Smart Steering Wheel

Dainis Boumber, Kyeongan Kwon, and Varun Prakash
CPS Course project report – Fall 2012
University of Houston

Abstract

Smart Steering Wheel's primary goal is to improve driver safety. Furthermore, we are motivated by a desire to learn more about the problems that arise when sensors are used to collect data about human body. In this project, we build a system that is placed around a steering wheel of a car. The system has various sensors that gather data when the vehicle is operated. Once the data is collected, it is stored for later analysis in a format that can be easily processed by various Data Mining tools. This data can be used, for example, to find patterns in driver behavior and analyze the relationship between driver actions, surroundings, and stress level. However, data analysis is a different problem altogether. Therefore, in this report, we focus on implementation details and challenges we faced while working on this project.

1.0 Introduction

On a large scale, we are attempting to look at different ways of quantifying the interactions between the human body and the environment. We chose the specific type of interaction because most other experiments in this area involve medical research, and few novel things can be done at a low cost. For a semester long project, this problem is not feasible to solve. We chose to take a subset of the problem and work on it. We attempt to build a complete system with a number of sensors that is a close to realistic set up as possible. This system is to gather various types of data and record it.

We planned to solve this problem by using 5-6 different types of sensors, including, but not limited to, accelerometer, GPS, pulse sensor, temperature sensors, pressure sensors, moisture sensors and others. We intended to use an Arduino board, a Mux shield and a bread-board to connect the sensors and process the data. We were going to wirelessly transmit the data to a laptop. For this purpose, an Arduino WiFi shield was used. All these components were soldered together and powered by a battery pack with USB. The sensors were attached to a wheel cover. The complete setup was installed in a car to be test-driven.

Overall, the experiment was successful. We faced some interesting challenges and had to take some shortcuts. In the end, we ended up using only three types of sensors, GPS, accelerometer and pressure. Even in this limited set up, the resulting data file has ten features. This problem is entirely due to the lack of time and components, for example there was no suitable pulse sensor, moisture sensor does not exist in the market and constructing one is beyond our expertise, etc. We faced a problem connecting all things together and stacking one shield on top of the other. For example, there exists a conflict between GPS and Mux shield's control pins. To get around this issue, we used a Master-Slave set up with two Arduino boards, which presented us with software challenges. WiFi would not work due to being shielded from network signal by other components and we ended up using an SD shield instead. Some wires were not long enough. In the end, the experiment did not collect sufficient data to talk about analysis, but it did prove the concept of a possibility of doing so. Therefore, it is a success.

2. 0 Design and Implementation

2.1 Using the device

The main interface with human users is the pressure sensors that are placed on the outer surface of the cover. This acts as the primary component to record the force with which the driver of a car grips the steering wheel, which is our parameter to judge driver stress and behavior. The pressure sensors (24" Long FSR sensors) are then connected in circuit with other components of the product using the breadboard. The other components viz. the GPS module and accelerometer are built into the central unit as they do not have the need to come in contact with the user and can function autonomously.

2.3 Requirements

The product is mainly required to withstand general driving conditions without offering barricading inconvenience to the task of driving itself. As the leading entity on the requirements list, the device must be able to successfully record continuous data of the driver behavior over a certain amount of time. It is also important, in our case, that the device stays intact due to the hard fact that there are many components that are connected merely using jumper wires. As for the technical requirements, the device is to get continuous readings from the pressure sensors and the accelerometer and add the GPS data to the file when there is a signal available.

2.2 Architecture

We mainly rely on stackable modules that are compatible with Arduino UNO. There are 3 main modules or "shields" that we make use of to carry out the functionalities of our device, the MUX shield, the GPS shield and the SD card data logger shield. It is on a trial and error basis that the combination of the shields is designed due to instances where two different shields (especially the MUX shield and GPS) would not work if placed together on the same Arduino board. To overcome this, a master –slave design is employed where we use two Arduino boards instead of one. A block diagram of the system is shown below.

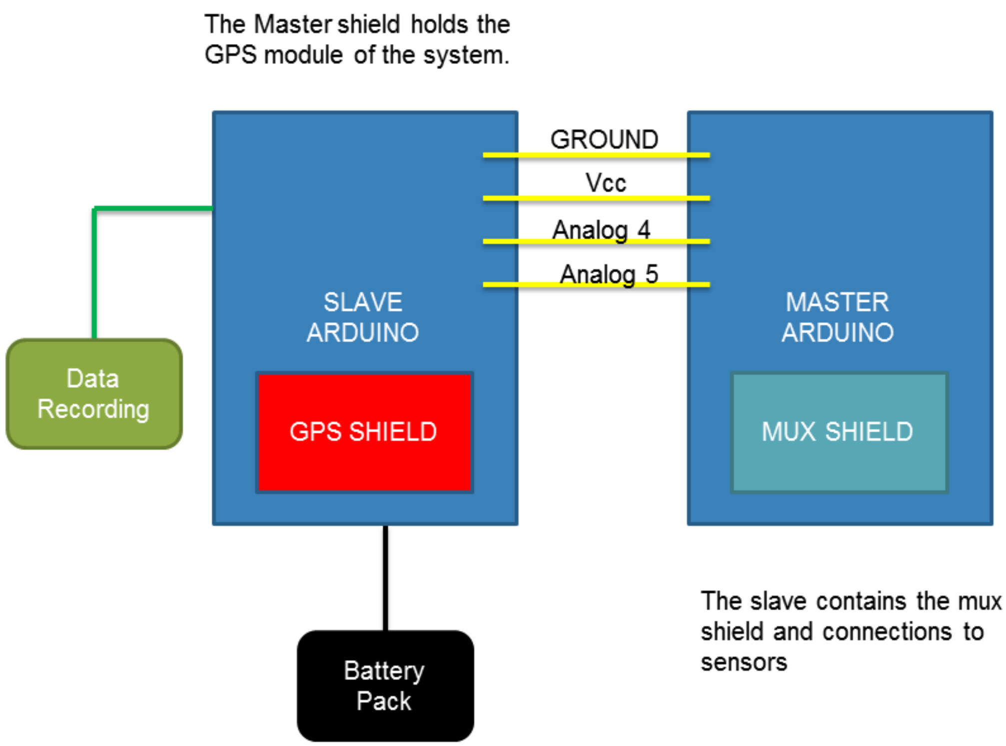


Fig 1: Block Diagram of the Architecture

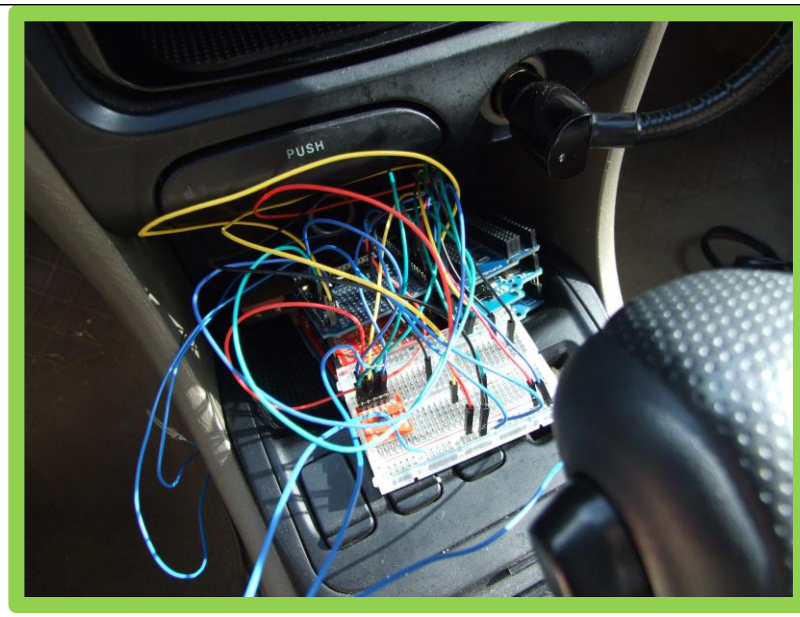
As represented above, the two Arduino boards work in unison to obtain data from the MUX shield and the GPS module at the same time. The shields are stacked on top of Arduino boards using stackable header pins. The Mux shield on the master generates analog data from sensors which is then wired into the slave arduino connected through 4 pins, namely the Voltage (Vin), Ground (Gnd), Analog 4 and Analog 5. The GPS module on the other hand, inputs digital data to the arduino board. The combined information from both boards is then written into a comma separated file on a 1 GB data card in the recording shield.

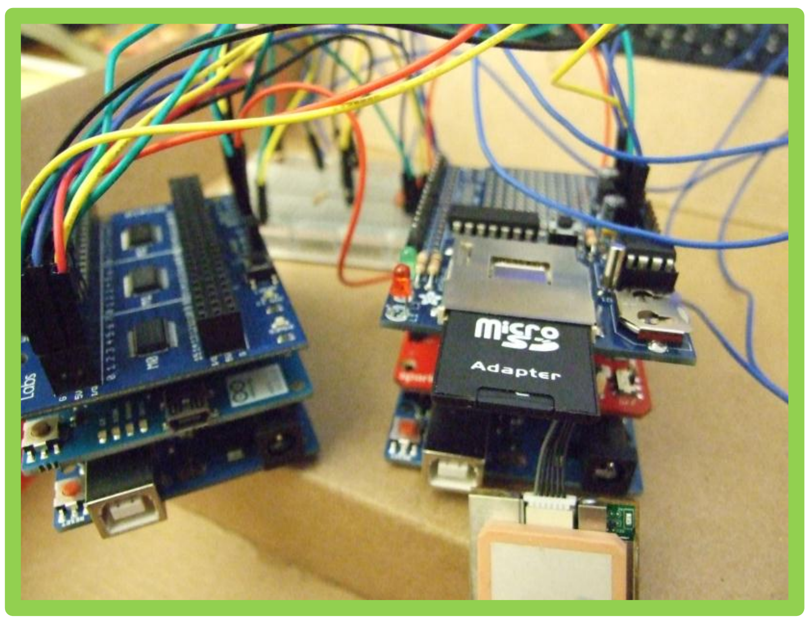
2.3 Hardware

The GPS shield and Data Logging shield are sold as sets of unassembled components. This way, a considerable effort was spent in assembling and soldering each component and stackable headers in order to make the shields stack upon the Arduino board. Other hardware includes the use of a Breadboard to obtain a better connecting field for the resistors and the accelerometer breakout.

Below figures show how cover looks with real world and

1. Left side with Smart Steering Wheel cover.	
2. Bottom side with Smart Steering Wheel cover. To avoid getting entangled lines, I setup two sensors to bottom faced. It is converged in one side, then, it give driver better experience compared to first our simulation.	

3. All Hardware puts down some space in Car. The space is enough to keep it and hold it well when driving.	
4. Figure with a driver and our product.	

5. Figure with all hardware being installed	
6. Figure with a SD card	

12.4 Software

We used two Arduino boards, one acted as a Master, the other one is a Slave. The software work consisted of reading sensor output, and writing it to an SD card. It was a little bit complex because the signal had to be transmitted between two boards, which was somewhat involved because transmission length was limited and the transmission was not 100% reliable. To further complicate things, Arduino is very simple and has no well-established synchronization libraries, so a some fine-tuning had to be done to avoid dead-locks and race conditions.

2.5 Contribution report

Task	Contributor	Status
Connections on breadboard	Varun	Success
Accelerometer	Varun	Success
Soldering GPS	Varun, Dainis	Success
Mux shield soldering	Varun, Dainis	Success
Master assembly	Dainis	Success
Slave assembly	Dainis, Varun	Success
Master-Slave connection	Dainis, Varun	Success
Master-Slave code	Dainis, Karl	Success
Ad-hoc wifi	Dainis	Failure
AP wifi	Dainis	Partial; does not work when more shields are installed
SD shield assembly	Varun	Success
SD shield installation	Varun, Dainis	Success
SD shield Valid. & Testing	Varun	Success
SD shield code	Varun	Success
Pressure sensor installation	Varun	Success
Wheel assembly	Varun	Success
Reports	Varun, Dainis, Karl	To be graded
Testing and debug code	Karl	Success
Assemble hardware with car	Karl	Success
Recoding videos, taking pictures and editing	Karl	Success
Presentations	Karl, Varun, Dainis	Success
Data collection	Karl	Success
Google Graph, Form and Arduino Reader	Karl	Success

3.0 Results

3.1 Requirements

Our team did meet all requirements completely. All of the sensors are working well and save data to SD card properly. It means all hardware is assembled with no problems. Of course, sometimes GPS has no signal even though outside. Although we still don't know why it happens occasionally, we had collected the data and can make graphs. Then, we can show these results during final presentation. Additionally, for example, pressure sensors react and give different values with different pressures when we are testing and driving. Also, accelerometer shows different X, Y and Z axis when it moves. Like pressure and accelerometer, GPS gives us some type data such as longitude, latitude, speed and etc. without problem.
In analysis, graphically we can say whether driver is nervous or not at some moment. Overall, we determine all requirement of our project to be fine.

3.2 Data

The device was tested in a live scenario by installing it on the steering wheel of a car and driving the car for a distance in and around the campus of the university. The driving has been kept as natural and as possible. The data obtained contains variation in pressure, accelerometer and a slight one in the GPS readings.

254	26	31.85	30.81	46.17	29	-96	121	2	11

The table shown above is an example annotation of how data is written into the comma separated files. A short explanation is given each of the components.

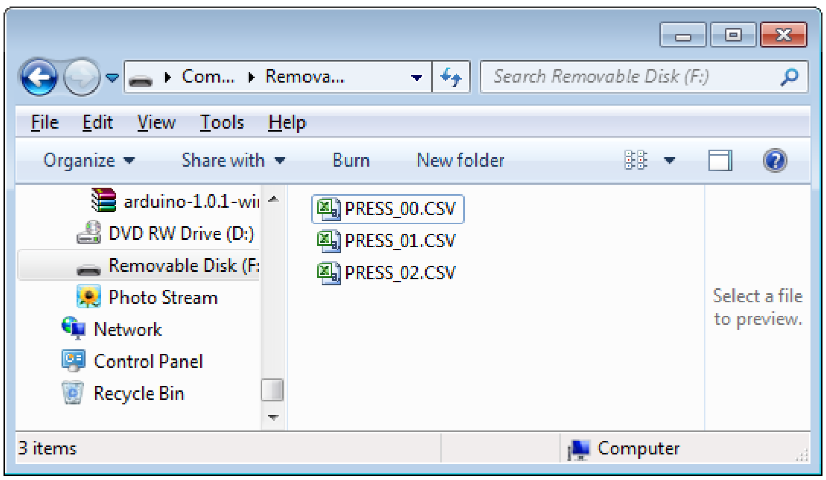
Column 1 [254] and Column 2 [26]: The raw sensor reading obtained from the first pressure sensor installed on the steering wheel. As such, the number does not have any unit as changes to pressure calculated in Newtons would not provide the required calibration to get fine pressure data.

Column 3 [31.85], Column 4 [30.81] and Column 5 [46.17]: These columns represent the X, Y and Z axis readings of the accelerometer. Sudden changes in the driving terrain such as a big pothole or a bump on the road causes large amount of change in the accelerometer readings in a short amount of time.

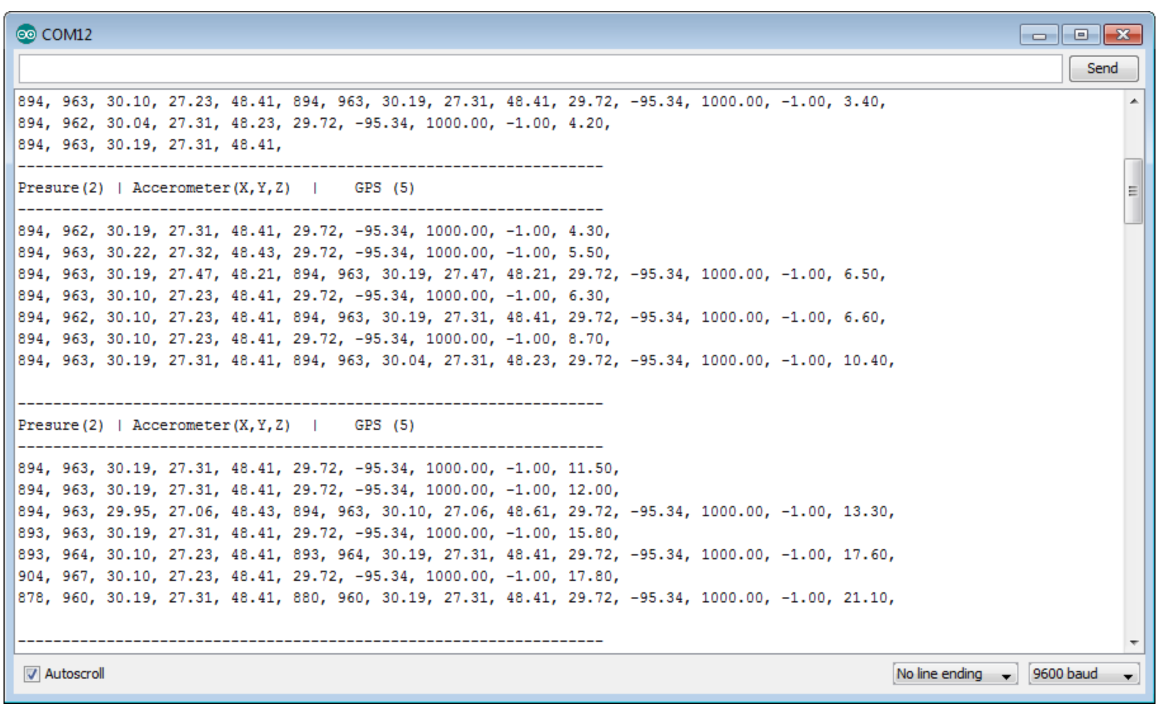
The remaining columns are readings from the GPS. They are arranged in the following manner.

Latitude – The Latitude of the current location of the driver.
Longitude – The Longitude of the current Location of the driver.
Course – The approximate heading or direction of the vehicle.
Speed – The speed with which the vehicle in the given course.
Altitude – Altitude of the location above sea level.

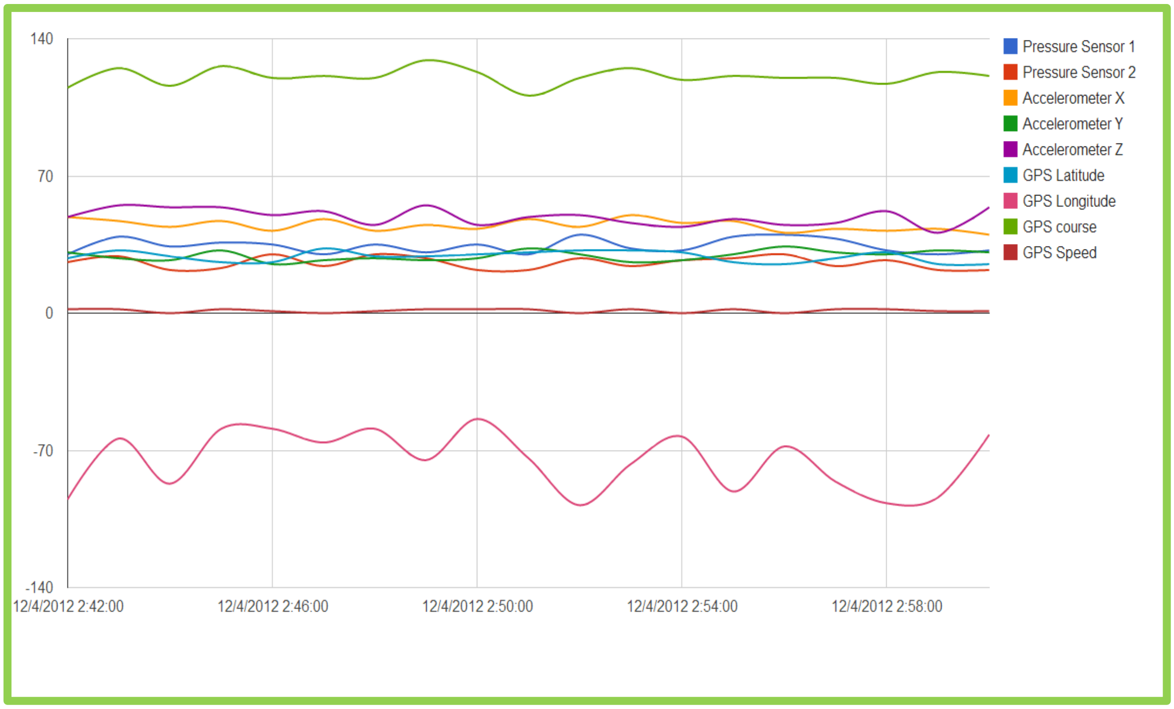
When running, the data save into SD card and the file is sequentially named as you can see in below figure. (A data file 'PRESS_OO.CSV' will be attached.)

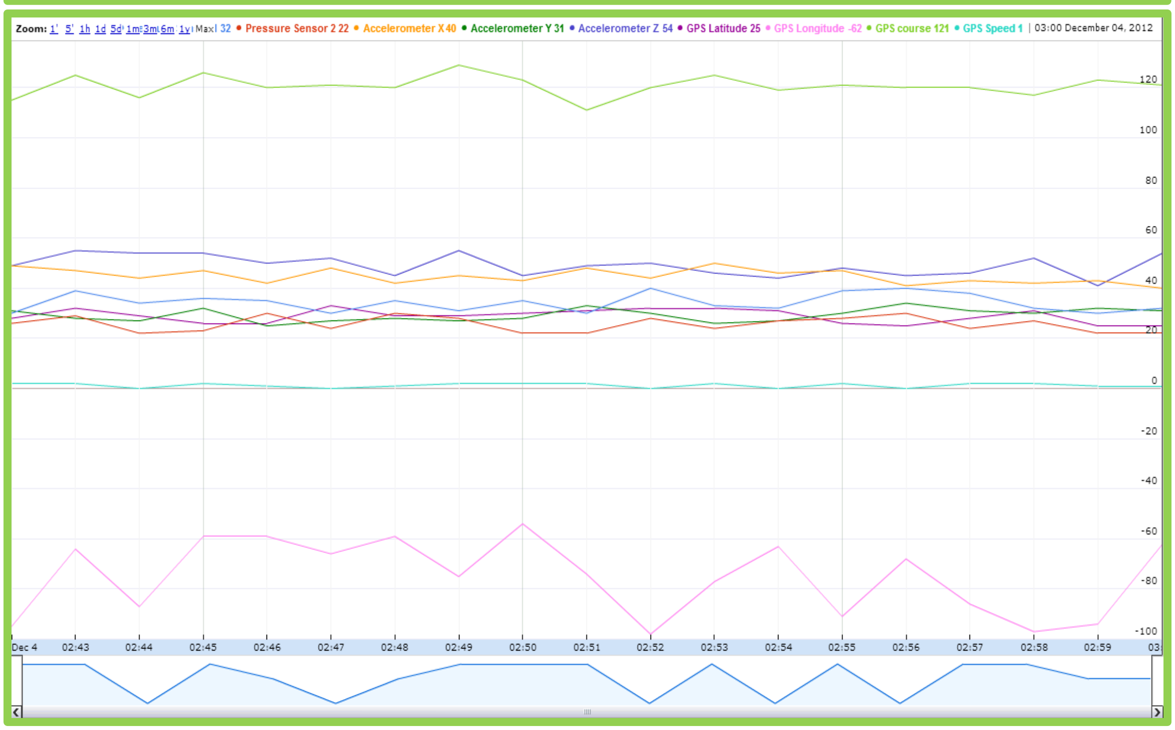


Data output example



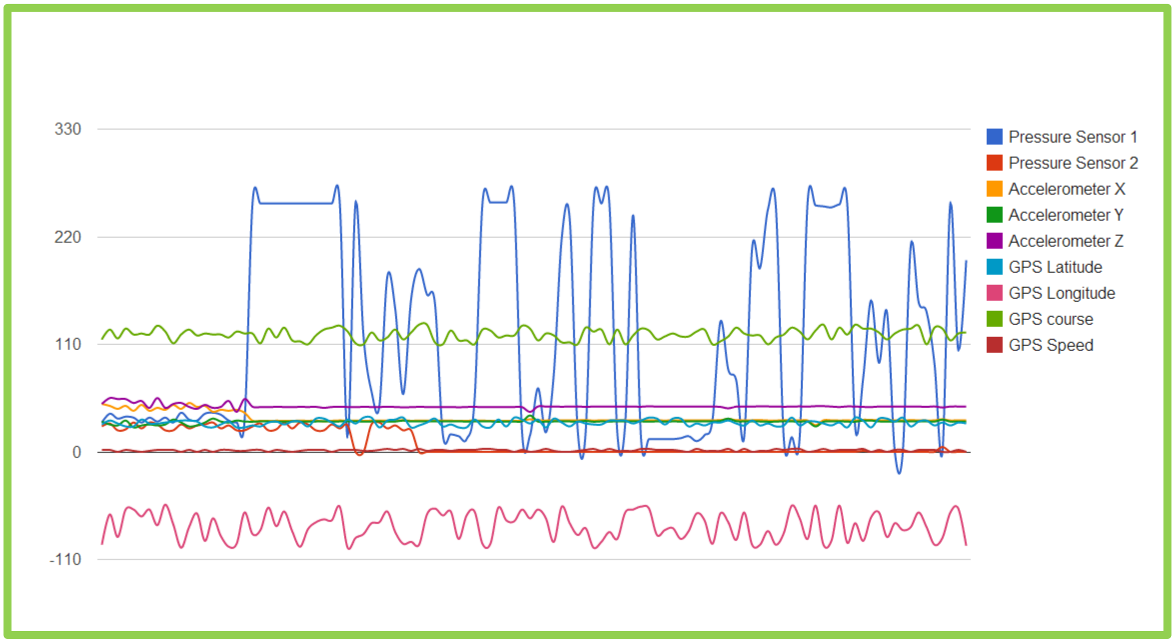
Below two graphs are from previous data.





This is new graph made with new data. (Note: Actually this data was combimed with previous and new one, and it was dicussed with Professor Gnawali.)

We can say that some of blue pressure sensors (1) have high value signal, and we can guess driver might be nervious whlie driving.



4.0 Results

Modeling human behavior in his day to day activities and extracting important information from it has been a long ventured area since the development of sophisticated recording and computation devices.

Our device aims at performing similar tasks with a particular operational scenario in mind. The task of driving has been one of the most common and well-practiced tasks in the recent history. Yet there are many incidents that involve fatal accidents of drivers. Car companies have long researched and produced sensors and devices that improve the quality of safety for a driver. But the inevitable fact that human error can cause the overriding of any safety measure needs to be, still, understood. In an attempt to understand driver behavior in various stress levels, surrounding changes and location, we have designed and implemented this device which helps researchers conduct a detailed study on effects and causes of accidents and other phenomena without merely rooting out the human error factor as inconceivable.

There have been many devices that record driver statistics such as alertness in the past. There have undoubtedly been many devices in the car itself that record and monitor the status of the car. But as far as the factor of human stress is concerned, we have not come across or heard of any devices which have a modus operandi similar to ours.

Conclusions

In this project, we learned a lot that we didn't know before. First of all, we learn how to design hardware, how to solder and combine it together, how to implement and debug embedded programming, and etc. Specially, we were able to learn how various sensors such as pressure, accelerometer and GPS work, and how these sensors give data to us. Also, we have built couple of things. It is following: Two Arduinos, GPS Shield, MUX Shield, SD card Shield, and breadboard with an accelerometer. We stuck two pressure sensors to wheel cover and it connects to what we built. Based on all of them, we might expand our imagination of our mind. Additionally, we can apply our knowledge into our future researches or improve our current research.