SenzCar

Technical Report

By: Kogul Balasubramaniam, Kyele Haynes, Samuel Dadet

# Declaration of Joint Authorship

We hereby declare that this technical report contains material that is the result of joint collaboration on the project. All of the team members have done their own research on the hardware and software components utilized in the project. As per the individual sensors used in the project (defined under requirements specifications), each team member has done their own extensive research on the sensor they have been assigned as well as performed data collection, analysis and interpretation on said sensor. Any key ideas involving each individual team members sensor has been written and drafted by themselves with critical revision from both the other team members. Research, primary contributions and experimental designs in regard to the PCB, physical wiring, enclosures, Raspberry Pi and Raspbian operating system will all be conducted, led, written and drafted by Kyele while taking into considerations any suggestions from other team members. The same outlines go for Sam and Kogul in regard to the android development and data linking portion of the project. As per the design and documentation of the database, all team members will help write and draft these sections. We declare that everything written in this technical report is the product of our own ideas and research. Any information that has been used or referenced from another source have been cited under the references section of the report. Finally, we, to the best of our knowledge claim that this report does not defy any copyright laws.

# Approved Proposal

***Proposal for the development of SenzerCar***

Prepared by Kogul Balasubramaniam, Kyele Haynes, Samuel Dadet  
*Computer Engineering Technology Students*https://github.com/KogulB/CENG355Project

**Executive Summary**

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with the following sensors and actuators YL-40 PCF8591 (0x48), AMG8833 IR Thermal Camera Breakout (0x69), BME280 Temp/Barometric/Humidity (0x77). The database will store Engine Temperature, Air Intake Pressure and Luminosity of the headlights. The mobile device functionality will include Being able to see basic maintenance information for the Baja racing vehicle. Checking the headlight luminosity coming from the car to check if it is reaching high levels. The app will respond by changing colors to let the vehicle owner know that something is wrong. The Air Intake Pressure will also be available to let the driver know if there is any concerns. The value can be seen in Hpa or PSI. Finally, The Engine Temperature reading will also be present to let the driver know if the engine is overheating or not. This value is also available to be seen in Celsius or Fahrenheit and will be further detailed in the mobile application proposal. I will be collaborating with the following company/department Baja Racing Vechicles. In the winter semester I plan to form a group with the following students, who are also building similar hardware this term and working on the mobile application with me Samuel Dadet, Kyele Haynes and Kogul Balasubramaniam. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

**Background**

The problem solved by this project is In today's society many automobiles come equipped with various sensors that provide the driver with basic maintenance information so he/she knows when to take it in for service. This informations helps the driver diagnose the problem and even possibly let them know before hand if there vehicle is in possible engine failure. Our projects focus is on providing a simple telemetry system for Baja racing vehicles. We provide simple maintenance information for the racer to diagnose their own vehicle. Information such as Air Intake Pressure, Engine Temperature and Headlight Luminosity.. A bit of background about this topic is Baja Racing Vehicles are raced in the SAE off-road motorsport competition every year by competing universities all over the world. The vehicles require less maintenance than your common automobile. However, some maintenance is still required. Our product is engineered towards helping the user distinguish if is vehicles is experience simple maintenance issues or is on the way to shutting down due to engine failure. Are three sensors used with a Rpi 3 powered with a step down converter connected to the vehicle provides crucial information that helps the driver diagnose there vehicle and provide simple maintenance..

Existing products on the market include [1]. I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content”[2] and have found and read [3] which provides insight into similar efforts.

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

* Java Docs from CENG 212 Programming Techniques In Java,
* Construction of circuits from CENG 215 Digital And Interfacing Systems,
* Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
* Micro computing from CENG 252 Embedded Systems,
* SQL from CENG 254 Database With Java,
* Web access of databases from CENG 256 Internet Scripting; and,
* Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

**Methodology**

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:  
 Phase 1 Hardware build.  
 Phase 2 System integration.  
 Phase 3 Demonstration to future employers.

*Phase 1 Hardware build*

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

*Phase 2 System integration*

The system integration will be completed in the fall term.

*Phase 3 Demonstration to future employers*

This project will showcase the knowledge and skills that I have learned to potential employers.

The brief description below provides rough effort and non-labor estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

We will be purchasing a Rpi 3 to use as our main circuit board and a Thermal Camera to detect heat signature. As well as a Light Sensor(PCF 8591) to detect if the headlight was left on or not as well as if its reaching high levels. A BME280 to detect the pressure of the intake in the vehicle.

**Concluding remarks**

This proposal presents a plan for providing an IoT solution for Our System uses the thermal sensor to detect the engine temperature and changes the background if engine is starting to overheat. The Light Sensor detects Headlight luminosity and then displays it as a value and changes application background depending on if the light is reaching high levels. Finally, a pressure sensor to detect the intake pressure.. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by [3]. I request approval of this project.

**References**

[1] Gilsdorf, J. M., Vercauteren, K. C., Hygnstrom, S. E., Walter, W. D., Boner, J. R., & Clements, G. M. (2008). An integrated vehicle-mounted telemetry system for VHF telemetry applications. Journal of Wildlife Management, 72(5), 1241-1246. Retrieved from http://ezproxy.humber.ca/login?url=https://search-proquest-com.ezproxy.humber.ca/docview/234179670?accountid=11530

[2] Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

[3] Leonardo Presoto de Oliveira, Marco Aurélio Wehrmeister, AndréSchneider de Oliveira, "Systematic Literature Review on Automotive Diagnostics", Computing Systems Engineering (SBESC) 2017 VII Brazilian Symposium on, pp. 1-8, 2017.

# Abstract

In today’s modern society off road cars have always been a popular method of racing on designated tracks. They provide a thrill to the driver and can be very entertaining when done safe. But since it is like any other mechanical machine it tends to develop problems. There are many problems that could develop throughout the lifespan of a car and our telemetry system is intended to detect certain issues or keep track of certain data that could be beneficial to the cars mechanics or just give any information that the owner might request. The driver would be able to make quick and basic repairs based on the information that our system would provide saving time, energy ando cost. Our system involves four crucial components (described in the requirement specifications). Each sensor will be programmed to collect certain crucial data for the vehicle and will record all of it into a database where it can be accessed by the driver whether it be past or present. The data can be viewed through our mobile application on any android device where the user can log in to get access to there vehicles data. It will also record the data in some type of graph so they can compare past and future information to see whether there was a decline in performance. Thanks to the simplicity of our casing (described in the report) our system will be able to be installed simply to any vehicle. Our design is being set out to make the consumers life a bit easier as well as help the driver know the condition of their vehicle no matter where they are.

# Table of Contents

[Declaration of Joint Authorship 2](#_Toc7111278)

[Approved Proposal 3](#_Toc7111279)

[Abstract 8](#_Toc7111280)

[Table of Contents 9](#_Toc7111281)

[Introduction 11](#_Toc7111282)

[Project Description 12](#_Toc7111283)

[Problem 12](#_Toc7111284)

[Solution 12](#_Toc7111285)

[Requirements Specifications 13](#_Toc7111286)

[Software 13](#_Toc7111287)

[Hardware 14](#_Toc7111288)

[Database 14](#_Toc7111289)

[Status Report 16](#_Toc7111290)

[Status Update #1 16](#_Toc7111291)

[Status Update #2 18](#_Toc7111292)

[Status Update #3 20](#_Toc7111293)

[Status Update #4 21](#_Toc7111294)

[Status Update #5 23](#_Toc7111295)

[Build Instructions 24](#_Toc7111296)

[Introduction 24](#_Toc7111297)

[System Diagrams 24](#_Toc7111298)

[Budget Plan/Bill 25](#_Toc7111299)

[The Git Repository 26](#_Toc7111300)

[Time Commitment 27](#_Toc7111301)

[Hardware Design/Assembly 28](#_Toc7111302)

[Power Up 30](#_Toc7111303)

[Production Testing 30](#_Toc7111304)

[Unit Testing 31](#_Toc7111305)

[Mass Production 33](#_Toc7111306)

[Android Application 34](#_Toc7111307)

[Conclusion 35](#_Toc7111308)

[Bibliography 36](#_Toc7111309)

# 

# Introduction

Baja Racing is a design competition hosted by the Society of Automotive Engineers (S.A.E). It's a tournament where various college and universities student come together to compete in a dirt road type of track, building their own vehicle to the specifications of the S.A.E board. Most of the students who participate in this competition are mostly automotive engineering students. Most students are focused on one path only and have very little knowledge of electronics and computer programming which would put them at a disadvantage if there is any issues that develop because of those disadvantages.

The objective of this telemetry system is to make life easier for the buggy racers in detecting fault in their vehicle, it would be a fair and just race if all buggy vehicles had our telemetry system in it. leveling the playing field for all participants. some of the unique approach about the telemetry system is that it can be used in smart vehicles and not just on buggy races even in other means of transportation like ships, airplanes and trains. In buggy races, a lot of drivers are stuck with mechanical, electrical, functional issues that they have no clue where or how to deal with. Some don’t even know what is wrong with their vehicle, it just won’t start. With our telemetry system, all those worries, and woes will not be present as you have exactly what is wrong with your vehicle right there on your app.

With our telemetry system, the driver would be able to detect exactly what is wrong with their vehicle. Our device would provide real time data which is being fed into the driver’s app and would notify the driver of any changes that should be done prior to start of race. Priority is given to the most crucial problem, indicating to the driver what is hindering the total usability of the vehicle. With the temperature sensor, if the engine is overheating, the BME sensor would send data to the phone app indicating to the driver that something has to be done immediately to ensure smooth sailing of the vehicle.

# Project Description

## Problem

In the world of SAE Baja Races a lot of participants are limited to very little knowledge of how their vehicle is running due to the students being in the realm of mechanical engineering and not electronics. A lot of problems most SAE Baja racers face are issues that could be solved with our telemetry system. These issues include; overheating engines, a lack of light in dark areas and low intake air pressure. Firstly, with simple SAE Baja vehicles there is no way to know if your engine is functioning at optimum capacity. Year after year participants use the same engine continuously some knowing its performance outside of the testing stages, which could cost racers the tournament. Next, dark areas are another possible issue in the SAE Baja racing world where racers would have to also worry about turning on and off their headlights. Lastly, another important metric of a running engine is the air-fuel ratio. Without any technology/sensors the pit teams and driver of the SAE Baja vehicles would not be able to easily locate and diagnose a fuel/air issue. Now with the modifications done to most SAE Baja vehicles without a sensor or device you won’t be able to accurately measure all these metrics. Most of the problems mentioned above concentrate mostly around the engine and that is because the engine serves as both the heart and brain of a SAE Baja vehicle and if these issues are not taken care of it could cause many issues for the driver and team of the vehicle.

## Solution

We came together to work on the problems discussed above. An AMG8833 sensor was programmed to take in a single value thermal reading. So, if the engine were to start to overheat, it detects this and sends the data to the database and then the app. The driver is then able to see this in real time on the Android application. This would help the driver adjust their driving style or allow them time to stop to avoid permeant damage to the engine. It could also help reduce the time it takes to find a probable issue with the engine. A PCF8599 was programmed to read the intensity of light in the racing environment in real time and as well send this data to the database and then the app. This could then be implemented with another system in the vehicle to automatically turn on the headlights when needed. Finally, there is the BME280. This sensor is a multi-functional sensor that reads altitude, temperature, humidity and barometric pressure. We made the most use out of the pressure sensor so it can get readings of the intake of the engine. The BME280 can detect if the engine has the right amount of air coming in which can then help determine the air-fuel ratio of the engine, a very important metric in determining how an engine is running and if there are any issues. All these sensors come together to make a simple telemetry system that provides just enough information for SAE Baja racers. All three sensors are programmed to display data in real time, helping drivers to identify any fault as quickly as possible and choose their course of action accordingly.

## Requirements Specifications

### Software

There are a few different types of software required for the project. The programming languages we will be using are Java (Android), Python and C. For the operating system on the Raspberry Pi we will be using Raspbian included with programming IDEs compatible with python and C and XRDP for remote access. All the software and connections on the Raspberry Pi will be setup by Kyele. The programming aspect of retrieving data from each of the sensors will be done as follows; Kyele – AMG8833, Kogul – PCF8591, Samuel – BME280. To develop the android application the most recent version of Android Studio is required as well as emulators running multiple different versions of android, Kit-Kat (4.4) or higher. Most of the development work for the application will be completed by Kogul and Samuel. Lastly, a firebase database that is capable of holding records every 5 minutes, 24/7 for up to 1 month is required which will be designed, created, and maintained as a joint effort.

### Hardware

The hardware includes a few main aspects which are; 3 different breakout board sensors, a Raspberry pi 3 (the development platform), smartphone running Android Kit-Kat (4.4) or higher and a 3D printed enclosure. Some other hardware requirements are an 8GB or higher micro SD card, a micro USB power cable for use with Raspberry Pi 3, PCB materials for connecting the sensors to the development platform and wall mounting brackets. As per the hardware work breakdown, each member is responsible for their own sensor; Kyele – AMG8833, Kogul – PCF8591, Samuel – BME280. They each have to completely understand how it functions and how it interacts with the development platform. This includes how it sends data, how it is powered and maximum and minimum specifications it can handle. As per the rest of the hardware, Kyele is responsible for designing the PCB required for connecting the sensors to the development platform with the help Sam and Kogul with their knowledge of the other sensors. He is also responsible for maintaining the Raspbian OS installed on the Raspberry Pi 3, such as installing any newly introduced software and performing updates and security patches. Each team member will have complete access to the Raspbian Operating System as we will all need to work on the software connections to the sensors. Lastly, all team members will work together on designing the final 3D printed enclosure.

### Database

The database was designed and implemented by Kogul. Each sensor is connected to the database through 3 separate Python scripts (one for each sensor) running on the Raspberry Pi. Currently through the prototype system we have implemented a Bash script to run the Python files back to back and repeat run time every 10 seconds. The database values currently hold data values that were originally updated every 10 seconds when Pi was running and all the user registration info of everyone registered with the one device. Each user has to have a special registration code unique to their system (“currently” hard coded as 12345) to be able to register; after successful registration the users name and email address is recorded to the database with a unique hashed ID. This user authentication is also used as login system to view all the data being presented through our android application. Each user that registers can login to the app and view all the latest readings coming from the database. Currently our system holds three sensor readings (described in the build instructions) . In the future we are hoping to update this system with more sensors to give the user more maintenance information about there Baja racing vehicle.

# Status Report

## Status Update #1

Our progress so far in the project has been very successful due to some preparations done in the previous semester.

**Hardware**

We will be using Kyele’s Raspberry Pi as our development platform; it is up to date and functioning as intended. His Raspberry Pi is communicating with the AMG8833 with no error. Sam and I also have our sensors (BME280 and PCF8591 respectively) communicating with our own Raspberry Pi’s. Our plan is to eventually switch our sensors over to Kyele’s platform to have all sensors working simultaneously with our application.

**Physical Format**

We still must connect our two sensors in conjunction with Kyele’s sensors on the Raspberry PI. This means we still must create and design a PCB that will support all three sensors.No progress has been made on the enclosure as the PCB needs to be created before we do this due to dimension issues.

**Current Progress**

So far this semester we have created the skeleton for the technical report, we have also planned out what each group member is going to contribute towards the project. Further planning is required to get started on the creation and maintenance of our database.

**Problems**

No problems have been discovered yet since we have not started implementing each of the sensors together. Problems that we might have in the future of our project is on the software side where we would have to collect data from our sensors and eventually be able to transfer them to our database. Since all three sensors are programmed using either C or Python it will prove to be a challenge to transfer data to our application which is primarily programmed using Java (Android Programming). Another challenge we might have to deal with is each of our sensors communicating.

**Financials**

Most of the hardware that is required had been purchased last semester. Some things that are still required is the materials for the enclosure and PCB. Although a new updated financial plan has not been created as of yet, we expect to draft one very soon with any updated purchases. Purchases are subject to change throughout the semester and the financial plan will be updated accordingly.

Please feel free to contact me or any of my colleagues through email if you have any further questions.

## Status Update #2

**Current Progress**

We recently changed our project to a telemetry system for a Mini Baja racing vehicle from our original idea, the “SenzRoom” room monitoring/sensing device. Even though this seems like a big change, the transition has not been very difficult. Most of the code we already have will work for the newly designed app. As per the PCB, I had to come up with a different design, but I had not yet had the original design printed.

**Hardware**

Since the project change, I have designed a PCB to plug into the GPIO pins on the Raspberry Pi which allows for wires to be plugged into it from the sensors spread around the vehicle. So far, I have had this board printed in the prototype lab. I have also designed the 3 small boards for each of the sensors but have yet to get them printed as well. I have not yet added the power management circuitry to power the Pi from the vehicle’s 12V battery but I may have found a relatively cheap step-down converter on Adafruit to do this for me. Lastly, the new case still needs to be designed.

**Problems**

One problem that Kogul encountered was that we had to change a big part of the android application since we do not need to intake multiple sets of data anymore. This was solved by creating a new app and importing the code from the old app with a new activity structure. Another problem that has been noted but not addressed yet is the casing for the Raspberry Pi. Currently we are using a plastic, laser cut case which would not stand in a rough, Baja environment. Thus, I must learn how to design, and 3D print a rugged case for the Raspberry Pi and each of the sensors.

**Financials**

So far financially, two more hardware components have been noted but not yet purchased. These are, the 12v to 5v step down converter and wiring from the Raspberry Pi to each of the 3 sensors.

## Status Update #3

**Current Progress**

We are on track in relation to out project schedule. The hardware aspect of this project which was pioneered by Kyele has seen enormous progress since the last report. the PCB has been printed out and most of the soldering has been done. The database aspect which is being handled currently by Kogul is being developed and some sensor values have been passed to the Realtime Database (Firebase). Data can be written and stored to the data base in real time.

**Hardware**

Using a very detailed layout of our individual sensors, Kyele has designed and engineered a PCB to plug into the GPIO pins on the Raspberry Pi which allows for wires to be plugged into it the sensors. So far, the PCB has been printed out to meet the specification of the sensors. Just a little bit of soldering left to be done. Wiring has been completed but so far, we have it connected through a breadboard and not in an actual encasing.

**Problems**

One problem that we encountered was in the software development aspect, we are working on integrating all three python files from each individual sensor to work as one. Another problem that we have encountered is importing some of the data into our android application. The readings are only able to take one sensors data (PCF8591) and display it on the app work still needs to be done for the other two sensors to integrate to the app.

**Financials**

Financially, there is one hardware component we have yet to purchase. which is the 12v to 5v step down converter.

## Status Update #4

**Current Progress**

We are still on track as per our project schedule. All the required PCBs have been printed and test fitted and progress has begun on the enclosure of all the boards. Since the last update we have got all data from the sensors uploading to the database in real time from the Raspberry Pi and appearing on the Android application.

**Hardware**

All four required PCBs (main board, and 1 board for each sensor) have been printed. Currently the main board is being used to connect to the 3 sensors on a breadboard. I still need three 4 pin headers to solder onto the 3 sensor boards before I can use them. I still need to purchase proper weather rated wiring to make up the wiring harness and the step-down converter on Adafruit to power the board from the 12V battery of the Baja vehicle. These 2 things can be done last as they are not needed for testing. Lastly, I have begun production on the 3D printed case but still have a lot to learn about the software, Solidworks.

**Problems**

So far, most of our problems have derived from the android application and how it gets and displays the data from the database. Kogul and Sam have to restructure the app as there are quite a few bugs currently with how the data is passed through activities and tabs. The app is also somewhat confusing in its current state and does not contain a working login page so creating a new project and restructuring the whole thing is our best course of action. One other issue we have is getting the data in the proper format from the AMG8833 Thermal Sensor. This sensor is currently set to output its data in an 8x8 grid of numbers representing the heat signatures exactly as it picks them up, but for our situation, this much info is not required. To solve this, we have to average out all the data the sensor outputs and then send it to the database.

**Financials**

So far financially we still remain in the same position since the beginning. Two more hardware components still need to be purchased which are; the 12v to 5v step down converter and wiring for the wiring harness, but as stated earlier these are not required as of yet.

## Status Update #5

**Current Progress**

As regarding the project schedule, we are on track, the reprint for the case is in and minor fixes on the 3 PCB’s were done. Since the last update our database can read real live data from the raspberry and display on the Android application.

**Hardware**

The main board for the PCB has been printed out and like I stated earlier, there were minor fixes done to the sensor PCB’s, Kyele perforated mountable holes on the PCB so they could be screwed in and stay in place. weather rated wiring was purchased and also which would harness the stepdown converter on the adafruit to power the board from the 12V battery. The enclosure is yet to be concluded, we originally planned to use SolidWorks but it was more or less a challenge so we opted to use Corel Draw.

**Problems**

Most of the problems we encountered prior to the previous status update have all been fixed like displaying the data from the database. Also, minor bug fixes to be cleaned in the code by I and Bala.

**Financials**

No significant changes have occurred in this department. We only purchased the 12v to 5v step down converter and the wiring for the wiring harness**.**

# Build Instructions

## Introduction

The purpose of this project is to build a telemetry system for a Baja racing vehicle. There are three sensors involved in this project which include a PCF8591, BME280 and AMG8833. This is a collaborative project that involves up to 3 members who each contributed to at least one sensor and the overall dynamics of this project. We put together a [Budget Plan](https://github.com/KogulB/CENG355Project/blob/master/documentation/Budget.xlsx) and a project [schedule](https://github.com/KogulB/CENG355Project/blob/master/documentation/Schedule.mpp) to help us complete the project in a successful manner.

## System Diagrams

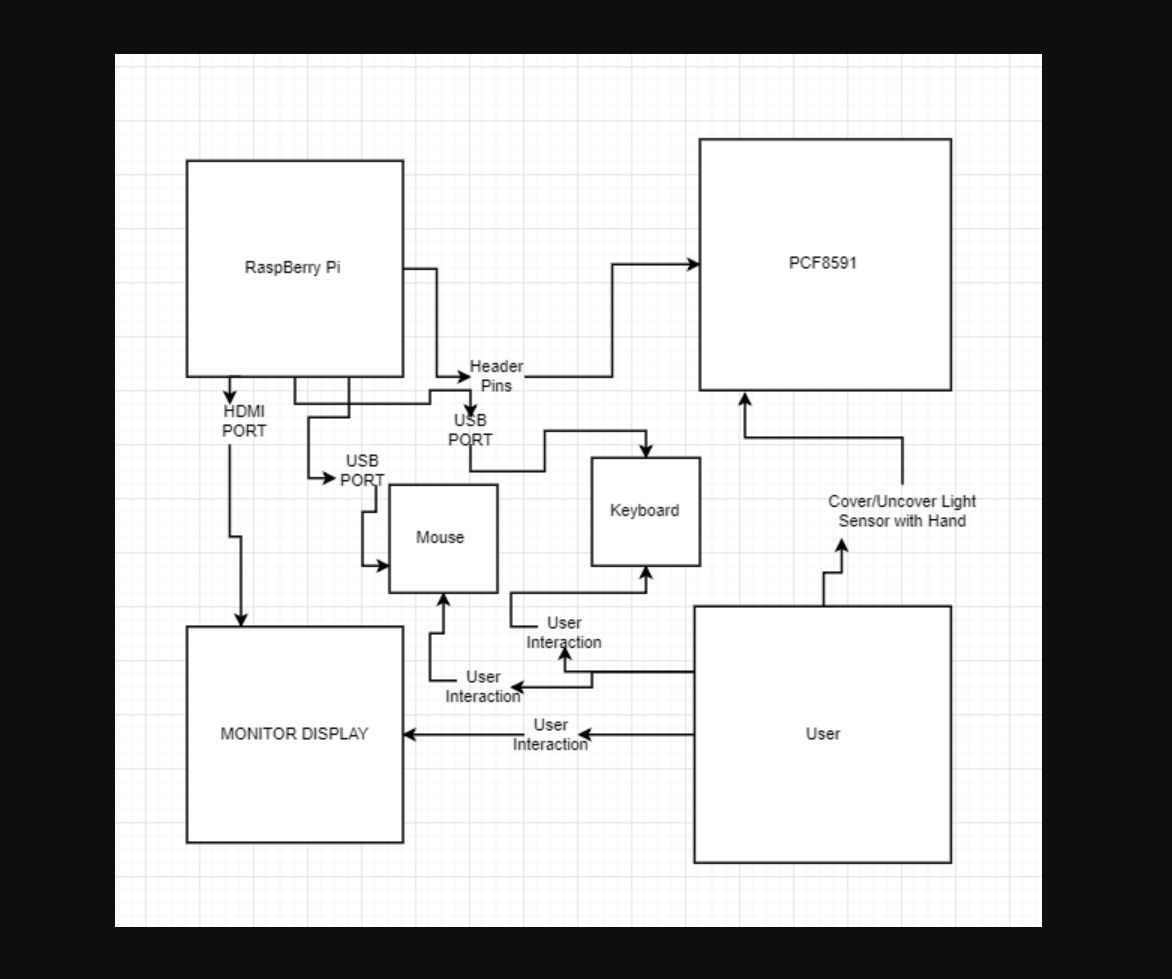


Figure 1 - System Diagram 1

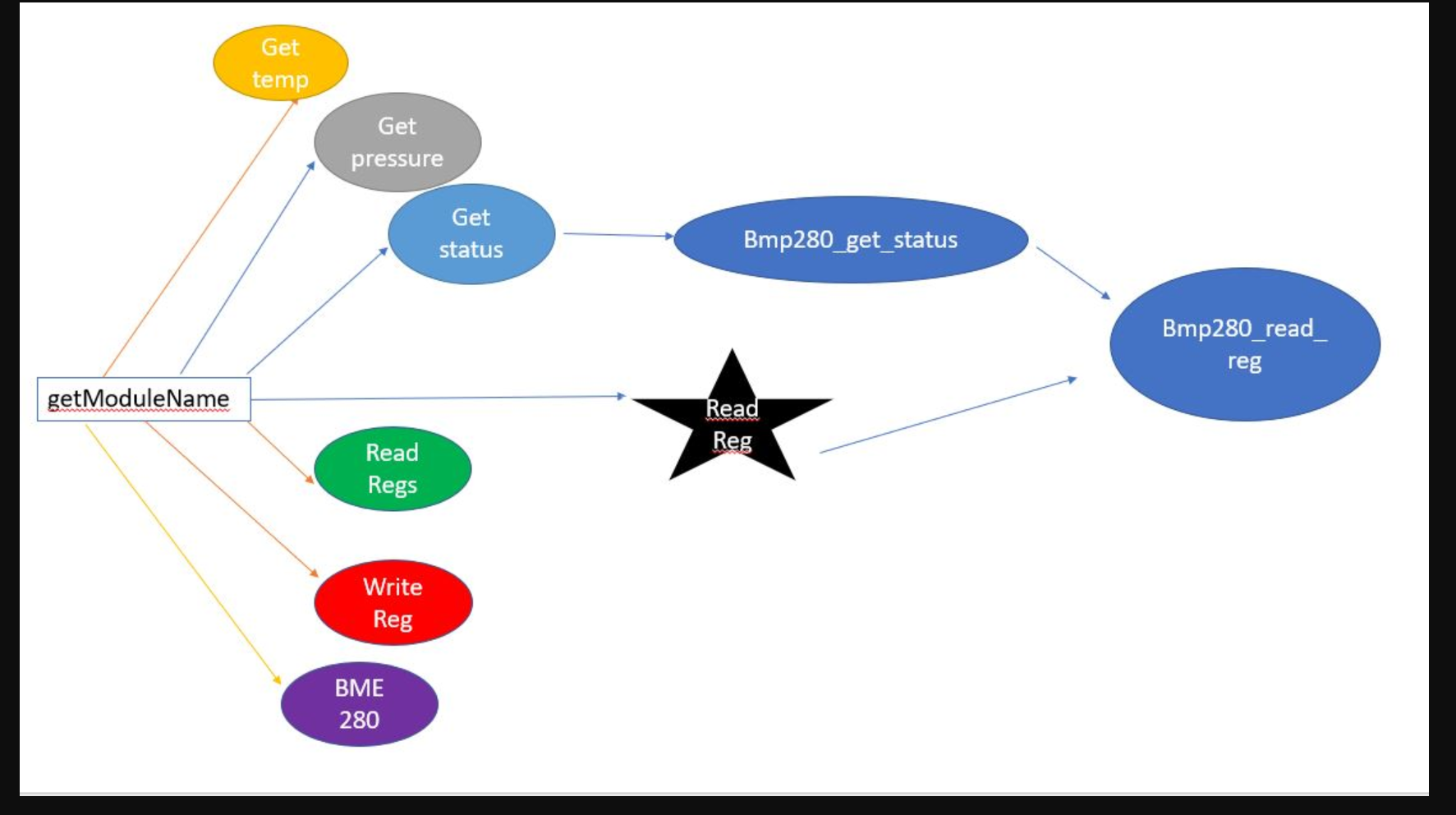


Figure 2 – System Diagram

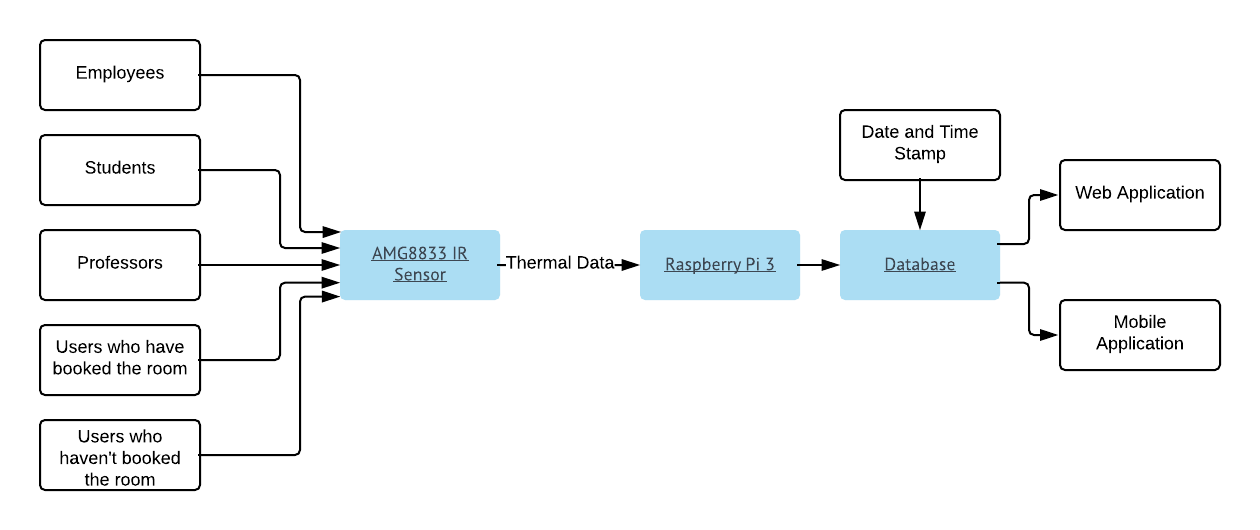


Figure 3 – System Diagram

## Budget Plan/Bill

The main components of the project were a Raspberry Pi 3 Kit which included a power cable and a default case. Memory card would have to be purchased separately but for our case we had one already. Other purchases made were the sensors and devices needed to run everything: PCF8591, BME280, AMG8833, Ethernet cable and adapter. An HDMI cable for initial setup and a breadboard for testing will also need to be purchased but in our case, we already had one of each. There are however some extra parts that were purchased for testing but were never user used. The link to the budget plan can be found [here](https://github.com/KogulB/CENG355Project/blob/master/documentation/Budget.xlsx).

## The Git Repository

The link to our publicly available Git repository is; <https://github.com/KogulB/CENG355Project>. This is where all files created and used have been stored over the course of the project. The repository is divided into 3 separate sections; Documentation, Hardware and Software.

First off, the Documentation directory includes any type of reporting done on the project as well as the proposal, budget and schedule. This allowed us to keep everything that was not directly associated with the creation of the physical project or the coding of the Android app.

The Hardware directory contains all files associated with the design creation of the four PCB’s and 4 acrylic cases used in the project as well as all pictures taken of the physical components. We went through four different versions of the main PCB that connects to the PI in Fritzing therefore there are 4 .fzz files as well as the extended Gerber files of the final version which were sent to the prototype lab for printing. For the three individual sensor boards we went through two different versions each. For casing, there are the Corel draw files for the main case and for the smaller cases (one file as they are all the same size).

Lastly, there is the Software directory which includes all files related to the Android application and the collection of data from the sensors which also reside on the Raspberry Pi. The files used on the Raspberry Pi include three python scripts to get data from each sensor and send it to the database as well as one bash script to run all the python scripts in succession with a delay.

## Time Commitment

It took about two hours per sensor for us to complete the research portion of our project because we each did our own research on a different sensor to find all the components and code needed to get the ICs up and running. So, in total research took about 6 hours (2 x 3 sensors).

The materials could take anywhere from 2-5 days to ship if they are ordered from Amazon and depending on the shipping method used. The sensors were however purchased from different websites (Links in the Budget) and took about 1 week to arrive.

After getting all the components listed in the budget, the next step is to test the connection from each sensor to the Raspberry Pi using a breadboard. Make sure to test each sensor separately to not short circuit the sensors. This should take roughly 2 hours.

After figuring out the connections, it took Kyele one hour creating the connections on Fritizing. He then spent approximately 2 more hours redesigning and moving around connections on the PCB by adding vias (holes in the PCB that can be soldered to allow the circuit to continue from the top of the board to the bottom or vice versa), making sure the connections don’t overlap and making the PCB overall more efficient. Luckily for anyone else attempting this project the Fritzing files are already created and are included in the Git repository. It also took another 2 hours to design the 4 different cases to fit the Raspberry Pi and the 3 separate sensors.

For physical assembly, the total time commitment was approximately 4 hours. This includes gathering all the header pins, sockets for each of the breakout board sensors and wiring. It also includes all the time taken to solder the vias, the header pins and sockets as well as fit and mount the PCB’s, sensors and Raspberry Pi into their respective cases.

Setting up the code for testing is not hard at all. For the PCF8591 sensor, code can be pulled from the StudentSenseHat Git repository. Since the PCF8591 was used in that project, testing took about 30 minutes. For the BME280 sensor, the test code was found online from the following [website](https://github.com/adafruit/Adafruit_CircuitPython_BME280/blob/master/examples/bme280_simpletest.py). The AMG8833 test code was also found online at this [link](https://github.com/adafruit/Adafruit_CircuitPython_AMG88xx/blob/master/examples/amg88xx_simpletest.py).

The Android application was also less time than estimated due to us having a “skeleton” version of the code from our Software Project Last year. Designing the pages of the app took approximately about 5 hours due to different inputs from each member of the group. The firebase database was connected to the app once designing was done. Figuring out user authentication took about 3 hours per 2 weeks (6 hours) due to the concept being a little tricky to implement. Overall the app took a total of 15 hours to design and finish because of small bugs that were eventually found through the app after the initial testing stage.

Total Commitment: 28 hours (excluding shipping)

## Hardware Design/Assembly

1. Design a PCB in Fritzing to connect to the GPIO of the Raspberry Pi with header pins for the wiring harness to the three separate sensor PCBs. File in Git.
2. Design 3 separate PCBs in Fritzing, one for each sensor. These are designed in a way where the wiring harnesses would not have to twist to connect to them. Another design aspect is to make them all the same size so a single case design will fit them all. Also, mounting holes are added to easily mount them to the cases. Files in Git.
3. Design a case to fit the Raspberry Pi and connected PCB in Corel Draw. Hole added at the top for all the cables to be routed through. File in Git.
4. Design a single case to fit the three other PCBs in Corel Draw, easiest way is to shrink down the design of the Raspberry Pi’s case to the proper dimensions. Files in Git.
5. Have the 4 PCBs and 4 cases printed. This was done through the Prototype lab at Humber College.
6. Soldering. There is quite a bit of soldering to be done between all 4 PCBs. All the vias need to be soldering with small pieces of wire, the header pins for the cabling and the Raspberry Pi and the sockets for each of the sensors.
7. Splicing the power cable. A Step-Down Buck converter is used to power the Telemetry system from the 12V battery of the Baja racing dune buggy. A micro USB cable needs to be cut and the power and ground wires spliced and soldered to the output end of the step-down converter. The input end then needs stripped so it is ready to connect to a 12V car battery.
8. Finally, the 3 PCBs need mounted to their respective cases and the Raspberry Pi mounted to it’s case and the wires run from the Raspberry Pi unit to each of the sensor units.

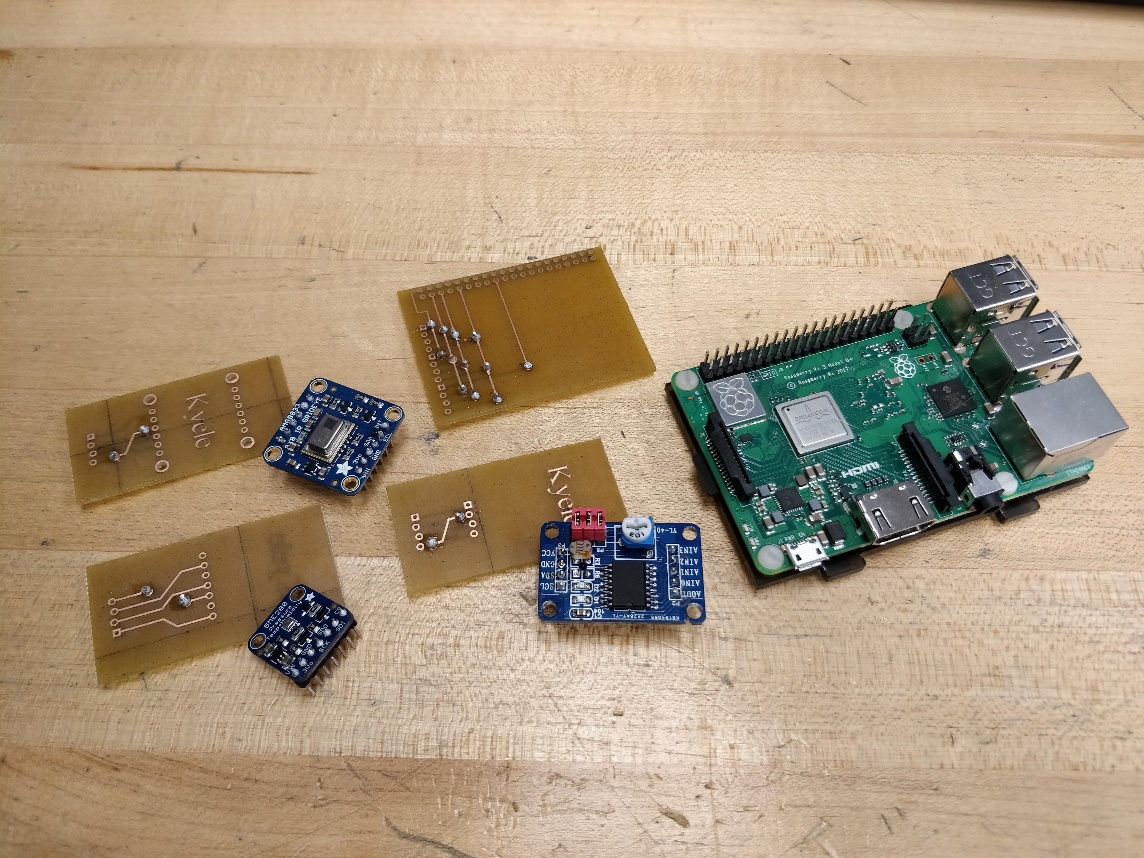


Figure 4 – All PCB’s and Sensors

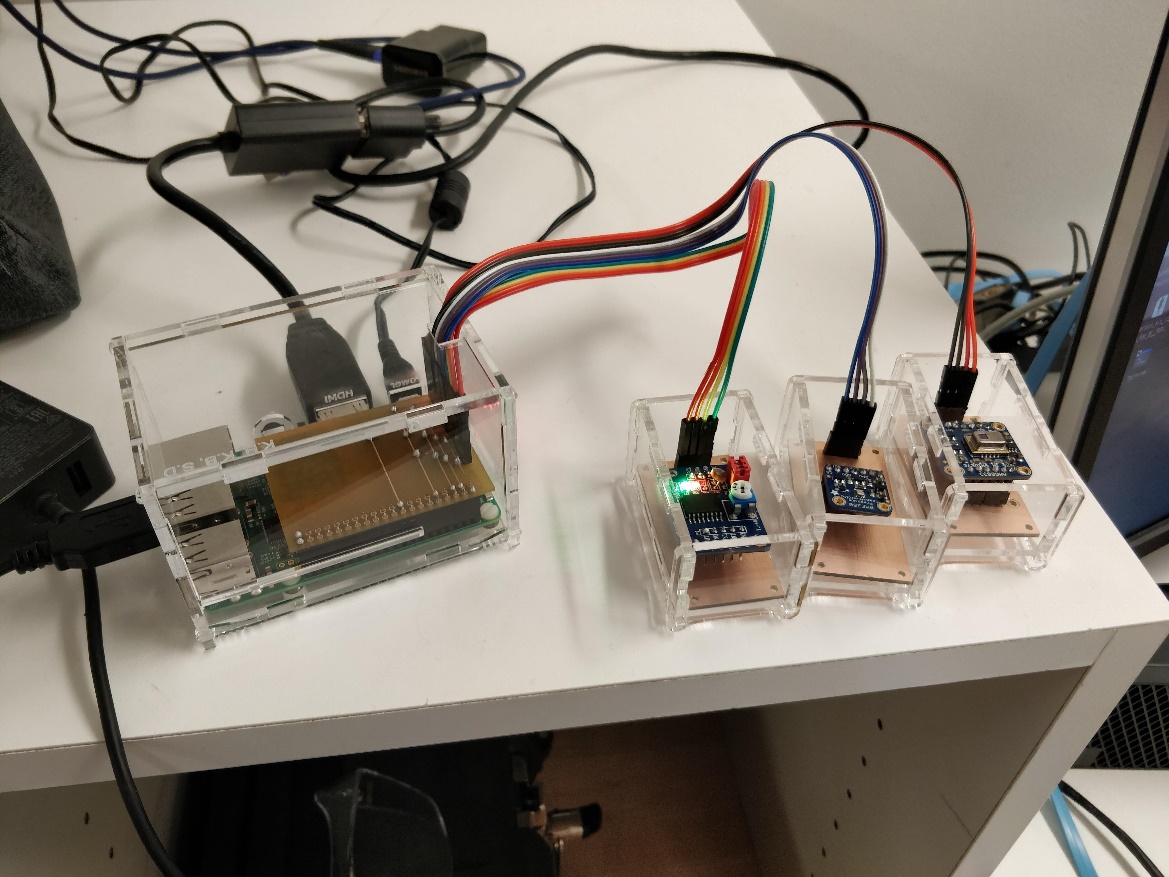


Figure 5 – Final form of the project

## Power Up

To be added when all parts are working on one enclosure

## Production Testing

After powering on the Raspberry Pi, make sure you test if the addresses of the sensors are correct;

1. Connect the Pi to a display using an HDMI cable/adapter
2. Turn on the Pi (wait for it to boot up)
3. Open a terminal window
4. Run the command sudo raspi config
5. Enable I2C Address
6. Exit out
7. Run sudo reboot
8. After reboot open another terminal window
9. Run the command sudo I2Cdetect -y 1
10. It should show that the addresses that are connected are 0x48, 0x69 and 0x77

## Unit Testing

1. Run the command sudo git clone <https://github.com/KogulB/>CENG355Project
2. This should clone the CENG355Project git to your root folder
3. Run cd / CENG355Project/Software
4. Then run sudo ./myProg.sh

* To test each sensor run “Python 3 <filename>.py”
* This should cause each sensor to run according to the python values and start giving temperature, light and pressure values.
  + Note that the PCF8591 (Light Sensor location) should be emitting a green light

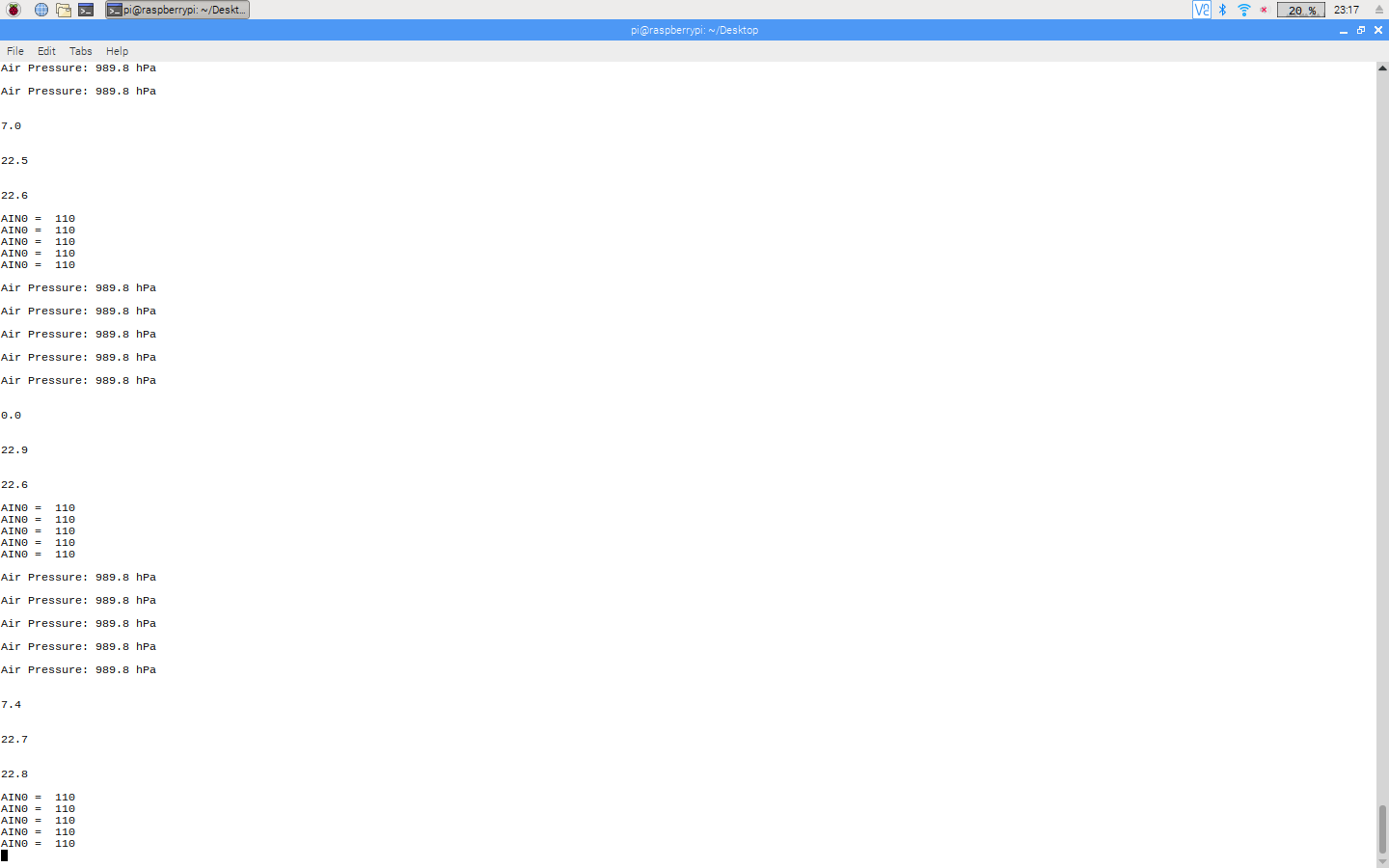


Figure 6 – Output of values uploaded to database



Figure 7 Firebase Database

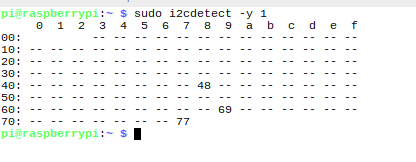


Figure 8 Sensor Connections

## Mass Production

If there was to be a mass production of this project, we believe that soldering the sensors directly to the PCBs would be much more efficient. This way, the case would be smaller allowing for a cheaper overall cost and smaller footprint when in use. Smaller connectors for the wiring harness between all the PCBs would also allow for a smaller footprint. Another Possible way of mass production is the casing for the diagram by adding a more 3D-printed water proof case that could stand the environments that a Baja racing vehicle brings.

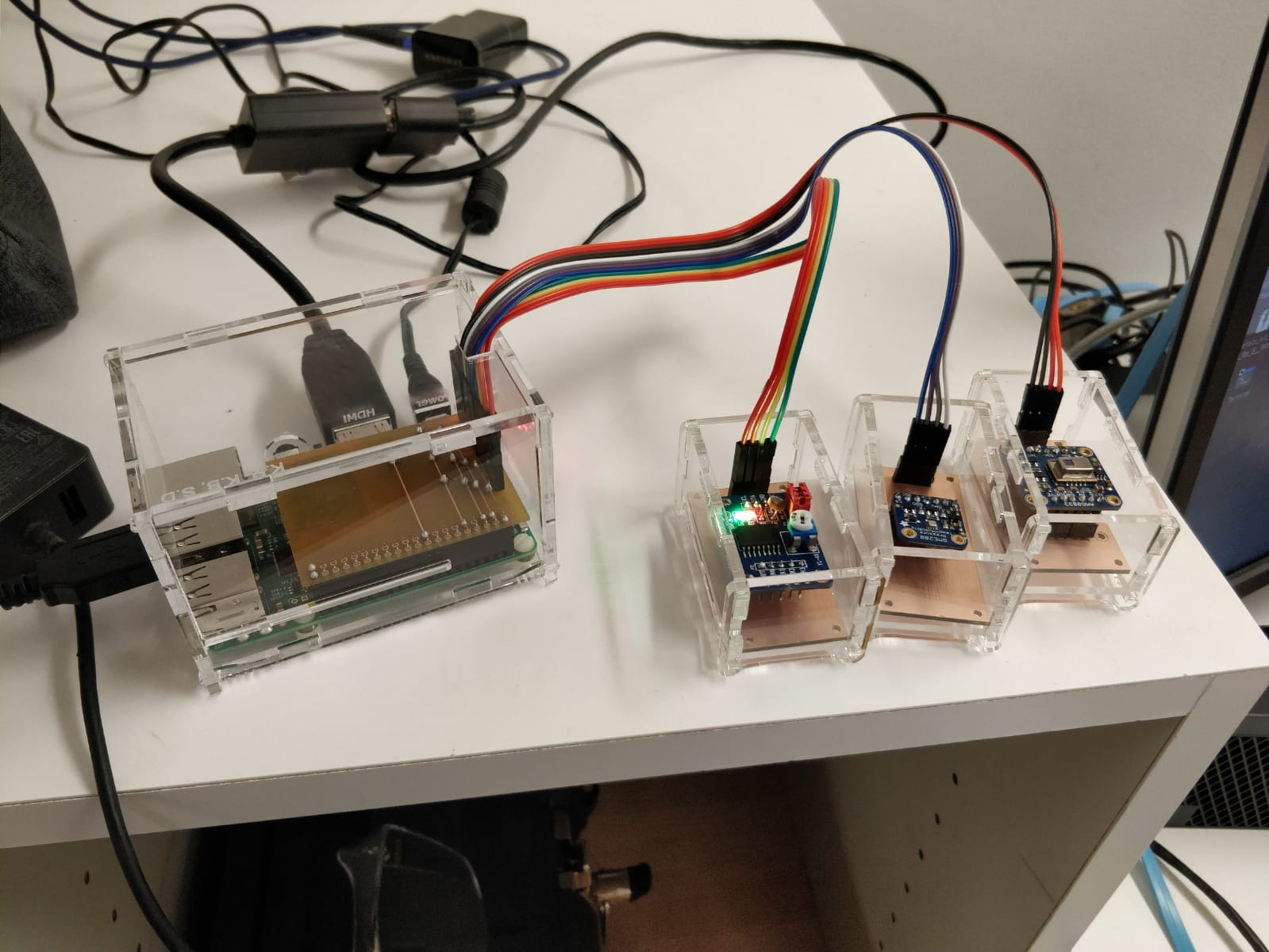
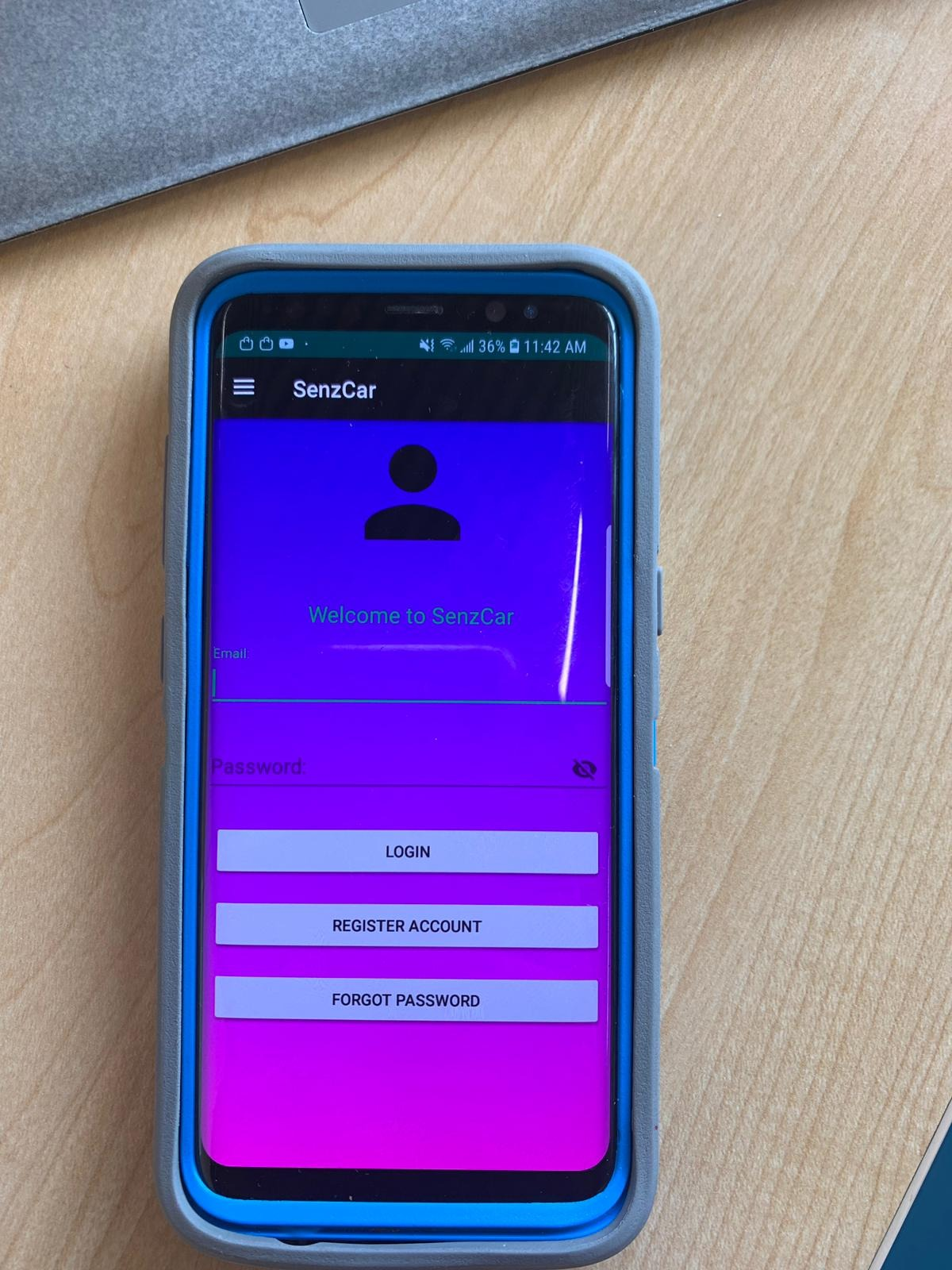
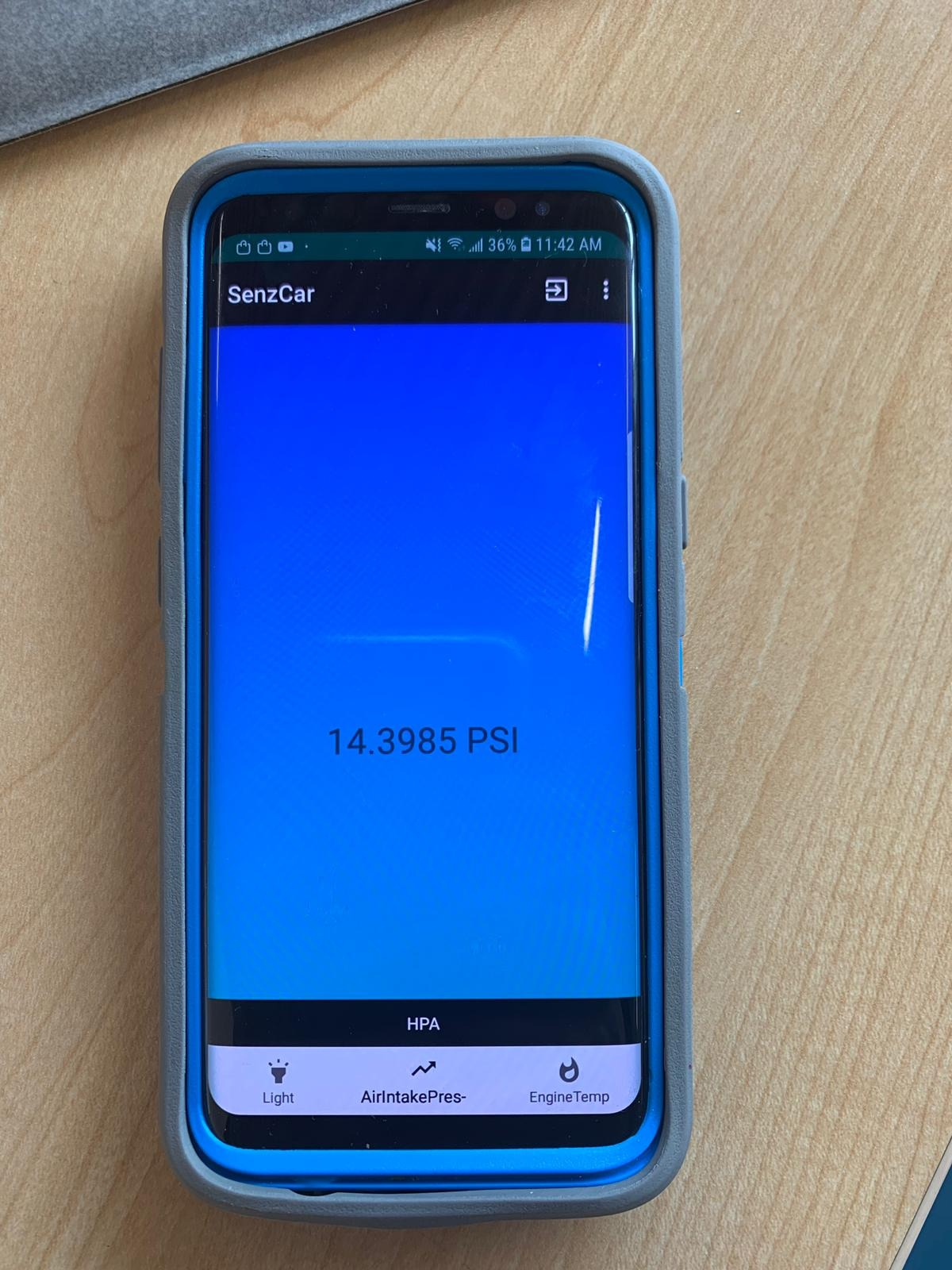


Figure 9 – Finished Product

## Android Application

The android application puts all the data gathered from the sensors together into one format that the user can interact with. There is a simple login and registration page that provides the user with a simple idea on how to get started for viewing the racing vehicle data. Once logged in the user is unable to go back to the previous page without the app logging him or her out which makes the security of your app and much better experience. Once logged in you are given three tabs that display the three sensor readings Light, Air Intake Pressure and Engine Temperature. Each tab displays its corresponding value. The engine temperature can be viewed in either Celsius or Fahrenheit and the Air intake pressure in PSI or HPA. The user also has the option of deleting account which eliminates there access to the account.



# Conclusion

In Conclusion, our telemetry system is designed to help Baha racing vehicle owners to better maintain their vehicle and be able to identify basic problems that require minimal solutions. Three sensors are implemented to monitor and interact with everyday use of the vehicle. Any data collected is then sent to a real-time database and can be viewed by the owner of the vehicle through an android application that essentially provides the latest data that was collected; Login privileges are required to view the information on each particular vehicle. Purpose of this is too save the driver money by fixing minimal issues by themselves as well as kind of provide a warning system for future potential issues. Each sensor provides its own data (as described in the report) and is all together locked in an enclosure running on a development platform. The system makes good use of all the information collected and can be viewed on a kind of graph of past and current data.

# Bibliography

Gilsdorf, J. M., Vercauteren, K. C., Hygnstrom, S. E., Walter, W. D., Boner, J. R., & Clements, G. M. (2008). An integrated vehicle-mounted telemetry system for VHF telemetry applications. Journal of Wildlife Management, 72(5), 1241-1246. Retrieved from http://ezproxy.humber.ca/login?url=https://search-proquest-com.ezproxy.humber.ca/docview/234179670?accountid=11530

[2] Institute of Electrical and Electronics Engineers. (2015, August 28). IEEE Xplore Digital Library [Online]. Available: https://ieeexplore.ieee.org/search/advsearch.jsp

[3] Leonardo Presoto de Oliveira, Marco Aurélio Wehrmeister, AndréSchneider de Oliveira, "Systematic Literature Review on Automotive Diagnostics", Computing Systems Engineering (SBESC) 2017 VII Brazilian Symposium on, pp. 1-8, 2017.

<https://github.com/adafruit/Adafruit_CircuitPython_BME280/blob/master/examples/bme280_simpletest.py>

<https://github.com/adafruit/Adafruit_CircuitPython_AMG88xx/blob/master/examples/amg88xx_simpletest.py>