Development of Android Based Real Time Monitoring System for Fire-fighters

by

Ganesan A/L Damudaren

16822

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical and Electronic Engineering)

MAY 2015

Universiti Teknologi PETRONAS

32610 Bandar Seri Iskandar

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Development of Android Based Real Time Monitoring System for Fire-fighters

by

Ganesan A/L Damudaren

16822

A project dissertation submitted to the

Electrical and Electronic Engineering Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL AND ELECTRONIC ENGINEERING)

Approved by,

____

(Dr. Micheal Drieberg)

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR, PERAK

May 2015

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

GANESAN A/L DAMUDAREN

ABSTRACT

Fire-fighters are trained rescuers especially in extinguishing hazardous fires and saving lives from dangerous situations however they tend to put their lives at risk while on the job. Statistic shows the number of deaths of fire-fighters while on the job are still high up to this year and a higher percentage of rapid fire progress and exertion are dedicated to the cause of death of fire-fighters while on the job. Therefore, a real-time monitoring on the physiological state for fire-fighters is something crucial to be done. However, the Fire and Rescue Department of Malaysia practices the traditional communication method which is by communicating via walkie-talkie. The practice of real-time assessment should be carried out by Fire and Rescue Department of Malaysia in order to avoid having a fire-fighter’s live at risk. This could be achieved by using the ARMOR (Android Based Real Time Monitoring System) whereby it can transmit voice data and physiological data such as heart rate, respirator rate, peak acceleration and posture. Based on the research and critical analysis, an android platform have been found to be a suitable selection as it supports Bluetooth Wi-Fi and Radio Frequency, accessible from any android devices and it is user friendly. As a result, a real-time intelligent monitoring system was successfully developed on an android platform. The physiological data for heart rate, respiration rate, posture, and peak acceleration was successfully transmitted and monitored on an android device at real-time.

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to my supervisor Dr. Micheal Drieberg and co-supervisor Dr. Irraivan Elamvazuthi for their guidance and determination to help me throughout the journey of completing my final year project. I was able to gain a better understanding and more knowledge regarding this project thanks to their continuous support and experience sharing throughout the period. Dr. Irraivan Elamvazuthi provided me with the equipment which helped me a lot throughout the completion of my final year project.

I would like to extend my appreciation and gratitude to the coordinators of FYP I and II from Electrical and Electronics department for assisting me throughout the period of completion of my final year project. I would like to thank for guiding me throughout the submission of all the important reports and advising me on the correction that needed to be done in order to produce a flawless report. Besides that, they have provided briefings that was very helpful in order for me to complete my final year project. Adding on, I would like to thank my external and internal examiners for giving me the feedback which have helped me to improve my project.

I would also like to convey my gratitude to my parents Mr. and Mrs. Damudaren for giving me the moral support and confidence throughout the period of my completion. Last but not least, I would like thank all my course mates for supporting and guiding me throughout my final year project completion.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Project Background

The background of study of this project is discussed in this chapter along with the problem statement of this project. The objective of this particular project and the scope have been discussed thoroughly in this chapter.

1.1 Background of Study

A fire-fighter is a rescuer generally trained in fire fighting, mainly to extinguish hazardous fires that threaten assets, and to save people from unsafe conditions such as collapsed or burning buildings and collided vehicles. The convolution of our current industrialized life which creates a greater distinction of hazards has made an increase in the skills needed in fire fighting technology and also widening the fire-fighter's rescue job. Rescue manoeuvres involves searching and taking out trapped victim from any kinds of hazardous conditions in which includes rescuing animals trapped on the tree.

An organized team is needed due to the complexity of fire fighting. During each fire rescue operations, the superior officer or commander will assign each of his fire-fighters to a specific duty. Sometimes there will be a necessary for a firemen's duty to change during a fire rescue operations. Depending on the severity of the disaster, sometimes fire-fighters will have to spend days rescuing victims and putting of the fire.

Saving a life or putting off the fire situates fire-fighters life at risk during each operation and this becomes an occupational hazard. Every year fire-fighter's life is taken away during a fire rescue operations. Figure 1.1 shows a survey on the statistics of work-related fire-fighter deaths which was conducted by NIOSH (National Institute of Occupational Safety and Health).

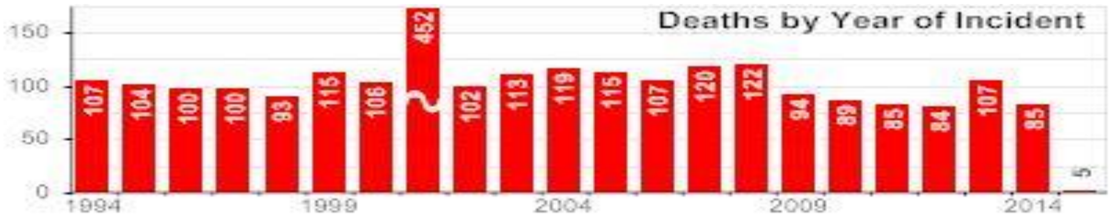

Figure 1.1: Fire-fighter Fatalities [1]

According to the statistic, since the beginning of time when fire fighters becomes operational till up to date, the occupational hazard have always existed. Fire fighting job can be voted as a life-threating occupation.

The National Fire Protection Association (NFPA) have conducted an analyses on the natures of duty associated with fire-fighter deaths. Contained within their report are the root of deadly injuries to fire-fighters [2]. The analyses conducted by NFPA gives a clearer picture of the cause of deaths as part of fire-fighter's occupational hazards.

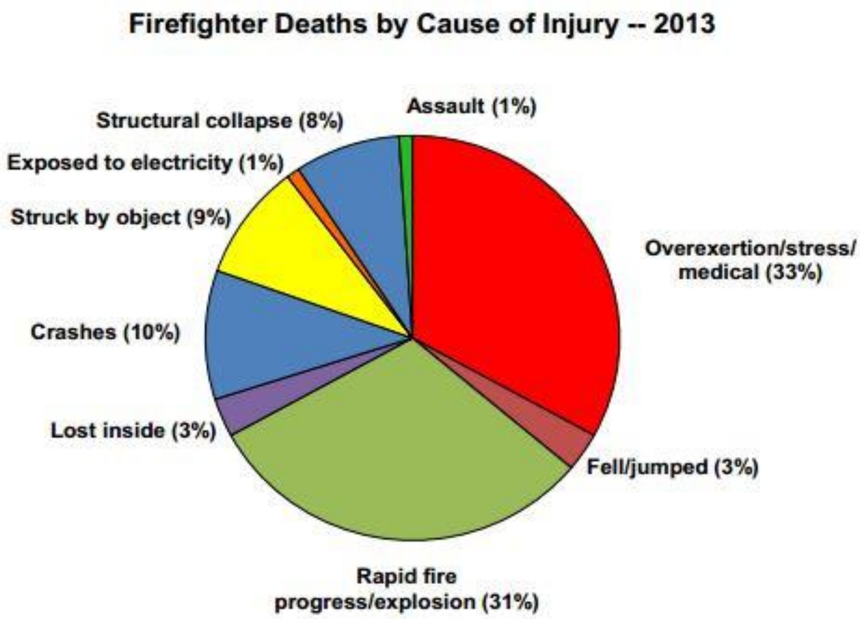

Figure 1.2: Cause of fire-fighter fatalities [2]

From Figure 1.2, it shows a minimal percentage of fire-fighters losing their lives due to fell or being exposed to electricity. Adding on, it also clearly states that the frequent cause of injury is due to overexertion or stress which is 33%. Fire-fighter's life is always at risk during any kinds of fire rescue operations. The occupational hazard of fire-fighter is unpredictable in which leads to a valid reason for monitoring fire-fighter's physiologically.

Real-time assessment of the physiological status and the status of mounted device (oxygen level) pressure of the fire-fighter is very crucial to be monitored during a fire rescue operations. This real-time assessment should be able to assess the baseline of physiological characteristic such as aerobic fitness, sleep history and heart rate.

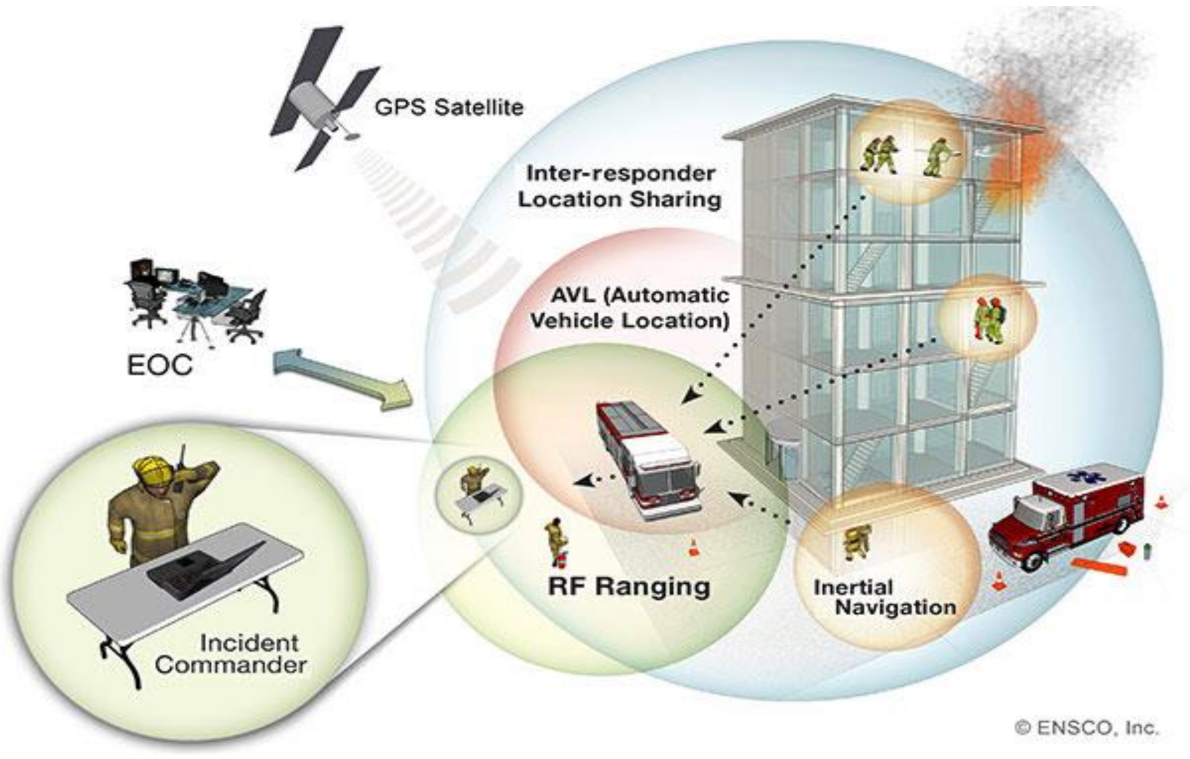

Figure 1.3 ENSCO Inc.'s Real-Time Assessment [3]

Figure 1.3 shows an overall view on communications method. It shows that fire-fighters which have been divided into two groups located in two different part of the building reports back to the incident commander wirelessly via radio frequency. Their fire fighting vehicle becomes a base of their communication.

Apart from that, it should be able to provide mission support such as improve "who, where, when" situational awareness, guide acute and chronic work or rest cycles, and reduce the likelihood of environmentally related injuries such as heat stroke which is quite common when it comes to fire rescue mission. Besides that, casualty evacuation to be facilitated and the quality of after action reviews to be improved.

1.2 Problem Statement

The current system that is being used by Fire and Rescue Department of Malaysia is still conservative whereby real-time assessment on the fire-fighters is not done. The risk of each fire-fighters who are exposing their lives in a fire rescue operation is still high due to lacking of real-time assessment system.

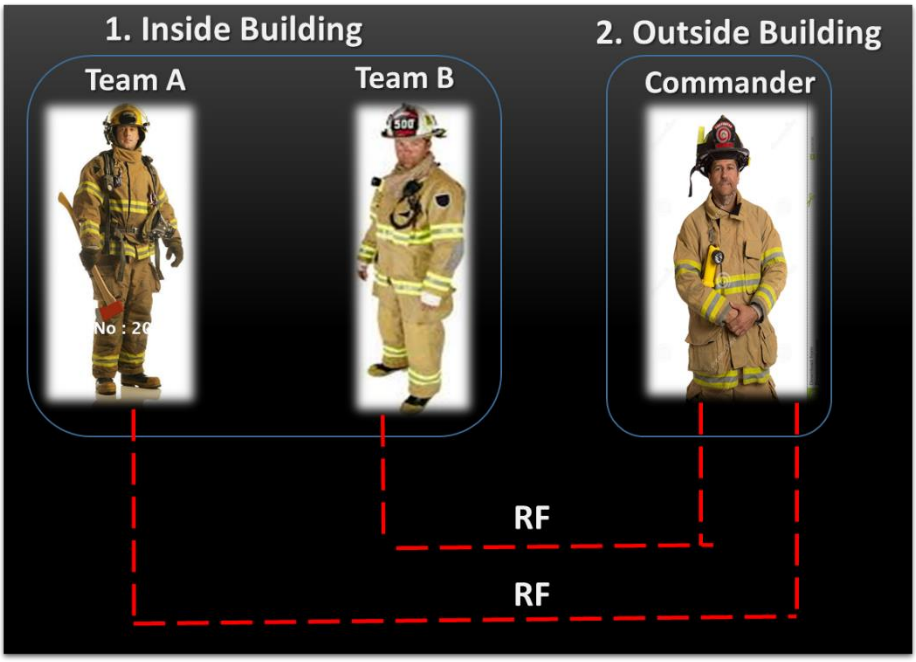

Figure 1.4 Typical Communication Method of fire-fighters

Figure 1.4 shows the typical communication method that is being used by Fire and Rescue Department of Malaysia. During a fire rescue mission, the commander will be positioned outside the building while his team(s) is sent into the building to accomplish the mission. The commander will only be able to get a voice feedback from his team and during any unfortunate event such as if any one of the team members fainted, the commander will not be able to get any accurate status from his team members. The basic walkie-talkie communication between fire-fighters is not sufficient to assure their safety during a high risk fire rescue operation.

Besides that, the walkie-talkie being used by the fire fighters in Malaysia does not consist of multi-channel terminal to receive data other than their voice data. The walkie-talkie that is currently being used will not be able to be used for real-time monitoring system.

1.3 Objective

The objective of this project are as follows:

- To assess the technologies for real-time assessment
- To develop a real-time monitoring system
- To validate the developed system through various trials

1.4 Scope of Study

The scope of this project is concentrated on the communication system. A scope of literature reviews has been carried out to determine and study the existing communication technologies that can be used as a platform for real-time monitoring system. A suitable terminal is to be selected to assist in real-time data monitoring system. Adding on, the programming language used is to be studied so that the device can be programmed accordingly to achieve the selective functions such as accessing and monitoring the physiological data at real-time. Various performance test is to be carried to validate its stability of communication especially in transferring and receiving data without any loss of data.

CHAPTER 2
Literature Review

There are plenty types of communication platform that can be used to transmit and receive data accurately and quickly. The best selection of the platform as the base carter the wireless communication as part of the real-time monitoring for the fire-fighters are discussed in this section. The selection of the communication platform is selected based on certain criteria in which is discussed in this section.

2.1 Real-Time Monitoring for Fire-Fighters

Real-time assessment and monitoring fire-fighters during rescue operations can be formed using top sensing technology integrated with wireless/radio-communications technology for a dependable and essential instrument. The use of wireless monitoring system is important as the incident response can be acquired in real-time accurately and quickly. Furthermore, use of wireless technology increases the mobility of the fire-fighters.

An important point to this new era of remote wireless detection monitors is the capacity to show alerts and information continuously that is available by means of the Internet whereby data from the first responders can be shared on emergency basis [4]. A radio channel is used as a platform to transmit real-time data accurately and quickly without any loss of data [5].

2.2 Communication Technologies

The transferring and receiving back of a data or an information is known as communication. Up to this date the technology used for the means of communication has increased in the sense of data transmission rate, ease of accessibility, number of users at the same time and convenient [6]. During the days before wireless communication technology existed, there was a need for telephones and internet for the means of communication.

The advancement of technology has allow the transition of landlines to wireless communication whereby it allows people to access from anywhere at any time anyhow. Faster internet is now available in wireless allowing user to access it easily and more conveniently. Following up with the latest communication trend, wireless communication technology has the highest influence in this current generation in sense of reliability and effective.

2.2.1 Existing Wireless Communication Platforms

Based on the Table 2.1 article regarding the design of physical bound acquirement and communication element constructed on CC2430, the usage of Zigbee technology helps to transmit data from medical sensors to monitoring equipment via wireless transmission, which reduces the usage of cable links [7].

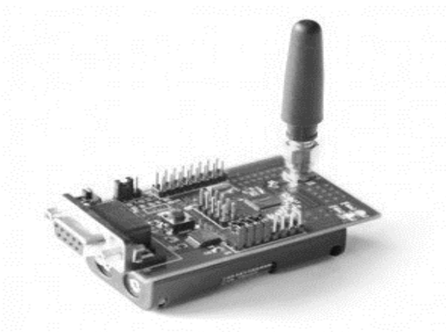

Figure 2.1 Zigbee CC2430 module

There is a mini monitoring network between the observing instrument and Zigbee sensor nodes. The physiological information is detected using a controller which is fitted on the sensor nodes then the information is transmitted via wireless transmission to the selected equipment [7]. Communication transmission system can be monitored for 24 hours due to its low power consumption [8].

Apart from that, wireless sensor network or more commonly known as WSN is a wireless network that comprise of independent devices such as low power consuming processor, flash memory, ADC and RF transceiver [5] [9]. Feedback can be received and monitored remotely via wearable monitoring system. A wearable system is used to detect the information and it consist of data collection hardware, remote centre and data analysis [10].

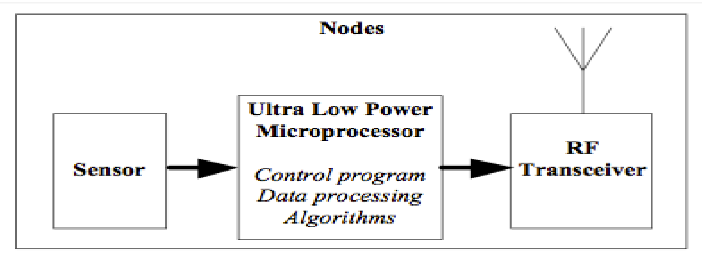

Figure 2.2 WSN Block Diagram

Based on another article which uses SoC or also known as System on Chip platform together with Bluetooth wireless network. Bluetooth module is set as the transmitter whereas the SoC is set as receiver platform [11]. Besides that, according to an article by Ramamurthy, Prabhu and Rajit, a smart sensor platform which is based on a patent pending technologies has a plug and play proficiencies. This supports hardware interfaces and communication desires for many sensors [12].

Another platform that was introduced by Motorola is TETRA portable terminal [13] which provides an upright functionality combined with GPS and non-stop encryption [14]. TETRA is an open standard for digital mobile radio communication and it also known as Terrestrial Trunked Radio [15]. TETRA terminals has quick access to voice and data service, besides that it has programmable interface which can be programmed according to the user. It also has radio call in the frequency band of 380 – 400 MHz, 806-870 MHz spectrum and RF channel bandwidth of 25 kHz [13] [15]. This Motorola TETRA provides convenience and hands free mobility [16].

Figure 2.3 Motorola TETRA MTP850 [13]

Last but not least, Android operating system is also commonly used as platform for wireless communication such as Bluetooth, Wi-Fi, 3G, Radio Frequency and others. Android operating system is constructed on Linux and it is an open source operating system which can be used freely by anyone [17]. Android developers announced Bluetooth Low Energy (BLE) in their operating system version 4.3(API Level 8) in which helps to improve the Android applications to receive and transmit data with other Bluetooth devices especially health monitoring devices [18]. Android operating system helps users to control wireless communication freely without any restrictions. Connectivity of the wireless communications depends on the capabilities of Android devices. Android operating system provides good data visualization through Graphical User Interface (GUI) [19]. There are rugged Android devices which can be used for tough and rough conditions such as tough smartphones which also has its own built in walkie-talkie radio [20] [21]. Figure 2.4 shows examples of tough smartphones by Runbo.

Figure 2. 4 Runbo Tough Smartphones [24]

2.3 Related Work

Table 2.1 shows the comparison of the available platforms for the wireless communications which have been summarised based on the articles reviewed.

Table 2.1 Platform Comparison

No	Platform	Communication	Authors	Title	Year	Merits	Demerits
1	Zigbee, CC2430	Zigbee	Qi Zhao and etc [7]	Design Of Physiological Parameter Acquisition And Communication Module Based On CC2430	2008	Low power usage. Cost Saving	No means of radio call. Bulky
2	Wireless Sensor Network	Bluetooh, Wi-FI	P.S. Pandian [9]	Wireless Sensor Network For Wearable Physiological Monitoring	2008	Low power usage	Radio call not available. Bulky
3	Wireless Sensor Network	Zigbee, Bluetooth, Wi-Fi	Jin Soo CHOI and MengChu ZHOU [5]	Recent Advances In Wireless Sensor Networks For Health Monitoring	2010	Low power usage. Fast data transmission.	No means of radio call.Bulky
4	SoC (System on Chip)	Bluetooth	Jzau-Sheng Lin and etc [11]	A Physiological Signal Monitoring System Based On An Soc Platform And Wireless Network Technologies In Homecare Technology	2009	Low power usage.	No means of radio call. Bulky
5	SoC, Wireless Sensor Network	Bluetooth, Zigbee, WiFi	Shyamal Patel and etc [10]	A Review Of Wearable Sensors And Systems With Application In Rehabilitation	2012	Low power consumption,	Radio call not available
6	Smart Sensor Platform	Wi-Fi, Bluetooth	Harish Ramamurthy and etc [12]	Wireless Industrial Monitoring And Control Using A Smart Sensor Platform	2007	Plug and Play	Radio call not available
7	TETRA portable terminal	Bluetooh, RF, Wi-FI	Motorola Solutions [13]	Motorola Tetra Terminals	2015	Portable, Radio call available, Programmable	Costly
9	Tough Android Phone	Bluetooh, RF, Wi-FI, 3G	Paul Strauss [20]	Runbo X3 Rugged Phone Wants to be the Rambo of Smartphones	2014	Radio Call, Friendly GUI,	Marginally higher
10	Android OS	Bluetooh, RF, Wi-FI, 3G	Yao Wang and etc [18]	Application of Android Mobile Platform in Remote Medical Monitoring System	2015	Radio call available, Friendly GUI	Limited RF depend on devices

2.4 Critical Analysis

There are certain important criteria that should be met in order to choose the suitable platform for the real-time monitoring system. Firstly it has to be a multi-channel terminal and must be able to receive multiple data layer at real-time. The platform must be able to receive data at a range within 4meters and also be able to transmit at long range more than 100meters. Secondly, it has to be portable and durable because it is very crucial that the device does not slow down the fire-fighters and must be able to cope up with tough handling. Lastly, the platform must be able to make radio call up to 10km range.

Based on Table 2.1 a comparison was made to choose the best platform which meets the criteria. The merits and demerit was also considered as part of the selections. Based on the comparison, Zigbee CC2430 platform which communicates via Zigbee has low power usage and has better coverage compared to Bluetooth but at the same time it has no means of radio call and it is bulky to be carried around. It does not serve the purpose of being portable terminal. Although it can receive and transmit data, but it does not have a good data visualization as compared to Motorola Tetra or any Android devices.

The wireless sensor network uses Zigbee, Bluetooth and Wi-Fi as the medium for communication for its platform. Although wireless sensor network has fast data transmission and consumes less power, it is not suitable to be used by the fire-fighters as the platform is bulky and mostly importantly it does not have a radio call function. Hence it does not meet criteria needed for the real-time monitoring system. Adding on, the system on chip (SoC) also uses less power and it is communicated via Bluetooth and Wi-Fi. Overall, this platform still does not meet the criteria needed. Apart from that, there is a smart sensor platform which has a plug and play function which is easy to be used and it transmit data via Wi-Fi and Bluetooth. Unfortunately it does not meet the requirement listed.

Next, the Motorola TETRA is a portable device which can be able to communicate and transmit data via Bluetooth at short range, Wi-Fi and Radio frequency at long range. Although the Motorola Tetra does meet the requirements, but yet the cost is higher compared to the other platforms. Besides that it has its own software development kit which could be a trouble as it is not an open source software like Android. Attending tutorial classes are the only way that the user could learn about the software development kit because it has limited source on the net. There will be difficulties in visual when comes to monitoring due to the small display size compared to an Android device which has a wider display. The Motorola Tetra cannot be user friendly device as it does not many functions that can be added.

Lastly, Android operating system meets all the criteria listed in order to develop a real-time monitoring system. Android operation serves as platforms for Bluetooth, Wi-Fi, 3G, Radio Frequency and others. Bluetooth can be used for short distance and besides that, Bluetooth Low Energy have introduced by Android which helps to improve the communication, whereas for long distance, Wi-Fi and Radio Frequency can be used. It is an open source which gives the freedom for user to work on it. Android applications can be developed using Android Studio which is a powerful Java-based development kit. Better data visualization can be created and user friendly GUI can be developed. This allows the fire-fighters to be monitored easily. Besides that, android platforms are also portable and affordable. There are tough smartphones available in the market which can be used by fire-fighters for any rough and tough situations.

2.5 Summary

As a conclusion, it can be clearly seen that Android operating system has a better functions as a communication platform compared to the other available platforms. It fulfils the criteria listed for the platform as part of the real-time monitoring system. Android devices are portable, supports wireless communications such as Bluetooth, Wi-Fi, Radio Frequency and others. Besides that, a better visual display with user friendly GUI can be programmed accordingly which contributes to the data visualization for the real-time monitoring system.

CHAPTER 3
Methodology

The research methodology used for this project, tools and software, organizational chart for this project are all discussed in detail in this section. This section provides information on the necessary steps that has to be taken in order to achieve the objective of this project

3.1 Project Methodology

Figure 3.1 shows the project or research methodology of this project.

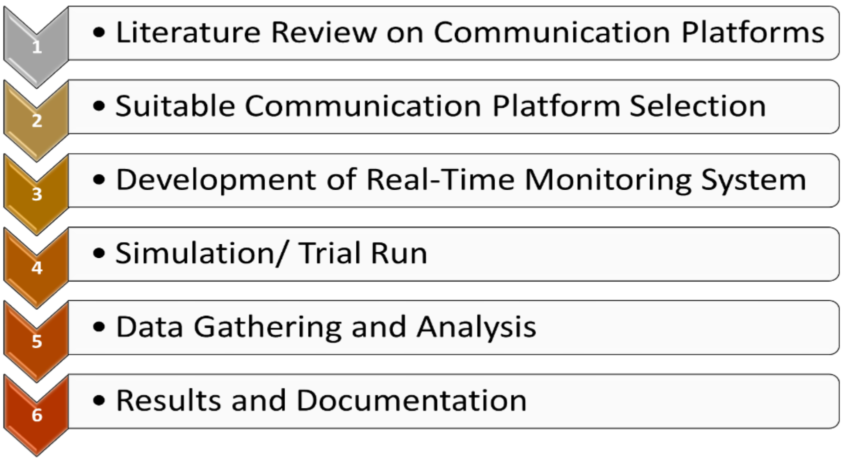

Figure 3.1 Project Methodology

Based on Figure 3.2, it shows that the project methodology have been divided into seven important step in order to achieve the objective. Step 1 is on the literature review of the wireless communication platform whereby reviews on past and current researches about various types' platforms which are used to carter wireless communications are analyzed. Step 2 is to identify the best platform based on the criteria for the real-time monitoring system. Next step will developing the real-time monitoring system on the selected platform. Step 4 is to simulate using the developed real-time monitoring system with as many trial runs. Then followed by step 5 whereby the data are gathered and analyzed. Lastly, step 6 is to tabulate the result and complete the documentation.

3.2 Gantt chart

Table 3.1 and 3.2 shows the project timeline for FYP1 and FYP2.

Table 3.1 Gantt chart FYP1

No	Project Flow		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Literature Review		■	■	■	■										
2	Platform Identification for Real-Time Monitoring System					■	■									
3	Development of Real-Time Monitoring System					■	■	■	■	■	■	■	■	■	■	■
4	Proposal Defence										●					
5	Documentation	Extended Proposal						●								
		Interim Report Draft Submission													●	
		Interim Report Submission														●

Table 3.2 Gantt chart FYP2

No	Project Flow		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Development of Real-Time Monitoring System(Debug Errors)		■	■	■	■	■									
2	Simulation/Trial Run					■	■	■	■	■						
3	Data Gathering and Analysis									■	■	■				
4	Project Viva														●	
5	Documentation	Progress Report							●							
		Final Report Draft											■	■		
		Dissertation												●		
		Hardbound Submission														●

Key Milestone

Progress

Table 3.1 and 3.2 shows the project timeline for FYP1 and FYP2. The duration taken for each project methodology has been clearly shown in the both of the table in which has been highlighted in blue whereas the key milestones in both the table has been highlighted in red circle. This would be the guidelines for every document submissions, due dates and presentations.

3.3 Tools and Hardware

Tools and hardware that will be used in this project are as follows:

- Physiological Monitor Device
- Oxygen Tank Pressure Level Data
- Android Smartphone

Software used to build the android platform are as follows:

- Android Studio
- Java Eclipse

3.4 System design

The concept of the real-time assessment for the fire-fighters is that the during a fire rescue operation fire-fighters are not only able to communicate with the commander outside the operation area, they are also unable to share their physiological status such as body temperature, heart rate signal and together with oxygen pressure tank level. Hence, during a rescue mission it is critical for the fire-fighters to be monitored at real-time. Figure 3.2 shows the concept of communication for the real-time monitoring system.

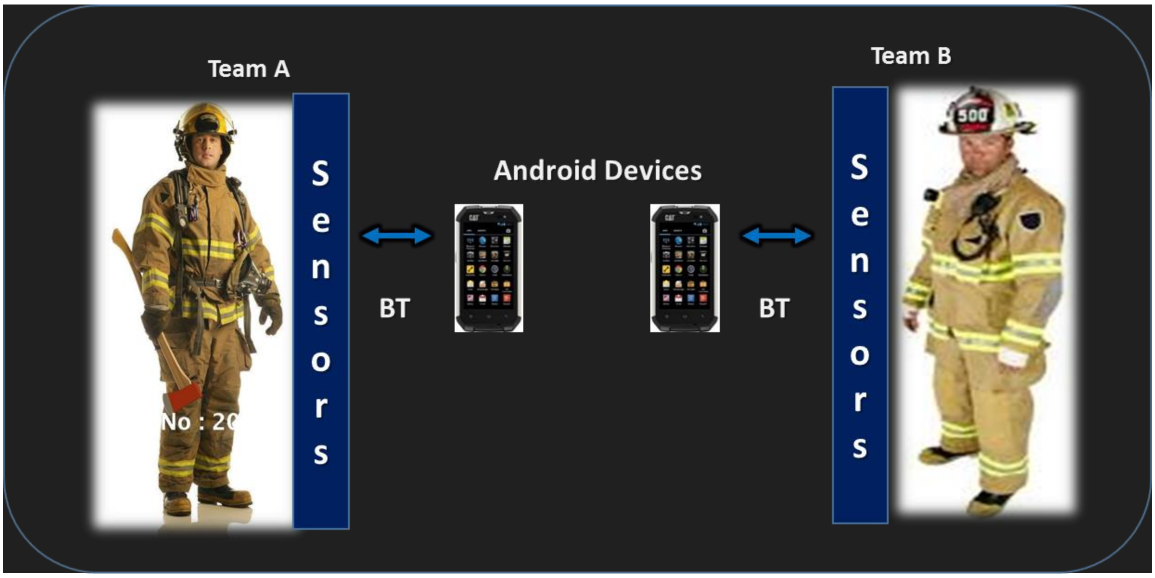

Figure 3.2 Concept of Communication

Based on concept of communication design as illustrated in Figure 3.2, the fire-fighters will be wearing a physiological monitoring devices that detects the physiological states such as heart rate, respiratory rate, peak acceleration and posture. The data will then be transmitted to an android smartphone. The data is transmitted via Bluetooth since the sensors on fire-fighters and the device carried by them are near, hence Bluetooth is the best option. The device is programmed using Android Studio and Java Eclipse to receive the data from the physiological monitoring device at every 1 seconds. As the data is transmits to the device at real-time, the data for each physiological state are shown in the smartphone accordingly.

The Figure 3.3 shows the system block diagram of the project. The sensors from the physiological and oxygen level monitoring devices are integrated with Bluetooth transmitter. The data transmitted by the devices via Bluetooth will be then received by the android smartphone. The data received by android smartphone will be pulled by the android application developed for this and displays its output accordingly. Besides that, it has been programmed to do intelligent sampling whereby it will trigger the fire-fighter on his physiological conditions. Hence the fire-fighters will be able to monitor himself by viewing this android application.

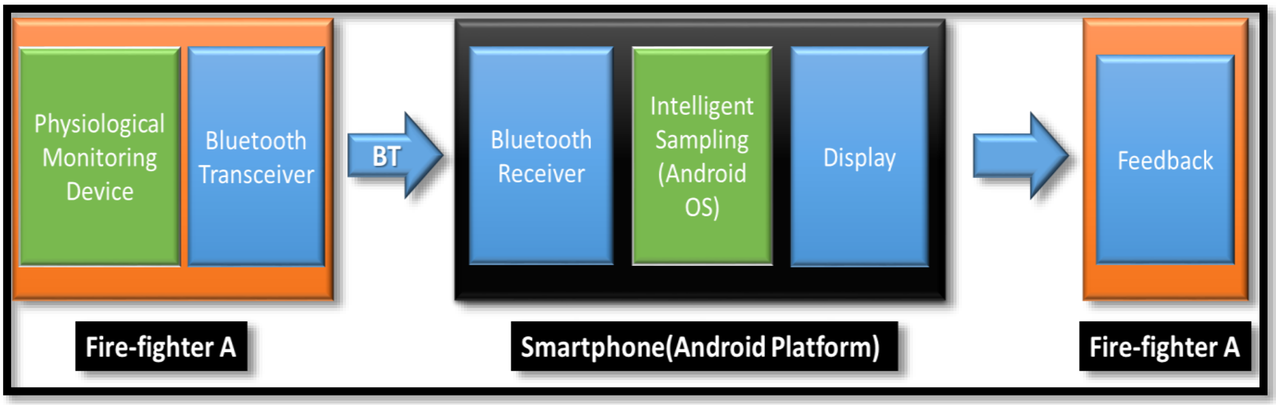

Figure 3.3 System Block Diagram

3.5 System Specification

Table 3.3 shows the parameters that is needed to be monitored from a fire-fighter during a fire and rescue and its specifications. The parameters listed are important to be monitored as it will help to save a fire-fighter's life.

Table 3.3 Parameters to be monitored

No.	**Parameters**	**Specifications**			
		Minimum	**Typical**	**Max**	**Unit**
1	Heart Rate	0	60-100	240	Beats per Minute (BPM)
2	Respiratory Rate	0	12-20	120	Breaths per Minute (BPM)
3	Posture	-180	0(Standing)	+180	Degrees (^{o})
4	Activity/ Acceleration	0	0(At Rest)	16	g-force (g) ; g=9.81m/s^2
5	Bluetooth Transmitter	2.4	-	2.835	Gigahertz (GHz)
6	Battery Voltage	3.5	3.9	4.2	Voltage(V)

From the Table 3.3, the physiological monitoring device is able to detect the heart rate ranging from 0 BPM to 240 BPM. The heart rate is measure in beats per minutes (BPM). The typical heart rate for a human being ranges between 60 BPM to 100 BPM at rest. The breathing rate ranges between 0 BPM to 120 BPM and measured in breaths per minute (BPM). The typical breathing for a human being ranges between 12 BPM to 20 BPM. The heart rate and breathing rate is depending on person's body size. Large sized people will tend to have higher breathing rate and heart rate.

The posture is measured in degrees by measuring the angle of inclination of the person. It ranges at $\pm 180^{o}$. A person at rest will have a typical posture at 0^{o}. For an example, as the person inclines forward, then the posture measurements will be more than 0^{o}, whereas if the person inclines backwards then the posture measurements will be less than 0^{o}. Next, the acceleration ranges from 0 g to 16 g measured in gravitational force, g which is multiplied by 9.81m/s^2. Typically it is zero acceleration for a person at rest, hence the gravitational force is 0g and it increases when the person is in motion.

All the data from the physiological monitoring device is transmitted via Bluetooth at a range of 2.4 GHz to 2.835 GHz. The battery life span of the physiological monitoring sensors are 24 hours in standby mode and 18 hours for active mode. The time taken for it to be fully charged is 3 hours. Furthermore it is water resistance and wash durability.

3.6 Summary

As a conclusion, the methodology of this project have been briefly explained on the concept of the project whereby the data from physiological will be transmitted via Bluetooth to the android based real-time monitoring system. The project methodology of this project have also been explained together with the Gantt chart for the entire project and tools used. Besides that the system specifications of the physiological monitoring device have also been discussed.

CHAPTER 4
Results and Discussion

In this chapter, the result of the overall project have been disclosed. Data gathered from this project have been discussed in this chapter.

4.1 Android Application

An android application have been developed as part of the android platform to facilitate in data transferring and receiving. The android app was developed using Android Studio and named as i-Rescue@UTP as shown in Figure 4.1. This android application will serve as the platform for the real-time monitoring system. It will receive the data from the physiological monitoring device and the physiological data will be displayed.

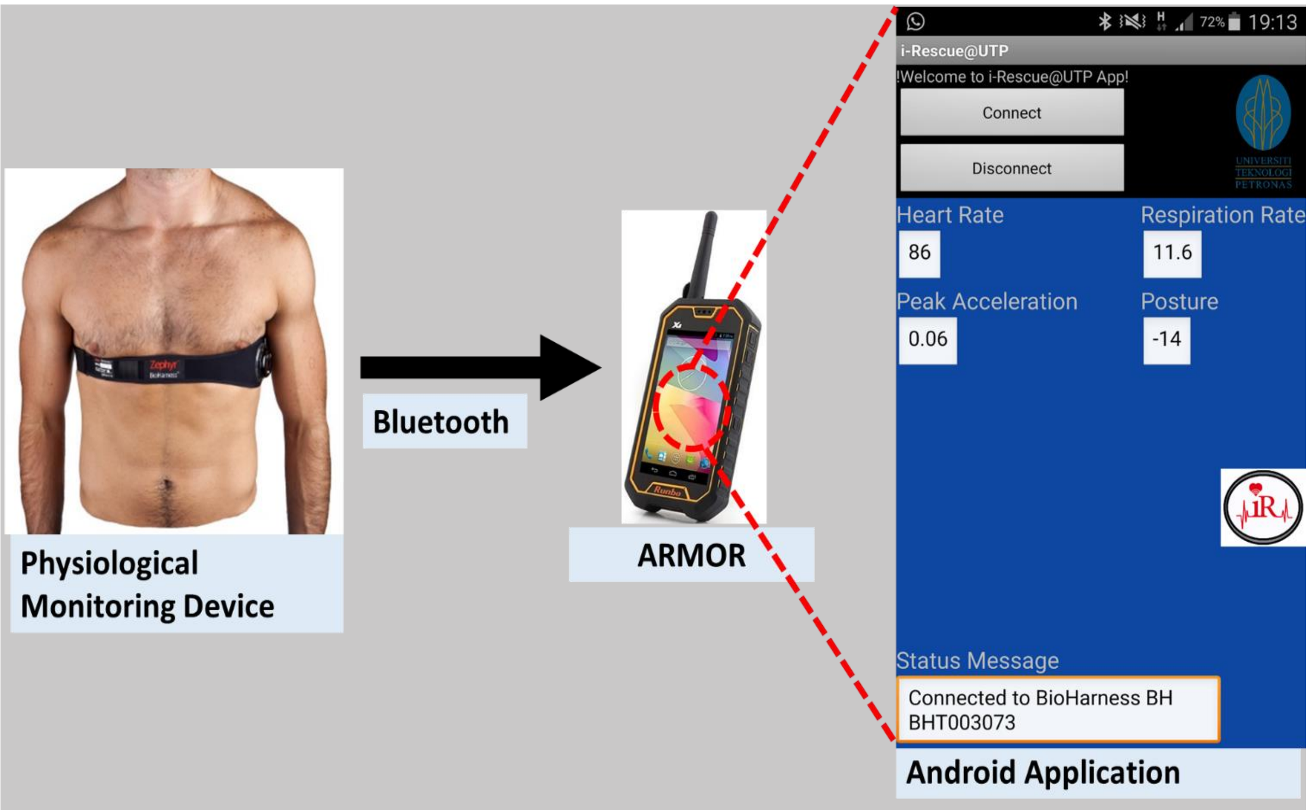

Figure 4.1 Android based Real-Time Monitoring (ARMOR)

Based on the Figure 4.1, at the top most left corner of the app, a text is printed "Welcome to iRescue@UTP App" welcoming the user to this application. At the bottom of the welcome text, there are two button labeled as "CONNECT" and "DISCONNECT" to assist the user in the connection of their smartphone to the physiological monitoring device. The data for each physiological state such as Heart Rate, Respiration Rate, Posture and Peak Acceleration will be displayed at the provided boxes once the Bluetooth connection is established. Last but not least, at the bottom most of the app there is a text box to display the status of the connection. This will assist the user to assure a successful connection.

4.2 Bluetooth Connection

The support for the Bluetooth network is provided by the Android platform, hence it permits access for a device to exchange data or information wirelessly [17]. It also allows a multipoint connection to take place. Figure 4.2 shows the flow diagram of the Bluetooth connection from the physiological monitoring device to the smartphone.

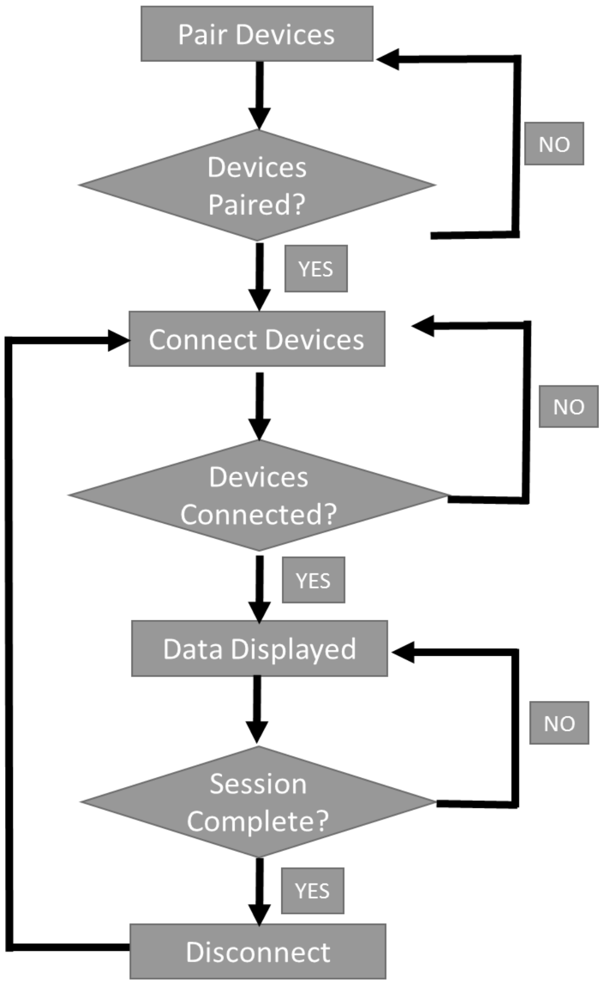

Figure 4.2 Flow Diagram of Bluetooth Connection

Figure 4.3 shows the coding written for the Bluetooth connection to occur. The Bluetooth physiological monitoring device has its own MacID. The i-Rescue@UTP application is programmed to be connected to that specific MacID, hence the data from the physiological device can be obtained accordingly. The MacID of the physiological monitoring device is “*E0: D7: BA: A7: FE: 9A”*.

```
Button btnConnect = (Button) findViewById(R.id.ButtonConnect);
if (btnConnect != null)
{
    btnConnect.setOnClickListener((v) → {
            String BhMacID = "E0:D7:BA:A7:FE:9A";
            //String BhMacID = "00:07:80:9D:8A:E8";
            //String BhMacID = "00:07:80:88:F6:BF";
            adapter = BluetoothAdapter.getDefaultAdapter();

            Set<BluetoothDevice> pairedDevices = adapter.getBondedDevices();

            if (pairedDevices.size() > 0)
            {
                for (BluetoothDevice device : pairedDevices)
                {
                    if (device.getName().startsWith("BH"))
                    {
                        BluetoothDevice btDevice = device;
                        BhMacID = btDevice.getAddress();
                        break;

                    }
                }
```

Figure 4.3 Coding for Bluetooth Connection

The Bluetooth connection will be connected once the physiological monitoring device is paired with the smartphone, this pairing is meant for security purpose. Hence, once paired, any other Bluetooth devices besides the physiological monitoring device will not be able to get connected to the physiological monitor device. The physiological monitoring devices works as one way terminal meaning once it gets paired and connected to one smartphone it will not get connected to other smartphones. This is helpful to prevent any confusion in securing the connection. In another words, only the user with the monitoring device will be able to use it without being interrupted by a third party.

Once the “CONNECT” button is tapped by the user, it indicates that the smartphone is ready to accept the data from the physiological monitoring device. This will allow the smartphone to receive the packet data from the monitoring device Bluetooth in which will be then stringed to the i-Rescue@UTP application. The user may tap on to the “DISCONNECT” button to end the data transmission and disconnect the monitoring device from the smartphone. This indicates the end of transmission of data.

Once the physiological monitoring device is paired and connected to the smartphone it will display at the status whether it has been connected or not. Figure 4.4 shows the coding for the status message to be displayed in the application. As soon as the smartphone is paired with the monitoring device and the “CONNECT” button is tapped then status will appear as “Connected to BioHarness” along with its device name indicating that the both of the devices have connected successfully. During any unfortunate event whereby the connection is lost or unable to connect to the monitoring device then the status will display “Unable to Connect” indicating that is an error in connection or in pairing between the smartphone and the monitoring device.

```
if(_bt.IsConnected())
{
    _bt.start();
    TextView tv = (TextView) findViewById(R.id.labelStatusMsg);
    String ErrorText  = "Connected to BioHarness "+DeviceName;
     tv.setText(ErrorText);

     //Reset all the values to 0s

}
else
{
    TextView tv = (TextView) findViewById(R.id.labelStatusMsg);
    String ErrorText  = "Unable to Connect !";
     tv.setText(ErrorText);
}
```

Figure 4.4 Connection Status

4.3 Receiving Data

This section will discuss on the route of the data transmission from the physiological monitoring device to the smartphone application which is the i-Rescue@UTP. Data transmitted by the physiological device is through packet data. Hence the header of the packet data should be identified as it is crucial to recognize the hexadecimal value for each physiological state so that the can be programmed to receive the data in orderly manner. Table 4.1 shows the hexadecimal value for the all five physiological state which are monitored. Since the data received from the physiological monitoring is in packet data. These values will be used to string the data of the specific physiological state to the app to display the reading accordingly. The data transmitted is in real-time and it transmits every 1 second.

Table 4.1 Hexadecimal Function

Physiological State	**Hexadecimal**
Heart Rate	0x100
Respiration Rate	0x101
Posture	0x102
Peak Acceleration	0x103

Figure 4.5 shows the coding for the physiological data that is converted from hexadecimal to strings.

```
final  Handler Newhandler = new Handler(){
    public void handleMessage(Message msg)
    {
        TextView tv;
        switch (msg.what)
        {
        case HEART_RATE:
            String HeartRatetext = msg.getData().getString("HeartRate");
            tv = (EditText)findViewById(R.id.labelHeartRate);
            System.out.println("Heart Rate Info is "+ HeartRatetext);
            if (tv != null)tv.setText(HeartRatetext);
        break;

        case RESPIRATION_RATE:
            String RespirationRatetext = msg.getData().getString("RespirationRate");
            tv = (EditText)findViewById(R.id.labelRespRate);
            if (tv != null)tv.setText(RespirationRatetext);

        break;
```

Figure 4.5 Data String

After getting the string for each physiological state, it is then stringed to the text box which will appear as the display in the app. The GUI of the app is programmed accordingly so that it can display the reading for each physiological state individually. Figure 4.6 shows the coding for the i-Rescue@UTP App GUI which have the strings written along. This ensure that the data for each physiological state receives are directed in the right text box accordingly. The id indicates the identification of the particular text in this case is for the heart rate. The text calls upon the string of heart rate in which the reading of the heart rate will be displayed. The rest of the coding for the android application is attached in Appendix I.

```
<TextView
    android:id = "@+id/HRTextBox"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Heart Rate"
    android:textSize = "20sp"
    android:layout_above="@+id/labelRespRate"
    android:layout_alignParentLeft="true"
    android:layout_alignRight="@+id/labelStatusMsg" />
```

Figure 4.6 GUI

4.4 Intelligent Monitoring

The android based real-time monitoring system does not only monitors and displays the value of the physiological states of the fire-fighter. It also sends feedback to trigger the fire-fighter and his teammates during any unfortunate events.

4.4.1 Faint Alert and Teammate Trigger

Figure 4.7 shows the flow diagram of the faint alert and teammate heart trigger.

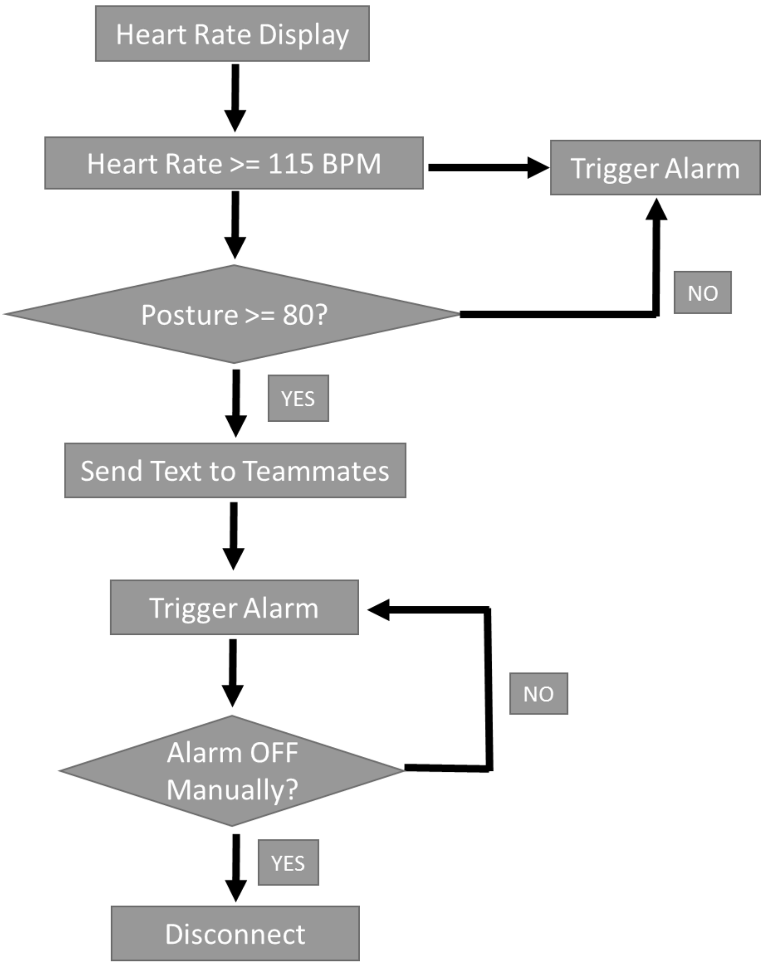

Figure 4.7 Flow Diagram for Faint Alert and Teammate Trigger

Based on Figure 4.7, as the heart rate crosses more the 115BPM it will begin to the trigger the alarm to warn the fire-fighter about his condition. If the posture of the fire-fighter is more than 80^0, it means that the fire-fighter's body is lying almost flat on the ground. Then it will immediately send SMS to his team member to alert that he have fainted. Figure 4.8 shows the alert system. Figure 4.8 shows the alert system which was successfully done.

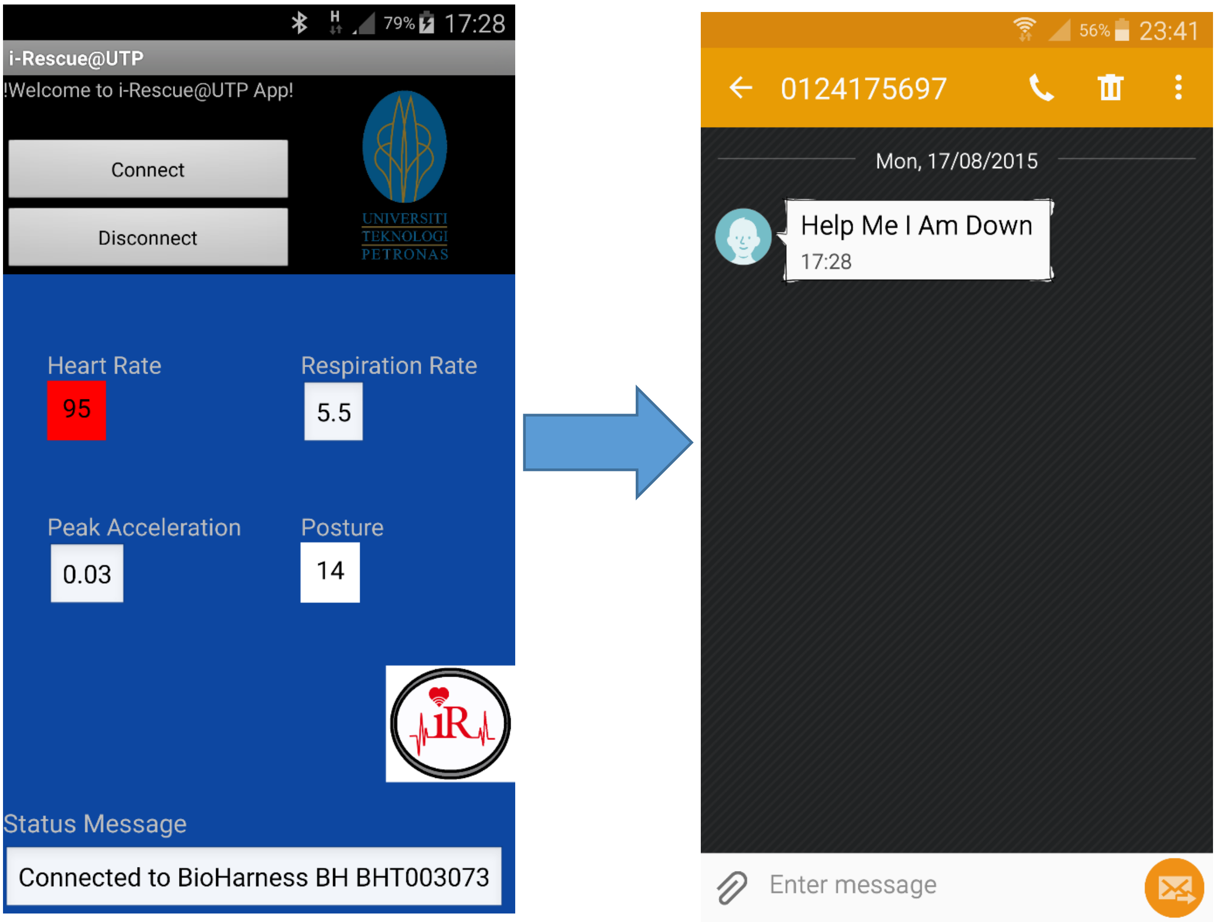

Figure 4.8 Alert System

4.4.3 Overexertion Warning

Figure 4.9 shows the flow diagram for overexertion warning whereby the fire-fighter will be warned if his heart rate and respiratory rate is too high. It alerts him to stay calm.

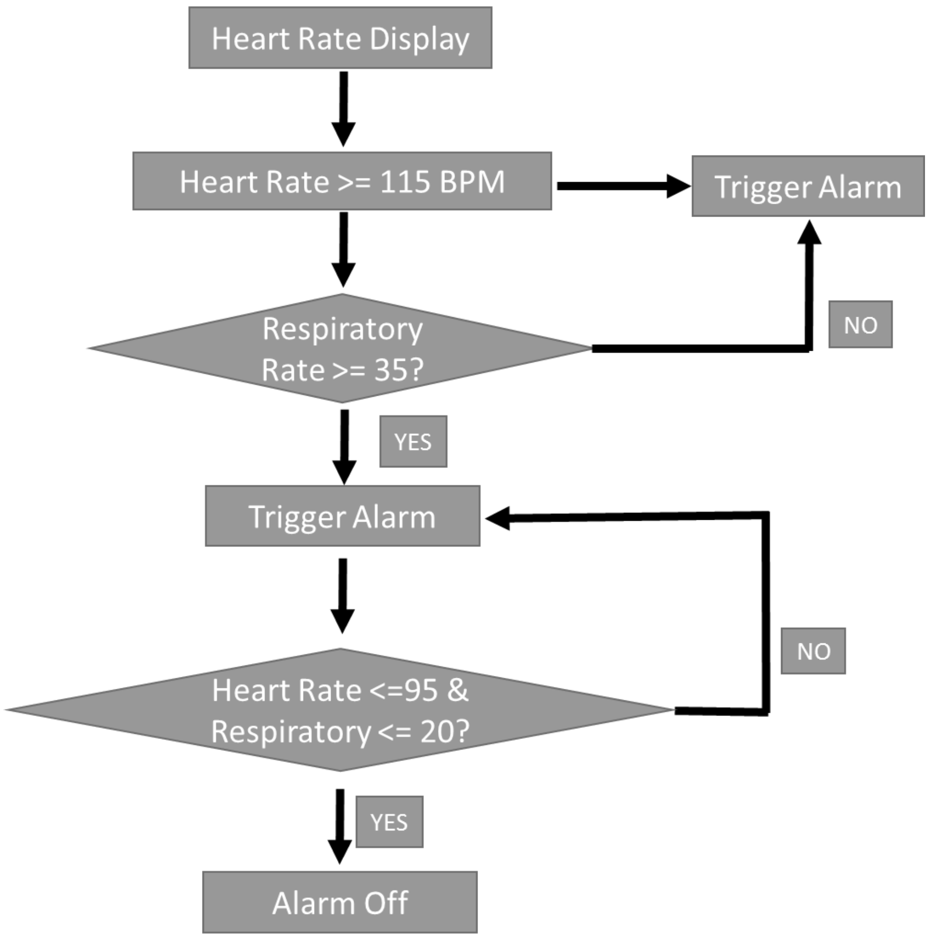

Figure 4.9 Flow Diagram Overexertion Warning

Based on Figure 4.9, it shows that when the firefighter’s heart rate increases more than 115BPM it will trigger the alarm. Adding on, if the respiratory rate is above 35 BPM then it will continuously trigger the alarm until the respiratory rate and heart rate reduces from the specified reading.

CHAPTER 5
Conclusion and Recommendation

5.1 Conclusion

In a nutshell, an android application was successfully created as part of the android platform for the smartphone to cater the data received from the physiological monitoring devices. The physiological data transmitted from the monitoring device have been successfully received by the android platform. The android applications is able to interpret and display the value for each physiological state individually such as the Heart Rate, Respiration Rate, Posture and Peak Acceleration. An android platform was successfully created with a friendly GUI to enable a real-time assessment to be done on physiological monitoring for the fire-fighters. The objective of this project was achieved.

5.2 Recommendation

As for future recommendation, the physiological data transfer via Bluetooth from the fire-fighters to the Android application will be programmed so that the physiological data will be sent via Wi-Fi to a tough pad held by the commander. This is to make sure that the commander is able to monitor the fire-fighters as well. The commander will also be able to trigger the fire-fighter according to their physiological conditions. Besides that, the oxygen pressure tank level data will also to be included in the monitoring system as well. This will help the fire-fighters to monitor their oxygen level in the tank as well as the commander will also be alert on the oxygen level in the tank for each fire-fighters.

References

[1] "Centers for Disease Contol and Prevention," 22 April 2013. [Online]. Available: http://wwwn.cdc.gov/wisards/fffmap/. [Accessed February 2015].

[2] F. R.F, L. P.R and M. J.L., "National Fire Protection Association," June 2014. [Online]. Available: http://www.nfpa.org/~/media/Files/Research/NFPA%20reports/Fire%20service%20statistics/osfff.pdf.

[3] "Ensco," ENSCO, Inc., 2015. [Online]. Available: http://www.ensco.com/products-services/gps-denied-geolocation-navigation/personnel-navigation.htm. [Accessed February 2015].

[4] *Using Wirelessly-Connected Monitoring Equipment,* Texas: RAE System,Inc, 2011.

[5] C. J.S and Z. M, "Recent Advance in Wireless Sensor Networks for Health Monitoring," *International Journal of Intelligent Control System,* vol. 15, no. 4, pp. 49-58, 2010.

[6] A. Szpak, "Types of Communication Technology," 2014. [Online]. [Accessed 17 March 2015].

[7] Q. Zhao, "Design of Physiological Parameter Acquisition," vol. 19, pp. 348-351, 2008.

[8] R. P and P. M.P, "Wireless Sensor Network for Continuos Monitoring a Patient's Physiological Conditions Using Zigbee," *Computer and Information Science,* vol. 4, no. 5, pp. 104-110, 2011.

[9] P. P.S., "Wireless Sensor Network for Wearable Physiological Monitoring," *Journal of Networks,* vol. 3, no. 5, pp. 21-29, 2008.

[10] P. S, P. H, B. P, C. L and a. R. M., "A review of wearable sensors and systems with application in rehabilitation," *Journal of Neuroengineering and Rehabilation ,* vol. 9, no. 21, pp. 1-17, 2012.

[11] L. J.S, H. S.Y, P. K.W and a. L. S.H, "A Physiological Signal Monitoring System Based on an SoC Platform and Wireless Network Technologies in Homecare Technology," *Journal of Medical and Biological Engineering,* vol. 1, no. 29, pp. 47-51, 2009.

[12] H. B. P. a. R. G. Ramamurthy, "Wireless Internet for the Mobile Enterprise Consortium," *Wireless Industrial Monitoring and Control using a Smart Sensor Platform,* p. 16, 2007.

[13] "Motorola Solutions," Motorola, 2015. [Online]. Available: http://www.motorolasolutions.com.

[14] "British Apco Journal," 05 March 2008. [Online]. Available: http://www.bapcojournal.com/. [Accessed 16 February 2015].

[15] K. Ammons, Writer, *Pros and Cons of TETRA vs. P25 and the Benefits of a Multi-technology Platform for TETRA, P25 Phase I / Phase II, and Mobile WiMax.* [Performance]. Power Trunk, 2014.

[16] M. T. Solutions, "Interconnecting TETRA systems," Motorola, Ltd., Hampshire, 2009.

[17] Android, "Android Developers," [Online]. Available: http://developer.android.com/index.html. [Accessed 20 06 2016].

[18] M. L. a. J. L. Yao Wang, "Application of Android Mobile Platform in Remote Medical," *nternational Journal of Smart Home,* vol. 9, no. 4, pp. 163-174, 2015.

[19] V. G. a. V. Ghodke, "Android Smartphone Based Health Monitoring System," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 2, no. 6, pp. 1721-1725, 2014.

[20] P. Strauss, "Technabob," Awesomer Media, 12 March 2013. [Online]. Available: http://technabob.com/blog/2013/03/12/runbo-x3-rugged-android-phone/. [Accessed 20 June 2015].

[21] Runbo, "Runbo Official Website," Tintele , 2015. [Online]. Available: http://en.runbo.net/about.php?id=10. [Accessed 19 June 2015].

[22] P. J.S., 01 April 2010. [Online]. Available: http://www.fireengineering.com/articles/print/volume-163/issue-4/Features/staffing-and-tactics-for-firefighter-survival.html.

[23] H. R, P. B.S and a. R. G, "Wireless Industrial Monitoring and Control using a Smart Sensor Platform," *IEEE Sensors Journal,* 2007.

[24] Adrian, "Android Authority," Android Authority, 16 January 2013. [Online]. Available: http://www.androidauthority.com/runbo-x3-x5-cheap-tough-android-4-0-smartphones-147941/. [Accessed 19 June 2015].

Appendix 1

Main Activity

```
package com.NewApp;

import android.app.Activity;

import android.graphics.Color;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.media.Ringtone;
import android.media.ToneGenerator;
import android.os.Bundle;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.util.Set;

import android.R.*;
import android.app.Activity;
import android.bluetooth.*;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.Vibrator;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;
import zephyr.android.BioHarnessBT.*;
import android.media.MediaPlayer;
import android.media.AudioManager;
import android.telephony.SmsManager;


public class MainActivity extends Activity {
   MediaPlayer beep;


   @Override
   protected void onStop() {
      super.onStop();
      beep.release();
   }





   /** Called when the activity is first created. */
   BluetoothAdapter adapter = null;
   BTClient _bt;
   ZephyrProtocol _protocol;
   NewConnectedListener _NConnListener;
   private final int HEART_RATE = 0x100;
   private final int RESPIRATION_RATE = 0x101;
   private final int SKIN_TEMPERATURE = 0x102;
```

```
   private final int POSTURE = 0x103;
   private final int PEAK_ACCLERATION = 0x104;
   private EditText editText;
   private boolean SMSFlag;



    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
      beep = MediaPlayer.create(this, R.raw.beepcen);
      SMSFlag = false;
        /*Sending a message to android that we are going to initiate a pairing 
request*/
        IntentFilter filter = new 
IntentFilter("android.bluetooth.device.action.PAIRING_REQUEST");
        /*Registering a new BTBroadcast receiver from the Main Activity context 
with pairing request event*/
       this.getApplicationContext().registerReceiver(new BTBroadcastReceiver(), 
filter);
        // Registering the BTBondReceiver in the application that the status of the 
receiver has changed to Paired
        IntentFilter filter2 = new 
IntentFilter("android.bluetooth.device.action.BOND_STATE_CHANGED");
       this.getApplicationContext().registerReceiver(new BTBondReceiver(), 
filter2);



      //Obtaining the handle to act on the CONNECT button
        TextView tv = (TextView) findViewById(R.id.labelStatusMsg);
      String ErrorText  = "Not Connected to BioHarness !";
       tv.setText(ErrorText);

        Button btnConnect = (Button) findViewById(R.id.ButtonConnect);
        if (btnConnect != null)
        {
           btnConnect.setOnClickListener(new OnClickListener() {
              public void onClick(View v) {
                 String BhMacID = "E0:D7:BA:A7:FE:9A";

                 adapter = BluetoothAdapter.getDefaultAdapter();

                 Set<BluetoothDevice> pairedDevices = adapter.getBondedDevices();

                 if (pairedDevices.size() > 0)
                 {
                        for (BluetoothDevice device : pairedDevices)
                        {
                           if (device.getName().startsWith("BH"))
                           {
                              BluetoothDevice btDevice = device;
                              BhMacID = btDevice.getAddress();
                                break;

                           }
                        }


                 }

                 //BhMacID = btDevice.getAddress();
                 BluetoothDevice Device = adapter.getRemoteDevice(BhMacID);
                 String DeviceName = Device.getName();
                 _bt = new BTClient(adapter, BhMacID);
                 _NConnListener = new NewConnectedListener(Newhandler,Newhandler);
                 _bt.addConnectedEventListener(_NConnListener);
```

```
            TextView tv1 = (TextView)findViewById(R.id.labelHeartRate);
              tv1.setText("000");


               tv1 = (TextView)findViewById(R.id.labelRespRate);
               tv1.setText("0.0");


               tv1 =     (TextView)findViewById(R.id.labelPosture);
               tv1.setText("000");

               tv1 =     (TextView)findViewById(R.id.labelPeakAcc);
               tv1.setText("0.0");
            if(_bt.IsConnected())
              {
                 _bt.start();
                 TextView tv = (TextView) findViewById(R.id.labelStatusMsg);
                 String ErrorText  = "Connected to BioHarness "+DeviceName;
                tv.setText(ErrorText);

                //Reset all the values to 0s

              }
              else
              {
                 TextView tv = (TextView) findViewById(R.id.labelStatusMsg);
                 String ErrorText  = "Unable to Connect !";
                tv.setText(ErrorText);
               tv.setTextColor(Color.RED);
              }
           }
        });
     }
     /*Obtaining the handle to act on the DISCONNECT button*/
     Button btnDisconnect = (Button) findViewById(R.id.ButtonDisconnect);
     if (btnDisconnect != null)
     {
        btnDisconnect.setOnClickListener(new OnClickListener() {
         @Override
         /*Functionality to act if the button DISCONNECT is touched*/
         public void onClick(View v) {
            // TODO Auto-generated method stub
            /*Reset the global variables*/
            TextView tv = (TextView) findViewById(R.id.labelStatusMsg);
             String ErrorText  = "Disconnected from BioHarness!";
             tv.setText(ErrorText);

            /*This disconnects listener from acting on received messages*/
            _bt.removeConnectedEventListener(_NConnListener);
            /*Close the communication with the device & throw an exception if
failure*/
            _bt.Close();

         }
        });
     }
  }
  private class BTBondReceiver extends BroadcastReceiver {
     @Override
     public void onReceive(Context context, Intent intent) {
        Bundle b = intent.getExtras();
        BluetoothDevice device =
adapter.getRemoteDevice(b.get("android.bluetooth.device.extra.DEVICE").toString());
        Log.d("Bond state", "BOND_STATED = " + device.getBondState());
     }
  }
  private class BTBroadcastReceiver extends BroadcastReceiver {
```

```
    @Override
    public void onReceive(Context context, Intent intent) {
       Log.d("BTIntent", intent.getAction());
       Bundle b = intent.getExtras();
       Log.d("BTIntent",
b.get("android.bluetooth.device.extra.DEVICE").toString());
       Log.d("BTIntent",
b.get("android.bluetooth.device.extra.PAIRING_VARIANT").toString());
       try {
          BluetoothDevice device =
adapter.getRemoteDevice(b.get("android.bluetooth.device.extra.DEVICE").toString());
          Method m = BluetoothDevice.class.getMethod("convertPinToBytes", new
Class[] {String.class} );
          byte[] pin = (byte[])m.invoke(device, "1234");
          m = device.getClass().getMethod("setPin", new Class []
{pin.getClass()});
          Object result = m.invoke(device, pin);
          Log.d("BTTest", result.toString());
       } catch (SecurityException e1) {
          // TODO Auto-generated catch block
          e1.printStackTrace();
       } catch (NoSuchMethodException e1) {
          // TODO Auto-generated catch block
          e1.printStackTrace();
       } catch (IllegalArgumentException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
       } catch (IllegalAccessException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
       } catch (InvocationTargetException e) {
          // TODO Auto-generated catch block
          e.printStackTrace();
       }
    }
  }


  final  Handler Newhandler = new Handler(){
     public void handleMessage(Message msg)
     {


        TextView tv;
        switch (msg.what)
        {
        case HEART_RATE:
           String HeartRatetext = msg.getData().getString("HeartRate");
           tv = (TextView)findViewById(R.id.labelHeartRate);
           System.out.println("Heart Rate Info is "+ HeartRatetext);
           if (tv != null) {
             tv.setText(HeartRatetext);

             if (Integer.parseInt(HeartRatetext) >= 85 )
             {

                tv.setText(HeartRatetext);
                tv.setBackgroundColor(Color.RED);
                tv.setTextColor(Color.BLACK);
                beep.start();
                if(!SMSFlag){
                   String phoneNo = ("0124175697");
                   String message = ("Help Me I Am Down");
                   SmsManager smsManager = SmsManager.getDefault();
                   smsManager.sendTextMessage(phoneNo, null, message, null,
null);
                   SMSFlag = true;
                }
```

```
                }
                else
                {
                    tv.setText(HeartRatetext);
                    tv.setBackgroundColor(Color.WHITE);
                    tv.setTextColor(Color.BLACK);
                    SMSFlag = false;


                }
            }


          break;

          case RESPIRATION_RATE:
              String RespirationRatetext =
msg.getData().getString("RespirationRate");
              tv = (TextView)findViewById(R.id.labelRespRate);
              if (tv != null)tv.setText(RespirationRatetext);

          break;

          case POSTURE:
              String PostureText = msg.getData().getString("Posture");
              tv = (TextView)findViewById(R.id.labelPosture);
              if (tv != null)tv.setText(PostureText);

            if (Integer.parseInt(PostureText) >= 70){
                //tv.setTextColor(color.darker_gray);
                //tv.setText(HeartRatetext);
                tv.setText(PostureText);
                tv.setBackgroundColor(Color.RED);
                tv.setTextColor(Color.BLACK);
                if(!SMSFlag){
                    String phoneNo = ("01126587716");
                    String message = ("Help Me I Am Down");
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phoneNo, null, message, null, null);
                    SMSFlag = true;
                }
            }

            else {
                //tv.setTextColor(color.black);
                //tv.setText(HeartRatetext);
                tv.setText(PostureText);
                tv.setBackgroundColor(Color.WHITE);
                tv.setTextColor(Color.BLACK);
                if(Integer.parseInt(PostureText) <= 60)
                    SMSFlag = false;
            }

          break;

          case PEAK_ACCLERATION:
              String PeakAccText = msg.getData().getString("PeakAcceleration");
              tv = (TextView)findViewById(R.id.labelPeakAcc);
              if (tv != null)tv.setText(PeakAccText);

          break;

          }
      }

    };

}
```