# Wearable Fall Detection Device

## EE 4OI6 – Engineering Design

Group #H14

Jingnan Chen #1073196
Sebastien Woo #1058846
Chao Zhang #1040310
Cheng Zhou #1057871

**ABSTRACT--With the rapid growth in the elderly population the need for more health related products to help aid the elderly is immense. A common injury among the elderly population is the occurrence of a sudden fall. Therefore the objective for this project is to create a fall detection system that will help aid elderly when a fall occurs. The system consists of a wearable device that will monitor the individual and recognize a fall. In addition to the wearable device, a smartphone application works in parallel with the device to increase fall detection accuracy and allows for other health monitoring features. Through our development, testing and analysis gave way to a successful product which will be further discussed in this report.**

## I. INTRODUCTION

The growing elderly population creates a whole new market for heath related products. In this day and age, the impact of technology in our daily lives is immense, and combining technology to help aid humans in their health is the future to further our relationship between man and machine. The sudden fall detection device is a way of providing a valuable service to the elderly population by looking after their well-being. Each year 1 in every 3 elderly over the age of 65 will encounter a sudden fall and it is one of the leading causes to fatal and non-fatal injuries among the elderly population [1]. In 2013 alone 25,500 people died in the United States from an unintentional fall [2]. These cases can vary from those who died instantaneously to those who did not get help immediately. The fall detection device helps provide a layer of security for cases where the user is not able to call for help. The impact of such a device can dramatically reduce the fatality from sudden falls.

The aim for this project is to create a wearable device that is able to detect a sudden fall. The implementation for this device will be further discussed in the report. The objective of the fall detection unit will be to automatically detect when a fall occurs without having the user to signal for help. There are many products out in the market right now that tackles this issue but each has their drawback. Our device takes the best features from all of these devices and merges them into our device. Products like the Philips Lifeline are wearable devices that have a push button that can be used to signal for help [3]. It is a great solution for general cases, but in the event of the user being unconscious the device would be useless. It is also limited to the range since it must communicate to a base station. Another product called MobileHelp, is also a wearable device but unlike the Philips Lifeline the Mobile Help can automatically detect falls and has a greater range [4]. The only con for this product is the subscription fee since the provided service requires annual fees. Mobile Help is an active system which is more effective than the Philips lifeline which is a passive system [5]. The development of our product is another stepping stone towards raising the bar in terms of sudden fall detection.

By creating a wearable device that is linked to a phone application will open up the possibilities for various features that purely hardware driven approaches fall short of. Therefore our device gives maximum freedom to the user since the range of the device is limited to the range of the user's cellular provider. Our assumption is that in the future the elderly population will have smartphones, so why not employ the smart phone to create a more sophisticated product in detecting falls. As a result when a sudden fall has been detected our fall detection device will send out a distress message using our fall detection android application. The distress message will be sent to a recipient that is responsible for the user. It will include a text message to alert the recipient and a link of the exact location using google maps. There are two major components to our fall detection device. The first component is the headset which includes sensors that calculates real time data of the user's acceleration and orientation in free space. It will stream data to the second component which is the android application. The phone application adds a second layer of accuracy to the fall detection device by using the onboard sensors which will be used in conjunction with the data from the headset to make accurate decisions for whether or not a fall has occurred. The device therefore has two modes, one mode is called the standalone mode which is the headset making the decisions and the second mode called the Master-Slave mode is when both the headset and phone application are paired. The phone application has many features such as a pedometer to track the number of steps taken, a graph to see real time data, data logger, distance traveled, and a calorie counter. This concept of a fall detection device will serve to be an important line of defense to mitigate the death and injury rate that comes from sudden falls.

## II. DESIGN METHODOLOGY

The method in deciding a fall was based on the data from the sensor on the head and the data on the phone. If the headset and the Android phone is connected, the phone will make the decision and the headset will wait for the trigger message from the phone. If the acceleration magnitude of the phone exceeds threshold or acceleration magnitude of the device is greater than threshold and the pitch and roll angle is greater than threshold angle, then the phone will trigger a time frame of 0.6 second. During this time frame the device will wait for the other conditions to be met. If the headset is not connected to the Android phone, the headset will be in a standalone mode (See Figure 1 and Figure 2 for flowcharts). Both devices calculate the acceleration magnitude as follows: $a = \sqrt{a_x^2 + a_y^2 + a_z^2}$ . From the IMU (Inertial measurement unit), we can have the Quaternion q = ($q_0$, $q_1$, $q_2$, $q_3$) with 4 elements from the DMP (Digital Motion Processor), and we can calculate the 3-axis gravity as follows [9]:

$$g_x = 2(q_1 q_3 - q_0 q_2)$$

$$g_y = 2(q_0 q_1 + q_2 q_3)$$

$$g_z = {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2$$

Thus we can calculate the pitch and roll angle as follows [8]:

$$pitch = \frac{g_x}{\sqrt{{g_y}^2 + {g_z}^2}}$$

$$roll = \frac{g_y}{\sqrt{{g_x}^2 + {g_z}^2}}$$

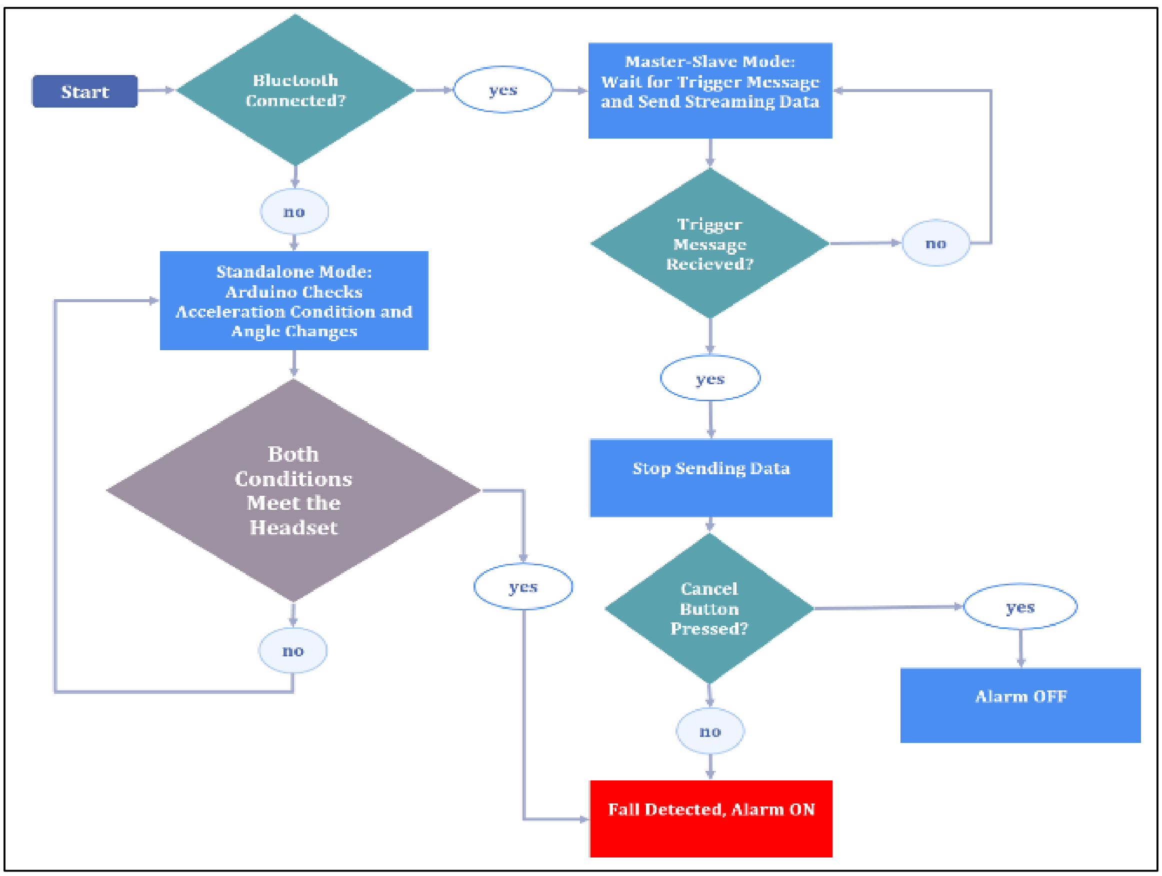


Figure 1 Algorithm Flowchart- Headphone Stand Alone Mode

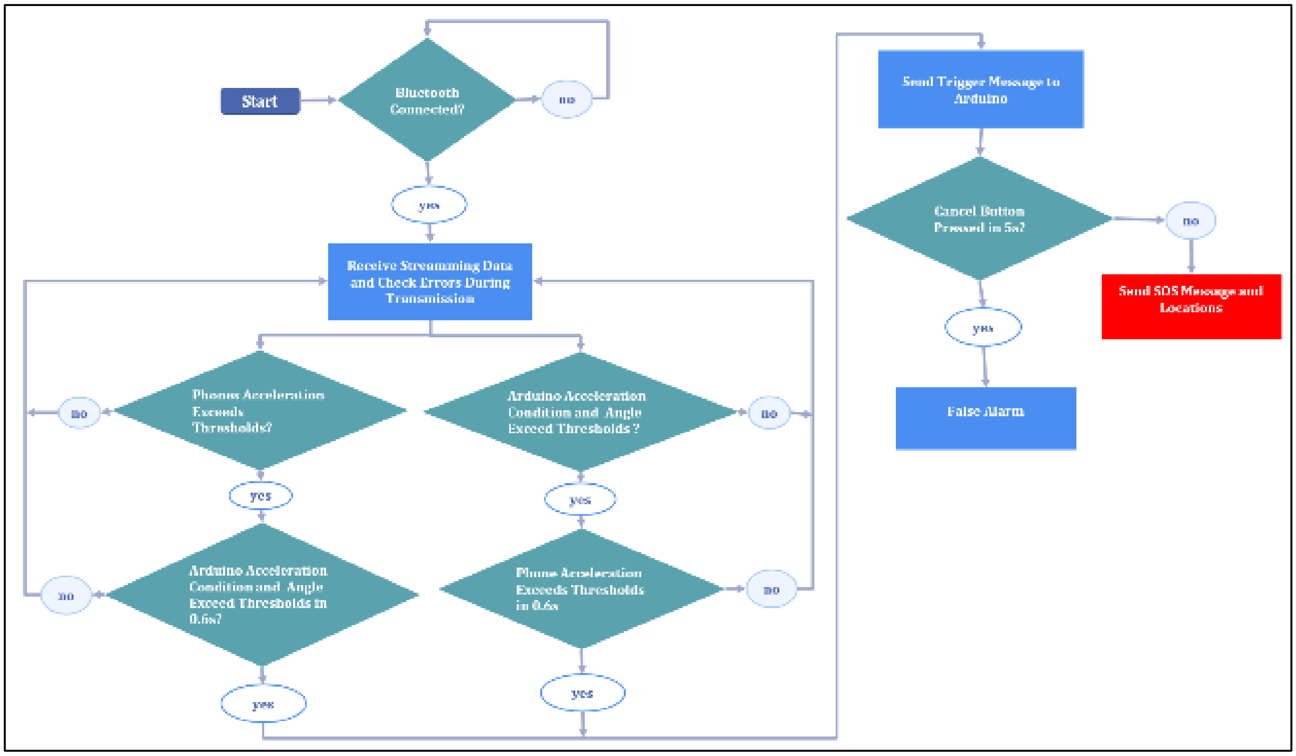


Figure 2 Algorithm Flowchart- Master Slave Mode

## III. ANALYSIS

### A. Fall Cases

Important parts of the testing that we conducted as a team were simulating fall types. By doing so allowed us to gather information about the acceleration produced from the head and waist in various fall cases. It was important to analysis our data so that our algorithm can be refined to increase the accuracy of detecting a sudden fall.

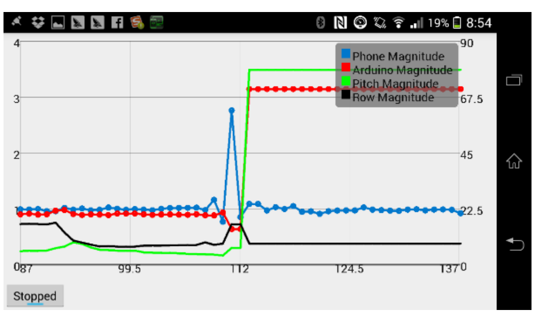


Figure 3 Fall Cases-Side Fall

The side fall is the most common fall type among elderly [6] and therefore was a case that needed to be detected. After several side falls, it can be seen from the graph that the acceleration from the head peaked almost twice the size of the acceleration from the waist. Also, the pitch angle changed which is a tell tail sign of a side fall. If the roll angle changed then it is most likely a front or back fall. The magnitude from the waist is relatively low when compared to the rest of the fall cases, and therefore our threshold for the phone sensor used the worst case value which was calculated to be 2G. This threshold for the phone gives us enough leeway to account for the worst possible case.

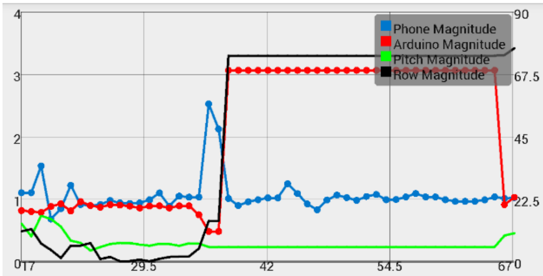


Figure 4 Fall Cases -Missing the Chair

For the elderly that have wheelchairs, a potential hazard is when they want to sit back down on their wheel chair. In the event where the elderly makes contact with the edge of the chair instead of the center, the chair can easily slide away resulting in a slow descending back fall. Another way to think about this situation is to imagine going to pre-school, and preforming the age old prank of pulling your friends chair before he/she sits down. This is the scenario we are trying to create. When preforming simulations for the fall, we notice that the time it took for both thresholds to be met was around 0.4 seconds give or take. So it was not always the case where both the head and waist acceleration were met at the same time but instead could lead or lag one another depending on the fall type. So to compensate for this matter, a time frame was used to check whether or not all condition has been met given the window size. Since the time lag between the two accelerations was around 0.4 seconds, we decided to use a time frame of 0.6 seconds to account for the worst case scenario. This test case is a prime example of how the analysis of our data led to the algorithm that we had developed.

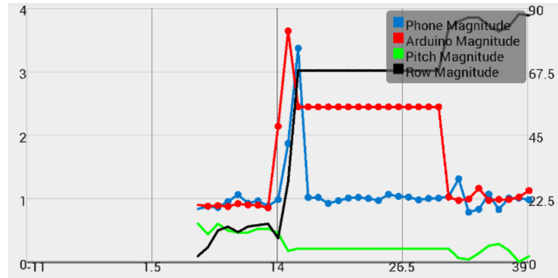


Figure 5 Fall Cases- Face Down Fall

The face down fall, is a very rare case and it usually happens when someone trips forward. As seen in the graph above, the face down fall can be easily identified from the rest because the sudden acceleration from the head peaked first before the waist. This is because in a face down fall, the head makes contact first then followed by the waist. The roll is another defining factor that this graph was a face fall because the roll angle is the angle is in line with forward and backward orientation. Although this fall type is very unlikely, it is a possibility that we have accounted for.

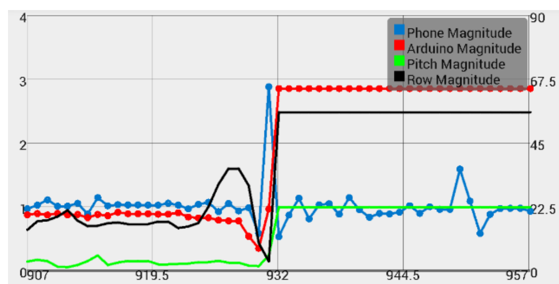


Figure 6 Fall Cases-Back Fall

A back fall is our last fall type and with this test our threshold for our angle was determined. In the event of a back fall, the natural instinct of a human being would be to keep their head up as high as possible to avoid making contact with the ground. So as a result, the change in angle was less in this case than the others. So to account for the worst case we had set the angle threshold to be 50 degree. The reason the angle is not around 90 degrees (which is what you would expect) is because the angle data actually lags behind the acceleration data. Also the other factor is because of our natural instinct of tilting out head away from danger. Another interesting characteristic in this fall was that the acceleration from the headset was lower than the other fall type. So our acceleration threshold for the headset was set to be 2.5G since the average acceleration in this particular fall was just under 3G.

B. Regular Activities

Regular Activities were tested to see the pattern that they had, which then can be compared to our fall cases. We categorized regular activities for the elderly in to 6 cases, which are Bend down, Hop, Spinning, Jogging, Sitting and Walking. Each case were performed five times, with the headset real time data and the phone sensor data being plotted on the application graph view, with a 0.2 second sampling time.

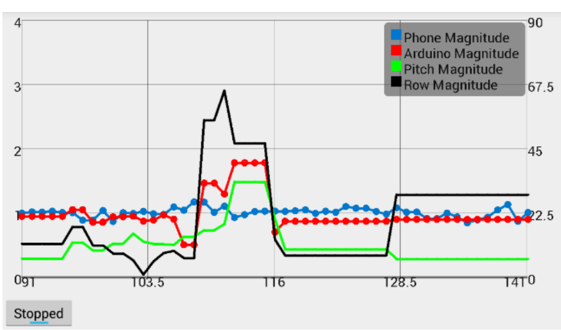


**Figure 7 Regular Activity- Bend down**

For a bend down case, a higher acceleration from the headset than phone acceleration can be seen. The reason is because the phone is positioned on the lower body whereas the headset is mounted to the head. Also roll angles and pitch angles each have an angle change of about 45 degrees, which is close to fall case. To distinguish this case from fall, we concluded that the phone acceleration does not meet the threshold; also neither pitch angle nor roll angle exceeds their thresholds.

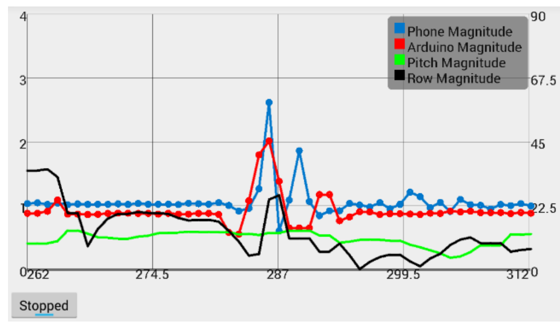


**Figure 8 Regular Activity- Hop**

For a hop case, the phone acceleration reached around 2.8G. This is the case because when performing a hop it requires a larger movement from the lower part of the body rather than the upper part. We conclude that in this case the fall alarm will not trigger as the headset acceleration magnitude and headset angle do not meet the threshold.

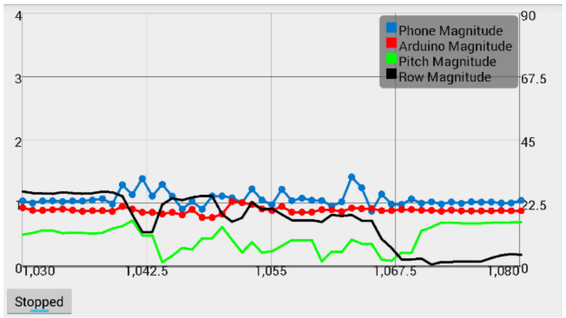


Figure 9 Regular Activity- Spinning

For a spinning case, low accelerations are shown on the graph for both lower part and upper part of the body. Also pitch angle and roll angle remain steady. The reason for this situation is that spinning will only change the yaw angle. We can distinguish this case easily from a fall since none of the data will exceed its threshold.

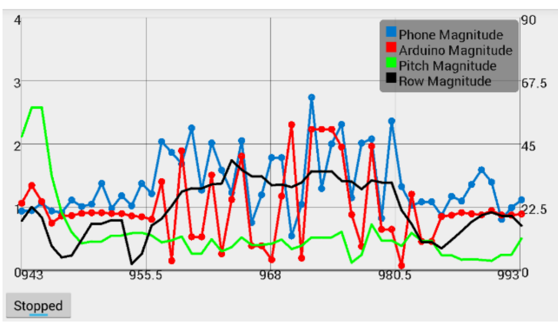


Figure 10 Regular Activity –Jogging

For the jogging case, fluctuations in accelerations below 2.8G can be seen in both headset and phone. It is due to the periodic movement of the upper and lower part of the body. However the pitch angle and roll angles still remain relatively steady below 40 degrees, so we can conclude that an alarm will not trigger since the angle thresholds are not met.

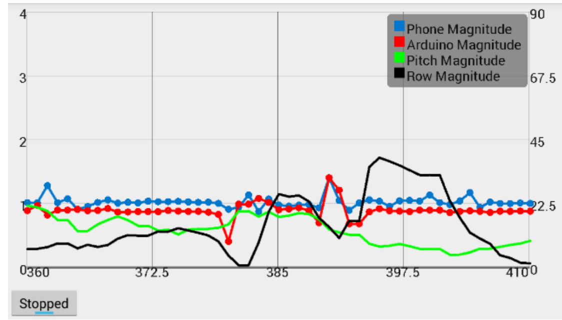


Figure 11 Regular Activity –Sitting and Standing up

For sitting down and the stand-up case, the spikes from the phone acceleration around 1.5G happens when user stands up. This case is also easy to distinguish from a fall as the headset acceleration magnitude and the angle changes do not exceed the threshold.

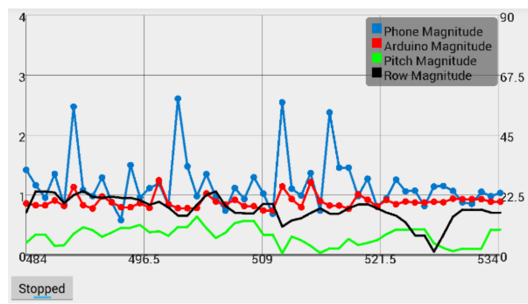


Figure 12 Regular Activity- Walking

For the walking case, both the phone acceleration and the headset acceleration are below 2.6G, and the angle changes are below 30 degrees. This case will not be considered as a fall, since all data spotted are within the thresholds.

## IV.IMPLEMENTATION

### A. Hardware-The headphone

The concept of wearable devices have been gaining popularity nowadays, therefore we decided to design a headset that will house a sensor to working conjunction with a smartphones sensors. We were planning to implement all the hardware parts on a safety helmet, however it cannot protect the component in a reasonable manner. So as a result we changed our mind to implement the device into an ear muff. The major components that made up the headset were an Arduino Uno, a protoshield, the IMU, Piezoelectric speaker, and the HC-05 Bluetooth module. A microcontroller (Arduino Uno) was needed to read the data from the sensors and control the other hardware components that will be integrated into the ear muff. The Uno was a perfect candidate for the task that we need it for. The amount of digital inputs/output were more than enough for our requirements and pairing that with the onboard Atmel 328 processor gave enough speed to running all of our components. Therefore a protoshield was needed to house all of our components since it keep the hardware manageable and organized. The piezoelectric speaker is used for the alarm which is triggered when a fall has been detected. By passing a frequency of impulses through the Arduino's digital outputs, the buzzer can create monotonic sounds. The buzzer is the best solution for a high efficiency speaker. Hooking up an 8 ohm speaker would have drawn too much power and was unnecessary for our application. The IMU (Inertial measurement unit) provides the 3-axis acceleration data, pitch and roll angles. We were planning to read the gyroscope sensor data and use it with the complimentary filter, since we will monitor the real time acceleration magnitude and the pitch and roll angles on the head. However, we read an article that performs the comparison between the DMP (Digital Motion Processor) Data Fusion and Complementary Filter. From a technophile mom block, Geek Mom Projects, "I can say that for pitch (rotation about the X-axis) and roll (rotation about the Y-axis), the calculations are fairly close, but the complementary filter seems to consistently lag the DMP." [10] Therefore through our research and testing we have concluded that the IMU would be best for our design. The last major component was the HC-05 Bluetooth module. This was a vital component in our design since it was responsible for communicating with the smartphone application. Modification to the HC-05 module was needed however to allow the Arduino to know if the Bluetooth module has be paired or not. There is no function on the Bluetooth library that determines whether or not the connection has been made. The only way would be to create a protocol. Through extensive research, we found out that one of the chips on the Bluetooth module has a pin that is high if a connection has been made, or is low if there is no connection [7]. By solving this issue, it made our software implementation a bit easier when it came to our communication protocol. The other components to the design included the 12V power supply, a push button to cancel the alarm, and a switch for the power supply. We have the following hardware parts in our design:

**Table 1 Table of Hardware Components and cost**

| Hardware | Cost |
| --- | --- |
| Arduino Uno | $25 |
| MPU 6050 | $15 |

| | |
|---|---|
| HC-05 Bluetooth Module | $20 |
| Arduino Proto Shield | $18 |
| 12V Power Pack | $4 |
| Push Potton | $2 |
| Ear Muff | Free |
| Switch | $2 |
| Jumper Cables | $4 |
| Buzzer | $2 |

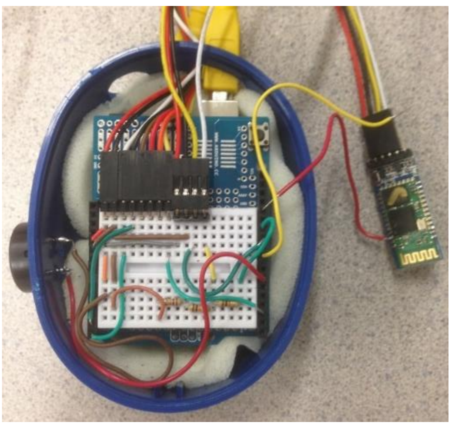

Figure 13 Arduino Board and Bluetooth Module

Figure 14 MP 6050

B. Software- Android Application

i Fall Detection Function

The android application for fall detection is designed in android SDK 21, and it is compatible with Android 4.3 or higher. This application implements various sensors and functions of the phone to perform reliable real-time Bluetooth data streaming, accurate fall detections and GPS coordinate tracking.

Emergency Contact Setup

On the main screen, the user needs to enter a cell phone number or import a number from his or her phone as an emergency contact. The distress message can also be edited. The changes on both emergency contact and distress message will be saved by choosing "Save Preference" on the title bar menu.

Bluetooth Connection and Data Processing

A Bluetooth connection with the headset is required for fall detection. By choosing "Search Bluetooth" on the title menu, the Bluetooth search activity will be opened. User can either choose an existing paired device to connect or scan for a new Bluetooth device to pair with. Once the Bluetooth connection is established, the program will start a background thread to continuously receive real-time data from the headset. The received data will be split by the character "/" and put into a FIFO queue. While the FIFO queue is not empty, the background thread will take the first element from the queue and split it with character ",". This will return an array that contains 6 strings, which are checksum, Pitch angle, Roll angle, and X, Y, Z accelerations respectively. To ensure data is not corrupted or lost during transmission, the sum of the last 5 strings will be compared to the value of checksum (unequal ones will be discarded).

Fall Decision Making

Once the application has a good set of data after data processing, the background thread will calculate the magnitude of acceleration based on received accelerations of three-axis. The application checks if a fall occurred based on both received data and acceleration magnitude of the phone. There are three cases for a possible fall: 1) the acceleration magnitudes from both headset and phone meet threshold, and pitch or roll angle difference meets threshold; 2) The acceleration magnitude of headset and either angle difference are meet threshold, and acceleration magnitude of the phone meets threshold within a 0.6 second window; 3) The acceleration magnitude of the phone meets threshold, and acceleration magnitude of the headset and either angle difference are meet threshold within a 0.6 second window. The case 2) and 3) are implemented by using a countdown thread in the background. While the countdown thread is running, the application will be only checking the fall condition for another end.

Fall Detected

If any of these three cases occurs, the application will notify the headset by streaming out a character "f" through Bluetooth. Meanwhile, a fall confirming thread will be started and running for 5 seconds. Within the 5 seconds window, the fall can be cancelled by pressing the physical button on the headset or pressing "Cancel" on the main screen. In the case that a fall has been cancelled from the headset, the application will stop the fall confirming thread and resume data receiving; otherwise, the application will notify the headset to cancel the alarm by out streaming a character "c", then stop fall confirming thread. If the fall is not cancelled within the window, the application will acquire the current GPS coordinate and append it as a Google Map link to the distress message then send off the distress message to the emergency contact.

Real-Time Plotting

The graph activity will be opened by choosing "Graph" from the title bar menu. The graph function is useful for fall pattern study and simulation by showing body acceleration and tilt angle change in real-time. This function is implemented by using a free open source Android library called "GraphView". The graph generates 4 plots in real-time on two Y-axis in a sample time of 0.2 second. The acceleration magnitude of phone and headset are plot in dotted line on primary Y-axis. The angle changes of Pitch and Roll are plot on the secondary Y-axis. Each data point values can be shown by tapping on the plots. Plotting can be paused or resumed by clicking on the toggle button on the bottom left. Through our testing, we have found that the average battery life is around 2 hours on a 2500 mAh battery powered android phone.


**Figure 15 Main Screen**

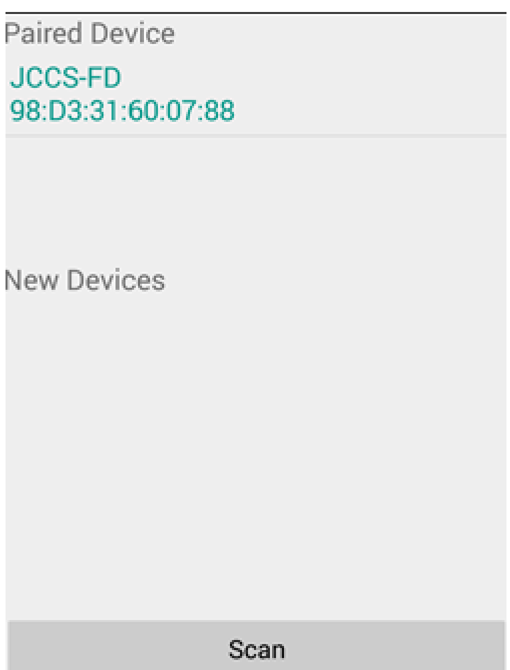


**Figure 16 Bluetooth Screen**

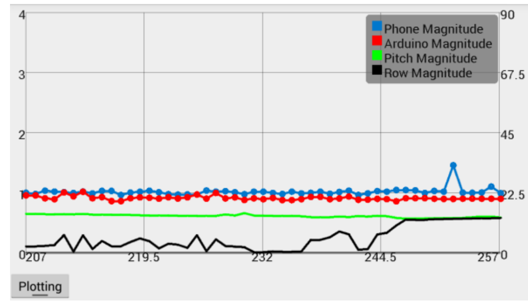


**Figure 17 Graph Screen (Landscape orientation)**

ii Pedometer Feature

The pedometer feature is implemented in the android application developed for the fall detection system. Taking advantage of new adds-on step detector sensor in android Kitkat 4.4 API, this pedometer feature gives the most accurate step count for androids and requires a low-power usage. The pedometer has 4 main functions including step counting, distance measuring, calorie calculation and data logging.

<u>Step Counting</u>

It counts total steps the user has walked and how many steps the user walks every day. It counts the total number of steps given that the phone is turned on. A daily step counting function is achieved by setting an alarm cycle for every 24 hours to reset the steps to zero through an alarm manager method.

<u>Distance Measuring</u>

The distance is calculated by the user's step length multiplied by the total steps detected within that day.

<u>Calorie Calculation</u>

Calories burnt are calculated by the product of the user input weight and distance measured, which then is multiplied by an exercise level factor (based on the units chosen). User preferences such as weight and step length are saved in the app by shared preferences.

## Daily Logging

Additionally, SQLite data base is used to save, update or delete any daily log which the user can used to configure. For example, a click on the ‘add’ button will log the user’s daily activity data into the list view, while a selection on a record in the list view deletes that piece of record. A graph view is provided as visual illustration of user’s activity trending.

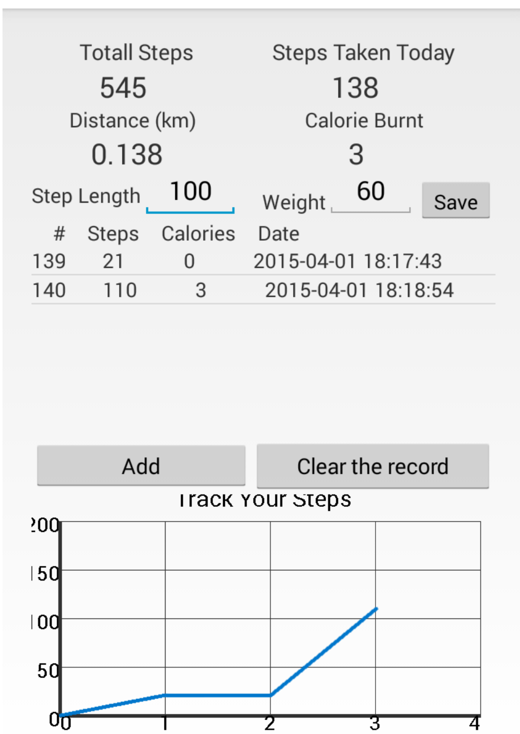


Figure 18 Pedometer Screen

Table 2: Test Object - Sebastien

| Sebastien (180 lb, 6 feet) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Normal Activties** | Bend down | Leaping | Sitting | Walking | Sneezing | Jogging | Spinning | **Sum** | **Accuracy** |
| Experiments | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 35 | 97.14 % |
| False Alarm | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| | | | | | | | | | |
| **Falls** | Face down | Sideways | Fall Back | Missed a chair | **Sum** | **Accuracy** | | | |
| Experiments | 5 | 5 | 5 | 5 | 20 | 80.00 % | | | |
| Detected | 5 | 4 | 4 | 3 | 16 | | | | |

Table 3 Test Object - Jingnan

| Jingnan (165 lb, 6 feet) | | | | | | |
|---|---|---|---|---|---|---|
| **Normal Activities** | Experiments | False Alarm | | **Falls** | Experiments | Detected |
| Bend down | 5 | 2 | | Face down | 5 | 4 |
| Leaping | 5 | 0 | | Sideways | 5 | 4 |
| Sitting | 5 | 0 | | Fall Back | 5 | 4 |
| Walking | 5 | 0 | | Missed a chair | 5 | 3 |
| Sneezing | 5 | 0 | | **Sum** | 20 | 15 |
| Jogging | 5 | 0 | | **Accuracy** | 75.00% | |
| Spinning | 5 | 0 | | | | |
| **Sum** | 35 | 2 | | | | |
| **Accuracy** | 94.29% | | | | | |

We performed test cases on Sebastien and Jingnan (See Table 2 and Table 3), who have different weight. We find out that the weight does affect the test results and accuracy. The test object that have more weight will have more accurate results with the threshold set earlier.

## V.SUSTAINABILITY AND SOCIAL IMPACT

There are social, economic and environmental implications that need to be considered when looking at technology for the elderly population. In the case of the wearable fall detection device, the main goal is to help prevent the loss of life and injuries. As a result this device will have to be worn by the user which will change the user’s lifestyle since the device will have to be on them at all times. Whether it is an ear piece, headset or necklace, the physical implementation of the device must be assessed. In addition, the ergonomics of our device would have to be studied over a period of time to insure that the product does not develop any long term injuries. Our goal for this product was for the user to wear it around the ear and the user might find discomfort in the device. Other than that, no major health issue would ever come from our device.

The economical benefit from this device comparing to other products on the market is incomparable, as it will be a low cost, one time investment. Whereas other products on the market take monthly payments for the services that they provide. Since both the phone and the device will be running on batteries, the only environmental concern is the consumption of batteries for the devices. Since smartphones in this day and age are rechargeable, this eliminates the need to purchase batteries. Therefore, rechargeable batteries will be bought for the wearable fall detection device to reduce environmental waste.

## VI. TEAM ORGANIZATION

The process from design to implementation took more than just deadlines. At the very beginning of our project we outline the major components that were required to meet our goal, and the components were hardware implementation and software/application implementation. The group was divided into their respective roles and shown in Table 2 illustrates the roles which each member took.

**Table 4 Table of Tasks Distribution**

| Tasks | Assigned to |
|---|---|
| Hardware Assembly | Sebastien Woo |
| Experiments and Data Collection | All group members |
| Arduino Programming | Jingnan Chen<br>Sebastien woo |
| Android Application Development | Cheng Zhou<br>Chao Zhang |

In order to hypothetically 'keep the ball rolling' certain mythologies were needed to help manage the effectiveness of the team. A model that some of us were familiar with was something called WIGs. WIGs stands for widely important goals, it is a model created by Sean Covey in his book called 'The 4 Disciplines of Execution'. It is a method for each individual stakeholder to create goals to meet, which solves the issues related to his/her area of interest. This model made the direction of the end result clear and it is a systematic approach to the problem. What made the model interesting was that everyone was held accountable for their work. A scoreboard was in placed and each person acts as the judge to the success for each individual. Therefore scheduling was made easy by implementing this model which led to our progress. A constant progression is always better than a stalemate because it is through our weekly goals which create momentum in the right direction for the following week. One things leads to next and this was the aim for WIGs. Everybody in the group was engaged in the weekly meetings discussing their finding and issues that they found. Deadlines were met almost every time and part of our success was due to our WIGs.

## VII.CONCLUSION

Throughout our research, analysis and implementation of the sudden fall detection device there is a clear understanding of the importance of such a device in our world today. Our aim of reducing the fatal and non-fatal injuries caused by sudden falls led to our design of a wearable fall detection devices with the added use of a smart phone application. Our final product is a combination of many months' worth of work and the end result was a functional prototype that worked exactly how we had wanted it to. Our deliverables ranged from creating a headset to detect a fall, to integrating a smartphone application which made fall detection more accurate and allowed for various features such as a daily logger, pedometer, calorie counter, real-time graph and most importantly the capability of sending out distress messages. As a result of our work, we had achieved each item from our deliverables, and with our accomplishment we have come to realize the potential for our product to grow by adding additional features. However, throughout our progress certain approaches in our development phase, we wished we would have done differently. One of the things that we should have done from the start was to have the app designers to work in parallel instead of having them working independently from each other. The problem that we ran into as a result was 2 separate applications that needed to be merged into one. Therefore the complication from merging the apps together could have all been avoided if it was development simultaneously on the same platform. Another approach that we wished to have done differently was to complete the development of the headset sooner to give more time for the app designers to test. The headset was the foundation of the project and therefore lots of time and effort was devoted towards it. Better scheduling and firm deadlines would have led to a larger buffer for other components of the project to develop sooner. Despite all of these shortcomings, the end results was a working prototype that met all of our requirements and deliverables.

There is a clear and viable market for the fall detection device and it is not just limited to the elderly. It had come to our attention from a professor at the department of Civil Engineering that our device was in fact a much desired technology in the field of construction and mining. Integrating our technology into the hardhats of miners and construction workers can help benefit their safety when a fall has occurred. For example a common injury among miners is a falling rock on a miner's head which incapacitates them leaving them unconscious. Therefore the application for such technology can be applied to various fields. Our future direction for our product is to make it a more compact and modular systems. We want it to be compact so that the user is not bothered by the presence of the device and a system that is modular so that it can be user configurable. Modularity for our device will allow the user to wear the device around their neck, wrist, head or around the ear. Commercializing our product to the masses will feature a better battery life, more health monitoring features on our phone application, and a better user interface with all of this being in a compact device. Having said all of this, our future incorporation will just be the next stepping stone for our product. The next generation of fall detection device may feature more user interface and better monitoring gadgets, but at the end of the day it is saving lives. Whether it is someone's grandma, grandpa, mom or dad, they can be insured that the fall detection device will be there to rescue them at a moment's notice.

VIII. REFERENCE

[1] , , "Falls Among Older Adults: An Overview". RetrievedJanuary , 2015 Available: http://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html

[2] , , "Injury prevention and control: Data and Statistics". RetrievedJanuary , 2015 Available: http://www.cdc.gov/injury/wisqars/

[3] , , "Philips Lifeline". RetrievedJanuary , 2015 Available: http://www.lifeline.ca/content/english

[4] Paul, M , "MobileHelp", [Online]. Available: http://medical-alert-systems-review.toptenreviews.com/mobilehelp-review.html

[5] , , "Automatic Fall Detection For Seniors", [Online]. Available: http://www.medicalalertadvice.com/fall-detection.php

[6] , , "Fall Prevention". RetrievedJanuary , 2015 Available: http://www.healthinaging.org/aging-and-health-a-to-z/topic:falls/

[7] , , "Arduino with HC-05 (ZS-040) Bluetooth module – AT MODE". RetrievedJanuary , 2015 Available: http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-at-mode/

[8] Aatish Bhatia, "Sensing your orientation: how to use an accelerometer", Jan 16, 2013, Available: http://physics.rutgers.edu/~aatish/teach/srr/workshop3.pdf

[9] "Using quaternions, how do I determine if an object is upside down from an original rotation?", Oct 2, 2013, Available: http://gamedev.stackexchange.com/questions/63000/using-quaternions-how-do-i-determine-if-an-object-is-upside-down-from-an-origin

[10] "MPU-6050 Redux: DMP Data Fusion vs. Complementary Filter", June 30, 2014, Available: http://www.geekmomprojects.com/mpu-6050-redux-dmp-data-fusion-vs-complementary-filter/

[11] "Android Kitkat - Low power sensor", Retrieved January,2015 Available: https://developer.android.com/about/versions/kitkat.html