Manifold Absolute Pressure Sensor / the barometric pressure in the intake manifold
P0113	High intake air temperature sensor
P0114	Inlet air temperature sensor fault

P0115	Engine Coolant Temperature Sensor Circuit Malfunction
P0120	Throttle Position Sensor Circuit Malfunction / Switch A
P0130	Malfunction of the oxygen sensor circuit (bank 1, sensor 1)

Table 35: Sample Diagnostic Trouble Codes

The diagnostic trouble codes in the table above, can assist the driver of the test vehicle in identifying possible issues when the diagnostic trouble code indicator appears on the vehicle's dashboard.

We used the Raspberry Pi to gather the information retrieved from the ELM327 connector. On initial setup, we connected the scanner to the single board computer, since the Raspberry Pi 4 Model B has Bluetooth compatibility, this process is simple. We utilized pyOBD which is an open source OBD II compliant scan tool software. The software is written in Python, and it is designed with compatibility with the ELM327. pyOBD will allow for communication from the vehicle's on-board computer to the Raspberry Pi via the ELM327. After successful connectivity, we must then identify the serial port the device is connected to. This is done by connecting the Raspberry Pi to a display to run the respective Python commands. Within the Python package, we can execute a series of commands that provide us with the option to query various parameters and information about the vehicle.

The integration of the ELM327 Bluetooth OBD II connector with the Raspberry Pi 4 Model B is crucial to the success of the TrackPack project. Its ability to retrieve vehicle parameters in real-time, coupled with the scanner's seamless connection to the on-board computer of the vehicle, makes it an ideal choice for our system. The benefits of having authorized access to a vehicle's parameters is pertinent, and we believe that the TrackPack system is enhanced with the help of the ELM327 Bluetooth scanner.

6.2 Testing

6.2.1 Optical System Testing

To evaluate the optical system during alignment and once the best possible focus is achieved, we collected a variety of images that will be brought into MATLAB to determine qualities such as the resolution of the system and the levels of distortion. To quantitatively measure the resolution of the system, we utilized a variety of resolution targets.

There are several different types of optical resolution targets available, each with its own specific features and purposes. Some of the most common types include:

- USAF resolution targets: These targets consist of a series of black and white bars arranged in groups of varying line pairs per millimeter. They are widely used for testing the resolution of cameras and lenses.
- Ronchi ruling targets: These targets consist of a series of equally spaced parallel lines and are often used to test the optical quality of telescope mirrors.
- Grid targets: These targets consist of a grid of equally spaced lines and are often used to test the resolution and distortion of wide-angle lenses.
- Star targets: These targets consist of a pattern of stars of varying sizes and orientations and are often used to test the resolution and contrast of optical systems.

For our purposes we tested our system again resolution targets such as the USAF resolution target, grid targets, and star targets. The targets will be placed at set distances from the lens system and images will be captured. The sensor will be tested in a variety of low light and bright light conditions to evaluate the performance in any situation we might encounter. The images of these resolution targets will be imported into MATLAB. Using the image processing toolbox included in MATLAB we can measure the digital image resolution, or how many pixels the target face markings cover. Using these measurements, we can then determine the exact resolution of the entire system.

6.2.2 Power Testing

To test the ability of the PCB's ability to regulate voltage into the 3.3V we need for most of the components and the 5V to power the Raspberry Pi for prototyping we used a breadboard to test the voltage. The Senior Design lab has the Keithley 2230-30-1 Triple-Channel Power Supply available that we can use to simulate a 12V input as well as read out the output voltage on the oscilloscope. The Raspberry Pi 4 requires a 5V DC input with a ±5% tolerance (4.75-5.25Volts). To see the voltage regulation, we measure the output on an open circuit with no load and against a full load condition similar to the needs of our design. For the breadboard we had a load of resistors to act as the power dissipation. The 3.3V rail has a greater tolerance for the components but will powering significantly more devices.

Device	Voltage Requirement V	Current Requirement A
Raspberry Pi 4	5V	3A
BerryGPS-IMU v4	3.3V	71mA
MSP430F168IPM	3.3V	48mA

Table 36: Device Power Requirements

The 5V rail at 3A gives us a resistor of 1.666 Ω with a minimum of 15W power rating at the load. 3.3V rail at 0.261A gives a resistor of 12.64 Ω with a minimum of 1W rating. To simulate a full test of the voltage regulation, the 3.3 volt regulator will be powered by the 5 volt rail , and the 5 volt rail will be adjusted so the total output is 15W while still sending 1W to the 3.3V rail.

$$15W - 1W = 14W => \frac{5V^2}{14W} = 1.7857\ \Omega$$

The test was then performed on a 2Ω resistance.

6.2.3 MSP430 Testing

To get experienced with the language that we plan to use in programming the GPS, accelerometer, and the gyroscope, we want to test out the code on a similar MSP microcontroller. To do this we used the MSPEXP430G2ET launchpad to test a generic setup to see how we would transmit our data back and forth from the MCU to our data retrieving components. In order to program the MSPEXP430G2ET we be used the provided USB port to connect the device to CodeComposer also available in the Senior Design Lab to set up our functions for establishing connection between the two-device addresses and debugging.

The board has large header pins that make connecting a design through jumper cables to a breadboard very easy for the user. We connected VCC to a 3.3V rail powered by the Keithley 2230-30-1 Triple-Channel Power, in the Senior Design Lab. With the plan of having the launchpad act as the master in this design we would need a component to simulate the slave since we do not have the IMU modules to test. To perform a simple test, we looked for small I^2C compatible devices that would be easy to test and debug on that has similar operating specs to ensure the function of the design. We chose a small Atmel chip the AT24C02D that would be easy to test out on a breadboard. IT provides 8192 bits of serial electrically erasable and programmable read-only memory (EEPROM) and works well in low power modes. In Code Composer we needed to first set up our code that will require device structure as a parameter so that it can support multiple slave devices on the same bus. Receiving data requires the master to send the START condition and slave device address byte with the R/W bit. Once the entire address is obtained, we can use it to store data to the EEPROM.

The function we set up receives the slave's device's address begins by setting the slave device address in the UCB0I2CSA register. The following transactions will therefore be directed at this device. To support all three I2C transaction formats we need to first consider the transmit buffer. If there are bytes to transmit, these are sent first, so check the size of the transmit buffer is greater than zero – if so, transmit the buffer. The actual writing of the buffer to the hardware is broken out into a separate function for the sake of keeping functions small and readable. Once the transmit is complete, and if there are no errors, then it's time to see if the master needs to read any data from the slave. If so, then call the receive function. If there are no bytes to receive, then the transaction is complete, and the master should issue the STOP condition by setting UCTXSTP in the UCB0CTL1 register.

Master transmitter slave receiver: The transmit buffer will have data and therefore the length should be non-zero. Data will be transmitted to the slave. The receive buffer will have a length of zero so master does not receive any data from the slave. Therefore, immediately after the transmission is complete the STOP condition will be set.

Master receiver slave transmitter: The transmit buffer will have a length of zero. Therefore, the transmit section of the function will be skipped. The length of the receive buffer should be greater than zero and therefore the master will read that number of bytes from the slave and then the STOP condition will be set.

Combined format: In this case both the transmit and receive buffers are greater than zero. Start by transmitting the required number of bytes. If no errors have occurred, a repeated START condition will be issued and the master will receive data from the slave. Once that is complete, the STOP condition will be set.

AT24C02D SPECS

- Low-Voltage Operation:
  - VCC = 1.7V to 3.6V
- Internally Organized as 128 x 8 (1K) or 256 x 8 (2K)
- Industrial Temperature Range: -40°C to +85°C
- I2C-Compatible (Two-Wire) Serial Interface:
  - 100 kHz Standard mode, 1.7V to 3.6V
  - 400 kHz Fast mode, 1.7V to 3.6V

With the AT24C02D on the breadboard, we connected the p1.6 and p1.7 from the launchpad (SDA and SCL) to pins 6 and 5 on the AT240C2D. Then in CodeComposer we create two new functions to read and write one byte to the EEPROM. We manually input the address and then the function will set the length and completes a signal back and forth between the master and slave. As well as seeing the data address that the EEPROM sends back to us in the CodeComposer window, we can also connect the connections on the breadboard to see the clock signal of the I^2C back and forth and compare with the information in the debug window.

7 Administrative Content

In this section, we will be covering the administrative aspect of TrackPack. We will introduce and outline the estimated budget for this project. Formulating a budget estimation is important so that we knew the approximation of the cost of this build. It is also important to take a closer look at the financial aspects of the project to ensure that we are within the planned budget and that there are no deviations that might impact on the project's completion.

Following the financial aspect of TrackPack, we will then discuss the planned schedule of this project. We will display the course of action with a corresponding timeline that we hope to achieve throughout the duration and completion of this design. It is imperative to establish a comprehensive schedule that will ensure the project stays on track and minimize any unexpected delays or disruptions that may hinder progress. Proper planning is essential to allow for sufficient time to complete each task efficiently and achieve the desired outcome. With a structured schedule, we can maintain our focus, organization, and momentum to meet our goals.

As part of our project management, we'll be comparing the planned schedule with the actual completion dates. With structured planning, this will assist us in making necessary adjustments to our approach moving forward.

Overall, the management section of the project is vital to its success. Being mindful of our estimated project budget and deadlines, this will help guide us in the right direction to ensure that TrackPack is completed within the allocated time and budget while also delivering the expected results.

7.1 Budget Estimates

Below, there is a table included with the estimated costs of the material needed for TrackPack. While we've done our best to ensure that these prices are accurate, it's important to note that they are not final and are subject to change once the final implementation begins. As the project's system requirements evolve, the overall cost of the system will need to be adjusted accordingly.

Item Number	Component	Quantity	Estimated Cost	Total
1	Completed PCB	1	$50.00	$50.00
2	IMU	1	$71.20	$71.20
3	Camera module	1	$50.00	$50.00
4	Lenses + mounting tube	4	$30.00	$120.00
5	Housing	1	$24.95	$24.95
6	MSP430 Microcontroller	1	$19.03	$19.03
7	OBD-II Connector	1	$13.99	$13.99
8	Jumper Wires	1	$6.98	$6.98
9	Pin Headers	1	$5.73	$5.73
10	12V to 5V - 3A Adapter	1	$8.99	$8.99
11	Display	1	$64.95	$64.95
12	MicroSD	1	$8.02	$8.02
13	Raspberry Pi 4 4GB	1	$55.00	$55.00
Total Estimated Budget:				$494.07

Table 37: Project Budget

7.2 Bill of Materials

Item Number	Component	Quantity	Estimated Cost	Total
1	Completed PCB w/MSP430 Microcontroller	1	$54.00	$54.00
2	IMU	1	$76.18	76.18
3	Raspberry Pi HQ Camera	1	$50.00	$50.00
4	Lens 1 + Lens 2 (LC1715-A)	2	$34.66	$69.32
5	Lens 3 + Lens 5 (LB1092-A)	2	$38.15	$76.30
6	Lens 4 (LD1357-A)	1	$37.84	$37.84
7	Housing	1	$26.61	$26.61
8	OBD-II Connector	1	$13.99	$13.99
9	Jumper Wires	1	$6.98	$6.98
10	Pin Headers	1	$5.73	$5.73
11	12V to 5V - 3A Adapter	1	$8.99	$8.99
12	Display	1	$79.94	$79.94
13	MicroSD	1	$8.58	$8.58
14	Raspberry Pi 4 4GB	1	$63.00	$63.00
15	Raspberry Pi HQ Camera	1	$50.00	$50.00
Total Budget:				$627.46

Table 38: BOM

7.3 Milestones

The implementation, design, and build of TrackPack will extend through two semesters, Spring 2023, and Summer 2023. Primarily, the first semester i.e., Senior Design I will focus on research and documentation to further support the team in the following semester. In the second semester i.e., Senior Design II, the team will begin executing the design and creation of TrackPack.

Milestone	Date	Members

SENIOR DESIGN 1		
Project Selection	1/17/2023 – 1/25/2023	Group 6
Divide and Conquer Report	1/26/2023 - 2/3/2023	Group 6
Divide and Conquer Revised Report	2/6/2023 -2/17/2023	Group 6
Research camera module	2/18/2023	George Gruse
Design lens array	2/20/2023	George Gruse
Research OBD II Integration	2/27/2023	Anjali Jodharam
PCB layout	2/28/2023	Myles Musanti
Test OBD II software	3/7/2023	Myles Musanti
Research Accelerometer Integration	3/15/2023	Kevin Singh
Order optical components	3/31/2023	George Gruse
60-page Draft	2/13/2023 - 3/24/2023	Group 6
Research GPS Integration	3/30/2023	Anjali Jodharam
60 page Revised & Upload to Website	3/27/2023 - 4/7/2023	Group 6
Optics Demo	4/11/2023	George Gruse
Research Gyroscope Integration	4/14/2023	Kevin Singh
PCB/electrical schematic	4/15/2023	Myles Musanti
Upload 3min Demo to Website	4/17/2023	Group 6
Upload 120-page draft to website	4/17/2023	Group 6
Final 120-page Report	4/8/2023 - 4/25/2023	Group 6
Final Report Submitted and Uploaded to Website	4/25/2023	Group 6
SENIOR DESIGN 2		
Order Parts	4/16/2023	Group 6
Build Prototype	5/2023	Group 6
Testing and Redesign	6/2023	Group 6
Finalize Prototype	7/2023	Group 6
Final Report/Presentation	7/19/2023	Group 6

Table 39: Senior Design I & II Tasks

7.4 Work Distributions

Ensuring the successful completion of this project within the allocated time frame, team organization plays an important role. Our group consists of diverse majors from electrical, computer, and optical backgrounds. We have devised a plan where each member of the team is assigned to respective tasks that align with their individual areas of expertise. We have structured this project in this way so that the team can maximize its efficiency and productivity and avoid potential bottlenecks or delays that may arise from task overlap or lack of specialization.

The team member with an electrical engineering background will be responsible for overseeing tasks related to schematic design and the integration of electrical components. This includes tasks ranging from designing and laying out electrical circuits to selecting the appropriate components for the project.

The team members with a focus in computer engineering will be assigned to software-based tasks. This includes coding and developing software that will be used to manage and control the electrical components, as well as troubleshooting and fixing any problems that arise.

The team member with a specialty in optic and photonic engineering will be responsible for designing and developing optical and photonic devices. These devices and systems include lasers, optical fibers, sensors, etc.

The following tables represent each task to be completed for this project and the team member assigned to them for Senior Design I.

Senior Design I Work Distribution	
Task	Team Member
Display Researching	Kevin
Storage Researching	Anjali
Power Supply Researching	Myles
PCB Researching	Myles
CMOS Sensor Researching	George
MCU Researching	Myles
Power Schematics/Testing/ Integration	Myles

OBD II Interface Researching	Anjali
GPS Researching	Kevin
Accelerometer Researching	Anjali
Design Lens Array Researching	George
Gyroscope Researching	Kevin
Assemble Lens and Adjust Tolerances	George
Order Relevant Design Parts	All Members

Table 40 : SDI Member Tasks

The following tables represent each task to be completed for this project and the team member(s) assigned to them for Senior Design II.

Senior Design II Work Distribution	
Task	Team Member
Assemble Camera	George
Software: OBD II Connectivity	Anjali
Soldering/Connections	All Members
Software: MSP430	Kevin & Anjali
Software: Graphic User Interface	Kevin & Anjali
PCB Fabrication/ Implementation	Myles
Power Schematics/Testing/ Integration	Myles

Table 41: SDII Member Tasks

8 Conclusion

TrackPack was initially conceived as a simple solution to meet the needs of car enthusiasts worldwide. However, as the research and development process unfolded, we realized that there were several engineering challenges that we needed to overcome to make the device both sophisticated and user-friendly.

We compared many relevant products on the market so that we could form a strong analysis on the favorable features and functionalities these devices had. We also examined what features we could add that relevant products were missing that could enhance the overall quality of TrackPack above fellow competitors.

We conducted thorough research on the hardware we wanted to implement in our final design. Thorough focus was dedicated to the following features and hardware. We analyzed the capabilities of four different single board computers and found that the Raspberry Pi 4 Model B was proven to have the best cost to processing power ratio. When it came to determining the best sensors, we chose cost and compatibility as our primary factors in consideration. We considered that consumers desire a user-friendly interface, so we ended up choosing the Raspberry Pi Touch Display since it fit all the criteria of being cost effective, compatible, and user friendly.

One of the challenges we encountered was guaranteeing cross compatibility among a wide range of vehicles. We also wanted to provide an ease of use to our consumers so that TrackPack could be easily installed and configured by users without requiring specialized technical knowledge. This required extensive research and optimization of the device's hardware and software to ensure that it could interface seamlessly with various vehicle models and provide accurate data. We diverted some of our focus on the design and aesthetics of the TrackPack device to provide a tasteful appearance and functionality of TrackPack. This required a balance of form and function, with a sleek and modern design that would be visually appealing to users while also providing the necessary features and capabilities.

After thoughtfully selecting the various hardware components for this design, we recognized the importance of project standards and constraints. The project standards for our hardware and software design influences the framework of the project development. The constraints imposed on TrackPack directly impacts whether the project is completed on time and if it is within the estimated budget. Considering this, we established clear workflow and goals for the team, as well as identifying any constraints that might impact the project's progress.

Upon the completion of parts selection, we began incorporating the preliminary hardware block schematics into an overall layout. The schematics assisted us in visualizing how the different components would be arranged and connected, which served as a guide for creating the final printed circuit board design. To ensure that the final product would function as intended, we established guidelines for testing the power circuit and sensors, as well as the firmware and software. This involved developing a comprehensive testing plan that would allow us to identify any potential issues or errors in the hardware or software before the final product was released. Programming and testing of the different components are assigned to the teammate with the respective specialty. Each teammate is responsible for ensuring the component they tested functions correctly. As the build of TrackPack progressed, we all collaborated for the overall project integration, bringing all the components together to ensure that they worked seamlessly.

Through testing our device and software as well as the creation of our three-minute demonstration video, we got a better understanding of how our hardware and software functions and got a head start on realizing the final design with the actual or final parts arrive in time for SDII. Any issues we may have debugging now have a frame of reference for what works to produce the desired results and we should be able to pass and problems we have in the building of our design easier.

TrackPack is a complex project that requires the coordination of many different components and processes. Our team consists of individuals with a diverse set of experiences and skills which has been helpful thus far with project management. It has been favorable that our team with different areas of expertise is able to assist with dividing different tasks and responsibilities accordingly. We each assumed different responsibilities according to our degree background and our individual skillset and this made the division of labor much easier which helped with reducing the amount of time needed for project management. If any team member ran into a problem or challenge, we could turn to our peers for support and advice. We also learned how to handle the task of working on such an intense project while dealing with other courses as well. Having great communication about what was required of each member and when it was needed and who was available greatly impacted our ability to create the work we have put forth. This not only helped to ensure that this research part of our project was completed on time and to a high standard, but also fostered a sense of collaboration and teamwork within the group.

This research paper has helped set a guideline and a solid framework for the development of TrackPack. In addition to sourcing the hardware needed for this design, we have also completed majority of the design of TrackPack. Tasks such as integration and software and hardware design have now been laid out through this research portion of this project. After this semester of extensive research and planning, this has set us up for success in the following semester. At the beginning of Senior Design II, we are now able to begin building TrackPack.

Appendix

[1] https://developer.nvidia.com/blog/jetson-nano-ai-computing/

[2] https://tinker-board.asus.com/product/tinker-board-s-r2.html

[3] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[4] https://www.raspberrypi.com/documentation/

[5] https://www.hardkernel.com/shop/odroid-n2-with-4gbyte-ram-2/

[6] https://all3dp.com/2/overclock-raspberry-pi-4-tutorial/#:~:text=For%20the%20Pi%204%2C%20the,containing%20the%20Raspberry%20Pi%20itself.

[7] https://en.wikipedia.org/wiki/Asus_Tinker_Board

[8] https://www.techpowerup.com/gpu-specs/jetson-nano.c3643

[9] https://www.intel.com/content/www/us/en/gaming/resources/cpu-clock-speed.html

[10] http://www.pidramble.com/wiki/benchmarks/power-consumption

[11] https://www.telerex-europe.com/content/files/pdfs/productPdfs/ASUS/90ME0031-M0EAY0.pdf

[12] https://www.u-blox.com/en/product/cam-m8-series

[13] https://www.sparkfun.com/GPS_Guide#:~:text=The%20update%20rate%20of%20a,significantly%20in%20the%20past%20second.

[14] https://en.wikipedia.org/wiki/Circular_error_probable

[15] https://www.everythingrf.com/community/understanding-gnss-sensitivity#:~:text=GNSS%20Sensitivity%20is%20defined%20as,to%20get%20a%20position%20fix.

[16] https://www.u-blox.com/en/product/zed-f9p-module

[17] https://www.canakit.com/raspberry-pi-lcd-display-touchscreen.html?cid=usd&src=raspberrypi

[18] https://learn.sparkfun.com/tutorials/gyroscope/all

[19] https://www.omega.com/en-us/resources/accelerometers

[20] https://www.althensensors.com/althenpedia/mems-sensors-what-are-they-and-how-do-they-work/#:~:text=MEMS%2C%20or%20Micro%20Electro%2DMechanical,as%20a%20change%20in%20capacitance.

[21] https://www.roadtovr.com/introduction-positional-tracking-degrees-freedom-dof/

[22] https://www.raspberrypi.com/software/

[23] https://grabcad.com/library/raspberry-pi-touchscreen-7-1

[24] https://grabcad.com/library/raspberry-pi-4-model-b-5

[25] https://grabcad.com/library/raspberry-pi-display-7-1

[26] https://rammount.com/pages/components

[27] https://www.raspberrypi.com/documentation/accessories/camera.html

[28] https://github.com/chethenry/pyOBD

[29] https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/

[30] https://www.lensrentals.com/blog/2011/03/the-development-of-wide-angle-lenses/

[31] http://phillipkpoon.net/jekyll/update/2017/02/27/Designing-A-Cooke-Triplet.html