DESIGN OF A NEURAL NETWORK BASED FALL DETECTION AND ALERT SYSTEM

NG YONG JIE

A report submitted in partial fulfilment of the requirements for the degree of
Bachelor of Mechatronics Engineering

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

I hereby declare that I have read through this report entitle "Design of a Neural Network Based Fall Detection and Alert System" and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Mechatronics Engineering.

Signature   : ....

Supervisor's Name   : ....

Date   : ....

I declare that this report entitle "Design of a Neural Network Based Fall Detection and Alert System" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : ....

Date : ....

To my beloved mother and father

ACKNOWLEDGEMENT

Firstly, I would like to thank my university, University of Technical Malacca (UTeM) for providing an opportunity for me to undertake my Final Year Project (FYP) in fulfilment of the requirement for Bachelor of Mechatronics Engineering. In performing my FYP in these few months, I had accepted useful assists and guideline from some kind and respected persons. I would like to show my gratitude to my project supervisor, Mr. Nik Syahrim bin Nik Anwar, for giving me a good guideline for my FYP. I would also like to express my deepest gratitude to my classmates, especially those who has directly and indirectly guided me in my training.

Moreover, I would especially like to thank my housemates, Tan Wei Chiang, Ooh Man Chun and Melvin Gan Yeou Wei for willing to share their knowledge and experiences during my training. I perceive the knowledge and experiences which I have gained throughout my FYP as a priceless component in my future career development. I also want to show my appreciation to all the lecturers and staffs from Faculty of Electrical Engineering, UTeM for willing to assist me and above all to your good companionship.

Finally, I must express my profound gratitude to my parents for providing me the opportunity to study at university and giving me support throughout my study life and this period of my FYP. This accomplishment would not have been possible without them. I perceive my FYP as a big milestone in my career development. I will make my great effort to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

ABSTRACT

Accidental falls are considered the major cause of accidents that could lead to paralysis, accidental deaths or psychological damage. In most of the fall accidents, external support is crucial in order to prevent major injuries. Thus, a system that automatically detects fall event could help to reduce fall events and efficiently improve the prognosis of fall victims. This project proposes a neural network based fall detection and alert system with SMS alert and GPS function. A GY-80 10 Degree of Freedom (DOF) Inertial Measurement Unit (IMU) module is mounted on a wearable waist-worn device to continuously record body movements and detect body postures. The GY-80 10DOF IMU module consists of BMP085 barometer, HMC5883L magnetometer, ADXL345 accelerometer and L3G4200D gyroscope. For this project, we only use the accelerometer and gyroscope for the fall detection. The tri-axial accelerometer measures the static acceleration of gravity with high resolution (4 mg/LSB) which enables measurement of inclination changes less than 1.0°. The gyroscope is device that measure or maintains rotational motion. A new neural network algorithm has been developed to accurately distinguish falls from different postural transitions during activities of daily living (ADL) including standing, walking, jumping, running, sitting and lying. A body temperature and heart pulse monitoring device was developed for this system to assist the rescue team know the body condition of the user during the fall occurs. The application of the system is implemented on the Android platform. Once a fall accident happens, the alert system will be triggered and send emergency messages, the actual location and body conditions of the user to the recipient. Fall and ADL simulations were performed by a group of subjects to test and to validate the performance of the system. The experiment results showed that the proposed system could obtain sensitivity of 95.5%, specificity of 96.4% and accuracy of 96.3%.

ABSTRAK

Kemalangan jatuh dianggap sebagai punca utama kemalangan yang boleh membawa kepada lumpuh, kematian akibat kemalangan atau kerosakan psikologi. Dalam kebanyakan kemalangan jatuh, sokongan luar adalah penting untuk mengelakkan kecederaan serius. Oleh itu, satu sistem yang dapat mengesan kejatuhan secara automatik boleh membantu untuk mengurangkan kejadian jatuh dan meningkatkan prognosis mangsa jatuh. Projek ini mencadangkan satu sistem pengesanan kejatuhan dan sistem amaran berasaskan rangkaian neural.bersama dengan fungsi SMS dan GPS. GY-80 10 DOF Inersia Unit Pengukuran (IMU) modul dipasang pada pinggang pengguna untuk terus rekod pergerakan badan dan mengesan postur badan. GY-80 10DOF IMU modul terdiri daripada BMP085 barometer, HMC5883L magnetometer, ADXL345 akselerometer dan L3G4200D giroskop. Untuk projek ini, kami hanya menggunakan akselerometer dan giroskop untuk mengesan kejatuhan. Akselerometer mengukur statik graviti dengan resolusi tinggi (4mg/LSB) yang membolehkan pengukuran pengubahan kecenderungan kurang daripada 1.0°. Giroskop adalah peranti yang digunakan untuk mengukur atau mengekalkan pergerakan putaran. Algoritma rangkaian neural baru telah dibangunkan untuk membezakan jatuh dari postur yang berbeza semasa aktiviti kehidupan harian (ADL) termasuk berdiri, berjalan, melompat, berlari, duduk dan berbaring. Peranti untuk pemantauan suhu badan dan jantung nadi telah dibangunkan untuk sistem ini untuk membantu pasukan penyelamat tahu keadaan badan pengguna semasa kejatuhan berlaku. Penggunaan sistem ini dilaksanakan pada platform Android. Apabila kemalangan jatuh berlaku, sistem amaran akan mencetuskan dan menghantar mesej kecemasan, lokasi dan keadaan badan pengguna kepada penerima. Simulasi jatuh dan ADL telah dijalankan oleh sekumpulan sukarelawan untuk menguji dan mengesahkan prestasi sistem. Keputusan eksperimen menunjukkan bahawa sistem yang dicadangkan itu boleh mencapai 95.5% sensitiviti, 96.4%kekhususan dan 96.3% ketepatan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATION

UTeM   -   Universiti Teknikal Melaka Malaysia

ADL   -   Activities of daily living

DOF   -   Degree of freedom

IMU   -   Inertial measurement unit

SMS   -   Short Message Service

GPS   -   Global Positioning System

MEMS   -   Micro-electromechanical systems

AHRS   -   Attitude heading reference system

IDE   -   Integrated development environment

ANN   -   Artificial neural network

BPNN   -   Back propagation neural network

TCP/IP   -   Transmission Control Protocol/Internet Protocol

HTTP   -   Hypertext Transfer Protocol

IOT   -   Internet of things

CHAPTER 1

INTRODUCTION

1.1   Introduction

Fall accident is one of the health risks which frequently happens, especially for the elderly who over the age of 65 years old. There are about 33% of the elderly are reported experience fall injuries at least once per year and 68% hospitalization of the elderly are fall-related [1]. Moreover, there are some sports activities like hiking will cause serious fall injuries for people. Every year, fall accidents will cause about 10,000 deaths among human aged 65 years and above in United State [1]. Therefore, in this thesis, we study and design a high performance human fall detection and alert system to reduce the percentage of fall injuries and improve medical care.

Over the last decade, the approach on fall detection is normally categorized into ambient-based, vision-based and wearable device-based [2]. However, ambient-based and visual-based methods are unpractical and can be significantly affected by the external environment. The interference factors which exist in the visual-based method will increase the difficulty for fall detection and the recognition rate is just about 50% to 70% [2]. Wearable sensors-based approach works by measuring and processing the initial signal of human motion, such as the acceleration and the angular rate. However, there are some difficulty to classify between fall activities and other activities which have similar signal profile with fall, such as running and lying down. Some researchers proposed a method which predicts falls using threshold of the angle and time. Since the fall events occur randomly, this method is unstable and has low recognition rate. By applying the neural

network algorithm, the presence of falls can be accurately detected less than 400ms before the collision happens [2].

1.2   Motivation

The main motivation of this project is to design a real time human fall detection and alert device for fall injuries reduction and medical care improvement. The fall accidents are a major cause of accidents with an average of 1042 cases annually in Malaysia. [3]. At the same time, population aged 65 years and above in Malaysia has increased by 0.2 percentage from 5.8% in 2015 to 6.0% in 2016 and the population of elderly is anticipated to increment to 7% by 2020 [4] [5]. As previously mentioned, fall accident is frequently happening, especially for the elderly who over the age of 65 years old [1]. There are about 33% of the elderly are reported experience fall injuries at least once per year and 68% hospitalization of the elderly are fall-related [1]. Figure 1.1 presents the cause of accidents in Malaysia from year 2005 to the year 2009.
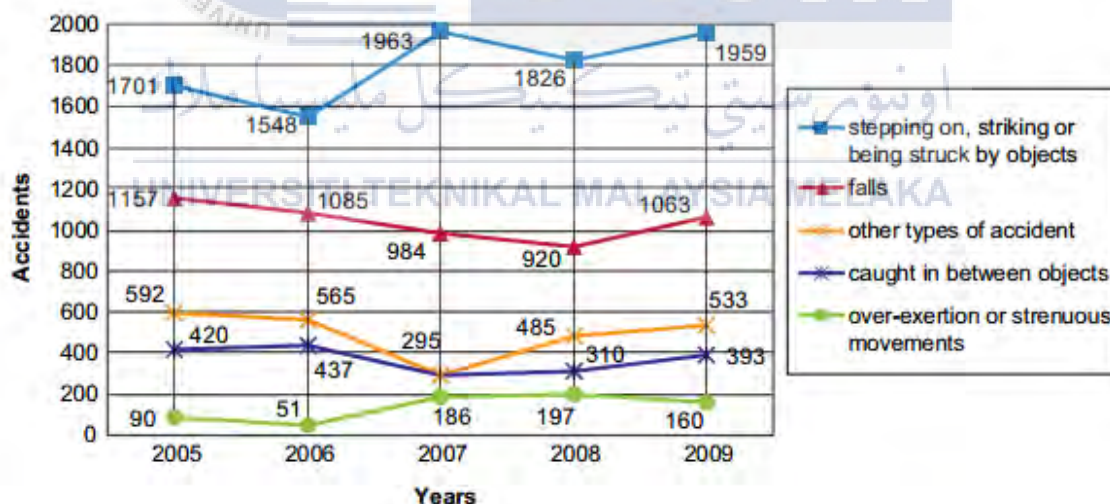
Figure 1.1: Cause of accidents in Malaysia (2005-2009) [3]

Moreover, falls can be a critical threat to the elderly who live alone. This is due to the injuries from a fall can lead to the risk of death or 'post-fall syndrome' if the victim unable receives any help immediately after a fall. Although the victim did not experience physical injury after a fall, it also may lead to psychological damage [6]. For those who experience serious fall, they might decrease their activities of daily living (ADL)

and self-care due to fear of falling again. This behaviour decreases their mobility, balance and fitness, leads to reduced social interactions, and increased depression. Figure 1.2 shows fall of the elderly.

Figure 1.2: Fall of elderly [3]

Falls are also consists of several types and can occur in different directions and at different speed. Faint falls and Crumble falls are examples of the unexplained falls. A faint fall happens where the victim loses his consciousness before the fall. Hence, the victim unable to seek any support on their own when falling. Generally, Faint falls occur at high speed and may cause critical injury for victims. On the other hand, Crumble fall is type of fall where the victim falls down slowly and sometimes takes the support of a wall or other objects. In order to distinguish the variety of falls, a fall detection algorithm is needed to separate different fall activities from other ADL.

There are plenty of possible solutions for reduce fall injuries, but none has been optimized. For examples, a physical alarm would be intrusive on the surrounding patients and a 24 hour call centre would be more costly. Therefore, this project will create a human fall alert application for Android smart phones which can be used at any place and also one of the lower-cost solutions. Furthermore, there are many reasons that fall accident occurs, for examples heart attack and heat stroke. So, knowing the cause of the victim's fall may be crucial to treatment. Therefore, our project includes a vital sign monitoring device to communicate with our fall detection system. When a fall accident happens, the body conditions of the victim such as body temperature and pulse rate will also send together with the alert message to the third party. This extra information will help the doctor and rescue team to know the conditions of the victim during the fall occurs.

1.3   Problem Statement

Accidental fall is one of the major causes of accidents that may lead to paralysis, accidental deaths or psychological damage. This type of accident has frequently happened among the elderly. For example, when the elderly alone at home, he suddenly falls down and loses consciousness, but no people found out and aware about this accident. This may cause they miss the best rescue time. Moreover, there are many reasons for the fall accident, such as heart attack and heat stroke. However, the doctor or the rescue team unable to know the reasons of fall for the victims immediately during the treatment.

Over the last decades, the fall detection and alert system can be used efficiently for reducing the dangers of the elderly falling. A fall detection device can be based on measurements of a number of different sensors. However, there are pros and cons regarding the utilization of different sensors in the system. For examples, the acoustic sensors and visual sensors may lead to some problems when implemented in the system. The acoustic noises that occur in the surrounding will easily influence the acoustic system [1]. The performance of the fall detection decreases due to the disturbance of acoustic noises. For visual-based sensor, the system's performance can be greatly affected by the external environment for examples due to lighting conditions, camera quality, background and occlusion size [1]. To overcome the limitations of acoustic and visual-based methods, wearable-based sensors will used in the system.

Data collected from both sensors can be major issues as to where the level of accuracy can be used to detect fall signals. Most of the researches use the threshold of the angle and time in the fall detection algorithm. Since the fall events occur randomly, this method is unstable and has low recognition rate. Therefore, the neural network algorithm which can easily learn and recognize the pattern of falls and other ADL is applied in the system. The accuracy of the human fall alert system is extremely correlated to the positions of the sensors. The complexity of activities done by our wrist will lead to high rate of fall detection if compare to the motion of other parts of our body such as head and waist. Moreover, the device on people's head will influence aesthetics [7]. Since the waist is the centre of gravity on the human body and truly reflects the posture of the trunk, waist worn is selected for our system design.

1.4   Objectives

The main objectives of this project are listed below:

1.   To design and develop a fall detection system using both accelerometer and gyroscope sensors.

2.   To implement neural network algorithm in order to accurately distinguish between falls and different postural transitions during activities of daily living (ADL)

3.   To design a vital sign monitoring device which consists of a pulse sensor and a temperature sensor in order to improve the emergency alert system.

1.5   Scope

The scope of the project focuses on the development of a human fall detection and alert system with a neural network algorithm. An Intel Edison with Mini Breakout Board and GY80 10 Degree of Freedom IMU module are used in this project. GY80-10 DOF Inertial Measurement Unit module consists of a L3G4200D (3-Axis Gyroscope), ADXL345 (3-Axis Digital Accelerometer), HMC5883L (3-Axis Magnetometer) and BMP085 (Barometric Pressure Sensor). However, only the accelerometer and gyroscope of the GY80 IMU module are utilized in this project. The accelerometer is being used in this project to measure the static acceleration of gravity for different activities while the gyroscope is used to measure angular velocity. The fall detection algorithm was developed and programmed into an Intel Edison board. The application of the system is implemented on the Android platform. The GPS and SMS function of the smart phones used in the Android Application as an emergency alert system. Besides that, an extra feature which consists of body temperature and heart pulse rate monitoring also include in this system to improve the alert function. Experiments and simulations of falls were carried out with 15 subjects to test the performance and validity of the system.

CHAPTER 2

LITERATURE REVIEW

2.1   Introduction

Over the past decades, many researchers developed the system to detect the presence of fall and then send an alert signal to the third party. This chapter presents a review of literature on the study related to the problem of falls in the world and various approaches to fall detection. Apart from that, wearable fall detection system and health care system will be further explained and analysed based on researchers' findings. This chapter aims to provide the overview of the ideas and background information in the report. Furthermore, this review reveals new ways to interpret and identifies any gaps in knowledge and research on the fall detection.

2.2   Study of Fall

Before starting to develop the human fall alert system, it is important to know and comprehend the meaning of the term of fall. The content of this thesis work will be easier to grasp once the understanding of the event of a fall is established. There are several of ways of defining a fall by different people. The World Health Organization defines a fall as an event when a person coming to rest without intention on the ground, floor or other lower level [8]. Falls are the major cause of inadvertent injury in elderly. A serious fall without any immediate detection and treatment might cause injury or accidental death. Even if the falls did not cause serious injury to the elderly people, falls also can cause lack of confidence and scare of falling. The fear of falling might cause them to become less active over time, which rise up their risk of falling [8]. To identify and address the factors that

contribute falls, the elders are encouraged to discuss all falls that occurred, including those which not cause any injury with their medical practitioners or other health professional [8]. With the advent of the technology, many fall detection and alert approaches were proposed to reduce the fall fatalities.

2.3   Methods for Fall Detection

In the last year, there are several of methods were developed for fall detection. The approaches on fall detection are generally classified into three different categories based on the deployed sensor technology, which are ambience device approach, vision-based approach, and wearable devices approach. Table 2.1 compares the advantages and disadvantages of the three major fall detection methods.

2.3.1   Ambience Device Approach

This kind of fall's detection approach utilizes multiple sensors to collect the data related person when the person is close to the sensors. The ambient device falls detection techniques include the acoustic-based and vibration-based methods. An acoustic-based fall detection system (acoustic FADE) that utilized a microphone array and beamforming was developed by Li et al [9]. The acoustic data in real environment were collected using the Microsoft Kinetic. This acoustic fall detection system able to achieve 80% accuracy under moderate background noise (SNR = 0 dB) and strong interference (SIR = −5 dB). For vibration-based fall detection, Zigel et al. [10] Proposed an automatic detection system using a combination of floor vibration and sound sensors, resulting in sensitivity of 97.5% and specificity of 98.6%. Sensitivity refers to the percentage of true falls that are accurately detected by the system while the specificity refers to the percentage of false fall alarms among ADL samples. The formula used to calculate these two parameters are shown in equation 3.11 and 3.12. The evaluation of the fall detection performance will further discuss in section 3.11. In their work, spectral and temporal features were extracted from the sensor's signals. Then, fall and non-fall activities were classified using a Bayes' classifier. Figure 2.1 shows the fall detection and classification block diagram for the system. However, ambient-based approach has low accuracy and high rate of false alarms.

This kind of methods unable is visually verified by a caregiver. Furthermore, these techniques are costly and easily inference by environment.
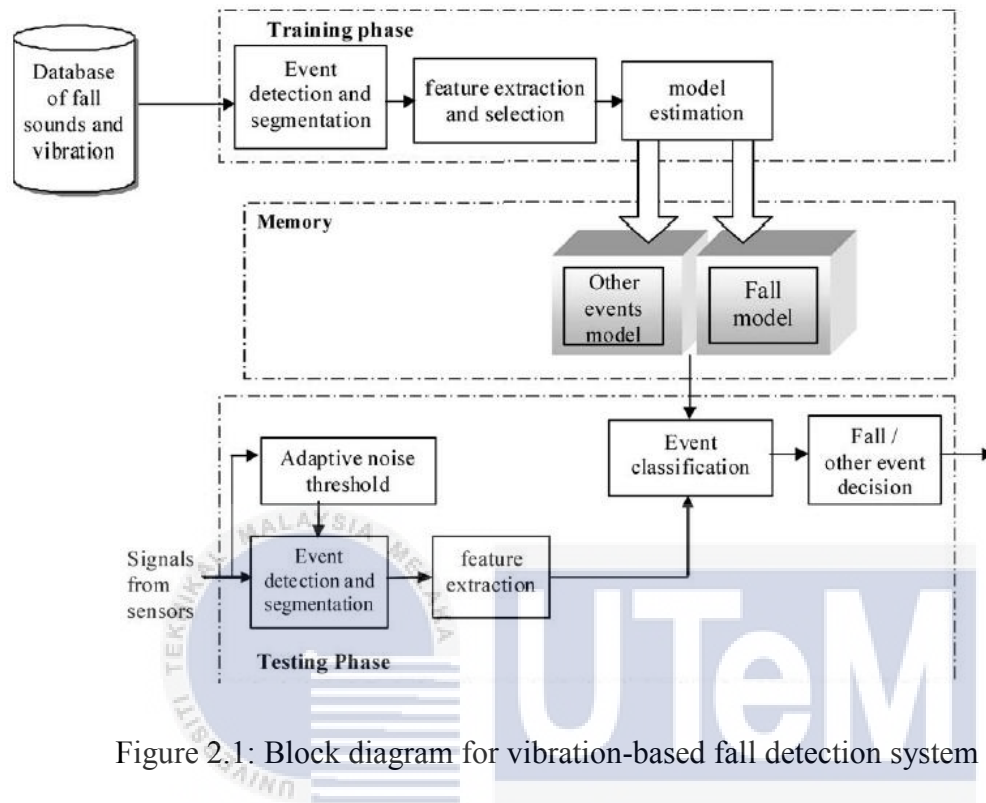
Figure 2.1: Block diagram for vibration-based fall detection system [10]

2.3.2 Vision-based Approach

With the rapid growth of the image recognition technique, computer vision method had been widely utilized in the fall detection system. This solution detects the presence of the falls using the extracted information from the captured image or video. In the study of fall detection using vision-based sensors, Yu et al [11] developed a vision-based fall detection system based on posture recognition with high sensitivity of (97.08%) and high specificity (99.2%). Yu and his research team applied a background subtraction algorithm in the fall detection system to extract the foreground human body and some post-processing was applied to improve the background subtraction results. Figure 2.2 and 2.3 shows the schematic representation and the flowchart of Yu's fall detection system respectively.

Moreover, Chua et al [12] proposed video-based fall detection based on human shape variation, resulting sensitivity of 90.5% and specificity of 90%. For this solution, three points of different regions of the human body were used to represent a human shape. This technique detects falls by shape analysis using the features extracted from the lines formed by these three points. However, there are some inference factors existing in the visual-based method, such as background, lighting conditions, camera quality and occlusion size. This will increase the difficulty to identify fall events and limit the recognition rate to about 50% to 70% [2].
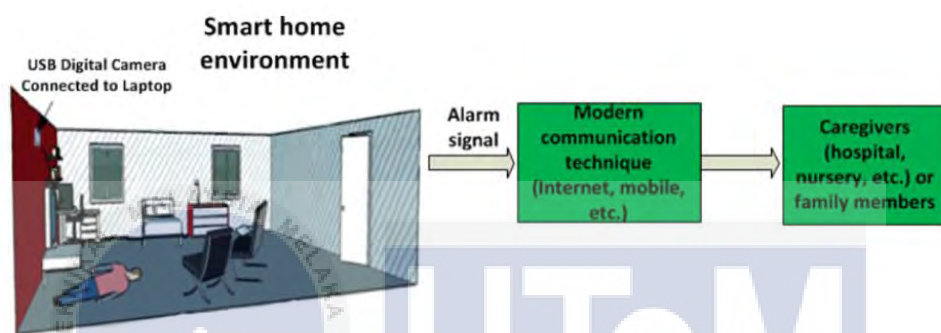
Figure 2.2: Block diagram of the vision-based fall detection system [11]
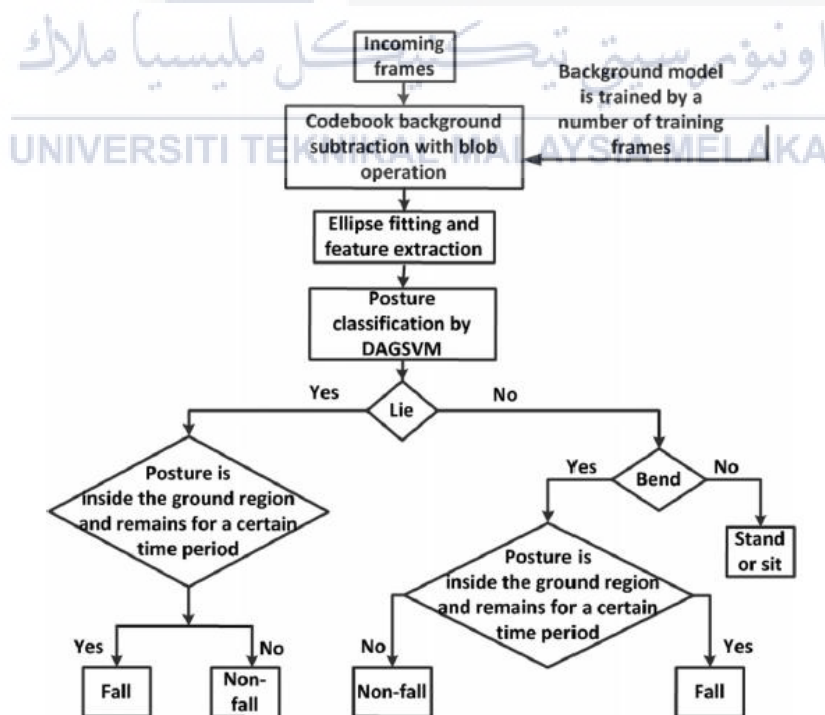
Figure 2.3: Flow Chart of the vision-based fall detection system [11]

2.3.3 Wearable Devices Approach

In recent years, most of the automatic real time fall detection systems developed are wearable device based [1, 2, 6, 7, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. Generally, the wearable electronic for fall detection is attached to the wristband and belts as a unit to continuous record motion data and detect the fall. The wearable devices use body-attached sensors like accelerometer and gyroscope to measure and acquire the characteristics of human motion. The wearable devices based fall detection can clearly distinguish the falls through analysing the kinematic data collected from the monitored individual. Moreover, this fall detection approach is widely applied due to its low cost and not limited to instrumented spaces. Table 2.1 shows the comparison of fall detection methods.

Table 2.1: Comparison of Fall Detection Methods

Methods	Advantages	Disadvantages
Ambience device	• Sensitive and responsive to the presence of a human	• Inaccuracy and high rate of false alarms. • Cannot be visually verified by caregiver. • Easily inference by environment.
Vision-based	• More information on the behaviour of a person can be obtained if compare to other methods • Does not require a person to carry any sensor or device	• Privacy concern • Difficulty to monitor the entire house area
Wearable devices	• Less costly • Easily operated • Not easily contaminated by environmental noise	• Needs to wear the gadget in shower, which has a higher occurrence rate of falling. • Old people might forget to wear the device frequently

2.4   Wearable Sensors

Basically, there are two ways for the wearable sensors utilized in the fall detection system. The systems are either using the smart phone's built-in sensors or the sensors mounted on a microcontroller to collect data. However, normally both ways use smart-phones as the platform for alert system. Smart-phones have drawn researchers' attention for its small size and affordable price.

2.4.1   Accelerometers

The accelerometer is commonly used for the measurement of acceleration and speed. Accelerometers are one of the popular types of sensors that frequently applied for fall detection. The shock received by the body upon impact detected by most fall-detection systems using accelerometers. Each system analyses the sensor data of accelerometer and apply it in fall detection by different methods.

Most of the researchers and developers developed a new automatic fall detection system using a single triaxial-accelerometer for data collection [13] [15] [16]. Aguiar et al. [6] and Bai et al. [14] used the smart-phone built in accelerometer in the fall detection system to capture the movement data. However, many false positive results might occur if only focus on large acceleration. This is due to some similarity of acceleration signals between fall activities and fall-like activities, for examples, jumping, running and lying down quickly. Figure 2.4 presents the flow diagram of fall detection.
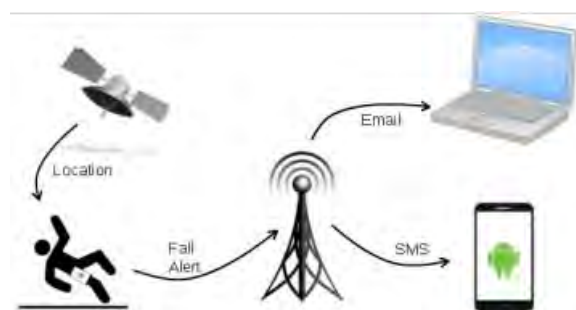
Figure 2.4: Flow diagram of fall detection [6]

2.4.2 Combination of Accelerometer and Gyroscope

An accelerometer-only system has its limitations, such as unable to measure movements with constant velocities. Therefore, some researchers proposed a real time fall detection system using both accelerometer and gyroscope [17][18][19]. In 2011, Jacob et al. [17] introduced a simple method to detect a human fall effectively using an accelerometer and two gyroscopes placed, as a single unit, in three different positions along the thoracic vertebrae. In their work, an accelerometer is utilized in the system to determine the acceleration that exists during human fall while the gyroscope is used to get the measurement of the angle in the fall.

Rakhman et al. [18] and Colón et al. [19] presented an approach for fall detection using accelerometer and gyroscope embedded in the smart-phones to obtain more accurate results of fall detection. In the research of Colón et al. [19], the smart-phone's built in accelerometer and gyroscope are applied in the system to detect the presence of falls and identify the location of the cell phone in the user's body (holster, chest, pocket, etc.). Figure 2.5 shows the axis of the gyroscope and accelerometer.
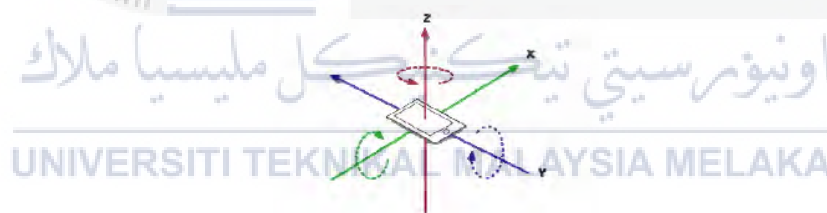
Figure 2.5: Axis of the gyroscope and accelerometer [18]

2.4.3 Micro-electromechanical Systems (MEMS) Based Sensor Module

A micro-electromechanical system (MEMS) is a technology that combines computers with miniaturized electro-mechanical elements such as sensors. In recent years, many researchers started to use the inexpensive and advance MEMS based sensors in the development of the fall detection system to improve the accuracy of fall detection. In 2015, Yuan et al [20] introduced a wearable fall detection system with MEMS based sensor module. The MEMS module used by Yuan et al consists of an ADXL312 triaxial

accelerometer, a BMG160 triaxial gyroscope and a HMC5883L magnetometer. Furthermore, a MEMS based fall detection system was developed by Shi et al [21]. A Micro Inertial Measurements Unit was utilized in this system to gather the raw human movement data from 21 segments of a human body. Figure 2.6 shows a compact size electronic module designed with MEMS-based sensor.

Figure 2.6: Wearable electronic modules [21].

2.4.4 Attitude Heading Reference System (AHRS)

The performance of the fall detection systems is extremely correlated to the measurement of subject orientation. To overcome the limitation of the single triaxial accelerometer and obtain an accurate measurement of subject orientation, some researchers utilized an Attitude Heading Reference System (AHRS) in the fall detection systems [2] [22]. In 2014, Pierleoni et al. [22] presented a high performance AHRS based fall detection system. The AHRS module was used by Pierleoni et al. to analyse the three dimension acceleration and calculate the orientation changes. STM32 with uC/OS-II real-time operating system is used as the main controller of the AHRS module. The hardware of AHRS module mainly consists of a tri-axial accelerometer ADXL345, two biaxial gyroscopes LPY530 and LPR530, a tri-axial magnetometer HMC5883, signal regulate circuit, high frequency filter circuit, watch-dog circuit, A/D reference power circuit and EEPROM data storage circuit. The typical frequency of AHRS module is sampled at 30Hz and able to adjust from 0Hz to 100Hz [22]. Figure 2.7 and Figure 2.8 show the physical structure and frame diagram of AHRS module.

Figure 2.7: Physical structure of AHRS module [22]
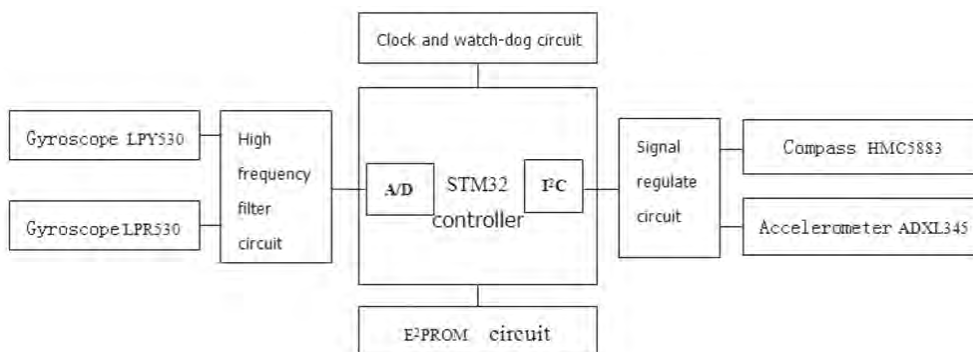
Figure 2.8: Frame diagram of AHRS module [22]

2.5   Sensor Placement

Besides the smart-phones built-in sensors, the literature present different body locations that used for sensors, including: waist [2, 13, 15, 16, 21], back [17], chest [20], feet [21], and ankle [22]. Since the fall detection device is used by a person for the whole day, therefore multiple sensor nodes may not be suitable to attach to a person. In order to reduce unnecessary discomfort, preferably only one sensor node should be placed on a person. The most suitable location for sensors has been investigate to obtain optimum and better fall detection performance.

In 2016, Özdemir [23] proposed a paper about the analysis on different body locations for wearable fall detection sensors. Several tests were carried by Özdemir with different sensor placements. The subject's motions were recorded with six groups of sensors, each with a tri-axis sensor (accelerometer, gyroscope and magnetometer), which is located on different body parts with special straps: head, chest, waist, right-wrist, right-thigh and right-ankle. The performance of single sensor groups with their location was calculated to identify the most satisfactory location for the sensor placement on the human body. Figure 2.9 presents the location-based average accuracies for single sensor units. Based on Figure 2.9, the sensor placed in waist positions achieved the best

performance with accuracy of 98.42%, followed by the thigh sensor with an accuracy of 97.89%. The sole reason for the best performance of the sensor unit on the waist position may due to the waist is the centre of gravity on the human body and truly reflects the posture of the trunk. Table 2.2 presents the summary on sensor placement and wearable sensors.

Figure 2.9: Location-based average accuracies [23]

Table 2.2: Summary on sensor placement and wearable sensors

Authors	Sensors Placement	Wearable Sensors
Nuttaitanakul and Leauhatong [13]	waist	3-axis accelerometer
Vallejo et al. [15]	waist	3-axis accelerometer
Dinh and Struck[16]	waist	3-axis accelerometer
Aguiar et al. [6]	Smart-phone	3-axis accelerometer
Bai et al. [14]	Smart-phone	3-axis accelerometer
Jacob et al. [17]	back	3-axis accelerometer and 3-axis gyroscope
Rakhman et al. [18]	Smart-phone	3-axis accelerometer and 3-axis gyroscope
Colón et al. [19]	Smart-phone	3-axis accelerometer and 3-axis gyroscope
Yuan et al. [20]	chest	MEMS-based sensor
Shi et al. [21]	Feet and waist	MEMS micro IMU module
Zhang et al. [2]	waist	AHRS module
Pierleoni et al. [22]	ankle	AHRS module

2.6 Fall Detection Algorithms

This section reviews fall detection algorithms based on neural network and the threshold based analysis methods which are widely applied in the fall detection system. Table 2.3 summarizes the paper on the fall detection system.

2.6.1 Wavelet Transform and Neural Network

In the paper presented by Zhang et al [2] and Nuttaitanakul et al. [13], the fall detection algorithm consists of 3 major steps which are signal acquisition, wavelet transform and activity classification. The block of proposed algorithm is presented in Figure 2.10.
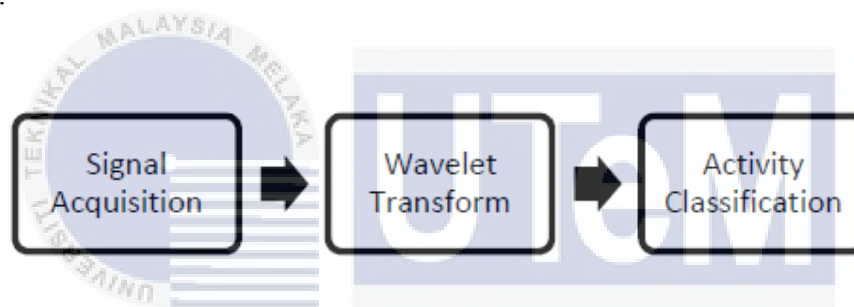
Figure 2.10: Block of proposed algorithm [13]

After the signals of the falls were gathered, the Daubechies wavelets were used in the fall detection system for feature extraction of the acceleration signals. The environment influences and restrictions of processing technology have created the signal noise for the data of sensors like accelerometer, gyroscope and magnetic compass. Zhang et al [2] reduced the signal noise of the collected sensor data with wavelet transform to improve the experiment data. The wavelet transform is a general and useful mathematical tool for feature extraction and noise reduction.

A signal can be constructed from its approximation and formula as follows [13]:

$$x(t) = \sum_{k=0}^{2^{N-j}-1} a_{j,k} 2^{-\frac{j}{2}} \emptyset(2^{-j}t - k) + \sum_{j=1}^{j} \sum_{k=0}^{2^{N-j}-1} d_{j,k} 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \qquad (2.1)$$

detailed coefficients of φ and ψ functions respectively. j and k are integer values which represent dilation and translation parameters of the transform. a and d coefficients can be calculated as follows:

$$a_{j,k} = \left(x(t), \emptyset_{j,k}(t) \right) = \sum_n h(n - 2k) a_{j-1,n} \qquad (2.2)$$

$$d_{j,k} = \left(x(t), \psi_{j,k}(t) \right) = \sum_n g(n - 2k) a_{j-1,n} \qquad (2.3)$$

where $h(n)$ is the low-pass filter of the scaling function, and $g(n)$ is the high-pass filter of wavelet function. These filters are constructed from the selected wavelet function and its corresponding scaling function as follows:

$$\emptyset_{j,k}(t) = \sqrt{2} \sum_n h(n) \emptyset(2t - n) \qquad (2.4)$$

$$\psi_{j,k}(t) = \sqrt{2} \sum_n g(n) \emptyset(2t - n) \qquad (2.5)$$

Figure 2.11 and Figure 2.12 show the original data of accelerometer before reduction of noise and processed data of wavelet after noise reduction respectively.
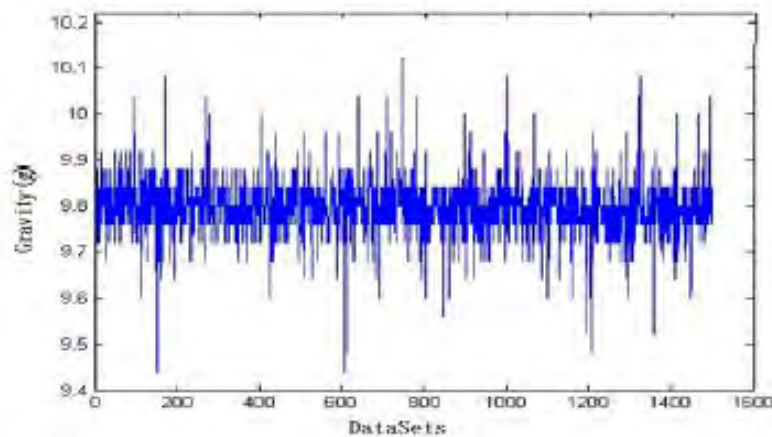
Figure 2.11: Z axis acceleration before noise reduction [2]
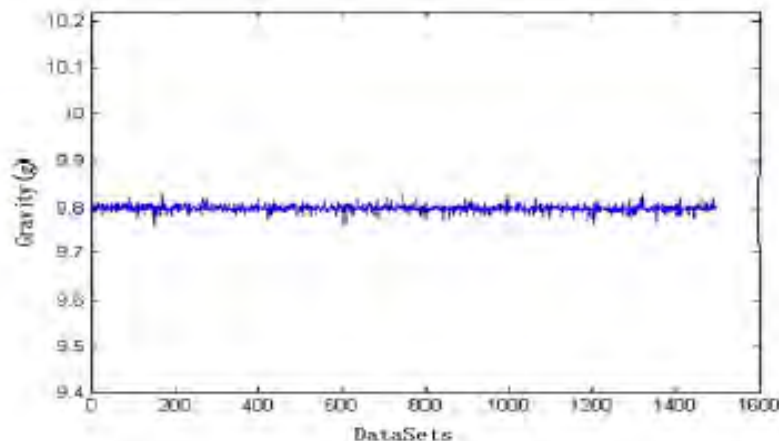
Figure 2.12: Z axis acceleration after noise reduction [2]

In the activities classification section, the Multilayer Perceptron neural network is used by Zhang et al [2] and Nuttaitanakul et al. [13] to categorize the activities of the decomposition signals. The MLP neural network is a very famous neural network applied for problem solving and pattern classification. Figure 2.13 shows the multilayer neural network structure.
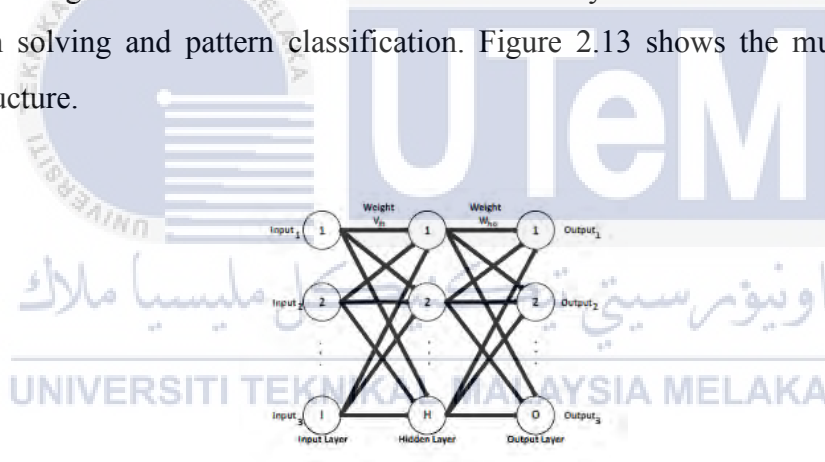
Figure 2.13: Multilayer neural network structure [13]

To design a BP neural network algorithm, there are several the processes involved. Firstly, the network weights are initialized with random values by a uniformly distribution whose mean is zero. The variance is selected by the rule that the standard deviation of induced local field is located in the transition area of the linear path, and the saturated part is in the sigmoid function. A sigmoid function is a mathematical function which has an "S" shaped curve and defined by the formula, $(x) = \frac{1}{1+e^{-\beta x}}$. Figure 2.14 presents the figure of sigmoid curve.

$$f(x) = \frac{1}{1+e^{-\beta x}}$$
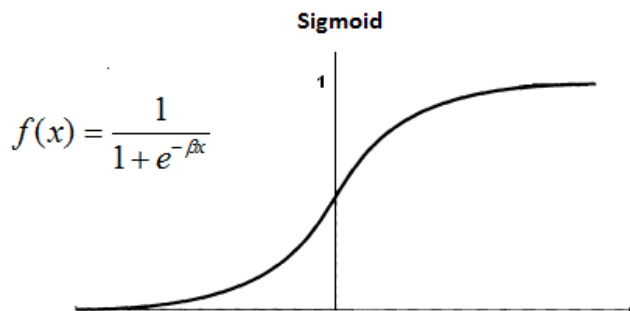
Figure 2.14: Sigmoid curve [13]

Secondly, a set of training input pattern which collected with different fall and ADL is applied to the input layer of the neural network. Next, the input sample sets propagated forward from layer to layer until the output pattern or signal is generated by the output layer. The convergence criteria of BP neural network are satisfied if the difference between the generated output signal and desired output signal fulfil the error threshold requirement. If the convergence criteria not satisfied, an error is calculated and then propagated backward from the output layers to the input layer for the weight adjustment in each layer. The weights are adjusted until the entire error signals reach the convergence criteria. The neural network structure of the fall detection system is shown in Figure 2.15.
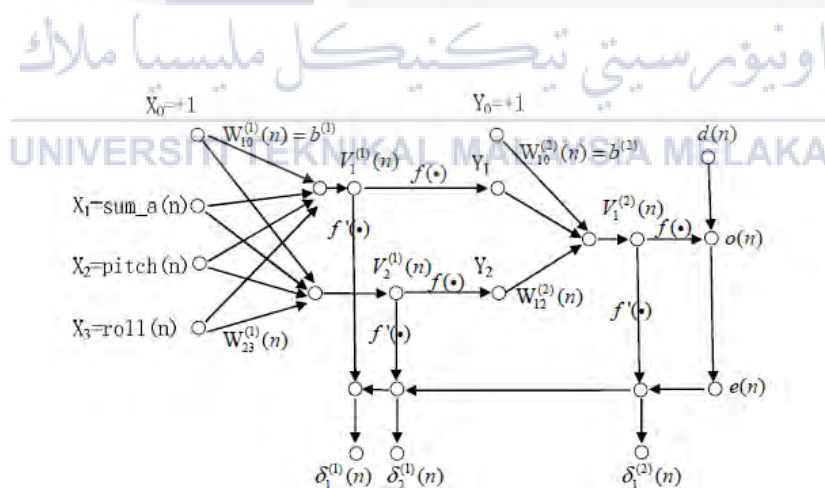
Figure 2.15: Neural network structure of the fall detection system [13]

2.6.2   Threshold Based Method

Most of the analyses applied are based on the use of a threshold in the signals. A fall will be detected if the threshold is exceeded. However, this approach is considered complicated in finding a suitable threshold value to achieve accurate detection on all types of falls. There are different results and performances presented on the use of threshold in the fall detection. In the literature, some researchers used a threshold for the accelerations, the angle of inclination or a combination of several thresholds. There is a high accuracy of fall detection result had achieved by some of these systems. However, most of the threshold-based fall detection have not tested under real-life conditions and there is no confirmation that these techniques function well with different subjects.

Most of the threshold based algorithms in the fall detection system makes use of triggering of the accelerations to distinguish between ADL and falls. Bai et al [14] analysed the measurement data obtained from a smart-phone's accelerometer to determine falls. The acceleration was measured with a 3-axis accelerometer and the sum vector were calculated with formula 2.6.

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \qquad (2.6)$$

where a is the sum-vector of axial accelerations, a_x is the acceleration on the x axis, a_y is the acceleration on the y axis and a_z is the acceleration on the z axis.

However, there are some limitations for this kind of method for fall detection. In real life, there are some easy and simple motions (such as jumping and running fast) that might be associated with high accelerations although falls normally have higher accelerations than other activities. Besides that, some falls (such as falls beginning with the person sitting on a chair or when the person reaches to hold on from something before falling) may have lower values of acceleration [15].

2.6.3 Multi-threshold Based Method

Some researchers utilized two or more sensors like accelerometer and gyroscope and developed a multi-thresholds algorithm for effective fall detection. In 2011, Jacob et al. [17] applied a multi-thresholds algorithm in the fall detection system to increase the effectiveness of fall detection. The multi-thresholds algorithm able to differentiate between ADL and falls using the resultant gravitational acceleration, angular change, angular velocity, and angular acceleration collected from the accelerometer and gyroscopes. However, there are some difficulties to obtain the appropriate threshold values. The formula for the calculation of the resultant gravitational acceleration, resultant angular velocity, resultant angular acceleration and total angular change are shown below [17].

$$Resultant\ Gravitational\ Acc. = \sqrt{A_x{}^2 + A_y{}^2 + A_z{}^2} \tag{2.7}$$

$$Resultant\ Angular\ Velocity = \sqrt{x^2 + y^2 + z^2} \tag{2.8}$$

$$Resultant\ Angular\ Acc. = \sqrt{(\frac{d}{dt}x)^2 + (\frac{d}{dt}y)^2 + (\frac{d}{d}(z))^2} \tag{2.9}$$

$$Total\ Angular\ Change = \sum \sqrt{x^2 + y^2 + z^2} \tag{2.10}$$

2.6.4 Attitude Angles Based (*Kalman filter*)

To reduce the false fall detection rate, Yuan et al [20] proposed an attitude angles based fall detection algorithm. In their work, three attitude angles (pitch, roll and yaw) of the wearable device fixed on the user are used to detect human fall. The Kalman filter algorithm was used by Yuan et al to determine and obtain the attitude angles. The Kalman filter is a common algorithm which used for attitude determination. The attitude angles for fall detection are quite complex but they have the ability to distinguish falling direction. The attitude angles can be determined through the simplified discrete Kalman filter as follows [20]:

$$X(n+1) = F(n)X(n) + w(n)$$
$$Y(n) = h[X(n)] + v(n) \tag{2.11}$$

$$\begin{cases} \hat{x}_n^- = F_{n-1}(\hat{x}_{n-1}^+) \\ P_n^- = F_{n-1}P_{n-1}^+F_{n-1}^T + W_{n-1}Q_{n-1}W_{n-1}^T \end{cases} \tag{2.12}$$

$$\begin{cases} K_n = P_n^- H_n^T \left(H_n P_n^- H_n^T + R_n^- \right)^{-1} \\ P_n^+ = (I - K_n H_n) P_n^- \\ \hat{x}_n^+ = \hat{x}_n^- + K_n \left[y_n - h_n(\hat{x}_n^-) \right] \end{cases} \tag{2.13}$$

Figure 2.16 shows the comparison results for real five kinds fall test (right fall, left fall, back fall, front fall and vertical fall) between accelerometer-based, gyroscope-based and attitude angle based.
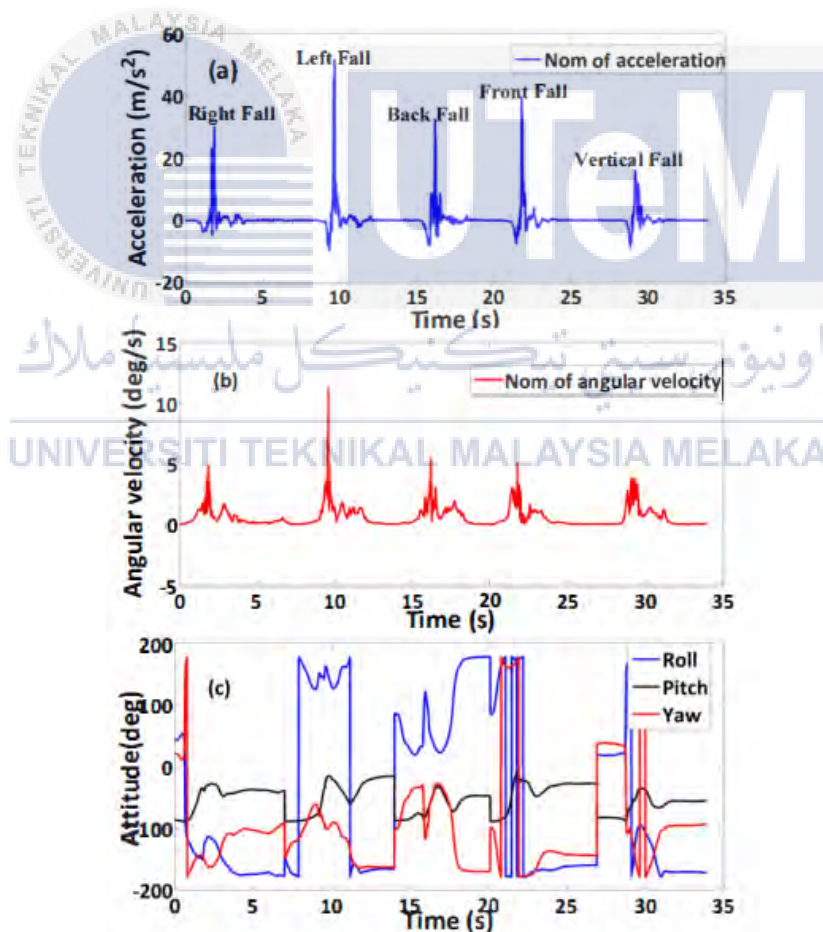
Figure 2.16: Test results for five types fall: (a) accelerometer-based, (b) gyroscope-based, (c) attitude angles-based [20]

Table 2.3: Summary of papers on fall detection.

Authors	Sampling Frequency	Algorithm	Falls	ADL	Sensitivity	Accuracy	Specificity
Nuttaitanakul and Leauhatong [2015]	200Hz	Waveform transform and multilayer perceptron neural network	Forward, backward, right side, left side, falling while standing	Walking, standing up from a chair, sitting down on a chair, lying down on a bed, getting up from a bed	-	85.6%	-
Vallejo et al. [15]	-	Artificial Neural Network	-	-	98.4%	98.74%	98.6%
Dinh and Struck[16]	100Hz	Neural network	Forward, backward, sideward and collapse	-	94%	90.3%	99.65%
Aguiar et al. [6]	67Hz	Decision Tree	-	-	97%	97.5%	99%
Bai et al. [14]	-	Threshold based	-	Running and jumping	-	96%	-
Jacob et al. [17]	1000Hz	Multi Threshold based	Forward, Sideways, forward knee flexion, sideways to the left, sideways to the right	Sitting down on a chair, standing up, walking with normal speed	100%	-	-
Rakhman et al. [18]	-	Threshold based	Forward, backward, left, right	Walk, run, sit down quickly, lying on the bed, bow, walk up stairs, walk down stairs	-	98%	-
Colón et al. [19]	-	Threshold based	Forward, backward, right, left	Walking	89%	81.3%	79%

Authors	Sampling Frequency	Algorithm	Falls	ADL	Sensitivity	Accuracy	Specificity
Yuan et al. [20]	100Hz	Kalman filter	Forward, backward, right and left	Walk, sitting, running, sitting down quickly, lying down quickly	-	-	-
Shi et al. [21]	-	J48 decision tree	Forward, backward, leftward, rightward, bend to fall, from stand to squat fall, from squat to stand fall, left foot forward slip, left foot backward slip	Bending, downstairs, lift-rising, lift-down, running, turn around, upstairs, walking, from stand to sit, from stand to squat, from squat to stand, from sit to squat	98.9%	97.792%	98.8%
Zhang et al. [2]	30Hz	Wavelet transform and neural network	Forward, lateral and backward	Walk, jump, sit, squat, bend and lay	-	98.2%	-
Pierleoni et al. [22]	-	Threshold based	Forward, backward, lateral fall to the left, lateral fall to the right, syncope	Lying on bed, walking, sitting on a chair, climbing two steps, standing after picking something	97%	98.52%	100%

2.7   Heart Rate and Body Temperature Monitoring System

Pulse rate and body temperature monitoring system is simply a system or device that calculates the heart beats per minute and measure the body temperature of human. However, timely measurement of the pulse rate and body temperature is important for better medical treatment. Mansor et al. [24] developed a health monitoring system named Home-smart Clinic. The designed system utilizes a temperature sensor, pulse rate sensor, Arduino microcontroller with Ethernet shield, and wireless communication device (Xbee) as the main components. The system consists of three main units, which are data acquirement unit, data processing unit and data communication unit. Data acquirement unit of the system consists of temperature sensor and pulse rate sensor for measuring and patient monitoring purpose. The measuring sensors of the system will connect to the Arduino with Ethernet shield to send the data to the data processing unit (database). This system communicates between each part in a real time monitoring, processing and reporting. From the diagnostic analysis, the doctor can access into the remote database to obtain the information of the pulse rate and body temperature of their patient(s). Figure 2.17 presents the block diagram for Home-smart Clinic.
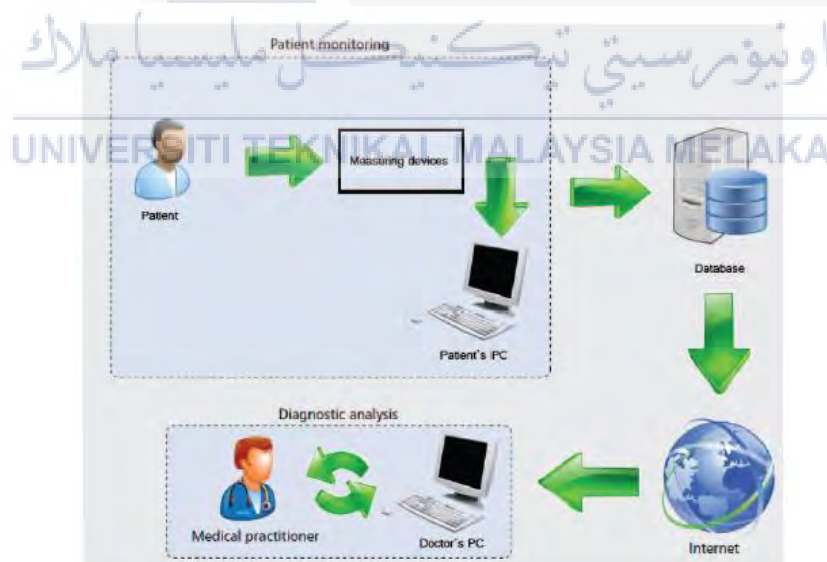
Figure 2.17: Block diagram for the Home-smart Clinic [24]

Asaduzzaman Miah et al.[25] proposed an embedded system for heart care monitoring purpose using Arduino UNO. An infrared Tx and Rx have been used in this system to measure the pulse by measuring the change in blood flow through one of the fingers. To filter out the unwanted noise and environmental interference, a noise filter has been designed by the author. A temperature sensor is used in this system for the measurement of body temperature. The information of pulse rate and body temperature will send to the android application for monitoring. The Arduino UNO microcontroller is connected to Bluetooth module. The Arduino UNO is programmed to calculate and measure the pulse rate and body temperature. The measured pulse rate and body temperature will send to the developed "Heartmate" android application via Bluetooth. Figure 2.18 shows the system block diagram of the system, Figure 2.19 presents the functional block diagram of the system.
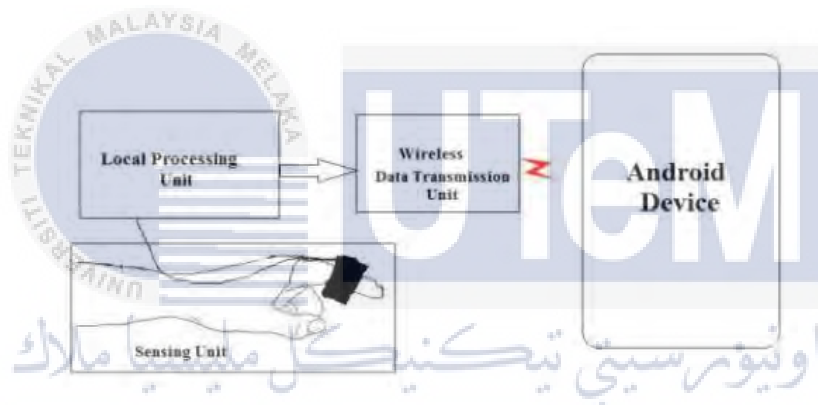
Figure 2.18: System block diagram of proposed system [25]
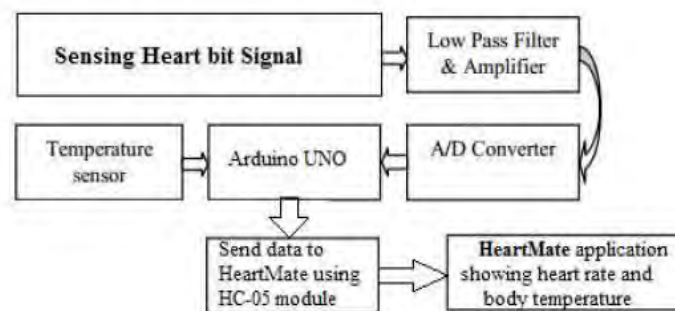
Figure 2.19: Functional block diagram of proposed system [25]

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will discuss the methodology of the project in detail. Project methodology is defined as a system of methods or procedure to successfully achieve the goals of the project. In this chapter, all the techniques and knowledge applied in the project will be described with table, diagram or detailed explanation.

3.2 System Flow Chart

The system flowchart lists out the process of the system in sequential order. This system flowchart also acts as a guideline for the methodology of this project. Figure 3.1 presents the system flow chart of the human fall detection and alert system.

Figure 3.1: System flow chart

3.3   Project K-Chart

K-Chart is a tool for systematically organizing research's scope of the system under study, methodology and results timelines in the form of a Tree Diagram. Figure 3.2 shows the K-chart for this project.
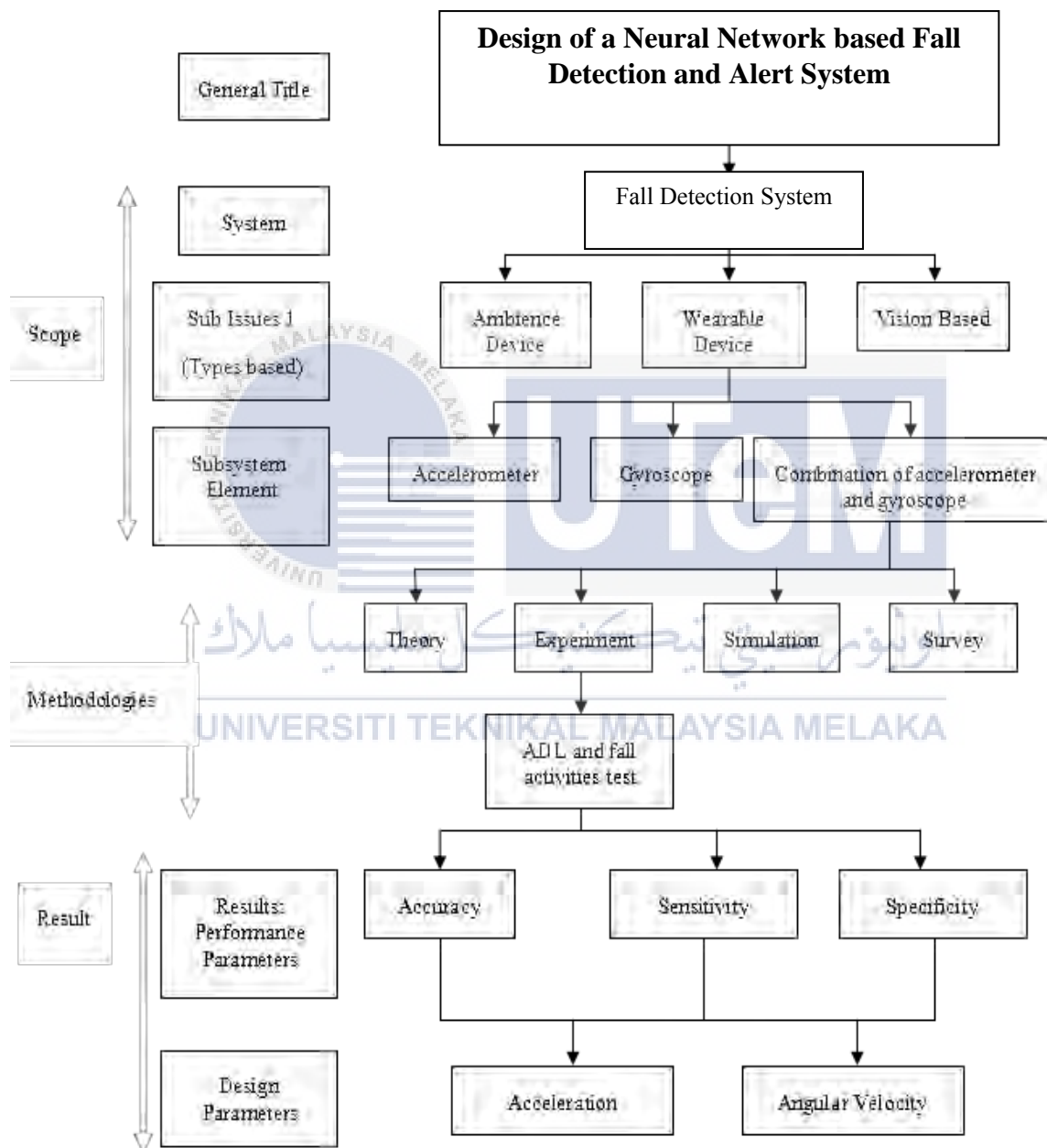
Figure 3.2: K-chart of fall detection and alert system

3.4 System Block Diagram

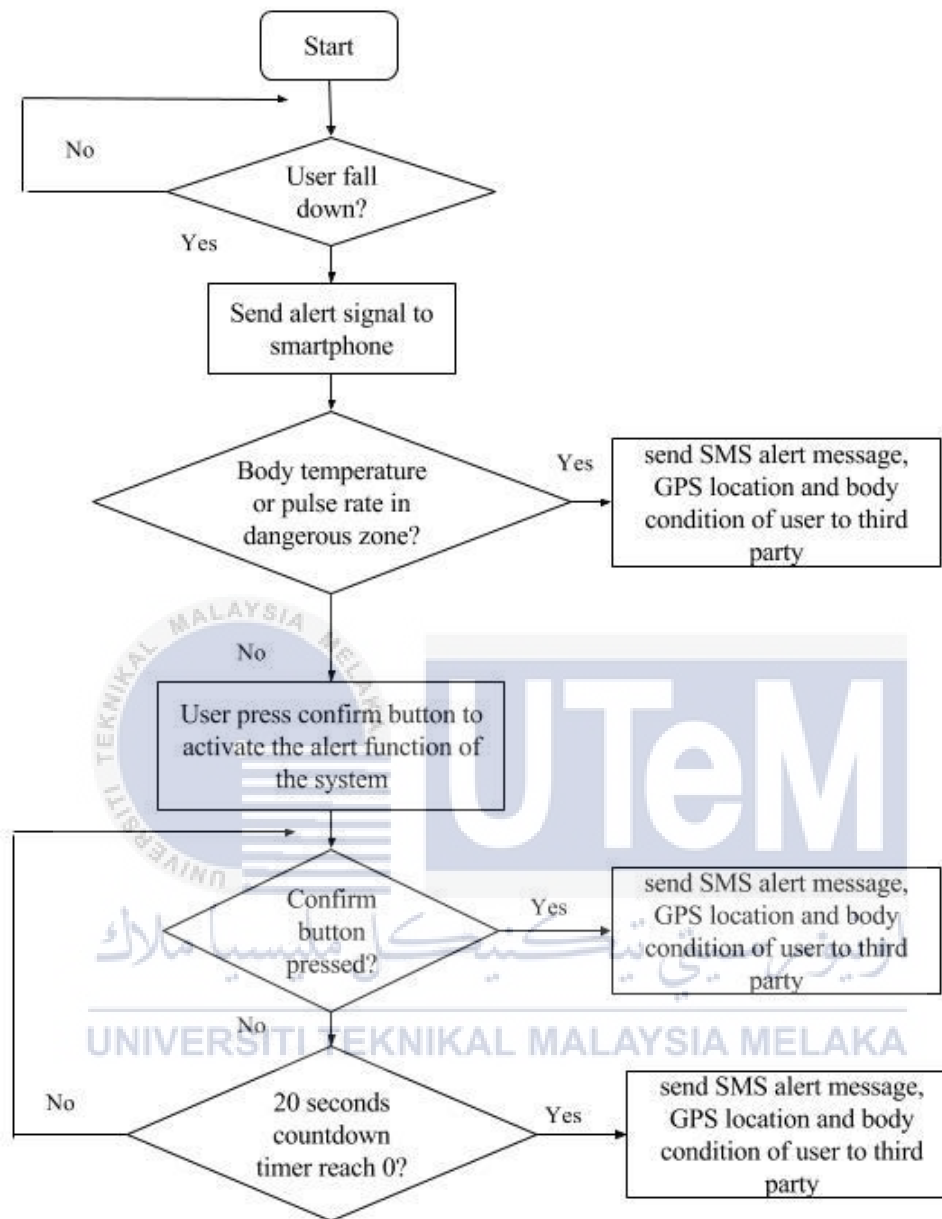Figure 3.3 presents the system block diagram of the system. The block diagram demonstrates an overview of the designed human fall detection system. A GY-80 10 DOF IMU module is mounted to the Intel Edison with Mini Breakout Board and worn on the waist of the human body. The module must be rigidly attached to the waist of the target in order to eliminate errors in the measurement. The sole reason for the best performance of the sensor unit on the waist position may due to the waist is the centre of gravity on the human body and truly reflects the posture of the trunk. The vital sign monitoring device which consists of LM35 temperature sensor and pulse rate sensor will be worn on the wrist of the user to measure the body temperature and pulse rate. The vital sign monitoring device will send the information on body condition of the user to the smart-phone when fall activities detected. Once the fall accident happens, the alert system will trigger and send emergency messages, body conditions and actual location of the user to his parents or friends.
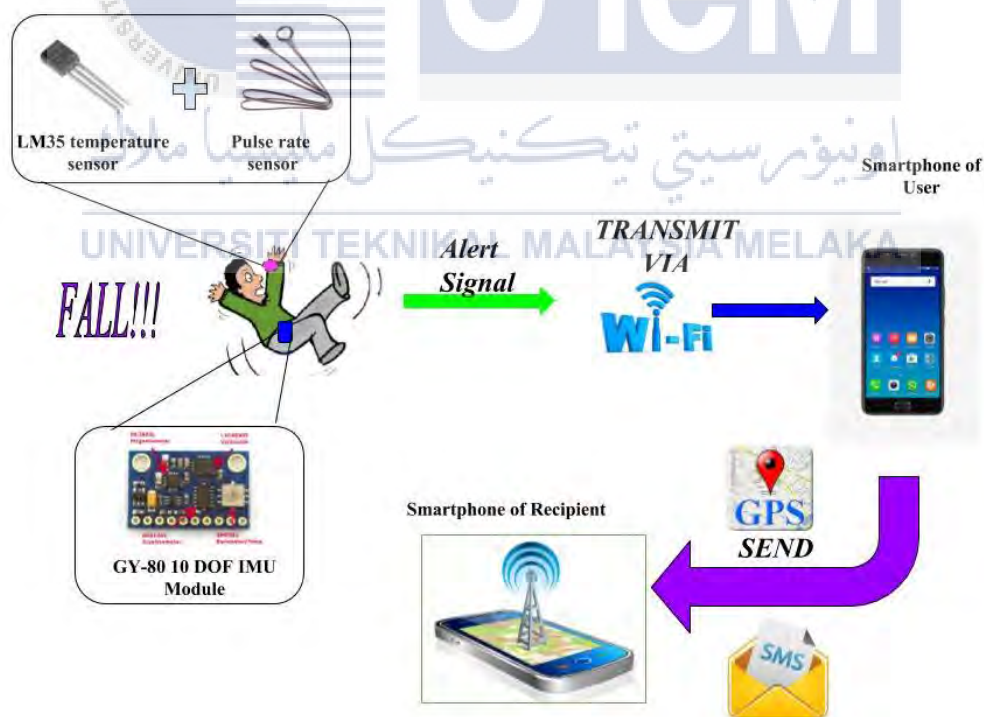
Figure 3.3: Block diagram of the human fall detection and alert system

3.5   Hardware Description for Fall Detection Device

The hardware platforms worn by subjects for fall detection are the Intel Edison with Mini Breakout Board and GY80-10 DOF Inertial measurement unit modules (GY80-IMU module). This section explains the details and specifications of the components used in the fall detection device.

3.5.1   Intel® Edison with Mini Breakout Board

The Intel Edison board is a small size processor, which designed with a large amount of tech features while still delivering the same robust strength of your go-to single board computer. This Edison board is powered by the Intel® Atom™ SoC dual-core CPU and consists of an integrated Wi-Fi, Bluetooth LE, and a 70-pin connector to attach a veritable slew of shield-like "Blocks" which can be stacked on top of each other [26]. This board has a low power consumption and small footprint. Therefore, it is suitable to utilize in the system that require high processing power, but with minimum power source and small footprint. The Intel® Edison with Mini Breakout Board is shown in Figure 3.4.

Figure 3.4: Intel® Edison with Mini Breakout Board [26]

3.5.2   GY80-10 DOF Inertial Measurement Unit Module

The GY80-10 DOF Inertial measurement unit modules (GY80-IMU module) mounted to the NodeMCU microcontroller to be used for data acquisition when human falls occurred. GY80-10 DOF Inertial measurement unit module consists of a L3G4200D (3-Axis Gyroscope), ADXL345 (3-Axis Digital Accelerometer), HMC5883L (3-Axis Magnetometer) and BMP085 (Barometer). The This module has a dimension of (25.8 x 16.8) mm and 3mm installing hole. For this project, only the accelerometer and angular rate sensor will be used to collect falling data. Although most of the smart phones consist of the integrated accelerometer and gyroscope, there are some difficulties for the real-time monitoring of the activities of the elderly using those sensors. Moreover, we can make further improvements using more sensors in our system if IMU module is used in this project. Therefore, external sensors might more appropriately apply the system. Figure 3.5 shows the figure of GY80 IMU module. Table 3.1shows the features of GY80 IMU module.
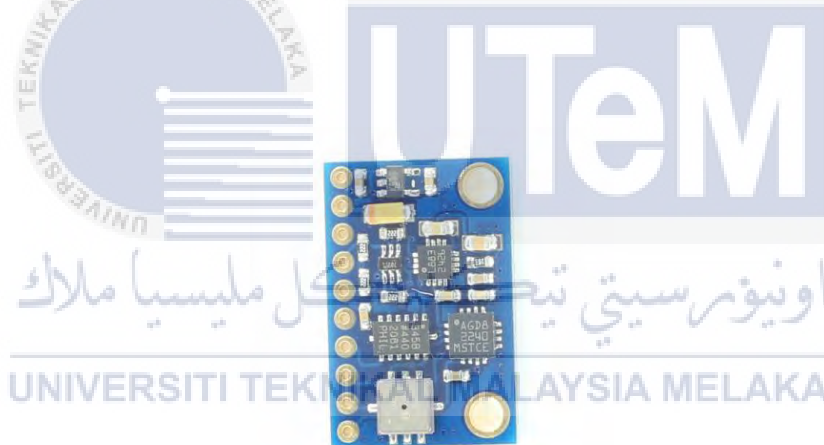

Figure 3.5: GY80-10 DOF Inertial measurement unit module [27]

Table 3.1: Features of GY80-10 DOF IMU

Feature	Specifications
Chips	-L3G4200D (3-Axis Angular Rate Sensor)
	-ADXL345 (3-Axis Digital Accelerometer)
	-HMC5883L (3-Axis Magnetometer)
	-BMP085 (Barometric Pressure Sensor)
Power supply	3-5V
Communication	IIC communication protocol (fully compatible 3-5 V system)
Module size	25.8 mm x 16.8 mm, installing hole 3 mm
Standard	2.54 mm put the needle interface, convenient bread plate porous plate experiment connection

3.5.2.1 ADXL345 (3-Axis Digital Accelerometer)

The tri-axial accelerometer (ADXL345) from GY-80 IMU module has a high resolution (13-bit) measurement at up to ±16 g. A Micro Electro Mechanical Systems (MEMS) accelerometer behaves as a damped mass on a spring which is displaced when it experiences acceleration. The acceleration of the sensor is determined by using the measurement from the mass displacement. An accelerometer at rest measures an acceleration of $g = 9.81$ m.s^{-2} (1g) straight upward due to its weight. A measurement of zero will be given when an accelerometer in free-fall. The datasheet of ADXL345 accelerometer is shown in Appendix B. Figure 3.6 presents the axes measurement of the accelerometer.
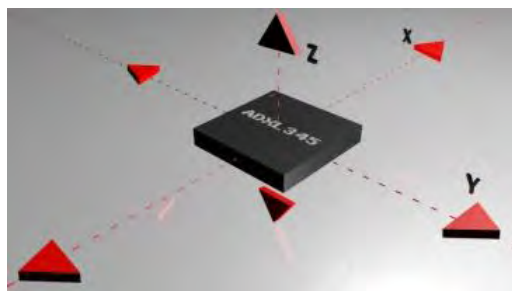

Figure 3.6: Axes measurement of accelerometer [28]

3.5.2.2 L3G4200D (3-Axis Angular Rate Sensor)

The L3G4200D (3-Axis Angular Rate Sensor) is a type of tri-axial gyroscope which able to provide unprecedented stability of the zero rate level and sensitivity over temperature and time with three selectable full scales (250/500/2000 degrees per second). The operating principle of MEMS gyroscopes is based on using a vibrating mechanical element to sense rotation. When angular velocity is applied to a gyroscope, two masses within the sensor oscillates, thus making the Coriolis force (Coriolis force is a force experienced in a rotating reference frame and is proportional to the rate of rotation) on each mass to act in the opposite direction, which results in a change in capacitance. The datasheet of L3G4200D gyroscope is shown in Appendix B. Figure 3.7 presents the axes measurement of the gyroscope.
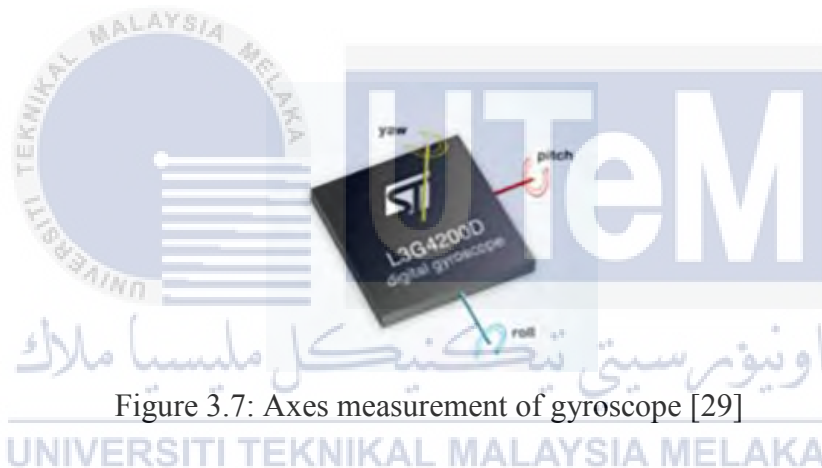
Figure 3.7: Axes measurement of gyroscope [29]

3.6   Hardware Description for Vital Sign Monitoring Device

For the vital sign monitoring device that included in this system, a LM35 temperature sensor and pulse rate sensor module are connected to NodeMcu Lua ESP-12E ESP8266 Wi-Fi Development Board Ver2 to measure the body temperature and pulse rate of the subjects. This section explains the details and specifications of the components used in the vital sign monitoring device.

3.6.1 NodeMcu Lua ESP-12E ESP8266 Wi-Fi Development Board Ver2

NodeMCU is an open source platform for Internet of Things (IoT). It consists of ESP-12 module based hardware and firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems. The term "NodeMCU" by default refers to the firmware rather than the development kits. The programming of firmware is based on the Lua scripting language. However, NodeMCU also able to be programmed using Arduino IDE. Figure 3.8 shows figure of NodeMcu Lua ESP-12E ESP8266 Wi-Fi Development Board Ver2.

Figure 3.8: NodeMcu Lua ESP-12E ESP8266 WIFI Board Ver2 [30]

3.6.2 Pulse Sensor

Pulse Sensor is utilized in our system to measure the heart rate of the user. The pulse sensor is a well-designed plug-and-play heart-rate sensor for Arduino. This sensor is commonly used by the students and researchers to easily incorporate live heart rate data into their projects. The sensor clips onto a fingertip or earlobe and connects to the microcontroller with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time. Figure 3.9 presents the figure of pulse sensor.

Figure 3.9: Pulse sensor [24]

3.6.3 LM35 Temperature Sensor

The LM35 temperature sensor is utilized as a part of our system to quantify the body temperature of the human body. LM35 temperature sensor is precision integrated-circuit temperature sensor which has an operating temperature over a −55˚ to +150˚C temperature range. The output voltage of LM35 is directly proportional to the Celsius (Centigrade) temperature. Moreover, no external calibration or trimming needed for the LM35 to achieve typical accuracies of ±1⁄4˚C at normal temperature and ±3⁄4˚C over a full −55 to +150˚C temperature range. Due to the low output impedance, linear output, and precise inherent calibration, the interfacing to readout or control circuitry of the LM35 become easier. It can be used with single power supplies, or with plus and minus supplies. Since LM35 only draws 60 µA from its supply, therefore it has very low self-heating, less than 0.1˚C in still air. The datasheet of LM35 temperature sensor is shown in Appendix D. Figure 3.10 presents the figure of LM35 temperature sensor.
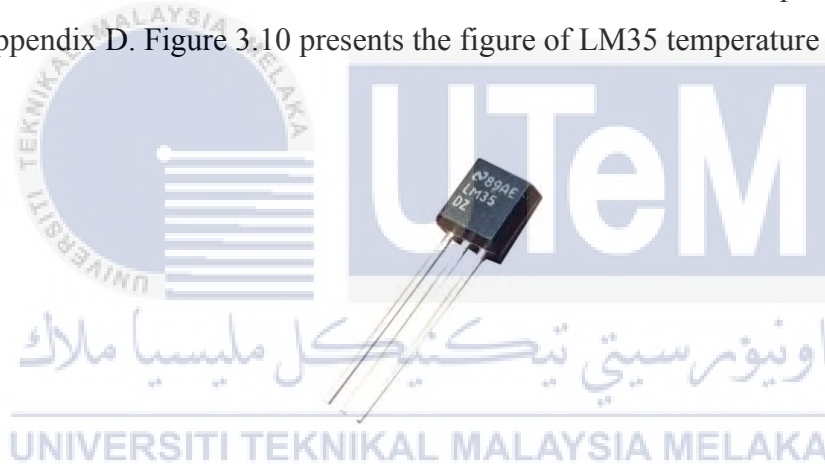
Figure 3.10: LM35 temperature sensor [25]

3.7   Software Description

3.7.1   MATLAB R2013a

MATLAB (MATrix **LAB**oratory) is a multi-paradigm numerical computing environment which developed by MathWorks. MATLAB is also a type of fourth generation programming language that normally used to solve linear algebra type problems and perform numerical computations. This program has a variety of toolbox available and able to perform several kinds of functions. Moreover, MATLAB can interfaces with programs written in diffirent programing languages, including C, C++, C#, Java, SQL,

Fortran, Python etc. Due to its advantages, MATLAB is widely used in engineering industry and academia. In this project, MATLAB is mainly used for plotting of data and implementation of fall detection algorithms.

3.7.2 Arduino IDE

The Intel Edison board is programmed using the Arduino IDE Software. Arduino IDE Software is an integrated development environment (IDE) which provides a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. Arduino hardware or other related hardware connected to Arduino IDE to upload programs and communicate with them.

3.7.3 Android Studio

Android Studio is the official software that consists of code editing, troubleshooting, testing, and performance tooling for Android platform development. The Android Studio is also an integrated development environment (IDE) which designed with the purpose of helping us to develop the highest-quality apps and accelerate the development for every Android device.

3.8 Design Process for Neural Network

An artificial neural network algorithm was designed to detect falls and classify the movement patterns. The training patterns were recorded and interpreted manually with 0 (no fall containing) and 1 (fall containing). Back propagation neural network (BPNN) was used as the training algorithm for the multilayer neural network. The output of the neural network was triggered by 0.45 to detect a fall [25]. If this threshold is greater than two times one after the other, an alarm is generated. The neural network algorithm was created, train, visualize, and simulate using Neural Network Toolbox of MATLAB software.

The design process for the neural network includes five major steps:

1. Data collection

2. Signal Processing

3. Configure the network

4. Initialize the weights and biases

5. Train the network

3.8.1   Fall and ADL Data Collection for Neural Network Training

In order to develop the fall detection algorithm and validate the proposed solution, two tests were carried out with the wearable electronic module placed on each subject's waist as shown in Figure 3.11. Test 1 is the activities of daily living (ADL) experiments while Test 2 is the fall activities experiments. The ADL included in the Test 1 is walking, sitting, running, jumping and lying down. The simulations of fall activities performed were fall forward, fall backward, fall left and fall right. Since it is not suitable for subject elderly people to perform falls, therefore the subject group included healthy young volunteers to simulate falling and ADL tasks. Figure 3.12 shows the data collection setup for both tests.

A group of 15 subjects which ages 20 to 40 years old was volunteered to simulate the falls. The weight varied from 45kg to 90kg, while the height was from 160cm to 185cm. Each of them performed each of the 4 different falls and 5 different ADL, giving a total of 90 (10x4+10x5) data sets. Acceleration and gyroscope data (in three dimensions) were gathered from accelerometer and gyroscope of the GY80-IMU module. The collected data were sampled at 100 Hz and recorded in Micro SD card for further processing. The Arduino coding for data collection is shown in Appendix E. Table 3.2 and 3.3 present the information for each test. Figure 3.13 shows the simulations of fall.
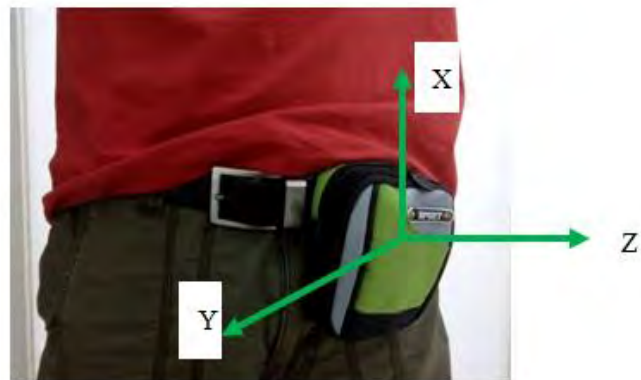
Figure 3.11: Sensor Placement
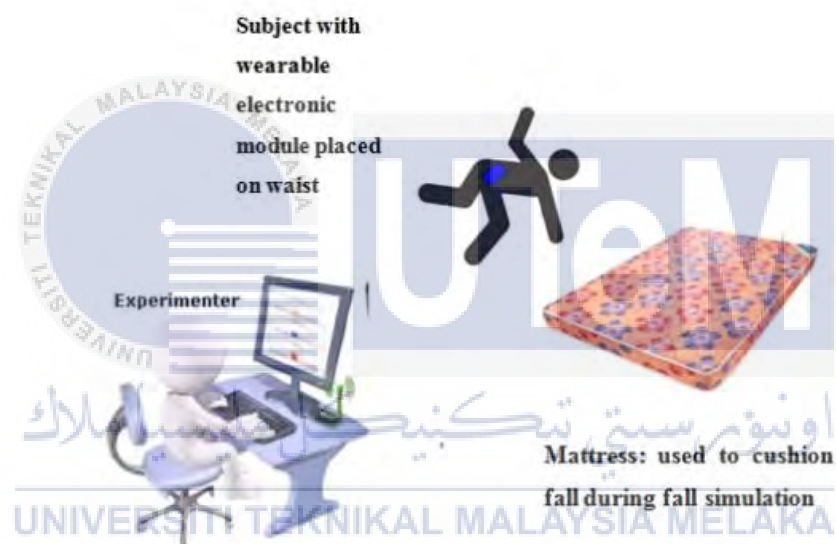
Figure 3.12: Data collection set-up

Table 3.2: Test 1 (ADL)

Subjects	15
Ages	20-40
Heights	160-185cm
Activities	Walk, Sitting, Running, Jumping, Lying Down

Table 3.3: Test 2 (fall activities)

Subjects	15
Ages	20-40
Heights	160-185cm
Activities	Fall forward, fall backward, fall left, fall right

Figure 3.13: Simulations of fall

3.8.2   Signal Processing

Signal processing of the signals from the GY80-IMU module such as transformation, filtering, and visualization will be performed using Signal Processing Toolbox of MATLAB software. The toolbox consists of algorithms for resampling, smoothing, and synchronizing signals, designing and analysing filters, estimating power spectra, and measuring peaks, bandwidth, and distortion. The toolbox also comprises parametric and linear predictive modelling algorithms. The Signal Processing Toolbox can be used for signal analysis in time, frequency, and time-frequency domains, patterns and trend identification, feature extraction, and custom algorithm development and validation.

3.8.2.1 Filtering and Feature Extraction

The collected data were filtered to reduce noise and compensate for gyroscope drift. In the latter case, a high-pass filter was applied to the angular velocity data from the gyroscope. The angular velocity is high-pass filtered with a second-order Butterworth filter. The gyroscope data are prone to drift and the angle derived from angular velocity continue to change during integration even when the sensor is stationary.

During fall detection, the acceleration or rotation of the body is not necessary is important in any specific dimension that is important. Therefore, some aggregate of the 3 dimensional sensor output may be suitable to be considered for the analysis. For this reason, Vector Magnitude (VM) is used as a data feature for feature extraction; VM can be calculated with the formula below:

$$\text{VM} = \sqrt{x^2 + y^2 + z^2} \qquad (3.1)$$

where x, y and z are the sensor readings for each axis.

3.8.3   Artificial Neural Network (ANN)

Artificial neural network (ANN) is an information-processing system based on the human brain. An ANN is a model of reasoning that has certain performance characteristics in a similar manner with biological neural networks. It consists of a set of interconnected simple processing units (neurons or nodes) which combine to produce output signals based on the received input signals for pattern classification and problem solving. The interconnected simple processing units have adjustable gains that are slowly adjusting through iterations influenced by the input-output patterns or signals presented to the ANN [31].

Basically, it is a supervised learning algorithm that requires a set of input pattern for classification training. The performance of an ANN is described by the figure of merit, which expresses the number of recalled patterns when input stimulus signals are applied, that could be complete, partially complete, or even noisy. When the ANN always outputs a desired pattern for every trained input pattern, it means the recalled pattern has achieved 100% performance.

Back propagation neural network is a common method of learning algorithm used in multilayer neural networks. The BP algorithm is a popular neural paradigm and widely used in recognizing patterns. In 1985, the PDP research group led by Dave Rumelhart proposed the BP algorithm based on the generalized delta rule [24]. The limitations of the perceptron algorithm able overcome by the BP algorithm. BP requires target patterns or signals before it can be applied in the systems as it a supervised learning algorithm.

Advantages of ANN:
- More like a real nervous system
- Solving problems that are too complex
- Quickly and relatively easily model
- Provide an analytical alternative to conventional techniques which are frequently limited by strict assumptions of normality, linearity and variable independence.

3.8.4 Process of Back Propagation Algorithm

The training input pattern obtained from the sensors is given to the network input unit. The input pattern is propagated forward in the network from layer to layer until the output pattern is generated by the output layer. If the generated output pattern is different from the target output, the error will determine and propagated backward through the network from the output layer to the input layer. The weights are adjusted as the error is propagated.

The three-layer network shown in Figure 3.14 is considered to derive the back propagation learning law. Let the indices i, j and k represent the neurons in the input, hidden and output layers, respectively. Generally, i indicates the number of input signals and k denotes the number of recognition patterns. x_1, x_2, x_3 until x_n are input signals which propagated forward through the network from the input layer to the output layer, and error signals, e_1, e_2, e_3 until e_l, propagated backward from the output layer to the input layer. The symbol w_{ij} refers to the synaptic weight for the connection between neuron i in the input layer and neuron j in the hidden layer. The symbol w_{jk} represents the synaptic weight between neuron j in the hidden layer and neuron k in the output layer.
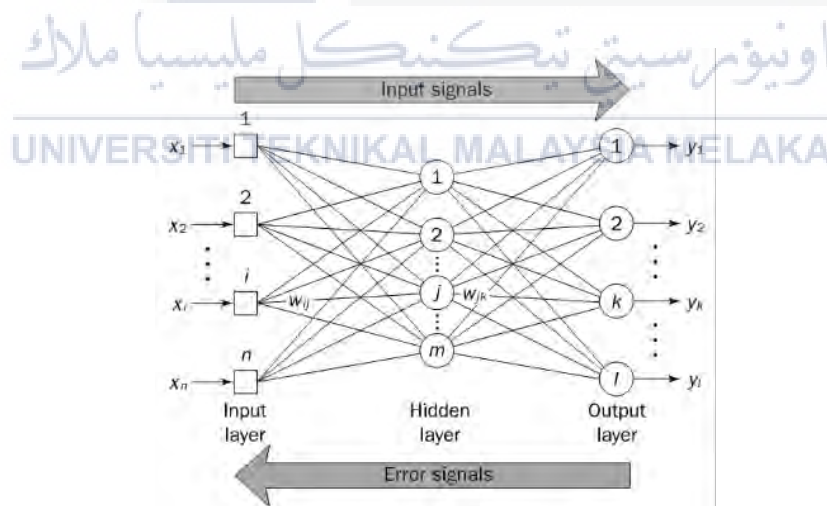

Figure 3.14: Multilayer neural network structure [32]

Step 1: Initialization

The synaptic weights and threshold θ of the network are set randomly by constantly distribution within a small range:

$$\left(-\frac{2.4}{F_i}, \ +\frac{2.4}{F_i}\right) \tag{3.2}$$

where Fi represents the sum of the inputs of neuron i in the network. The weight is initialized based on a neuron-by-neuron basis.

Step 2: Activation

The back-propagation neural network is activated by putting in the inputs $x_1(p)$, $x_2(p)$, $x_3(p)$, until $x_n(p)$ and target outputs $y_{d,1}(p)$, $y_{d,2}(p)$, $y_{d,3}(p)$, until $y_{d,n}(p)$.

    (a)   The actual outputs of the neurons in the hidden layer, y_j are calculated:

$$y_j(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) \cdot w_{ij}(p) - \theta_j\right] \tag{3.3}$$

where sigmoid denotes the sigmoid activation function, n refers to the number of inputs of neuron j in the hidden layer, p is the number of iteration and θ is the threshold.

    (b)   The actual outputs of the neurons, in the output layer, y_k are calculated:

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p) \cdot w_{jk}(p) - \theta_k\right] \tag{3.4}$$

where m refers to the number of inputs of neuron k in the output layer and θ is the threshold.

Step 3: Weight training

The weights in the back-propagation network are updated propagating backward the errors associated with output neurons.

(a) The error gradient, δ_k for the neurons in the output layer is calculated:

$$\delta_k(p) = y_k(p) \cdot [1 - y_k(p)] \cdot e_k(p) \qquad (3.5)$$

where $e_k(p) = y_{d,k}(p) - y_k(p)$ and $y_{d,k}(p)$ denotes the target output of neuron k at iteration p.

The weight corrections are calculated:

$$\Delta w_{jk}(p) = \alpha \cdot y_j(p) \cdot \delta_k(p) \qquad (3.6)$$

where y_j represents the output of neuron j in the hidden layer, the symbol $\delta_k(p)$ denotes the error gradient at neuron k in the output layer at iteration p and α is the learning rate.
The weights of the output neurons are updated:

$$\Delta w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p) \qquad (3.7)$$

(b) The error gradient for the neurons in the hidden layer is calculated:

$$\delta_j(p) = y_j(p) \cdot [1 - y_j(p)] \cdot \sum_{k=1}^{l} \delta_k(p) w_{jk}(p) \qquad (3.8)$$

where l refers to the number of neuron in the output layer.
The weight corrections are calculated:

$$\Delta w_{ij}(p) = \alpha \cdot x_i(p) \cdot \delta_j(p) \qquad (3.9)$$

The weights of the hidden neurons are calculated:

$$\Delta w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \qquad (3.10)$$

Step 4: Iteration
The iteration p is increased one by one, and then return to Step 2. The process is repeated until the selected error criterion is satisfied.

3.8.5 Neural Network Traing using the MATLAB Neural Network Tool

After the configure the network and initialize the weights and biases, the collected data were used to train the neural network using MATLAB Neural Network Tool as shown in Figure 3.15.
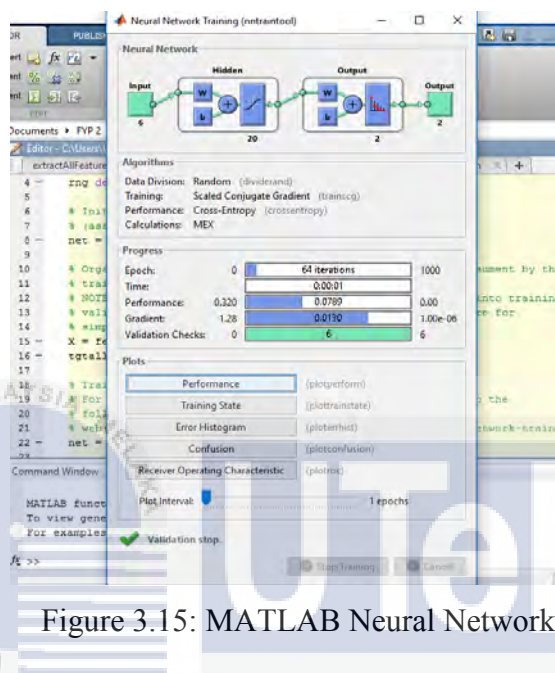

Figure 3.15: MATLAB Neural Network Tool

3.9 Development of Android Application

The system is implemented on the Android platform that allows receiving via the Wi-Fi interface. Two Android Applications were developed using the Android Studio software. The Android Application communicates with the microcontroller through TCP/IP (Transmission Control Protocol/Internet Protocol) using HTTP method. The Android Application consists of short message service (SMS) and global positioning system (GPS) location service on the smart phone. The actual location is determined using the integrated GPS module of the smart phone. Once the fall accident happens, the alert system will trigger and the application will obtain the information of body temperature and pulse rate of the user from the vital sign monitoring device. Then, alert messages which consist of the location information of and body conditions of the user will send to third parties by SMS. Figure 3.16 shows the flow chart for the designed Android Application.
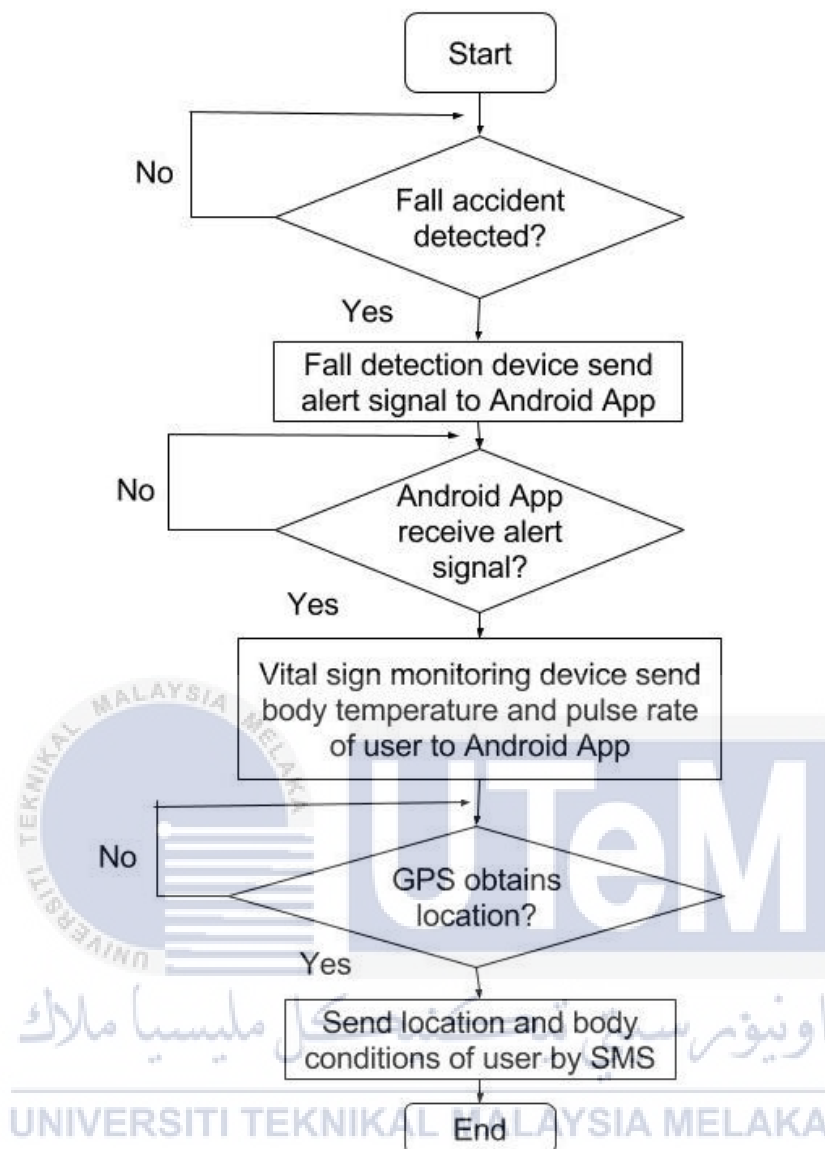
Figure 3.16: Flow Chart of Fall Alert Android Application

3.10 Performance Measurement for Vital Sign Monitoring System

The pulse rate sensor and temperature sensor have been tested on an individual to verify the sensor's performance. For performance checking of our system, we measured the heart rate and body temperature of a person for 10 minutes with wearing the vital sign monitoring device. The pulse sensor was mounted in a finger ring while the temperature sensor worn on the wrist. After the user has worn the vital sign monitoring device, then run the "Health Monitoring" android application. The "Health Monitoring" android application will display the body temperature and pulse rate of the user. At the same time, the

heartbeat measured in traditional way (i.e. calculate number of pulses in one minute) which represents the actual measured heart rate. While measuring body temperature, we measured the actual temperature through a digital thermometer.

3.11   Real Experiments

Several testing and experiments were carried out to test and validate the performance and functionality of the system. The aim of the experiments is to debug the program code, fine-tuning and improving the overall performance of the product.

A controlled test was carried out with the human fall detection and alert system. This test is similar to the previous experiment in Section 3.8 where both fall activities and ADL simulations were performed by a group of volunteers. 15 volunteers were involved in testing this system. Every subjects aged ranging from 18 to 40, height ranging from 160cm to 185cm, and weight ranging from 40kg to 90kg. The system was tested against 4 types of falls and 4 types of ADL activities. Each activity was repeated for three times by the subjects. Table 3.4 and 3.5 presents the details of the evaluation test 1 and 2.

Table 3.4: Evaluation test 1 (ADL)

Subjects	15
Ages	18-40
Heights	160-185cm
Activities	Walk, Sitting, Running, Jumping, Laying Down
No. of times action repeated	3 times
Total ADL performed	(15x 5x3)=225

Table 3.5: Evaluation test 2 (fall activities)

Subjects	15
Ages	18-40
Heights	160-185cm
Activities	Fall forward, fall backward, fall left, fall right
No. of times action repeated	3 times
Total Fall performed	(15x4x3)=180

3.11.1 Performances Evaluation Method

The results of the experiments categorized based on the four possible conditions. The numbers of occurrence for the four conditions are used for the calculations of the accuracy, sensitivity and specificity of the fall detection algorithm. The 4 possible conditions for the output of the system are presented below:

- True Positive (TP) denotes the number of falls that successfully detected.
- True Negative (TN) denotes the number of non-fall activities that not detected as fall events.
- False Positive (FP) denotes the number of non-fall activities that detected and announced as a fall by the system.
- False Negative (FN) denotes the number of falls that occurs but does not announce by the system.

The calculations of sensitivity, specificity and accuracy of fall detection are based on the values of those four possible conditions. The formula for sensitivity, specificity and accuracy are as shown below:
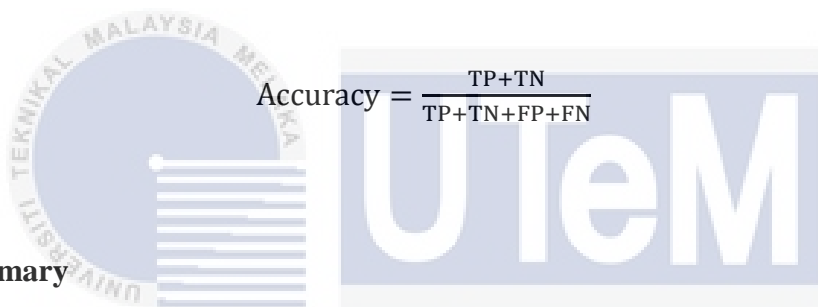
Sensitivity relates to the percentage of true falls that are accurately detected by the system, 100% denoting that all falls are detected.

$$\text{Sensitiviy} = \frac{TP}{TP+FN} \tag{3.11}$$

Specificity represents the percentage of false fall alarms among ADL samples, 100% denoting that no false alarms are detected.

$$\text{Speci city} = \frac{TN}{TN+FP} \tag{3.12}$$

Accuracy denotes the percentage of true discrimination between falls and ADL, 100% accuracy representing 100% sensitivity and specificity.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.13}$$

3.12   Summary

With the aim of collecting data for neural network based fall detection algorithm, two tests which include fall activities and ADL were designed. 15 young healthy volunteers were recruited to simulate fall and ADL in the experiments. The GY80 IMU module strapped to the waist was used to gather data relating to acceleration and angular velocity of subjects. The sampling frequency was performed at 100 Hz. After the data was collected, the signals were processed for the development of fall detection algorithm. A back propagation neural network algorithm was designed to detect falls and classify the movement patterns. Two Android Application which able to communicate with the fall detection and alert system were developed for the vital sign monitoring and emergency alert purpose. A controlled test was carried out with the completed human fall detection and alert system to test the performance and validity. This test is similar to the data collection experiments where both fall activities and ADL simulations were performed by a group of volunteers. The accuracy, sensitivity and specificity of the fall detection algorithm were calculated based on the results of the experiments. The next chapter describes the output of the research.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter highlights the outputs of our project which includes the results and discussions obtained from experiments of section 3.8.1, 3.10 and 3.11. Besides that, this chapter will also present the prototype and Android Application developed for this project.

4.2 Data Collected from Test 1 and 2

Test 1 is the activities of daily living (ADL) experiments while Test 2 is the fall activities experiments. Both tests were performed with 15 healthy volunteers. The data collected from Test 1 and 2 were tabulated followed by presented in graphs.

Test 1: Each of the subjects performed the ADL activities with the IMU module placed on each subject's waist. The ADL included in the test 1 is walking, sitting, running, jumping and lying down. The magnitude of the accelerations and angular velocity are calculated using the values of x, y and z axis for the plotting of graphs. Figure 4.1 and 4.2 show the acceleration versus time graph and angular velocity versus time graph for ADL and fall activities respectively.

Test 2: Each of the subjects simulated falls with the IMU module placed on each subject's waist. Fall simulations performed are forward, fall, backward fall, fall left and fall right. Figure 4.3 to 4.10 show the acceleration versus time graph and angular velocity versus time graph for different fall activity.

Based on the results obtained from Test 1 and 2, there is obvious accelerations and angular velocity changes when the subject performs any activities. During running and jumping activities, the acceleration and angular velocity were changing vigorously and frequently. Based on Figure 4.3 to 4.10, the acceleration and angular velocity will suddenly increase when the any of the fall activities occurs. In general, different activities will show different patterns of signals, therefore these signals can be used in classification of fall events.
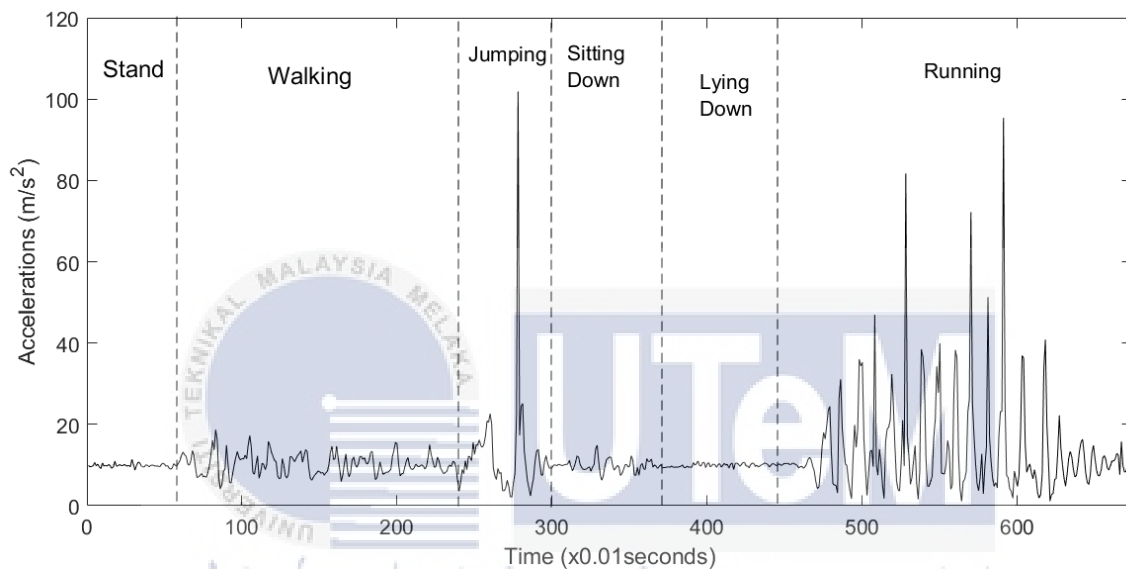

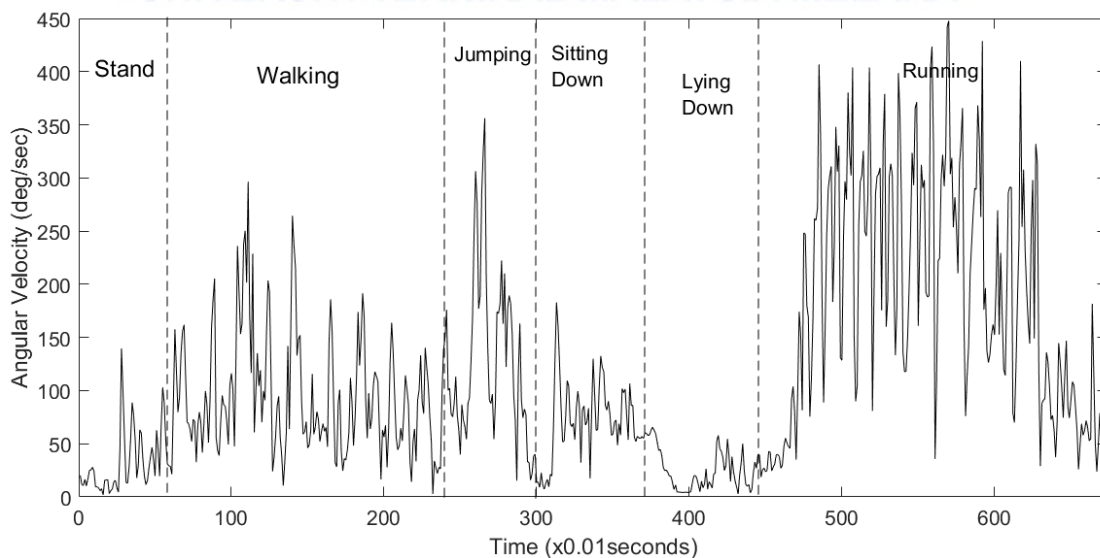Figure 4.1: Acceleration versus time graph for ADL

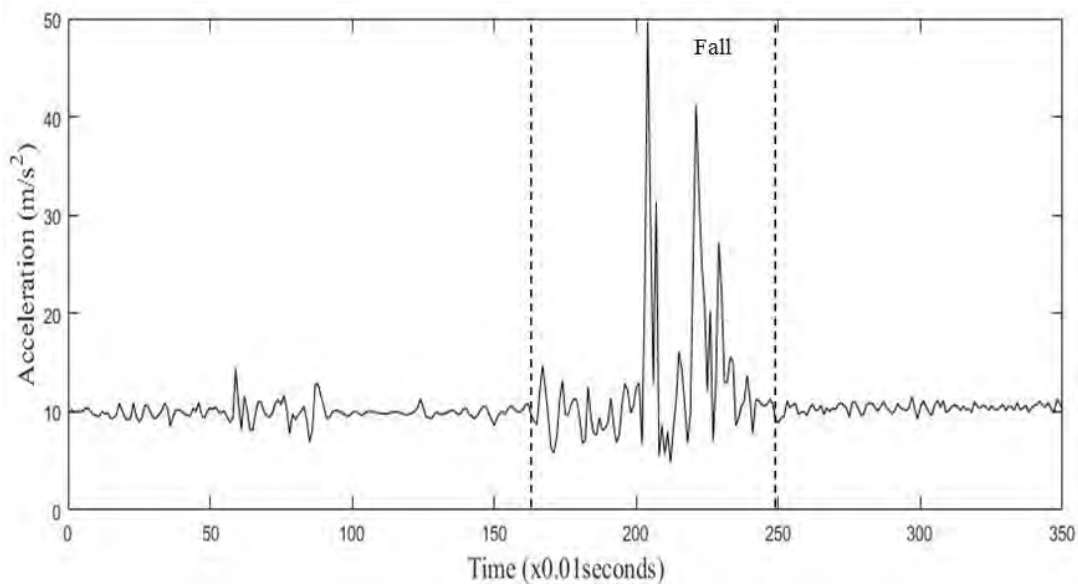Figure 4.2: Angular velocity versus time graph for ADL

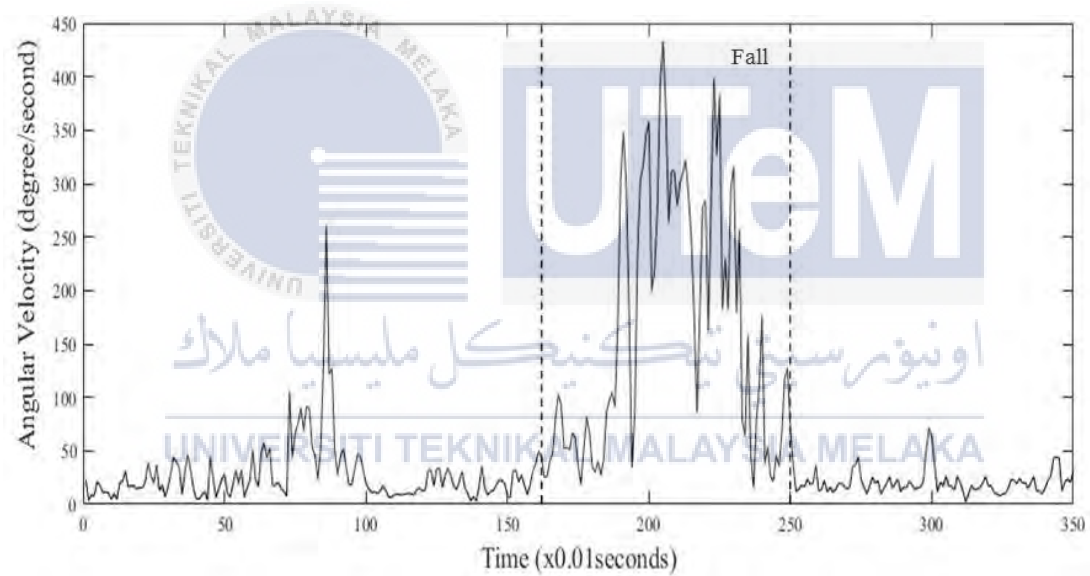Figure 4.3: Acceleration versus time graph for a fall forward motion

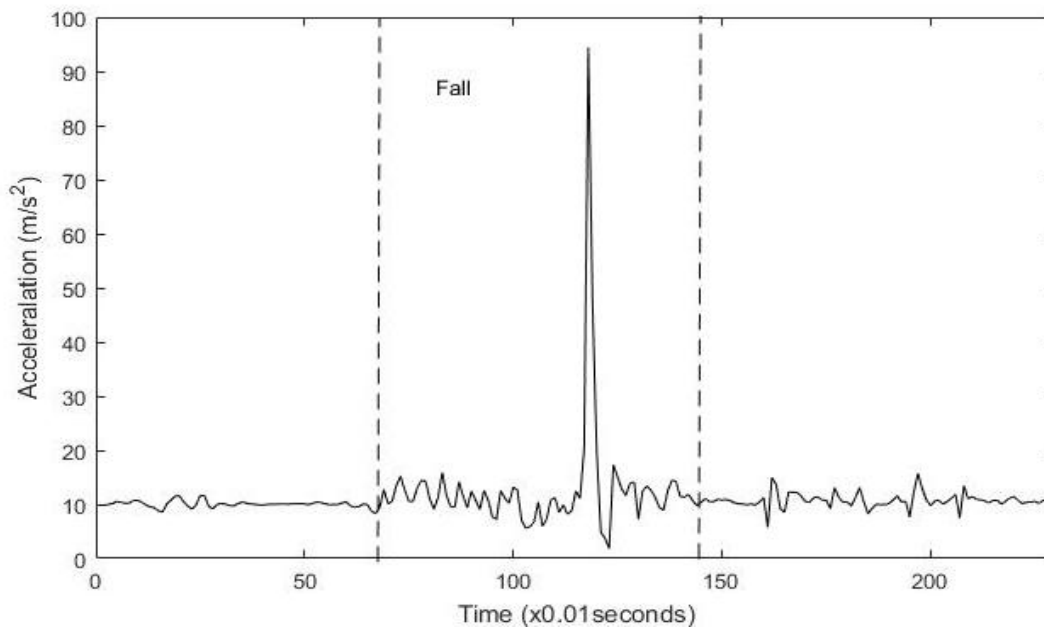Figure 4.4: Angular velocity versus time graph for a fall forward motion

Figure 4.5: Acceleration versus time graph for a fall backward motion
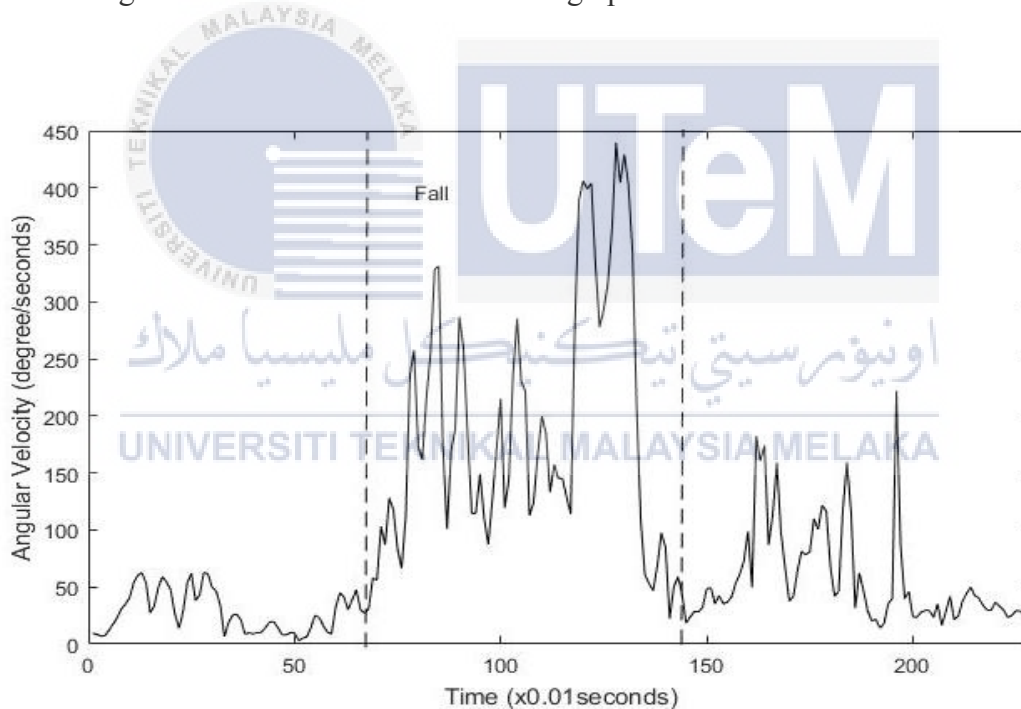

Figure 4.6: Angular velocity versus time graph for a fall backward motion
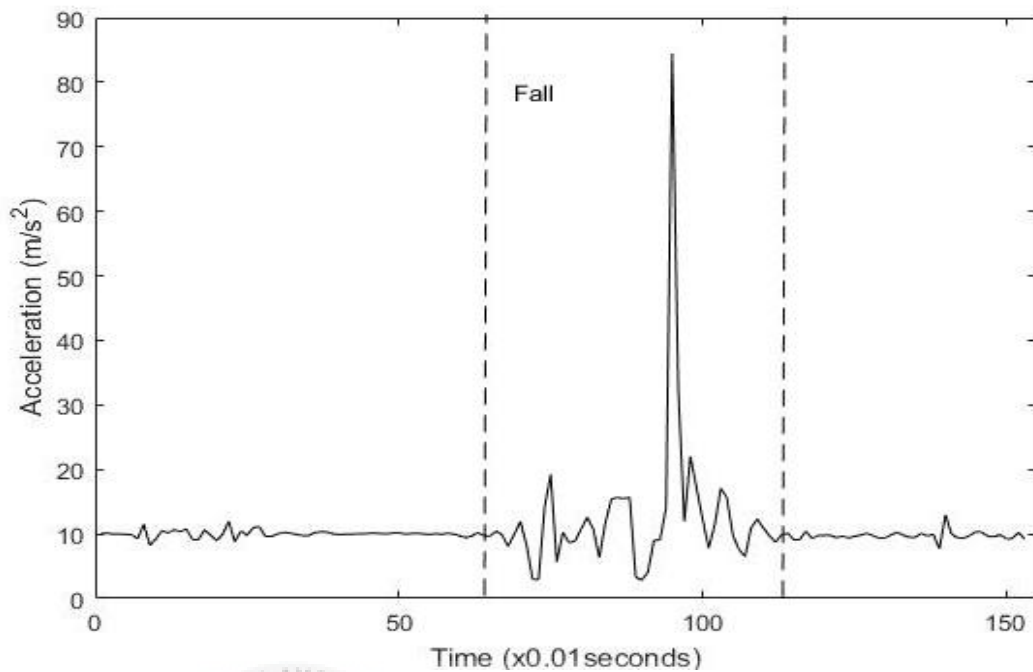
Figure 4.7: Acceleration versus time graph for a fall left motion
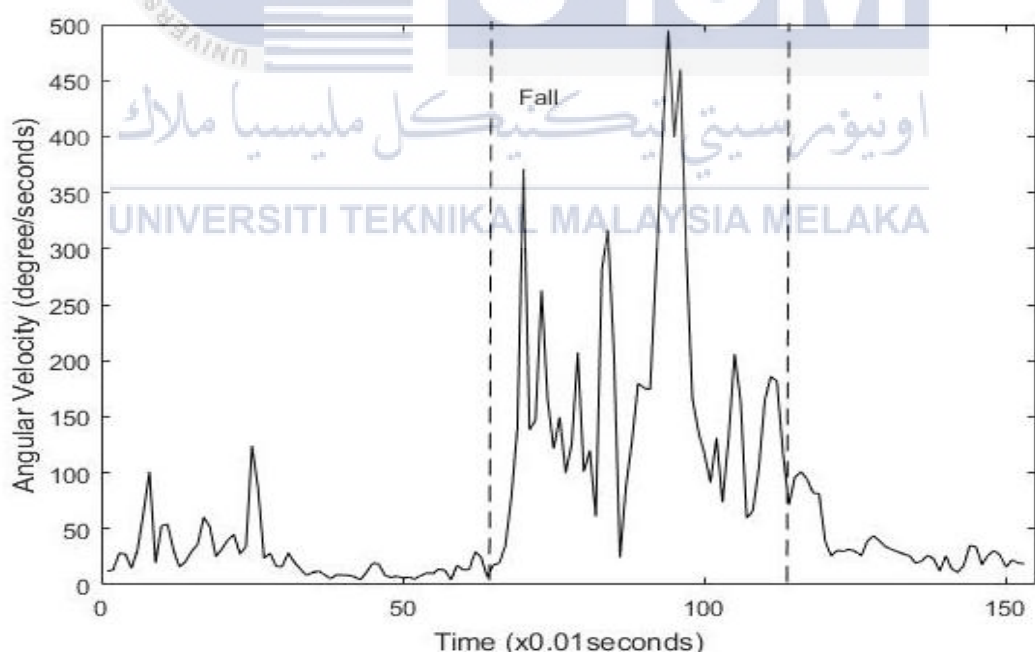

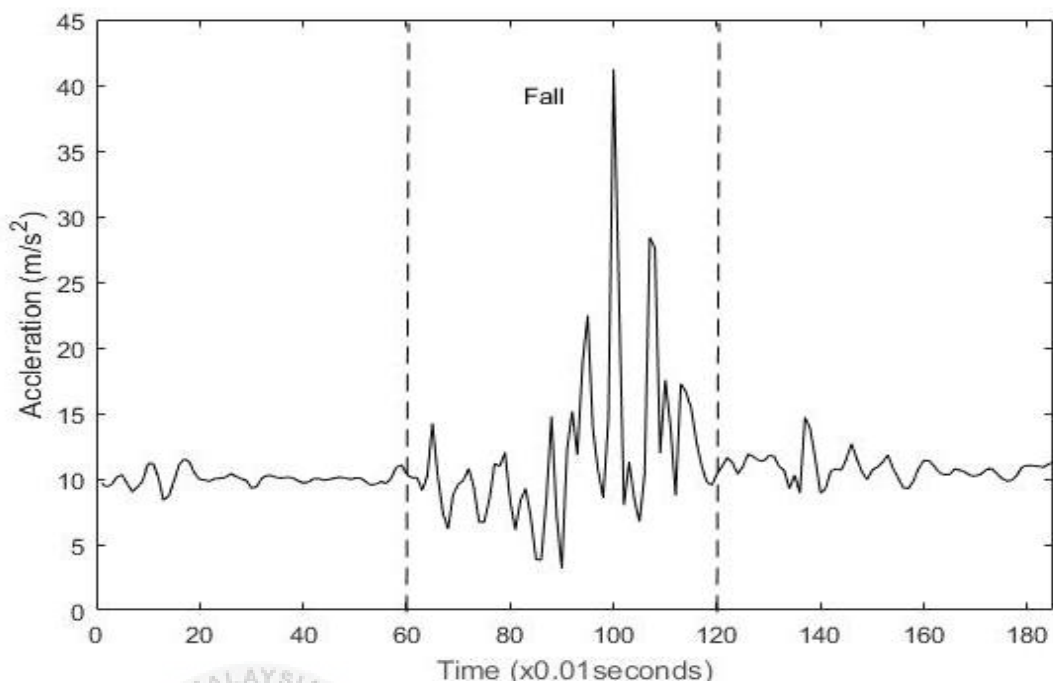Figure 4.8: Angular velocity versus time graph for a fall left motion

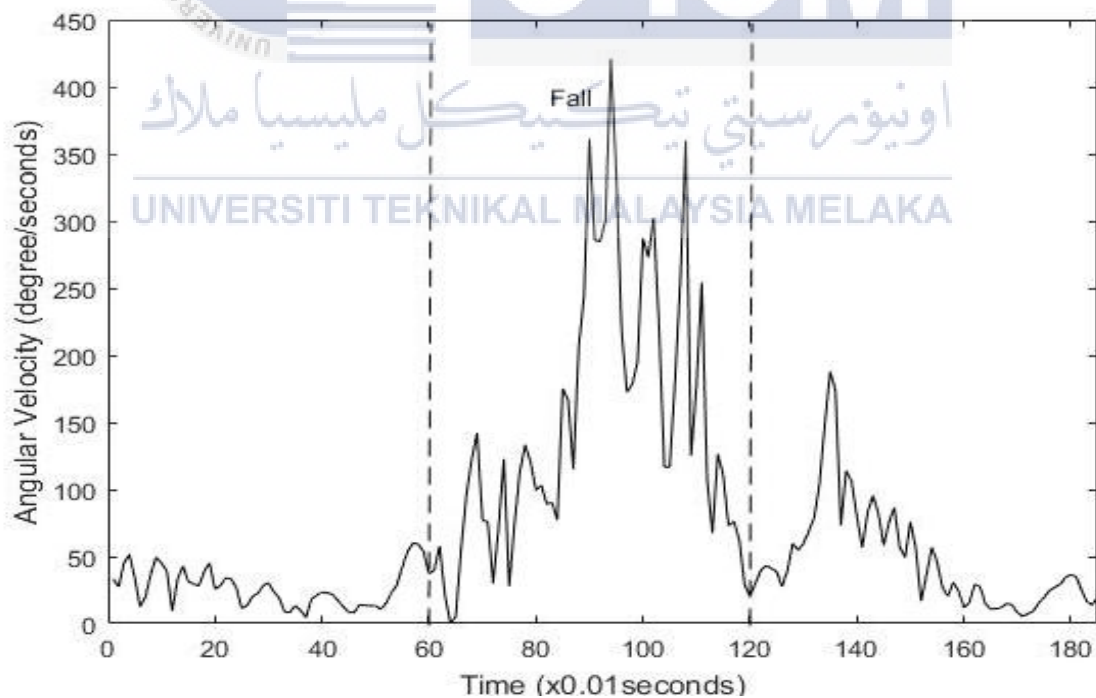Figure 4.9: Acceleration versus time graph for a fall right motion

Figure 4.10: Angular velocity versus time graph for a fall right motion

4.3   Neural Network Confusion Matrix

After the signal processing process, the collected data were used to train the neural network using the MATLAB Neural Network Tool. A confusion matrix was used to validate the trained network more systematically. To quantitatively evaluate the performance of the neural network algorithm, one would normally measure the predictions over a whole test data set, and compare them against the known class values. The ultimate prediction performance can be represented visually in a number of different ways. The confusion matrix is a square matrix that summarizes the cumulative prediction results for all couplings between actual and predicted classes, respectively. Normally it is good practice to use a test set different from the training set. This ensures that the results are not biased by the particular training dataset used. Figure 4.11 presents the confusion matrix for our trained neural network. The confusion matrix shows that the trained network has achieved a high accuracy of 95.7% in overall.
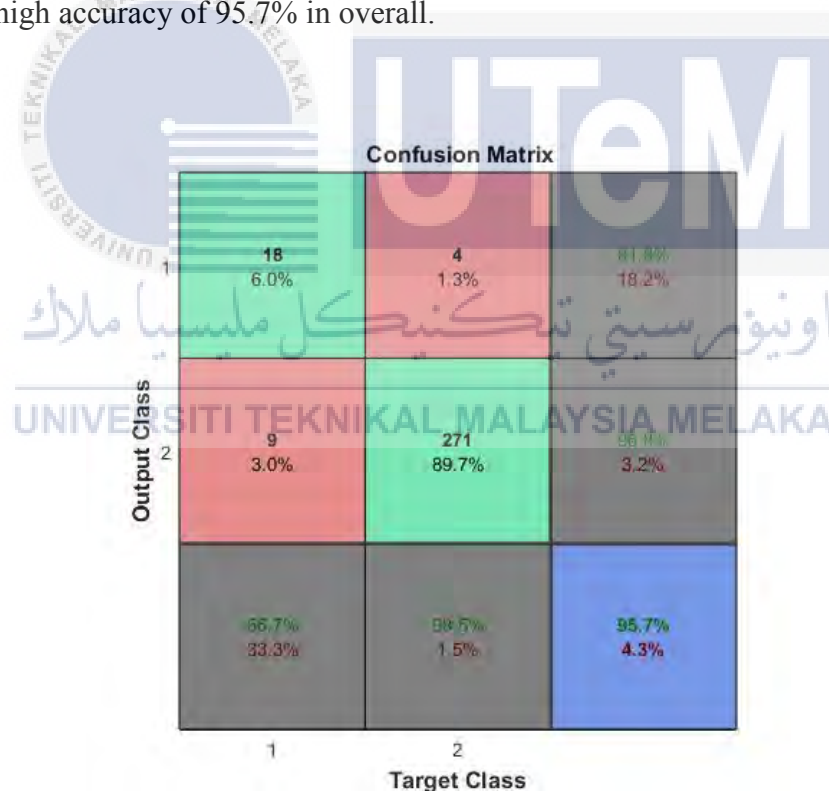
Figure 4.11: Confusion matrix for trained neural network

4.4   Prototype

A prototype casing was designed to protect the components of our system from damage. The casing was drawn using software SolidWork 2015 and printed by 3D printer. The casing was designed with a length of 67mm, width of 60mm and height of 89mm. Figure 4.12 shows the explore view for the designed casing while Figure 4.13 shows the orthographic drawing for the designed casing. Figure 4.14 presents the printed prototype casing for fall detection system.

For the vital sign monitoring device, the prototype was completed using a small container and a wrist band. A LM35 temperature sensor and a pulse rate sensor were connected to NodeMCU board and placed inside the small container. Figure 4.15 shows the prototype for the vital sign monitoring device.

Figure 4.12: Explored view for the designed casing

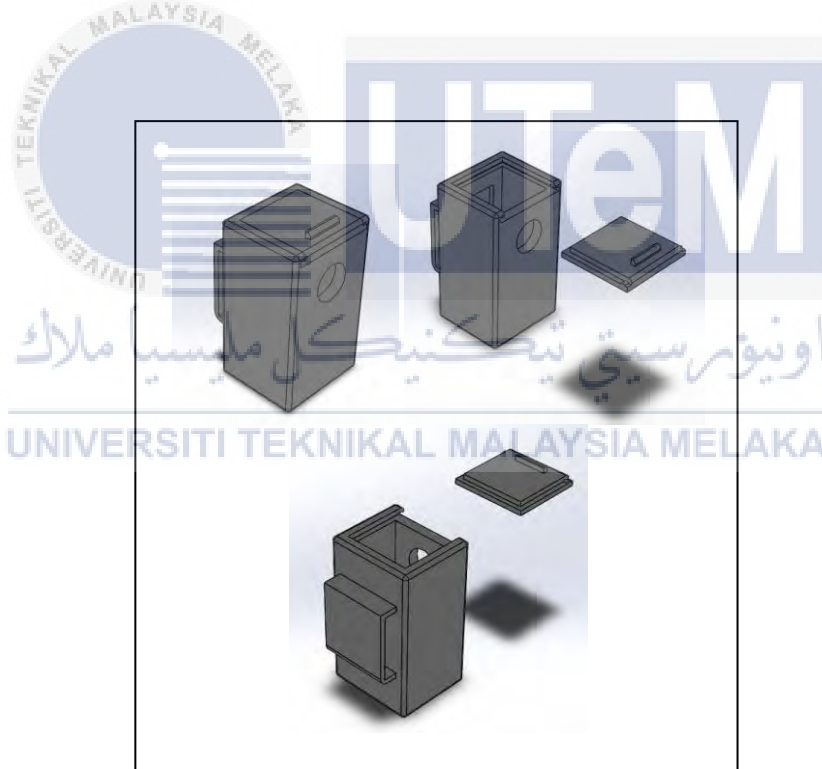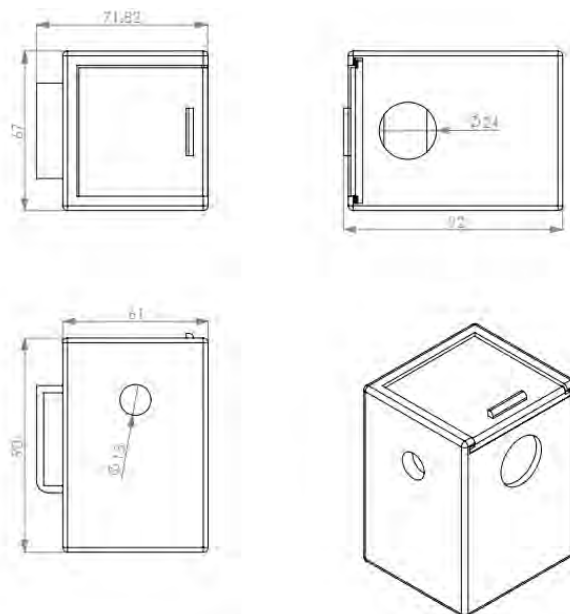Figure 4.13: Orthographic drawing for the designed casing

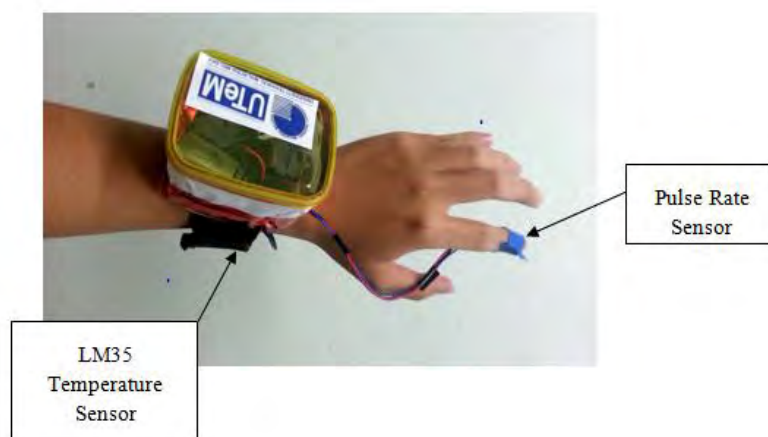Figure 4.14: Prototype casing for fall detection system

Figure 4.15: Prototype for the vital sign monitoring device

4.5 Developed Android Application

For this project, two Android Application had been developed. One for the health monitoring purpose, another one for the alert purpose. Figure 4.16 shows the Android Application for the health monitoring. This application will displays the body temperature, heart pulse rate and the fall status of the users. Besides that, the normal range for the body temperature is set from 35°C to 37.5°C, while the heart pulse rate range from 60BPM to 130BPM. If the body temperature or heart pulse rate outside the normal range, the Android Application will show "Abnormal" for the status.

For the fall alert Android Application, the alert screen will pop out when the system detects any fall occurs as shown in Figure 4.17 The users may press the "Yes" or "No" button to decide whether need to send the alert message. In addition, the system also includes a 20 second countdown timer function. If the user already loses consciousness because of the fall and the countdown timer has reach zero without any response, the application will automatically activate the alert function and send the alert message together with GPS location and body conditions of the users to the third party. Figure 4.18 presents the alert messages that sent to the third party.

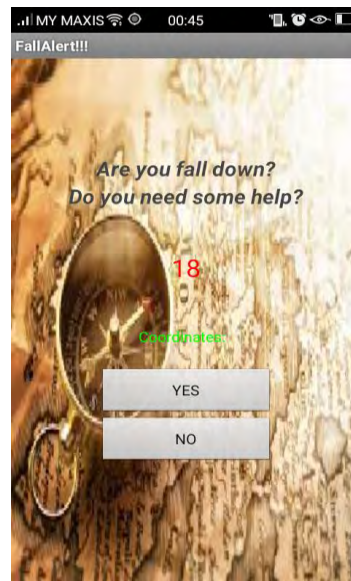Figure 4.16: Health monitoring application

Figure 4.17: Emergency alert screen

Figure 4.18: Alert message

4.6 Performance Results for Vital Sign Monitoring System

Table 4.1 presents the comparison of pulse rate using pulse sensor and manual calculation. Table 4.2 shows the comparison of the body temperature using temperature sensor and a digital thermometer. As experimental results indicate, there are some errors between the measured pulse rate and the actual pulse rate. The analog value of corresponding blood flow is affected by noise due to the tiny movement of the finger while measuring pulse rate through our system. For this reason the pulse rate measured by the pulse sensor is slightly deviated from the actual pulse rate. For the body temperature, there is also some variance in the measured value. This is due to the LM35temperature sensor not designed especially the measurement of the body temperature, therefore it might have some variance in the body temperature measured the LM35 temperature sensor.

Table 4.1: Comparison of the pulse rate using pulse sensor and manual calculation

Time (Minute)	Pulse Rate (Bpm)	
	Pulse Sensor	Manual Calculation (Actual Heartbeat)
1	78	80
2	79	79
3	78	78
4	80	80
5	83	83
6	82	81
7	78	79
8	83	83
9	84	85
10	87	85

Table 4.2: Comparison of the body temperature using temperature sensor and digital
thermometer

Time (Minute)	Body Temperature on Wrist (Degree Celcius)	
	LM35 Temperature Sensor	Digital Thermometer (Actual Body Temperature)
1	35.3	35.3
2	35.3	35.4
3	35.4	35.4
4	35.2	35.2
5	35.3	35.3
6	35.4	35.3
7	35.3	35.2
8	35.1	35.1
9	35.1	35.1
10	35.3	35.3

4.7 Validation Results for Fall Detection

The results obtained from the evaluation experiment 1 and 2 for section 3.13 is shown in Table 4.3 below. A total of 180 falls and 225 activities of daily living were performed. The sensitivity, specificity and accuracy of the system were calculated using the formula 3.11, 3.12 and 3.13. The results obtained show that our system has a high performance with sensitivity of 95.5%, specificity of 96.4% and accuracy of 96.3%. By applying neural network, we are able to accurately classify fall events and other fall-like activities, which are difficult to distinguish using threshold method. Due to the similarity of signal profile, the ADL motion like jumping and running are more likely to produce False Positive results. The accuracy of the neural network system can be improved by including more input patterns in the training phase.

Calculations:

$$\text{Sensitiviy} = \frac{173}{173+7} \text{ x } 100\%$$
$$= 95.5\%$$

$$(4.1)$$

$$\text{Speci city} = \frac{217}{217+8} \text{ x } 100\%$$
$$= 96.4\%$$

$$(4.2)$$

$$\text{Accuracy} = \frac{173+217}{173+217+7+8} \text{ x } 100\%$$
$$= 96.3\%$$

$$(4.3)$$

Table 4.3 Validation Results

| Total Falls	180	Total ADL	225
True Positive (TP)	173	True Negative (TN)	217
False Negative (FN)	7	False Positive (FP)	8
Sensitivity: 95.5%			
Specificity:96.4%			
Accuracy:96.3%			

CHAPTER 5

CONCLUSION AND FUTURE WORK

In conclusion, the objective of this project had achieved since a human fall detection and alert system was successfully developed using both accelerometer and gyroscope. A GY-80 IMU module which consists of an accelerometer and gyroscope is mounted to the Intel Edison board as the fall detection device. The device must be rigidly attached to the waist of the target in order to eliminate errors in the measurement. The sole reason for the best performance of the sensor unit on the waist position may due to the waist is the centre of gravity on the human body and truly reflects the posture of the trunk. Two experiments had been conducted to collect the data of ADL and fall activities. Based on the results obtained from both experiments, we can see the difference between ADL and falls activities. During running and jumping activities, the acceleration and angular velocity were changing vigorously and frequently. When any of the fall activities occurs, the acceleration and angular velocity will suddenly increase. In general, different activities will show different patterns of signals, therefore these signals can be used in classification of fall events. The collected data was used to train the neural network for our system.

To accurately distinguish between fall activities and ADL, the trained neural network has implemented in our system. Two evaluation tests were carried out to validate the performances of the neural network algorithm in fall detection,. The real experiments were conducted and evaluated based on the correct alarms and false alarms announced by the system. As the experimental results indicate, this fall detection method able to detect and distinguish between the falling activities and ADL with high sensitivity, high specificity and high accuracy with values of 95.5%, 96.4% and 96.3% each. This means that most of the fall will be detected and the users will rarely disturb by false alarms. In overall, our system will present a high performance for fall detection.

In addition, a vital sign monitoring device was successfully developed to improve the emergency alert function in our system. We able monitor and send the information of body temperature and pulse rate when any fall accident occurs. Moreover, we included the SMS and GPS function in the alert system. With the SMS and GPS function in this system, the parents or friends of the users of our system able to receive the alert message with location when any fall accident occurs.

In future work, more simulated activities will be added in this system to improve the performance. Moreover, we may figure out some safe way to perform fall simulation with the elderly to obtain more useful data for our system. Besides that, the vital sign monitoring device can be modified and improve in the future, especially in term of size to improve the efficiency and performances of the device. With the technology of fall detection and Internet of Things, we hope that the wearable fall detection and alert system will provide reliable fall detection which can minimize fall injuries for elderly.
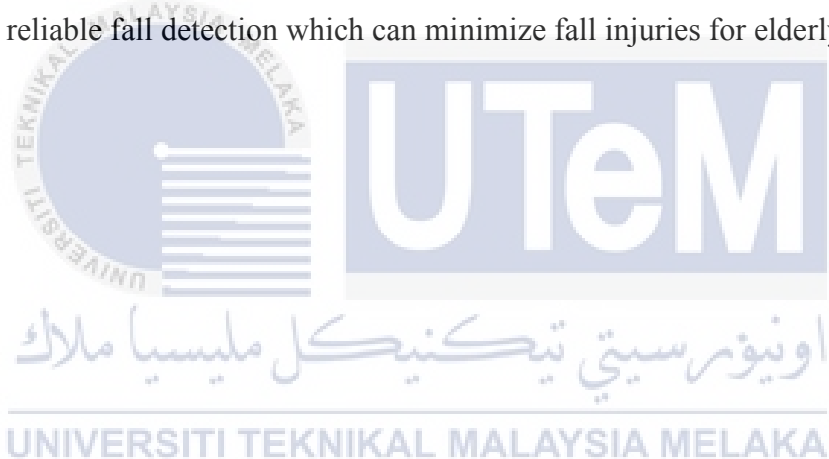
REFERENCES

[1]   W. Qu, F. Lin and W. Xu, "A Real-Time Low-Complexity Fall Detection System on the Smartphone," *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, Washington, DC, 2016, pp. 354-356.

[2]   Q. Zhang, G. Tian, N. Ding and Y. Zhang, "A fall detection study based on neural network algorithm using AHRS," *Information and Automation (ICIA), 2013 IEEE International Conference on*, Yinchuan, 2013, pp. 773-779.

[3]   H. Y. Chong and T. S. Low, "Accidents in Malaysian Construction Industry: Statistical Data and Court Cases," *International Journal of Occupational Safety and Ergonomics*, vol. 20, no. 3, pp. 503–513, 2014.

[4]   Current Population Estimates, Malaysia, 2014-2016. Department of Statistic Malaysia., 2016. [Online]. Available: http://www.statistics.gov.my [Accessed: 31-November-2016].

[5]   Dr Azidah Hashim, Overview of Malaysia's Integrated Telehealth Project. The International Medical Journal, 2003. [Online]. Available: http://www.eimjm.com/Vol2-No1/Vol2-No1-I4.htm [Accessed: 11-December-2017].

[6]   B. Aguiar, T. Rocha, J. Silva and I. Sousa, "Accelerometer-based fall detection for smartphones," *Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on*, Lisboa, 2014, pp. 1-6.

[7]   D. Chen, Y. Zhang, W. Feng and X. Li, "A wireless real-time fall detecting system based on barometer and accelerometer," *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Singapore, 2012, pp. 1816-1821.

[8]   "Definition of a fall," *Department of Health & Human Services*, Sep-2015. [Online]. Available: https://www2.health.vic.gov.au/ageing-and-aged-care/wellbeing-and-participation/ healthy-ageing/falls-prevention/definition-of-a-fall. [Accessed:31-December-2016].

[9]   Y. Li, K. C. Ho and M. Popescu, "Efficient Source Separation Algorithms for Acoustic Fall Detection Using a Microsoft Kinect," in *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 745-755, March 2014.

[10]   Y. Zigel, D. Litvak and I. Gannot*, "A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound—Proof of Concept on Human Mimicking Doll Falls," in *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858-2867, Dec. 2009.

[11]   M. Yu, A. Rhuma, S. M. Naqvi, L. Wang and J. Chambers, "A Posture Recognition-Based Fall Detection System for Monitoring an Elderly Person in a Smart Home Environment," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1274-1286, Nov. 2012.

[12]   J. L. Chua, Y. C. Chang and W. K. Lim, "Intelligent Visual Based Fall Detection Technique for Home Surveillance," *2012 International Symposium on Computer, Consumer and Control*, Taichung, 2012, pp. 183-187.

[13]   N. Nuttaitanakul and T. Leauhatong, "A novel algorithm for detection human falling from accelerometer signal using wavelet transform and neural network," *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Chiang Mai, 2015, pp. 215-220.

[14]   Y. W. Bai, C. H. Yu and H. C. Wu, "Design and implementation of fall detection and voice response detection in a smart phone," *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, Montevideo, 2014, pp. 781-785.

[15]   M. Vallejo, C. V. Isaza and J. D. López, "Artificial Neural Networks as an alternative to traditional fall detection methods," *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, 2013, pp. 1648-1651.

[16]   C. Dinh and M. Struck, "A new real-time fall detection approach using fuzzy logic and a neural network," *Proceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for Personalized Health*, 0slo, 2009, pp. 57-60.

[17]   J. Jacob *et al.*, "A fall detection study on the sensors placement location and a rule-based multi-thresholds algorithm using both accelerometer and

gyroscopes," *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, Taipei, 2011, pp. 666-671.

[18] Z. Rakhman, L. E. Nugroho, Widyawan and Kurnianingsih, "Fall detection system using accelerometer and gyroscope based on smartphone," *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering*, Semarang, 2014, pp. 99-104.

[19] L. N. V. Colón, Y. DeLaHoz and M. Labrador, "Human fall detection with smartphones," *2014 IEEE Latin-America Conference on Communications (LATINCOM)*, Cartagena de Indias, 2014, pp. 1-7.

[20] X. Yuan, S. Yu, Q. Dan, G. Wang and S. Liu, "Fall detection analysis with wearable MEMS-based sensors," *2015 16th International Conference on Electronic Packaging Technology (ICEPT)*, Changsha, 2015, pp. 1184-1187.

[21] G. Shi, J. Zhang, C. Dong, P. Han, Y. Jin and J. Wang, "Fall detection system based on inertial mems sensors: Analysis design and realization," *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, Shenyang, 2015, pp. 1834-1839.

[22] P. Pierleoni, A. Belli, L. Palma, L. Pernini and S. Valenti, "A versatile ankle-mounted fall detection device based on attitude heading systems," *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, Lausanne, 2014, pp. 153-156.

[23] Özdemir, "An Analysis on Sensor Locations of the Human Body for Wearable Fall Detection Devices: Principles and Practice," *Sensors*, vol. 16, no. 8, p. 1161, 2016.

[24] H. Mansor, S. S. Meskam, N. S. Zamery, N. Q. A. M. Rusli and R. Akmeliawati, "Portable heart rate measurement for remote health monitoring system," *2015 10th Asian Control Conference (ASCC)*, Kota Kinabalu, 2015, pp. 1-5.

[25] M. Asaduzzaman Miah, Mir Hussain Kabir, M. Siddiqur Rahman Tanveer and M. A. H. Akhand, "Continuous heart rate and body temperature monitoring system using Arduino UNO and Android device," *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, Khulna, 2015, pp. 183-188.

[26]   "Intel® Edison and Mini Breakout Kit," *DEV-13025 - SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/13025. [Accessed: 31-Mac-2017].

[27]   "GY-80 10DOF IMU," *GY-80 10DOF IMU - ePro Labs WiKi*. [Online]. Available: https://wiki.eprolabs.com/index.php?title=GY-80_10DOF_IMU. [Accessed: 31-Mac-2017].

[28]   "SparkFun Triple Axis Accelerometer Breakout - ADXL335," *Learn at SparkFun Electronics*. [Online]. Available: https://learn.sparkfun.com/tutorials/accelerometer-basics. [Accessed: 31-Mac-2017].

[29]   "3-axis digital gyroscope claims breakthrough in motion-control realism for mobile phones and gaming consoles | EE Times," *EETimes*. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1313681. [Accessed: 31-May-2017].

[30]   "How to setup NodeMCU drivers and Arduino IDE," *Marginally Clever Robots*, 24-Feb-2017. [Online]. Available: https://www.marginallyclever.com/2017/02/setup-nodemcu-drivers-arduino-ide/. [Accessed: 31-May-2017].

[31]   M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. 2nd Edition, New York: Addison Wesley, 2005.

[32]   S. Kumar, Neural networks: a classroom approach.2nd edition, New Delhi: Tata McGraw-Hill, 2013.

APPENDIX A

RESEARCH GANTT CHART (FYP 1 & FYP 2)

TITLE OF ACTIVITY	TIME LINE									
	2016				2017					
	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
1. Title selection and objective identification	▓									
2. Literature review		▓	▓							
3. Design of Methodology			▓	▓						
4. Data Acquisition and Processing										
5. Neural Network Algorithm Design										
6. Android Application Design										
7. Experiments and Improvements							▓	▓		
8. Report Preparation								▓	▓	
9. Final Presentation and Submission of Final Report										▓

APPENDIX B

Datasheets for ADXL345 3 axis Accelerometer

ANALOG DEVICES

Data Sheet

3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Accelerometer

ADXL345

FEATURES

Ultralow power: as low as 23 µA in measurement mode and
0.1 µA in standby mode at V_S = 2.5 V (typical)
Power consumption scales automatically with bandwidth
User-selectable resolution
Fixed 10-bit resolution
Full resolution, where resolution increases with g range,
up to 13-bit resolution at ±16 g (maintaining 4 mg/LSB
scale factor in all g ranges)
Embedded memory management system with FIFO
technology minimizes host processor load
Single tap/double tap detection
Activity/inactivity monitoring
Free-fall detection
Supply voltage range: 2.0 V to 3.6 V
I/O voltage range: 1.7 V to V_S
SPI (3- and 4-wire) and I²C digital interfaces
Flexible interrupt modes mappable to either interrupt pin
Measurement ranges selectable via serial command
Bandwidth selectable via serial command
Wide temperature range (−40°C to +85°C)
10,000 g shock survival
Pb free/RoHS compliant
Small and thin: 3 mm × 5 mm × 1 mm LGA package

APPLICATIONS

Handsets
Medical instrumentation
Gaming and pointing devices
Industrial instrumentation
Personal navigation devices
Hard disk drive (HDD) protection

GENERAL DESCRIPTION

The ADXL345 is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I²C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than 1.0°.

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion by comparing the acceleration on any axis with user-set thresholds. Tap sensing detects single and double taps in any direction. Free-fall sensing detects if the device is falling. These functions can be mapped individually to either of two interrupt output pins. An integrated memory management system with a 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor activity and lower overall system power consumption.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm × 5 mm × 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

Figure 1.
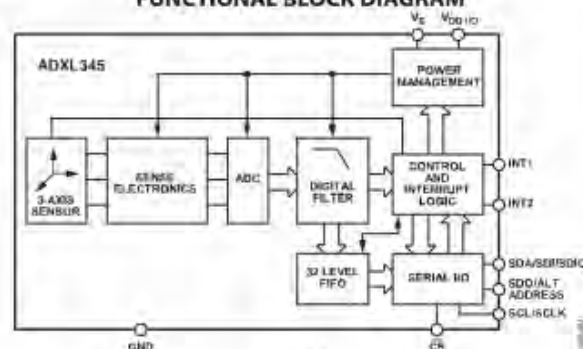
Rev. E

Document Feedback

SPECIFICATIONS

$T_A = 25°C$, $V_S = 2.5 V$, $V_{DD I/O} = 1.8 V$, acceleration = 0 g, $C_S = 10 \mu F$ tantalum, $C_{I/O} = 0.1 \mu F$, output data rate (ODR) = 800 Hz, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Test Conditions	Min	Typ[1]	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range	User selectable		±2, ±4, ±8, ±16		g
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[2]			±1		%
OUTPUT RESOLUTION	Each axis				
All g Ranges	10-bit resolution		10		Bits
±2 g Range	Full resolution		10		Bits
±4 g Range	Full resolution		11		Bits
±8 g Range	Full resolution		12		Bits
±16 g Range	Full resolution		13		Bits
SENSITIVITY	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	All g-ranges, full resolution	230	256	282	LSB/g
	±2 g, 10-bit resolution	230	256	282	LSB/g
	±4 g, 10-bit resolution	115	128	141	LSB/g
	+8 g, 10-bit resolution	57	64	71	LSB/g
	±16 g, 10-bit resolution	29	32	35	LSB/g
Sensitivity Deviation from Ideal	All g-ranges		±1.0		%
Scale Factor at X_{OUT}, Y_{OUT}, Z_{OUT}	All g-ranges, full resolution	3.5	3.9	4.3	mg/LSB
	±2 g, 10-bit resolution	3.5	3.9	4.3	mg/LSB
	±4 g, 10-bit resolution	7.1	7.8	8.7	mg/LSB
	±8 g, 10-bit resolution	14.1	15.6	17.5	mg/LSB
	±16 g, 10-bit resolution	28.6	31.2	34.5	mg/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
0 g OFFSET	Each axis				
0 g Output for X_{OUT}, Y_{OUT}		−150	0	+150	mg
0 g Output for Z_{OUT}		−250	0	+250	mg
0 g Output Deviation from Ideal, X_{OUT}, Y_{OUT}			±35		mg
0 g Output Deviation from Ideal, Z_{OUT}			±40		mg
0 g Offset vs. Temperature for X-, Y-Axes			±0.4		mg/°C
0 g Offset vs. Temperature for Z-Axis			±1.2		mg/°C
NOISE					
X-, Y-Axes	ODR = 100 Hz for ±2 g, 10-bit resolution or all g-ranges, full resolution		0.75		LSB rms
Z-Axis	ODR = 100 Hz for ±2 g, 10-bit resolution or all g-ranges, full resolution		1.1		LSB rms
OUTPUT DATA RATE AND BANDWIDTH	User selectable				
Output Data Rate (ODR)[1,3]		0.1		3200	Hz
SELF-TEST[5]					
Output Change in X-Axis		0.20		2.10	g
Output Change in Y-Axis		−2.10		−0.20	g
Output Change in Z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range ($V_{DD I/O}$)		1.7	1.8	V_S	V
Supply Current	ODR ≥ 100 Hz		140		μA
	ODR < 10 Hz		30		μA
Standby Mode Leakage Current			0.1		μA
Turn-On and Wake-Up Time[7]	ODR = 3200 Hz		1.4		ms

APPENDIX C

Datasheets for L3G4200D 3-axis Gyroscope

L3G4200D

MEMS motion sensor:
ultra-stable three-axis digital output gyroscope

Preliminary data

LGA-16 (4x4x1.1 mm)

Features

- Three selectable full scales (250/500/2000 dps)
- I²C/SPI digital output interface
- 16 bit-rate value data output
- 8-bit temperature data output
- Two digital output lines (interrupt and data ready)
- Integrated low- and high-pass filters with user-selectable bandwidth
- Ultra-stable over temperature and time
- Wide supply voltage: 2.4 V to 3.6 V
- Low voltage-compatible IOs (1.8 V)
- Embedded power-down and sleep mode
- Embedded temperature sensor
- Embedded FIFO
- High shock survivability
- Extended operating temperature range (-40 °C to +85 °C)
- ECOPACK® RoHS and "Green" compliant

Applications

- Gaming and virtual reality input devices
- Motion control with MMI (man-machine interface)
- GPS navigation systems
- Appliances and robotics

Description

The L3G4200D is a low-power three-axis angular rate sensor able to provide unprecedented stability of zero rate level and sensitivity over temperature and time. It includes a sensing element and an IC interface capable of providing the measured angular rate to the external world through a digital interface (I²C/SPI).

The sensing element is manufactured using a dedicated micro-machining process developed by STMicroelectronics to produce inertial sensors and actuators on silicon wafers.

The IC interface is manufactured using a CMOS process that allows a high level of integration to design a dedicated circuit which is trimmed to better match the sensing element characteristics.

The L3G4200D has a full scale of ±250/±500/±2000 dps and is capable of measuring rates with a user-selectable bandwidth.

The L3G4200D is available in a plastic land grid array (LGA) package and can operate within a temperature range of -40 °C to +85 °C.

Table 1.   Device summary

Order code	Temperature range (°C)	Package	Packing
L3G4200D	-40 to +85	LGA-16 (4x4x1.1 mm)	Tray
L3G4200DTR	-40 to +85	LGA-16 (4x4x1.1 mm)	Tape and reel

2 Mechanical and electrical characteristics

2.1 Mechanical characteristics

Table 4. Mechanical characteristics @ Vdd = 3.0 V, T = 25 °C, unless otherwise noted[1]

Symbol	Parameter	Test condition	Min.	Typ.[2]	Max.	Unit
FS	Measurement range	User-selectable		±250		dps
				±500		
				±2000		
So	Sensitivity	FS = 250 dps		8.75		mdps/digit
		FS = 500 dps		17.50		
		FS = 2000 dps		70		
SoDr	Sensitivity change vs. temperature	From -40 °C to +85 °C		±2		%
DVoff	Digital zero-rate level	FS = 250 dps		±10		dps
		FS = 500 dps		±15		
		FS = 2000 dps		±75		
OffDr	Zero-rate level change vs. temperature[3]	FS = 250 dps		±0.03		dps/°C
		FS = 2000 dps		±0.04		dps/°C
NL	Non linearity[4]	Best fit straight line		0.2		% FS
DST	Self-test output change	FS = 250 dps		130		dps
		FS = 500 dps		200		
		FS = 2000 dps		530		
Rn	Rate noise density	BW = 50 Hz		0.03		dps/sqrt(Hz)
ODR	Digital output data rate			100/200/400/800		Hz
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3.0 V. The operational power supply range is specified in *Table 5*.

2. Typical specifications are not guaranteed.

3. Min/max values have been estimated based on the measurements of the current gyros in production.

4. Guaranteed by design.

APPENDIX D

Datasheets for LM35 Temperature Sensor

November 2000

National Semiconductor

LM35
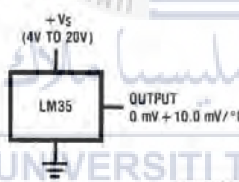# Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ˚ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼˚C at room temperature and ±¾˚C over a full −55 to +150˚C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1˚C in still air. The LM35 is rated to operate over a −55˚ to +150˚C temperature range, while the LM35C is rated for a −40˚ to +110˚C range (−10˚ with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.
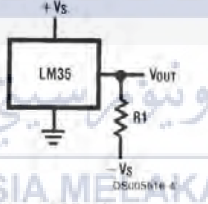
Features

- Calibrated directly in ˚ Celsius (Centigrade)
- Linear + 10.0 mV/˚C scale factor
- 0.5˚C accuracy guaranteeable (at +25˚C)
- Rated for full −55˚ to +150˚C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08˚C in still air
- Nonlinearity only ±¼˚C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

FIGURE 1. Basic Centigrade Temperature Sensor
(+2˚C to +150˚C)

Choose $R_1 = -V_S/50$ μA
$V_{OUT} = +1,500$ mV at +150˚C
= +250 mV at +25˚C
= −550 mV at −55˚C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

TO-46
Metal Can Package*
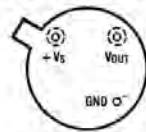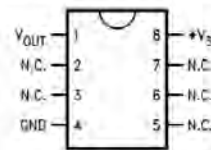

BOTTOM VIEW
DS005516-1

*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

SO-8
Small Outline Molded Package

N.C. = No Connection

DS005516-21

Top View
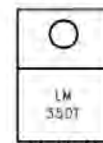**Order Number LM35DM**
See NS Package Number M08A

TO-92
Plastic Package

BOTTOM VIEW
DS005516-2

Order Number LM35CZ,
LM35CAZ or LM35DZ
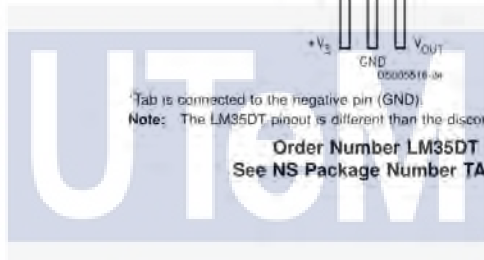**See NS Package Number Z03A**

TO-220
Plastic Package*

DS005516-24

*Tab is connected to the negative pin (GND).
Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

APPENDIX E

Arduino Coding for Data Collection

```
#include <SD.h>

#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_ADXL345_U.h>

#include <L3G.h>

/* Assign a unique ID to this sensor at the same time */

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

L3G gyro;

const int chipSelect = 15;

void displaySensorDetails(void)

{   sensor_t sensor;

    accel.getSensor(&sensor);

    Serial.println("------------------------------------");

    Serial.print   ("Sensor:        "); Serial.println(sensor.name);

    Serial.print   ("Driver Ver:    "); Serial.println(sensor.version);

    Serial.print   ("Unique ID:     "); Serial.println(sensor.sensor_id);

    Serial.print   ("Max Value:     "); Serial.print(sensor.max_value); Serial.println(" m/s^2");

    Serial.print   ("Min Value:     "); Serial.print(sensor.min_value); Serial.println(" m/s^2");

    Serial.print   ("Resolution:    "); Serial.print(sensor.resolution); Serial.println(" m/s^2");

    Serial.println("------------------------------------");

    Serial.println("");

    delay(500);
```

```
}

void displayDataRate(void)

{

  Serial.print    ("Data Rate:        ");

  switch(accel.getDataRate())

  {

    case ADXL345_DATARATE_3200_HZ:

      Serial.print    ("3200 ");

      break;

    case ADXL345_DATARATE_1600_HZ:

      Serial.print    ("1600 ");

      break;

    case ADXL345_DATARATE_800_HZ:

      Serial.print    ("800 ");

      break;

    case ADXL345_DATARATE_400_HZ:

      Serial.print    ("400 ");

      break;

    case ADXL345_DATARATE_200_HZ:

      Serial.print    ("200 ");

      break;

    case ADXL345_DATARATE_100_HZ:

      Serial.print    ("100 ");

      break;
```
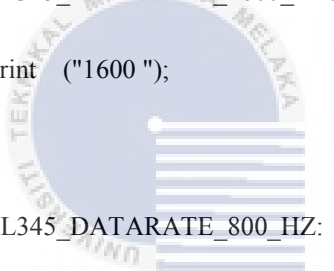
```
case ADXL345_DATARATE_50_HZ:

  Serial.print    ("50 ");

  break;

case ADXL345_DATARATE_25_HZ:

  Serial.print    ("25 ");

  break;

case ADXL345_DATARATE_12_5_HZ:

  Serial.print    ("12.5 ");

  break;

case ADXL345_DATARATE_6_25HZ:

  Serial.print    ("6.25 ");

  break;

case ADXL345_DATARATE_3_13_HZ:

  Serial.print    ("3.13 ");

  break;

case ADXL345_DATARATE_1_56_HZ:

  Serial.print    ("1.56 ");

  break;

case ADXL345_DATARATE_0_78_HZ:

  Serial.print    ("0.78 ");

  break;

case ADXL345_DATARATE_0_39_HZ:

  Serial.print    ("0.39 ");

  break;
```

```
      case ADXL345_DATARATE_0_20_HZ:

        Serial.print    ("0.20 ");

        break;

      case ADXL345_DATARATE_0_10_HZ:

        Serial.print    ("0.10 ");

        break;

      default:

        Serial.print    ("???? ");

        break;

  }
  Serial.println(" Hz");

}

void displayRange(void)

{

  Serial.print    ("Range:          +/- ");

  switch(accel.getRange())

  {

    case ADXL345_RANGE_16_G:

      Serial.print    ("16 ");

      break;

    case ADXL345_RANGE_8_G:

      Serial.print    ("8 ");

      break;

    case ADXL345_RANGE_4_G:
```
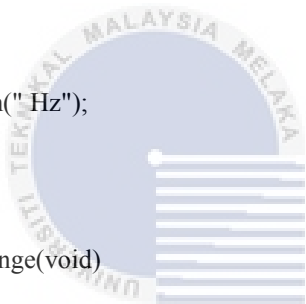
```
      Serial.print    ("4 ");

      break;

    case ADXL345_RANGE_2_G:

      Serial.print    ("2 ");

      break;

    default:

      Serial.print    ("?? ");

      break;

  }

  Serial.println(" g");

}

void setup()

{

  // Open serial communications and wait for port to open:

  Serial.begin(9600);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for Leonardo only

  }

  Serial.print("Initializing SD card...");

  // see if the card is present and can be initialized:

  if (!SD.begin(chipSelect)) {

    Serial.println("Card failed, or not present");

    // don't do anything more:

    return;
```
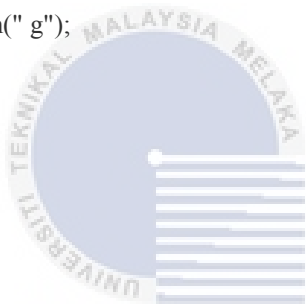
```
}

Serial.println("card initialized.");

  /* Initialise the sensor */

if(!accel.begin())

{

  /* There was a problem detecting the ADXL345 ... check your connections */

  Serial.println("Ooops, no ADXL345 detected ... Check your wiring!");

  while(1);

}

/* Set the range to whatever is appropriate for your project */

accel.setRange(ADXL345_RANGE_16_G);

// displaySetRange(ADXL345_RANGE_8_G);

// displaySetRange(ADXL345_RANGE_4_G);

// displaySetRange(ADXL345_RANGE_2_G);

/* Display some basic information on this sensor */

displaySensorDetails();


/* Display additional settings (outside the scope of sensor_t) */

displayDataRate();

displayRange();

Serial.println("");

  Wire.begin();

if (!gyro.init())

{
```

```
    Serial.println("Failed to autodetect gyro type!");

    while (1);

  }

  gyro.enableDefault();

}

void loop()

{sensors_event_t event;

  accel.getEvent(&event);

   gyro.read();

  // make a string for assembling the data to log:

float x =gyro.g.x* 0.00875;

float y =gyro.g.y* 0.00875;

float z =gyro.g.z* 0.00875;

File datalog = SD.open("datalog.txt", FILE_WRITE);

  // if the file is available, write to it:

  if (datalog) {

  datalog.print(event.acceleration.x); datalog.print("            ");

  datalog.print(event.acceleration.y); datalog.print("            ");

  datalog.print(event.acceleration.z);datalog.print("            ");

  datalog.print(x); datalog.print("            ");

  datalog.print(y); datalog.print("            ");

  datalog.println(z);

    datalog.close();}

    // print to the serial port too:
```

```
    /* Display the results (acceleration is measured in m/s^2) */

Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print("    ");

Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print("    ");

Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print("    ");Serial.println("m/s^2 ");

  Serial.print("G ");

Serial.print("X: ");

Serial.print(x);

Serial.print(" Y: ");

Serial.print(y);

Serial.print(" Z: ");

Serial.println(z);

}
```