EE318 Electronic Design Lab Project Report, EE Dept, IIT Bombay, April 2013

Fall Detection and Boundary Monitoring for Young and Old

Group No: 5
Shibani Santurkar (100010008) <shibani@iitb.ac.in>
Sahil Agarwal (10D070017) <sahil_agarwal@iitb.ac.in>
Supervisors: Dinesh Sharma, Dipankar

Abstract

The main aim of our electronics based project is to detect human falls. This we achieved by taking input from a three-axis accelerometer. After some computing to ascertain a fall, an “emergency” SMS is sent to a pre-stored mobile number through a GSM module. The SMS also contains the latitude and longitude of the wearer obtained by a GPS module. This entire kit is worn like a belt around the waist.

We had another major feature in our project. It consists of the user inputting the points of a boundary of his choice. If the wearer moves outside this boundary another “emergency” SMS will be sent. In terms of hardware this feature additionally requires only a user interface.

1. Introduction

Falling is a serious health issue among the elderly population resulting several times in critical injuries like hip fractures. With elderly who live alone, not being found for hours after a fall is quite common and can have drastic consequences. We undertook this project to give a feasible and inexpensive solution in the form of an electronics gadget.

There has already been research in fall detection algorithms but many research papers and projects involve the use of gyroscopes. To cut down on costs (gyroscopes are expensive) we developed our own fall detection algorithm using a three-axis analog accelerometer.

Boundary Monitoring is also essential for both the very young as well as the old. Children sometimes go far away from a safe place and parents have no way to know this until when it is too late. Even the elderly need to be monitored similarly.

2. High level Description and Block Diagram

Our objective was to make a gizmo that would act as a safety device for both children and the elderly. It would have two essential features, to detect and report their falls to a care-giver and to ensure that they remain in some sort of safe zone. To realize these objectives we needed some mechanism to sense falls for which we decided to use a single accelerometers unlike more conventional prototypes which use several accelerometers and gyroscopes. We also needed some mechanism to report the situation to the guardian along with the position of the wearer. We decided to use SMS as a mode of communication for which we picked the SIM – 300 module. To locate the wearer we decided to use GPS data. Even though this has slightly poor reception indoors, our objective was primarily to cater to outdoor applications since several Bluetooth based gadgets exist for home based monitoring. We wanted to provide a user- interface wherein the user could easily enter the perimeter and other details. We hoped to add a boost convertor and a battery monitoring system ass the power system of the gadget. Ultimately we hoped to provide something that could be worn around the waist and would be economical.

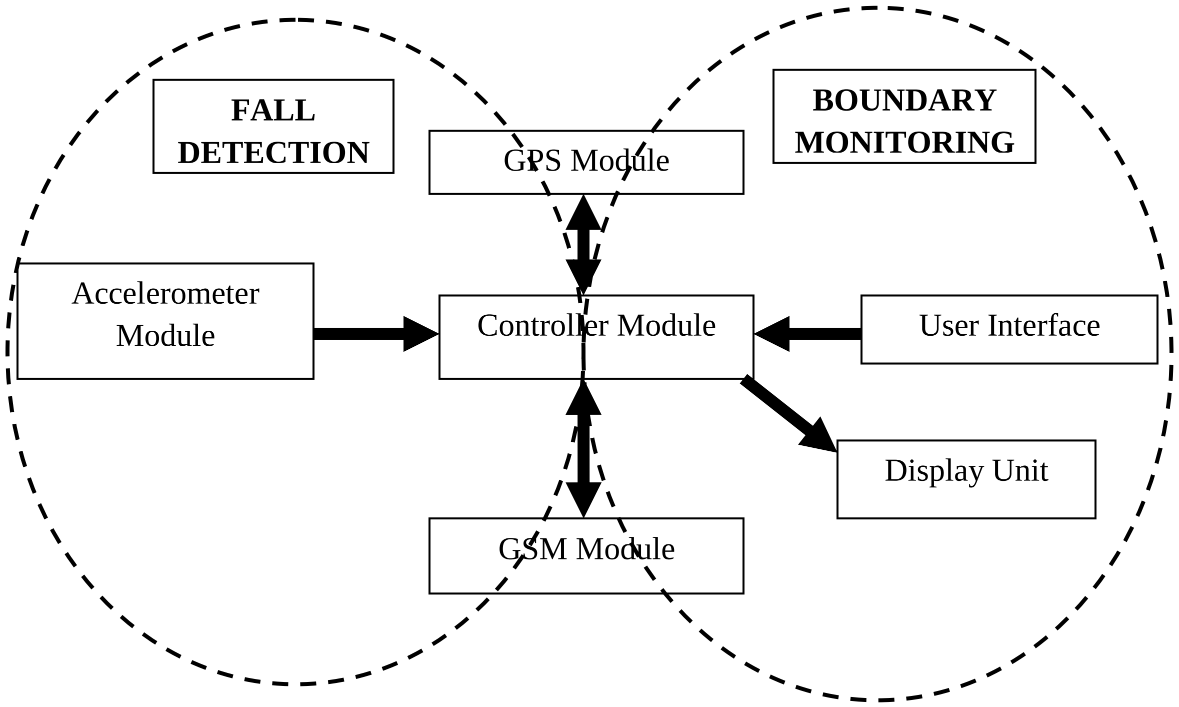

Figure 1: Block Diagram of system

3. Hardware Design/Interfacing

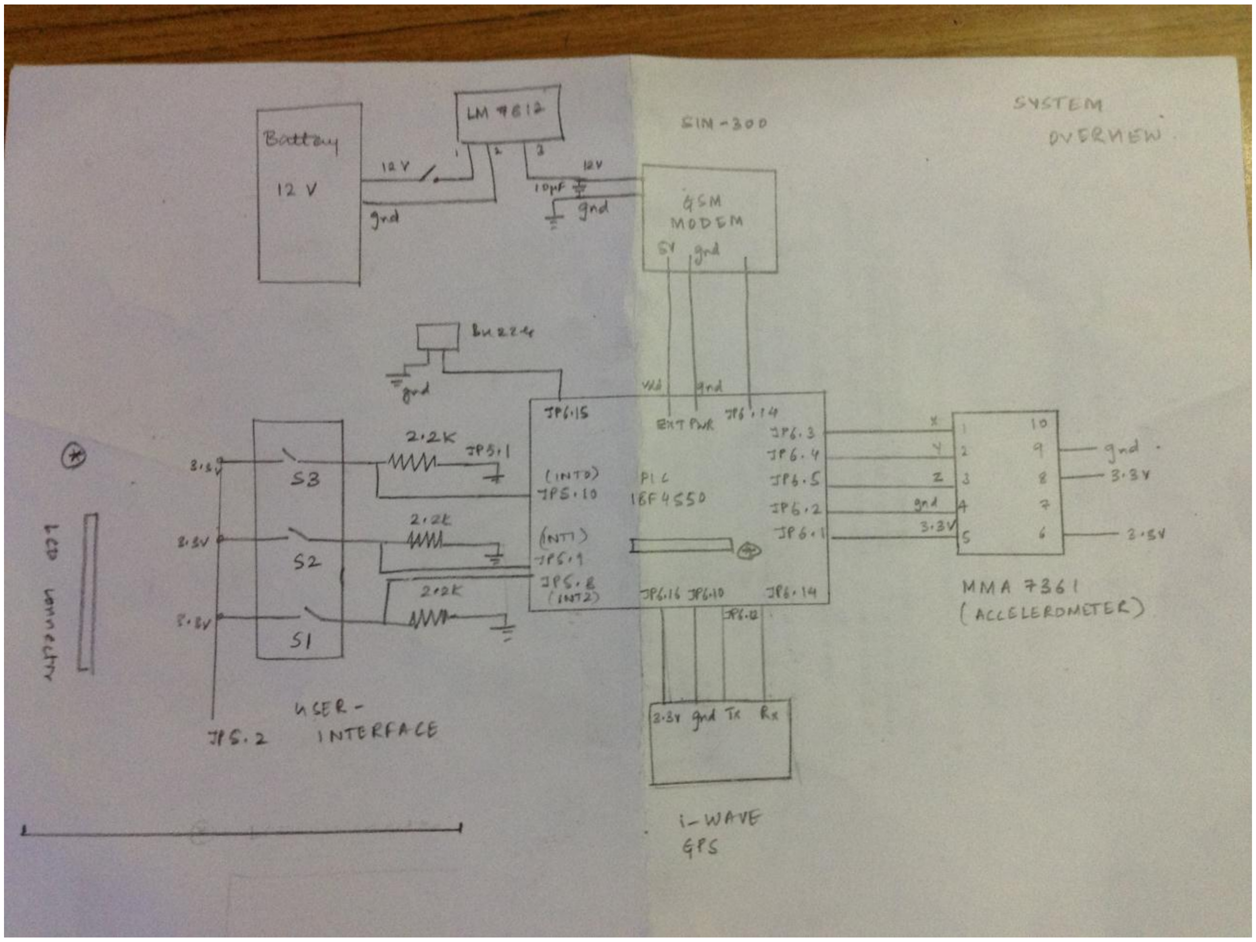

Figure 2: System Overview
(This diagram shows the interconnections of our various modules)

Controller Module

We used the Aurum v1.2 (a PIC18F4550-based board).

- The accelerometer module used the onboard ADC.
- The serial interface was used by the GPS and GSM modules.
- External interrupts were used by the User Interface.

Accelerometer module

This module used the MMA7361L ±1.5g, ±6g Three axis Accelerometer Module (Two Line) from Nex Robotics.

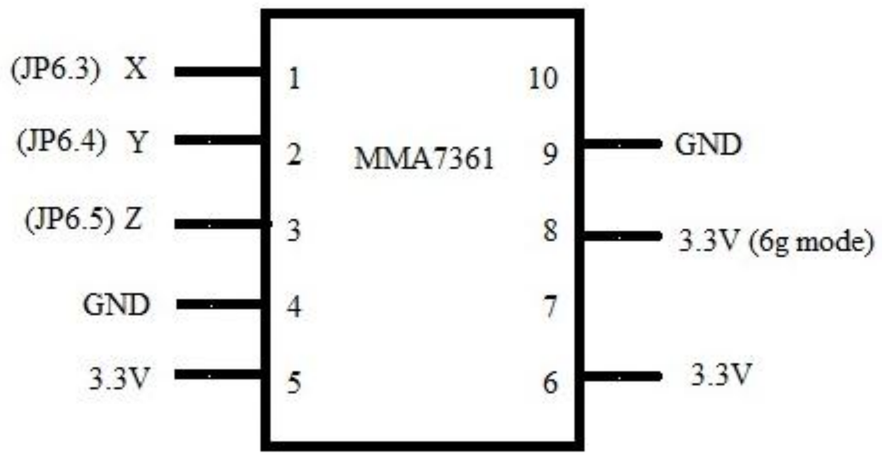

Figure 3: MMA7361 Accelerometer Interfacing
(This diagram shows which pins of the MMA7361 are connected to specific pins of the micro-controller)

GPS module

We used iWave GPS Module User Manual (iW-PRDLT-UM-01-R2.0).

It uses NMEA protocol. We used this module because of its reliability, speed and operating voltage. It has a very fast start up time of around 45 seconds and has an operating voltage of 3.3V which is easily obtainable from the Aurum board.

The format we were interested in was GGA which looks as follows:
$GPGGA,002153.000,3342.6618,N,11751.3858,W,1,10,1.2,27.0,M,-34.2,M,,0000*5E

Table 1: GGA Format

Name	Example	Unit	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	002153.000		hhmmss.sss
Latitude	3342.6618		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	11751.3858		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	10		Range 0 to 12
HDOP	1.2		Horizontal Dilution of Precision
MSL Altitude	27.0	meters	
Units	M	meters	
Geoid Separation	-34.2	meters	Geoid-to-ellipsoid separation. Ellipsoid altitude = MSL Altitude + Geoid Separation.
Units	M	meters	
Age of Diff. Corr.		sec	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*5E		
<CR> <LF>			End of message termination

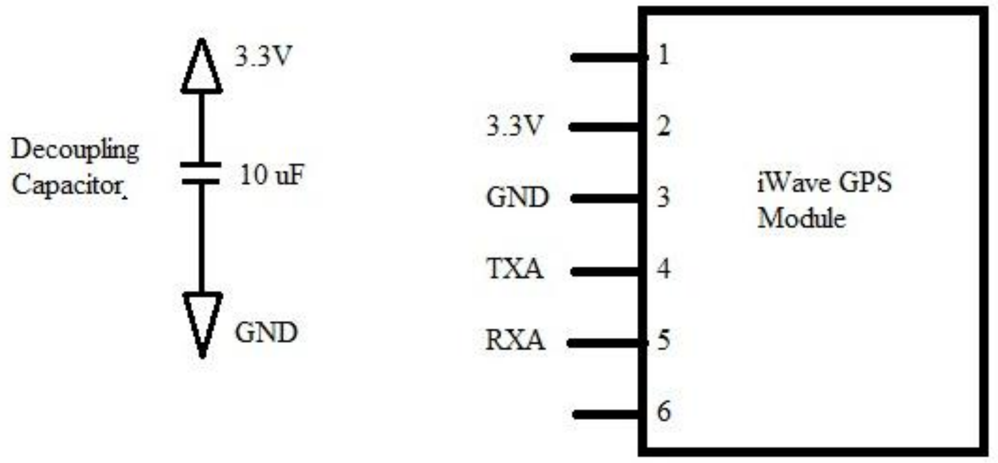

Figure 4: Interfacing of GPS Module
(This figure shows which a representative diagram of the GPS module)

GSM Module

We used the module available in WEL. This module uses SIM300 IC.

Testing of this module was carried out through the HyperTerminal. The module was serially connected to the PC via MAX-232 circuit.

The GSM modem is operated using the serial interface of the microcontroller. It communicates at a baud rate of 9600 bps and it understands only AT commands.

Basic syntax:

These AT commands have the format of “AT<x><n>”, or “AT&<x><n>”, where “<x>”is the command, and “<n>”is/are the argument(s) for that command. An example of this is “ATE<n>”, which tells the DCE whether received characters should be echoed back to the DTE according to the value of “<n>”. “<n>” is optional and a default will be used if missing.

In order to send a SMS we need to send the following three main commands:

```
at+cmgf=1                         //configuring modem in text mode
at+cmgs="8082270432”              //pre-stored number
"EMERGENCY"                       //content of the message
```

The communication between uC and modem is **full duplex** with the transmit of the uC connected to receive of the modem and vice versa.

Testing Module

A major portion of our project was interfacing with different modules via the serial interface. For this we used the MAX 232 IC so as to be able to observe the serial data on the computer. This IC is essentially a level convertor. The computer serial port requires inputs to lie between plus and minus 12V whereas uC outputs tend to be between 0V and 5V. So the MAX 232 performs this level conversion.

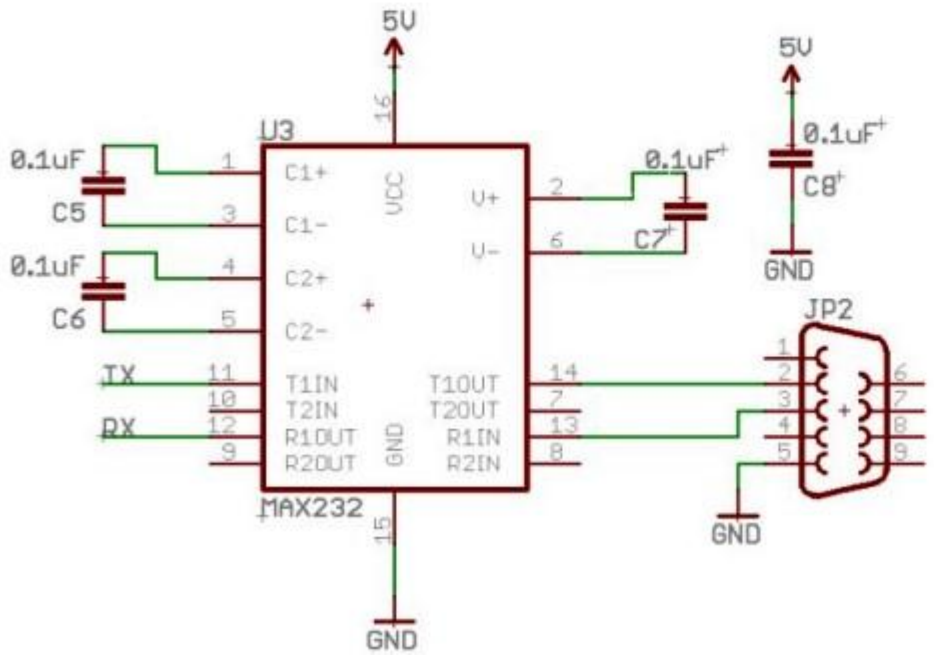

Figure 5: Circuit Diagram for MAX-232 Circuit

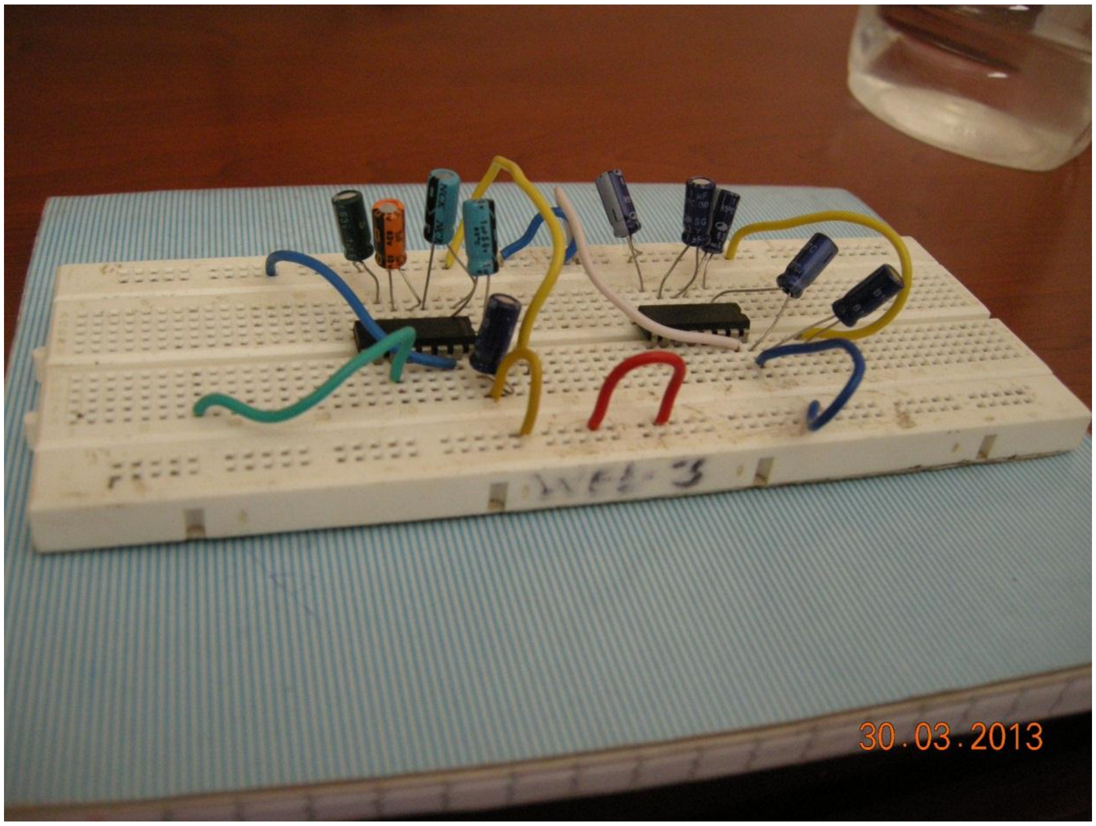

Figure 6: Circuit diagram of MAX-232 interface for serial communication

User Interface Module:

- Our user interface comprises of three push buttons, an LCD screen and a buzzer.
- The main push-button enables the user to scroll through the menu, giving the user the choice to enter his own mobile number, and the perimeter points.
- The LCD screen is used as a display unit to show relevant information.
- If a fall is detected the buzzer goes off. If the user has recovered he can press any of the push buttons which will prevent the sending of the emergency message.

Battery:

- We are using a 12V, 1500 mA Ni-mh battery.

4. Software Design

Quite a large part of the microcontroller code is devoted to interfacing with the different hardware modules. Other than that the two main algorithms written have been described below.

Fall Detection Algorithm

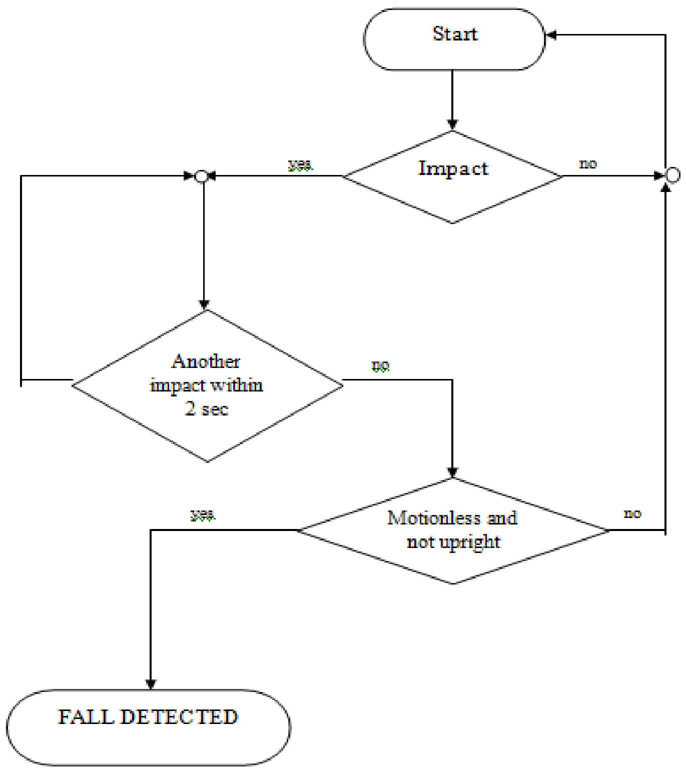

Figure 7: Flowchart of the Fall Detection Algorithm

Instead of storing data, we employed the use of state diagrams based on this flowchart.

Boundary Monitoring Algorithm

Wrote an algorithm to detect whether a point (defined by its X and Y coordinates or equivalently its latitude and longitude) is inside or outside a user defined perimeter using the ray casting algorithm.

What is the algorithm?
Take any point **inside** a polygon (concave or convex). If it is a source of a ray (in any direction) the ray will cut the polygon in **ODD** number of places.

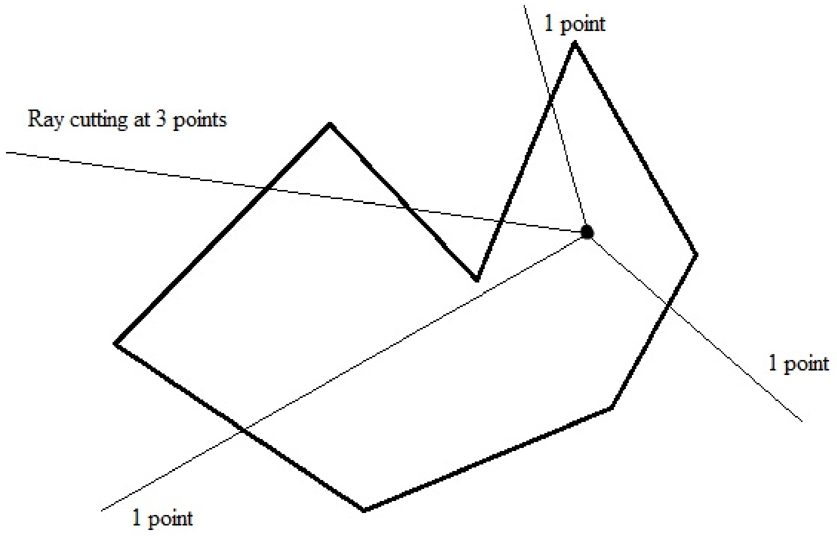

Figure 8: Point inside the polygon

Conversely, take any point **outside** any polygon. If it is a source of a ray (in any direction) the ray will cut the polygon in **EVEN** number of places.

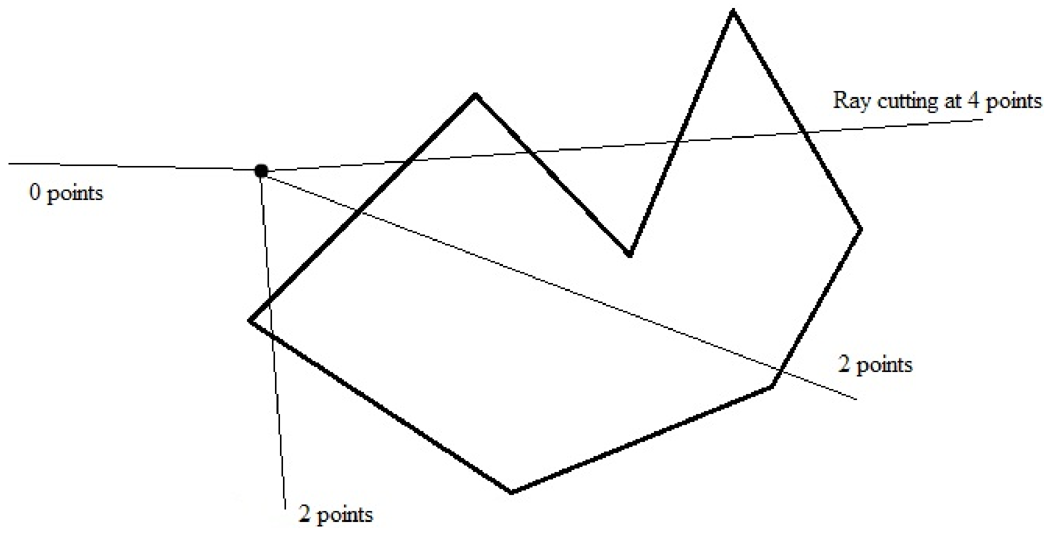

Figure 9: Point outside the polygon

Our code:
We sent a ray in a direction of 0 degrees with the X axis (latitude axis) and count the number of times it cut the polygon. If the count is odd the wearer is safely inside the boundary; else he/she is outside the boundary.

5. Results/Objectives achieved

- Interfacing of the accelerometer and developing a fall detection algorithm.
- Developing an algorithm for perimeter monitoring and using real time GPS information as an input to this algorithm.
- Interfacing the GSM modem and sending text messages.
- Interfacing the GPS and the setup for the perimeter monitoring module, i.e., we are able to record a set of points whenever the user presses a push button.
- We have setup the required user interface.

6. Problems faced

- We initially started with a digital accelerometer but since we were finding it difficult to interface using I2C protocol, we switched to the analog one.
- Debugging the MAX232 circuit for serial communication with the PC took a long time. Setting up the USART also was time consuming due to problems in baud rate and synchronisation (for interfacing of uC, PC and GSM module).
- We faced some major timing issues with the GPS module in changing its baud rate, configuring the module to get only relevant data, storing all the data in a buffer but managed to get it working after a lot rigorous effort.
- The SIM300 chip is not working without the entire GSM module. This module needs 12V supply which is forcing us to design a Boost converter (since our battery is 3.7V – standard cell phone battery).

7. Future Work:

- A good idea would be to develop a battery monitoring system and alert the user when the battery needs to be charged.
- Further miniaturization of the prototype is required.

8. Completed Product :

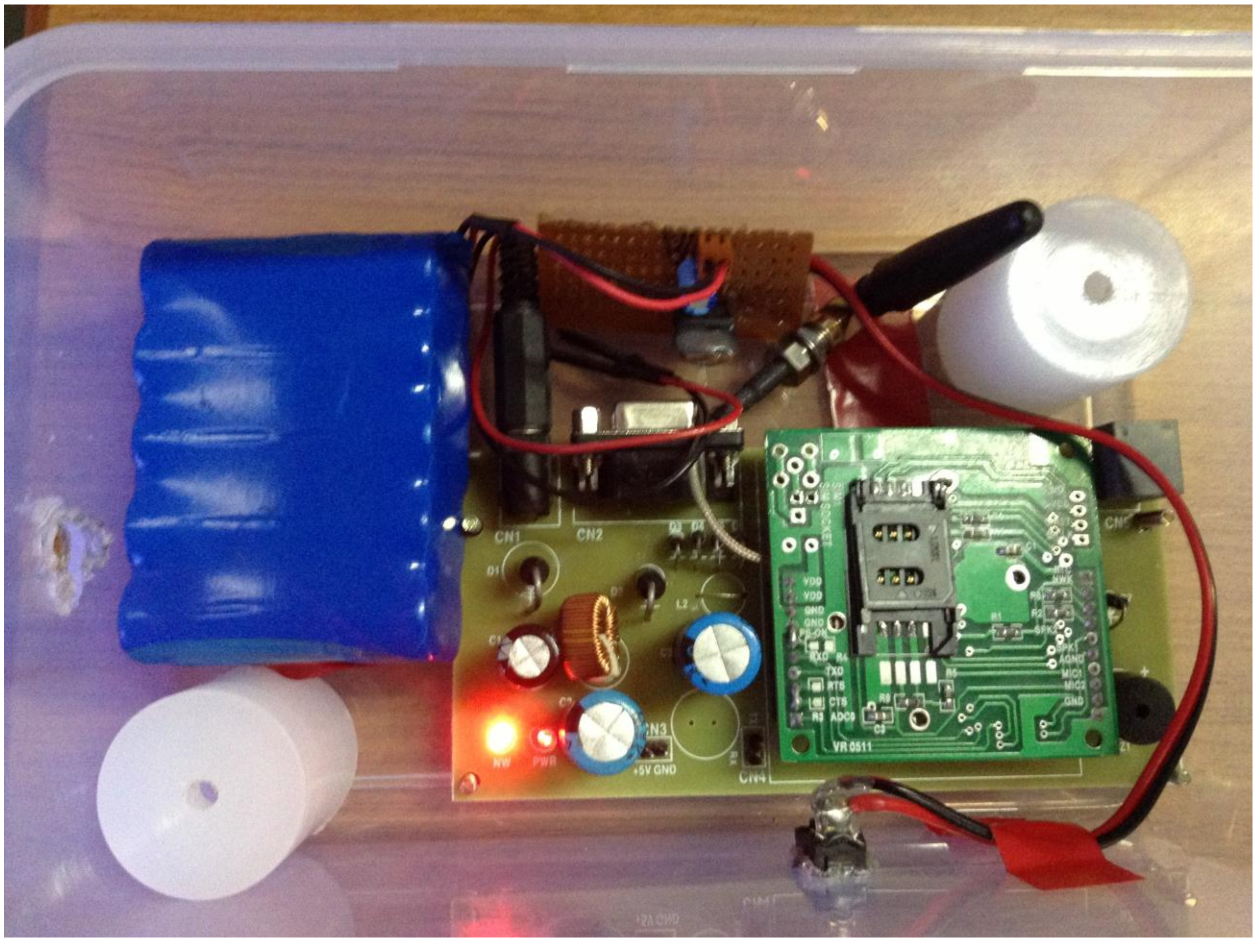

Figure 10: Bottom layer of our prototype with battery, GSM modem

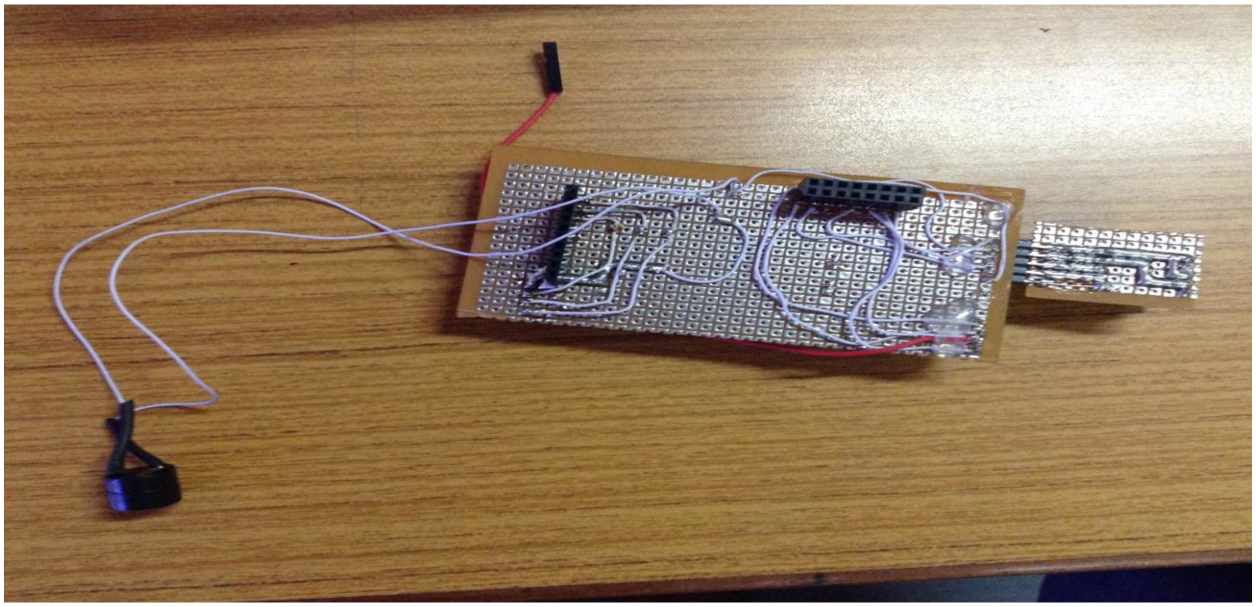

Figure 11: Bottom view of our Break-out board

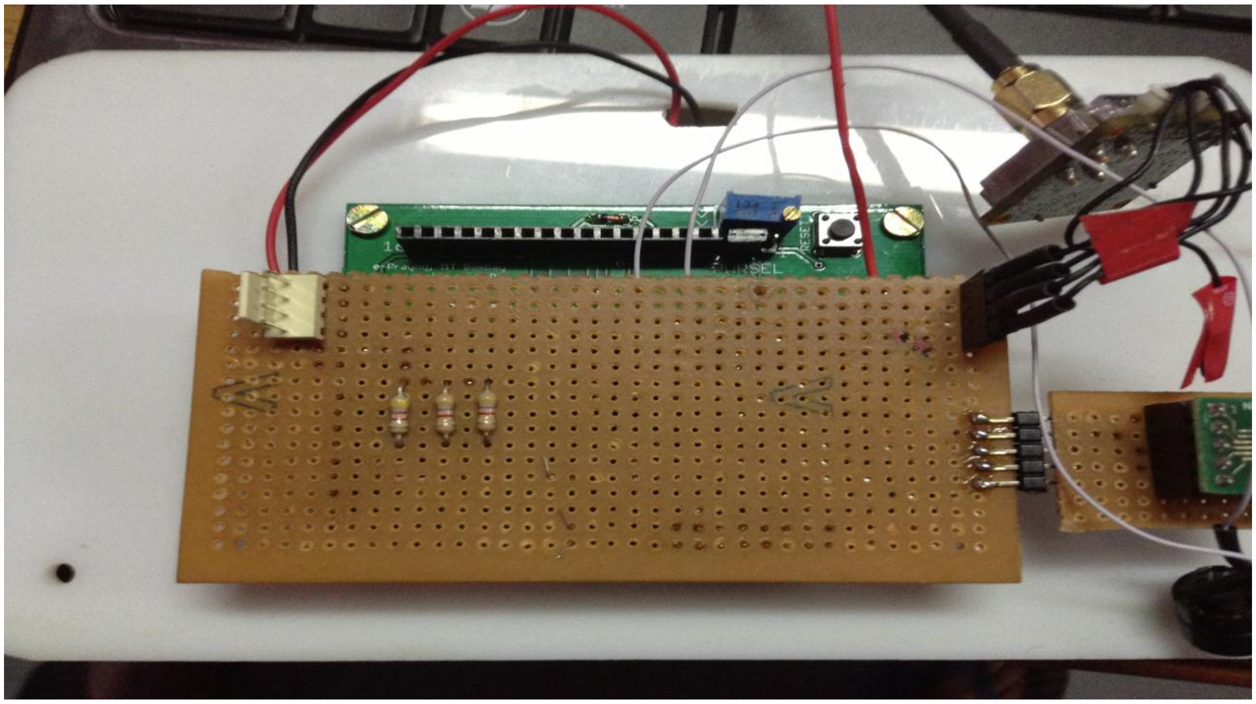

Figure 12: Top view of our Break-out board mounted on the micro-controller

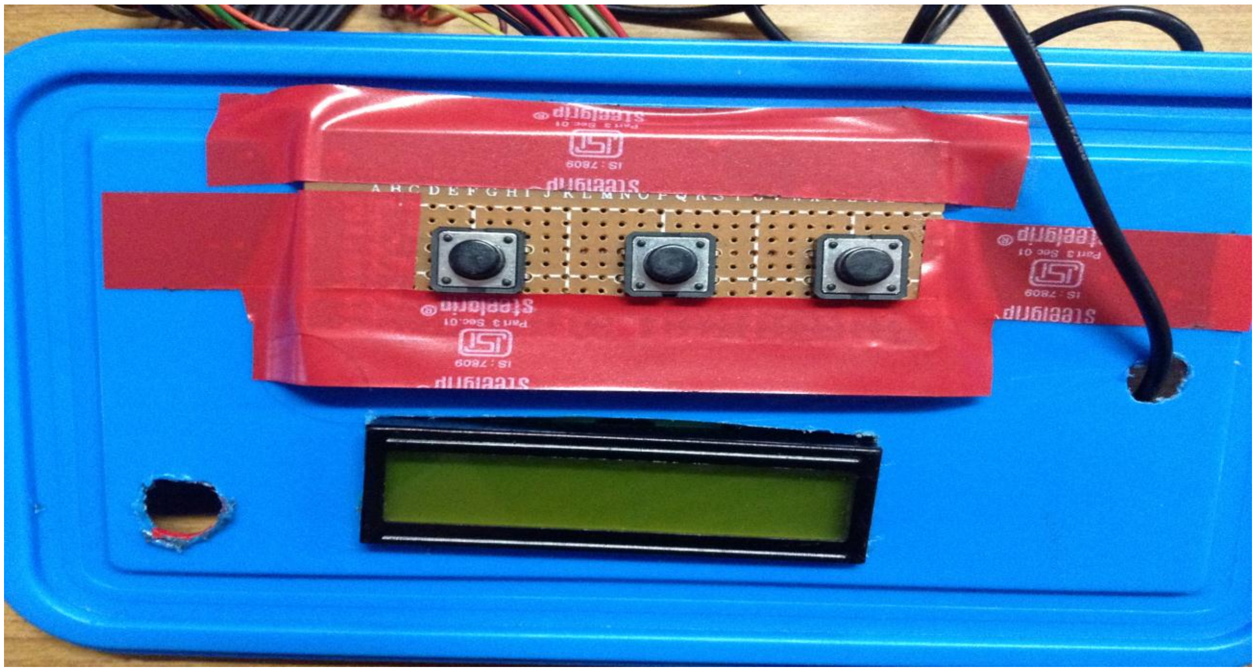

Figure 13: User interface of prototype

References

1. Accelerometer datasheet: http://www.nex-robotics.com/images/downloads/MMA7361L.pdf
2. Accelerometer manual: http://www.nex-robotics.com/images/downloads/MMA7361L%20%C2%B11.5g,%20%C2%B16g%20Three%20axis%20Accelerometer%20Module%20(Two%20Line).pdf
3. AT Command Set for GSM modem http://www.owen.ru/uploads/re_pm01_list_command.pdf
4. Library for compiler used for uC http://ww1.microchip.com/downloads/en/devicedoc/mplab_c18_libraries_51297f.pdf
5. http://www.ece.tufts.edu/ee/194HHW/papers/01617246.pdf
6. http://digitalcommons.mcmaster.ca/cgi/viewcontent.cgi?article=1044&context=ee4bi6