Android Based Monitoring Human Knee Joint Movement Using Wearable Computing

Sanjana K.S [#1], Vidit Jain [#2], Vignesh P [#3], Vishal H [#4]

Department of Computer Science and Engineering,

M.S. Ramaiah Institute of Technology, Bangalore 560054, India

[1]sanjanasukumar2012@gmail.com
[2]viditjain27@gmail.com
[3]vigneshp8080@gmail.com
[4]vishalh08@gmail.com

Abstract: ***In today's fast moving lifestyle, incidents regarding health issues are surfacing every day. One of the major issues relating to medical concern is weakness in joints caused due to excess stress. Hence there is a demand in the market for wearable devices to measure the rapid movement of human joints when under recovery. We are developing a prototype by implementing two ADXL335 accelerometer sensors which is placed around the joints in order to detect the amount of stress in the knee, thereby providing the correct information to an individual's family physician. This system will be highly helpful for athletes and also for people who are recovering from a knee surgery as it ensures speedy rehabilitation since it constantly monitors the knee. It uses a Renesas RL78/G12 microcontroller along with a HC05 Bluetooth module to communicate with the Android application where the data obtained from the sensors are graphically represented using a pie chart which displays the frequencies of different activities performed by the patient such as walking, running, etc. Whenever there is excess stress on the patient's knee due to undesired body posture or activity performed, a voice alert is played on the android application to ensure that the patient refrains from doing that activity or changes the body posture. Also during unforeseen situations, a SMS alert is sent to the registered mobile number and an Email alert along with the location of the patient, using GPS, is sent to the intended recipients. In order to make sure that the Android application occupies reasonable space, the accumulated data can be cleared once the physician has seen it.***

Keywords: ADXL335, Renesas RL78/G12, HC05 Bluetooth module, Android application, SMS, GPS.

1. INTRODUCTION

If an individual, say an athlete, is injured in the knee utmost care has to be taken in-order to recover as soon as possible. For this purpose, the amount of stress on the knee due to an athlete's day-to-day activity needs to be monitored at every instant of time. This is shown in Figure 1. If the athlete bends his injured knee too much accidently, then the chances of damage to the injured knee would increase and the recovery process would take more time.

Figure 1 Monitoring Human Knee Joint

To facilitate proper correction for speedy recovery from damage caused at the joints, in this case the knee joint, we propose a prototype which uses accelerometer sensors placed on both the side of the knee, one above the joint and the other below the joint. The readings are obtained by monitoring the changes in the positions of the two accelerometers. Any movement that occurs at the knee joint where the accelerometers are placed will cause these sensors to generate an analog output voltage corresponding to the change in gravity. These voltages are fed as input to the microcontroller. The microcontroller further processes this information and delivers a digital output which is transmitted via Bluetooth to an Android based application. The Android application is responsible for receiving and storing this data and to display it graphically so that it is visually better and is faster to analyze than textual information. This data will help the physician to monitor the athlete's knee during rehabilitation. Whenever the bend around the joints is more, the athlete is alerted and hence can make the necessary adjustments.

2. LITERATURE SURVEY

The field of Body Sensor Network (BSN) is very useful in constantly monitoring the body's movements in rehabilitation activities [1]. In our case we consider that the patient wants to recover quickly after an arthritis operation and patient along with the doctor needs to constantly monitor the different activities performed by the patient to ensure that there is not much of stress or strain applied on the knee due to excessive activity. Thus this data is very crucial for physiotherapists who ensure that the effectiveness of the rehab program is quite high and patient achieves a timely recovery [2]. As our concentration is on monitoring the movements in the knee joint we have used a 3-axes accelerometer which gives more precise readings [1].

The objective of [1] is fall detection and it uses a 2-axes accelerometer sensor with an ATmega8 microcontroller and thus the readings of the sensors need not be very accurate and thus they have used a 2-axes variant. However, in our system the readings need to very accurate as it is used by the Android application to determine whether the patient is sitting, standing, walking etc. Thus the aim of [1] is to estimate fall detection but the idea of using accelerometer sensors has been quite efficient and we would thus be using it in our system too. We found that the accelerometer sensors suit our requirements [3] [6] [7].

The aim of [2] is to determine the knee joint angle using gyroscope and flex sensors for rehabilitation purposes. From this paper we got the idea of monitoring body joints as BSN is a field which is rapidly growing and needs relatively easy to use and comfortable wearable computing to monitor the body joints. Thus we are concentrating on monitoring the knee joint movement rather than the angle as it wouldn't be an effective way to monitor the patient's movements. Instead we would be using a pie chart that attractively displays the data and yet effectively monitors the patient's knee to ensure quick recovery. Renesas RL78 is a 16-bit microcontroller widely used in various industrial and medical applications [4] [5]. Thus we have chosen it for our system. Further [3] gives an overview of the advantages and characteristics of using accelerometer sensors. This was a motivation for us to use this sensor in our system too. Also we wanted the patient to actively monitor his knee by providing him with a very efficient GUI. Thus we came up with idea of an Android application which communicates with the Renesas Microcontroller [4]. The Android application also provides persistent storage by keeping a history of previous readings that it received via Bluetooth. This makes sure that ADL i.e. activity of daily living [1] is monitored on a long term basis. This is feasible by interfacing a Bluetooth chip with the microcontroller and thus it transfers the sensors data via Bluetooth and hence it's not dependent on internet connectivity. Also we have used an emergency button which when presses trigger a buzzer which can be deactivated by pressing the reset button. This ensures that if the patient is in tremendous pain he can gather the attention of the people around him and also an Email alert is sent to the doctor from the Android application along with the patient's GPS location. In addition to this a SMS alert is also sent to the intended recipients.

The prototype that we have proposed is shown in Figure 2. It efficiently takes ideas from different papers and integrates them to achieve maximum benefit and also ensures that the design is cost effective but at the same time efficiently solves the problem at hand. Thus the tradeoff between performance and cost has been effectively managed.

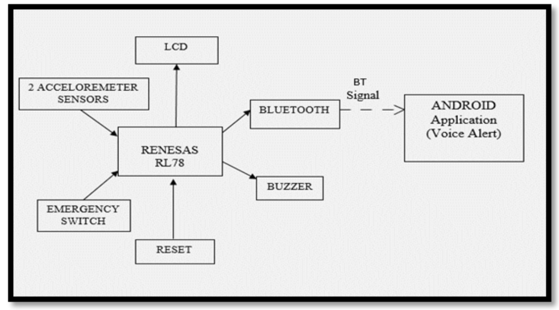

Figure 2 Proposed System Design

3. DESIGN

3.1 Introduction

The different modules needed and their descriptions for the Knee Joint Movement Monitoring system are given below:

- **Accelerometer sensors Module**: This module is responsible for identifying the patient's activity such as walking, running etc. ADXL335 sensors are interfaced with the Renesas RL78/G12 microcontroller. The aim of this module is to read the data from the pins to which these sensors are connected to.
- **Microcontroller Module:** This module is the most important module in the project. An embedded c program is written and flashed to the microcontroller. This program ensures that the data is being read from the accelerometer sensors and this analog voltage is converted to a digital value and is sent to the Android application using the Bluetooth module. Further suitable messages are constantly displayed on the LCD screen which is also interfaced to the microcontroller. Depending on the voltage value, the patient's activity is identified
- **Bluetooth Module**: This module is responsible for interfacing a Bluetooth chip with the Renesas Microcontroller to ensure that data communication takes place between the Android Application running on the patient's phone and the Microcontroller.
- **Database Module**: This module's function is to store the data in the Android Application and will be helpful when the patient or the doctor needs to monitor the physical activities performed by the patient.
- **Email Module**: This module's function is to send an Email alert to the intended recipient's Gmail account along with the current position of the patient which is obtained using GPS when the emergency button is pressed.
- **SMS Module**: This module deals with a SMS being sent to the intended recipient when the emergency button is pressed.
- **Android Application Module**: This module's function is to receive the data sent by HC05 Bluetooth module and to pictorially represent

the activities performed by the patient using a pie chart. Further the application has a login page for authentication and also plays out a voice alert when the patient's posture applies excess stress on the knee joint. Also the Android Application has a record of all the activities performed by the user in the past with the help of a SQLite Database.

3.2 Architecture Design

Figure 3 shows the flow diagram for this project. It gives a design overview of the project.

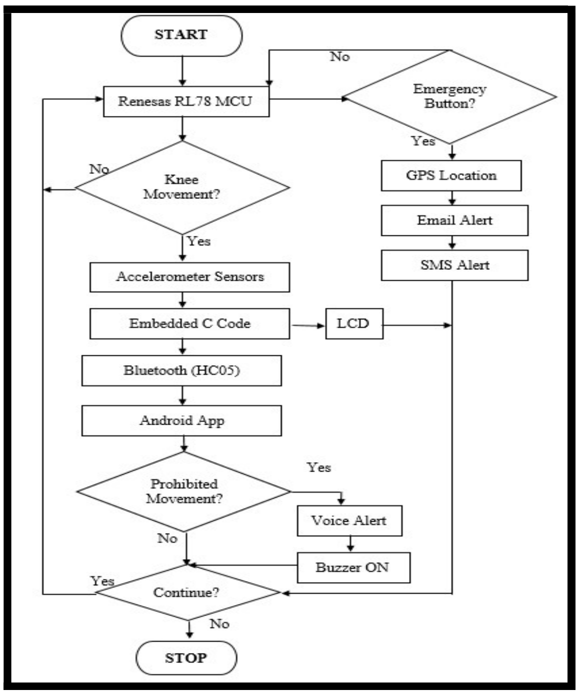

Figure 3 Flow Diagram

The system architecture is given in Figure 4. This diagram depicts the different modules that were discussed in the design introduction and the relationship between the modules and how they communicate with one another. The dotted line between the Bluetooth module and the Android application indicate the Bluetooth signal used for communication between the two and also a voice alert is played when the emergency button is pressed. All incoming arrows to Renesas RL78 indicate input features and all outgoing arrows indicate output features respectively. The dotted line between Android application and Email Alert indicates the usage of Internet connection to send the alert. The dotted line between the Android application and the SMS alert indicates the usage of some money in sending the message. In Figure 4, SMS alert, database and Email alert are the modules associated with the Android application whereas LCD, Buzzer, Bluetooth, Accelerometer sensors and Emergency switch are the hardware components associated with the microcontroller. The system architecture helps in designing the UML diagrams like the Class diagram, Sequence Diagram and Data flow diagram.

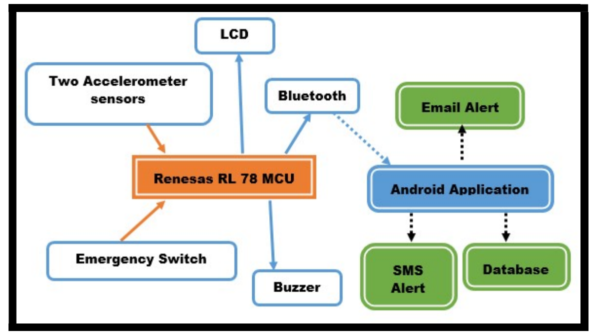

Figure 4 System Architecture

3.3 Class Diagram

The relationship among various classes and their interdependencies are effectively modeled and designed using the class diagram with the help of association, aggregation, composition and generalization which is shown in Figure 5.

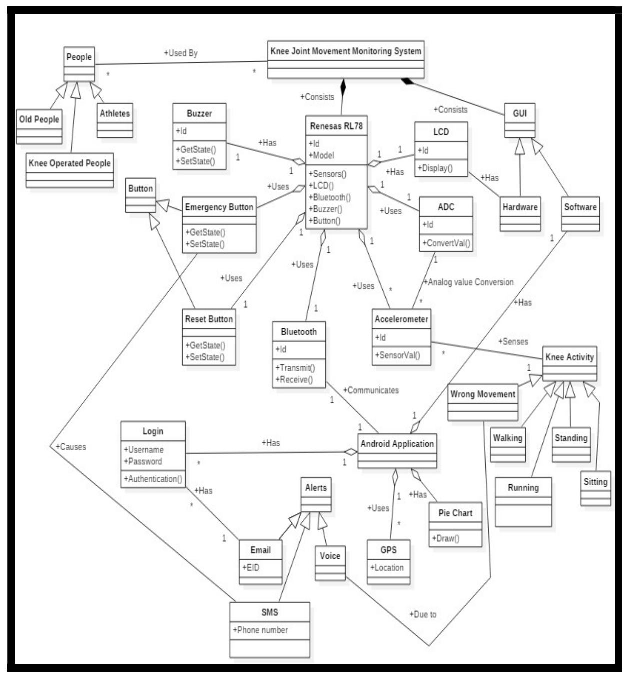

Figure 5 Class Diagram

3.4 Sequence Diagram

The sequence diagrams for this project are given in Figure 5, Figure 6 and Figure 7. These are helpful in designing the order of occurrence of events and the corresponding actions that needs to be performed.

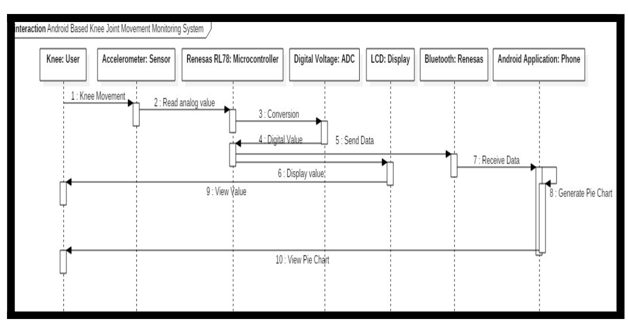

Figure 6 Sequence Diagram for the Overall System

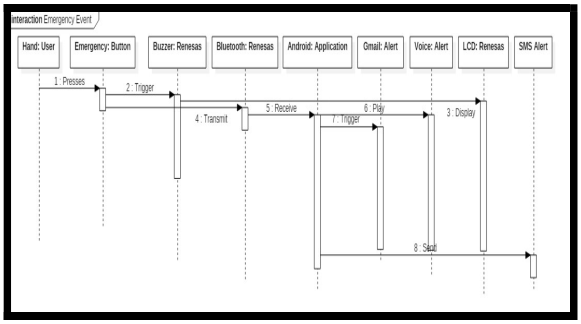

Figure 7 Sequence Diagram for the Emergency Event

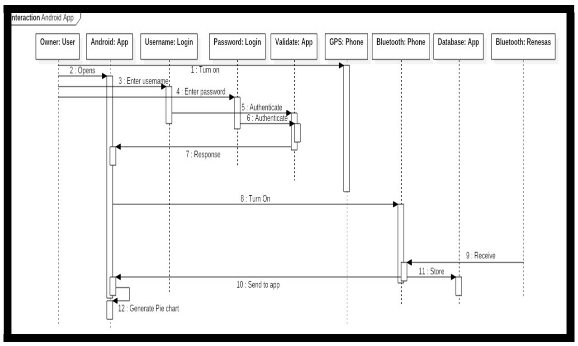

Figure 8 Sequence Diagram for Android Application

3.5 Data Flow Diagram

The data flow diagram is responsible for depicting the flow of data among the various entities and modules in the system. It is shown in Figure 9.

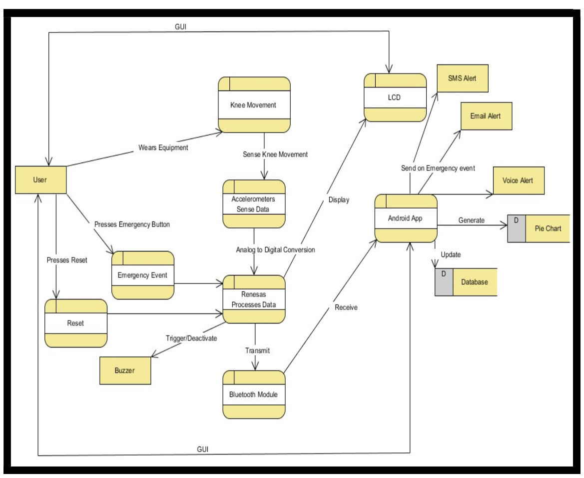

Figure 9 Data Flow Diagram

4. IMPLEMENTATION

4.1 Tools Introduction

The hardware components needed to build this prototype are as follows.

- **Renesas RL78/G12 microcontroller** is the core component of this system. RL78 is a 16-bit microprocessor which has CISC architecture. Flash memory is the type of memory that is present inside this MCU. We are using a 30 pin variant of this MCU along with 16kb ROM capacity. It has a variety of household applications. Thus this MCU is named according to the Renesas Standard as R5F102AAA and belongs to the G12 family. The meaning of this name is shown in Figure 10. The Renesas development board used for the project and is shown in Figure 11.

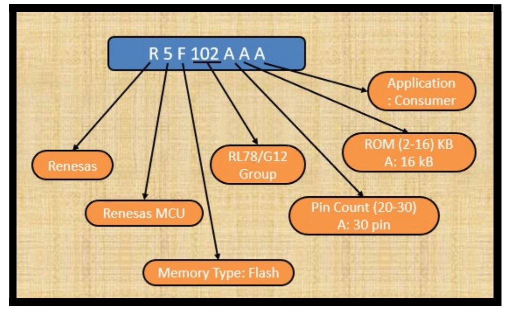

Figure 10 Meaning of R5F102AAA

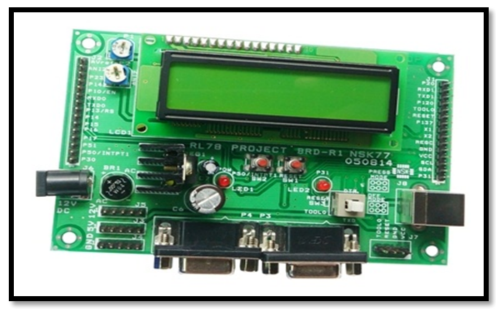

Figure 11 Renesas Development Board

- **3 axes Accelerometer sensors (ADXL335)** are the ones used in this prototype to determine the activity of the user by placing one of them above the knee joint and the other one below. They work on the basis of gravitational force rather than acceleration force. We are using a capacitive variant of this sensor. Thus whenever there is a motion in the knee, the movable plate of the capacitor moves towards one of the fixed plates which produces a difference in capacitance across the fixed plates which is amplified to produce an analog voltage as the output. The 3-axes (x, y and z) accelerometer sensor is shown in Figure 12.

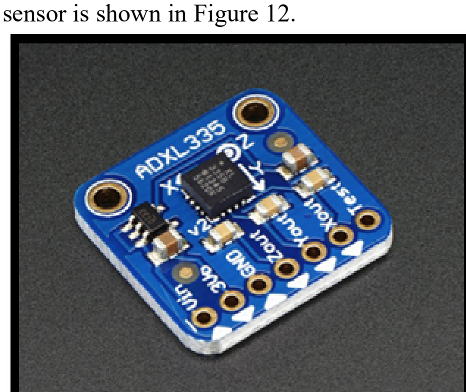

Figure 12 ADXL335 Sensor

- **HC05 Bluetooth module** is a type of SPP (serial port protocol/profile) module and is interfaced to the RL78 MCU using UART (Universal Asynchronous Receiver Transmitter) to provide communication between the android application and the MCU. Figure 13 shows the HC05 Bluetooth module.

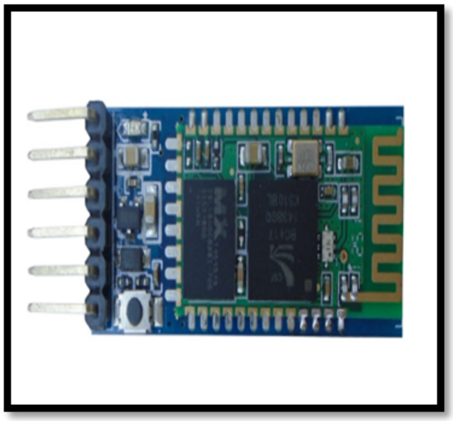

Figure 13 HC05 Bluetooth Module

- **NAND flash** board is needed to flash the Embedded C code into the Renesas MCU. This board is shown in Figure 14. NAND flash is a type of flash memory which does not need power to retain the data.

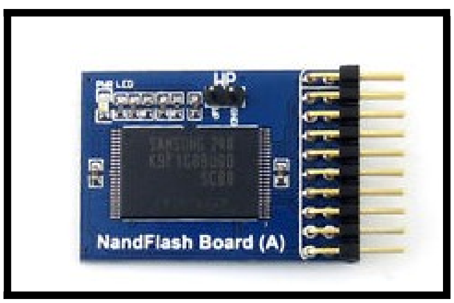

Figure 14 NAND Flash

- In addition to these main components, one 16*2 LCD, buzzer, an android phone and a knee cap is used. These are shown in Figure 15. Also a 5v power cable, flash cable and few jumper wires are needed.

Figure 15 Additional Components

4.2 Overall View of the Project

Embedded C code is written using Cube Suite+ IDE to perform the following functions

- Read the analog voltage output of the sensors and convert them to digital values.
- Determine the posture or activity of the patient from these values and assign a unique code for it.
- Display these digital values along with the activity on the LCD.
- Send the unique code to the Android application via Bluetooth with the help of HC05.
- Trigger the buzzer when the patient does a knee movement which is prohibited.
- Display Emergency Alert on the LCD when the emergency button is pressed.

The Android application is written using Eclipse IDE to perform the following functions

- Connect and pair to HC05 module by turning on Bluetooth.
- It has a login page to authorize the user.
- Also a field called Emergency number is present to send SMS alert when the emergency button has been pressed.
- To generate a pie chart.
- To clear the existing data.
- To send an Email alert along with the location of the user using internet and GPS.

Figure 16 gives the Overall view of the project in terms of Implementation.

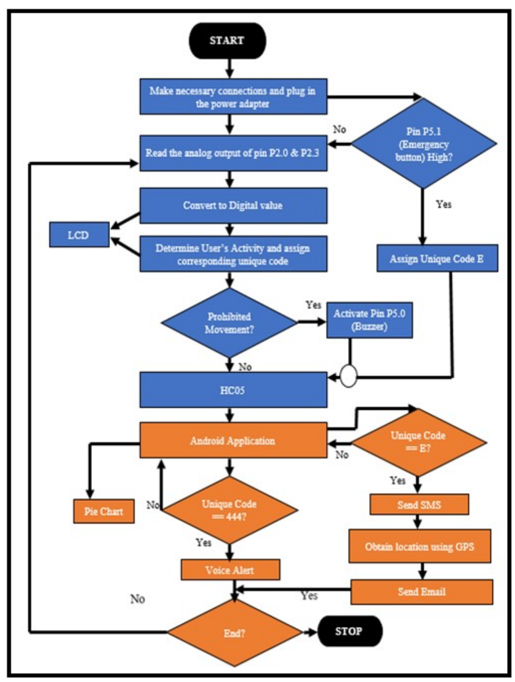

Figure 16 Overall View of the project in terms of implementation

In the above Figure, all the blue colored boxes correspond to the tasks performed by the embedded C code and manually whereas the orange colored boxes correspond to the tasks performed by the Android Application.

4.3 Algorithm and Implementation Details

This algorithm deals with the hardware aspects of the project.

Input: Renesas MCU with all necessary connections and prototype worn by user.

Output: Determining user's knee activity and sending data.

1. Avolt ← Sensor Value due to knee motion.
   //Avolt= analog voltage output
2. Dvolt ← ADC (Avolt)
   //Dvolt= Digital Voltage
3. DispLCD(Dvolt)
   //Function to display data on LCD
4. If (Dvolt >= Ut && Dvolt<= Lt)
   // Ut= Upper Threshold; Lt=Lower threshold
   I. DispLCD("Walking")
   II. UART_Send("Unique_code")
5. If (Dvolt >= Ut1 && Dvolt<= Lt1)
   I. DispLCD("Running")
   II. UART_Send("Unique_code")
6. If (Dvolt >= Ut2 && Dvolt<= Lt2)
   I. DispLCD("Sitting")
   II. UART_Send("Unique_code")
7. If (Dvolt >= Ut3 && Dvolt<= Lt3)
   I. DispLCD("Standing")
   II. UART_Send("Unique_code")
8. If (Dvolt >= Ut4 && Dvolt<= Lt4)
   I. DispLCD ("Wrong Movement")
   II. UART_Send("Unique_code")
   III. Buzzer = On
   IV. Delay(5s)
   V. Buzzer = Off
9. If (Ebutton == High)
   I. UART_Send("Emergency_Code")
10. Loop to Step 1.

The above algorithm deals with the hardware code which is flashed into the Renesas MCU and is written using Cube Suite+ IDE in embedded C. The ADC is a successive approximation ADC which produces a digital output by using the Equation 1.

$$D\_val= ((A\_val* Vref) / (2^n -1)) * 10 \ldots\ldots\ldots\ldots (1)$$

From D_val, each digit is extracted and converted to its ASCII value for displaying on the LCD.

In the Equation 1, Vref= 5v and 100 is a constant and A_val is the Analog value and D_val represents the Digital value of the voltage.

Table 1 Unique Codes for each activity

Sl. No.	Activity Name	Unique Code
1	Walking	111
2	Running	222
3	Sitting	333
4	Standing	555
5	Emergency	E
6	Wrong Move	444

Table 1 gives the details of the activity name and the corresponding unique code that is sent as we are using serial communication and it would be better to use small codes rather than sending the activity names. Table 2 gives the details of the commands for implementing the LCD.

Table 2 LCD Details

Sl. No.	LCD Function	Command
1	2lines and 5*7 matrix	0x38
2	Display ON, Cursor blinking	0x0E
3	Clear display Screen	0x01
4	Shift cursor right	0x06
5	Data Instruction	RS=1
6	Command Instruction	RS=0

Table 3 gives the threshold values for each activity.

Table 3 Activity Details

Sl. No.	Voltage Value range	Activity
1	155 to 175	Sitting
2	185 to 190	Standing
3	132 to 136	Walking
4	140 to 147	Running
5	146 to 153	Wrong Movement

The algorithm mentioned below deals with the Android application.

Input: Data from HC05 module

Output: Knee Joint Monitoring Application

1. Enter username and password.
2. If (uname == "admin" && pwd == "*****")
   [1] Logged in successfully.
   [2] Bluetooth = On
   [3] Connect to Knee cap
   [4] If (Bluetooth Key =" ****")
      Connected successfully
      Code ← Received data
      If (Code == 111)
        Wc++ // Walking count
      Else if (code == 222)
        Rc++ //Running count
      Else if (code == 333)
        Sitc++ // Sitting count
      Else if (code == 555)
        Stdc++ // Standing count
      Else if (code == 444)
        [i] Wmc++ //Wrong count
        [ii] Play Voice alert
      Else if (code == E)
        [i] Send SMS
        [ii] Fetch Location
        [iii] Send Email Alert
   [5] Else
      Wrong Bluetooth Key
3. Else
   Wrong Username/Password

With the help of the algorithms written above, the hardware code has been implemented using Cube Suite+ in embedded C. There are three classes namely the main class, Knee_ADC and Knee_LCD. Knee_ADC performs the task of converting the analog voltage output of the accelerometer sensors into digital value and then they are converted into their respective ASCII characters for displaying on the LCD module. Knee_LCD performs the task of initializing the LCD and sending data as well as command instructions to display the data using the information given in Table 2. The main class detects the various activities of the patient and sends corresponding unique codes mentioned in Table 1 and also is responsible for triggering the buzzer in case of wrong movements. It also makes calls to the functions present in Knee_ADC and Knee_LCD. In addition to these certain libraries have been imported to support certain features of LCD.

The android application has been implemented using Eclipse (Kepler) with Android ADT and SDK plugins. BluetoothSPPConnection class is responsible for connecting the android application to the HC05 Bluetooth module so that data can be transferred. Child_Movement_Graph_activity class performs the function of constructing the pie chart with the values of the following arrays: colors, ar_values and ar_namelist. Constant class is used for defining and initializing the counter. DatabaseHelper class inserts the activity name and count into a SQL database so that it can be fetched for drawing the pie chart. However, if the data is cleared then the database is truncated and the pie chart will be drawn using the new values. DeviceListActivity class in conjunction with BluetoothSPPConnection is responsible for connecting and pairing to HC05. Login_activity class is responsible for handling the events pertaining to the login page of the android application. Mail class is used for sending email alerts. MainActivity class is the core class which calls the functions from the other classes. Also its computes the location of the patient by using GPS and sends SMS alerts in case of unforeseen situations.

5. TESTING AND RESULTS

5.1 Introduction

In our project we have used two different programming languages namely Embedded C for programming the microcontroller and Java for developing the Android application. Thus we used Junit framework for testing the various components of the application and a tool called Flash Magic was used for testing the HC05 Bluetooth module. The testing of hardware components was manually done with the help of Cube Suite+ whereas for the android application, it was automated using Junit framework.

Also the Android Virtual Device Manager (AVD) was used to perform integration testing on the unit tested components of the Android application. An Android Virtual Device (AVD) is “an emulator configuration that lets you model an actual device by defining hardware and software options to be emulated by the Android Emulator”. Apart from performing functional testing AVD also aids in nonfunctional testing (compatibility/portability testing) as it can simulate different android phones from brands like Nexus, Samsung, and Nokia etc.

A framework called Robotium has been used for performing automated testing on the Android Application in conjunction with Junit. Robotium is “An open-source test framework for writing automatic gray box testing cases for Android applications. With the support of Robotium, test case developers can write function, system and acceptance test scenarios, spanning multiple Android activities”. When running the test case on the emulator or an actual device, you will see the values being entered into the UI components or see them being clicked as if you were doing it yourself. After the tests are completed, the JUnit view in Eclipse will show which tests have failed and passed. Figure 18 and Figure 19 shows the Junit and Robotium testing that was performed.

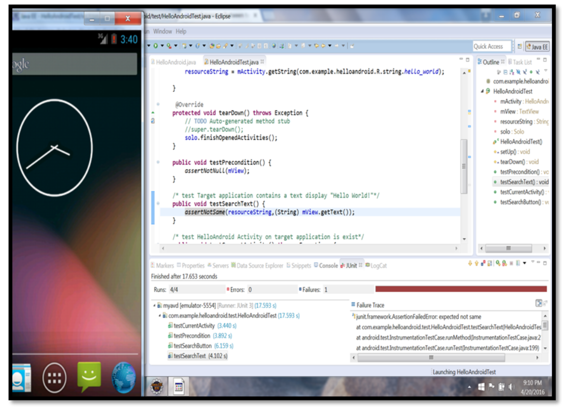

Figure 17 Junit Testing (Fail)

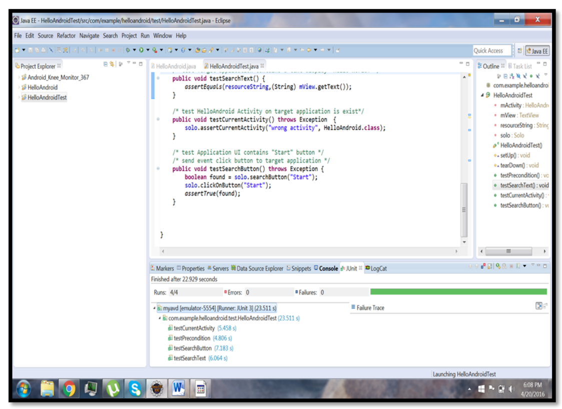

Figure 18 Junit Testing (Fail)

5.2 Snapshots

Figure 19 shows the process of pairing with the HC05 Bluetooth module. Only if the correct pin is entered the phone gets paired with HC05.

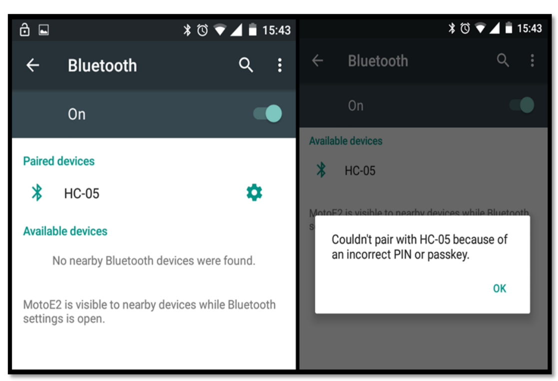

Figure 19 Pairing with HC05

Figure 20 shows the messages received by the android application for the different activities of the patient with its unique code. The received data gets accumulated in the SQLite database and when the generate graph button is pressed a pie chart as shown below is generated.

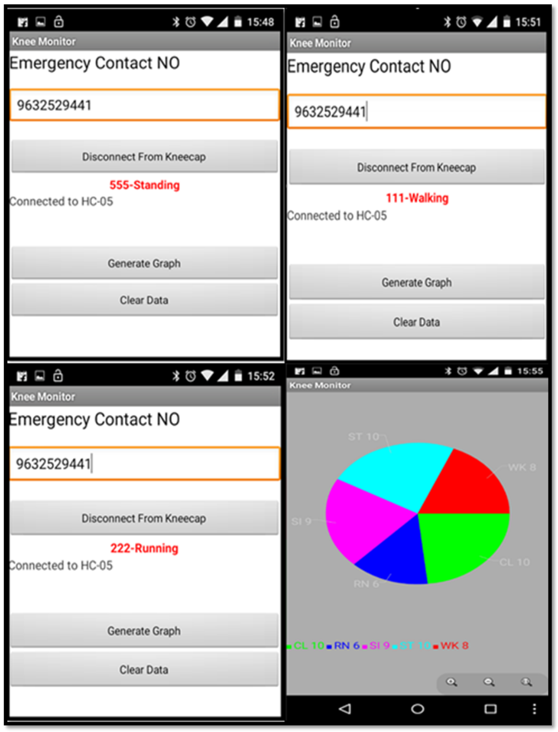

Figure 20 Android Application

Figure 21 shows the hardware components used in this prototype and the real time demo which was performed when the patient wearing the prototype was standing.

Figure 21 Working

Figure 22 shows the entire hardware kit that was built with the knee cap which is wearable for implementing the prototype real time.

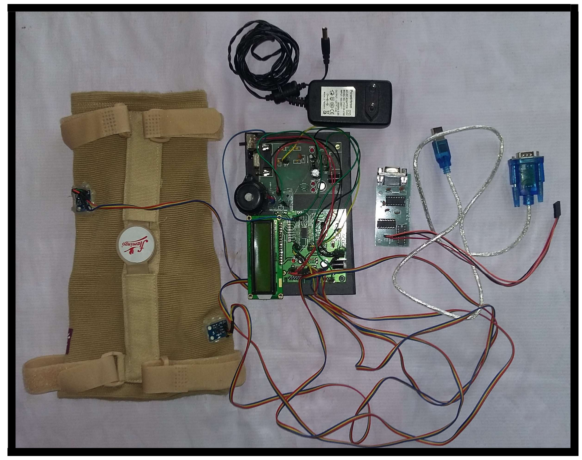

Figure 22 Hardware kit

6. CONCLUSION

The system developed is a prototype in which the microcontroller needs constant power supply for it to run. Also the prototype has been highly customized for a single patient. This project helps athletes, old people who have knee problems and people who have undergone knee surgery by constantly monitoring the different activities performed by the patient, which affects their knee. Also only certain postures such as walking, running, sitting and standing have been used to demonstrate the concept. The two major sections are the hardware code which is implemented using Cube Suite+ in embedded C and the Android application which is implemented using Eclipse Kepler with Android ADT and SDK plugins. The most important aspect to be considered in the future is to make the entire prototype wearable and to ensure that the fabrication costs involved due to it is low so that the product can be released in the market. This is because the cost of fabrication is quite high as it needs to ensure that the device is small, efficient and can be worn without causing any discomfort. Further the system must extend its variety in identifying the different postures and activities performed by the user. This system can be integrated with smart devices such as watches etc. and also the IOS version of the application is to be developed.

REFERENCES

[1] Masdar, A.; Ibrahim, B.S.K.K; Hanafi, D.; Jamil, M.M.A.; Rahman, K.A.A, "Knee Joint Angle Measurement System Using Gyroscope and Flex-sensors for rehabilitation", Biomedical Engineering International Conference (BMEiCON), 2013 6th, vol., no., pp.1, 4, 23-25 Oct. 2013.

[2] Lee, Y.; Kim, J.; Son, M.; Lee, M.; "Implementation of accelerometer sensor module and fall detection monitoring system based on wireless sensor network", Proc. 29th IEEE EMBS Annual International Conference, pp. 2315–2318, Aug 23–26, 2007, Lyon, France.

[3] Jeethendria, A.; Senthilmuguran, S.; "Monitoring Human Joint Movement Using Wearable Computing", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 4, April 2014

[4] Amin Ahmadi; Edmond Mitchell; Chris Richter; Francois Destelle; Marc Gowing; Noel E. O'Connor; Kieran Moran; "Toward Automatic Activity Classification and Movement Assessment During a Sports Training Session", IEEE Internet of Things Journal, Vol. 2, No. 1, February 2015.

[5] http://www.renesas.com/products/mpumcu/rl78/index.jsp

[6] Aszkler; Craig; "Acceleration and Shock and Vibration Sensors, in Sensor Technology Handbook", edited by Jon S. Wilson, 137-159. Burlington: Elsevier, 2005.

[7] http://www.sensorwiki.org/doku.php/sensors/accelerometer

[8] Alexander G. Dean; James M. Conrad; "Creating Fast, Responsive and Energy-Efficient Embedded Systems using the Renesas RL78 Microcontroller", Renesas Books, 2012.

[9] Alexander G. Dean; "Embedded Systems Design, Analysis and Optimization using the Renesas RL78 Microcontroller", Renesas Books, 2012.