EMBEDDED SYSTEM FOR MONITORING HUMAN ACTIVITIES USING 3-AXIS ACCELEROMETER

by

HIMABINDU YAMANA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

DECEMBER 2007

To my Mother, my Father, my Teachers and God

ACKNOWLEDGEMENTS

At the outset,I would like to thank my supervising professor Dr.Jung-Chih Chiao for constantly motivating and encouraging me, and also for his invaluable advice during the course of my master's studies. I wish to thank Dr.Enjun Xiao and Dr.Nikolai Stelmakh for their interest in my research and for taking time to serve in my defense committee.

I am grateful to all the teachers who taught me during the years I spent in school in India and in the Unites States. Finally, I would like to express my deep gratitude to my father and mother for their sacrifice, encouragement and patience. I would like to express my sincere gratitude to my friend Hari Hara Subramanian for his sagacious support all through this process. I am extremely thankful to all lab-mates, friends and roommates for their moral support.

I would like to acknowledge the unflagging financial and moral support given by Kavuri Srinivas and Moturi Madhuri. I am also thankful to Electrical Engineering department for giving me an opportunity to serve as Teaching Assistant which supported me during the course of my thesis.

November 26, 2007

ABSTRACT

EMBEDDED SYSTEM FOR MONITORING HUMAN ACTIVITIES USING 3-AXIS ACCELEROMETER

Publication No. _____

HIMABINDU YAMANA, M.S.

The University of Texas at Arlington, 2007

Supervising Professor: Dr. Jung-Chih Chiao

The last decade has witnessed a surge of interest in new sensing and monitoring devices for health care. In several medical and non-medical applications it is necessary to measure the human activities. Detection and monitoring of human activities requires the use of external sensors around the body. The reported design provides a systematic approach for characterization of states. The design employs a 3-axis accelerometer, a microcontroller and RS232 based communication protocol for sensing the activities of a person and the output will be seen on a PC with a graphical user interface.

Experiments were conducted on human test subjects and various samples are collected to estimate the accuracy of the system. The major achievements of the system are improved accuracy and precision than the previous proposed systems. The proposed sys-

tem minimizes the caregiver's effort and serves a good monitoring system for bedridden elderly person.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Human activity monitoring is systematic monitoring of human movement. Human activity is the key in determining wellbeing of a human. With the growth of electronics and information systems it is possible to collect relevant physiological information and share them between local and global information systems for the purpose of health and as a monitoring system. In recent years, number of promising prototypes of both wearable and implantable monitoring devices has emerged. These devices find their application in the areas listed below.

Patients with specific physiological conditions or convalescent's conditions can be monitored with wearable embedded systems in an integrated e-health environment, to provide better and effective health care. They also find application for collecting key physiological information of sportsmen during both effort and recuperation phases, to provide suitable feedback about body functions in a range recognized to provide optimal training and health effects. Embedded systems are used for enhancing the security and safety of the individuals, as they find applications in biometric authentication and monitoring for vigilance. They can also be used as embedded alert systems which send messages to remote locations. For example elderly and disabled citizens can make use of these systems when they are in potentially dangerous situation. There are also applications in psychiatry and psychology for detection psychomotor drug effects and hyper-hypo-activity in children and adults. With growth in the fields of micro/nano-electronics, MEMS/NEMS, wireless sensors and actuators it is possible to increase the performance

and sophistication of Wearable and implantable systems. This thesis represents a follow on to unique approach in this field conceived by Kondraske that focuses on the concepts of "activity states" (eg: walking, standing, etc) and revisits a second-generation Investigation of Human Activity State Detection and Monitoring System [1] which extended the first-generation system [2]. This system differs from its predecessors, by deviating from the concept of recording the amount of activity but to concentrate on the activity states. The interest now is on developing a prototype which can be used for real time activity state detection. In addition, this thesis extends the activity states which are more suited to the real-time world, provides a JAVA based user-interface for using the system, a new algorithm for detecting and discriminating activity states and also improves the positioning and accuracy of the system by using latest hardware.

1.2 Literature review

Human activity monitoring can be classified into external monitoring systems and wearable monitoring systems. These techniques can also be grouped as direct way of recording human activity. Since these techniques involve actual observation using human observers or video cameras or using devices which are worn by subjects. Among indirect methods, physiological parameters are one of the common sources of activity information, but they are inadequate in providing information for discriminating between individual actions.

1.2.1 External monitoring devices

External activity monitoring devices that are not worn or attached to the subject. Rather they "observed" from a distance. They do not move along the subject and hence the subject is confined to a defined space in case he/she has to be monitored.

1.2.1.1 Video Human Activity Monitoring

There are several systems, which are able to detect human activity using videos.

The system [3] described in this paper is made up of three cameras. Two of these cameras are active and are part of a binocular system. They operate either as a set of three static cameras or as a set of one fixed camera and an active binocular vision system. The human activity is monitored by extracting several parameters that are useful for their classification. The system enables the creation of a record based on the type of activity. These logs can be selectively accessed and provide images of the humans in specific areas.

This system [4] automatically recognizes complex human activities embedded in video sequences acquired with a large scale view in order to monitoring wide area (car parking, archaeological site, etc) with a single static camera. The recognition process is performed in two steps: at first the human body posture is estimated frame by frame and then the temporal sequences of the detected postures are statistically modeled. Body postures are estimated starting from the binary shapes associated to humans, selecting as features the horizontal and vertical histograms and supplying them as input to an unsupervised clustering algorithm. The Manhattan distance is used for both clusters building and run-time classification. Statistical modeling of the detected postures is performed by discrete hidden Markov models.

This system [5] uses a hierarchical method for human detection and activity recognition in MPEG sequences. The algorithm consists of three stages at different resolution levels. The first step is based on the principal component analysis of MPEG motion vectors of macroblocks grouped according to velocity, distance and human body proportions. This step reduces the complexity and amount of processing data. The DC DCT components of luminance and chrominance are the input for the second step, to be

matched to activity templates and a human skin template. A more detailed analysis of the uncompressed regions extracted in previous steps is done at the last step via model-based segmentation and graph matching. This hierarchical scheme enables working at different levels, from low complexity to low false rates.

1.2.2 Wearable monitoring devices

Wearable monitoring devices are devices which are attached to the subject. They could be strapped or hooked on to the subject under test.

1.2.2.1 Electromyography:

Electromyography (EMG) [6] is a technique for evaluating and recording physiologic properties of muscles at rest and while contracting. EMG is performed using an instrument called an electromyography, to produce a record called an electromyogram. An electromyography detects the electrical potential generated by muscle cells when these cells contract, and also when the cells are at rest. To perform intramuscular EMG, a needle electrode is inserted through the skin into the muscle tissue. A trained medical professional (most often a physiatrist, neurologist, or physical therapist) observes the electrical activity while inserting the electrode. The insertional activity provides valuable information about the state of the muscle and its innervating nerve. Normal muscles at rest make certain, normal electrical sounds when the needle is inserted into them. Then the electrical activity when the muscle is at rest is studied. Abnormal spontaneous activity might indicate some nerve and/or muscle damage. Then the patient is asked to contract the muscle smoothly. The shape, size and frequency of the resulting motor unit potentials are judged. Then the electrode is retracted a few millimeters, and again the activity is analyzed until at least 10-20 units have been collected. Each electrode track gives only a very local picture of the activity of the whole muscle. Because skeletal mus-

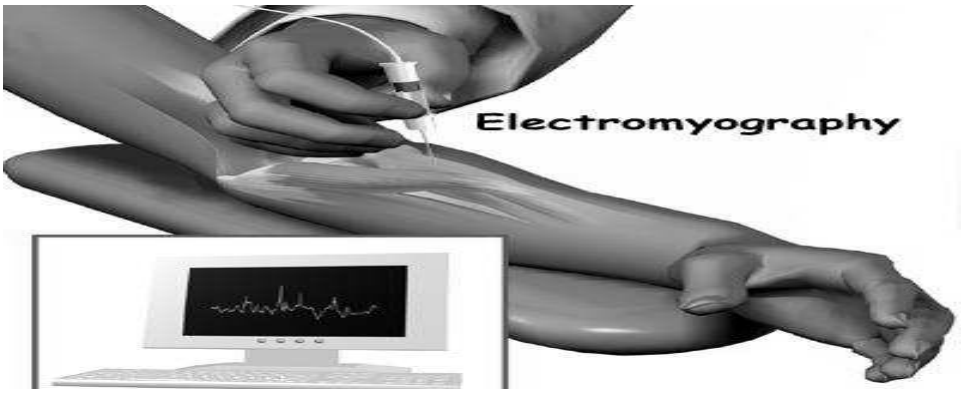

Figure 1.1 Electromyography

cles differ in the inner structure, the electrode has to be placed at various locations to obtain an accurate study. Intramuscular EMG may be considered too invasive or too specific in some cases. A surface electrode may be used to monitor the general picture of muscle activation, as opposed to the activity of only a few fibers as observed using a needle.

1.2.2.2 Polysomnography:

Polysomnography [7] is a comprehensive recording of the bio-physiological changes that occur during sleep. The polysomnogram, or PSG, is usually performed at night, when most people sleep. PSG monitors many body functions including brain (EEG), eye movements (EOG), muscle activity or skeletal muscle activation (EMG), heart rhythm (ECG), and breathing function or respiratory effort during sleep. For a polysomnogram, the EEG will generally consist of four "exploring" electrodes and two "reference" electrodes. The exploring electrodes are usually attached to the scalp near the central and occipital portions of the brain via a paste that will conduct electrical signals originating from the neurons of the cortex.

Figure 1.2 Polysomnography

1.2.2.3 Activity Monitoring System

The Activity Monitor System [8] provides a way of monitoring and summarizing an individual's physical activity. Like a pedometer, the AWare Activity Monitor System can count steps, but it can also accurately summarize the type and level of physical activity of the monitored person on a minute-by-minute basis. This information is sent wirelessly to a receiving console where it can be recorded, displayed on a local computer display.

The AWare Activity Monitor Wearable is a small (1.4 x 1.8 x 0.6 , or 35.6mm x 45.7mm x 15.2mm) light-weight (0.85oz, or 23g) body-worn device that continuously monitors wearers' physical activity, 24 hours a day, for up to six months on a single battery. The Activity Monitor Wearable continuously measures, interprets, and summarizes the wearer's activities, wirelessly communicating with the Activity Monitor Console or other receiving device up to 300ft (100m) away. The AWare Activity Monitor Wearable uses a MEMS +- 3G, three-axis accelerometer sampled at 50Hz with 8 bits of resolution per axis. The Activity Monitor Wearable is capable of streaming this data continuously (telemetry mode), summarizing it the form of an activity journal (steps taken, hours

slept, activity type and level), or a combination of the two. The Wireless Activity Monitor includes a three-color LED and an ergonomic molded button for user interaction.

1.2.2.4 Human Activity Detection and Monitoring System

Activity can mean many things. It can reflect higher-level concepts, in which a global index reflecting the amount of activity is desired, such as mobility in bed. On the other hand extreme details regarding the amount and the timing of the activities or activity patterns may be of interest as in monitoring activity of skeletal muscles. Many early attempts were made to measure activity level over time which was termed as human activity monitoring. kondraske observed that while humans engage in activities which are often quite complex dynamically, there are distinct patterns that lead as to identify specific states such as standing, walking, etc. Information regarding the frequency, duration and transitioning of such activity states over a period of time can be highly beneficial to the effective diagnosis, treatment and facilitation of individuals with various pathologies (eg., neurological, psychological, orthopedic, etc.,). Additional benefits can be obtained in behavioral research, pharmaceutical clinical trails, and many other investigative endeavors. Based on this, a prototype system dubbed the HASDMS-I was developed and evaluated [2]. Recent literature review indicates that this approach remains unique; it is the only effort found that ventures into activity state detection.

In the first generation system, the parameters used to discriminate activity states are the angle between long axis of the thigh and gravity and its frequency of movement in the sag ital plane. The sensing of the axis and the plane is done by electronic accelerometers. In situations where there is slight, intermittent or thigh movement, the angle of orientation is utilized to determine whether the activity is lying/sitting or standing, the frequency of the gait cycle along with the orientation of the thigh is utilized to determine whether the activity is walking or running.

This system relies solely on the orientation and motion of the thigh to determine the specific activity state. The sensing units is worn on the thigh. Therefore, with the thigh in a (relatively) horizontal orientation, so movement (or slight intermittent motion) is interpreted as a state of lying or sitting. larger movements of the thigh when in a relatively horizontal orientation result in an unknown state that may be due to activities such as stair climbing or bicycle riding. When the thigh is oriented closer to vertical, the possible activity states include standing, walking and running. These states are distinguished by zero , low or high changes in thigh movement respectively. The information is passed on to a host computer which displays the activity states in various formats.

1.3 Contributions

Given the amount of work done in previous thesis, this thesis concentrates on following aspects.

1. Improve the micro-controller to get better sensitivity
2. Improve the resolution of the analog to digital converter
3. Improve precision of the state detection
4. Conduct experiments with human subjects to define criteria for activity state detection
5. Evaluate results obtained and provide recommendation for next phase of research

The emphasis is not on a new devices, but rather on a tool and approach to facilitate research aimed at improving state detection performance and applicability to wider activity states.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 describes the architecture of the state detection system and its various components. Chapter 3 gives an in depth description of the software implementation of the system. Chapter 4 discusses, how various experiments conducted on human subjects and their results. Finally Chapter 5 outlines conclusions and future work.

CHAPTER 2

HARDWARE DESIGN

Shorter hospital stay and better community care are expected to be the future trend of national health services. In order to evaluate the effectiveness of remote rehabilitation program, physical therapists must assess the motion characteristics during standing up, walking, sitting up and so on. Given this background, we have developed a wearable system for monitoring the human activities from by employing a 3-axis accelerometer. We reviewed position sensing technology for human movement, compare different technologies and present analysis issues and current and potential application to human activity analysis[10]. The General Architecture and the detailed design of the proposed system, 3-Axis Human Activity State Detection and Monitoring System (3-HASDMS) has been discussed.

2.1 General architecture

It mainly comprises of two basic modules

- User Interface
- Sensing and Processing Unit

For User Interface we chose the a PC, capable of communicating to the Sensor processing module and at the same time to the external world. User Interface has the ability to process large amounts of data at high data rates and produce a human readable format.

Sensing and Processing Unit consists of a microcontroller and a sensor interfaced to the microcontroller. Purpose of the sensor is to sense the motions of a human be-

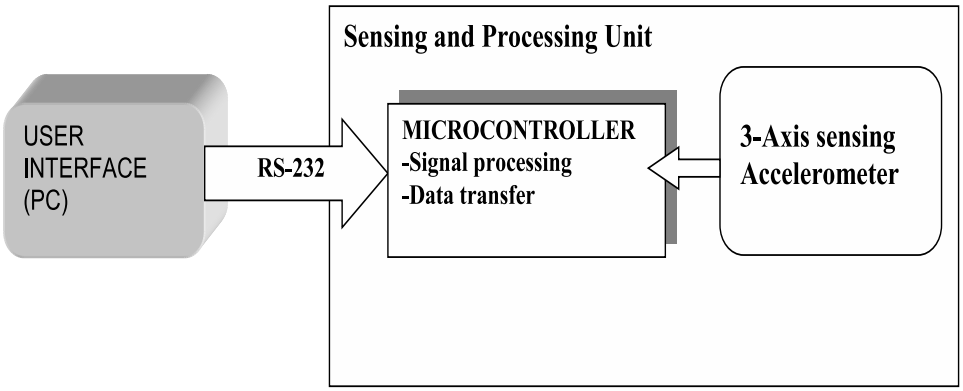

Figure 2.1 Architecture

ing and notify them to the microcontroller for further processing. There are different types of sensor which provide motion based information relevant for activity recognition such as accelerometers, gyroscopes and magnetic field sensors. As this design accounts for the human activities we opted for a low power, tri-axial Accelerometer (ADXL330). The microcontroller PIC18F4520 used is computational and high enduring with an economical price. This microcontroller has Universal Synchronous Asynchronous Receiver and Transmitter (USART) and a 10-bit up to 13 channel Analog-to-Digital converter module with which the family of microcontrollers will be a logical choice for this particular application. Main function of the microcontroller is to get the analog data from the accelerometer and convert the data back to digital format using its built-in Analog-to-Digital conversion module and send the data to the user interface using the serial communications.

Detailed discussions about the components used in the architecture are addressed below:

1. PIC18F4520
2. ADXL330
3. MAX232 and DB9 connector for Serial communications

2.2 Microcontroller PIC18F4520

The choice of microcontroller has many advantages than any other general purpose microprocessor as they require external memory and many other external peripherals for typical applications. The particular microcontroller PIC18F4520 follows Harvard Architecture which has different memory spaces for program and data and facilitates fast execution of instructions.

Peripheral highlights of PIC18F4520 include

- 40-Pin device with 36 pins available for I/O ports
- RISC based Architecture with 83 instructions and most executed in one cycle
- Data memory organized as 4Kx8-bits with 8-bit data bus and 12 bit address bus
- 1536 bytes of SRAM available
- Program memory organized as 1M x16-bits (21-bit address, 16-bit data bus) with 16K words of flash available
- Internal Analog-to-Digital converter, PWM and timers
- DC to 40MHz clock with instruction cycle time upwardly bounded by a 10MIPS limit
- 100,000 re-programmable Data and Program Memory
- Five different I/O ports named A, B, C, D, E but only one of it supports both analog and digital inputs
- Enhanced addressable USART module which supports RS-435 and RS-232 Using internal oscillator

2.2.1 Internal Analog-to-Digital converter

PIC18F4520 has an internal Analog-to-Digital Converter which is used to convert the analog signals from the accelerometer. ADC module can process 100ksps (kilo samples per second). ADC module supports 13-inputs and allows conversion of analog signal to a

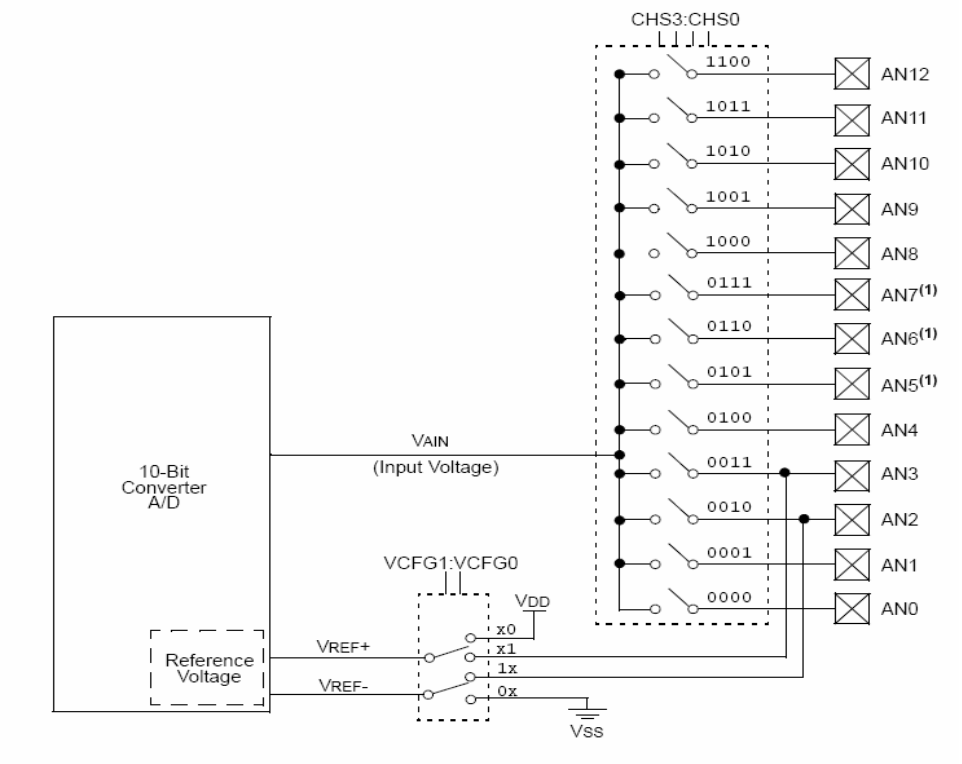

Figure 2.2 Analog-to-Digital converter

corresponding 10-bit digital number. Registers ADCON0, ADCON1, and ADCON2 are keys in configuring the functions of port pins, A/D conversion clock source, programmed acquisition time and justification consequently. The block diagram of ADC module with reference to the PIC18F4520 data sheet by microchip is shown in Figure 2.2

The analog reference voltage is software selectable to either positive or negative supply voltage(V_{dd} or V_{ss}) or voltage level on the RA3/AN3/V_{ref+} & RA2/AN2/CV_{ref-}/CV_{ref} pins. The ADC module has a unique feature of operating in the sleep mode by deriving the conversion clock from the A/D internal RC Oscillator. Once the conversion is started the same clock source should clock until the conversion is completed. The output of sample and hold is input into the converter, which generates the results via Successive Approximation. Successive Approximation is one of the analog to digital conversion

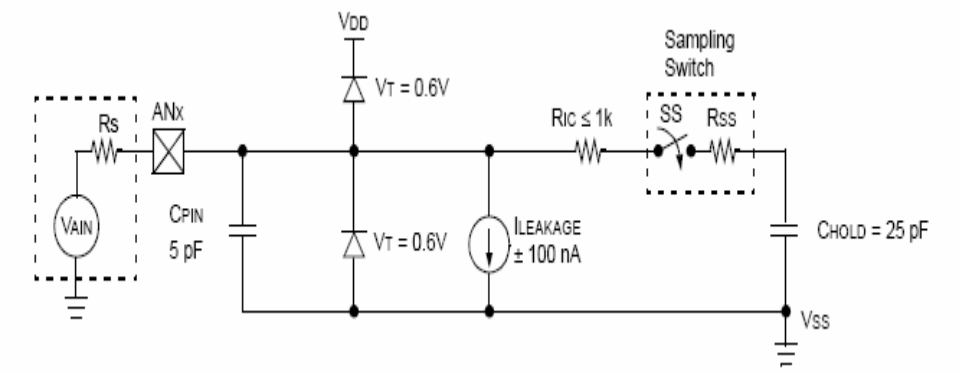

Figure 2.3 Analog input model of Analog-to-Digital converter

method where different values of bits are tried from MSB to LSB to get a binary number that equals the analog input. Analog input model showing the working of ADC module shown in Figure 2.3

For the A/D convertor to meet the specified accuracy the charge holding capacitor (CHOLD) should be allowed to charge till it reaches the input channel voltage level. From the above figure the source impedance (R_s) and internal sampling switch impedance (R_{ss}) directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch impedance (R_{ss}) varies over the device voltage. The source impedance affects the offset voltage at the analog input due to pin input leakage current so maximum recommended impedance for analog sources is 2.5KΩ. The discharge phase is used to initialize the value of the capacitor array. The Array will be discharged before every sample, this feature helps to optimize the unity gain amplifier, as the circuit always needs to charge the capacitor array, rather than charge and discharge based on the previous measures.

2.3 ADXL330

The ADXL330 is a small, thin, low power, complete three axis accelerometer with signal conditioned voltage outputs, all on a single monolithic IC. This MEMS based

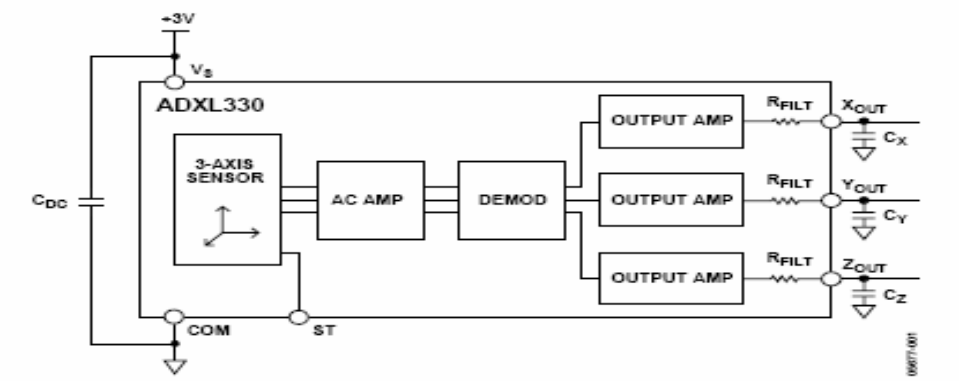

Figure 2.4 Block diagram of ADXL330

device measures acceleration with a minimum full-scale range of +/- 3g. Functional block diagram with reference to Analog Devices data sheet is shown in Figure 2.4

The ADXL330 uses a single structure for sensing the X, Y and Z axes. As a result, the three axes sense directions are highly orthogonal with little cross axis sensitivity. Mechanical mis-alignment of the sensor die to the package is the chief source of cross axis sensitivity. It can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration. It has 10,000g shock survival. The user selects the bandwidth of the accelerometer using the C_x, C_y, C_z capacitors at the X_{out}, Y_{out}, Z_{out} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz for X and Y axes, and the range of 0.5 Hz to 550 Hz for the Z axis. Pin configuration of the device and the operating principle is shown in Figure 2.5

The ADXL330 is an acceleration measurement system on a single monolithic IC. It contains poly-silicon surface micro machined sensor and signal conditioning circuitry to implement open-loop acceleration measurement architecture. The sensor is a polysilicon surface micro machined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor

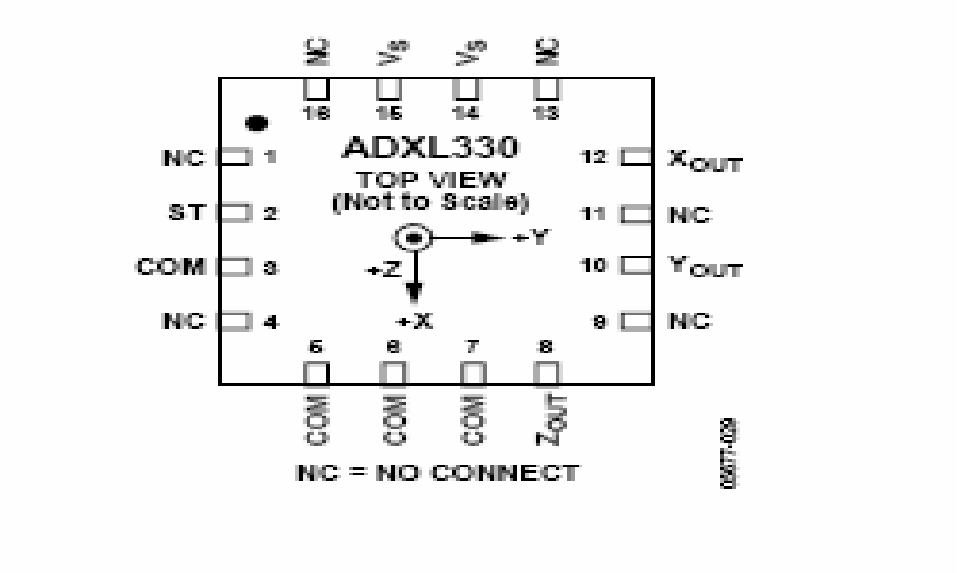

Figure 2.5 Pin configuration of ADXL330.

that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180 degrees out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase- sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration. The output signals are analog voltages that are proportional to acceleration. Rather than using additional temperature compensation circuitry, innovative design techniques ensure high performance is built-in to the ADXL330. So, there won't be neither quantization error nor non-monotonic behavior, and as a result the temperature hysteresis will be very low. ADXL330 has the provision for band-limiting the X_{out} , Y_{out} and Z_{out}. Capacitors must be added at these pins to implement low-pass filtering for anti-aliasing and noise reduction.

Table 2.1 Specifications of ADXL330.

Parameter	Range
Power Supply	2.0 - 3.6 V
Sensitivity	270 - 330 mV/g
Operating Temperature	-25^o to 70^oC
Zero g bias level	1.2 - 1.8

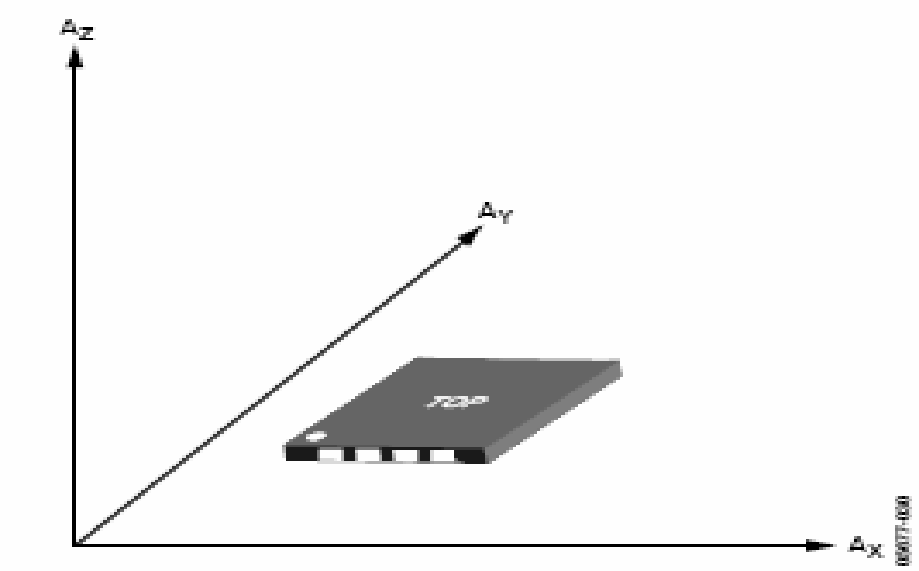

Figure 2.6 3-Axis sensing directions of ADXL330.

2.3.1 Axes of Acceleration Sensitivity

The voltage increase when the acceleration across that particular sensitive axis increases. The typical output response with respect to the orientation of the sensor is shown below. The accelerometer is placed in different positions as illustrated in Figure 2.7 and the values listed against that position are verified experimentally by collecting the 3-Axis data.

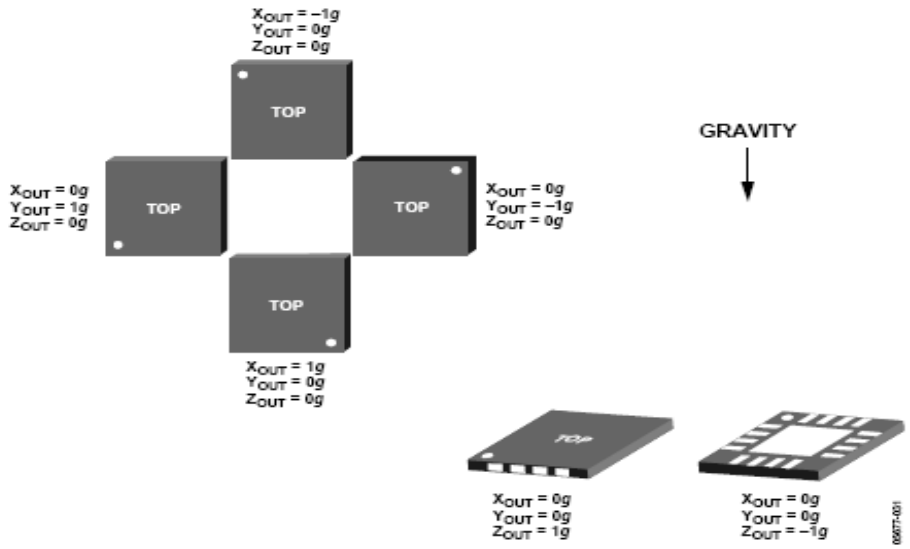

Figure 2.7 Accelerometer orientation.

2.4 Serial Communications

We need to establish a means of communication between the microcontroller and PC, where the data will be further processed into human readable format. PIC18F4520 supports EUSART (Enhanced Universal Synchronous Receiver Transmitter). The EUSART is one of the two serial I/O modules. EUSART can be configured as a full-duplex asynchronous communication system that can communicate with peripheral devices such as CRT terminals and personal computers. EUSART can produce only TTL level logic signals. So, level transition is required for system compatibility. This level transition is achieved by an MAX232 chip, which meets the TIA/EIA-232 F and ITU recommendation V.28. This Chip can operate with single 5V supply voltage and can handle up to 120kbits/sec. MAX-232 input thresholds are both TTL and CMOS compatible. The typical output voltage swing is +/-8V. A DB9 connector is required to establish communication with the serial port of PC. Hardware connections are as shown in Figure 2.8.

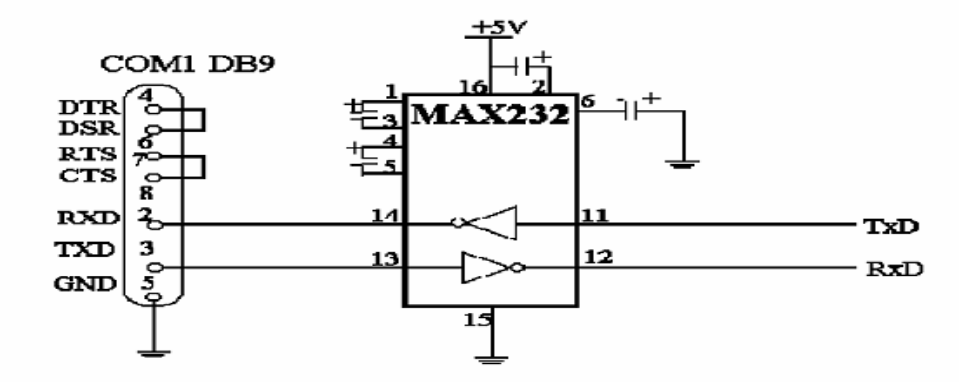

Figure 2.8 Integrated functional diagram a 3-Axis sensing system

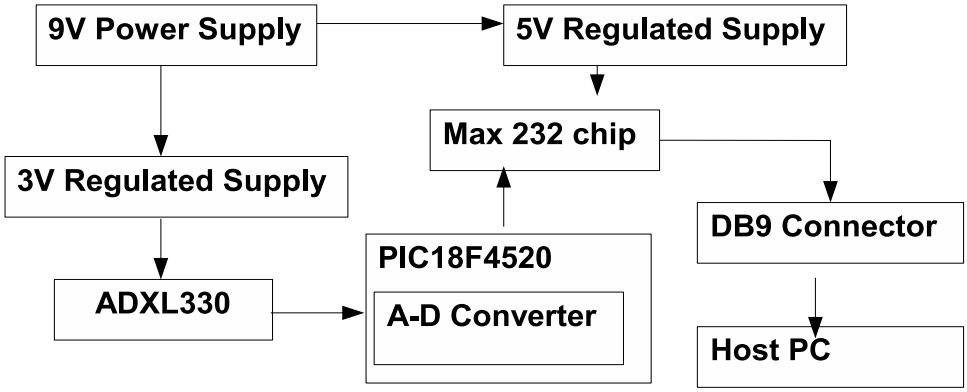

Figure 2.9 Integrated functional block diagram

2.5 Integrated Functional Diagram a 3-Axis Sensing System

The functional diagram shows the integrated functional block diagram of a 3-Axis Acceleration sensing system, which can be used for wide range of applications including motion and tilt sensing, gaming etc. The system can operate at various frequencies but the present design operates at the HS-PLL frequency which is 4 times the employed crystal frequency. In the present design we employed a 10 MHz crystal oscillator so, our overall system frequency will be 40 MHz. We used a 9V alkaline battery. LM2931, 5V 100mA voltage regulator is used to supply the operating voltage of 5V to the MAX232 chip. Another voltage regulator LE30 which converts 9v to 3V to provide the supply voltage for the PIC18F4520 and accelerometer.

ADXL330, a three axis acceleration sensing device provides analog voltages on three axis X, Y, Z which are proportional to the accelerations. As the device generates analog voltage, it is interfaced to the analog port A of PIC18F4520. X, Y, Z pins of the ADXL330 are connected to the AN2, AN1 and AN0 of PIC18F4520. The micro controller forwards the analog voltages to the internal A-to-D converter on three different channels, which converts the analog signal to a 10-bit digital number.

Pin numbers 25 and 26, Transmit and Receive pins, of PIC microcontroller are connected to the corresponding transmit and receive input pins 11 and 12 of MAX232 for the conversion of the TTL logic level to the PC compatible signal levels.MAX232 converts the voltages and passes the signals from the Transmit and Receive output pins 14 and 13 to the DB9 connector. The schematic diagram showing the internal pin connections including ADXL330,PIC18F4520,MAX232 and DB9 connector are shown in Figure A.1. Assembly of the components on a PCB according to the schematic can be seen in Figure A.2.

DB9 connector is used to communicate with the serial communication port of the PC. Here we used a Serial- to- USB converter cable to communicate serially to the PC using a USB port and the corresponding drivers are loaded. The detailed Software will be discussed in Chapter 3.

CHAPTER 3

SOFTWARE DESIGN AND IMPLEMENTATION

The circuit design to meet the criteria stated is build on a PCB board and is illustrated in Chapter 2. In order to correlate with the design issues we need to embedded some software into the system in a manner to make it work for our specific application. There are two different modules for our specific application. They are:

1. Assembly level module
2. User Interface module

3.1 Assembly Level Module

PIC18F4520 on the board should communicate with the accelerometer through its analog port to get the data from the accelerometer. Programming of microcontroller is done using its own instruction set. Steps followed to interface the analog device ADXL330 with the PIC18F4520 to convert the analog data to digital is as follows:

1. Initialization of hardware includes setting serial buffer transmission rate and deciding on transmission mode
2. Configure the A/D module:
    - Configure analog pins, Voltage reference and digital I/O (ADCON1)
    - Select A/D input channel (ADCON0)
    - Select A/D Acquisition time (ADCON2)
    - Select A/D Conversion clock (ADCON2)
    - Turn on A/D conversion module (ADCON0)
3. Configure A/D interrupt (if desired)

- Clear ADIF bit
- Set ADIE bit
- Set GIE bit

4. Start Conversion
   - Set Go/Done bit (ADCON0 register)
5. Wait for A/D conversion to complete by either
   - Polling for the Go/Done bit to be cleared
   - Waiting for the A/D interrupt
6. Read A/D Result Registers (ADRESH: ADRESL); clear ADIF bit if required
7. For next conversion, go to step 1 or 2 as required. A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next conversion starts

Configuring different registers is the key in determining the functionality of the chip to perform as we required. Registers which plays key role during this specific programming are:

1. ADCON0
2. ADCON1
3. ADRESH & ADRESL
4. ADIE
5. ADIF
6. TXREG

Brief discussion about each of the Registers is as follows.

ADCON0: This register controls the operation of the A/D module. ADCON0 register can be represented as shown below:

Bits 7-6 are unimplemented. Bits 5-2 are used as analog channel selection bits ranging from AN0 to AN12. Bit 1 is A/D conversion status bit. Bit 0 is A/D on

ADCON0 REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Figure 3.1 ADCONO register.

ADCON1 REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0(1)	R/W(1)	R/W(1)	R/W(1)
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Figure 3.2 ADCON1 register.

bit. The program uses AN0, AN1 and AN2 channels for the conversion of X, Y and Z bits respectively. A/D conversion status bit is used in determining whether the A/D conversion is done or not.

ADCON1: ADCON1 is used in configuring the functions of the port pins. The operations of pins RAADCON1 register is represented in Figure 3.2.

Bits 7-6 are unimplemented. Bits 5-4 are voltage reference configuration bits. Bits 3-0 are A/D port configuration control bits. We have selected the configuration control bits in such a way all the channels will be analog. In Addition to ADCON0 and ADCON1 there is one more register called ADCON2 used in selection of result format, acquisition time and clock speed.

ADRESH & ADRESL: Higher and lower eight bits of the A/D conversion result are stored in the ADRESH and ADRESL registers respectively. When the bit1 of ADCON0 is set the A/D conversion starts and when it is cleared the A/D conversion result pair is loaded with the conversion result. When the bit is cleared or aborted abruptly the ADRESH and ADRESL continues to hold the previous values and they won't be updated with the partially completed A/D conversion sample.

PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

Figure 3.3 PIE1 register.

PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

Figure 3.4 PIR1 register.

ADIE: Peripheral interrupt enable register (PIE) contain the individual enable bits for the peripheral interrupts. Bit 6 of PIE1 is A/D converter interrupt enable bit. PIE1 register organization is shown in Figure 3.3

Set and Reset of this particular bit enables and disables the A/D Interrupt.

ADIF: Peripheral interrupt Request register (PIR) contain the individual flag bits for the peripheral interrupts. PIR1 register organization is shown in Figure 3.4.

Bit 6 of PIR1 is A/D conversion interrupt flag. Set of this bit represent that A/D conversion is done and it should be cleared through software.

TXREG: The heart of the EUSART Asynchronous Transmitter is Transmit Shift Register (TSR).The shift Register obtains its data from the Read/Write Transmit Buffer Register TXREG. The TXREG is loaded with data in the software. The TSR is not loaded until the Stop bit has been transmitted from the previous load. As soon as the stop bit is transmitted, the TSR is loaded with the new data from the TXREG (if it is available).Once the TXREG register transfers the data to the TSR register, TXREG will be empty and TXIF(bit 4 in PIR1) is set. TXREG can be said as a key in EUSART transmission. Asynchronous Transmission setup is explained as follows:

- Initialize the SPBRGH and SPBRGL with appropriate baud rate. We have selected a baud rate of 9600 as it will match with the MAX232 bit rate range. Set or clear the BRGH and BRG16 bits as required to achieve the desired baud rate
- Enable the asynchronous serial port by clearing the bit SYNC and setting bit SPEN
- Mode of transmission either 8-bit or 9-bit is can be selected. In our program we selected 8-bit transmission
- Enable the transmission by setting bit TXEN which will also set TXIF bit
- Load data to the TXREG register (starts transmission).Data in the TXREG is flushed out at a rate of 9600 bits per second

  Flow chart of the Activity detection module is shown in Figure 3.5.

3.2 Programming Modules at Assembly Level

Some subroutines are written to support certain functions. Some of those subroutines are discussed below:

Serial_init_hi: This module takes care about setting the serial communication baud rate and Transmission mode. By sending a value of 255 into the SPBRG register sets the baud rate to 9600 bits per second.

Wait_ms: This routine helps in making the program wait until the Acquisition time of 20msec before it proceeds to another set of data.

Serial_put_data: Checks whether the Transmit buffer is empty, if not it will keep trying until there is some data. If it detects some data on the ports it flushes the data to the Serial communication cable.

Read_X: This particular module reads the value of X from the ADXL330 accelerometer through pin4 of the PIC microcontroller.

Read_Y: This module reads the value of Y from the accelerometer through pin3 of the microcontroller.

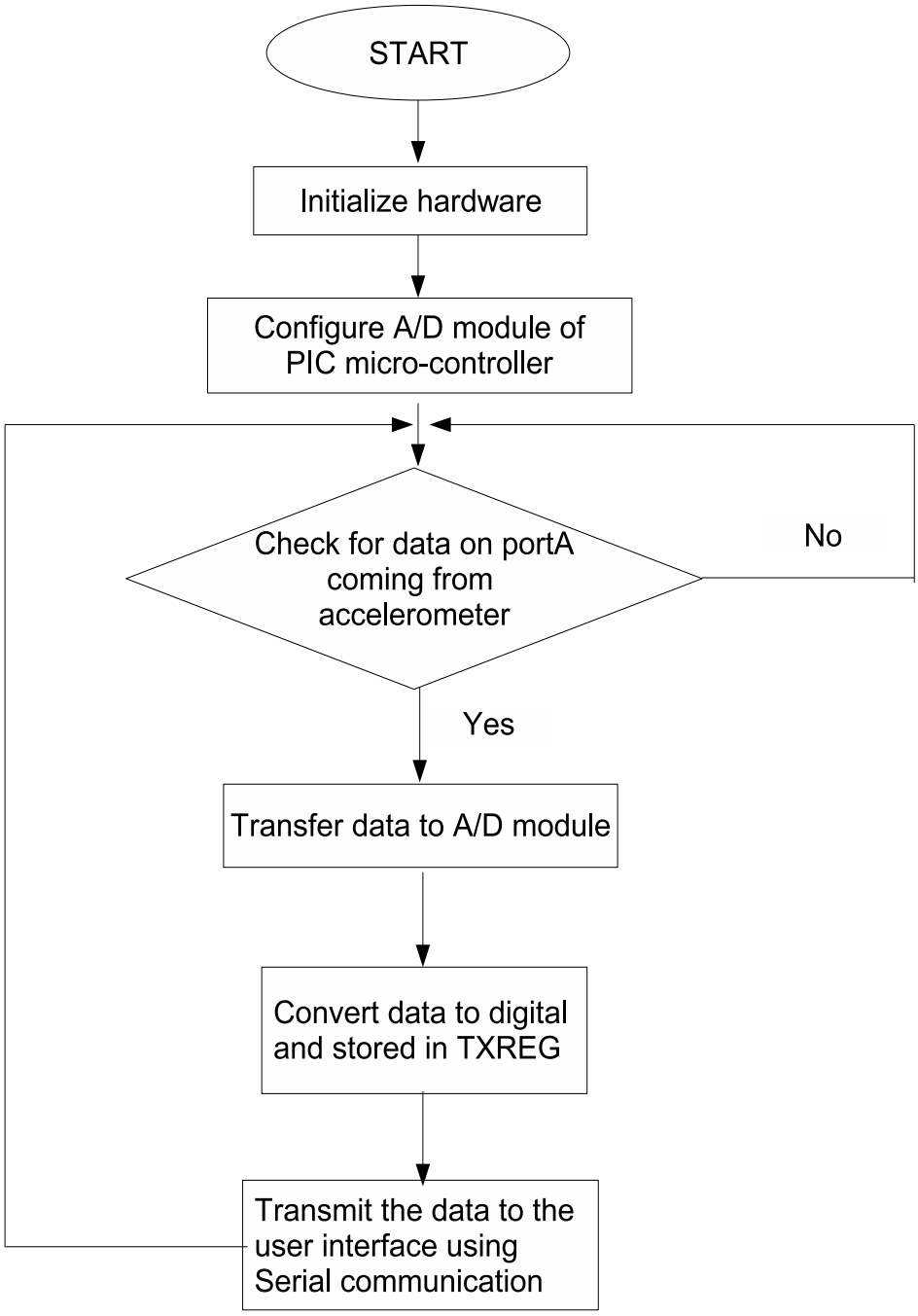

Figure 3.5 Activity state detection flow chart at assembly level.

Read_Z: This module reads the value of Z from the accelerometer through pin2 of the microcontroller.

The data read from these three above modules will be simultaneously transferred to the A/D converter module and converted to digital data using three different channels like AN0, AN1 and AN2.The converted data will be made available for the TXREG register to flush the data to the Serial communication cable with the help of Serial_put_data module.

3.3 User Interface Module

The user interface or human-machine interface is the aggregate of means by which people interact with a particular machine, device, computer program or other complex tool. Here in our situation user interface is the interface between the device developed and the human beings.

Initially we collected the data from the serial communication port using hyper terminal. But, hyper terminal is capable of displaying the data only in ASCII format. As ASCII format won't be user friendly we developed a user interface program which can read data from the Serial communication port using JAVA. The JAVA interface developed is a Graphical User Interface (GUI), which allows user to interact with a computer and computer controlled devices which employ graphical icons, visual indicators or special graphical elements called widgets along with text, labels or text navigation to represent the information and actions available to user. There are two different principles used in the GUI design namely Object Oriented user interfaces (OOUI's) and application oriented interface. This JAVA GUI interface with graphs representing the acceleration in all the three axes is developed using NETBEANS 5.5.1 environment. This JAVA program also writes an output file which creates a log of time and corresponding state of a person.

The user interface design using Flowchart is explained in Figure 3.6. User interface specification should be matched with the hardware design like the baud rate and number of transmission bits.

We selected the baud rate as 9600 bits per second and 8-bit transmission mode same as selected in the Assembly level program. Besides that user interface allows the user to select the port number from a list of ports that are active for the data collection from that port to start. The data collected from the port at the rate of selected baud rate will be in digital form. The JAVA code then converts the digital data into analog voltage according to the reference voltage applied at the hardware side. This analog voltage will be converted to three axes acceleration according to the Analog devices data sheet. As we won't be able to precisely notify the states of the system accurately we converted the accelerations in three axes to angle made with the earth. Depending upon the 3-axis angles different states like SIT/LIE, STAND, INVERSE LIE, WALK, and DROP will be detected and displayed in the output file. Several modules are written in JAVA to perform all the above functionality. Some of the important modules are discussed briefly.

Comportopen: This module detects all active serial ports connected to the PC and provides an opportunity to the user to detect the appropriate port. This module also sets the baud rate, transmission bits including start and stop bits. This module has the capability of differentiating the data coming from different axes.

ReadQueueConsumer This module reads the 3-Axes data from the Comportopen and converts to binary data using bintodec function. This module finally converts the binary data into angle from which it calls three different modules called XGraph, YGraph and ZGraph.

XGraph: XGraph is a time series chart where we can dynamically see an update of X-axes values with a range of +3 and -3 on Y-axis plotted against time in the X-axis.

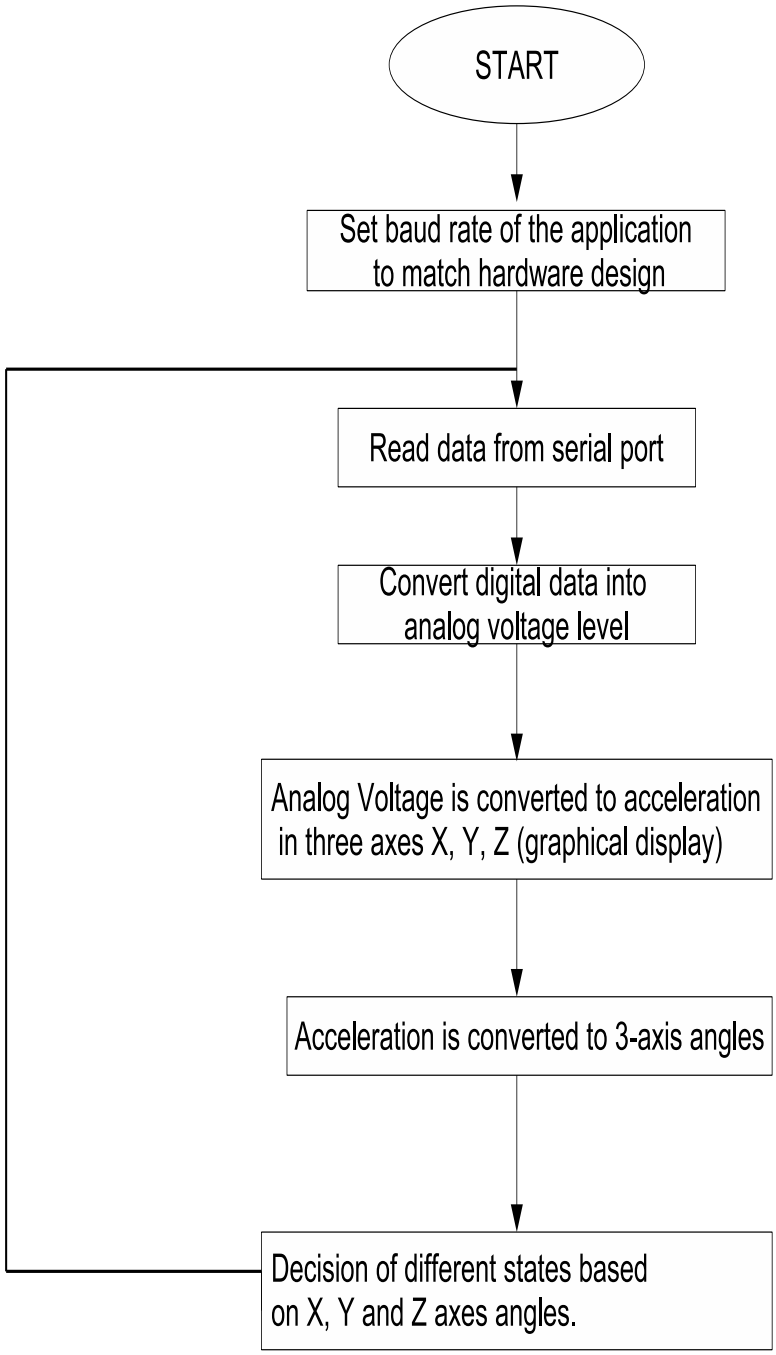

Figure 3.6 User interface flow chart.

YGraph: YGraph is a time series chart where we can dynamically see an update of Y-axes values with a range of +3 and -3 in accordance with the time.

ZGraph: ZGraph is a time series chart where we can dynamically see an update of Z-axes values with a range of +3 and -3 in accordance with the time. In addition to this there will be a text file written dynamically with time and the corresponding state.

Appendix B.1 addresses the Assembly level code that has been developed as part of the system design.

CHAPTER 4

SYSTEM EVALUATION AND EXPERIMENTAL RESULTS

Basic idea of the 3-axis motion sensing prototype is to measure the body movements of the elderly persons using a 3-axis accelerometer. A feature of the prototype developed is the sensor is compactly assembled as a wearable unit.

4.1 Objective

Estimation of human motion is important enabling technologies for realizing a pervasive computing environment. Objective of the experiment is to evaluate the system developed in real time environment by proposing an improved method for estimating states from accelerometer data is introduced. One major focus of the test is to verify the validity and accuracy of the device and compare the same with the previous developed systems. A series of experiments for testing the effectiveness of the proposed method has been performed and its results are presented in the rest of the chapter.

4.2 Characterization of Activity States

There are enormous varieties of activities possible for humans. Characterization of each state is key in monitoring the human activities[19]. Activities like Sit/Lie,Lie facing down, Stand, Walk and Drop are taken as our major concern for the analysis of the prototype developed. One major task is to identify, characterize and discriminate between different states chosen. First step towards characterization is defining the descriptions for each state. The description attempt to focus on features which are unique to each state and help identify it.[15].

Lying Down: Lying down is described as resting the body horizontally on the surface. The upper and lower extremity long bones are maintained in a relaxed horizontal position.

Sitting Down: Sitting is described as supporting the body on the posterior aspect of the thigh. The upper long bone is nearly vertical while the long bone of the thigh is horizontal. These positions are controlled by certain environmental issues (furniture, etc).

Standing: The fundamental standing position is described as being that of an erect body which is extended, with feet together. Long bones of the upper and lower extremities and the spine are usually in a near vertical position. Standing actually requires modest amount of muscle activity to maintain the body at equilibrium.

Inverse Lie: Inverse lie is same as lying down except that resting the body facing down to the surface. The upper and lower extremity long bones are maintained in a relaxed reverse horizontal position.

Walking: Human walking is a process of locomotion by which the erect moving body is supported ultimately by one leg and then the other. The position of the bones will be similar to the Stand position but with some motion to it which caused a range of deviation in the angle.

Drop: Drop can be described as a descending from a higher position to a lower. Here in our analysis we have thrown the sensor board from a certain height and acceleration to a lower position.

Angle and acceleration information of different states is given in the following table:

4.3 Sensor Placement

Placing the sensor at the right position is one more key in distinguishing the states proposed. The state angle and acceleration information discussed in Table is in regard

Table 4.1 Angle and acceleration for different states

Activity	Angle(X)	Angle(Y)	Angle(Z)	Acceleration	Deviation
Lie/sit	Horizontal	Horizontal	Vert. varied	Small	Small
Standing	Vertical	Vertical	Horiz. varied	Medium	Small
Inverse Lie	Horiz. but opposite	Horiz. but opposite	-	Small	Small
Walking	Vert. varied	Vert. Varied	Horiz. varied	Medium	Large
Drop	Horiz. varied	Horiz. Varied	Vert. varied	Large	Large

to the sensor placed on the thigh. One disadvantage of placing the sensor on the thigh is we are not able to distinguish between Sit and Lie properly as both shows the same set of X, Y, Z axes angles. In our present thesis we tried placing the sensor at different locations of the body, but all the places have some set of overlapping states. For example if we place the sensor on my leg we are not able to sense the difference between Sit and Stand but it is able to distinguish between Sit and Lie down.

4.4 Experimental Setup

Experiments are conducted on live human beings which responds to the command given by the coordinator. We mounted just the accelerometer sensor on a separate board in order to make it movable. The sensor board is made worn on the thigh of a person and asked to follow the sequence of states said by the coordinator. Once the environment is set physically, we need to "RUN" the program in JAVA NETBEANS environment. Physical setup will be seen in Figure 4.1[1]. In this sequence each state will last for around 10-20 seconds. NETBEANS JAVA environment will produce 3 dynamic graphs for 3-axis acceleration as a function of time. In addition to the graphs there will be an output file written which keeps log of the time with that corresponding state opposite

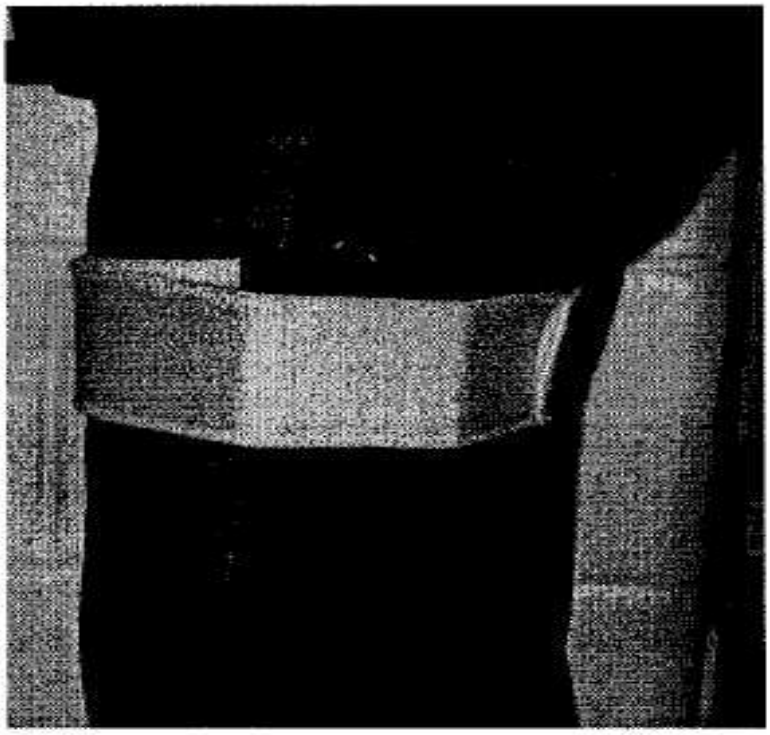

Figure 4.1 Person wearing a sensor attached to the thigh is shown.

to it is written to *output.txt* file. Example snapshot of the *output.txt* file for Sit/Lie has been shown in Figure C.1.

4.5 Orientation of 3-Axis Accelerometer

Orientation of the accelerometer and their axis description is shown in the 4.2.

On the accelerometer we can see a dot placed which has to be taken as the reference for the X axis, the one perpendicular to it as Y-axis, the axis perpendicular to both of these is Z-axis. The board worn on the thigh will be arranged in such a way that the X-axis will be parallel to earth and Y and Z are perpendicular to the direction of earth. The direction f earth gravity is in Figure 4.3.

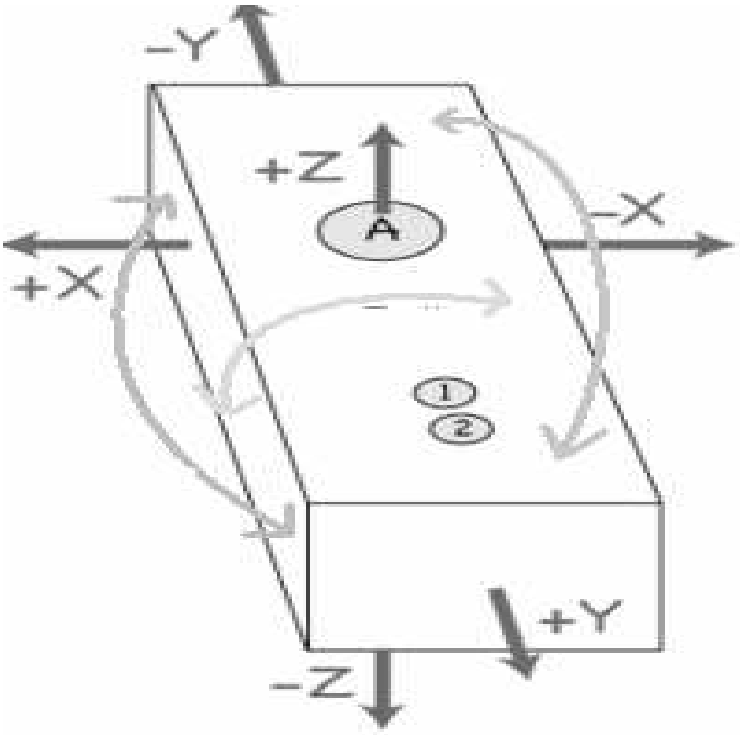

Figure 4.2 Rotation of axis.

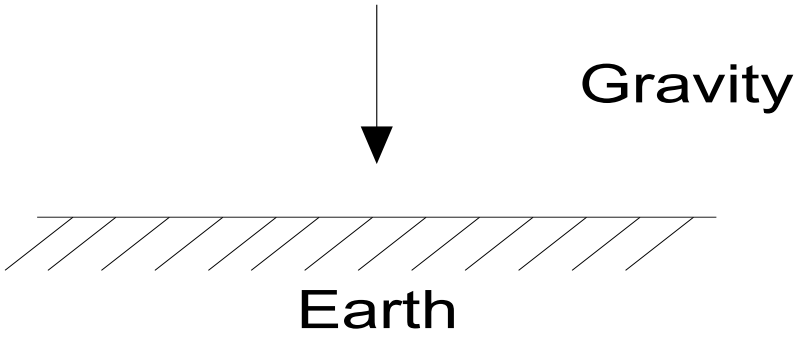

Figure 4.3 Direction of gravity

Table 4.2 Acceleration and the sensitivity observed at the three axes.

Axes/Parameter	X	Y	Z
Voltage(V)	2.987	2.987	2.987
Sensitivity	298mV/g	298mV/g	298mV/g
0g reference voltage	1.4935V	1.4935V	1.4935V

Voltage supply given to the accelerometer and the sensitivity observed at the three axes is given in Table 4.2

4.6 Acceleration and Angle Calculations

In order to determine the acceleration and angle of tilt represented by Δ, the A/D converter on the microcontroller will be sampled by the ADC channel. The digital 10-bit number is collected by the serial port of the PC and is converted to an analog voltage in the JAVA interface and is called V_{out} in general. Voltage at all three axes X_{vol}, Y_{vol}, Z_{vol}.

$X_{vol} = (X_{bin} * 2.987)/1024$

$Y_{vol} = (Y_{bin} * 2.987)/1024$

$Z_{vol} = (Z_{bin} * 2.987)/1024$

where X_{bin} , Y_{bin} , Z_{bin} are the binary values read through the serial port.

4.6.1 Acceleration calculation

X_{acc}, Y_{acc} and Z_{acc} can be determined from the three axes voltage as follows

$X_{acc} = (0_g voltagereferenceofX - X_{vol})/(\Delta V/\Delta g)$

$Y_{acc} = (0_g voltagereferenceofY - Y_{vol})/(\Delta V/\Delta g)$

$Z_{acc} = (0_g voltagereferenceofZ - Z_{vol})/(\Delta V/\Delta g)$

where 0g voltage reference of X, Y and Z is 1.4935

$(\Delta V/\Delta g) = 0.2987$

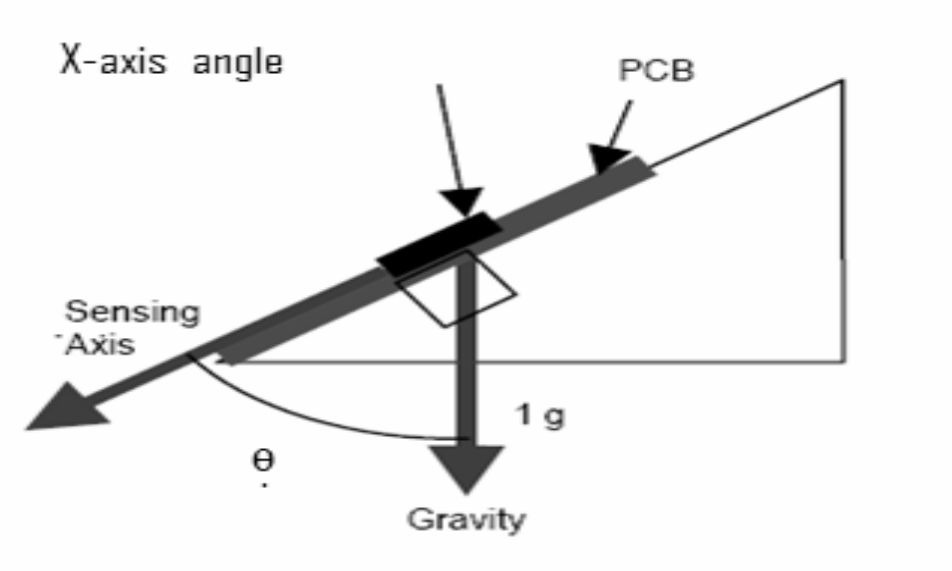

Figure 4.4 Angle calculation.

The acceleration is compared with the 0g offset to determine whether it is positive or negative acceleration, if the value is greater than offset then the acceleration is said to be positive and if it is less than offset then the acceleration is said to be negative.

4.6.2 Angle calculation

Acceleration should be converted to the angle information such that we can pass on these values to the position determining algorithm to specify the position of the subject. The general output voltage formula is given as follows:

$V_{out} = V_{offset} + (\Delta V/\Delta g * 1_g * sin\theta)$

- V_{offset} is Accelerometer 0_g offset
- 1g is earth's gravity
- θ is the angle of tilt

Example showing the tilt θ across the sensing axis is shown in the Figures 4.4 and 4.5

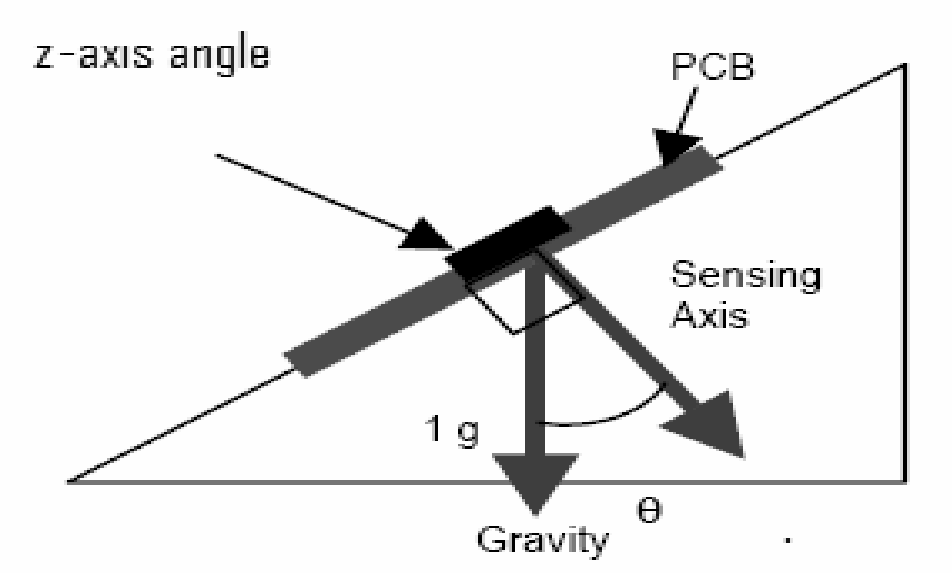

Figure 4.5 Angle calculation.

By re-arranging the V_{out} equation given above we get the tilt or angle equation as follows:

$\theta = arcsine(V_{out} - V_{offset})/(\Delta V/\Delta g)$

The typical output of the capacitive micro- machined accelerometer is more like a sine function. The figure shows the analog output voltage sweep from the accelerometer for degrees of tilt ranging from -90^o to 90^o. The change in degree of tilt is directly corresponds to the change in acceleration due to a changing component of gravity acted on the accelerometer. The slope of the non-linear curve shown in Figure 4.5 will be the sensitivity of the device. As the device is tilted from 0^o its sensitivity decreases. Because of this non-linearity, the degree resolution of our application must be at 0^o or 90^o to ensure lowest resolution is still with in the required application resolution.

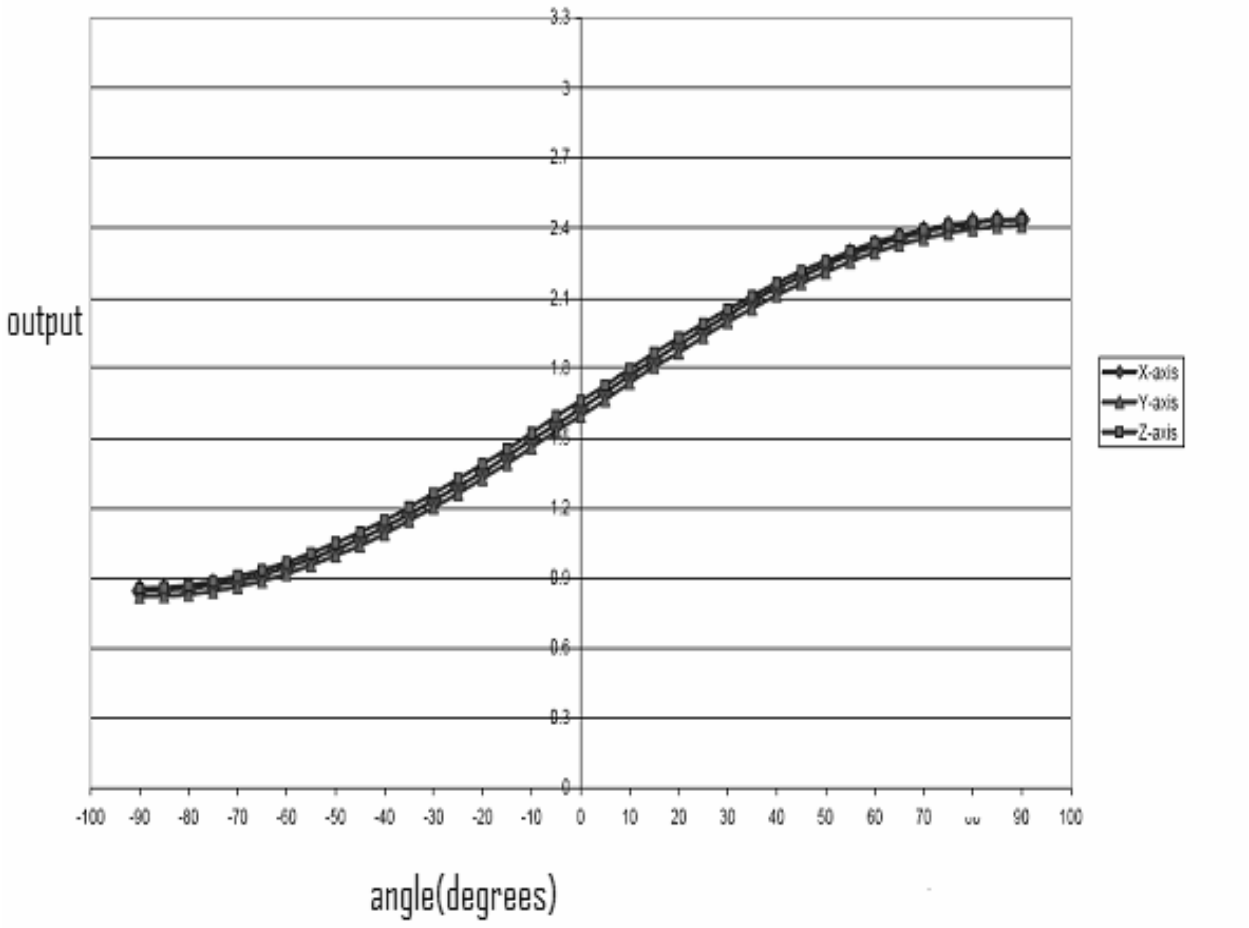

Figure 4.6 Comparison of change in voltage as a fuction of angle

4.7 Decision Criteria and Algorithm

The acceleration and angle corresponding to the position of the accelerometer are calculated and stored in the corresponding registers. Depending upon the angles in X, Y and Z axis we made decisions for different states. Range of Acceleration values is given in the following table:

Range of angle values in each direction to decide upon different states is as follows:

Table 4.3 Range of acceleration.

States/Axis	Sit/Lie	Inverse Lie	Stand	Walk
X	0_g to -0.15_g	0_g	0_g	$\geq -0.35 \& \leq -0.15_g$
Y	0_g	0.15_g	$\geq -0.9_g \& \leq -1_g$	≥ -0.8 to $\leq -1_g$
Z	$\geq 0.9 \& \leq 1_g$	$\leq -0.9_g \& \geq -1_g$	-0.15_g	$\geq -0.35 \& \leq 0_g$

Table 4.4 Range of angle.

States/Axis	Sit/Lie	Inverse Lie	Stand	Walk
X	-	-	≥ 0 and ≤ 10	≥ 0 and ≤ 20
Y	-	-	≥ 70 and ≤ 90	≥ 55 and ≤ 90
Z	≥ 70 and ≤ 90	≤ -70 and ≥ -90	-	-

For deciding upon the Walk state we have taken one more criteria i.e keeping track of three consecutive values of X, Y and Z such that for a walk state the values of the consecutive will be changing over the range specified in the Table 4.4.

Summary of different states is graphically illustrated in Figures 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14 and 4.15. In all these graphs acceleration is represented in g and angle is represented in degrees.

4.7.1 Summary of graph analysis

- In Sit/lie state we observe an appreciable angle only in the direction of Z
- In Stand state we observe a maximum angle only in the direction of Y
- In case of Walk the angle stays the same as of the stand but with some deviation in the Y angle due to continues movement and change in acceleration
- The angle observed in Lie facing down will be exactly the opposite of the Sit/Lie state

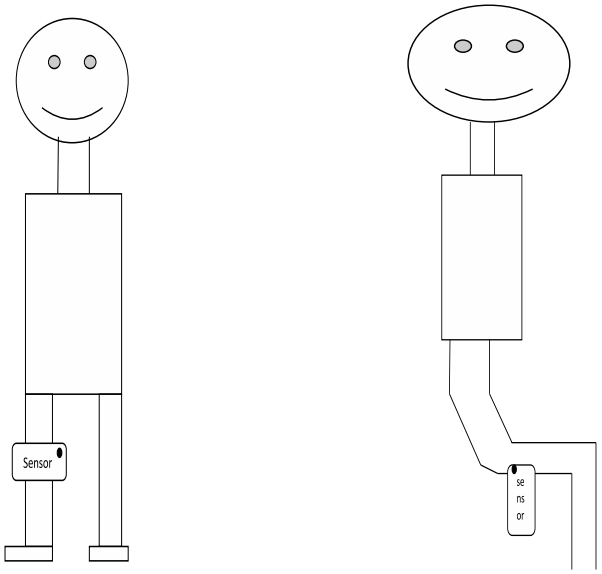

Figure 4.7 Stand and Sit/Lie positon of a person with sensor wearing on his thigh

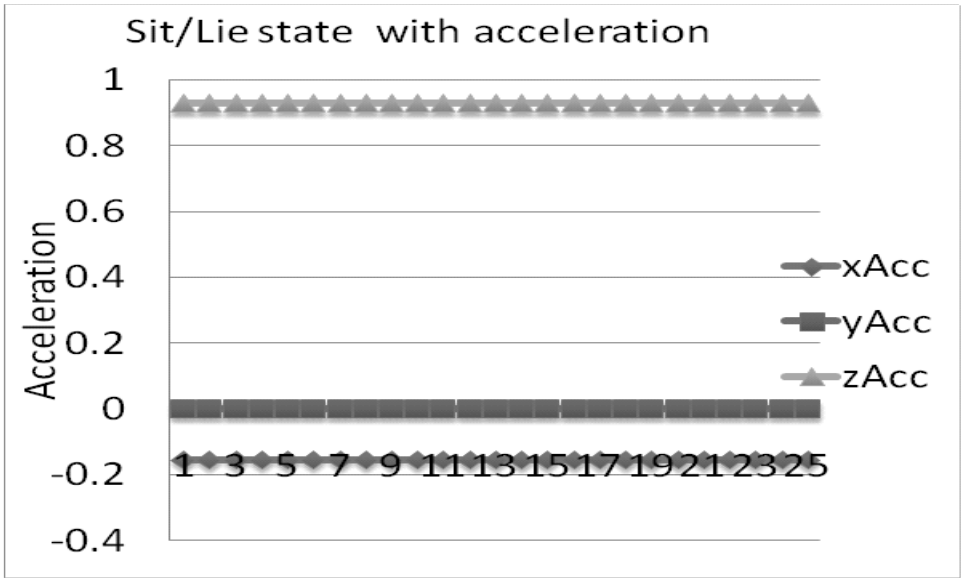

Figure 4.8 Acceleration change during “Sit/Lie” state

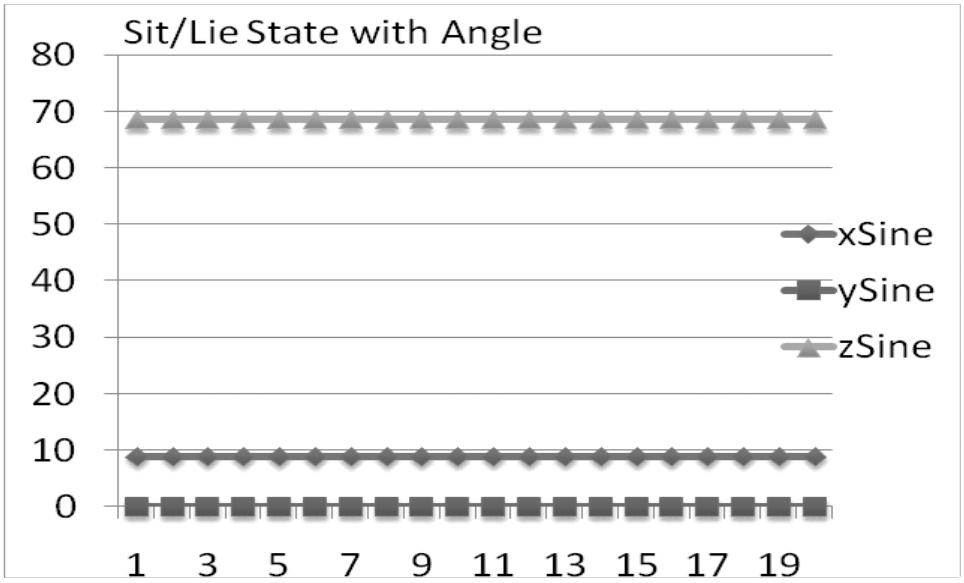

Figure 4.9 Angle change during "Sit/Lie" state

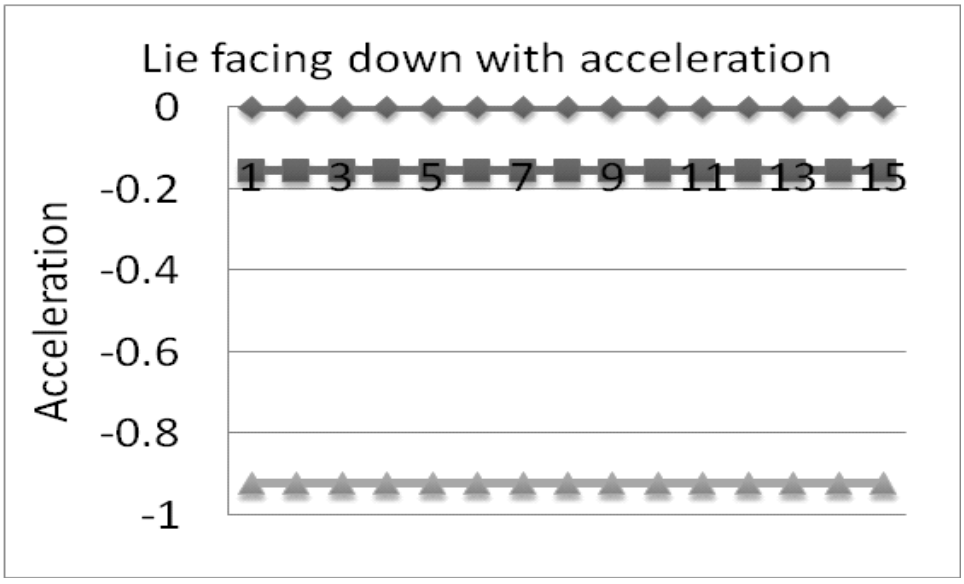

Figure 4.10 Acceleration change during "Lie facing down" state

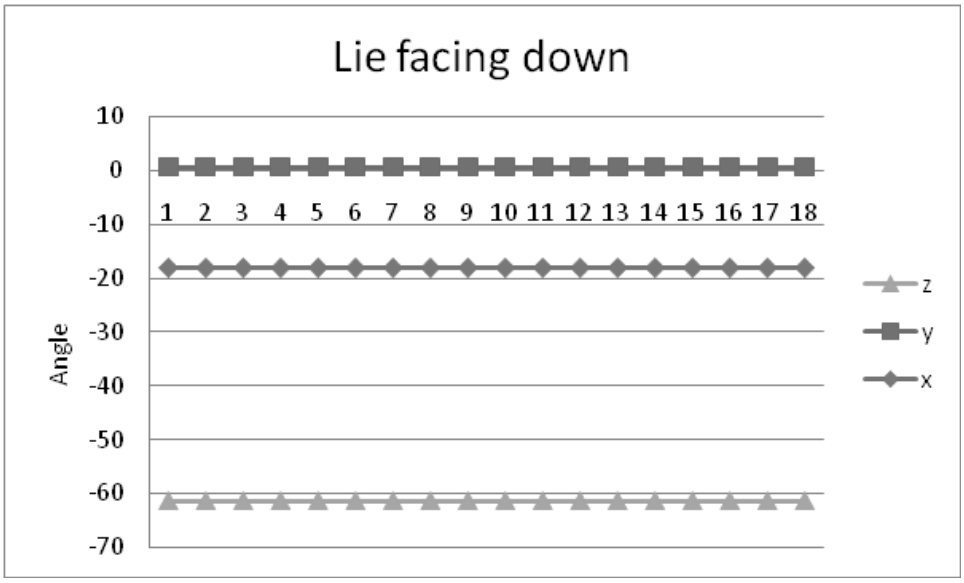

Figure 4.11 Angle change during "Lie facing down" state

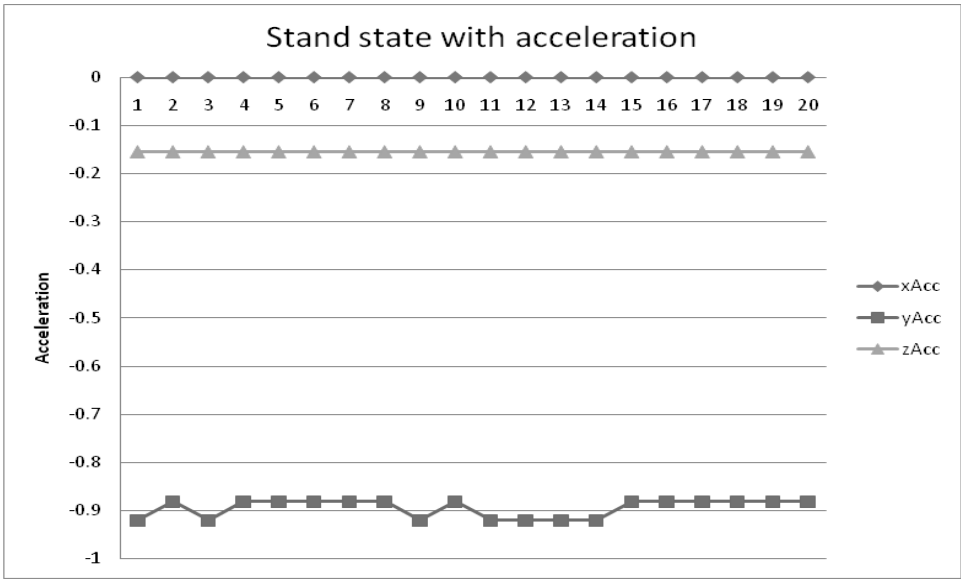

Figure 4.12 Acceleration change during "Stand" state

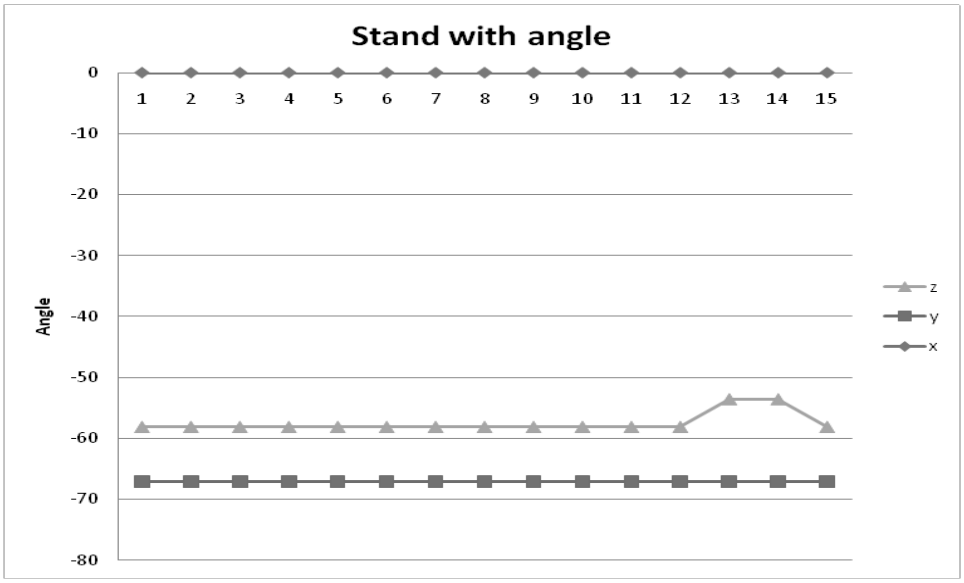

Figure 4.13 Angle change during "Stand" state

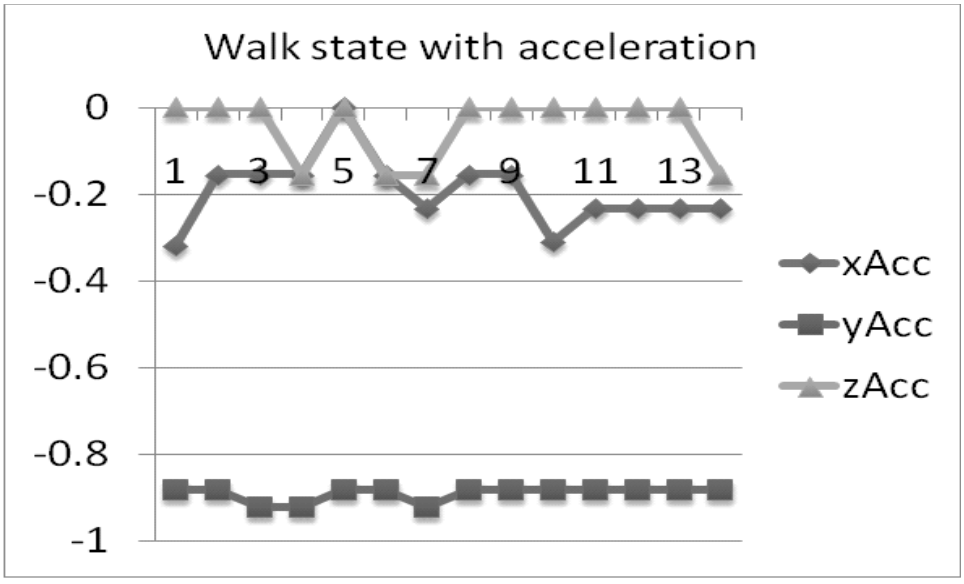

Figure 4.14 Acceleration change during "Walk" state.

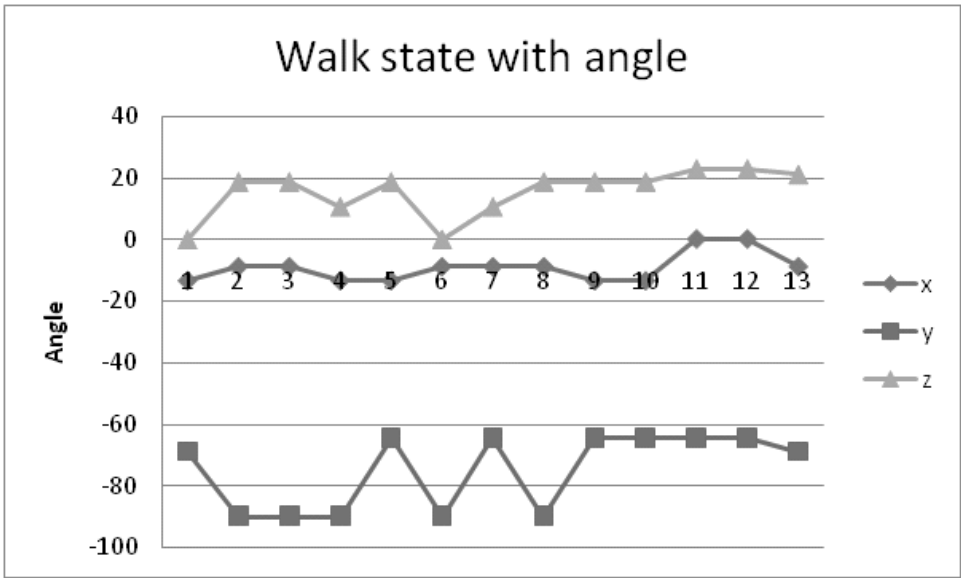

Figure 4.15 Angle change during "Walk" state.

4.7.2 Drop State

We have observed a sudden rise in the acceleration of 2.14 g in the direction of subject drop. So, if the acceleration is observed more than 2.14g then the state is declared as Drop in that particular direction.

4.8 System Evaluation

The position of the sensor when a person is siting or standing is shown in 4.7. Once the sensor is set physically on the person's thigh,each state is maintained for about 10-20 seconds. During this period 100 samples were collected. By taking numerous samples of a single state we calculated the mean of all the samples and ,finally evaluated the performance of the proposed activity monitoring system is shown in the Table 4.5. System evaluation represents the correctness of each state for the samples collected. Table 4.5 represent the percentage of times a particular state is recognized correct.

Table 4.5 System evaluation.

Actual/Detected	LIE/SIT	STAND	INV-LIE	WALK	DROP	TRANSITION
LIE/SIT	98%	-	-	-	-	2%
STAND	-	97%	-	-	-	3%
INV-LIE	-	-	92%	-	-	8%
WALK	-	11%	-	80%	2%	7%
DROP	2%	-	-	4%	91%	3%

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Tasks Done

Human activity monitoring system with 3-axes sensing is used to detect the basic activities of a human being with increased performance in different aspects than previous designs.

Different tasks done as part of thesis are:

- Hardware design capable of sensing the data from the accelerometer and converting the analog data to digital
- *RS*232 protocol based communication to transmit the digital data to the PC through serial communication port
- JAVA interface to provide a graphical user interface showing the X, Y and Z axes acceleration dynamically
- Experiments done on different human test subjects got numerous samples for different states and mean of the samples is calculated to determine the accuracy of the states

5.2 Cost per unit

Cost of all the individual components in building the circuit and the manufacturer/Distributor name is addressed in Table 5.1.

5.3 Percentage of Error

The percentage of error is shown in Table 5.2.

Table 5.1 Cost per unit.

Component	Manufacturer/Distributor	Unit price($)
PIC18F4520	Microchip	3.35
MAX232	Microchip	0.43
ADXL330	Analog devices	5.45
PCB board	Twin board	5.85
LED(red)	Mouser	0.05
Header	Mouser	0.04
0.1μcapacitor	Mouser	0.06*3
10μcapacitor	Mouser	0.05
	Grand total	$ 15.4

Table 5.2 Percentage of error.

STATE Error	percentage(%)
Sit/Lie	2
Stand	3
Reverse Lie	8
Walk	20
Drop	9

5.4 Advantages over Previous designs

In addition to the emphasis on the research methodology, this thesis investigated the previous generation models in terms of data acquisition, hardware design, software design and provided an improvised version. Present design has achieved the goals set by the previous design. Improvements realized over the previous systems are as follows:

- 3- axis sensing to detect and monitor the activities.
- Reduced cost than previous systems. Present design can be realized almost at a cost less than the second generation system.
- Improved accuracy by shifting the threshold voltages in the decision making algorithm.
- Low probability of error.

- Increased precision by adoption of 10-bit ADC.
- Due to the advancements in VLSI technology we can avoid using external memory and ADC units by employing a PIC microcontroller.
- Better software design.

5.5 Applications of the System

The main application of the proposed system is to serve the bedridden elderly person. Increase of the aged persons is becoming a serious social problem. In the care house of elderly only a few care workers work for many aged people. One serious problem is injury of aged due to slipping and falling down on the floor[. Once they broke their legs and feet it is not easy for them to recover sooner. One simple answer to protect such cases is to fasten the patient onto the bed. But such kind of treatment is neglecting the human right of the patient. Keeping all the above in view the proposed system will be able to serve the bed ridden elderly person in a better way. If this wearable system is worn on the thigh of a person the person is free to move with minimum level of supervision. Besides the application quoted above there are some tilt based applications for the proposed system. They are:

- In cell phone market and hand held electronics market tilt applications can be used for controlling menu options, e-compass compensation, image rotation, or function selection in response to different tilt measurements.
- In medical markets, tilt is used for making the blood pressure sensors more accurate.

5.6 Future Work and Recommendations

The present design is capable of achieving the proposed objectives, but still can be extended in many ways. Some of the recommendations are:

- Employing a wireless technology makes the system portable, so the system can be extended using a wireless module[16].
- If we use a good gyroscope instead of a 3-axis accelerometer the system can be extended for detecting free-motion state detection.
- The system can be designed for multitude of product, such as game controllers, virtual reality input devices, HDD portable products, computer mouse, cameras, projectors, washing machines and personal navigation systems.
- The system can be extended for detecting and monitoring different states like climbing the stairs etc., taking the present design as the basis.

APPENDIX A

CIRCUIT DIAGRAMS

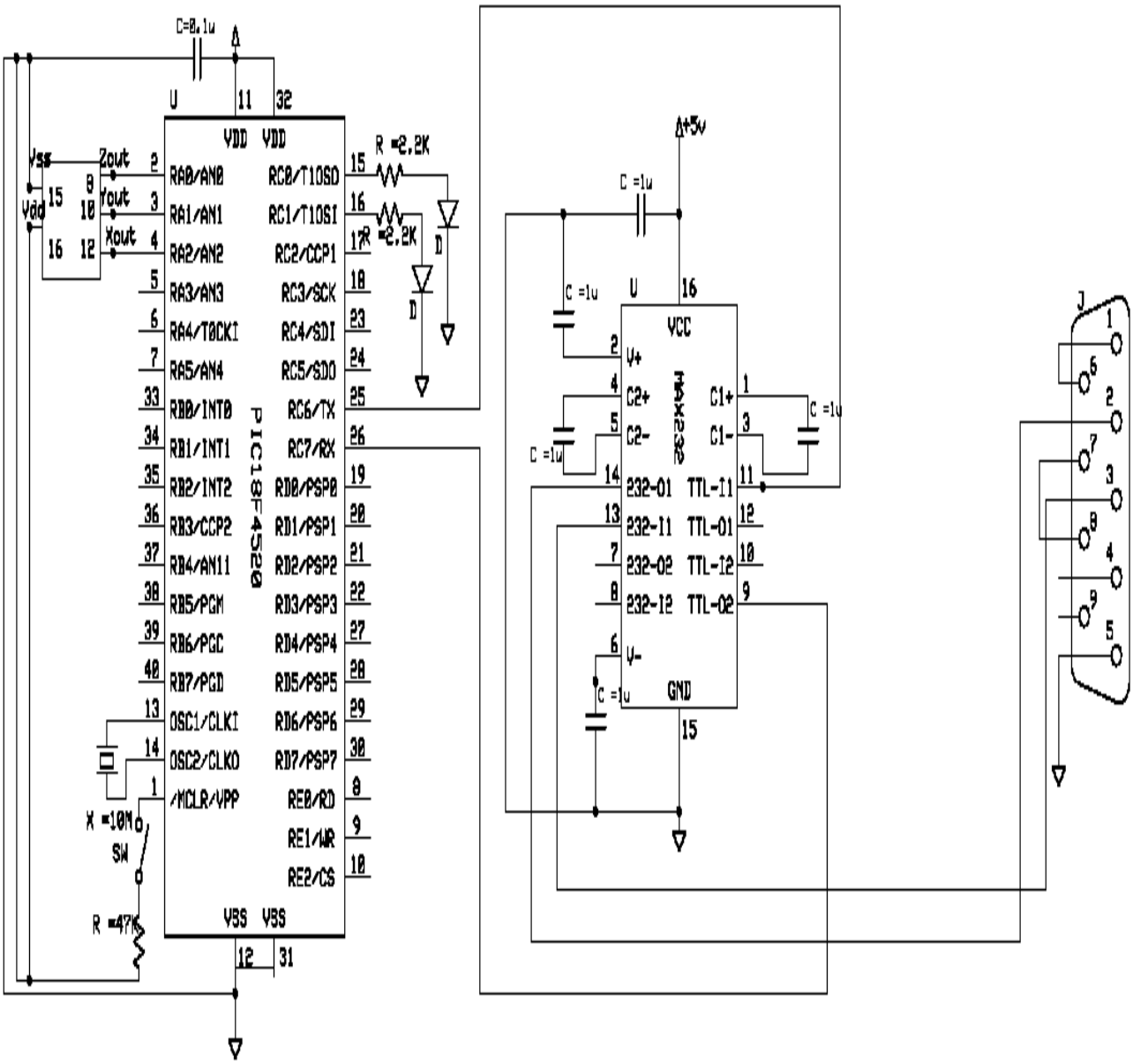

Figure A.1 Schematic diagram showing the pin connections of ADXL330, PIC18F4520 , MAX232 and DB9 connector.

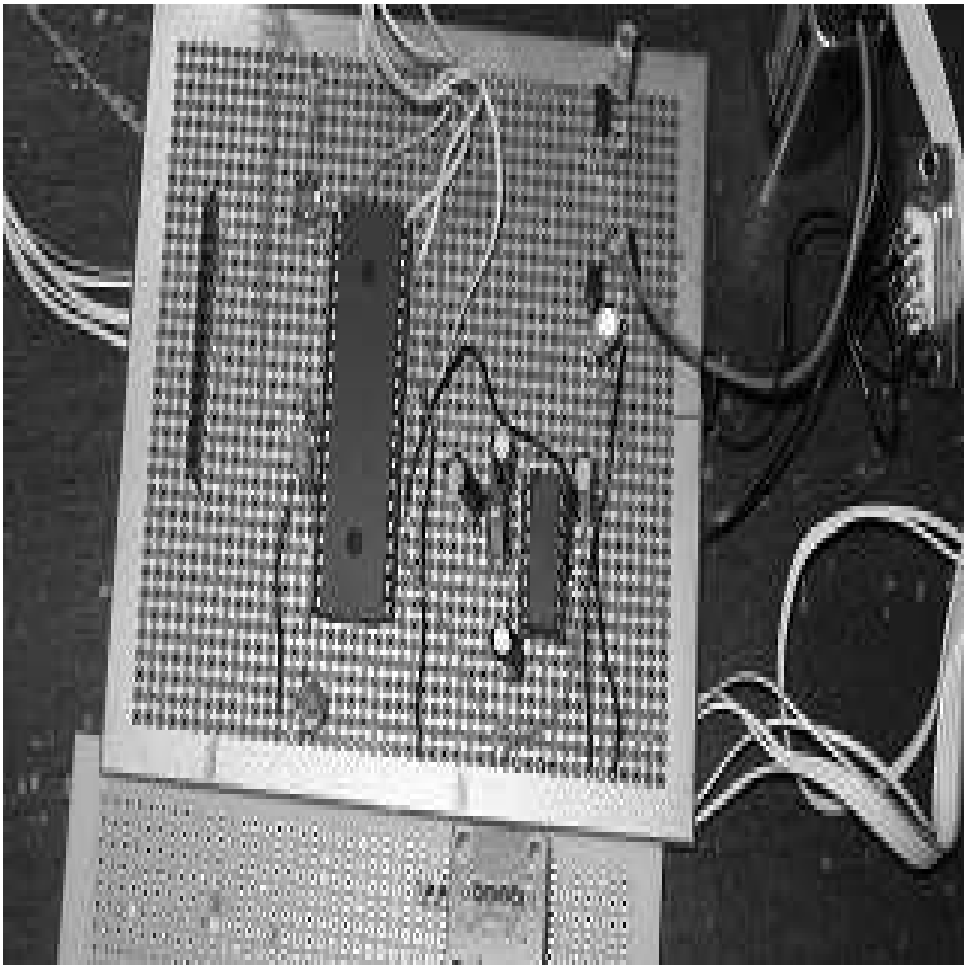

Figure A.2 Components assembled on a PCB board according to the schematic

APPENDIX B

CODE

B.1 Assembly Code

```
;-------------------------------------------
    ACCELEROMETER INTERFACE MODULE
;-------------------------------------------
; Assembler directives and device include
;-------------------------------------------
radix dec               ;make base-10 the default radix
errorlevel 0, -302      ;suppress bank warning messages
#include "P18F4520.inc"
;-------------------------------------------
; Macros
;-------------------------------------------
tblptr          macro   addr
                movlw   addr & 0xFF
                movwf   TBLPTRL
                movlw   (addr >> 8) & 0xFF
                movwf   TBLPTRH
                endm
;-------------------------------------------
; Pin aliases
;-------------------------------------------
#define PIN_X         LATA, 2
#define PIN_Y         LATA, 3
#define PIN_Z         LATA, 4
#define BAUD_19200    255
#define RLED          LATC, 1
```

```
;-----------------------------------------
; Variables
;-----------------------------------------
 XVALUE_L  equ    40h
 XVALUE_H  equ    41h
 YVALUE_L  equ    42h
 YVALUE_H  equ    43h
 ZVALUE_L  equ    44h
 ZVALUE_H  equ    45h


;-----------------------------------------
; Reset, Interrupt, and Startup Handlers
;-----------------------------------------
org     00h             ;force assemble
                        ;to place code here
goto    main
;-----------------------------------------
str_X       DB        "X", 0
str_Y       DB        "Y", 0
str_Z       DB        "Z", 0
org 0x50
#include "c:\project1\functions.inc"
-----------------------------------------
;main routine startpoint;
main
      bcf TRISC,1
```

```
      bsf RLED
      movlw     BAUD_19200
      call      serial_init_hi
      bsf       INTCON,GIE   ; to set the interrupts both
                             ;global and pheripheral.
      bsf       INTCON,PEIE
      bsf       PIE1,TXIE
label call read_X
      call read_Y
      call read_Z
      goto label

;-------------------------------------------

;-------------------------------------------
;subroutines
;-------------------------------------------
read_X  clrf PORTA
        clrf LATA
        bsf TRISA,2  ; initialization of
                     ;port A to input
        movlw 09h
        movwf ADCON0 ;initializing ADCON0
        movlw 80h
        movwf ADCON1
        movlw 20h
```

```
        call wait_ms ;wait till the
         ;acquisition time elapses
        movlw 0Bh ; start a to d conversion
        movwf ADCON0
        bcf   PIR1, ADIF
        bsf   PIE1, ADIE
        ;bsf   INTCON,GIE
        tblptr  str_X
        call    serial_const
loop  btfsc ADCON0,1 ;test whether the
;a to d conversion done?
        bra loop        ;no
        movf ADRESL,W
        movwf XVALUE_L
        call serial_put_data
        movf ADRESH,W
        movwf XVALUE_H
        call serial_put_data
        return

read_Y  clrf ADCON0
        clrf ADCON1
        bsf TRISA,1
        movlw 05h
        movwf ADCON0 ;initializing ADCON0 register
         ;to on the AtoD converter
```

```
        movlw 80h
        movwf ADCON1
        ;bsf ADCON2,ADFM
        movlw 20h
        call wait_ms
        movlw 07h
        movwf ADCON0
        bcf   PIR1, ADIF
        bsf   PIE1, ADIE
        ;bsf   INTCON,GIE
        tblptr  str_Y
        call    serial_const
loop1   btfsc ADCON0,1
        bra loop1
        movf ADRESL,w
        movwf YVALUE_L
        call serial_put_data
        movf ADRESH,W
        movwf YVALUE_H
        call serial_put_data
        return
read_Z clrf ADCON0
       clrf ADCON1
       bsf TRISA,0
       movlw 01h
       movwf ADCON0 ;initializing ADCON0 register
```

```
         ;to on the A to D converter
        movlw 80h
        movwf ADCON1
         ;bsf ADCON2,ADFM
        movlw 20h
        call wait_ms
        movlw 03h
        movwf ADCON0
        bcf   PIR1, ADIF
        bsf   PIE1, ADIE
        ;bsf   INTCON,GIE
        tblptr  str_Z
        call    serial_const
loop2   btfsc ADCON0,1
        bra   loop2
        movf  ADRESL,w
        movwf ZVALUE_L
        call  serial_put_data
        movf  ADRESH,W
        movwf ZVALUE_H
        call  serial_put_data
        return

sleep
end
;-----------------------------------------------
```

```
SUPPORTING FUCTIONS
;-----------------------------------------------
;-----------------------------------------------
; Functions
;-----------------------------------------------
LOOP_OUTER      equ     0ah  ;used by wait_10us
LOOP_INNER      equ     0bh
LOOP_MS         equ     0ch  ;used by wait_ms
LOOP_100MS      equ     0dh  ;used by wait_100ms

wait_10us       movwf   LOOP_OUTER   ;1 tick
wait_10us_loop  movlw   32   ;1 tick / 10us (N=32)
   movwf   LOOP_INNER      ;1 tick / 10us
wait_10us_loop2 decfsz  LOOP_INNER, F   ;N+1 ticks / 10us
   bra     wait_10us_loop2 ;2(N-1) ticks / 10us
   decfsz  LOOP_OUTER, F   ;1 tick / 10us + 1 tick
   bra     wait_10us_loop  ;2 ticks / 10us - 2 ticks
   return                  ;2 ticks

wait_ms         movwf   LOOP_MS ;store time in 1ms resolution
wait_ms_loop    movlw   100  ;wait 1000.4 us
   call    wait_10us
   decfsz  LOOP_MS, F
   bra     wait_ms_loop
   return
```

```
wait_100ms      movwf   LOOP_100MS ;store time in 100ms resolution
wait_100ms_loop movlw   100  ;wait approx 100ms
   call    wait_ms
   decfsz  LOOP_100MS, F
   bra     wait_100ms_loop
   return

init_hardware   bcf     LATC, 0 ;turn-off green LED
 bcf     LATC, 1  ;turn-off red LED before turning on output
 bcf     INTCON2, RBPU   ;enable pullups on port B for inputs
 movlw   7
 movwf   LATE
 movlw   84h
 movwf   ADCON1
 movlw     0fh
 movwf    TRISB
 bcf     TRISC, 0        ;make C.0 an output
 bcf     TRISC, 1        ;make C.1 an output
 clrf    TRISD           ;make D.0-7 outputs
 clrf    TRISE           ;make E.0-2 outputs
 return

wait_pb_press   btfsc   PORTB, 3  ;is pb pressed?
 bra     wait_pb_press   ;no, try again
 return
```

```
serial_init_hi  movwf   SPBRG  ;set serial baud rate
   movlw   64h             ;9-bit, tx on, hi-speed (BRGH) on
   movwf   TXSTA           ;set tx mode
   movlw   0D8h             ;9-bit, continuous rx. ADDEN = 1
   movwf   RCSTA           ;set rx mode
   return

serial_getc call serial_kbhit ;make sure data is ready to read
  andlw    1
  bz       serial_getc
  movf     RCREG, W        ;receive
  return

serial_kbhit   btfsc    PIR1, RCIF  ;has data been received?
  retlw    1               ;yes: return 1
  retlw    0               ;no:  return 0
;Initialize bidirectional serial
serial          macro   rate
  movlw   00001000b       ;select 16-bit baud rate gen divisor
  movwf   BAUDCON
  movlw   rate & 0xFF     ;set serial baud rate
  movwf   SPBRG
  movlw   (rate>>8)&0xFF
  movwf   SPBRGH
  movlw   00100100b       ;8-bit, tx on, hi-speed (BRGH) on
  movwf   TXSTA           ;set tx mode
```

```
  movlw   10010000b       ;8-bit, continuous rx
  movwf   RCSTA           ;set rx mode
  endm

serial_putc    btfss    PIR1, TXIF      ;is tx buffer empty?
  bra      serial_putc     ;no, try again
  movwf    TXREG           ;transmit
  return


serial_put_address  btfss    PIR1, TXIF      ;is tx buffer empty?
bra      serial_put_address ;no, try again
bsf      TXSTA, 0
movwf    TXREG           ;transmit
return

serial_put_data  btfss    PIR1, TXIF      ;is tx buffer empty?
  bra      serial_put_data    ;no, try again
  bcf      TXSTA, 0
  movwf    TXREG           ;transmit
  return

serial_const   tblrd*+                  ;read with post-increment
;for next character
  movf     TABLAT, W       ;read character value
  bz       sc_done         ;if null, end of the string done
```

```
  call     serial_putc     ;transmit
  bra      serial_const    ;get next char
sc_done        return

text_lcd_cmd   movwf    PORTD           ;write data (D.0-7)
  bcf      PORTE, 0        ;clear RS (E.0)
  bcf      PORTE, 2        ;clear R/W (E.2)
  movlw    5               ;50 us delay
  call     wait_10us
  bsf      PORTE, 1        ;set E (E.1)
  movlw    20              ;200 us delay
  call     wait_10us
  bcf      PORTE, 1        ;clear E (E.1)
  movlw    10              ;10 ms delay
  call     wait_ms         ;slowest command is >4ms
  return

text_lcd_data  movwf    PORTD           ;write data (D.0-7)
  bsf      PORTE, 0        ;set RS (E.0)
  bcf      PORTE, 2        ;clear R/W (E.2)
  movlw    5               ;50 us delay
  call     wait_10us
  bsf      PORTE, 1        ;set E (E.1)
  movlw    20              ;200 us delay
  call     wait_10us
  bcf      PORTE, 1        ;clear E (E.1)
```

```
  movlw    5               ;50 us delay
  call     wait_10us
  return

text_lcd_init  movlw  38h ;function set to 8bit data, dual line
call     text_lcd_cmd
movlw    38h             ;set two times
call     text_lcd_cmd
movlw    06h             ;entry mode: move to right, then down
call     text_lcd_cmd
movlw    0Ch             ;display on without cursor or blinking
call     text_lcd_cmd
movlw    01h             ;clear command
call     text_lcd_cmd
return

text_lcd_const tblrd*+ ;read with post-increment
movf     TABLAT, W       ;read value
xorlw    0               ;set flags
bz       tlp_done        ;if null, done
call     text_lcd_data   ;print character
bra      text_lcd_const  ;get next char
tlp_done       return
```

B.2 Java Code of ReadQueueconsumer Containing majority of the modules

```
/*
```

```
 * ReadQueueConsumer.java
 *
 * Reads from data from port and makes decision
 *
 */

package serailportreader;



import com.sun.corba.se.internal.ior.JavaCodebaseComponent;
import java.util.*;
import java.io.*;
import org.jfree.ui.RefineryUtilities;
import org.jfree.data.time.Millisecond;
import java.text.DecimalFormat;

public class ReadQueueConsumer extends Thread{

    private Vector printVector;
    private Vector accVector;
    private Vector angVector;
    private int orderCount;
    PrintWriter printwriter = null;

    /** Creates a new instance of ReadQueueConsumer */
    public ReadQueueConsumer() {
```

```
 printVector = new Vector();
 accVector = new Vector();
 angVector = new Vector();
 orderCount = 0;
 try
 {
 printwriter = new PrintWriter
 (new FileWriter("output.txt"));
 printwriter.println("xAcc;yAcc;zAcc;xSine;ySine;zSine");
 printwriter.close();
 }catch(Exception e) {
  e.printStackTrace();
 }
}

public void run()
{
 final XGraph xdemo = new XGraph("X Acceleration Graph");
 xdemo.pack();
 RefineryUtilities.centerFrameOnScreen(xdemo);
 xdemo.setVisible(true);
 final YGraph ydemo = new YGraph("Y Acceleration Graph");
 ydemo.pack();
 RefineryUtilities.centerFrameOnScreen(ydemo);
 ydemo.setVisible(true);
```

```
final ZGraph zdemo = new ZGraph("Z Acceleration Graph");
zdemo.pack();
RefineryUtilities.centerFrameOnScreen(zdemo);
zdemo.setVisible(true);

while(true) {
try {
 Thread.sleep(1);
 if(SerialReadByEvents.readQueue.size() > 0) {
 int readValue = ((Integer)SerialReadByEvents.
                   readQueue.remove(0)).intValue();
 if(readValue == 88) {
   System.out.println("Found first x");
   if(orderCount == 8) {
   System.out.println("Found an another" +
   "x with in count");
   processReadValues();
  }
 System.out.println(" Removing Noise, order Count" +
                                orderCount);
 orderCount = 0;
 System.out.println("Removing Elements");
 printVector.removeAllElements();
 printVector.add(new Integer(readValue));
 } else {
 orderCount++;
```

```
 printVector.add(new Integer(readValue));
 }
}
} catch(Exception e) {
e.printStackTrace();
}
   }
}

public void processReadValues() {
try {
int xVal binaryAdd(((Integer)printVector.elementAt(2))
.intValue(),
((Integer)printVector.elementAt(1)).intValue());
int yVal = binaryAdd(((Integer)printVector.elementAt(5))
.intValue(),
    ((Integer)printVector.elementAt(4)).intValue());
int zVal = binaryAdd(((Integer)printVector.elementAt(8))
.intValue(),
       ((Integer)printVector.elementAt(7)).intValue());
double xVolt = (double) xVal*2.978/1024;
double yVolt = (double) yVal*2.978/1024;
double zVolt = (double) zVal*2.978/1024;
double xAcc = (double) (1.489 - xVolt)/0.3;
double yAcc = (double) (1.489 - yVolt)/0.3;
double zAcc = (double) (1.489 - zVolt)/0.3;
```

```
double xAngle = getAngle(xAcc);
double yAngle = getAngle(yAcc);
double zAngle = getAngle(zAcc);

accVector.add(new Double(xAcc));
accVector.add(new Double(yAcc));
accVector.add(new Double(zAcc));
angVector.add(new Double(xAngle));
angVector.add(new Double(yAngle));
angVector.add(new Double(zAngle));

printwriter = new PrintWriter(new FileWriter
("output.txt", true));

if(angVector.size() == 9 && accVector.size() == 9)
 {
  double x1 = Math.ceil(((Double)accVector.elementAt(0))
  .doubleValue());
  double y1 = Math.ceil(((Double)accVector.elementAt(1))
  .doubleValue());
  double z1 = Math.ceil(((Double)accVector.elementAt(2))
  .doubleValue());
  double x2 = Math.ceil(((Double)accVector.elementAt(3))
  .doubleValue());
  double y2 = Math.ceil(((Double)accVector.elementAt(4))
  .doubleValue());
```

```
 double z2 = Math.ceil(((Double)accVector.elementAt(5))
 .doubleValue());
 double x3 = Math.ceil(((Double)accVector.elementAt(6))
 .doubleValue());
 double y3 = Math.ceil(((Double)accVector.elementAt(7))
 .doubleValue());
 double z3 = Math.ceil(((Double)accVector.elementAt(8))
 .doubleValue());
 //printwriter.println("3 acc" + x1 + " " + y1 + " "
 + z1 + " " +
 x2 + " " + y2 + " " + z2 + " " + x3 + " " + y3
 + " " + z3);
double xA1 = Math.ceil(((Double)angVector.elementAt(0))
.doubleValue());
double yA1 = Math.ceil(((Double)angVector.elementAt(1))
.doubleValue());
double zA1 = Math.ceil(((Double)angVector.elementAt(2))
.doubleValue());
double xA2 = Math.ceil(((Double)angVector.elementAt(3))
.doubleValue());
double yA2 = Math.ceil(((Double)angVector.elementAt(4))
.doubleValue());
double zA2 = Math.ceil(((Double)angVector.elementAt(5))
.doubleValue());
double xA3 = Math.ceil(((Double)angVector.elementAt(6))
.doubleValue());
```

```
double yA3 = Math.ceil(((Double)angVector.elementAt(7))
.doubleValue());
double zA3 = Math.ceil(((Double)angVector.elementAt(8))
.doubleValue());
//printwriter.println("3 angles" + xA1 +" " + yA1
+ " " + zA1 + " " +
xA2 + " " + yA2 + " " + zA2 + " " + xA3 +" "
+ yA3 + " " + zA3);

java.sql.Time tm = new java.sql.Time(System.
currentTimeMillis());
int hours = tm.getHours() + 1;
int mins = tm.getMinutes();
int secs = tm.getSeconds();

if((xA1 == xA2) && (xA2== xA3) && (xA1 == xA3)
&& (yA1 == yA2) && (yA2== yA3) && (yA1 == yA3)
&&(zA1 == zA2) &&
(zA2== zA3) && (zA1 == zA3))
{
if(zA1 > 65 )
printwriter.println("Time:" + hours + ":" + mins
+ ":" + secs + " sit/lie");
if(yA1 > 55)
printwriter.println("Time:" +  hours + ":" + mins +
":" + secs + " stand");
```

```
 if(zA1 < -55 )
 printwriter.println("Time:" +  hours + ":" + mins +
 ":" + secs + " reverse lie");
 }
 else
 {
 if((yA1 > 55) && (yA2 > 55) && (yA3 > 55) &&
 ((xA1 != xA2) && (xA2 != xA3) || (zA1 != zA2)
 && (zA2 != zA3)))
 printwriter.println("Time:" +  hours + ":" + mins
 + ":" + secs + " walk");
 if(((xA3 > xA2)&&(xA2 > xA1)) || ((yA3 > yA2)
 && (yA2 > yA1))
 || ((zA3 > zA2) && (zA2 > zA1)))
 printwriter.println("Time:" +  hours + ":" + mins + ":"
 + secs + " drop");
 if((xA1 != xA2)&&(xA2 != xA3) && (yA1 != yA2)
 && (yA2 != yA3)
 && (zA1 != zA2) && (zA2 != zA3))
 printwriter.println("Time:" +  hours + ":" + mins + ":"
 + secs + " transition");
}
angVector.remove(0);
angVector.remove(0);
angVector.remove(0);
accVector.remove(0);
```

```
  accVector.remove(0);
  accVector.remove(0);
}
XGraph.xseries.add(new Millisecond(), xAcc);
//y-voltage
YGraph.yseries.add(new Millisecond(), yAcc);
//z-voltage
ZGraph.zseries.add(new Millisecond(), zAcc);
printwriter.close();
     } catch(Exception e) {
         e.printStackTrace();
     }
 }

 public double getAngle(double accVal) {
     double sineVal, tempVal;
     DecimalFormat df = new DecimalFormat("##.##");
     if (accVal <0 ) {
         if(accVal < -1.0)
             accVal = -1.0;
         accVal = -accVal;
         tempVal = java.lang.Math.asin(accVal);
         tempVal = -tempVal;
         sineVal =(tempVal*180)/3.14159265;
     } else {
         if(accVal > 1.0)
```

```
            accVal = 1;
        tempVal = java.lang.Math.asin(accVal);
        sineVal =(tempVal*180)/3.14159265;
    }
    return Double.parseDouble(df.format(sineVal));
}

public int binaryAdd(int higherBit, int lowerBit)
{
    String lowerString = dec2binary(lowerBit,8);
    String higherString = dec2binary(higherBit,8);
    //System.out.println("lowerByte :"  + lowerBit
    + " " + lowerString);
    //System.out.println("higherByte :" + higherBit
    + " " +higherString);
    String totalString = higherString + lowerString;
    //System.out.println("TotalByte :" +totalString
    + " TenBits: "
    + totalString.substring(0,10));
    int totalBits[] = convertStringBits(totalString.
    substring(0,10));
    int returnDec = bin2Dec(totalBits);
    return returnDec;
}

public int bin2Dec(int[] binaryNum) {
```

```
    int returnDec=0;

    for(int place=0; place < 10; place++) {
        returnDec += java.lang.Math.pow(2,place)*
        binaryNum[place];
    }
    return returnDec;
}

public int[] convertStringBits(String bits) {
    int returnBits[] = new int[10];

    StringBuffer revBits = new StringBuffer(bits);
    revBits.reverse();

    for(int i=0; i<returnBits.length;i++) {
        returnBits[i] = Integer.parseInt(revBits.
        ubstring(i,i+1));
    }
    return returnBits;
}

public String dec2binary(int val, int numbits){
    int mask = 1;
    String Message = "";
```

```
        //left-shift a 1 into proper position
        mask <<= numbits-1;

        int spacer = 0;
        for(;mask!= 0; mask>>>=1){
            if((val & mask) !=0) Message = Message + "1";
            else
                Message = Message + "0";
        }
        return(Message);
    }
}
```

APPENDIX C

PROGRAM OUTPUT

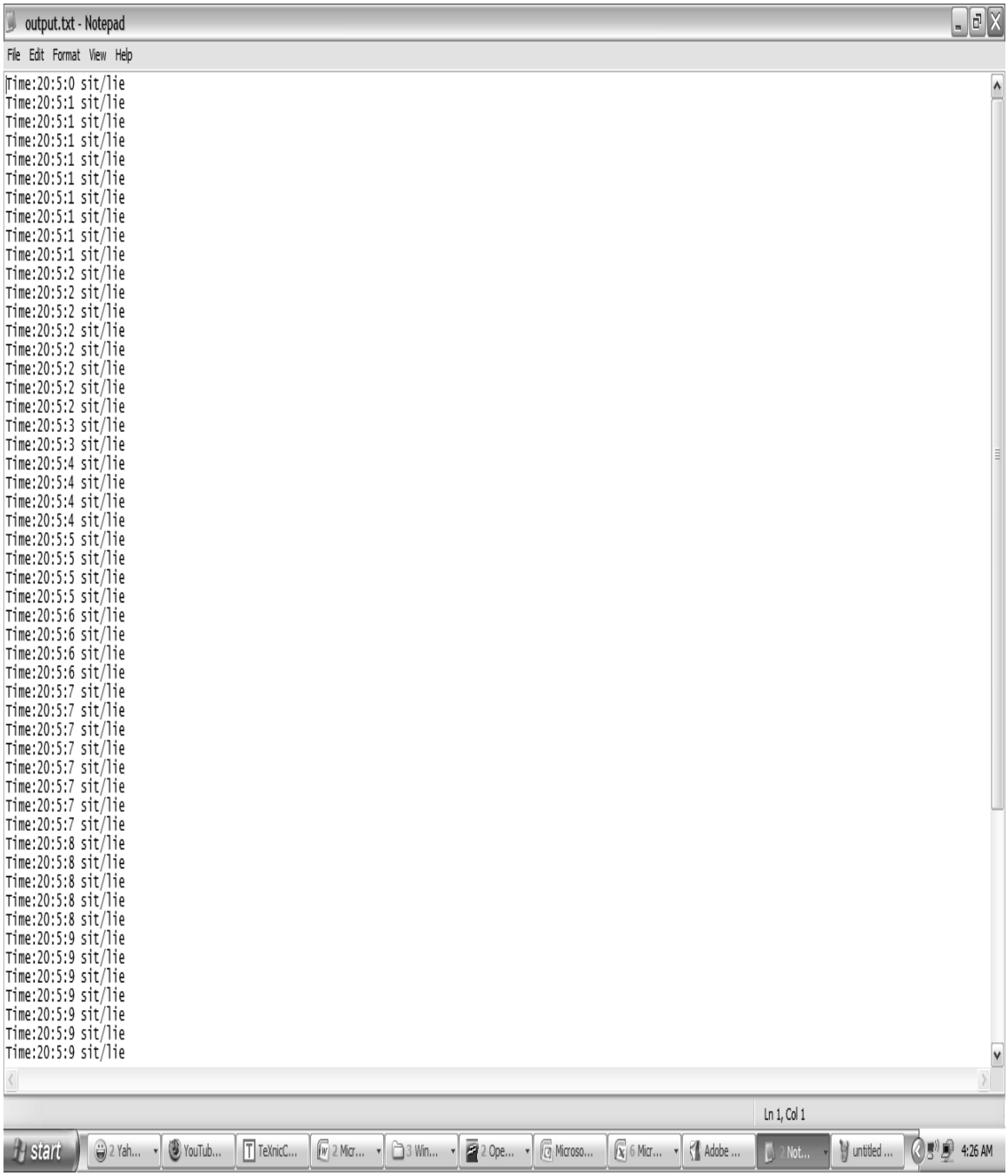

Figure C.1 Example snapshot of Output.txt file for Sit/Lie state

REFERENCES

[1] J. F. DSouza, "Investigation of human activity state detection and monitoring," Master's thesis, The University of Texas at Arlington, 2003.

[2] G. Ganapathy and G. V. Kondraske, *Microprocessor-based instrumentation for ambulatory behavior monitoring.* Lippincot, 1990.

[3] P. Peixoto, J. Batista, and H. Arajo, "Real-time human activity monitoring exploring multiple vision sensors." [Online]. Available: citeseer.ist.psu.edu/393541.html

[4] M. Leo, T. D'Orazio, I. Gnoni, P. Spagnolo, and A. Distante, "Complex human activity recognition for monitoring wide outdoor environments," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4.* Washington, DC, USA: IEEE Computer Society, 2004, pp. 913–916.

[5] B. Ozer, W. Wolf, and A. N. Akansu, "Human activity detection in mpeg sequences," in *HUMO '00: Proceedings of the Workshop on Human Motion (HUMO'00).* Washington, DC, USA: IEEE Computer Society, 2000, p. 61.

[6] Electromyography, "http://en.wikipedia.org/wiki/electromyography."

[7] Polysomnography, "http://en.wikipedia.org/wiki/polysomnography."

[8] Aware, "http://www.awaretechs.com/wirelessaccelerometer.html."

[9] M. U. Rue Shao Dong, Motohiro Tanaka and T. Ishimatsu, "Intelligent monitoring system of bedridden elderly," in *Proceedings of SPIE*, 2005, pp. 604 020–4.

[10] B. Sweetman, "A measurement of limb activity in a back pain study in industry," in *Proceedings of the Third International Symposium on Ambulatory Monitoring*, London, 1979, pp. 231–237.

[11] R. Ramsay, "Clinical usefullness of ambulatory eeg monitoring of the neurological patient," in *Proceedings of the Fourth International Symposium on Ambulatory Monitoring*, Belgium, 1981, pp. 234–243.

[12] e. K.W Eatanable, "Estimation of sleep stage based on heart rate fluctuation and body movement," in *Proceedings of SICE Annual conference*, Sapporo, 2004, pp. 2153–2156.

[13] e. H.Andoh, "Network health monitoring system in the sleep," in *Proceedings of SICE Annual conference*, Sapporo, 2004, pp. 1421–1424.

[14] A. Schmidt, H. W. Gellersen, and M. Beigl, "A wearable context-awareness component," in *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 1999, p. 176.

[15] G.Gopinath, "Human activity state detection and monitoring system," Master's thesis, The University of Texas at Arlington, 1988.

[16] T. Burchfield and S.Venkatesan, "Accelromter based human abnormal movement detection in wireless sensor networks," Master's thesis, The University of Texas at Dallas, 2005.

[17] W. P. Jonghun Baek, Geehyuk Lee and B.-J. Yun, *Accelerometer Signal Processing for User Activity Detection*. Springer Berlin / Heidelberg, 2004.

[18] G. F. Dreher and P. R. Sackett, *Situation specificity of behavior assessment center validation strategies: A rejoinder to Neidig and Neidig*, 1984, pp. 187–190.

[19] S. Epstein and P. Meier, *Constructive thinking: A broad coping variable with specific components*, 1989, pp. 332–350.

[20] L. J. Cronbach, *Coefficient alpha and the internal structure of tests*, 1951, pp. 297–334.

[21] D. han and Schmitt, *Video-based versus paper-and-pencil method of assessment in Chapter 1 Subheading 5: Subgroup differences in test performance and face validity perceptions*, 1977, pp. 143–149.

[22] N. R. Cliff, *The eigenvalues-greater-than-one-rule and the reliability of components*, 1988, pp. 276–279.

[23] R. B. Cattell, *The screen test for number of factors*, 1966, pp. 245–276.

[24] S. R. Briggs and S. R. Cheek, *The role of factor analysis in the development and evaluation of personality scales*, 1986, pp. 663–678.

[25] W. C. Borman, M. A. Hanson, S. H. Oppler, E. D. Pulakos, and L. A. White, *Role of supervisory experience in supervisor performance*, 1993, pp. 443–449.

[26] W. C. Borman, L. A. White, E. D. Pulakos, and S. H. Oppler, *Models of supervisory performance*, 1991, pp. 863–872.

[27] L. E. Atwater, *Beyond cognitive ability: Improving the prediction of performance*, 1992, pp. 27–44.

[28] L. Atwater and F. Yammarino, *Personal attributes as predictors of superiors and subordinates perceptions of military academy leadership*, 1993, pp. 645–668.

[29] K. Bollen and R. Lennox, *Conventional wisdom on measurement: A structural equation perspective*, 1991, pp. 305–314.

[30] W. F. Cascio and H. Aguinis, *Applied Psychology In Personnel Management.* Prentice Hall Publishing, 1970.

[31] E. H. Cattell, R. B. and M. Tatsuoka, *Handbook for the Sixteen Personality Factor Questionnaire.* Institute for Personality and Ability Testing, 2005.

BIOGRAPHICAL STATEMENT

Himabindu Yamana was born in Chirala, Andhra Pradesh,India in 1984. She received her B.E. degree from St.Ann's College of Engineering and Technology, Chirala, India in 2005 and her M.S. degree from The University of Texas at Arlington in 2007, both in Electronics Engineering. During her period of study she worked as an intern in Traulsen, Fort-worth, Texas. She also worked as Teaching Assistant for the EE Department, The University of Texas at Arlington. Her research interests include VLSI designing and Embedded devices.