A Thesis Report on

WEARABLE FALL DETECTION SYSTEM

Submitted to

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY IN ELECTRONICS & COMMUNICATION ENGINEERING

By

B. NARAYANA MURTHY 18P11A4014

Under the Esteemed Guidance of

DR.A.PULLA REDDY SIR M .Tech, Ph.D.

Associate Professor

Department of ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE (AUTONOMOUS)

(Accredited by NAAC with ‘A’ Grade, Approved by AICTE, New Delhi, Affiliated to JNTU Anantapur)

Renigunta road, Tirupati-517506, Andhra Pradesh, India

2018-2022

CHADALAWADA RAMANAMMA ENGINEERING COLLEGE

(AUTONOMOUS)

Department of Electronics & Communication Engineering

Certificate

This is to certify that the project work entitled **"wearable fall detection system"** is a bonafide work done by **B.NARAYANA MURTHY (18P11A0414)** in the Department of **"ELECTRONICS & COMMUNICATION ENGINEERING"**, and submitted to ***Chadalawada Ramanamma Engineering College (Autonomous), Tirupati*** is a project work carried out by them under my guidance during the academic year 2020-2021.

GUIDE	HEAD
DR .A .PULLA REDDY SIR M.Tech,Ph.d	**Dr.P.KRISHNAMURTHY**
Associate Professor.	
Department Of ECE	Associate Professr &
	Head of the department of

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We are hereby declare that the project work on **“WEARABLE FALL DETECTOR”** done by us under the guidance of DR.PULLA REDDY SIR**,** **M.Tech,Ph.D** in **CHADALAWADA RAMANAMMA ENGINEERING COLLEGE (Autonomous)** is submitted in partial fulfillment of the requirements of the requirements of the award of Degree of Bachelor of Technology.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that, we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our project report submission.

Signature of the Students

B.NARAYANA MURTHY **18P11A0414**

ACKNOWLEDGEMENT

A grateful thanks to **Dr. CHADALAWADA KRISHNAMURTHY**, Chairman, C.V.S. Krishna Murthy Theja Charities for providing education in this esteemed Institution.

We express our gratitude to our principal **Dr. P. SANJEEVA RAYUDU**, **M.Tech, Ph.D.** Principal, Chadalawada Ramanamma Engineering College, Tirupati, for providing all the facilities and supporting us to complete this project work.

Our sincere thanks **to Dr. P. KRISHNA MURTHY**, **M.Tech, Ph.D.** Associate Professor and Head of the Department of Electronics & Communication Engineering, Chadalawada Ramanamma Engineering College, Tirupati, for providing infrastructural facilities to complete our project work.

We are greatly indebted to our guide, **DR.PULLA REDDY SIR**, **M.Tech,Ph.d.** Associate Professor 1 of Electronics & Communication Engineering, Chadalawada Ramanamma Engineering College, Tirupati, who encourages us in all spheres of activities, constant encouragement and keen interest enriched to complete this work.

We extend our thanks to all the Teaching staff of the Department of E.C.E. for their support, encouragement and inspired us. We also thank the Non-Teaching staff of the Department of E.C.E. for being helpful in many ways in successful completion of our work.

The chain of our gratitude would be definitely incomplete, if we forget to thank all my friends of Chadalawada Ramanamma Engineering College, Tirupati for their constant support and encouragement. Finally, we thank all those who helped us directly or indirectly in successful completion of our project work.

Contents

List of Figures

List of Tables

Abstract

As we all know the fall detector system is very useful for older people. This is because it can notify the individual or family member when it detects a fall and reduces the risk of delay in medical attention. So it leads to the development of various types of automatic fall detector systems. Nowadays, we can also find fall detectors in smartwatches, fitness trackers, and other types of wearables. IoT-based fall detector devices can save lives in an emergency. For this reasons I choosen to build an IoT Fall Detector Using MPU6050, NodeMCU ESP8266, and Blynk Application.

Acknowledgements

The completion of this project was made possible by Professor John McNeill, PhD for providing the project with a direction and a focus as well as technical expertise, Professor Ulkuhan Guler, PhD for providing great technical expertise, and William Appleyard for his help in obtaining the resources needed for the construction of our project.

Additionally, we would like to thank the NECAMSID lab as well as Analog Devices and Allegro Microsystems for helping sponsor our project. Furthermore, we would like to thank Michael Coln at Analog Devices for taking the time to meet with us via Skype several times as well as providing us with valuable advice and his technical expertise. Lastly, we would like to thank PhD student Sulin Li for providing us with several soldering techniques that were crucial to the success of our project.

Executive Summary

To culminate one's education at Worcester Polytechnic Institute (WPI), students are required to complete the Major Qualifying Project (MQP), a capstone, in their respective majors. Though some teams begin their project with a solidified idea, our project was introduced to us with the general description of "analog applications". With no concrete direction on where to begin, we began a process of brainstorming possible ideas for our MQP. Ten possible project ideas were brainstormed and they were: an EEG, a continuation of the skin impedance spectrometer MQP, an ADC design, an application demo for chips, a continuation of the smart shirt MQP, a fall detection system, heart rate monitoring, DUI prevention, a posture monitoring system, and a LIDAR crash prevention system.

After brainstorming ten possible project ideas and conducting preliminary research on each, we needed to critically examine each idea and decide which ones had a significant concentration of analog material and was the most feasible to complete in three terms. To help guide the decision process, we developed ten decision criteria that we felt our project should include. The ten criteria were as follows: relevant and new to industry, solves an interesting and important problem, want it to work and function on time, improves skills and technical growth, interesting, leads to a paper at a conference or a journal, creates a cool 60 second video, leads to a business or patent, leads to future work (MS or MQP), and wins the outstanding MQP award.

With these criteria being used as guidelines, we then used various decision methods to narrow down the potential MQP choices. The first decision method we used was a decision matrix to generate a numbered score for each idea. Using the decision criteria, we created weights, scored each project idea in the criteria categories, and calculated the final scores for each project idea. There were distinct number separations for all ideas except for the fall detection system, posture monitoring system, and skin impedance spectrometer which scored 65, 65, and 47 respectively. As a result, the team decided to combine the fall detection system and posture monitoring system into one device, not only due to its application in the health field, but also for the various applications the device could target.

Using the fall detection and posture monitoring system as the idea for our MQP, we proceeded to develop several functions the device could include. From this second brainstorming period, we developed several applications the device could target. These include the elderly, hospital patients, active individuals, people under the influence, parents and children, students, and office workers. This system is composed of an inertial measurement unit (IMU) and will alert the user if they have poor posture and will also alert a caretaker if the individual wearing the system has fallen.

Once we determined what the functions of the device would be, the next step was choosing the components that would satisfy each of the device's functions. We used a microprocessor to control the entire system. The microprocessor needed to be compact, inexpensive, lightweight, easy to use, and have enough memory to control all the system requirements. Initially we chose the TI CC2650MODA, a microprocessor that met most of the specifications. However, as we progressed through the system integration, difficulties programming the SPI and Bluetooth communication became prominent. There were little resources online to help with the troubleshooting process. As a result, we switched to the ATMEGA328-PU microcontroller.

For fall detection and posture monitoring, the team wanted a device that had the capability of measuring the position of a human body in space and in real time. Initially we explored accelerometers and gyroscopes separately, but after researching IMU's, decided to focus research there. We compared individual accelerometers and gyroscopes to IMU systems and decided to choose the LSM9DS0, an IMU, sold through Adafruit.

With numerous system functions it was crucial that we found a power source that not only had the ability to support all of the features, but also was lightweight enough to be carried on the waist of a human body. We explored four battery options and decided on the 3.7V Lithium Polymer Ion Battery, as it was relatively small, lightweight, and had a capacity of 500mAh.

To control the system, we decided to use Bluetooth Low Energy (BLE) along with a compatible phone application. The first Bluetooth module we researched was the HC-06 which is compatible with Arduino. The second module we researched was the RedBear BLE Shield which is also compatible with Arduino, however, it comes in the form of a shield. Due to the very similar specifications, we chose the HC-06 module as it was a lower cost and would save space on the future PCB with the system components.

After choosing the necessary components, we proceeded to the construction, testing, and debugging stages of our project. Here we constructed the physical systems for each of the devices functions and wrote the necessary code to control each of these functions. The accelerometer records data in m/s2 and alerts the user and caregivers when their motions increases outside the threshold of 6.9 m/s2 to 10.2 m/s2 . If the threshold is met, the phone application will display the message “Fall Detected,” sending a text message to the caretaker with the individual’s location. Additionally, the data can be visualized through the application or a Python script, allowing for future data manipulation.

For posture tracking, the IMU continuously saves the values of the x, y, and z coordinates relative to its position in space. The gyroscope records data in degrees per second (dps) and alerts the user and when their posture leaves the threshold of -4 dps to 4 dps. If the threshold is met, the phone application will display the message “Stand Up Straight” to alert the user that they need to adjust their posture. These values can then be displayed on a graph and the data can further be analyzed and manipulated. The overall system works as expected, and we believe that with more refining, the project has the ability to become an on-market product.

Our project comprised of four phases, as outlined in Figure 1. Between August 2018 and October 2018, Phase 1 and Phase 2 were completed, and Phase 3 was initiated. Phase 3 was completed between October 2018 and December 2018. Phase 4 was initiated in January 2019 and will be completed in March 2019.

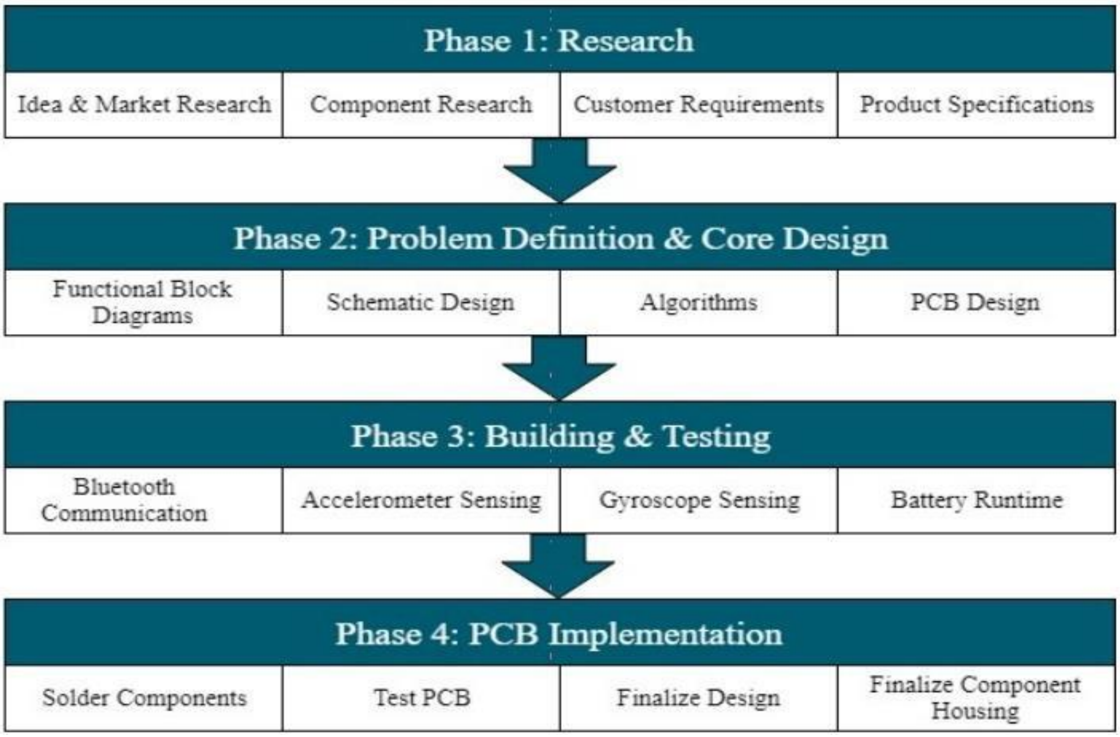


Figure 1: Project flow chart

Chapter 1

Introduction

The topic of our Major Qualifying Project (MQP) was analog applications. Within this discipline, we were able to brainstorm our own ideas based off evaluation criteria that we determined. Our team consisted of three individuals whom all had varying backgrounds within our majors of Electrical and Computer Engineering. Throughout the first few weeks of the project, we began to propose various ideas to one another and soon, two stood out.

Two different problems amongst the health industry in the United States are that people are susceptible to falls as well as poor posture. While these are two separate issues, we decided as a team that one device could be the solution to both of these problems. The leading cause of death in the geriatric population, according to the Center for Disease Control, is falling [1]. These tragedies could be prevented through postural corrections that have been shown to improve balance. Equilibrium and balance have shown to directly decrease with degeneration of a person’s posture [1]. While the elder population would be a main target for our device, fall detection and imperfect posture are widespread issues that can be applied to a wide variety of individuals.

Per our advisor’s suggestion, we began to research various target populations for our device other than the geriatric population. Each year, three million people are treated in emergency departments for fall injuries [2]. Additionally, 90 percent of the United States’ population lean forward with their necks, which often gets worse when working on a computer, driving or watching television in bed [3]. To aid in this research, we added several different use cases for our device other than the elderly.

We decided on a fall detection and posture monitoring device that would utilize two sensors to build an overall smart system. The initial goal of our team was to develop a working, wireless system, in which various sensors were able to transmit data to a phone application via Bluetooth. The first sensor would be an accelerometer used to monitor the acceleration of the user and the second would be a gyroscope used to measure the posture of the individual. To accomplish this goal, we designed, implemented, and tested our small device to ensure that it properly detected a fall or a person’s poor posture and sent a notification to the phone application via Bluetooth.

Chapter 2

Brainstorm

2.1 Proposed Projects

After deciding that the project would target the healthcare industry, we began to propose several ideas to each other based off our own interests. As a result, a list of ten project ideas were brainstormed and then discussed. We then narrowed the initial project ideas down to three solid ideas which can be categorized into the following:

◦ Fall Detection System

◦ wearable fall detector

2.1.1 Fall Detection System

All around the world, falls are a major public health problem. Around 646,000 fatal falls occur each year, making it the second leading cause of unintentional injury and death. Death rates due to falls are the highest among adults over the age of 60 years [4]. In 2015, the total medical costs for falls totaled more than $50 billion while Medicare and Medicaid covered around 75% of these costs [2]. In addition, the implementation of effective fall prevention strategies would cause a 20% reduction of falls among children ten and under. This would result in a net savings of over $120 million dollars each year [4].

One of the project ideas was to design a device that monitors whether a person has fallen. If a fall is detected, loved ones, caregivers, or medical personnel, would be notified and if the person did not respond within a certain amount of time, help would be sent to them. This system would use an accelerometer and a microprocessor to interpret the data. The data would then be sent to a mobile application in order to display the data and send notifications. There are several fall detection monitors in the market, however, they are expensive and some require the use of a landline or the push of a button. Some of these competitors can be seen in Table 2.1. While these devices are similar, they are expensive and require a monthly fee after purchase.

Table 2.1: Fall detection device competitors

GoSafe [5]	GoLiveClip [6]	Sense4Care [7]
		
$99.95 (device fee) $44.95 per month	$299 (device fee) $14.95 per month for app	$174
Wearable pendant Fall detection capabilities Autoalert capabilities Requires a landline	Clips onto clothing Fall detection capabilities Pairs with a smartphone Auto-alert capabilities	Clips onto clothing Fall detection capabilities Pairs with a smartphone Auto-alert capabilities

Chapter 3

Decision Methods

3.1 Evaluation Criteria

We prepared a list of criteria to determine which of the ideas were most suitable for our project. This allowed us to conduct an analysis of each idea and compare them with each other. Below is each criteria we used for analysis.

Relevant and New to Industry

One of the most important criteria we considered when evaluating ideas was that they had to be new and relevant to the industry. We wanted to find topics within an industry that were creating a lot of 'buzz' because they were new and innovative ideas. While a particular topic may be relevant to an industry, it may also be too saturated. We did not want to work on a project that had been completed several times before in the industry, since it would not have any room for improvement.

Solves an Interesting and Important Problem

One criteria that we considered most important was that the project would solve an interesting and important problem. We wanted our project to be meaningful and beneficial to society. While the project had to be interesting to each member, it also had to solve an important problem as well as aid various individuals.

Works and Functions on Time

While we wanted to complete an advanced project, it was important that we kept in mind our three academic term time frame. We defined a working and functioning project as one that could be demonstrable to an audience as well as exceeded the standards of all of the set criteria for the project. It was also important to understand the time frame since there are several components in the design including the prototyping, testing, and writing process of the project.

Improves Skills and Technical Growth

It was important to us that we work on a project where we have the flexibility to be exposed to technical aspects that we had not yet encountered. Therefore, it wasimportant for the project to be technically diverse so that each member had the opportunity to develop a variety of skills.

Interesting for Team

The project idea had to be interesting for each member as it would ensure that we were motivated throughout the year. If we were motivated and driven towards the end goal, the project would have a higher probability of being successful.

Leads to Paper at Conference or in a Journal

One of our goals was to publish a paper at a conference or in a journal. While this may not have been our most important goal, we believed that this would be a major achievement and honor.

A Cool 60 Second Video

We believed that a cool, 60 second video would provide a brief overview of the entire project. It would be able to describe our project clearly to a non-professional audience as well as attract them and make them interested in learning more about the design.

Business / Patent / Start up

Another one of our goals was to either start a business and/or apply for a patent. While this may conflict with the other goal of publishing a paper at a conference or journal, we believed it should be considered if the business evaluation of the project is positive. In addition, one member of the team was pursuing a minor in business, so this could also be an opportunity to use their skills in pursuing a patent.

Leads to Future Work (MS, MQP)

We believed another important goal was that the project would lead to future work, such as for another MQP or a Masters degree. We thought this was important as it would contribute to the WPI community and allow future students the foundation to add their own ideas by completing further research on our project.

Outstanding MQP Award

Lastly, one of our final goals for the MQP was to try and win the Outstanding MQP Award. This award would be an honor for our team as it would indicate that the idea made a significant contribution to the community. The award would also commend our three-term effort and represent all of our hard work.

3.2 Decision Matrix

To judge the ten ideas we brainstormed, we used a decision matrix. The full decision matrix can be found in Appendix A. A decision matrix is a logical approach to narrowing down project ideas. It is beneficial as it weights the choices purely based on numbers, excluding personal opinions [17]. However, this method is also disadvantageous as it may be unsuccessful at providing a clear choice since options may be close in value due to the weighted criteria.

The decision matrix used the weighted decision criteria to rank the ten brainstormed project options. We ranked each criterion as either a one, two, or three; where a one was least important, two was medium importance, and three was most important. Table 3.1 shows the decision criteria weights.

To calculate the scores, the weights of the criteria and the rankings given to each project idea were multiplied and the total was summed. The higher score indicated that the project better fit the criteria described.

Table 3.2 shows the final scores for the top three project ideas. The top three ideas were the fall detection system, the posture monitoring system, and the skin impedance spectrometer.

Table 3.2: Decision matrix scores

Project Idea	**Score**
Fall Detection System	65

3.3 Final Decision

Since both the fall detection system and the posture monitoring system tied as a result of the decision matrix and had a significantly higher score than any of the other designs, we decided to combine the two ideas and make a smart system. Both of these ideas were relevant to the industry and solved an important problem. Not many devices in the market currently connect to smartphones, and none combine the technology of an accelerometer and gyroscope to detect both falls and imperfect posture. We believed that we could make the device work and function within the time constraints and that it could be easily demonstrated working in a video. While this topic was interesting to our team, it was also believed that the construction of the device would improve the skills and technical growth of each member.

Chapter 4

Background Research & Product Specifications

4.1 Target Markets

Using the fall detection and posture monitoring system as the basis for the MQP project, we proceeded to target several markets where the device could possibly succeed. From this second brainstorming period, we developed eight target markets where both the fall detection or posture monitoring technology could be applied.

4.1.1 Fall Detection and Prevention System

In order to attract the proper market, it was important to determine who would benefit from wearing a fall detection device. While falls are most commonly associated to the elderly, we did not want the device to only focus on a single group of people. Below are the different categories of people that we believed would benefit from the device.

Elderly

One of the main target markets for the device was to target it towards the elderly, including those in a nursing home and those who live alone (Figure 4.1). This would provide urgent care to people if needed and reduce the medical costs associated with falls. If falls are not dealt with in time, they can cause functional impairment and a significant decrease in a person’s mobility, independence, and life quality [4]. In addition, some falls may be preventable and it is important to recognize when a person may be about to fall and prevent the situation from occurring.

Figure 4.1: Elderly fall detection target market

Hospital Patients

Another market would be to target hospital patients of all ages whom are under the risk of falling, so that if they got up from their room and happened to fall, hospital staff would be notified immediately (Figure 4.2). Patients who fall while in a hospital add health care costs for both the patient and the facility. Also, they will have to stay longer and use more hospital resources, taking up space and time. Hospital patient are at a higher risk of falling due to a variety of factors including muscle weakness, medications, and dizziness [20]. It is important to both detect a fall as well as to prevent them by recognizing when a person may be more susceptible to a fall.

Figure 4.2: Hospital patient fall detection target market

Active Individuals

Active individuals would be another main target for the fall detection device. These individuals could include rock climbers, hikers, and bikers who all participate in risky activities (Figure 4.3). These individuals have a high risk for falling and these falls may be life-threatening due to their environments. A fall detection system would allow them to participate in the activities that they love and give them peace of mind that if an accident were to happen, someone would be notified to provide them with help.

Figure 4.3: Active individuals fall detection target market

People Under the Influence

Another group of people would be people under the influence of drugs or alcohol as the fall detection system could sense that their walking patterns are not normal and indicate that they may be susceptible to a fall. A person who is under the influence is more likely to fall due to their visual impairment and loss of balance (Figure 4.4). Therefore, it would be beneficial for them to wear one of these devices in case they were to accidentally fall and injure themselves.

Figure 4.4: Under the influence fall detection target market

Parents and Children

Finally, last group of people that we could target with the device would be children and their parents (Figure 4.5). Many parents would have peace of mind knowing that their children could go out and play and that if they were to fall and injure themselves, they would be notified immediately.

Figure 4.5: Children fall detection target market

4.2 System Requirements

When deciding on which components will best suit our system, we first needed to decide the most important requirements for our product. We decided to create specific product requirements for each component of our product.

4.2.1 Accelerometer

Below are the system requirements for the accelerometer component in our device:

◦ Sensitivity of at least 1 mg/LSB

◦ Range of at least 4g

◦ Low cost (< $4)

◦ 8 mA worst case current usage

◦ Small size ◦ −25 ◦C to 40 ◦C temperature range

◦ I2C or SPI communication

An important requirement for the accelerometer of our system is that it must have high sensitivity so that it can precisely detect a fall and measure when a person is more susceptible to a fall. It also needs to have a suitable range so that it will be able to correctly detect a fall. The accelerometer we choose must have a low cost so it is as cost efficient as possible. In addition, we also want our accelerometer to have a low current usage so that the user would not have to recharge the battery often. Another requirement is that the accelerometer is on the smaller size so that it does not make our product too large since it will be worn by the user. Finally, the accelerometer needs to have a suitable temperature range as well as SPI or I2C communication.

4.2.2 Gyroscope

Below are the system requirements for the gyroscope component in our device:

◦ Sensitivity of at least 10 mdps/LSB

◦ Low cost (< $4)

◦ 8 mA worst case current usage

◦ Small size ◦ −25 ◦C to 40 ◦C temperature range

◦ I2C or SPI communication

An important requirement for the accelerometer of our system is that it must have high sensitivity so that it can precisely detect a change in a person's posture. It also needs to have a high output range. The gyroscope we choose must have a low cost so that the device is as cost efficient as possible. Additionally, we also want our gyroscope to have a low current usage so that the user would not have to recharge the battery often. Another requirement is that the gyroscope is on the smaller size so that it does not make our product too large since it will be worn by the user. Finally, the gyroscope needs to have a suitable temperature range as well as SPI or I2C communication.

4.2.3 Microcontroller

Below are the system requirements for the microcontroller component in our device:

◦ Low cost (< $10)

◦ 12 mA worst case current usage

◦ Small size

◦ At least 12 GPIO pins

◦ −25 ◦C to 40 ◦C temperature range

◦ I2C or SPI communication

An important requirement is that the microcontroller we choose must have a low cost so that our product is as cost efficient as possible. In addition, we also want our microcontroller to have a low current usage so that the user would not have to recharge the battery often. It would be feasible that the user could go at least a day without needing to recharge the battery. The microcontroller will use a lot of the power so we want the current usage to be as low as possible. Another requirement is that the microcontroller must be on the smaller size so that it does not make our product too large since it will be worn by the user. Additionally, our microcontroller needs to have a suitable temperature range and can be integrated with

Bluetooth. It also needs to be able to interface with SPI or I2C to accommodate both the accelerometer and gyroscope and have a sufficient number of GPIO pins.

4.2.4 Wireless Communication

Below are the system requirements for the microcontroller component in our device:

◦ Range of at least 2 meters

◦ Low cost (< $4)

◦ Integrates with phone

An important requirement for the wireless communication module is that it must be able to interface with a phone or application. An important part of our system is that the data can be sent to an app where it can be processed and viewed by the user. Another requirement is that the module must have a range of at least 2 meters. While this is a close range, we believe that the user will always have their phone close to their body.

4.2.5 Battery

Below are the system requirements for the battery component in our device:

◦ Rechargeable

◦ Weight less than 15 grams

◦ Low cost (< $10)

◦ Nominal voltage of at least 3.3V

◦ Small size

◦ Capacity of at least 500 mA/h

4.3 Component Research & Specifications

When deciding which components will best suit our system, we used the system requirements we determined in Section 4.2 to help us decide between our options. Below is the analysis of our component decisions based of the requirements we already determined.

4.3.1 Accelerometer & Gyroscope

The fall detection and posture monitoring system incorporates a sensing device to track the both the positioning and motion of the user for applications both in fall detection as well as posture correction for people in everyday life.

To provide the user with relevant information on their posture and motion, we researched the accuracy of an accelerometer, gyroscope, as well as an inertial measurement unit (IMU). An IMU incorporates an accelerometer, gyroscope, and magnetometer in one chip [22]. This is beneficial to the system as we will utilize the accelerometer and gyroscope and it will save space on the PCB and could potentially be a lower cost. We compared the price and accuracy of the three different sensors. The results of the three different component options are shown below in Table 4.1. After comparing the cost, current usage, and additional features of each sensor, we ultimately selected the LSM9DS0 IMU.

Table 4.1: Comparison of sensors

		ADXL362 (Acc) [23]	**L3GD20H (Gyro) [24]**	**LSM9DS0 (IMU) [25]**
Cost		\$3.97	\$3.42	\$6.33
Current Usage	Measurement Mode	3 μA	6.1mA	6.35 mA
	Standby Mode	10 nA	2 mA	2 mA
	Power Down Mode	—	5 μA	6 μA
Voltage Supply		1.8V to 3.3V	2.4V to 3.6V	-0.3V to 4.8V
Sensitivity	—	1 mg per LSB	—	—
	Output Range: 245 dps	—	8.75 mdps/LSB	
	Output Range: 2000 dps	—	70 mdps/LSB	
	Linear acceleration FS = ±2 g	—	—	0.061 mg/LSB
	Linear acceleration FS = ±16 g	—	—	0.732 mg/LSB
Size of Chip		12 mm^3	9 mm^3	16 mm^3
Temperature Range		−50 °C to 150 °C	−40 °C to 85 °C	

This system uses the LSM9DS0 to track the users movement as well as their posture (See Figure 4.9). The sensor combines an accelerometer, a gyroscope, and magnetometer and transmits data to the microcontroller through a digital I2C output. Additionally, this sensor satisfies our requirements of a least a 1 mg/LSB accelerometer sensitivity and 10 mdps/LSB gyroscope sensitivity. The accelerometer output range has a wide range of ±2 g to ±16 g which satisfies our requirement of at least ±4 g. The worst case current usage also satisfies our 8 mA requirement and the combined cost of both the accelerometer and gyroscope is under eight dollars. Although all of the sensor options satisfied our requirements, the LSM9DS0 combines both the accelerometer and gyroscope which will be a lower cost, lower current usage, and save us space on our PCB. Additionally, the LSM9DS0 has a higher sensitivity range for both the accelerometer and gyroscope measurements.

Figure 4.9: LSM9DS0 component

Although the sensor is capable of handling different sensitivities, the ranges that we chose can be seen in Table 4.2.

Table 4.2: Sensor characteristics

	Accelerometer	Gyroscope
Range	+/-2g	+/-245 dps

4.3.2 Microcontroller

To process the data received from the sensor, we investigated a variety of microcontrollers. The microcontroller needed to have the ability to control every peripheral used by the fall detection and posture monitoring system. The data processed by the microcontroller will be sent to the user wirelessly through Bluetooth. While some microcontrollers have built in Bluetooth modules, others can be interfaced with an external Bluetooth modules through either SPI or I2C. Additionally, we considered the microcontroller’s ease of use, the amount of instructional resources available, and the cost. Table 4.3 compares these characteristics against two different microcontrollers.

Table 4.3: Microcontroller comparisons

	CC2650MODA (Bluetooth Integrated) [26]	**ATMEGA328-PU (No Bluetooth Integrated) [27]**
Cost	\$9.90	\$3.25
Current Usage	9.4 mA (Active mode) 1 μA (Standby mode) 100 nA (Shutdown mode)	19 mA (Active mode)
Size of Chip	16.9 mm x 11 mm	35.544 mm x 7.62 mm
Interface	I2C, UART, SPI	
Temperature Range	-40 to 85 C degree	
Integrated with BLE?	Yes	No
GPIO Pins	15	Digital: 14 Analog: 8
Embedded with RTOS?	Yes	No
Supply Voltages	1.8V to 3.8V	6V to 20V

Initially, we selected the CC2650MODA for the device due to the integrated Blue tooth capabilities and low cost. However, after encountering multiple problems when trying to set up the SPI communication between the MCU and sensor, we chose to switch to the ATMEGA328-PU and found an external Bluetooth module (See Figure 4.10). While the ATMEGA328 does not satisfy our 12 mA worst case current usage and is a much larger size than the CC2650MODA, it does satisfy our 12 GPIO pin minimum requirement as well as our low cost. Even though these requirements were not met, it was important to consider our three-term time frame as well as the skillset of the team members. We decided as a team to use the ATMEGA328 as the device's microcontroller as it was easier to use and had many more instructional resources available.

Figure 4.10: ATMEGA328-PU component

4.3.3 Wireless Communication

The ATMEGA328-PU itself does not have built in Bluetooth capabilities, so we decided to use an external Bluetooth module to send data to the phone app. We explored two different Bluetooth modules, the RedBear BLE Shield v2.1 and the HC-06 module sold through Adafruit. Both were very similar, however, we chose the module over the shield due to the additional space the module made available on the PCB and the module was easier to integrate with the phone application. Table 4.4 shows the comparison between the two different Bluetooth modules.

Table 4.4: Bluetooth comparison

	RedBear BLE Shield v2.1 [28]	HC-06 [29]
Price	$19.90	$3.99
Dimensions	53.44mm × 62.17mm × 11.17mm	27mm × 13mm × 2mm
Voltage	5V	3.6V to 6V

The final product uses the Arduino ATMEGA328-PU to process the data and the HC-06 module to transmit the data wirelessly and the data can be visualized from an Android App (See Figure 4.11). The Bluetooth module meets the requirement of a two meter range and is inexpensive compared to the RedBear BLE Shield v2.1.

Figure 4.11: HC-06 module

4.3.4 Battery

An important part of the fall detection and posture monitoring system was the power source. We needed a power supply which would not only be able to power the whole system, but also keep it running for a reasonable amount of time. It was also important that the battery was lightweight so that the user is able to carry it on their back for a long period of time. Finally, the battery needed to be at a low cost to keep the overall of price of our system low. Table 4.5 displays the comparison between the different batteries that we researched.

Table 4.5: Battery comparisons

	3.7V Lithium Polymer Ion [30]	**RJD2450 [31]**	**Panasonic NiMh AAA [32]**	**ML2430 [33]**
Current Capacity	500mAh	200mAh	800mAh	100mAh
Cost	$7.95	$13.18	$2.81	$3.82
Weight	10.5g	6.5g	2.9g	4.1g
Nominal Voltage	3.7V	3.7V	1.2V (for 1)	3V

Based on the above specifications we selected the 3.7V Lithium Polymer Ion battery to power the system due to its nominal voltage, light weight, and low cost (See Figure 4.12). The cost is under the $10 requirement as well as the weight is below 15 grams. The battery is rechargeable and its nominal voltage is greater than 3.3V.

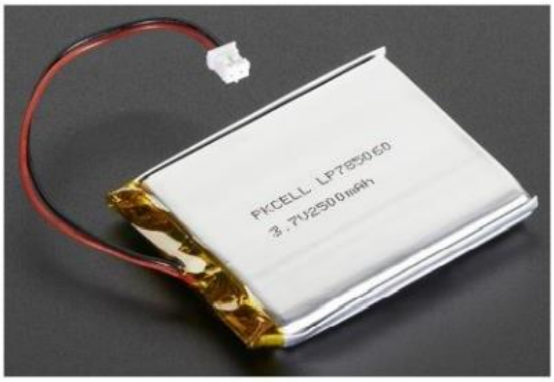


Figure 4.12: 3.7V Polymer Ion battery component

After selecting our battery and system components, it was important that we conducted a power analysis to determine the runtime of the overall system. To start, we calculated the total current draw from all of the components in our system (See Table 4.6)

Table 4.6: Total current usage of components in the system

Component	**Current Usage (Worst Case)**
Sensor	6.35 mA
Microcontroller	19mA
Bluetooth Module	8mA
Total	33.35mA

Next we divided the current capacity of the battery by the total current usage of the system as seen in Equation 4.1.

$$Runtime = \frac{Battery\ Capacity}{Total\ Current\ Usage} \times \frac{1}{PowerBoost\ 500\ Efficiency} \tag{4.1}$$

The results indicated that with the worst case current usage, the battery will last up to roughly 14 hours which is less than a day. However, we agreed that the average person is only awake for 14 to 18 hours a day. Therefore, a runtime of 14.39 hours will provide the user with a one days use. To calculate the overall time to charge the battery once it has completely died, we will divide the battery capacity by the charge rate current in Equation 4.2. The standard charge rate current of a micro-usb port is 500mA.

$$Charge\ Time = \frac{Battery\ Capacity}{Charge\ Rate\ Current} \tag{4.2}$$

The maximum time to charge the battery was calculated to be 1 hour and with a 20% efficiency lost, the total charge time would be around 1.2 hours. This is a very short charge time and would not inconvenience the user if they needed to charge the device during the day.

Chapter 5

Financial Analysis

A key, non-technical aspect of our project was the financial analysis. The Electrical and Computer Engineering Department at WPI requires students to complete an engineering design course. One of the outcomes of this course is to be able to demonstrate a working knowledge of financial, scheduling, and administrative elements of the design process. In this course, students create and adhere to project schedules, as well as estimate possible Return-of-Investment (ROI) if their product were to be manufactured [34]. We decided to utilize the material learned in this course and apply it to our fall detection and posture monitoring system. This chapter will look at the bill of materials for bulk production, initial investment required costs, as well as the ROI analysis.

5.1 Bill of Materials

A bill of materials (BOM) is a comprehensive inventory of the raw materials, assemblies, subassemblies, parts and components, as well as the quantities needed to manufacture a product [35]. To summarize, it is the complete list of items that are required to build a product. The bill of materials for the production of a single device is shown in Table 5.1, where costs are assumed for bulk production (>4000pcs) [36].

5.2 Initial Investment

An initial investment would be made if we wanted to take the product into market. The numbers seen in Table 5.2. were extracted from a lecture by Professor W. Michalson in ECE 2799. Manufacturing costs are typically divided into three categories. The first category is direct materials which is the cost of the materials that become part of the finished product [37]. For example, a direct material for our product would be the FR-4 used to make the PCB and soldering lead. The second category is direct labor which is the cost of individual's wages who are physically involved in converting raw materials into the finished product [37]. For example, the wages of the people operating the PCB Assembly machine would be considered direct labor for our system. The third category is the factory overhead or manufacturing overhead which is all of the other costs incurred in the manufacturing activity which cannot be directly traced to physical units in an economically feasible way [37]. An example would be the wages of the people that verify assembly/component soldering as well as the depreciation on the factory equipment. Shipping costs are the costs incurred when the finished product is shipped from the manufacturer to the company's warehouse and then to the customer [37].

5.3 Break-even Analysis

Break-even analysis is a technique widely used by production management and management accountants. It is based on categorizing production costs between those which are ”variable” (costs that change when the production output changes) and those that are ”fixed” (costs not directly related to the volume of production). Total variable and fixed costs are compared to sales revenue to determine the breakeven point, which is the level of sales volume at which the business makes neither a profit nor a loss (Equation 5.1). A fixed cost is an expense that does not change with an increase or decrease in the number of goods or services produced or sold. Fixed costs are expenses that have to be paid by a company, independent of any business activity [38]. It is one of the two components of the total cost of running a business, the other being variable costs.

$$N = \frac{Fixed\ Cost}{Unit\ Price\ -\ Unit\ Cost}$$

where N is the number of units needed to be sold to break even

For our project, a few of the fixed costs would include paying the engineers, using lab equipment, and renting space. Our conservative pricing and cost analysis has led to the result that around 20,000 pieces would need to be sold before our business could be profitable.

Chapter 6

Hardware Design

This chapter details the hardware design of the Fall Detection and Posture Monitoring System and elaborates on each system block.

6.1 Overall Design

The overall block diagram of the Fall Detection and Posture Monitoring System is depicted in Figure 6.1. Each module and its specifications are elaborated upon below.

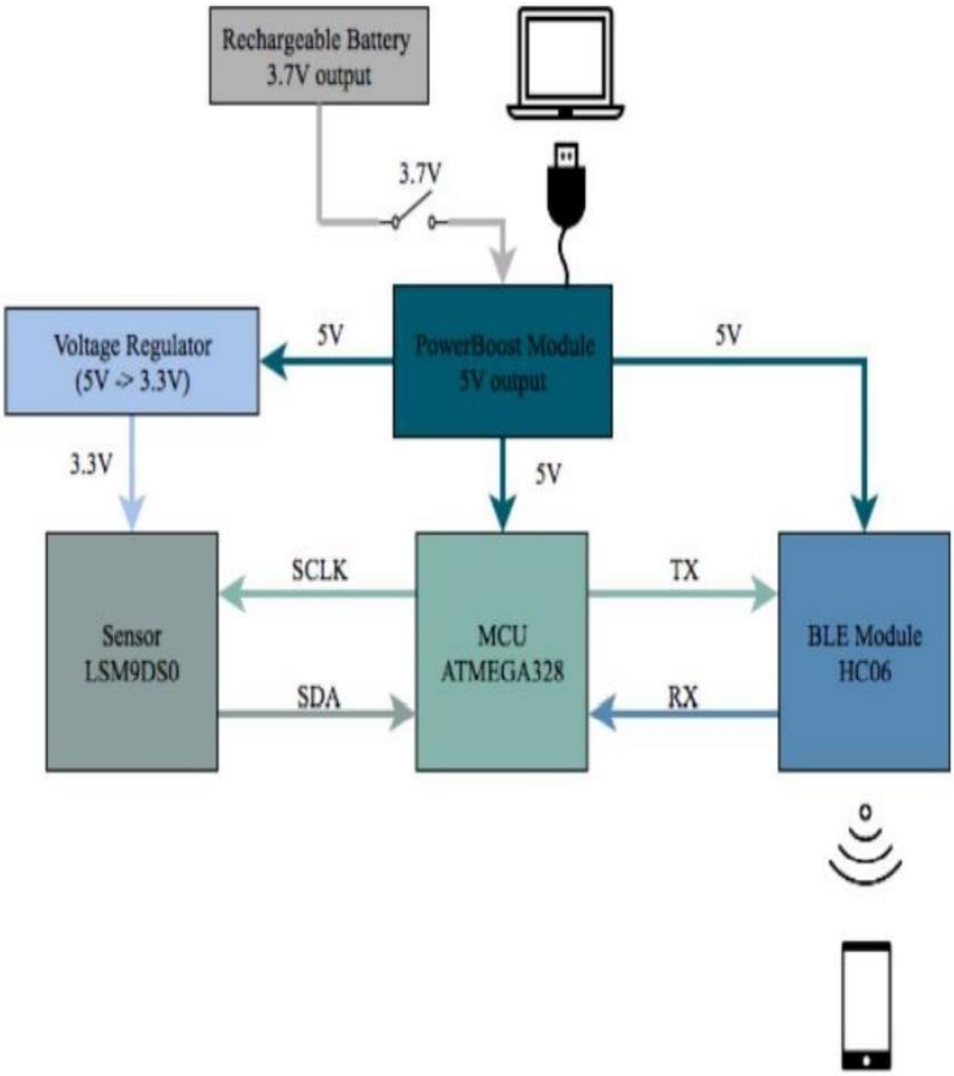


Figure 6.1: System block diagram

6.2 System Blocks and Specifications

6.2.1 Power Supply

The 5V PowerBoost Charger and the 3.7V Lithium Ion Polymer Battery provided by Adafruit is used to both power and charge the fall detection and posture monitoring system. The benefits of using these components are that it includes a built in boost circuit to provide 5V with the 3.7V power supply so that we do not need an extra voltage regulator in our system. Also, it allows our system to be charged via microusb rather than needing an external battery charger. In the design, the team utilizes the micro-usb port to charge the 3.7V Lithium Ion Polymer Battery which then provides the necessary 5V to the ATMEGA328-PU microcontroller (Figure 6.2). A voltage regulator is also used to scale the 5V in order to provide the necessary 3.3V to the sensor.

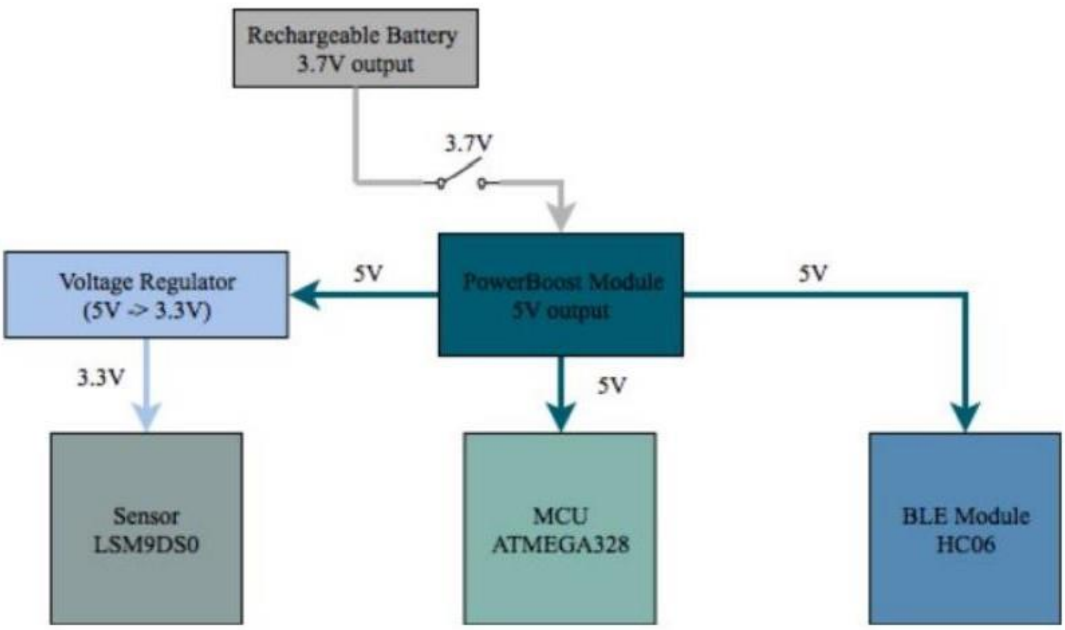


Figure 6.2: Power supply block diagram

6.2.2 Sensor

We chose the LSM9DS0 as our sensor since it had a 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer all in one. The sensor will be placed on the waist and calibrated using that point as the zero. The person's orientation is measured by the sensor and if deviations of more than the specific threshold is detected in the x, y, or z direction, an alert will be sent to the user's smartphone. A voltage regulator will scale the 5V from the power boost and supply the sensor the necessary 3.3V to function. Data is sent to the microcontroller through InterIntegrated Circuit (I2C) protocol. The data line connects to pin SDA and the clock line connects to pin SCL (Figure 6.3).

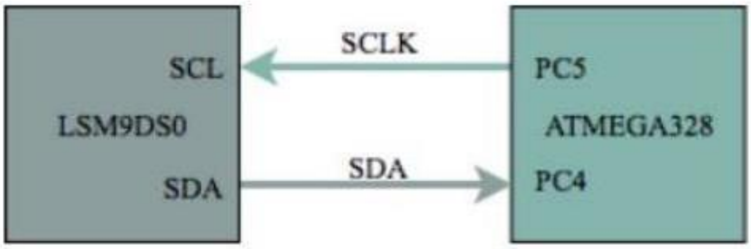


Figure 6.3: Sensor block diagram

Chapter 7

Software Design

This section will briefly outline the strategies and methods used to code both the Arduino and the Android app. The language used by the Arduino is C++, and the environment coded in is Arduino's own IDE [39]. Arduino code is structured so that there are two main functions that must be modified. The 'setup' function is used to enact initializations that should only happen once at the beginning of the code, and the 'loop' function is used to run processes repeatedly while the Arduino is functioning. The Android app, in contrast, uses MIT App Inventor 2 which is a free, cloud based service that allows people to make their own mobile application. This tool utilizes a block based programming language for users to build apps and run them on an Android mobile device [40].

7.1 Arduino IDE

The following sections outline the three main parts of the Arduino code for our Fall Detection and Posture Monitoring System. These main sections are the setup of the sensor, I2C communication, and Bluetooth. The full code can be viewed in Appendix B.

Sensor Setup

The code written to acquire data from the sensor is the simplest part of the Arduino code. Certain variables are declared at the beginning of the code that have values to the Tx and Rx pins being used. These variables are then used with functions from built-in Arduino libraries corresponding to the LSM9DS0 sensor. For the accelerometer sensor, we look at the three axes of the accelerometer data to see if there has been a change and the data is outside of the current threshold. If this is the case, then a byte integer flag is set to 6. For the gyroscope sensor, we convert the data into a three-axes Euler angle. This data is compared to the normalized posture created by the user and if there is an irregular posture is detected, then a different byte integer flag is set to 7. Both the accelerometer and gyroscope code are repeated a single time in the 'loop' function, therefore updating their values every iteration of the loop.

I2C Communication

The LSM9DS0 comes equipped with its own Arduino library that easily implements the communication necessary to sample the accelerometer, gyroscope, and magnetometer. The library allows an object to be declared, which holds all the functions and variables needed to run the accelerometer. After this object is created and the settings initialized in the 'setup' function, the accelerometer and gyroscope are looked at using a library specific function in the 'loop' code. Following this function, the x, y, and z accelerations and angles are obtained from the object that was declared in the beginning. The code then analyzes the values to determine if the person is walking or positioned in an irregular manner. If this is found to be true, a flag is set to the value of 1, indicating that a problem has occurred

Bluetooth Setup

The HC-06 module is able to run with its own library, which integrates with the Arduino. To implement this, a serial monitor is initialized and named in the 'setup' function. This then allows the Arduino to read values from, or write values to the characteristics built into the serial monitor. In the 'loop' function, functions were used to write the values of the various flags, described as seen in Figure 7.1, which then allows them to be to sent to the central device (the app).

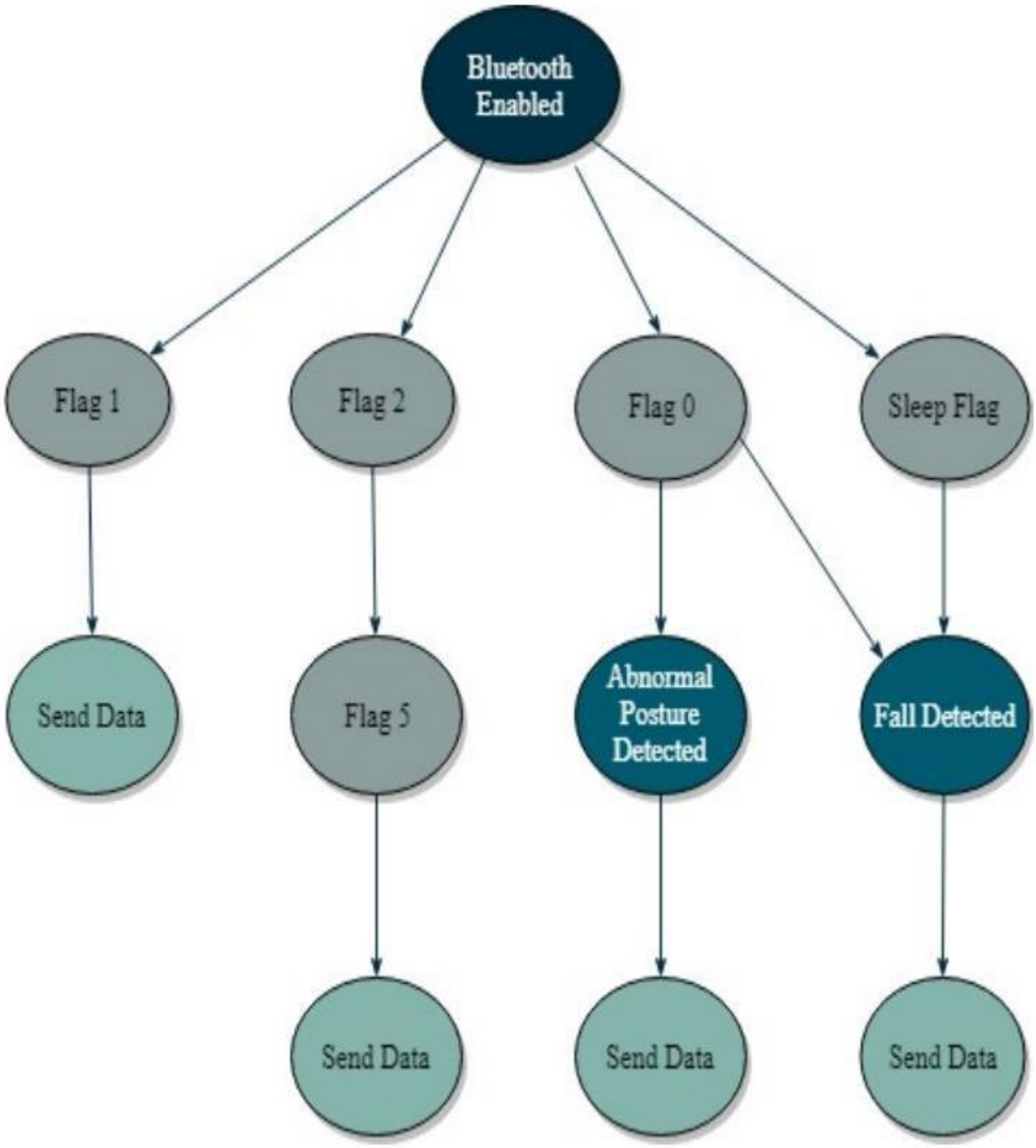
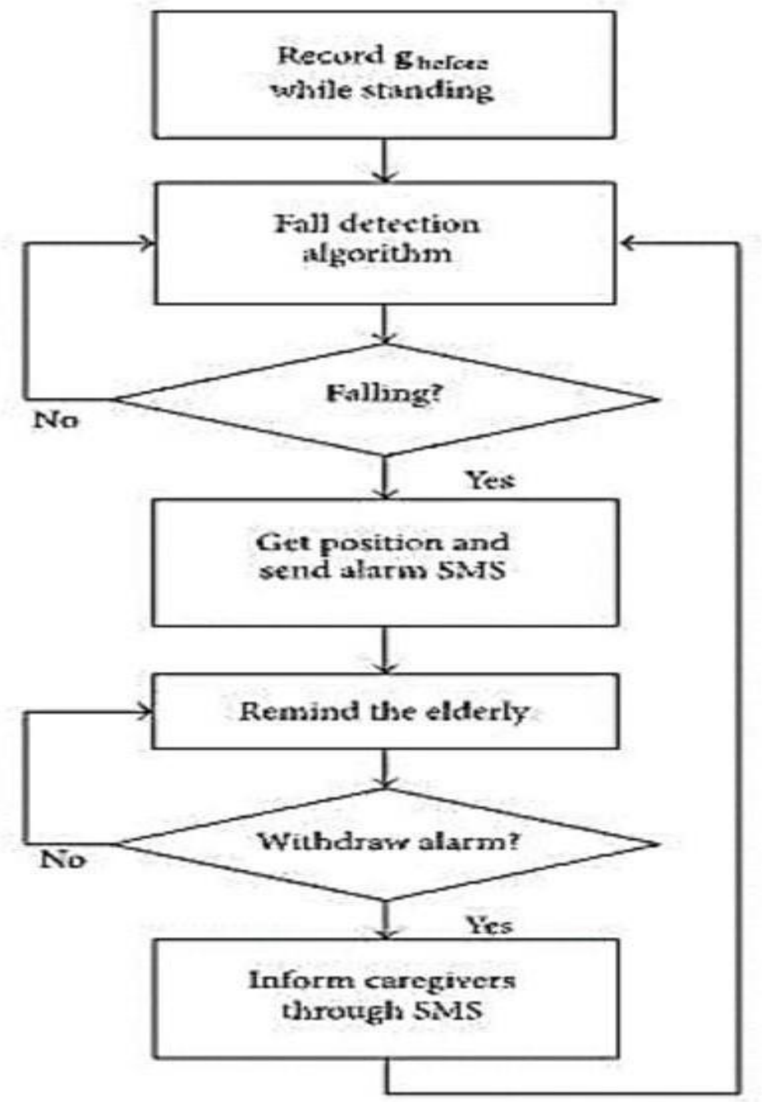


Figure 7.1: Bluetooth flag functionality flowchart

Each flag is a single byte of data because the function used to write the data requires only a byte of data as a parameter. These flags are sent when an abnormality was detected in the user's walking or posture. In addition, the data from the accelerometer and gyroscope is sent to the App for graphing purposes. This functionality is included once at the end of the 'loop' function, after each flag has had a chance to update its value based on its corresponding sensor.

7.2 Algorithms for Detection

All of the detection algorithms are on the Arduino code side. The MCU will read a flag from the app, analyze the current condition, and then judge whether to send an alert and/or data to the app. The app will then use the alert and data to send a posture message, a fall alert message, or graph the accelerometer and Euler angle data. The full software flowchart can be seen in Figure 7.2.



7.2.1 Fall Detection Algorithm

When we first developed our algorithm for fall detection, we only looked at the Y-axis data, which is the gravity axis when a person is standing. Whenever the magnitude of the Y-axis acceleration became zero, we assumed that the user had fallen down. However, this "naive algorithm" did not work when a person fell down on stairs and their body had an angle to the ground. In this situation, the magnitude of the Y-axis acceleration would not be zero.

The next version of the algorithm focused on the X-axis and Z-axis. We did several tests to see the X-axis and Z-axis acceleration changes under different motion conditions, including jumping, running, walking, falling when the Y-axis is equal to zero, and falling when the Y-axis is not equal to zero. (Detailed test results are included in Chapter 9.) When the magnitude of the X-axis or Z-axis is greater than 6.9 m/s2 and lower than 10.2 m/s2 , the user falls down. The reason of the lower boundary is that we assume that 45 degrees to the ground is the largest angle a person can fall. Under that falling condition, the Y-axis as well as the X- or Z-axes will split the gravity acceleration. Since the gravity on the earth is around 9.8 m/s2 , we then calculated that the Y-axis and X or Z-axes which are equal to the gravity/sqrt(2) ≈ 6.9 m/s2 (Figure 7.3).

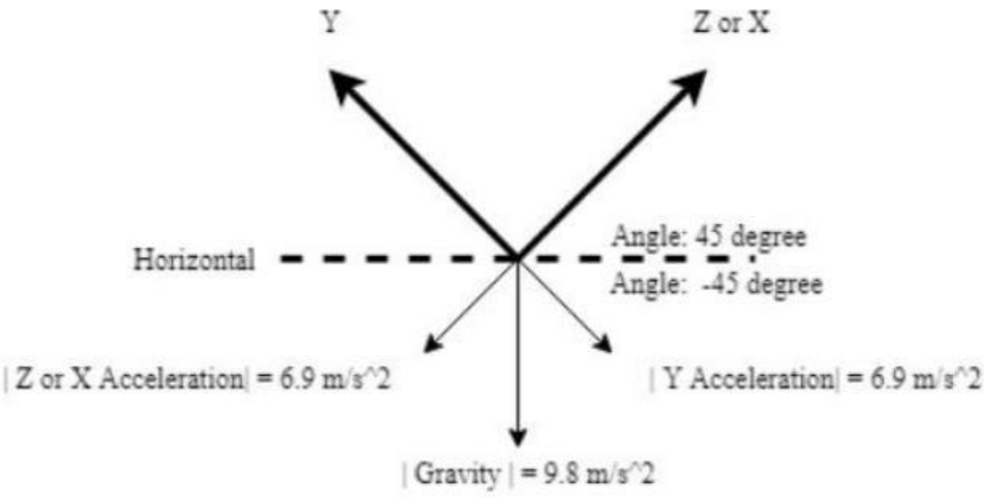


Figure 7.3: 45 degree fall acceleration analysis

Another feature of any falling condition is that all of the acceleration axes will stay the same until the user stands up. To accommodate this functionality, we added a debounce algorithm for our the fall detection algorithm. When the X-axis or Z-axis acceleration remains in the threshold range for more than two second, the MCU will send a fall detection flag to the app.

7.3 software code

```
#include <Wire.h>

#include <ESP8266WiFi.h>

const int MPU_addr=0x68; // I2C address of the MPU-6050

int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;

boolean fall = false; //stores if a fall has occurred

boolean trigger1=false; //stores if first trigger (lower threshold) has occurred

boolean trigger2=false; //stores if second trigger (upper threshold) has occurred

boolean trigger3=false; //stores if third trigger (orientation change) has occurred

byte trigger1count=0; //stores the counts past since trigger 1 was set true

byte trigger2count=0; //stores the counts past since trigger 2 was set true

byte trigger3count=0; //stores the counts past since trigger 3 was set true

int angleChange=0; // WiFi network info.

const char *ssid = "12345"; // Enter your Wi-Fi Name

const char *pass = "123456789"; // Enter your Wi-Fi Password

void send_event(const char *event);

const char *host = "maker.ifttt.com";

const char *privateKey = "owkigF01nmCboTZmPH7d32w3YFT4nQVPqB9zjK6lrWK";

void setup(){

 Serial.begin(115200);

 Wire.begin();

 Wire.beginTransmission(MPU_addr);

 Wire.write(0x6B); // PWR_MGMT_1 register

 Wire.write(0);     // set to zero (wakes up the MPU-6050)

 Wire.endTransmission(true);
```

```
Serial.println("Wrote to IMU");
 Serial.println("Connecting to ");
 Serial.println(ssid);
 WiFi.begin(ssid, pass);
 while (WiFi.status() != WL_CONNECTED)
 {
   delay(500);
   Serial.print(".");              // print ... till not connected
 }
 Serial.println("");
 Serial.println("WiFi connected");
}
void loop(){
mpu_read();
ax = (AcX-2050)/16384.00;
ay = (AcY-77)/16384.00;
az = (AcZ-1947)/16384.00;
gx = (GyX+270)/131.07;
gy = (GyY-351)/131.07;
gz = (GyZ+136)/131.07;
// calculating Amplitute vactor for 3 axis
float Raw_Amp = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
int Amp = Raw_Amp * 10; // Mulitiplied by 10 bcz values are between 0 to 1
Serial.println(Amp);
if (Amp<=2 && trigger2==false){ //if AM breaks lower threshold (0.4g)
  trigger1=true;
```

```
  Serial.println("TRIGGER 1 ACTIVATED");
  }
 if (trigger1==true){
  trigger1count++;
  if (Amp>=12){ //if AM breaks upper threshold (3g)
    trigger2=true;
    Serial.println("TRIGGER 2 ACTIVATED");
    trigger1=false; trigger1count=0;
    }
 }
 if (trigger2==true){
  trigger2count++;
  angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5); Serial.println(angleChange);
  if (angleChange>=30 && angleChange<=400){ //if orientation changes by between 80-
100 degrees
    trigger3=true; trigger2=false; trigger2count=0;
    Serial.println(angleChange);
    Serial.println("TRIGGER 3 ACTIVATED");
      }
  }
 if (trigger3==true){
   trigger3count++;
   if (trigger3count>=10){
      angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
      //delay(10);
      Serial.println(angleChange);
```

```
      if ((angleChange>=0) && (angleChange<=10)){ //if orientation changes remains
between 0-10 degrees
          fall=true; trigger3=false; trigger3count=0;
          Serial.println(angleChange);
            }
      else{ //user regained normal orientation
         trigger3=false; trigger3count=0;
         Serial.println("TRIGGER 3 DEACTIVATED");
      }
    }
  }
 if (fall==true){ //in event of a fall detection
   Serial.println("FALL DETECTED");
   send_event("fall_detect");
   fall=false;
   }
 if (trigger2count>=6){ //allow 0.5s for orientation change
   trigger2=false; trigger2count=0;
   Serial.println("TRIGGER 2 DECACTIVATED");
   }
 if (trigger1count>=6){ //allow 0.5s for AM to break upper threshold
   trigger1=false; trigger1count=0;
   Serial.println("TRIGGER 1 DECACTIVATED");
   }
  delay(100);
   }
```

```
void mpu_read(){
 Wire.beginTransmission(MPU_addr);
 Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
 Wire.endTransmission(false);
 Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
 AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
(ACCEL_XOUT_L)
 AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
 AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
(ACCEL_ZOUT_L)
 Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
 GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
 GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
 GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
 }
void send_event(const char *event)
{
  Serial.print("Connecting to ");
  Serial.println(host);
    // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("Connection failed");
    return;
  }
```

```
  // We now create a URI for the request
 String url = "/trigger/";
 url += event;
 url += "/with/key/";
 url += privateKey;
 Serial.print("Requesting URL: ");
 Serial.println(url);
 // This will send the request to the server
 client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");
 while(client.connected())
 {
   if(client.available())
   {
     String line = client.readStringUntil('\r');
     Serial.print(line);
   } else {
     // No data yet, wait a bit
     delay(50);
   };
 }
 Serial.println();
 Serial.println("closing connection");
 client.stop();
}
```

Chapter 8

Results and References

8.1 Code testing

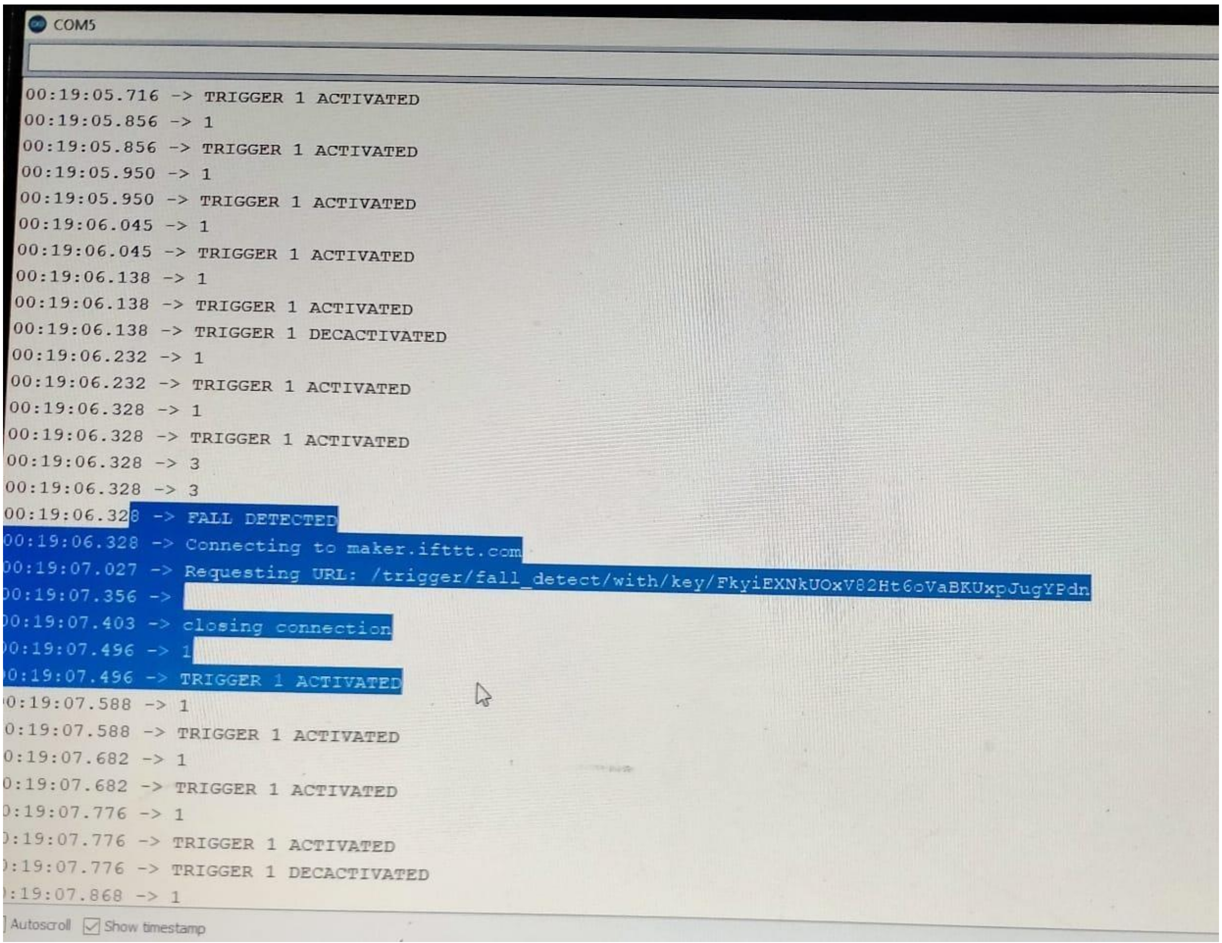


Fig:8.1 code testing

8.2 Model graph

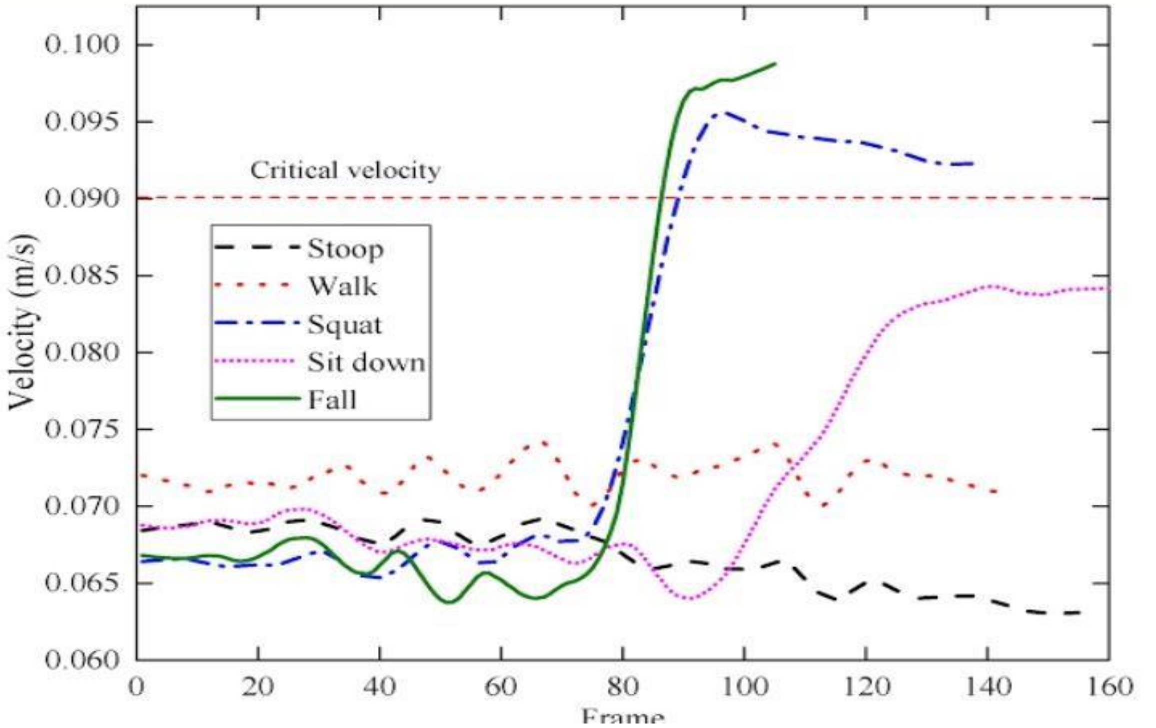


Fig:8.2 Model graph

8.3 References

[1] Y. He, Y. Li, and S.-D. Bao, "Fall detection by built-in tri-accelerometer of smartphone," in Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 184–187, IEEE, Hong Kong, China, January 2012.

View at: Google Scholar

[2] Q. V. Vo, G. Lee, and D. Choi, "Fall detection based on movement and smart phone technology," in Proceedings of the IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), pp. 1–4, IEEE, Ho Chi Minh City, Vietnam, February-March 2012.

View at: Google Scholar

[3] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, "A smartphone-based fall detection system," Pervasive and Mobile Computing, vol. 8, no. 6, pp. 883–899, 2012.

View at: Publisher Site | Google Scholar

[4] M. Yuwono, B. D. Moulton, S. W. Su, B. G. Celler, and H. T. Nguyen, "Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems," Biomedical Engineering Online, vol. 11, no. 1, p. 9, 2012.

View at: Publisher Site | Google Scholar

[5] S. S. Khan and J. Hoey, "Review of fall detection techniques: a data availability perspective," Medical Engineering & Physics, vol. 39, pp. 12–22, 2017.

View at: Publisher Site | Google Scholar

[6] T. Theodoridis, V. Solachidis, N. Vretos, and P. Daras, "Human fall detection from acceleration measurements using a recurrent neural network," in Precision Medicine Powered by pHealth and Connected Health, pp. 145–149, Springer, Berlin, Germany, 2018.

View at: Google Scholar

[7] P. Cheng, L. Tan, P. Ning et al., "Comparative effectiveness of published interventions for elderly fall prevention: a systematic review and network meta-analysis,"

International Journal of Environmental Research and Public Health, vol. 15, no. 3, p. 498, 2018.

[8] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, and O. Tanaka, "A wearable airbag to prevent fall injuries," IEEE Transactions on Information Technology in Biomedicine, vol. 13, no. 6, pp. 910–914, 2009.

View at: Publisher Site | Google Scholar

[9] Z. Zhong, F. Chen, Q. Zhai et al., "A real-time pre-impact fall detection and protection system," in Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1039–1044, IEEE, Auckland, New Zealand, July 2018.

View at: Google Scholar

[10] C.-F. Lai, M. Chen, J.-S. Pan, C.-H. Youn, and H.-C. Chao, "A collaborative computing framework of cloud network and WBSN applied to fall detection and 3-D motion reconstruction," IEEE Journal of Biomedical and Health Informatics, vol. 18, no. 2, pp. 457–466, 2014.

View at: Publisher Site | Google Scholar

[11] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, "Fall detection system for elderly people using IoT and big data," Procedia Computer Science, vol. 130, pp. 603–610, 2018.

[12] Y. He, Y. Li, and S.-D. Bao, "Fall detection by built-in tri-accelerometer of smartphone," in Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 184–187, IEEE, Hong Kong, China, January 2012.

[13] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, "A smartphone-based fall detection system," Pervasive and Mobile Computing, vol. 8, no. 6, pp. 883–899, 2012.

[14] Z. Zhong, F. Chen, Q. Zhai et al., "A real-time pre-impact fall detection and protection system," in Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1039–1044, IEEE, Auckland, New Zealand, July 2018.

[15] D. Ajerla, S. Mahfuz, and F. Zulkernine, “Fall detection using physical activity monitoring data,” in Proceedings of the 7th International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, Big Mine 2018, ACM SIGKDD, London, UK, April 2018.

[16] Z. Wang, Y. Zhang, Z. Chen et al., “Application of relief algorithm to selecting feature sets for classification of high resolution remote sensing image,” in Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 755–758, Beijing, China, July 2016.

[17] L. N. V. Colon, Y. de la Hoz, and M. Labrador, “Human fall detection with smartphones,” in Proceedings of the 2014 IEEE Latin-America Conference on Communications (LATINCOM), pp. 1–7, IEEE, Cartagena de Indias, Colombia, November 2014.

Result analysis

To measure the performance of the developed system, several measurements are used. Specificity and sensitivity are the two statistical measures used to measure the performance of binary classifiers. Accuracy is another term used to measure the overall performance of a classifier model. Sensitivity is the proportion of actual positives that have been correctly iden- tified as positives by the classifier.