OXYHEALTH MONITORING SYSTEM

By

NC Noman Waheed

NC Muhammad Zeeshan Sadiq

NC Muhammad Uzair Zubair

NC Abu Bakar Saadat

Submitted to the Faculty of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfilment for the requirement of a B.E Degree in Electrical Telecommunication Engineering

JUNE 2017

CERTIFICATE

It is hereby certified that the contents and form of the project entitled "Oxyhealth Monitoring System", submitted by the syndicate of

1) Noman Waheed
2) Muhammad Zeeshan Sadiq
3) Muhammad Uzair Zubair
4) Abu Bakar Saadat

has been found satisfactory as per the requirement of the B.E. Degree in Electrical (Telecom) Engineering.

Name (Supervisor): **<u>Asst. Prof. Dr. Mir Yasir Umair</u>**

Signature: ____________________________

Date: ____________________________

ABSTRACT

OXYHEALTH MONITORING SYSTEM

According to the latest WHO data published in May 2014, the deaths resulting from car accidents reached 30,310 or 2.69% of total deaths in Pakistan [1]. Beside other causes, one of the main causes of deaths is the delay in providing first-aid. So with this motivation we designed a system which helps to provide immediate first-aid in case of an accident or any severe condition. The Oxyhealth Monitoring System (OMS) provides continuous monitoring of health parameters of a car's driver such as saturation of oxygen in the blood (SpO2) and heart rate (HR) using optical bio-sensing chip. Therefore, in case of an accident the health parameters will be sent to the concerned helpline or any desired contact number via GSM module. Hence, a first-aid would be provided to the injured person by determining its location coordinates via GPS.

COPYRIGHT STATEMENT

DEDICATED TO

Allah Almighty,

Faculty/Supervisor for their Help,

And our parents for their support and blessings.

ACKNOWLEDGMENTS

We are without a doubt especially grateful to our respected supervisor Dr. Mir Yasir Umair for his support and supervision all through the length of our project. It is not often that one finds a supervisor that dependably finds the ideal opportunity for listening to the little issues and detours that unavoidably manifest over the span of performing project. His specialized and publication exhortation was basic to the consummation of this project and has shown us incalculable lessons and experiences on the workings of scholarly research in general.

Our parents get our most profound appreciation and love for their devotion and the numerous times of support amid our undergraduate studies that gave the foundation to this work. Without them it was an impossible task to complete.

TABLE OF CONTENTS

1. CHAPTER 1: Introduction..1

1.1 Overview……………..……………,………………………………………………….……..1

1.2 Problem Statement….………………………………………………………………...……1

1.3 Objectives..2

1.4 Approach 3

1.4.1 Wrist-band Wireless Device……………………….………………………………....4

1.4.2 Receiver Section..5

2. CHAPTER 2: Literature Review..6

2.1 Existing literature 6

2.2 Problem Formulation..8

3. CHAPTER 3: Design and Development...9

3.1 Project Design………………………………….……………………………………..9

3.1.1 Required Modules…………………………….…………………………………..9

3.1.1.1 Optical Bio-sensing Chip MAX30100………………….……………………………9

3.1.1.2 GSM Module SIM900D……………………………………………………...…...11

3.1.1.3 GPS Module NEO-6M…………………..……………………………………12

3.1.1.4 Bluetooth Module HC-05………………………………………………..……….…13

3.1.1.5 GY-521 3-Axis Accelerometer ……………………………………………..…...14

3.1.1.6 Arduino Mega 2560……………………………………………………………………15

3.2 Detailed Design………………………………………………………………………...16

3.2.1 Wrist-Band Wireless Device……………………………………………………16

3.2.2 Receiver Section……………………………………………………………..18

3.3 Methodologies and Algorithms……………………………………………………...19

3.3.1 Extraction of SpO2 Measurements…………………………………………….……19

3.3.2 Extraction of Heart Rate Measurement……...…………………………………………21

4. CHAPTER 4: Analysis and Evaluation...23

4.1 Hardware Configuration…………………………….…………………………………...23

4.1.1 Wrist-Band Wireless Device Circuit Diagram………………………………...23

4.1.2 Receiver Section Circuit Diagram……………………………………………24

4.1.3 Optical Sensing chip (Pulse Oximeter) and Accelerometer Circuitry…………..25

4.2 Graphical Simulation……………………………………………………………….25

4.3 SMS Alert Screenshot……………………………………………….………………...……26
5. CHAPTER 5: Conclusion..**27**
5.1 Achievements…………………………………………………………………….…...27
5.2 Applications………………………………..…………………………………….27
5.3 Suggestions for Future Work………………………………………….……….…28
6. CHAPTER 6: References..**29**
Appendix A……………...………………………………………………………………30
Appendix B 31
Appendix C...32

LIST OF FIGURES

Figure 1-01: Prototype of Wrist-Band Wireless Device……………………………...…...……4
Figure 1-01 Receiver Section for displaying and sending the data…………………………...5

Figure 2-01: Schematic diagram of light (IR and Red) absorbance by a pulse sensor…..….7
Figure 2-02: Typical resting PPG signal ……………………………………….………………8

Figure 3-01: Optical Sensing Module……………………………………….……………9
Figure 3-02: MAX30100 System Block Diagram……………………………….…………10
Figure 3-03: GSM Module SIM900D…………………………………………….…...……11
Figure 3-04: GPS Module NEO 6M…………………………………………….…………12
Figure 3-05: Bluetooth Module HC-05………………………...…………………………...13
Figure 3-06: GY-521 MPU6050……………………………………………………..………14
Figure 3-07: Label Diagram of Arduino Mega 2560……………………………………………15
Figure 3-08: Prototype of Wrist-Band Wireless Device……………………………...……17
Figure 3-09: Receiver Section for displaying and sending the data…………………………18
Figure 3-10: Typical SaO2 Calibration Curve……………………………………….……20
Figure 3-11: Absorption of oxy and deoxy haemoglobin in the R and IR regions…………22

Figure 4-01: Wrist-Worn Transmitter Section Circuit Diagram ………………………..……..23
Figure 4-02: Receiver Section Circuit Diagram ……………………………………….……..24
Figure 4-03: Pulse Oximeter and Accelerometer Circuitry ………………………..……..25
Figure 4-04: Graph of Pulse Sensor when not mounted on Wrist……………………..……25
Figure 4-04: Graph of Pulse Sensor when mounted on Wrist………………………………26
Figure 4-05: Sending Alert on Cell Number……………………………………………..…26

LIST OF TABLES

Table 3.1: Technical Specifications of Arduino Mega 2560……………………………………..…16

LIST OF ABBREVIATIONS:

OMS: Oxyhealth Monitoring System

WHO: World Health Organization

SpO2: Saturation of Peripheral Oxygen

HR: Heart Rate

PPG: Photoplethysmogram

ACC: Accelerometer

LMS: Least-Mean-Square

GSM: Global System for Mobile

GPS: Global Positioning System

O2Hb: Oxygenated Haemoglobin

HHb: Deoxygenated Haemoglobin

CHAPTER 1: Introduction

1.1 Overview

Oxyhealth Monitoring System is basically a wireless non-invasive medical device that enables us to monitor the health of a car's driver. The project is designed on Arduino board to enable parallel processing of the multiple inputs and is programmed using Arduino IDE. The target would be the monitoring of arterial oxygen saturation (SpO_2 level) and Heartrate (HR) using optical bio-sensing chip [2]. Finally, the location of a car's driver is detected via GPS and sends an SMS alert to the concerned helpline.

1.2 Problem Statement

From the last five years, there is a rapid increase in the research of biomedical field all over the world. Because the world demands more of health monitoring devices from engineering disciplines due to a large number of prevailing diseases worldwide. In this regard, OMS would be an affordable wearable remote physiological monitoring device for car drivers or the individuals who perform tasks in high-risk situations, such as soldiers and firemen, as well as individuals at remote mining. The use of this system will ensure rapid response and make the location of the individual more efficient. It has the following advantages:

i. In a war zone setting, this device would empower medics to evaluate the wellbeing status of warriors more rapidly, which would facilitate the conveyance of medicinal treatment as well as triage of injured people.

ii. Optical sensor signifies a promising stage for remote physiological checking since the readings of SpO_2 and HR can be extracted from the PPG signal, which can be acquired by a wearable optical sensor. These estimations would empower a medic to make more precise diagnostics for restorative treatment and triage of injured soldiers.

1.3 Objectives

The fundamental goal of this project is to build a hardware for monitoring the health status of a car's driver and provide immediate response to the concerned helpline along with the tracking of the vehicle in case of an accident. The wireless device is to be worn by the car's driver and its receiver section circuit board would be installed on a dashboard of a vehicle. The device would have military as well as civil applications. This device is implemented on Arduino board exploiting its multiple processing capabilities to monitor and locate the respective individual. The PPG (Photoplethysmogram) and SpO_2 differential algorithms would be implemented on bio-sensing chip. The main objectives are as below:

Objective 1

To operate an optical pulse sensor and test the precision using algorithm developed by Johnston and extract SpO_2 *and Heart-Rate from IR PPG signals [3].*

Objective 2

To test the accuracy of the accelerometer with respect to the falling motion of car's driver in case of an accident [4].

Objective 3

To develop a code for GSM and GPS module for sending SMS alert and detecting the location respectively.

Objective 4

To interface all the modules with Arduino board and final testing of the designed system.

1.4 Approach

The project basically comprises of two sections: Setting up wrist-band wireless device for measuring the health parameters as shown in fig.1-01 and the configuration of a receiver section, shown in fig 1-02, for displaying and sending the data to the respective health-care centre.

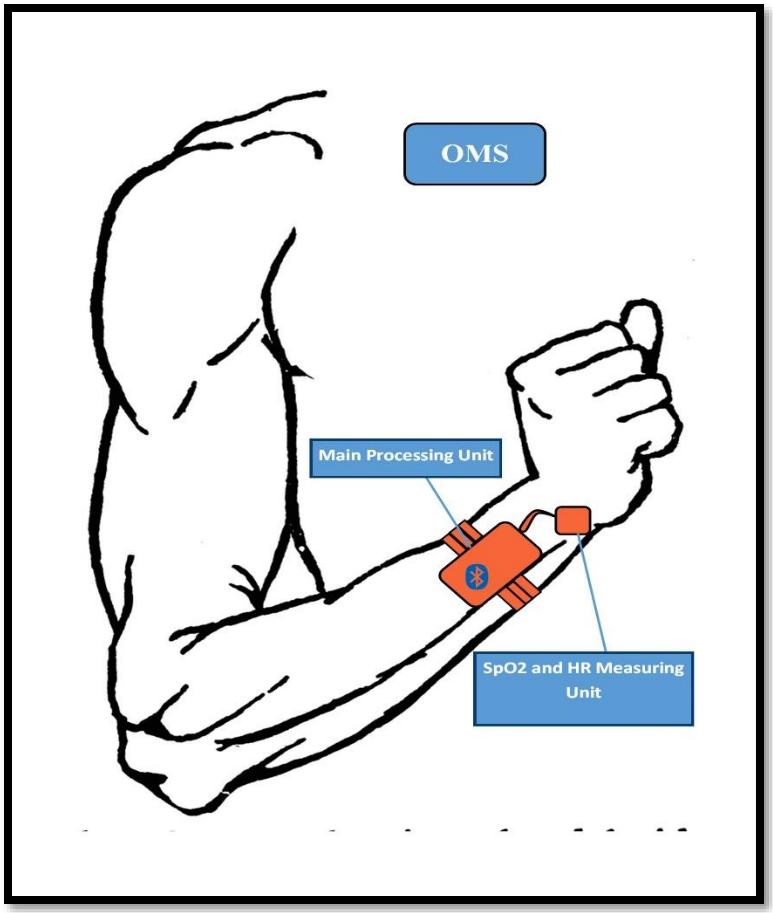


Figure 1-01: Prototype of Wrist-Band Wireless Device

1.4.1 Wrist-Band Wireless Device

This consists of an optical sensor, accelerometer sensor, Bluetooth module, Arduino Mega2560 and a battery.

Arduino Mega 2560 performs all the processing of the optical bio-sensing chip that monitors the health parameters of the car's driver non-invasively, and a Bluetooth module that sends the health status continuously to the receiver section along with the body orientation determined by the accelerometer sensor.

1.4.2 Receiver Section

Receiver part as shown in fig.1-02, uses Arduino mega 2560 for displaying and processing the data (SpO2 and Heart-rate) received via Bluetooth module and then sends the information containing the health-parameters and location of a car driver to the registered health-centre using GSM and GPS modules respectively.

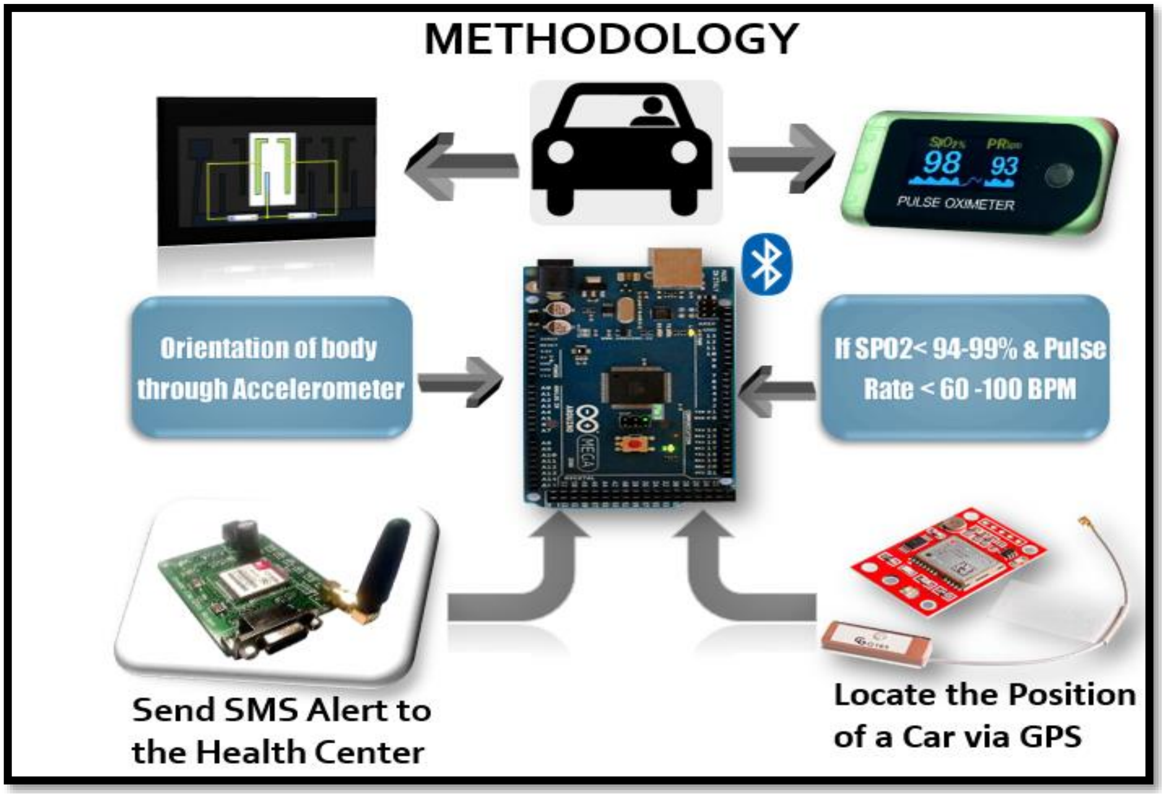


Figure 1-02 Receiver Section for displaying and sending the data

The above figure describes the functioning and interfacing of different modules with Arduino board. Pulse oximeter always monitors the SpO_2 level and HR of a person and sends the signal to Arduino if they do not meet the normal ranges. Hence, GSM will the alert SMS and GPS will send the location to the desired helpline.

CHAPTER 2: Literature Review

2.1 Existing literature

i. Johnston developed the algorithms for the measurement of SpO2 and HR.[3]

   a. The SpO_2 estimation in view of PPG signals demands a count of the normalized ratio of "R"(R/IR) i.e. absorption of red light by the blood over absorption of infrared light. An assortment of techniques found for getting the values utilized to compute R, specifically, the amplitudes of the AC parts of Red and IR light obtained via PPG signals. The amplitudes of individual heartbeats can be utilized to give exact AC values.

   b. The derivatives of PPG can be utilized to distinguish beat pinnacles and estimate the heart rate. PPG signal's derivative can be utilized to find the upward incline toward the start of a cardiovascular event. The consequent zero-intersection, showing an incline move, is utilized to decide the area of the pinnacle. This technique has the benefit of being steady and robust.

ii. Gary W. Comtois describes the SpO2 and HR extraction from signs that were prepared by a LMS calculation utilizing the acceleration signals of the body. This review was made to decide the probable impact of ACC axis selection on estimation precision. The information demonstrates that enhancements in SpO2 and HR were acquired by using every acceleration axis independently and in addition the summed of the accelerometer axes (X+Y+Z). [4]

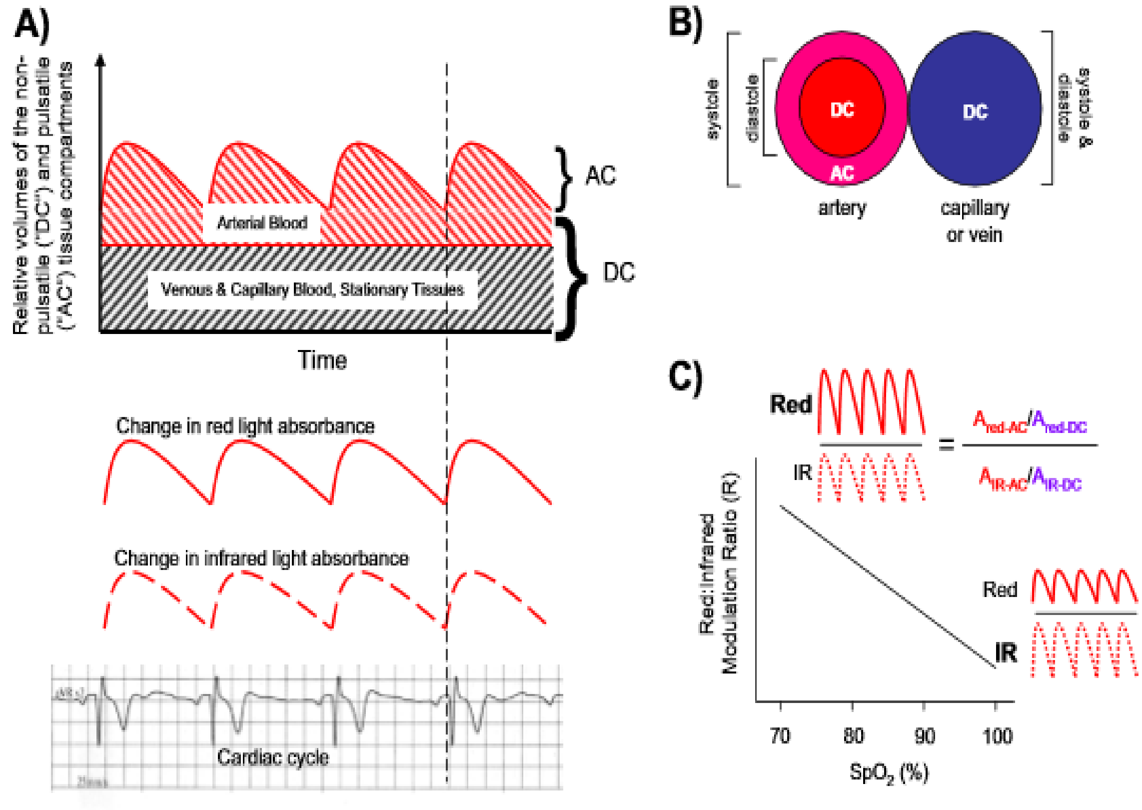


Figure 2-01: Schematic diagram of light (IR and Red) absorbance by a pulse sensor. (A) In a man with great cardiovascular function (B) A cross-sectional outline of an artery and a vein (C) An alignment (standard) curve of the Red: IR Modulation Ratio in connection to the SpO_2.

iii. Edward D. Chan, states that oxygenated haemoglobin (O2Hb) absorbs more near-infrared light than deoxygenated haemoglobin (HHb) and HHb assimilates more red light than O2Hb, which is the basic working principle of an optical sensor. Since optical pulse sensors are adjusted for SpO_2 in the vicinity of 70 and 100%. Shown values underneath 70% ought to just be considered subjectively exact and not quantitatively. Mistaken SpO_2 readings can happen with conditions that decrease blood vessel perfusion [5].

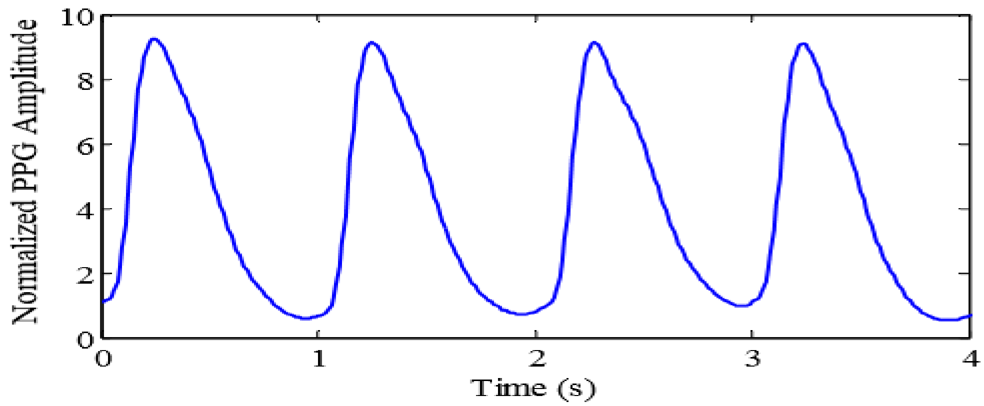


Figure 2-02: Typical resting PPG signal

2.2 Problem Formulation

We observed that most of the existing implementations make use of the Independent Component Analysis technique such as SpO_2 and HR analysis and accelerometer functionality. Substantial work has been done in the discipline of bio-medical engineering and isolation of health monitoring system of a particular source. Most of the implementations have been limited to the formation of only medical device using biomedical sensors. To make the health monitoring system worthwhile and real-time with better accuracy, our designed project make use of optical bio-sensing chip [2], GSM module, GPS module and accelerometer sensor. The hardware to be used is the Arduino kit and the hardware description language is 'Arduino IDE'. The number of optical bio-sensing chips would be increased to achieve better results of the IR and Red light passing through the skin. This project will be stand alone as well as be able to be mounted on a dashboard of a vehicle provided that pulse oximeter is attached to a person's body.

CHAPTER 3: Design and Development

3.1 Project Design

The project design is discussed as follows:

3.1.1 Required Modules

i. Optical Bio-sensing Chip MAX 30100

ii. GSM Module SIM900D

iii. GPS Module NEO-6M

iv. Bluetooth Module HC-05

v. GY-521 Accelerometer Sensor

vi. Arduino Mega 2560

3.1.1.1 Optical Bio-sensing Chip MAX30100

The MAX30100, shown in fig.3-01, is an incorporated heartbeat oximetry and heartrate sensor solution. It joins two LEDs, a photodetector, optimized optics,

Figure 3-01 Optical Sensing Module

and low-noise signal processing to recognize beat oximetry signals. The sensor is set on a fringe purpose of the body, for example, a fingertip, wrist, ear projection or the nose.

Working:

In our case we placed it on the wrist of the body. The sensor incorporates two LEDs, shown in fig 3-02, red light lies in the visible range (660 nm) and the other led lies in the infrared range (940 nm). The rate of oxygen in the body is computed by measuring the intensity from every recurrence of light after it transmits through the body and afterward figuring the proportion between these two intensities. SpO2 depends on measuring the light intensity that has been constricted by body tissue. For every wavelength of light, the DC value is expelled from the signal leaving the AC part, which reflects the oxygenation level

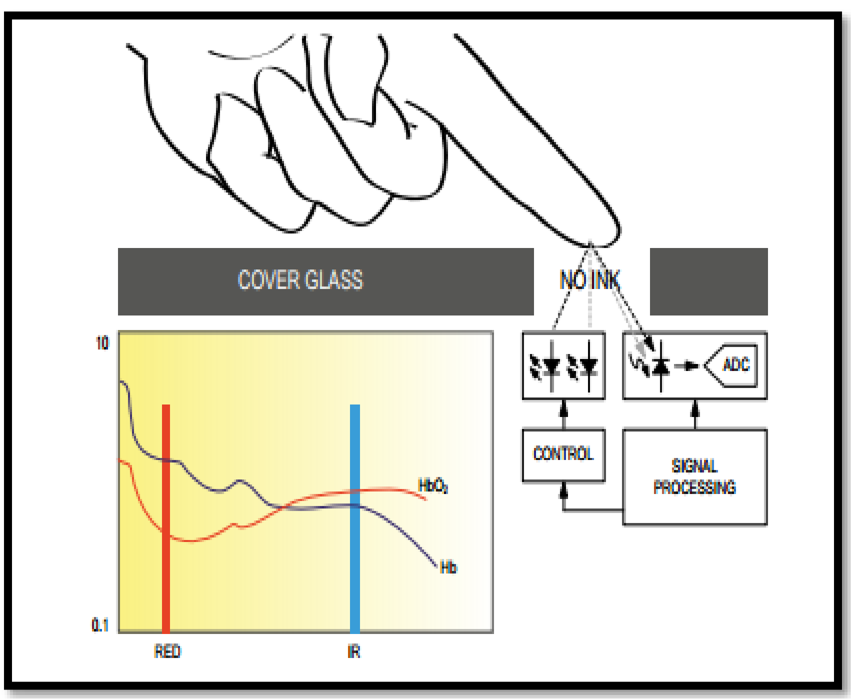


Figure 3-02 MAX30100 System Block Diagram

in the arteries. The RMS value is ascertained by averaging the square of the signal over various heart beat cycles (HR).

3.1.1.2 GSM Module SIM900D

SIM900D GSM module, shown in fig 3-03, is used for sending an SMS alert to the desired helpline or any family member’s number. SIM900D conveys different GSM/GPRS standards i.e. 850MHZ, 900MHz, 1800MHz and 1900MHz required for the execution of SMS, voice services, Data services and Fax in a little form factor and with low consumption of power . The dimensions of SIM900D are (33 x 33 x 3) mm, which fits all the space prerequisites in M2M applications, especially for small and compact demand of design. Further, the SIM900D has GPRS multisport class 10/8 (optional) and supports the various coding schemes which include CS-1, 2, 3 and 4.

Figure3-03 GSM Module SIM900D

3.1.1.3 GPS Module NEO-6M

The NEO-6 module, shown in fig. 3-04, brings the elite of the u-blox6 position feature to the smaller than expected NEO Inc. form factor. U-blox6 has been built with the reason for accomplishing low power utilization with lower costs. Employing keen power utilization is a leap forward for the applications that required low-power. The receivers join a high level integrated capacity with an adaptable availability choices in a little package. Further, the DDC interface gives availability and empowers collaborations with u-blox6 LEON and LISA remote module.

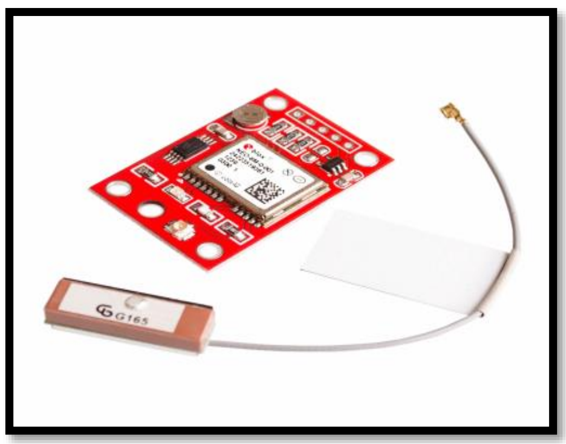


Figure 3-04 GPS Module NEO 6M

It has the following features:

- Under 1 second time-to-first-settle for hot and aided begins
- SuperSense ® Indoor GPS: - 162 dBm tracking sensitivity
- Anti-jamming technology
- Support SBAS (WAAS, EGNOS, MSAS, GAGAN)

• 50 channel situating engine with more than 2 million viable correlators

• 5Hz position refresh rate

• Operating temperature range: - 40 TO 85°C

• UART TTL attachment

• EEPROM to store settings

• Build in 18X18mm GPS antenna

3.1.1.4 Bluetooth Module HC-05

HC-05, shown in fig. 3-05, is a wireless Bluetooth innovation standard for trading information over short ranges (utilizing short-wavelength UHF waves in the ISM band from 2.4GHz to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). Range is around 10 Meters (30 feet).

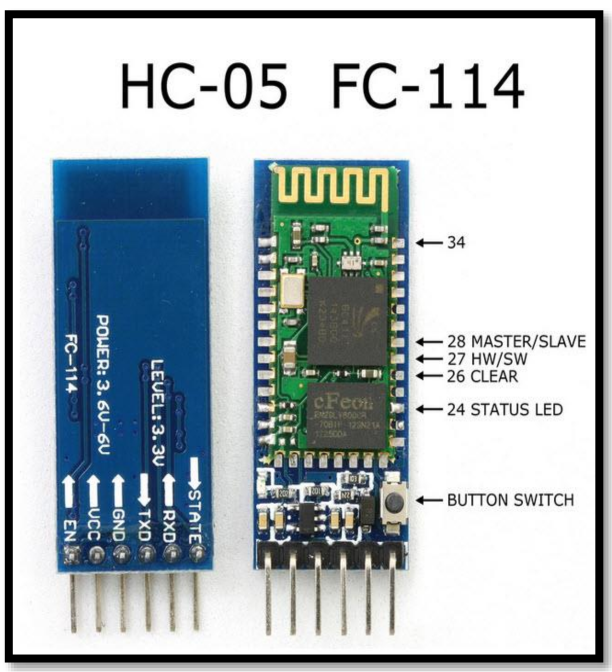


Fig 3-05 Bluetooth Module HC-05

Specifications:

- Up to +4dBm RF transmit control
- Sensitivity: - 80dBm
- With edge connector
- Default Baud rate: 38400, Data bits: 8, Stop bit: 1, Parity: No parity
- UART interface with programmable baud rate
- PIO control
- Low Power Operation: 1.8 to 3.6V
- With integrated antenna

3.1.1.5 GY-521 3-Axis Accelerometer

The GY-521, shown in fig. 3-06, contains both a 3-Axis accelerometer and a 3-Axis Gyroscope permitting estimations freely, however all based around similar axes, along these lines wiping out the issues of cross-axis errors when utilizing separate gadgets.

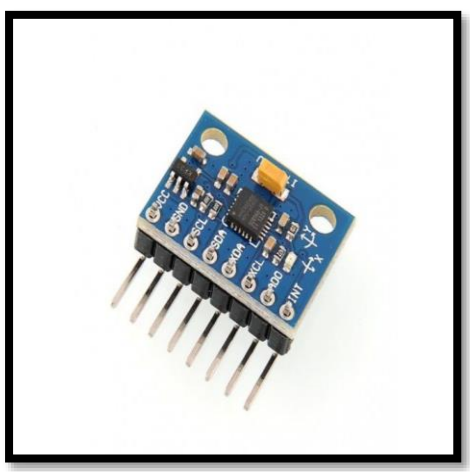


Figure 3-06 GY-521 MPU6050

Specifications:

- Gyroscope ranges are ± 250, 500, 1000 and 2000 °/s
- Accelerometer ranges are ±2, ±4, ±8 and ±16g
- Voltage range is 3.3V - 5V (the capability of incorporating a low drop-out voltage regulator)

3.1.1.6 Arduino Mega 2560

The Arduino Mega 2560 microcontroller toolkit is based on ATmega2560. Mega 2560 has 54 digital I/O pins (15pins can be utilized as PWM outputs), 16 simple sources of info, 4 UARTs ports (equipment serial ports), crystal oscillator of 16 MHz, a DC power jack, an ICSP header, a USB connection, and a reset button. It contains a battery port needed to support the microcontroller; which can be simply connected to a PC with a USB cable or can be controlled with an AC-to-DC adapter.

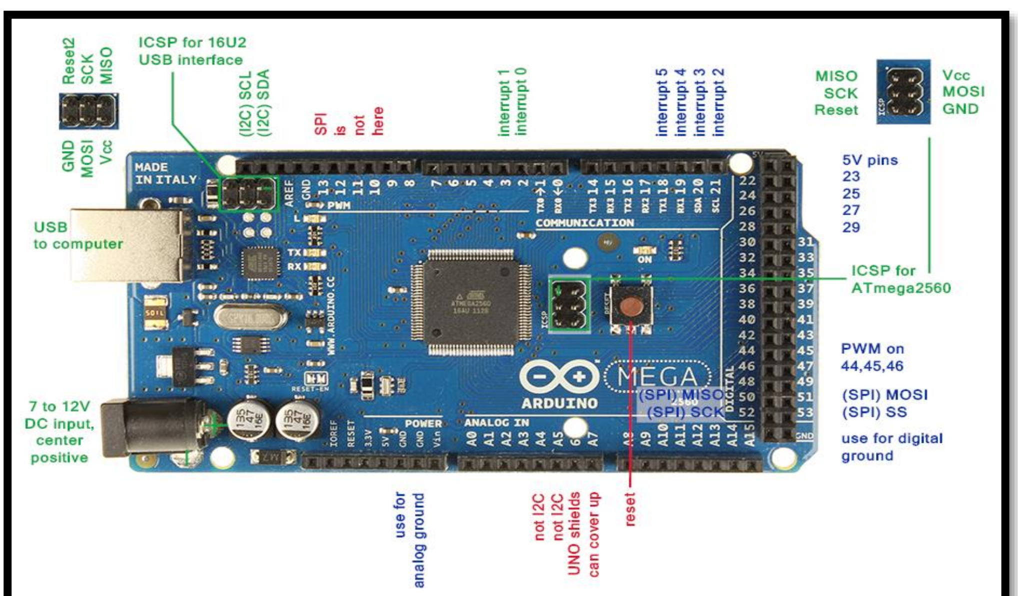


Figure 3-07 Label Diagram of Arduino Mega 2560

3.1.1.6.1 Technical Specifications

Arduino Mega 2560 technical specifications are shown in the below table:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Table 3.1.Technical Specifications of Arduino Mega 2560

3.2 Detailed Design

The project basically comprises of two parts:

3.2.1 Wrist-Band Wireless Device

Wrist-band wireless device, shown in fig 3-08, is used for measuring the health parameters. It consists of following components:

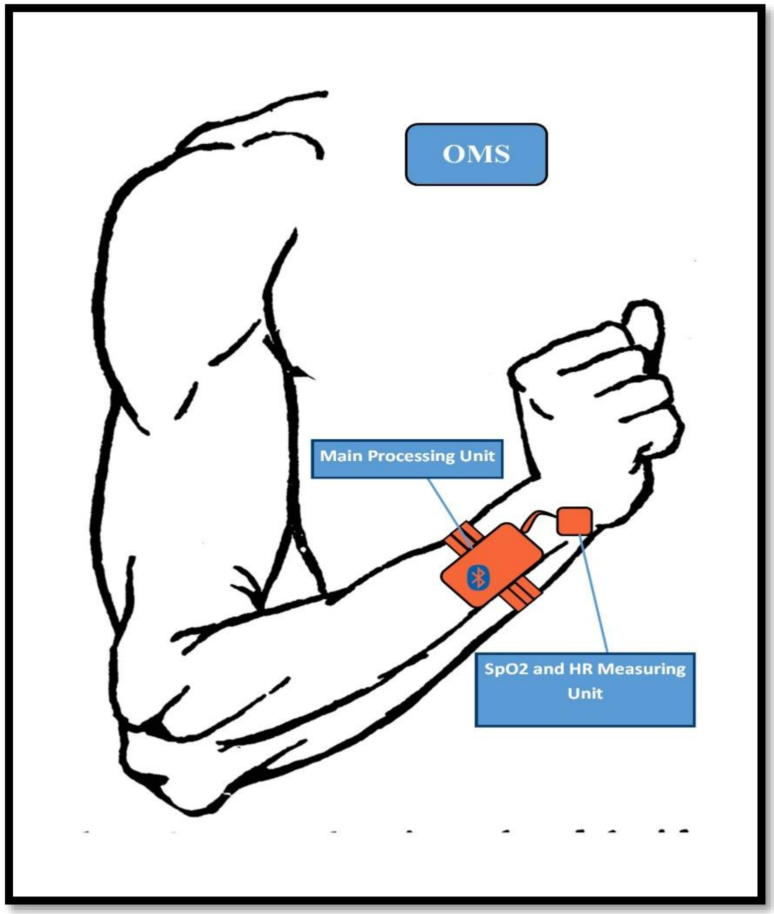


Figure 3-08 Prototype of Wrist-Band Wireless Device

i. **Optical Sensor**: The optical sensor module consists of the optical transducer that measures SpO_2 and Heart-rate based on calibration of PPG graph obtained via absorption of IR light and red light by O_2HB and HHB.

ii. **Accelerometer Sensor:** The orientation of a body is determined via ACC sensor for accurate measurement of health parameters with respect to the body movement.

iii. **Arduino Mega 2560:** All the modules of a wrist-worn oximeter are processed by Arduino Mega 2560.

iv. **Bluetooth Module:** The health status acquired by the sensor modules is transmitted to the receiver circuit wirelessly via Bluetooth module.

v. **Battery:** 9V battery is used as a power source of wireless transmitter section.

3.2.2 Receiver Section

Receiver part as shown in fig.3-09 consists of following components:

i. **Bluetooth Module:** Used for receiving the data (SpO2 and Heart-Rate) wirelessly from transmitter circuit that actually monitors the health parameters.

ii. **GSM Module:** GSM module is used for sending the alert SMS to the health-centre in case of an accident of any critical situation. Alert SMS

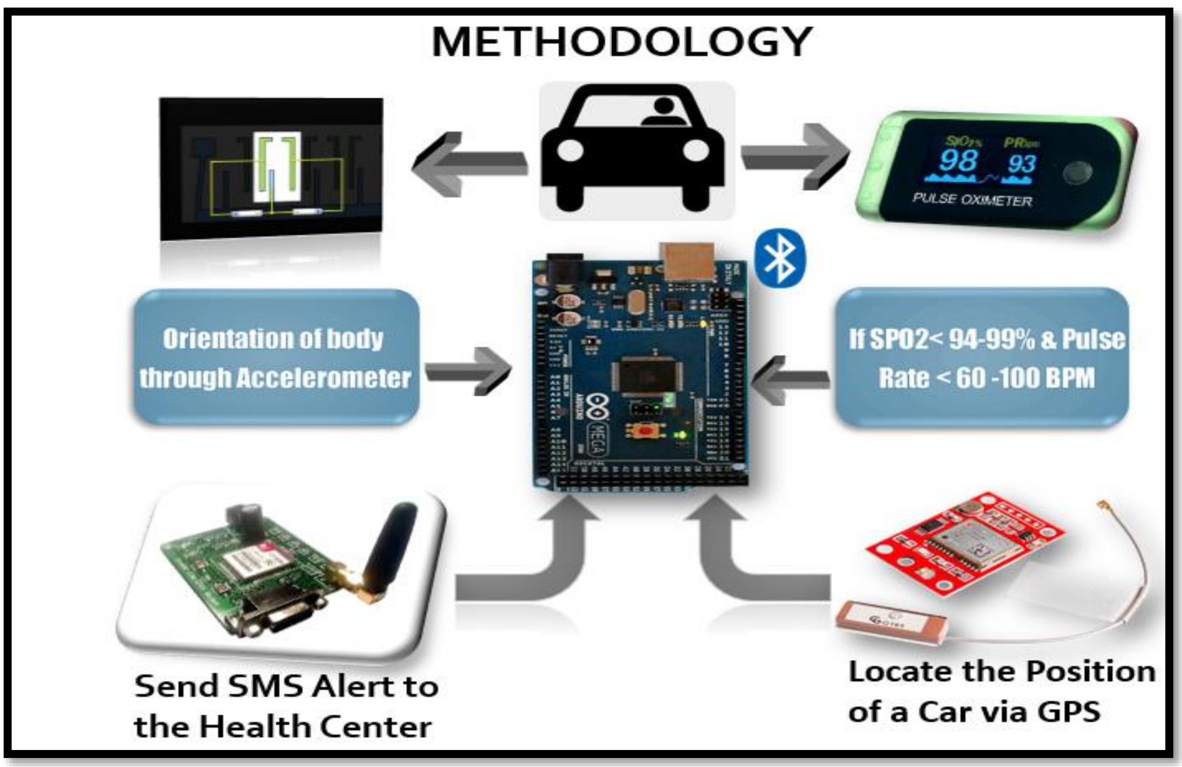


Figure 3-09 Receiver Section for displaying and sending the data

contains the actual reading of SpO2, Heart-Rate and GPS coordinate of that particular geographical location.

iii. **GPS Module:** GPS module is used to determine the latitude and longitude of the accident location of the car.

iv. **LCD 16x4:** LCD is used for knowing the health status (SpO2 and Heart-Rate) continuously.

v. **Arduino Mega 2560:** Arduino mega 2560 employs the processing of all the modules of receiver section attached to it.

3.3 Methodologies and Algorithms

3.3.1 Extraction of Spo_2 Measurements

SpO2 measurement is gotten from the algorithm created by Johnston. In like manner, the SpO_2 calculation based on Differential technique was intended to eliminate the derivatives that did not match the figured limit in order to create more exact SpO_2 readings. For the most part, amid resulting information collection, SpO_2 estimations diminished extensively during free movement. During tireless movement artefacts, R and IR PPG signals with similar amplitudes are obtained using the formula given below. This delivers an R-estimation of approximately 1 which thus yields SpO_2 readings close to 85%. [2]

SpO2 is gotten from the respective concentrations of both HbO2 and Hb in the arterial vessels. The ratio of R/IR light absorbed in a tissue bed is used for calculation. Since, the ratio of normalized Red light (ACR/DCR) to normalized Infrared light light (ACIR/ACIR) known as the ratio-of-ratios, or R value (3.1), is used in (3.2) to find out SpO_2, where A and B are constants determined amid exact calibration of an optical pulse sensor. A typical characteristic curve is appeared in Fig 3-10.

$$R = \mathrm{AC}R/\mathrm{DCR} \div AC\mathrm{IR}/DC\mathrm{IR} \qquad (3.1)$$

$$SpO_2 = A-B\ .\ R \qquad (3.2)$$

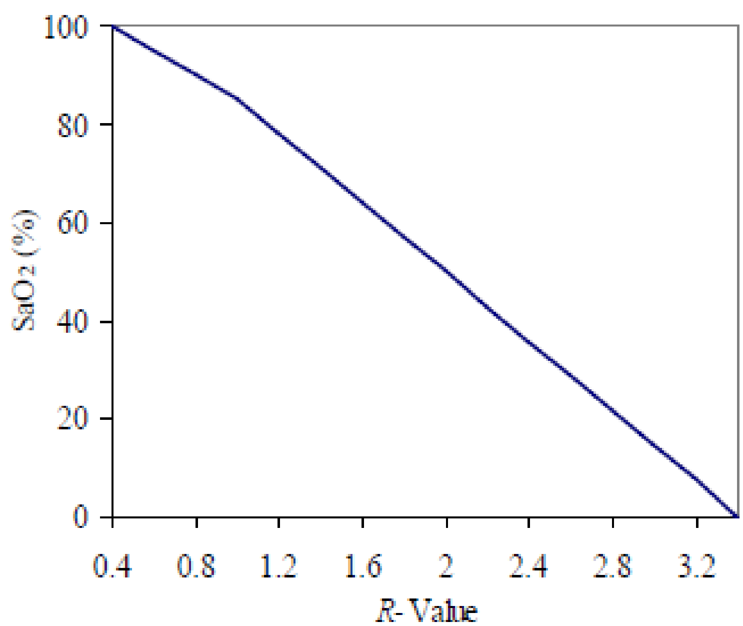


Figure 3-10 Typical SaO_2 characteristic curve

SpO_2 readings are acquired noninvasively from PPG signals by using frequency and time domain calculations. Scharf et al have recommended that Fast Fourier Transform (FFT) can be used for implementing the algorithm based on spectrum analysis for the estimation of SpO_2 with clinically adequate precision [6-8]. In a review that included the outline of various SpO_2 extraction calculations for conditions at rest, it was exhibited that a time based calculation in view of differential changes in PPG signals gave exact SpO_2 estimations, albeit spectral analysis created similar outcomes [3].

3.3.2 Extraction of Heart Rate Measurement

As a rule, the HR estimations are extracted by using the approach called Time-Based Averaging which gives more precise readings. Since, the polar heart rate readings were likewise calculated at the start of an experiment utilizing an averaging routine in light of a steady length window. HR estimations obtained amid rest utilizing the Peak Count-Based Averaging calculation devised by Johnston gave the absorption of IR and Red light by HbO2 and HHb for particular wavelength as appeared in fig. 3-10. The readings were less precise when contrasted with the Time-Based Averaging calculation for a normal HR of 95bpm (beats/min). The shorter averaging time likely gives less accurate readings for higher values in the algorithm of Peak Count-Based Averaging, since it combined 10 beats. [3]

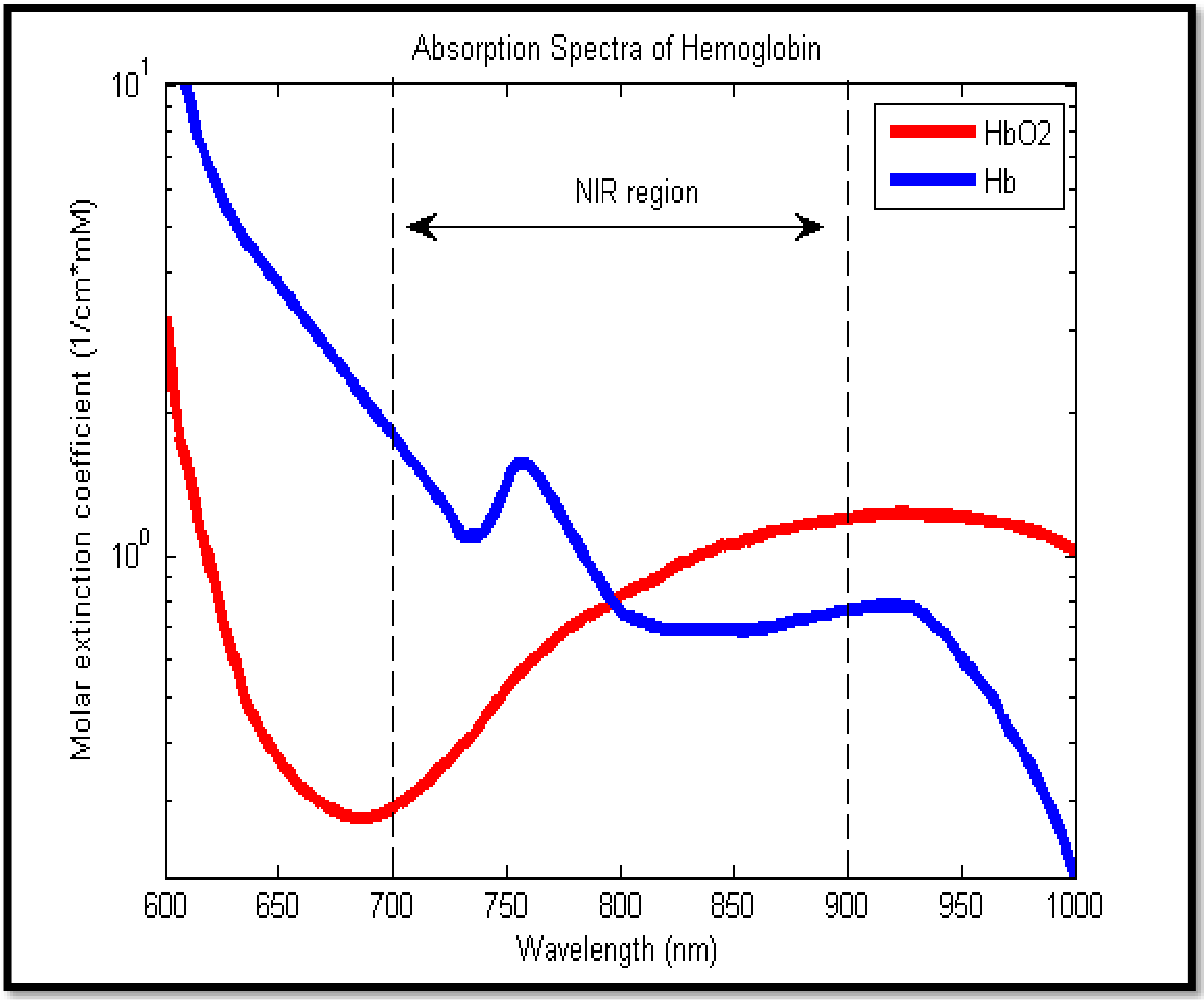


Figure 3-11 Absorption of oxy and deoxy haemoglobin in the R and IR regions

CHAPTER 4: Analysis and Evaluation

4.1 Hardware Configuration

4.1.1 Wrist-Band Wireless Device Circuit Diagram:

The circuit diagram of wireless wrist-band is shown below:

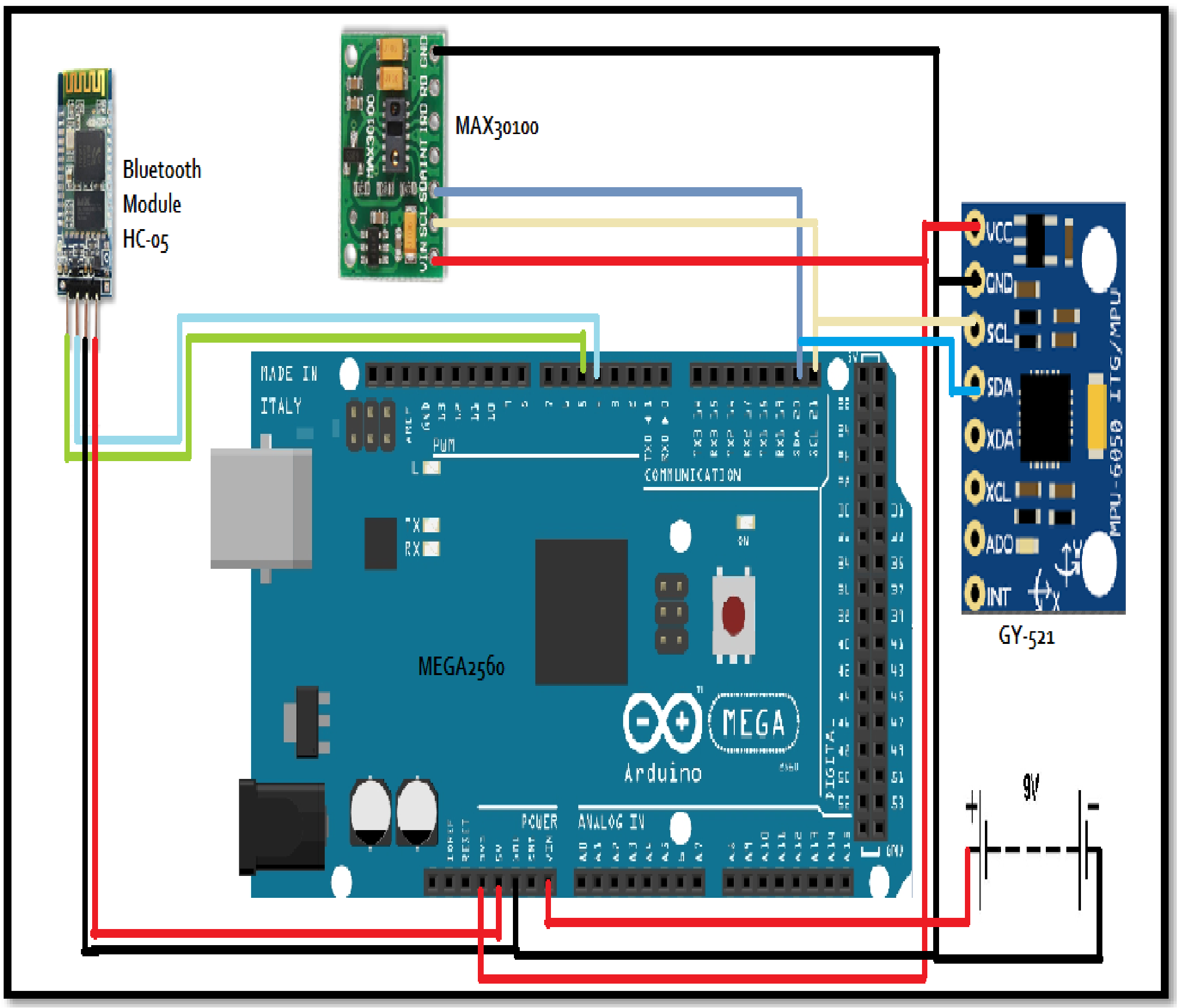


Figure 2-01: Wrist-Worn Transmitter Section Circuit Diagram

4.1.2 Receiver Section Circuit Diagram:

Receiver section circuit diagram is shown below:

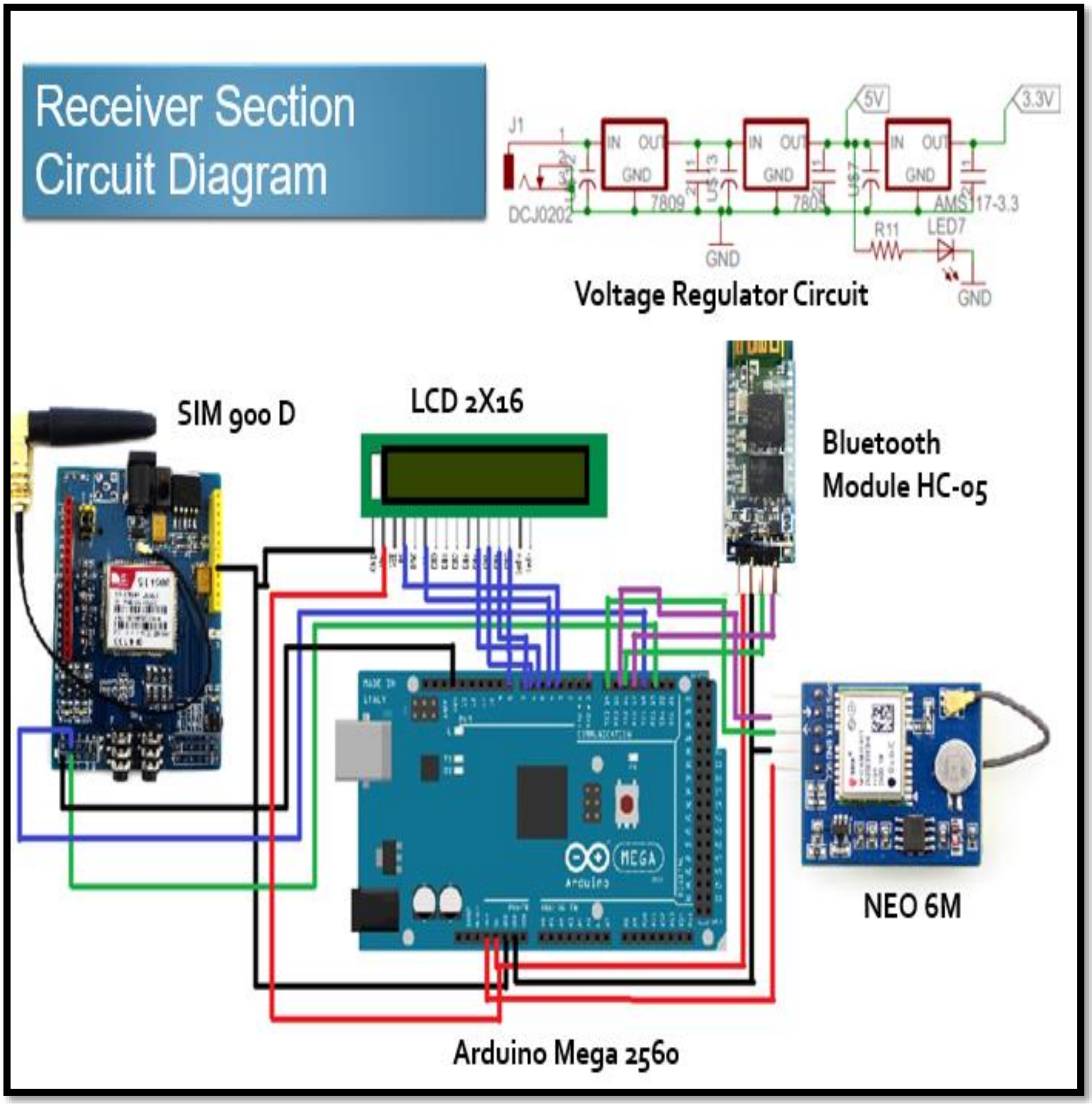


Figure 4-02: Receiver Section Circuit Diagram

4.1.3 Optical Sensing chip (Pulse Oximeter) and Accelerometer Circuitry

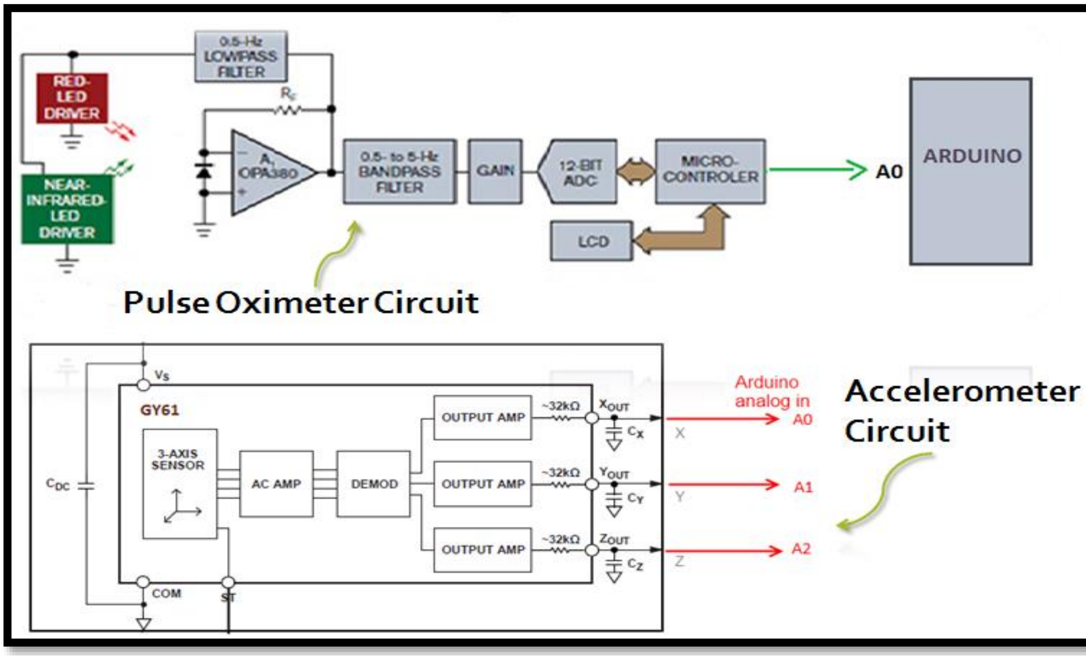


Figure 4-03: Pulse Oximeter and Accelerometer Circuitry

4.2 Graphical Simulation

Figure 4-04 shows the graphical simulation of heart beat pulse when the optical bio-sensing chip is not mounted on the wrist.

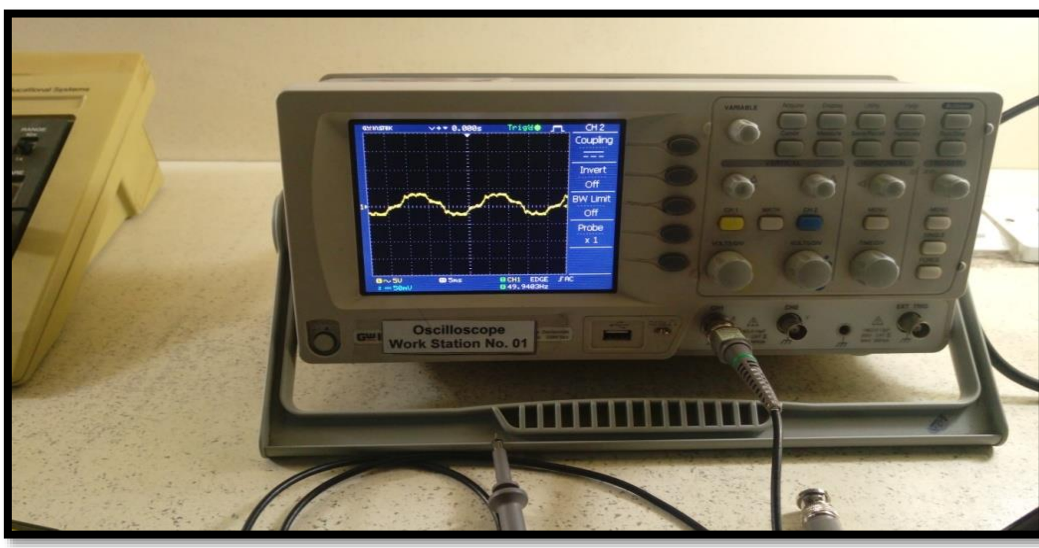


Figure 4-04: Graph of Pulse Sensor when not mounted on Wrist

Figure 4-05 shows the normal heart-rate pulse when the optical bio-sensing chip is mounted on the wrist.

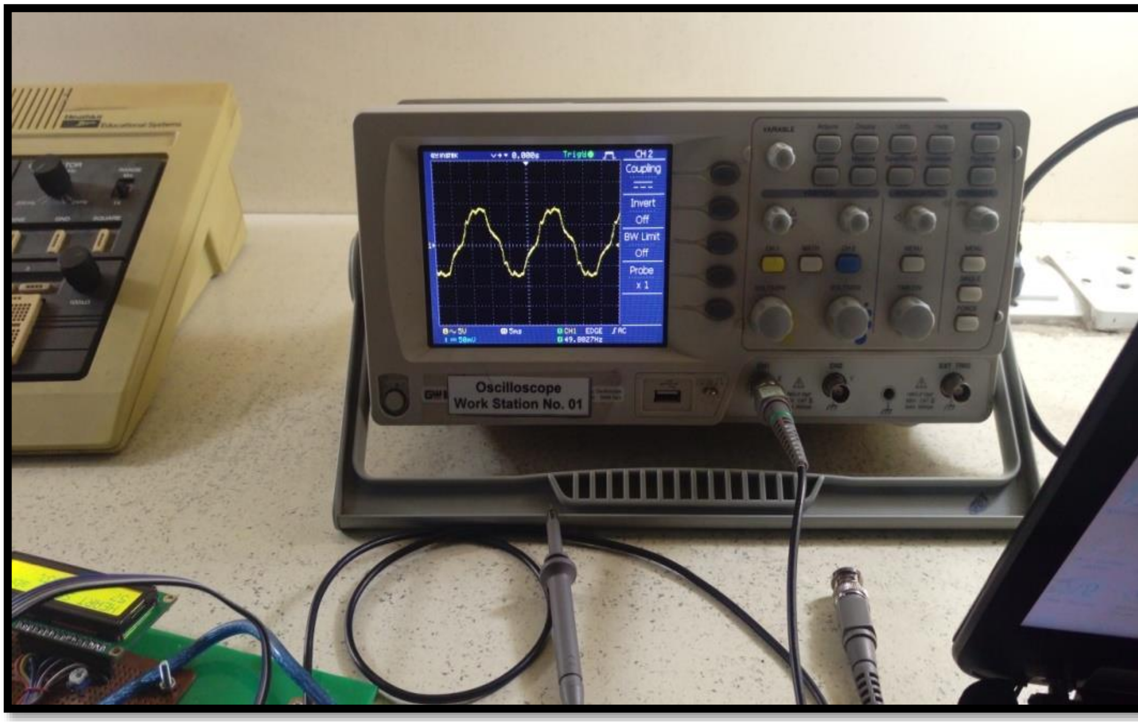


Figure 4-04 Graph of Pulse Sensor when mounted on Wrist

4.3 SMS Alert Screenshot

Fig 4-05 shows the SMS alert with all the required emergency parameters

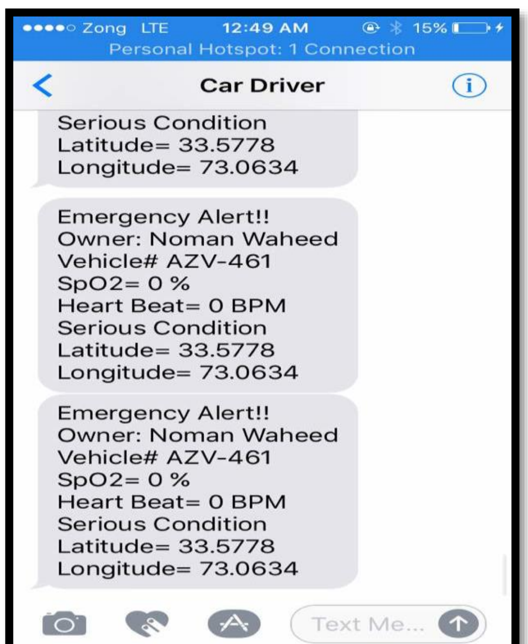


Figure 4-05: Sending Alert on Cell Number

CHAPTER 5: Conclusion

5.1 Achievements:

A wireless, wearable, health monitoring system has been produced based on little wrist-mounted optical sensor. The battery-operated gadget utilizes a light in weight optical pulse sensor and consolidates an annular photodetector to decrease the consumption of power. The device has a wireless correspondence abilities to transfer the readings of SpO_2, HR and body orientation to a receiver section. The data could upgrade the capacity of immediate responders in order to provide more effective medical aid, accordingly saving the lives of those persons who are in critical situation.

5.2 Applications:

Hardware finds its usefulness in the following areas:

- soldiers at combat situation
- persons in remote mining
- vehicle drivers
- for fireman
- patients in hospital

5.3 Suggestions for Future Work:

It is proposed that this system can be implemented in the Health-Centers or Rescue Centers such as 1122 for the ease of detecting the location of a car's driver in case of an accident or any severe injury. Also this system can be deployed in war zones for the continuous health monitoring of soldiers and the provision of immediate first-aid in case of any casualty.

"There is always room for improvements."

Andrew Bogut

CHAPTER 6: References

[1] http://www.worldlifeexpectancy.com/pakistan-road-traffic-accidents

[2] Josiane P. Lafleur, Recent advances in lab-on-a-chip for biosensing applications, in *"Biosensors and Bioelectronics"*Department of Pharmacy, University of Copenhagen, Copenhagen, Denmark Volume 76, 15 February 2016, Pages 213–233.

[3] W. S. Johnston, "Development of a Signal Processing Library for Extraction of SpO2, HR, HRV, and RR from Photoplethysmographic Waveforms," M.S. Thesis, Department of Biomedical Engineering, Worcester Polytechnic Institute, Worcester, MA, 2006

[4] Gary W. Comtois,"Implementation of Accelerometer-Based Adaptive Noise Cancellation in a Wireless Wearable Pulse Oximeter Platform for Remote Physiological Monitoring and Triage", M.S. Thesis, Department of Biomedical Engineering, Worcester Polytechnic Institute, Worcester, MA, 2007

[5] Edward D.Chan, Pulse oximetry: Understanding its basic principles facilitates appreciation of its limitations, Respiratory Medicine, Elsevier publisher,2013,pp 789-799

[6] T. L. Rusch, R. Sankar and J. E. Scharf, "Signal Processing Methods for Pulse Oximetry," Computations in Biology and Medicine, vol. 26, no. 2, pp. 143 – 159, 1996.

[7] J. E. Scharf and T. L. Rusch, "Optimization of Portable Pulse Oximetry through Fourier Analysis," presented at 12th Southern Biomedical Engineering 133 Conference, pp. 233 – 235, 1993.

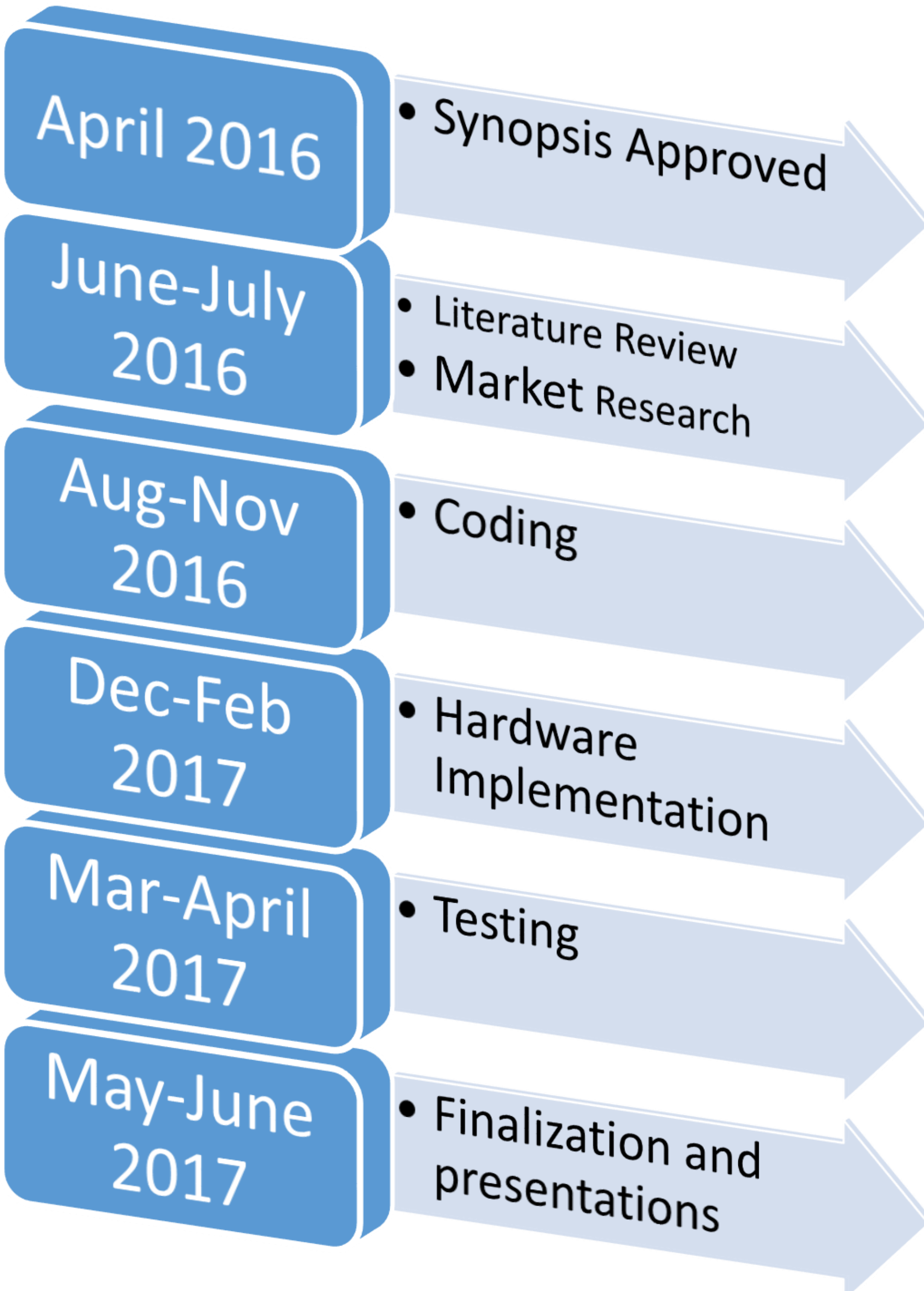
[8] T. L. Rusch, J. E. Scharf and R. Sankar, "Alternate Pulse Oximetry Algorithms for SpO2 Computation," presented at 16th Annual International Conference of the IEEE EMBS, pp. 848 – 849, 1994.

Appendix A: Synopsis

Oxyhealth Monitoring System(OMS)

Extended Title: Oxyhealth Monitoring System (OMS)
Brief Description of The Project / Thesis with Salient Specifications: The proposed project will monitor the level of oxygen saturation (SPO2) in the blood, pulse rate of the person by using pulse oximeter and will measure the orientation of the body through accelerometer. Hence, the SMS alert is sent to the desired numbers in case of a severe condition. Furthermore, the position of a person is detected through GPS.
Scope of Work : The project incorporates biomedical sensor, detectors and communication system that simultaneously works and make a reasonable health monitoring system.
Academic Objectives : The proposed project will enable us to understand and apply the techniques of Electronic Circuits, Programming, Wireless Communication and Mobile communication.
Application / End Goal Objectives : This project can be used to monitor the health of soldier at combat situation, person at remote mining, fireman and car's driver health and provide immediate first aid in case of injury or accident.
Previous Work Done on The Subject : Previously the research work is done only on detection of SPO2 level of patient using fingertip pulse oximeter
Material Resources Required: Arduino, GSM Module, Accelerometer, GPS Module
No of Students Required : 4 **Group Members:** Noman Waheed Muhammad Zeeshan Sadiq Muhammad Uzair Zubair Abubakar Saadat
Special Skills Required: Proteus, Arduino IDE

Appendix B: Project Progress



Appendix C: Code

Wrist-Band Wireless Device Code:

```
#include <Wire.h>
#include <SoftwareSerial.h>
#include "MAX30100_PulseOximeter.h" // include optical pulse sensor library
#include <MPU6050.h>

SoftwareSerial ss(4, 5);
#define REPORTING_PERIOD_MS     1000

MPU6050 mpu;
PulseOximeter pox;

uint32_t tsLastReport = 0;

void onBeatDetected()
{
   Serial.println("Beat!");
}

void setup()
{
  Serial.begin(115200);           //initialization begin
   ss.begin(9600);

  Serial.println("Initialize MPU6050…");  //accelerometer sensor initialization

    while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Couldn't find a valid MPU6050 sensor, check wiring!!!");
    delay(500);
  }
  mpu.calibrateGyro();
  mpu.setThreshold(3);

  checkSettings();
  Serial.print("Initializing pulse oximeter..");
  if (!pox.begin())
  {
      Serial.println("FAILED");
      for(;;);
  }
  else
  { Serial.println("SUCCESS");
  }
```

```
 for(int i=0;i<=30000;i++)

    {beatTest();}


}

void checkSettings()
{
  Serial.println();

  Serial.print(" * Sleep Mode:        ");
  Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

  Serial.print(" * Clock Source:      ");
  switch(mpu.getClockSource())
  {
    case MPU6050_CLOCK_KEEP_RESET:     Serial.println("Stops the clock and keeps the
timing generator in reset"); break;
    case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with external 19.2MHz
reference"); break;
    case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with external 32.768kHz
reference"); break;
    case MPU6050_CLOCK_PLL_ZGYRO:      Serial.println("PLL with Z axis gyroscope
reference"); break;
    case MPU6050_CLOCK_PLL_YGYRO:      Serial.println("PLL with Y axis gyroscope
reference"); break;
    case MPU6050_CLOCK_PLL_XGYRO:      Serial.println("PLL with X axis gyroscope
reference"); break;
    case MPU6050_CLOCK_INTERNAL_8MHZ:  Serial.println("Internal 8MHz oscillator");
break;
  }

  Serial.print(" * Gyroscope:         ");
  switch(mpu.getScale())
  {
    case MPU6050_SCALE_2000DPS:        Serial.println("2000 dps"); break;
    case MPU6050_SCALE_1000DPS:        Serial.println("1000 dps"); break;
    case MPU6050_SCALE_500DPS:         Serial.println("500 dps"); break;
    case MPU6050_SCALE_250DPS:         Serial.println("250 dps"); break;
  }

  Serial.print(" * Gyroscope offsets: ");
```

```
Serial.print(mpu.getGyroOffsetX());
  Serial.print(" / ");
  Serial.print(mpu.getGyroOffsetY());
Serial.print(" / ");
  Serial.println(mpu.getGyroOffsetZ());

  Serial.println();
}

void gyroScope()
{

  Vector RawGyro = mpu.readRawGyro();
  Vector NormGyro = mpu.readNormalizeGyro();

  Serial.print(" Xraw = ");
  Serial.print(RawGyro.XAxis);
  Serial.print(" Yraw = ");
  Serial.print(RawGyro.YAxis);
  Serial.print(" Zraw = ");
  Serial.println(RawGyro.ZAxis);

  delay(500);
}

void beatTest()
{
  // Make sure to call update as fast as possible
    pox.update();

    // Asynchronously dump heart rate and oxidation levels to the serial
    // For both, a value of 0 means "invalid"
    if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
        Serial.print("HEART RATE:");
        Serial.print(pox.getHeartRate());
        Serial.print("BPM / SpO2:");
        Serial.print(pox.getSpO2());


        tsLastReport = millis();
    }
}
void loop()


{
```

```
  for(int i=0;i<10;i++)
  {

for(int i=0;i<=30000;i++)

{
beatTest();

}
  }
       ss.print("$N1ok,");
       ss.print(pox.getHeartRate());




 ss.print(",");
       ss.print(pox.getSpO2());
       ss.println("*");


 Serial.print("$N1ok,");
      Serial.print(pox.getHeartRate());
      Serial.print(",");




Serial.print(pox.getSpO2());
      Serial.println("*");

  if(pox.getHeartRate() < 20 || pox.getSpO2() < 50)
  {


gyroScope();
  }


delay(1000);
}
```

Receiver Section Code:

```
#include <LiquidCrystal.h> //library include for LCD
//#include <Keypad.h>
LiquidCrystal lcd(3, 4, 5, 6, 7, 8);
#include <TinyGPS.h>

String inputString = "";             // a string that hold incoming data
boolean stringComplete = false;      // whether the string is complete or not
String sms_string="";
String gps_string="";
String new_smss="";


   boolean sms_r=0;
   boolean gps_data=0;
   unsigned int my_spo2=0;
   unsigned int my_heart_beat=0;


 boolean new_sms=0;
   boolean condition=0;
   TinyGPS gps;
  float latitude=0;
  float longitude=0;


  void setup() {

// initialize serial:
  pinMode(13,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(11,INPUT_PULLUP);
  digitalWrite(8,LOW);digitalWrite(9,LOW);digitalWrite(10,LOW);

  Serial.begin(115200);
   Serial1.begin(19200);
    Serial2.begin(9600);
     Serial3.begin(9600);

     lcd.begin(16, 2);
     lcd.print("OXYHEALTH SYSTEM");
     delay(1000);
```

```
   inputString.reserve(200);       // 200 bytes reserved for input String:

  simple_gsmtap();


simple_gsmtap();
  ecco_off();
  text_mode();//0314-9359751
 sms_forwarding();
  delete_sms();
}
void get_response()
{
  long int t=millis();
  do
   {
   serialEvent1() ;
  }while((millis()-t)<1000);

}

void sms_forwarding()

{

   Serial.println("Sms Forwarding");
  Serial1.println("AT+CNMI=1,2,0,0,0");

get_response();
}

void simple_gsmtap()
{
   Serial.println("GSM Tap");
  Serial1.write('A');delay(10);
  Serial1.write('T');delay(10);
  Serial1.write(13);// clear the string:
get_response();
}

void ecco_off()
{
  // Serial.println("Sending AT");
  Serial.println("Eco off");
```

```
  Serial1.write('A');
  delay(10);
  Serial1.write('T');
  delay(10);
  Serial1.write('E');delay(10);
  Serial1.write('0');
delay(10);



 Serial1.write(13);// clear the string:


get_response();
}

void text_mode()
{
  Serial.println("Set modem to Text mode");
//   Serial.println("Sending AT");
  Serial1.print("AT+CMGF=1");

  Serial1.write(13);// clear the string:
get_response();
}
void delete_sms()
{


  Serial.println("Delete Sms");
//   Serial.println("Sending AT");


Serial1.print("AT+CMGD=1,4");

  Serial1.write(13);// clear the string:
get_response();
}
void send_sms()
{
  Serial1.println("AT");get_response();
  Serial1.print("AT+CMGS=");Serial1.write("");
  Serial1.print("03135961860");

Serial1.write("");//03135961860
  Serial1.write(13);get_response();
```

```
 Serial.println(" Spo2=");
 Serial1.print(my_spo2);
 Serial.println("HB");
Serial.print(my_heart_beat);
Serial1.print("latitude=");
 Serial1.println(latitude);
 Serial1.print("78");


Serial1.print("longitude=");
 Serial1.print(longitude);
 Serial1.print("34");
 Serial1.write(26);
 get_response();get_response();get_response();
}
////
void send_sms_sc()
{
 Serial1.println("AT");get_response();
 Serial1.print("AT+CMGS=");Serial1.write("");

 Serial1.print("03135961860");Serial1.write("");////03135961860
 Serial1.write(13);get_response();

 Serial1.println("Emergency Alert!!");delay(10);
 Serial1.println("Owner: Noman Waheed");delay(10);
 Serial1.println("Vehicle# AZV-461");delay(10);
 Serial1.print("SpO2= ");
 Serial1.print(my_spo2);
 Serial1.println(" %");
 Serial1.print("Heart Beat= ");
 Serial1.print(my_heart_beat);
 Serial1.println(" BPM");

Serial1.println("Serious Condition");delay(100);
 Serial1.print("Latitude= ");
 Serial1.print(latitude);
 Serial1.println("78");
 Serial1.print("Longitude= ");
 Serial1.print(longitude);
 Serial1.print("34");
 Serial1.write(26);
 get_response();get_response();get_response();
 delay(1000);
}
```

```
void loop()
{

  serialEvent() ;
  serialEvent1() ;
  serialEvent2();
 //get_gps();f
  // serialEvent3();
  if (stringComplete||!digitalRead(11)) {


 Serial1.print(inputString);
    inputString = "";
    get_gps();
    send_sms_sc();
    Serial.println();
    Serial.println();
    Serial.println();
    Serial.println("longitude--->");
      Serial.println(longitude);
    Serial.println("latitude--->");
      Serial.println(latitude);
    Serial.println();Serial.println();
    stringComplete = false;
  }
  if(new_sms)
  {
    String nsms_d=new_smss;new_smss="";
    nsms_d.trim();
    if(nsms_d.startsWith("$L1on"))
    {
      Serial.println("LED one On");digitalWrite(13,HIGH);
    }
    if(nsms_d.startsWith("$L2on"))
    {
      Serial.println("LED 2 On");digitalWrite(9,HIGH);
    }
    if(nsms_d.startsWith("$L3on"))
    {

 Serial.println("LED 3 On");digitalWrite(10,HIGH);
    }
    //
    if(nsms_d.startsWith("$L1off"))
    {
```

```
      Serial.println("LED one Of");digitalWrite(13,LOW);//8
    }
    if(nsms_d.startsWith("$L2off"))
    {
      Serial.println("LED 2 Of");digitalWrite(9,LOW);
    }


 if(nsms_d.startsWith("$L3off"))
    {
Serial.println("LED 3 Of");digitalWrite(10,LOW);
    }
    //

 new_sms=0;
  }
  if(sms_r)
  {

    String data=sms_string;
    data.trim();
    String new_value=data;
    if(new_value.startsWith("$N1")&&new_value.endsWith("*"))//$N1,220,123*
      {


          int yu=0;
          int xy=0;
          String vstring="";
          String cstring="";
          int sonv=new_value.indexOf("*");
           for(yu=0;yu<sonv;yu++)
       {
        if(new_value.charAt(yu)==',')
        {
          xy++;
      //   Serial.print("Comma at ");Serial.println(yu);
          goto over_next;


}

        if(xy==1)//$N1,220,22*
        {
          vstring=vstring+new_value.charAt(yu);
        }
```

```
      if(xy==2)
     {
      cstring=cstring+new_value.charAt(yu);
     }
     over_next:
     delay(1);
    }


   int my_hb=vstring.toInt();
   int my_spo=cstring.toInt();

   my_spo2=my_spo;

   my_heart_beat=my_hb;


     lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("HB=");
  lcd.setCursor(3,0);
  lcd.print(my_heart_beat);
  lcd.setCursor(6,0);
  lcd.print("BPM");

  lcd.setCursor(0,1);
  lcd.print("SPO2=");

 lcd.setCursor(5,1);
  lcd.print(my_spo2);
  lcd.setCursor(7,1);
  lcd.print("%");
   if(new_value.startsWith("$N1ok"))
   {
     condition=0;Serial.println("Condition ok");
   }else
   {


condition=1;Serial.println("Condition bad");
     get_gps();
     send_sms_sc();
   }

         }
      sms_r=0;
```

```
  }
}
////GPS

void get_gps()
{
  bool newData = false;

unsigned long chars;


  unsigned short sentences, failed;

  // For one second we parse GPS data and report some key values
  for (unsigned long start = millis(); millis() - start < 1000;)

 {
    while (Serial3.available())
    {
      char c = Serial3.read();
      Serial.write(c);
      if (gps.encode(c))
        newData = true;
    }
  }

  if (newData)
  {
    float flat, flon;
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);
    latitude=flat;
    longitude=flon;
    Serial.print("LAT=");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 4);
    Serial.print(" LON=");
    Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 4);
    Serial.print(" SAT=");


Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 : gps.satellites());
    Serial.print(" PREC=");
    Serial.print(gps.hdop() == TinyGPS::GPS_INVALID_HDOP ? 0 : gps.hdop());
    Serial.println();Serial.println();
    Serial.println();Serial.println();
    Serial.print("#########    ");
```

```
    Serial.print(latitude,6);Serial.print("       ");
    Serial.println(longitude,6);
    Serial.println();Serial.println();
    Serial.println();Serial.println();
    Serial.println("**********");
  }

  gps.stats(&chars, &sentences, &failed);
  Serial.print(" CHARS=");


Serial.print(chars);

 Serial.print(" SENTENCES=");
  Serial.print(sentences);
  Serial.print(" CSUM ERR=");
  Serial.println(failed);
  if (chars == 0)
    Serial.println("** No characters received from GPS: check wiring **");
}
void serialEvent() {
  while (Serial.available())
  {
    // get the new byte:
    char inChar = (char)Serial.read();

    inputString += inChar; // add it to the inputString:
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
void serialEvent1() {
  while (Serial1.available()) {


    // get the new byte:
    char inChar1 = (char)Serial1.read();
   if(inChar1=='$')


{
    new_smss="";
   }
```

```
   if(inChar1=='*')
    {
   new_sms=1;
   }
   new_smss+=inChar1;
    Serial.write(inChar1);
  }
}

void serialEvent2() {


while (Serial2.available()) {


// getting new byte:



 char inChar2 = (char)Serial2.read();


  if(inChar2=='$'){sms_string="";}

if(inChar2=='*'){sms_r=1;sms_string+=inChar2;}

 if(!sms_r)
    {
    sms_string+=inChar2;
    }

 Serial.write(inChar2);
  }

}
```