ELDERLY HOMES HEALTH TRACKING SYSTEM

By

Aishee Mondal

Jeep Kaewla

Sanjana Pingali

Final Report for ECE 445, Senior Design, Spring 2023

TA: Akshat Sanghvi

3 May 2023

Project No. 6

Abstract

This paper presents the motivation, design, outcomes, and cost for the product created, a centralized health monitoring device for Elderly Homes. This product ensures immediate assistance in the case of irregular heartbeats, irregular blood oxygen levels, or a fall in the form of a notification on the web application. It also aims to continuously measure the person's heart rate, blood oxygen levels, and location, displayed on a web application. This product was created by integrating a MAX30102 (pulse oximeter), MPU6050 (Inertial measurement unit), NEO-6M (GPS), and a beeper to sound an alert.

The final product was able to estimate heart rate with an accuracy above 90%, the blood oxygen was able to predict accuracy within 2%, and the GPS module predicted distance location within 20 m of the actual location. It can detect a fall with 83.33% accuracy and display all the above on the web application.

Contents

1. Introduction 1
2 Design 2
2.1 Sensor Subsystems 2
2.1.1 Inertial Measurement Unit 2
2.1.2 Heart Rate and Blood Oxygen Sensor 4
2.1.3 Global Positioning System (GPS) 4
2.2 Power Subsystem 5
2.3 Control Subsystem 7
2.4 Database and Backend Subsystem 8
2.5 User Interface Subsystem 8
3. Design Verification 10
3.1 Sensor Subsystem 10
3.1.1 Inertial Measurement Unit 10
3.1.2 Heart Rate and Blood Oxygen Sensor 11
3.1.3 GPS Sensor 12
3.2 Power Subsystem 12
3.3 Control Subsystem 13
3.4 Backend and Database Subsystem 13
3.5 User Interface 14
3.5.1 Web Application 14
3.5.2 Beeper 15
4. Costs 16
4.1 Parts 16
4.2 Labor 16
5. Conclusion 18
5.1 Accomplishments 18
5.2 Uncertainties 18
5.3 Ethical considerations 18
5.4 Future work 18

References 19
Appendix A Requirement and Verification Table 20

1. Introduction

The project is motivated by the current lack of smart health monitoring and emergency response devices that are intuitive for elderly people to use. The target use case for this device is mainly Elderly Homes, Retirement Centers, and Assisted Living Centers. Mainly in these locations, the other smart health monitoring and emergency response devices on the market are not cost-effective and customized to the elderly person's needs with too many additional features that are not useful. In addition, it would be greatly beneficial to monitor multiple elderly at once; most smart monitoring devices lack this centralization aspect. The emergency response mechanism on most current devices of this nature is usually connected to one phone and can be monitored only by one person. We bring in our project to bridge this gap in the market.

Our device is a cost-effective and innovative solution to this problem, estimated at around $56.73 per node. It offers continuous centralized monitoring of multiple devices using a web application that can be used in Elderly Homes, Retirement Centers, and Assisted Living Facilities. The project also focuses on the design's intuitive use and optimum wearability with a box on the belt and a wristband for heart rate and blood oxygen. The elderly person just needs to put on the belt and the wristband and then switch on the device, which is very intuitive and uncomplicated. The fast assistance of our emergency response is covered in two ways, with a beeper on the device itself to attract attention and a notification popping up on the web application. Our high-level requirements are outlined keeping in mind the solution. The first high level requirement is that this device can monitor heart rate, blood oxygen level, GPS location, and fall detection of the person and enables the staff to view these results through the web application within 90 seconds ± 30 seconds of measurement. The next high level requirement is that notification is sent to the web application within 90 seconds ± 30 seconds when an irregularity is observed for a specific elderly person. These irregularities include the following: The heart rate is less than 40 BPM or greater than 150 BPM, the blood oxygen level is less than 96%, or a fall is detected, and the person is immobile for more than 30 seconds. The last high-level requirement is that the beeper on the belt emits an alert to attract immediate attention within 30 seconds ± 10 seconds of an irregularity (as defined above in the first high-level requirement) being detected.

The sections ahead describe the design and tests that have been simulated with results that verify the requirements have been met, along with the costs outlined. Overall, the requirements have been met, and we have a fully functional minimum viable product on a Printed Circuit Board. The heart rate, blood oxygen, location of the person, and the fall detection algorithm are working as expected and are being continuously updated on the web application according to the person. The notification system also works as expected. The microcontroller detects an irregular heart rate, blood oxygen, or a fall, and a notification appears on the web application. The beeper also emits an alarm whenever an irregularity is detected. The project also keeps track of ethical considerations in its design and implementation, keeping in mind that it needs to be worn for an extended period.

2 Design

In this section, we will discuss our design choices during this project. This project has five subsystems: Sensor Subsystem, Power Subsystem, Control Subsystem, Database and Backend subsystem, and User Interface Subsystem.

The health data collected by the sensors are sent from the microcontroller to the database over Wi-Fi. The web application displays data from the database. Any irregularity triggers a notification on the web app, and the beeper emits an alarm.

2.1 Sensor Subsystems

Our project has three primary sensor modules that allow us to collect the necessary health and location data. These sensor modules include the inertial measurement unit, heart rate and blood oxygen, and GPS modules.

2.1.1 Inertial Measurement Unit

The inertial measurement unit serves to collect the data we need for the fall detection algorithm. There are three inputs to the fall detection algorithm: root mean square acceleration, roll (orientation angle around the X-axis), and pitch (orientation angle around the Y-axis). In this project, we selected the MPU6050 chip as it has a built-in digital motion process (DMP), which facilitates the roll and pitch angle calculation.

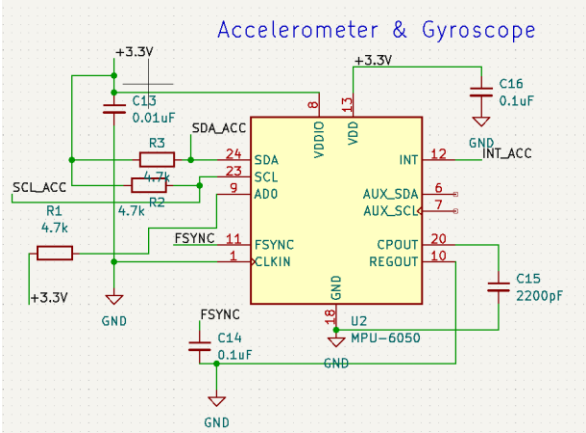

Figure 1. The schematic of the Inertial Measurement Unit

2.1.1.1 Sensor Placement

We decided to place MPU6050 on the belt wearable system as this increases the accuracy of the fall detection algorithm, as shown in the graph in Figure 2. This also ensures that the wearable wrist system

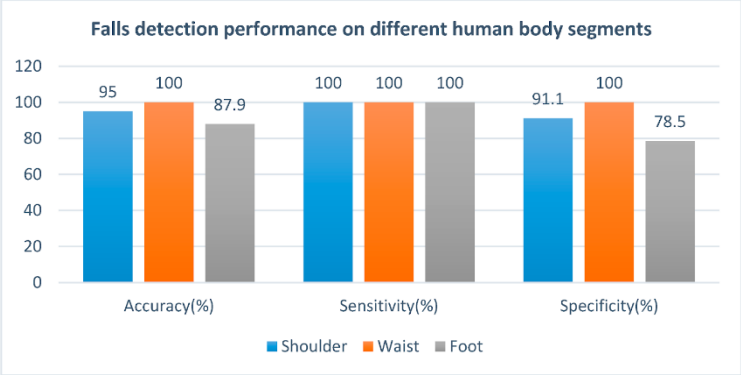

Figure 2. Fall detection accuracy, sensitivity, and specificity performance when placing the IMU on different human body parts [1].

will be compact.

2.1.1.2 Obtaining Fall Detection Inputs

The first input for the fall detection algorithm is the root mean square acceleration. MPU6050 outputs the acceleration in the X, Y, and Z axis. The equation to calculate the root mean square acceleration using the output from the MPU6050 is

$$A_{rms} = \sqrt{a_x{}^2 + a_y{}^2 + a_z{}^2} \tag{1}$$

Where a_x is the acceleration in the x-axis, a_y is the acceleration in the y-axis, and a_z is the acceleration in the z-axis.

The remaining two inputs are the pitch and roll angle. There are two ways to obtain the roll and pitch angle. The first way is to use a complementary filter. The formulas to calculate the orientation angles using the complementary filter are

$$Pitch = arc\tan 2(a_y, a_x) \tag{2}$$

$$Roll = arc\tan 2(-a_x, \sqrt{a^2{}_y + a_z{}^2}) \tag{3}$$

where a_x is the acceleration in the x-axis, a_y is the acceleration in the y-axis, and a_z is the acceleration in the z-axis. The second approach is to calculate the orientation angles by converting the quaternion, a way to represent a body in a 3-dimensional space, obtained from the digital motion processor to the roll and pitch angle. We decided to use the second approach because the digital motion processor provides a high-frequency orientation with lower latency than the complementary filter. DMP is also more accurate and abstracts the quaternion to Euler's angle conversion, allowing us to focus on our fall detection algorithm [2].

2.1.1.3 Fall Detection Algorithm

There are three phases in the fall detection algorithm, which are the impact, posture, and immobility phases [1]. The impact phase occurs when there is a sudden change in acceleration greater than 1.5 times gravity. The 1.5 times gravity threshold is set by taking an average of the measured acceleration from our falling experiments. Within two seconds, there should be a sudden change in the orientation greater than 50 degrees; this is the posture phase. The posture phase enables us to differentiate high acceleration activities, such as jumping and running, from a fall. Following the posture phase is the immobility phase. This is when the fallen person remains immobile for at least 30 seconds. The purpose of the immobility phase is to consider the possibility that the fallen person might be able to get up on their own. In this case, extra assistance is not required.

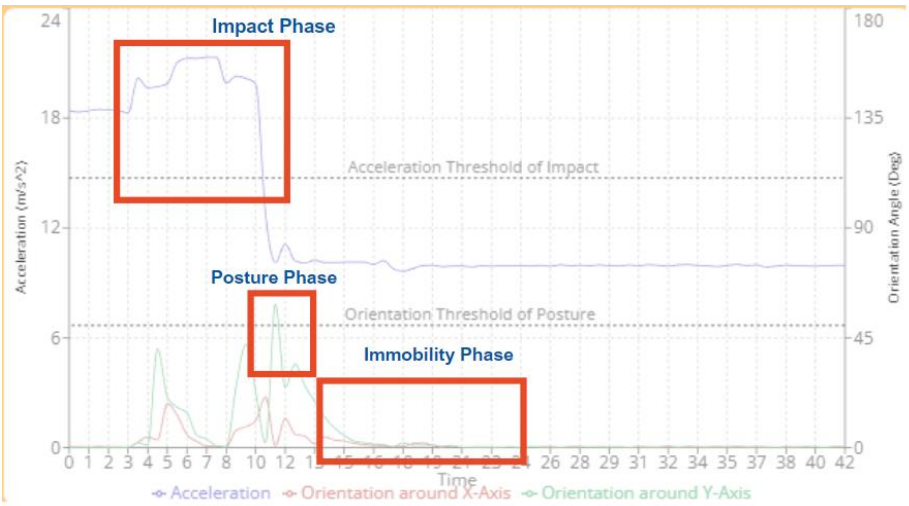

Figure 3. The three phases of the fall detection algorithm

2.1.2 Heart Rate and Blood Oxygen Sensor

2.1.2.1 Module Description

The heart rate and blood oxygen sensor used is MAX30102. The MAX30102 communicates with the microcontroller through the Inter-Integrated Circuit (I2C) communication protocol with a default address of 0x68. This chip contains two Light-Emitting Diodes (LEDs), an Infrared LED and a Red LED [3].

2.1.2.2 Sensor Placement

This chip contains a glass encasing with two LEDs and a photodetector that needs to be in direct contact with the skin. While most of our components are placed in a box on the belt, this MAX30102 chip is embedded in a wristband and connected to the main circuit board in the box. This bracelet is made of webbing to ensure that most aspects of the circuit are covered with a small hole exposed for the photodetector and LEDs. The wrist was selected as an ideal point for sensor placement as the skin around the wrist is thin and convenient for elderly people to wear without too much hindrance.

2.1.2.3 Data Collection

The amount of light reflected back from the two LEDs onto a photodetector can determine the amount of oxygen in the blood. The amount of light reflected back from the Infrared LED determines the amount of Infrared radiation absorbed by blood and, therefore, how oxygenated the blood is [3]. The Infrared radiation had to be above a certain threshold (Infrared value of 10,000) to detect a heartbeat. The heart rate can be determined based on the time interval between these peaks in Infrared values.

The blood oxygen measurement requires Infrared as well as red light absorbed values. This is because deoxygenated blood absorbs radiation from the visible range (red) of the Ultraviolet (UV) spectrum and oxygenated blood from the infrared range [3]. As a result, these values determine the extent to which the blood is oxygenated and deoxygenated, which are then compared to determine blood oxygen levels.

2.1.3 Global Positioning System (GPS)

2.1.3.1 Module Description

The GPS sensor consists of the NEO6MV2 chip and a sensitive ceramic patch antenna with a sensitivity of -161dbM [4]. The GPS sensor is able to pinpoint the location of the person using latitudes and

longitudes. The antenna is able to track location using 22 satellites over 50 channels and enables the sensor to work in all outdoor and some indoor locations. In order to work in the best possible conditions, the path antenna needs to be parallel horizontally to the NEO6Mv2 chip. The operating voltage for this sensor is 2.7-3.6 V, and according to our design, it receives a voltage supply of 3.3 V, which meets the operational requirements [4]. The antenna is connected to the chip using an Ultra-small coaxial connector. The sensor module is then connected to the printed circuit board using 01x04 female connectors and jumper wires.

2.1.3.2 Data Collection

The GPS sensor is connected to an ESP32 microcontroller using the universal asynchronous receiver-transmitter (UART) protocol to read the encoded latitude and longitude. The GPS sensor connects to the microcontroller using the transmit-data(TXD1) and the receive-data(RXD1) pins over UART communication with a 9600 bits/second sampling rate. The latitude and longitude values obtained by reading the data are then converted to an address using an Application Programming Interface (API), which is then used to display the person's current location on our web application.

Figure 4. The web application shows the current address of the person rendered using GPS.

2.2 Power Subsystem

The power subsystem consists of a 5 V rechargeable battery and a 3.3 V linear regulator. As this product was intended to be used by the elderly daily, a rechargeable battery would allow them to charge it without any external assistance.

Below, the circuit schematic can be seen in Figure 5.

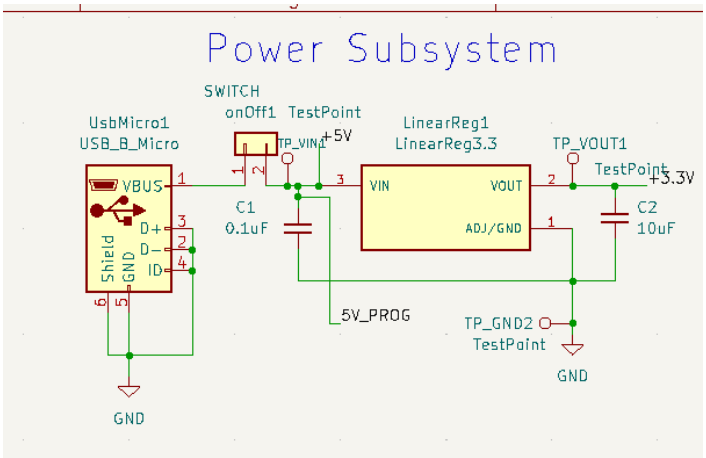

Figure 5. Schematic for Power Subsystem

The MPU6050, NEO-6M, ESP32, and the beeper have a maximum voltage rating of around 3.7 V. Hence, a linear voltage regulator of 3.3 V is used to supply the constant voltage of 3.3 V to these modules. A circuit depicting the current requirements of these modules is shown below in Figure 6.

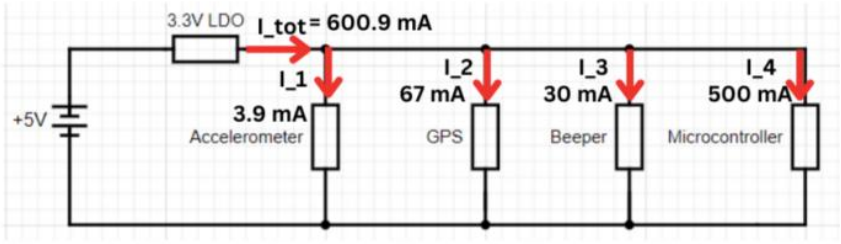

Figure 6. Circuit depicting the printed circuit board and current flowing through the modules.

The total current required by the printed circuit board is 600.9 mA, as depicted above in Figure 6. According to the datasheet for the 3.3 V regulator [5], a minimum drop-out voltage of 1.2 V is needed to ensure the regulator functions as expected. This would entail a minimum voltage of 4.5 V for the battery. Hence, a battery of 5 V was chosen.

The MAX30102 has a maximum voltage rating of 5 V. The module consists of two regulators: a 3.3 V regulator to supply both the LEDs with 3.3 V and a 1.8 V for the rest of the sensor. This led to a supply of 5 V directly from the battery.

The regulator chosen was of packaging type TO-220, as seen in the datasheet [5]. This packaging type allows for a minimum increase in temperature for each watt of power (50 degrees Celsius/Watt) as compared to other packing styles for this regulator. This can be seen in the calculation shown below:

IMU: At 3.3 V, for resistance of 10 $k\Omega$, current supplied, $I_1 = 3.9$ mA
GPS At 3.3 V, current supplied, $I_2 = 67$ mA
Beeper At 3.3 V, current supplied, $I_3 = 30$ mA
Microcontroller: At 3.3 V, current supplied, $I_4 = 500$ mA

Adding up all the current, we get total current from the voltage regulator at 3.3 V, $I_{total} = I_1 + I_2 + I_3 + I_4$
$= 3.9 + 30 + 500 + 67$ mA $= 600.9$ mA
Voltage drop, $V = V_{in} - V_{out} = 5$ - 3.3 V = 1.7 V
Power consumed, P $= V * I_{total} = 1.7 * 600.9 * 10^{-3}$ $W = 1.02153$ W
Temperature change, $\Delta T = P * \theta_{Ja} = 50 * 1.02153 = 51.0765$ degree Celsius.

If we add the temperature difference to the room temperature, max temperature achieved = 25 degree C + 51.0765 degree C = **76.0765 degree Celsius.**

Since this is well within the maximum temperature of 125 degree C of the voltage regulator, the circuit is safe for operation within the wearable device of the belt and the voltage regulator will not blow up.

Figure 7. Calculation of temperature increase of system due to the linear regulator.

2.3 Control Subsystem

The microcontroller used in the printed circuit board is the ESP32-WROOM-32E with an integrated Wi-Fi module. This microcontroller with the integrated Wi-Fi module is very useful for us to send the collected data over to our database in MongoDB using POST requests over a Hypertext Transfer Protocol (HTTP) connection. The microcontroller has to be programmed by putting the ESP32 in boot mode. This is achieved by driving the IO0 pin on the ESP32 low and then the Enable pin on the ESP32 low, which is equivalent to switching on the ESP32 in boot mode.

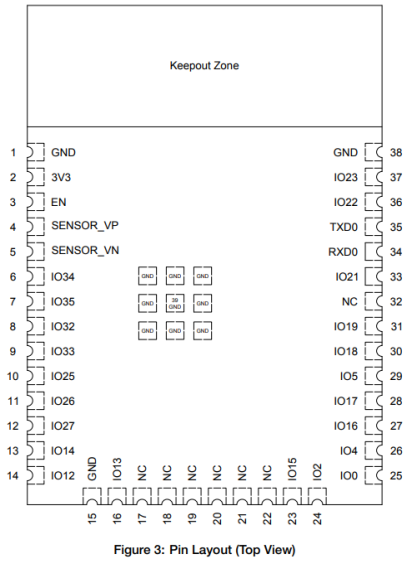

Figure 8. Pinout Representation of the ESP32-WROOM-32E [6]

The programming of the ESP32 is accomplished using button logic implemented on our PCB. The Boot button connects to the IO0 button and is active low, enabling the IO0 pin to receive a low signal when

pressed. The Enable button connects to the Enable pin of the microcontroller and is also active low, which drives the Enable pin low when the button is pressed.

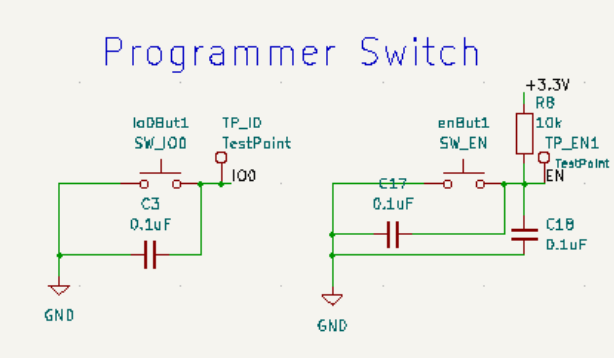

Figure 9. Schematic showing the design of the programming of the ESP32 using buttons.

2.3.1 Integrated Wi-Fi Module

The Wi-Fi module on the ESP32 enables us to connect to a Wi-Fi network and establish an HTTP connection with an HTTP client. In this case, our HTTP client is the backend API running on a server. Our microcontroller collects all the data from the sensors. It serializes it into one JavaScript Object Notification (JSON) body, which is then used to send a POST request to the backend server and update the database accordingly with new entries.

2.4 Database and Backend Subsystem

The primary function of this subsystem is to send the data from our database to the client that requested it and save health data to be stored in our database for later access.

2.4.1 Database

We chose to use MongoDB to store all of our health data as it is faster, more scalable and gives more flexibility to our data structure compared to MySQL.

2.4.2 Backend

The backend that we designed supported two endpoints: the GET and POST request endpoints. The frontend web application uses the GET request to obtain the health data from MongoDB to be displayed on the user interface. The GET endpoint supports WHERE, LIMIT, SORT, and SELECT operations. The microcontroller, on the other hand, uses POST requests to store health data in MongoDB.

2.5 User Interface Subsystem

The User Interface Subsystem consists of two aspects: the web application and the beeper.

2.5.1 Web Application

The web application displays the current heart rate and blood oxygen readings for different individuals on a centralized interface.

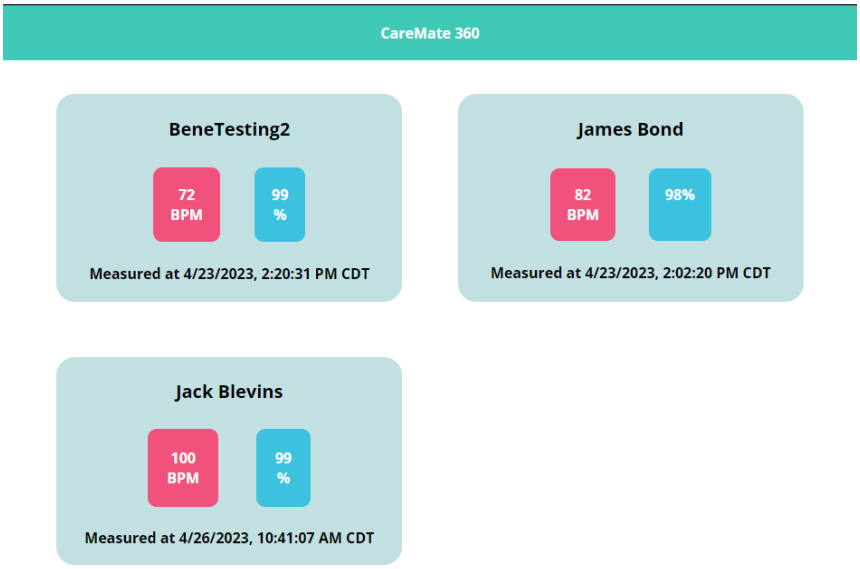

Figure 10. The interface of all the individuals being tracked.

Clicking on the widget of a specific individual gives health trends and location data as well as the timestamps of when the measurements were taken, as seen below in Figure 11.

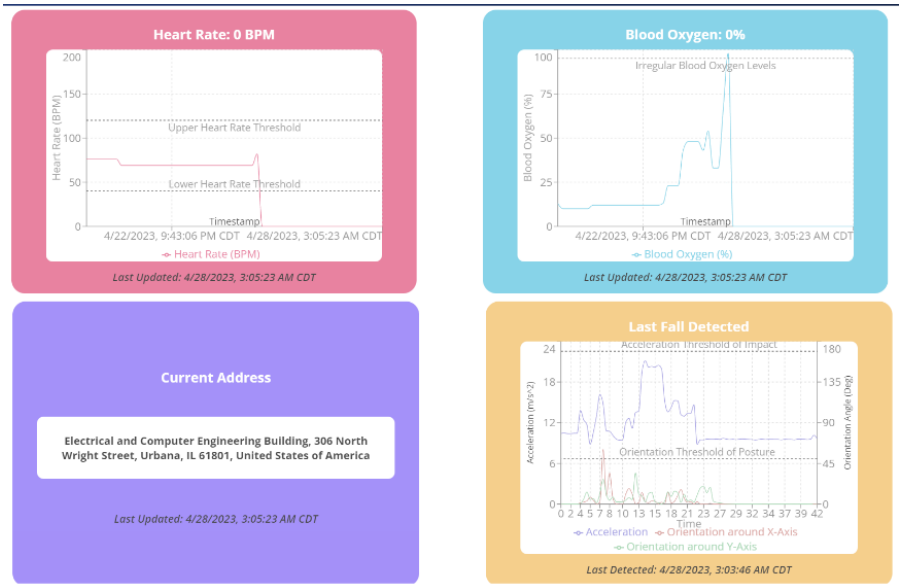

Figure 11. Interface depicting an individual's vitals being measured.

2.5.2 Beeper

The beeper is connected to the microcontroller, which drives the pin high when an irregularity has occurred. It has a minimum loudness of 85 decibels at the supplied voltage of 3.3 V to attract the attention of those around in the case of an emergency [7].

3. Design Verification

In this section, we will describe the requirement and the verification obtained from simulating tests to fulfill these requirements. We will go over each subsystem and the specific requirements that apply to that subsystem. A comprehensive list of these requirements and verification is also included at the end of the document in Appendix A.

3.1 Sensor Subsystem

We will describe the requirements and verifications obtained from testing each sensor module in detail. In particular, we have three sensor modules - the IMU, Heart Rate and Blood Oxygen Sensor, and the GPS.

3.1.1 Inertial Measurement Unit

The requirement we set for the inertial measurement unit is being able to detect a fall with 70% accuracy. To test this requirement, we simulated a fall 12 times, falling forward, backward, left, and right three times each. The testing data are shown in Table 1. We achieved an accuracy of 83.33%, as shown by correctly detecting falls in 10 out of 12 trials, which fulfilled our subsystem requirements. Table 1 shows the results of our fall detection trials, where F stands for forward, B for backward, L for left, and R for right.

Table 1. The results from the fall detection trials.

Trials	1	2	3	4	5	6	7	8	9	10	11	12
Fall Direction	F	F	F	B	B	B	L	L	L	R	R	R
Fall Detection	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes

Our fall detection algorithm can also differentiate a fall from other high-acceleration activities, such as a jump or a run. As shown in Figure 12 below, when we try to simulate a jump, the orientation threshold of posture is never crossed. Consequently, our fall detection algorithm does not mistake these high-acceleration activities as a fall, which decreases the likelihood of a false positive in our algorithm.

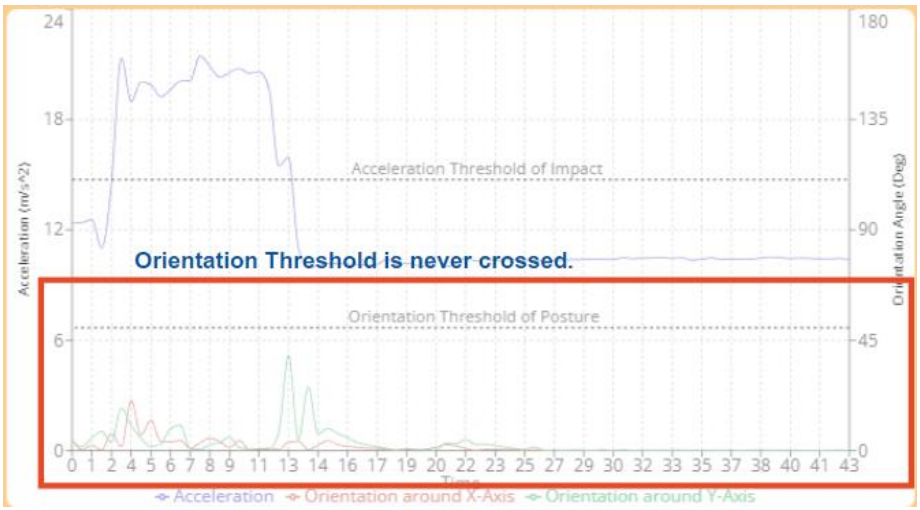

Figure 12. The output graph of the jumping activity.

3.1.2 Heart Rate and Blood Oxygen Sensor

3.1.2.1 Heart Rate Sensor

The requirement for the heart rate sensor was that the device would measure heart rate in Beats Per Minute (BPM) with 90% accuracy compared to the heart rate (BPM) measured by an oximeter.

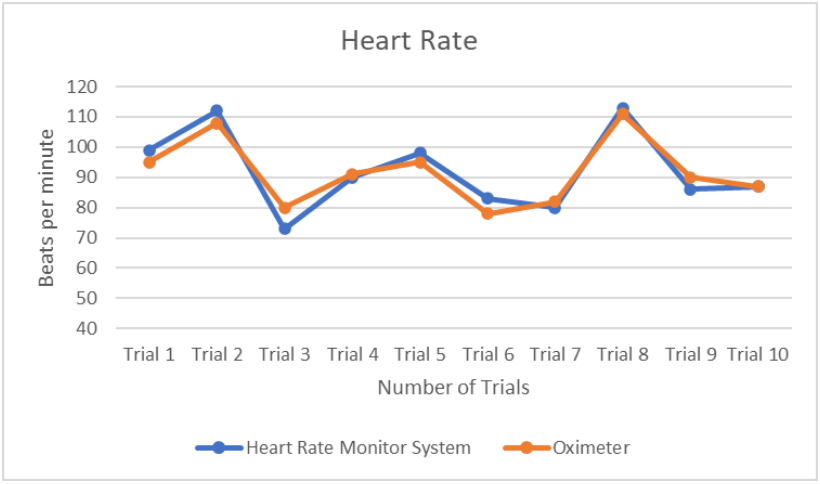

Figure 13. The verification of the Heart Rate requirement with ten trials

As seen on the graph, ten trials were taken, and the readings were compared to those taken by the oximeter, and Table 2 below shows the percentage error for each trial. We can see that the error percentage is less than ten for every trial.

Table 2. The results from the fall detection trials.

Trials	1	2	3	4	5	6	7	8	9	10
Error (%)	4.211	3.704	8.75	1.099	3.158	6.410	2.439	1.802	4.444	0

3.1.2.1 Blood Oxygen Sensor

The requirement for the Blood Oxygen sensor was that the device should measure blood oxygen level within 2% accuracy of the blood oxygen level measured by the oximeter.

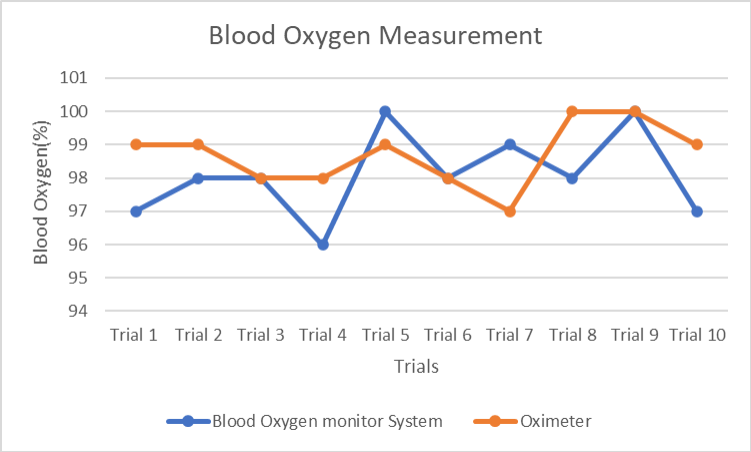

Figure 14. The verification of the Blood Oxygen requirement with ten trials

3.1.3 GPS Sensor

The requirement for the GPS sensor was that it should be able to measure the actual location of the place within 20 m of what is measured by Google Maps. We calculate the latitude and longitude of the location as given by Google Maps and find the distance between the location given by Google Maps and the one measured by the GPS sensor. This data is plotted on the graph, as shown in Figure 15.

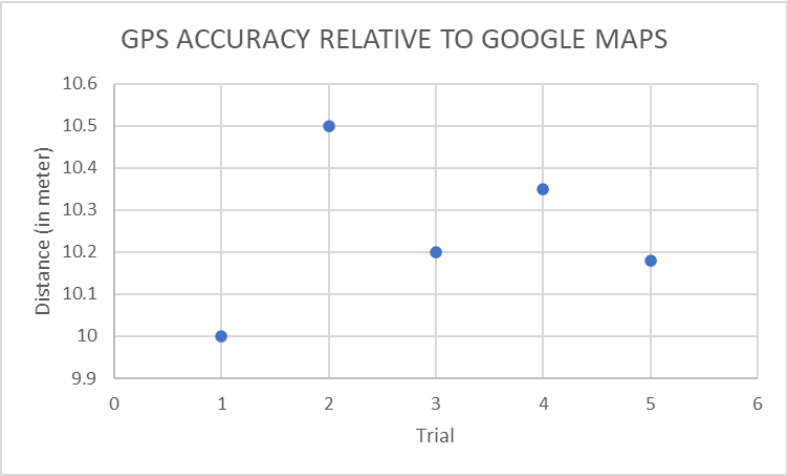

Figure 15. The distance between the location on Google Maps and the one measured by the GPS sensor.

3.2 Power Subsystem

The requirement for the power subsystem was to ensure that 3.3 V was provided to the sensor modules, as mentioned in the Design section, after the voltage was output from the linear regulator. This was verified by probing the output voltage of the linear regulator, as seen in Figure 16.

Figure 16. Verification of the Voltage drop to 3.3 V

3.3 Control Subsystem

The requirement here is that the control subsystem can collect all the data and serialize it into JSON, making it easy for us to send POST requests over Wi-Fi to our backend server, which is an HTTP client. As shown in Figure 17 below, the entry has been updated in our database using a POST request from the microcontroller, and all the fields have been filled out with the correct format as specified.


```
timestamp: 2023-04-28T04:49:50.963+00:00
heartRate: 93
bloodOxygen: 98
acceleration: 9.8213452
▾ pr: Array
▾ notification: Array
    0: 0
    1: 0
    2: 0
gpsLat: "40.114667"
gpsLong: "-88.227480"
▸ historyIMU: Array
createdAt: 2023-04-28T04:49:50.964+00:00
__v: 0
```


Figure 17. The output of the newest entry updated into the MongoDB database

3.4 Backend and Database Subsystem

The requirements for this subsystem are to support GET and POST requests from clients.

To verify that the backend supports GET requests, we chose a random entry from the database and tried to retrieve the same document using its id. We successfully retrieved the same document, as shown in Figure 18.

Figure 18. The verification of the GET endpoint.

Similarly, to verify that the backend supports POST requests, we sent a POST request to store a new document in our database. We then confirmed that the same document was saved, as shown in Figure 19 below.

Figure 19. The verification of the POST endpoint.

3.5 User Interface

3.5.1 Web Application

The web application requirement was to ensure that an individual's health data would be updated within 90 ± 30 seconds of measurement. Since the web application updates the timestamp along with the measured sensor data, this result is verifiable by observing how fast the time stamp is updated.

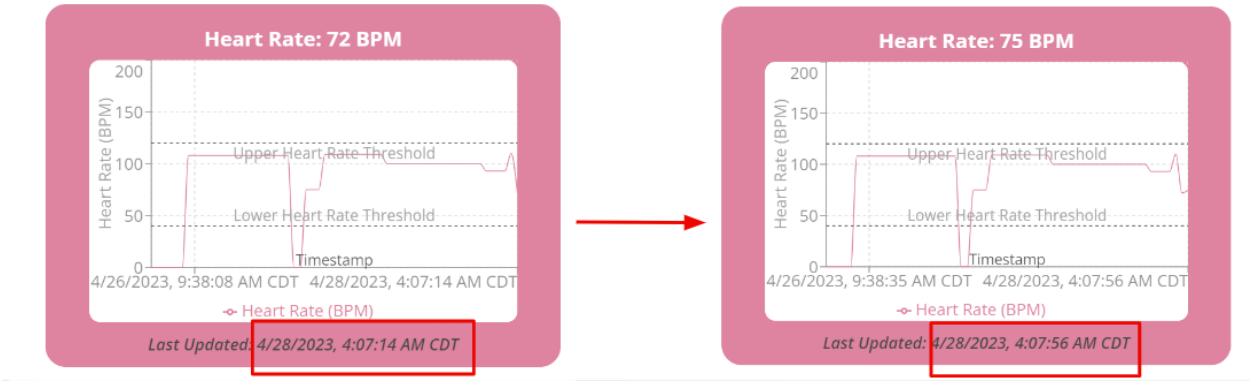

Figure 20. The duration between two consecutive timestamp updates on the web application

As seen in Figure 20, the timestamp was updated in 42 seconds, meeting the requirement of updating the timestamp within 90 ± 30 seconds.

3.5.2 Beeper

The requirement for the beeper was to emit an alert within 30 ± 10 seconds of an irregularity occurring.

This requirement was verified by simulating an irregularity and checking the time it took for the beeper to emit an alarm after this irregularity was detected. There were ten trials taken, and a graph with the results of these trials can be seen below in Figure 21.

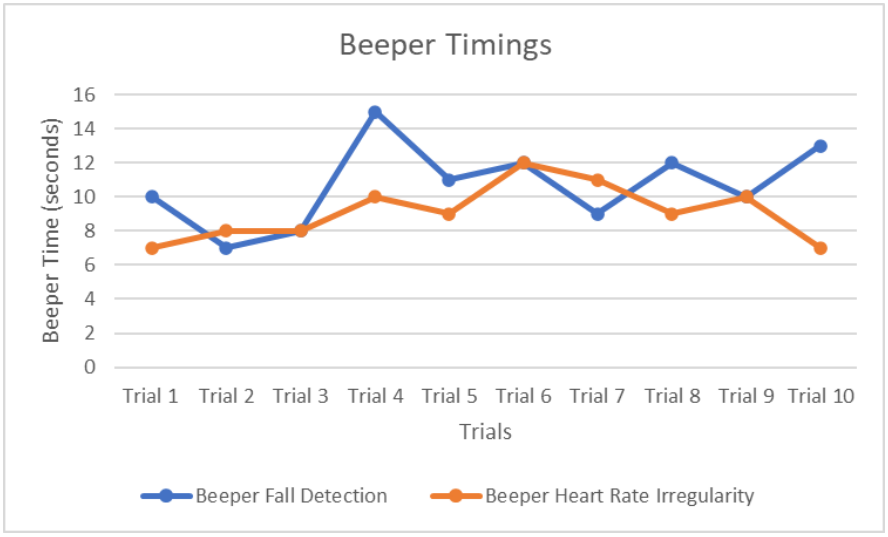

Figure 21. A graph of trials demonstrating the time it had taken the beeper to emit an alert

4. Costs

4.1 Parts

Table 3, shown below, contains our costs for all parts bought in the development and used in the final product. It also mentions the manufacturer, the cost for each part, and the total costs incurred over the semester.

Table 3 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
ESP32-Devkit-32E	Mouser	10.00	2	20.00
ESP32-WROOM-32E	DigiKey	2.7	2	5.4
NEOGY6MV2	Amazon	8.99	2	17.98
Module MAX30102	Amazon	2.996	3	8.99
MPU6050	Mouser	8	2	16
GY-521 MPU6050	Amazon	3.296	3	9.89
RA11131121 Switch	DigiKey	0.6	2	1.2
LD1117V33	DigiKey	0.76	3	2.28
USB Battery Pack - 2200mAh Capacity	Adafruit	14.95	1	14.95
Micro-USB-B USB2.0	DigiKey	1.17	2	2.34
WST-1203UX Beeper	DigiKey	1.46	2	2.92
Tactile buttons	DigiKey	0.81	4	3.24
01 x 05 Female Conn	DigiKey	0.47	3	1.41
01 x 04 Female Conn	DigiKey	0.47	2	0.94
4.7 kΩ Resistors	DigiKey	0.1	10	1
10 µF Capacitor	DigiKey	0.44	4	1.76
0.01 µF Capacitor	DigiKey	0.1	10	1
1 µF Capacitor	DigiKey	0.11	4	0.44
2200 pF Capacitor	DigiKey	0.35	4	1.4
Total				**113.14**

4.2 Labor

Typically, as a Computer Engineering graduate from ECE at Illinois, the hourly rate of the Engineer would be $50/hour. Since all three of us are Computer Engineering majors, we estimate the Cost per hour for each member would be $50/hour. We then estimate that each of us would be working 25 hours per week for the 11 weeks of the semester as defined in Table 4 on the next page.

Table 4 Labor Costs

Labor Cost	Hours per week	Total number of Hours (11 weeks)	Cost ($)/hour	Multiplier	Total Cost over 11 weeks
One member	25	275	50	2.5	34,375
Three members	75	825	50	2.5	103,125

5. Conclusion

5.1 Accomplishments

The main accomplishment was that our group satisfied all our high level requirements that were initially set out. The first high level requirement was measuring vitals (heart rate, blood oxygen) as well as detecting the location and the most recent falls, and updating the web application with this data within 90 30 seconds. The next high level requirement was displaying the notification on the web application within 90 30 seconds of the irregularity occurring. The last high level requirement was that the beeper would emit a sound within 30 ± 10 seconds of an irregularity occurring.

Another accomplishment was being able to integrate the sensors and microcontrollers along with a web server and database and allowing them to communicate wirelessly with each other. A major accomplishment also included working with and making a fully functional PCB, with limited prior hardware experience.

5.2 Uncertainties

An uncertainty that our group had encountered while testing our printed circuit board was with loose wiring connecting the MAX30102 sensor module to the main printed circuit board. Connectors were used to ensure the connections were stable, however, our group still encountered problems with loose wiring when powering up the board. This would only be fixed by pressing the wires down against the connectors while powering the board.

5.3 Ethical considerations

Due to the fact that our project must be worn by an elderly individual during all times of the day and since the heart rate and blood oxygen sensor module was in contact with the individual's skin, our group made sure to follow IEEE code 7.8.II.9, which states that the device must not harm the individual [8].

Our project follows this ethics code by ensuring that the box encases the circuit board and the entire power subsystem and that there are no dangling wires outside the box. Additionally, although the heart rate and blood oxygen module does have to be in contact with a person's skin, webbing was used to cover a majority of the circuit so it does not come into contact with the person's skin and only a small hole was created so that the photodetector and LEDs within the sensor would be in contact with the person's wrist.

5.4 Future work

After building and testing the project, there were some aspects that we would consider changing for future work. The first aspect would be making the box and its connections more robust and secure, as g we realized there might be an issue damaging the box, when a person falls.

Another aspect we believe would be more effective is additional data analysis added to the web application. Data analysis such as average daily, weekly, hourly heart rate and blood oxygen readings would be more useful in understanding a person's underlying health conditions.

Finally, integrating machine learning models could improve accuracy of fall detection. By analyzing previous fall detection data trends, it is possible to be able to improve the prediction of when a fall occurs. We could also use Machine Learning and health data to understand different underlying health trends for a particular person, and provide recommendations through the web app.

References

[1] A. Mao, X. Ma, Y. He, and J. Luo, "Highly portable, sensor-based system for Human Fall Monitoring," MDPI, 13-Sep-2017. [Online]. Available at: https://www.mdpi.com/1424-8220/17/9/2096. Accessed May 2023.

[2] D. Debra, "MPU-6050 Redux: DMP Data Fusion vs. complementary filter," Geek Mom Projects, 19-Nov-2015. [Online]. Available at: https://www.geekmomprojects.com/mpu-6050-redux-dmp-data-fusion-vs-complementary-filter/Accessed May 2023.

[3] Last Minute Engineers, "Interfacing MAX30102 pulse oximeter and heart rate sensor with Arduino," Last Minute Engineers, 06-Feb-2022. [Online]. Available at: https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/#:~:text=The%20MAX30102%20works%20by%20shining,through%20light%20is%20called%20Photoplethysmogram. Accessed: May 2023.

[4] Last Minute Engineers, "In-depth: Interface ublox NEO-6M GPS module with Arduino," Last Minute Engineers, 26-Jun-2022. [Online]. Available at: https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/. Accessed: May 2023.

[5] "Adjustable and fixed low drop positive voltage regulator LD1117." [Online]. Available at: https://www.st.com/content/ccc/resource/technical/document/datasheet/99/3b/7d/91/91/51/4b/be/CD00000544.pdf/files/CD00000544.pdf/jcr:content/translations/en.CD00000544.pdf. Accessed: May 2023.

[6] "Espressif ESP32-WROOM-32E datasheet." [Online]. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf. Accessed: May 2023.

[7] "WST-1203UX: Digi-key electronics," *Digi*. [Online]. Available at: https://www.digikey.com/en/products/detail/soberton-inc/WST-1203UX/1245301. Accessed: May 2023.

[8] "IEEE Code of Ethics", IEEE. [Online]. Available at: https://www.ieee.org/about/corporate/governance/p7-8.html Accessed: May 2023.

Appendix A Requirement and Verification Table

Table 5 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Detect if a person has fallen with 70% accuracy.	1. Pretends to fall on a soft surface (falls forward, right, left, and backward directions three times each) and remain immobile for 30 seconds. See if a notification pops up on the web application. The beeper should also emit a sound.	Y
2. Provide a heart rate measurement with 90% accuracy compared to an Oximeter	2. Compare the current heart rate displayed on the web application to that of the oximeter and see if the two measurements are within 90% accuracy.	Y
3. Provide an accurate ± 2% oxygen level compared to an Oximeter	3. Compare the current blood oxygen displayed on the web application to that of the oximeter and see if the two measurements are within 2% accuracy.	Y
4. Provide the location of where the person is within 20 m of the actual location	4. Collect the latitude and longitude values from the GPS. Check the actual latitude and longitude using Google Maps. Calculate the distance between the two.	Y
5. To provide 3.3V ± 0.2V to the wearable belt subsystem	5. Use a multimeter to measure the voltage output of the LDO by placing one probe on the voltage regulator output pin and the other probe on the ground test point.	Y
6. Process all sensor data in JSON format and send it to the MongoDB database	6. The MongoDB database should have a new entry with all the fields with values in the correct format.	Y
7. The backend subsystem should support GET requests	7. Pick a random entry to be retrieved by id and verify that the same document is returned after sending a GET request by id.	Y
8. The backend subsystem should support POST requests	8. Make a POST request to store a new entry to MongoDB. Verify if the same entry is added to the database.	Y
9. Display sensor data on the web application within 90 seconds ± 30	9. Compare the two consecutive timestamps displayed on the front	Y

seconds after measurements.	end.	
10. Emit an alarm with a latency of 30 seconds ± 10 seconds when an irregularity occurs.	10. When an irregularity is observed, the beeper should emit the alarm within the latency of 30 seconds ± 10 seconds.	Y
11. Provide the location of where the person is within 20 m of the actual location	11. A notification should pop up on the web application within the latency of 90 seconds ± 30 seconds.	Y