# CSAT - Car Safety Assessment Tool

Ryan Abuasi, Vince Seeraj, Akihiro Yamamoto

**Abstract**—If people are different, then so are drivers. Accidents occur every year due to poor driving which negatively impacts both the driver and the potential victims of the accident. Despite there being many laws to ensure vehicle use safety, deaths caused by automobiles continue to be a problem throughout the world. Authorities and other experts do all that they can but there is only so much an outside entity can do to influence the safety of a driver. This is where our proposed solution for this problem comes in. The "Car Safety Assessment Tool" (or CSAT) is a device which allows one to track certain parameters of their driving such as speed, location, and how hard they hit the brakes. In this way, drivers can track down their weaknesses and problems as it pertains to driving. Furthermore, the CSAT can be used to collect data and obtain information about how safe or dangerous a driver can be behind the wheel. Different sensors and parts from the Raspberry Pi 3 B+ and the Adeept Ultimate Starter Learning Kit are used to create this device. Depending on whether the driver is speeding or quickly turning or changing speeds, the CSAT will gather data based on the actions performed by the driver. Collecting this information will better help drivers understand their problems and potentially become better drivers. The CSAT aims to let drivers know how safe they are as a driver based on an algorithm that is based off the data that has been collected.

**Index Terms**—Raspberry Pi, Accelerometer Sensor, Gyroscope Sensor, CSAT, API

## 1 Introduction

The **C**ar **S**aftey **A**ssesment **T**ool or CSAT is our team's approach to demonstrate the use of a cyber-physical system in a real world scenario. The CSAT is aimed to helped people learn about their driving style so they can improve themselves. The CSAT will utilize physical sensors to gather data about the vehicle it is in such as the speed, direction, and acceleration at which the car is moving. This information will then be sent to a Raspberry Pi which will be executing code allowing it to communicate with the sensors [1]. With the information received from the sensors and background code (this report will go into further detail about the background code in a later section) the CSAT will be able to determine how safely the driving is driving a vehicle.

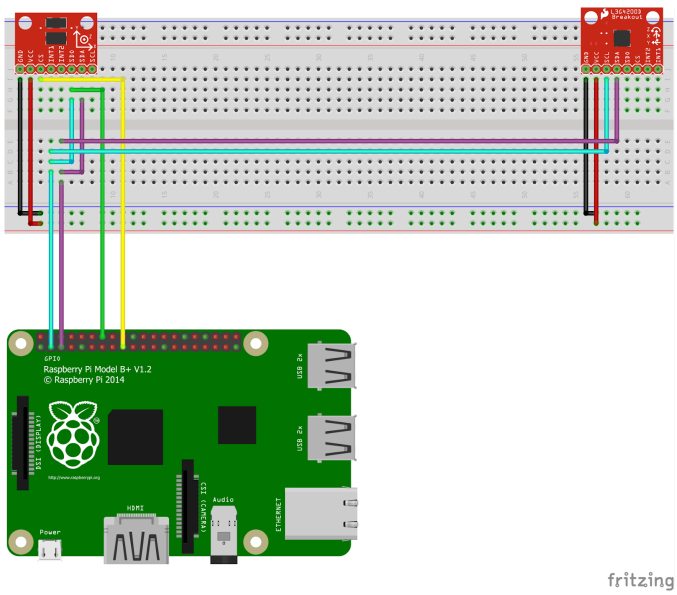


Figure 1. Raspberry Pi with both sensors

### 1.1 Project Proposal

Not everyone has the same amount of experience when it comes to driving automobiles and that leads to some drivers lacking in expertise of certain driving techniques such as turning and utilizing the brakes. Some drivers accelerate and break too fast without even knowing what wear they're causing to their vehicle and potentially to bystanders and other drivers on the road. Other drivers unknowingly cause major damage to their vehicle due to their driving style [2]. Our team wants to create a device that

PP-2017 V2.1

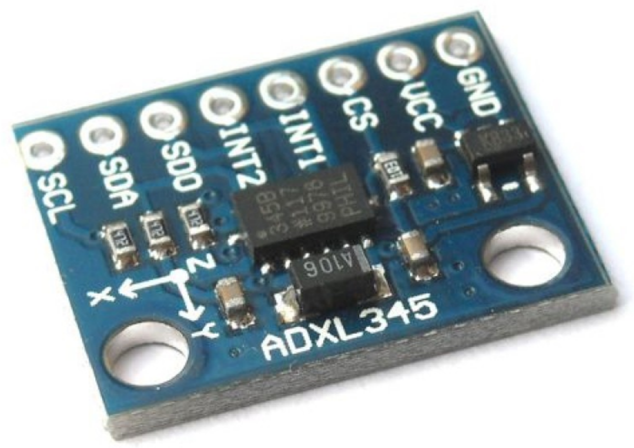


Figure 2. Accelerometer Sensor

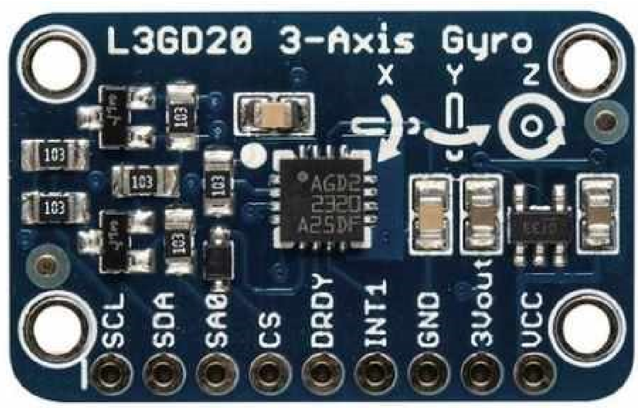


Figure 3. Gyroscope Sensor

can assist drivers on the way they drive. Using a Raspberry Pi, an accelerometer sensor, and a gyroscope sensor, we want to create a device for this purpose. The two sensors will be connected to the Raspberry Pi and the device itself will be powered by the vehicle once it has been ignited. From here, this is where the Google API software takes into affect and enacts its tasks upon the Raspberry Pi. This is how data and information regarding the parameters of the driver will be collected. The Raspberry Pi will call information from the Sensors, then input the data into an algorithm that will then let the user know about their driving style. The intentions of the device is for users to use the information obtained from the device to help them drive better and safer. Thus creating a safer environment and create less of an impact on their vehicle.

## 2 Equipment Used

For the CSAT project our team decided to go with a minimal amount of parts. This is mostly because our team aims to have a ”bare-bones” safety detection system with no extra features. This is to ensure that the team will have a sufficient amount of time to discover how to process the signals received from each component. The equipment we decided to use are:

- Accelerometer: ADXL345
- Gyroscope: L3GD20H
- Google Maps API
- Raspberry Pi 3 B+

We chose to use these items as each of them are inexpensive and friendly to unfamiliar users who intend to use it for ”do it yourself” projects such as our team. The ADXL345 and L3GD20H were both chosen as the peripherals of choice since they both can communicate with the Raspberry Pi via I2C which is crucial to our project.

### 2.1 I2C Implementation

To allow communication between the ADXL345 and L3GD20H our team will be using the I2C interface. I2C was selected as the I2C interface is a more viable option compared to SPI. I2C allows better communication between multiple peripherals as well as the ability to have two bidirectional open-drain lines which allows us to share the Serial Data Line (SDL) and Serial Clock Line (SCL) between both peripherals.

## 3 Model

Using the I2C interface standard, the Raspberry Pi interacts with both sensors, the ADXL345 accelerometer and the L3GD20 gyroscope, to read in values of how fast the vehicle is driving and what position the vehicle is in. The entire

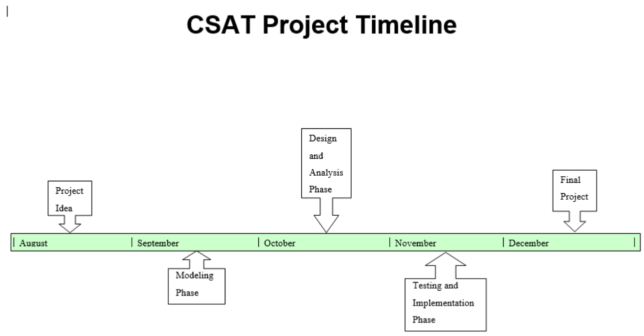


Figure 4. Process intended for the CSAT

system will be powered by the automotive power socket in which it will be placed on the dashboard of the targeted vehicle. Unfortunately, as will be mentioned later on, we were only able to successfully implement the accelerometer to calculate the acceleration and store the appropriate data values.

## 4 Design

As originally intended, the CSAT was supposed to implement the Raspberry Pi with the ADXL345 accelerometer and the L3GD20 gyroscope. The ADXL345 accelerometer along with the L3GD20 gyroscope will be interfaced with the Raspberry Pi using the I2C protocol. I2C was selected as the preferred protocol since it effectively allows communication between a master and multiple slaves.

The ADXL345 accelerometer was chosen to detect the vehicles speed and record the data. The L3GD20 gyroscope was chosen to detect the positioning of the vehicle. Both these peripherals together with the Google Maps API would in theory allow us to detect how fast the vehicle is moving, where the vehicle is located, and lastly use Google Maps API to determine if the driver is driving safely or not depending on how fast the vehicle is travelling.

## 5 Results

Due to the issues and complexity we encountered throughout the project, compromises were made. Since we were unable to implement the gyroscope it was cut from the model. This resulted in the model only containing the single

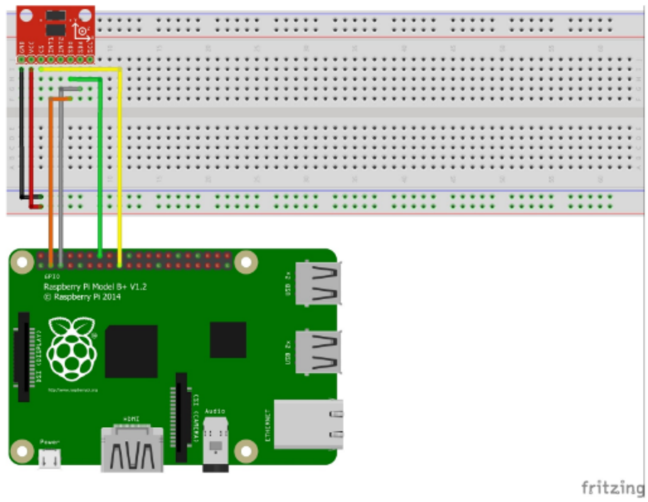


Figure 5. Final Raspberry Pi System

ADXL345 accelerometer, the new model can be seen in Fig. 4.

In this model the ADXL345 accelerometer was fully implemented with the highest resolution capable. The ADXL345 accelerometer successfully recorded values in $m/s^2$ and then converted the value to $mph$. Using these recorded values the CSAT could tell the driver to slow down depending on what the speed limit of the road is. During testing however, the CSAT was not very accurate. The output of the device was not matching the speed the speedometer of the car was outputting. Our team was unable to narrow down the cause of this issue, however we did narrow down a list to possible failure points:

1) The ADXL345 is unable to read speed properly while in a moving vehicle
2) The code used is not correctly recording data
3) The code used has incorrect values for resolution and range
4) The RaspberryPi is not provided enough power from the car outlet

Although the CSAT was not fully functional it was able to:

1) Convert the values it received
2) Store the values received for later use
3) Use values received for meaningful purposes such as informing the driver if they were driving too fast

## 6 Analysis

The CSAT project was not fully implemented however, we were able to achieve results with the accelerometer. Our team was unsuccessful with the project due to issues with the L3GD20 gyroscope along with Google Maps API. The team was unable to receive meaningful values from the L3GD20, this was mostly due to inadequate experience with digital signal processing. The team was also unsuccessful with the implementation of Google Maps API. A combination of both these faults lead to an incomplete project.

## 7 Issues

Upon working on CSAT, our team encountered a number of issues. Some problems we encountered were with the Google API and the Gyroscope sensor. Our team had difficulty implementing the Google API to our C++ code.

On the contrary Google API is available only on the the coding languages Java, JavaScript, Ruby, .NET, Objective-C, PHP and Python [3]. The Google API can not be implemented into the C++ code. created an issue because we intended on using the Google API with the Gyroscope to determine the location and position of the vehicle.

Our team also experienced issues with the Gyroscope. Although we were able to implement and run the L3GD20H sensor, we obtained no usable results relevant to our project.

### 7.1 Google Maps API

Implementation of the Google Maps API proved to be a particular challenge for our team. Such roadblocks included being well-oriented with how the API works itself which we have no prior experience with.

Furthermore, the language we intended to program the CSAT with, C++, was not available for the API. Another problem found with the Google Maps API dealt with financial costs including a credit card verification which was even limited itself due to the time limit constraints.

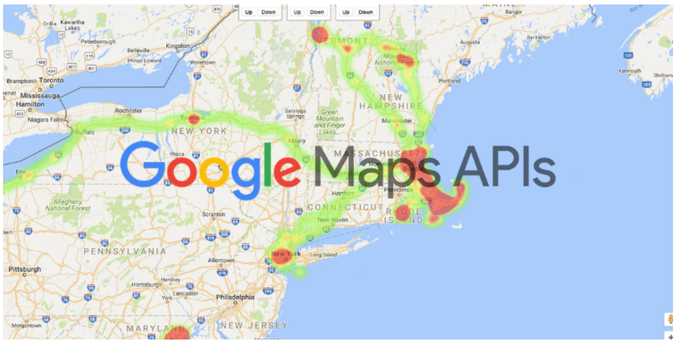


Figure 6. Google Maps API

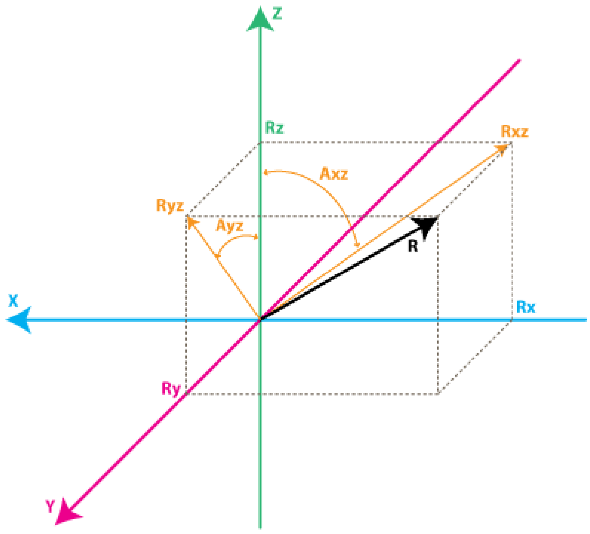


Figure 7. Tutorial of gyroscope usage

### 7.2 Gyroscope

While the program to run the gyroscope was already created, the main issue came in the form of the experimental phase in order to obtain the results. Although the L3GD20 gyroscope would give us data, our team was not able to process the data into meaningful data that we could utilize. The original intention was to use the gyroscope in conjunction with the Google Maps API. Without meaningful data from the gyroscope the team was unable to do this. The API was meant to act like a "glue" for the accelerometer and the gyroscope to work together and determine whether or not the driver is driving safely [4].

## 8 Potential Solution

In finding a method to solve our issues, we started looking for alternatives which led us

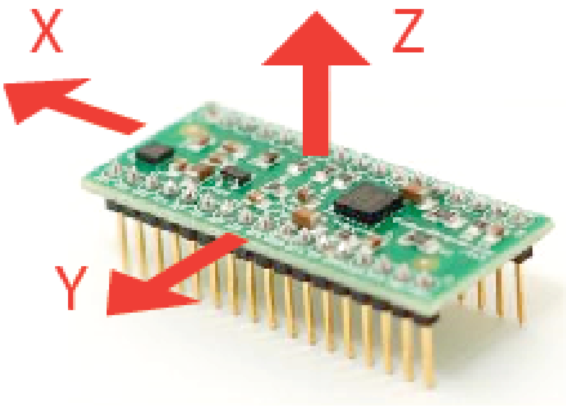


Figure 8. How an IMU would be implemented

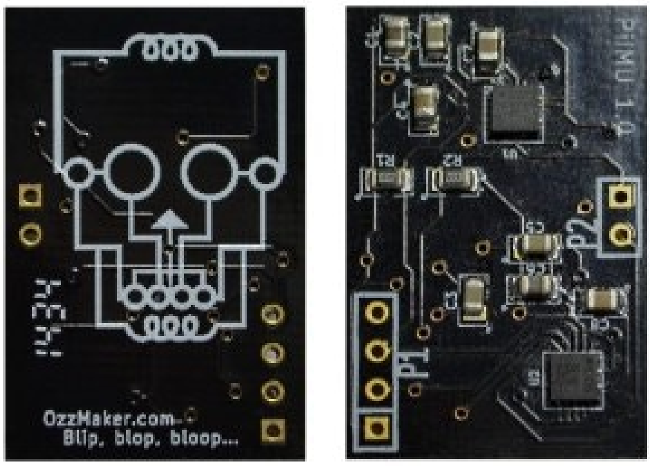


Figure 9. BerryIMU

to the BerryIMU. The BerryIMU is an inertial measurement unit that combines both the gyroscope and the accelerometer functions into one part. In addition, the BerryIMU contains a magnetometer as well as capabilities to measure altitude, temperature, and pressure.

Naturally, we were primarily interested in the accelerometer and gyroscope functions for the CSAT implementation. However, we ended up looking for alternatives too late as we started to look near the end of our intended completion deadline. The BerryIMU is a feasible solution alternative that could have been used. [5].

## 9 Individual Contributions

Throughout the project the team divided the work equally throughout the members.

Ryan Abuasi:
- Project Proposal
- Final Paper
- Presentation
- Google API research
- Fritzing Diagrams
- Programming

Vince Seeraj:
- Project Proposal
- Presentation
- Final Paper
- Wiring the circuit
- Gyroscope research
- Alternative solution

Akihiro Yamamoto:
- Project Proposal
- Presentation
- Final Paper
- Accelerometer research
- Data processing/recording
- IMU research

## 10 Conclusion

The CSAT was built with the intention to assist people to learn how to drive safer. It uses I2C connectivity to read the data received by two sensors, the Accelerometer and the Gyroscope sensor. The Raspberry Pi B+ was then supposed to use that information to give feedback to the user. Although a fraction of CSAT has been completed, there is still a good amount of work that needs to be done.

## References

[1] "Speed limits | Roads API." [Online]. Available: https://developers.google.com/maps/documentation/roads/speed-limits

[2] D. Chen, K.-T. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible sensing of vehicle steering with smartphones," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '15. New York, NY, USA: ACM, 2015, pp. 1–13. [Online]. Available: http://doi.acm.org/10.1145/2742647.2742659

[3] "What programming languages does the api support?" [Online]. Available: https://support.google.com/code/answer/54878?hl=en

[4] G. GangsterFollow, "Accelerometer & Gyro Tutorial." [Online]. Available: https://www.instructables.com/id/Accelerometer-Gyro-Tutorial/

[5] "BerryIMU v2 - An accelerometer, gyroscope, magnetometer and barometric/altitude sensor," Jan. 2015.