**PROJECT**

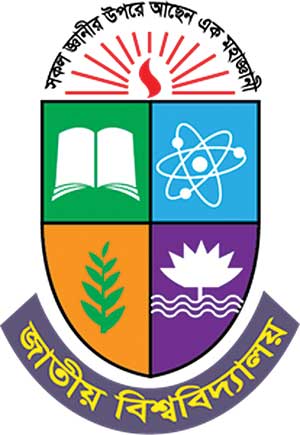
ON

DESIGN AND IMPLEMENTATION OF

IOT BASED HEART BEAT DETECTION AND MONITORING SYSTEM

Electronics and Communication Engineering

National University of Bangladesh



Prepared by

Roll No. :

Registration No. :

Session :

A project paper submitted in partial completion of the requirements for the degree of Bachelor of Science in Electronic and Communication Engineering.

**DECLARATION**

I, Roll: .......... Reg: ......... declares that this project report entitled “IoT based Heart beat Detection and Monitoring system” and the work presented in it is my own, I confirm that,

* This work is done fully by the author as a part of undergraduate degree at this university.
* This is admitted that any part of this project report has not been published or submitted in any other institution.
* All main sources of help is greatly acknowledged and taking advice from my supervisor.

Signature

Date / /

Supervised By

**Runa Rukhsana khan**

Associate Professor

Department of ECE

National University of Bangladesh.

Signature

Date / /

Submitted By

**Md Ariful Islam**

Roll-1300006

Reg-1213007

Department of ECE

National University of Bangladesh.

**TABLE OF CONTENTS PAGE NO.**

**Chapter One: Introduction**

* 1. Introduction 3
  2. Objective 3
  3. Problem Statement 4
  4. Project Scope 4
  5. Limitations of Project 5

**Chapter Two: Literature Review**

2.1 Book Name: Pulse detector through difference amplifier 6

2.2 Journal Name: Microcontroller based Heart Rate Monitor 6

2.3 Journal Name: heart rate monitoring in the estimation energy 7

2.4 Journal Name: Heart rate monitor applications and limitations 7

**Chapter Three: Theoretical Analysis**

3.1 Introduction 9

3.2 Heart Rate 9

3.3 Types of Heart rate monitors 9

3.4 Introduction to Heart Rate monitor concept 10

3.5 Hardware and Software Technology Background 10

3.5.1 The Arduino Uno-Board 10

3.5.2 The Heart Rate Sensor 11

3.5.3 The Pulse Sensor Unit 11

3.5.4 LCD Display Unit 11

3.5.5 ESp8266 12

3.6 Summary 13

**Chapter FOUR: Research Methodology**

4.1 Introduction 15

4.2 Design 15

4.3 Block Diagram 16

4.4 Hardware Requirements 17

* 1. Sensor Introduction 17

4.6 Arduino Uno 18

4.7 Esp8266 20

4.8 Budget Costing 21

**CHAPTER FIVE: PROTOTYPE/PRODUCT DEVELOPMENT**

5.1 Introduction 23

5.2 Hardware Design 23

5.3 Project setup

5.4 Thing Speak Setup 24

5.5 Working Explanation 26

5.6 Software Design 27

5.7 Coding 28

**CHAPTER SIX: TESTING AND RESULTS**

6.1 Introduction 37

6.2 Heart Beat Senor 37

6.3 Arduino UNO Controller 37

6.4 Software Tools to Write Program 38

6.5 Uploading and Testing 41

**CHAPTER SEVEN: CONCLUSION AND SUGGESTION**

7.1 Introduction 44

7.2 Conclusion 44

7.3 Discussion 45

7.4 Future works 45

**List of Figures**

3.1 Two Different Wavelength

3.2: Photo Dectector Reaction 10

3.2: Pin out of the LCD 11

3.3: how sensor works 12

3.4: ESP8266 12

4.1: HRM block diagram 16

4.2: PIN out diagram of the Arduino Uno 18

4.3: Esp8266 20

5.1: Circuit Diagram 23

5.2: Final project setup 25

5.3: Thing speak channel setup 25

5.4: Thing speak key generate 26

5.5: Arduino IDE 27

6.1: Installing Options 38

6.2: Installation Folder 39

6.3: Installing 39

6.4: Basics Blink 40

6.5: Basics Tools Board 40

6.6: Basic Port Setting 41

6.7: Uploading Program 41

6.8: Bpm Show on Web and LCD 42

## List of Tables

Table 4.3: Budget and Costing 21

## ABSTRACT

This project consists of two subsystems, which are hardware system and software system. Basically, the hardware system consists of Arduino microcontroller and other electronic circuits for detecting the heart pulse rate. The heart pulse rate scanning is based on the use of LDR sensitivity. The Arduino will be able to detect the pulse change, counting the pulse per minute, send to PC through ESP8266 and also display the actual status on LCD. The software system consists of application program that connect and receive data through serial, display the result per minute and heart rate status information with data logging capabilities. There are a number of methods that can be used to detect the presence of the heart pulse. For example, the ECG wave can be used to produce a synchronized pulse corresponding to each beat of the heart. Other techniques utilize pressure differentials due to the pulse or optical methods that cause the pressure pulse to interfere with a light beam. This project using super bright red LED. Once the circuit is given a 5 Volt DC supply, The LED will continuously flash red light onto the LDR surface. This indicates that the circuit works and after programming the Arduino it can produce output for Heart Rate measurement. The last part of this project is to design and develop Human Machine Interface (HMI) running on windows base platform with the capability to display the heart beat pulse signal (digital) and the heart beat pulse per minute. Project is done for the use as a future reference for those who want to do a research on heart rate monitor system.

**CHAPTER ONE**

**INTRODUCTION**

## 1.1 Introduction

Telemedicine is the most important step in cutting costs and increasing service quality in health care. The traditional telemedicine systems mostly enable communication between health professionals in order to give doctors in remote locations access to specialist’s knowledge and monitoring of patients remotely for home care or emergency applications. Essentially, these systems provide an extension of hospital environment and connect diagnostic equipment at home with hospitals using fixed telephone or satellite networks. Although these systems provide many benefits for its users there are still many limitations.

One of the main limitations is the lack of mobility that hinders their usage in many scenarios. Besides that, a more general problem of today’s health care is the insufficient availability of data conceding the status and medical history of the patient, both to the medical personnel and to the patient himself. Frequent measurements of vital signs could give indications about the current status of chronic illnesses and are necessary for optimization of the treatment, but would incur significant cost.

Recent advancements in development of wide area wireless technologies have made possible development of new generation of telemedicine systems that should provide mobile, wearable and flexible health monitoring systems. Such systems will enable constant monitoring of health data and constant access to the patient regardless of patient’s current location or activity and with a fraction of cost of the regular face-to- face examination.

## 1.2 Objective

The main objectives of this project are to design and develop hardware base arduino Uno with the capabilities to detect heart beat pulse from human body and transmit and receive data between PC using Wi-Fi module (ESP8266).

To design and develop program that can calculate the heart beat pulse receive and transmit the data to the PC for monitoring purpose.

Basically, the system consists of software and hardware system. The software consists of computer, mobile or any other smart device to display the output over the internet. At hardware part, system consists of heartbeat sensor module, Arduino Uno circuits, LCD for quick display and ESP 8266 module.

## 1.3 Problem Statement

From the patient's view, the existing care system creates an uncomfortable feeling because they cannot move freely throughout the monitoring progress. Patients should be in the hospital and if patients with serious heart problems, they are more limited area of within the designated wards only.

In addition, patients also need make an appointment with the doctor before the checkup and sometimes it’s very difficult to do because of the vacancy doctors and patients different. Patient's emotional state should also be considered during the meeting with doctor because there is a fear or phobia patients during trips to the doctor and have patients who are afraid to come to the hospital while giving effect to the data taken by the doctor.

## 1.4 Project Scope

The scope of the project covers the design of the following: Hardware Controller

* The heart beat scanning mechanism is based on the use of LDR sensitivity.
* The Arduino will be able to detect the pulse change, counting the pulse per minute, send to PC through serial data communication and display the actual status on LCD.
* 2 way communications are available between hardware controller and PC.
* User only needs to place his or her thumb and the controller will automatically give the result after several seconds.

## 1.5 Limitation of Project

The main challenge for this project is system running on power supply. If the power supply run out of power the system is nonfunctional.

Another challenge is the unstable and high sensitivity nature of the Light- dependent resistor alternatively called an LDR. LDR can be sensitive to high levels of ambient infrared light example strong sunlight. If this maybe the case try excluding the light by covering the sensor with some dark fabric example by placing your hand inside a cloth b

**CHAPTER TWO**

**LITERATURE REVIEW**

This chapter describes and discusses the research from various sources such as textbooks, articles and the internet. It consists of information which is vital in the development of this project.

Related works are :

**2.1. Book Name: Pulse detector through difference amplifier**

Writer Name: Norwahidah Mohd Noor

University: Kolej University, Teknikal, Kebangsaan ,Malaysia.

Descriptions:

This book presents the principles and working of pulse detecting system using difference amplifier. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart pulse rate is to be monitored the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

**2.2 Journal Name: Microcontroller based Heart Rate Monitor using Sensor.**

Writer Name: Sharief F. Babiker, Liena Elrayah Abdel-Khair, Samah M. Elbasheer

University: University of Khartoum , Khartoum, Sudan.

Available online at : [www.uofk.edu/uofkej](http://www.uofk.edu/uofkej)

Descriptions:

This paper presents the design and development of a microcontroller based heart rate monitor using fingertip sensor. The device uses the optical technology to detect the flow of blood through the finger and offers the advantage of portability over tape-based recording systems. The important feature of this research is the use of Discrete Fourier Transforms to analyze the ECG signal in order to measure the heart rate. Evaluation of the device on real signals shows accuracy in heart rate estimation, even under intense physical activity. The performance of HRM device was compared with ECG signal represented on an oscilloscope and manual pulse measurement of heartbeat, giving excellent results. Our proposed Heart Rate Measuring (HRM) device is economical and user friendly.

**2.3.Journal Name: The use of heart rate monitoring in the estimation of energy expenditure**

Writter Name: Cessay SM, Pentice Am, Scott Am

University: University of Cambridge, MRC Dunn Nutrition Unit.

Available Online at: [www.ncbi.nlm.nih.gov/pubmed](http://www.ncbi.nlm.nih.gov/pubmed)

Descriptions:

This paper presents of heart rate monitoring in the estimation of energy expenditure. A modified heart rate (HR) method for predicting total energy expenditure (TEE) was cross-validated against whole-body calorimetric (CAL). Minute-by-minute HR was converted to energy expenditure (EE) using individual calibration curves when HR exceeded a pre-determined 'FLEX' value designed to discriminate periods of activity. ('FLEX' HR was defined as the mean of the highest HR during rest and the lowest HR during the lightest imposed exercise.) Sedentary EE (below FLEX) was calculated as the mean EE during lying down, sitting and standing at rest. Sleeping EE was calculated as basal metabolic rate (BMR) predicted from standard equations.

**2.4. Journal name: Heart rate monitoring applications and limitations.**

Writter Name: Achten J, Jeukendrup AE

University: University of Birmingham, Edgbaston, Birmingham, United Kingdom.

Available Online at: [www.ncbi.nlm.nih.gov/pubmed](http://www.ncbi.nlm.nih.gov/pubmed)

Descriptions:

Over the last 20 years, heart rate monitors (HRMs) have become a widely used training aid for a variety of sports. The development of new HRMs has also evolved rapidly during the last two decades. In addition to heart rate (HR) responses to exercise, research has recently focused more on heart rate variability (HRV). Increased HRV has been associated with lower mortality rate and is affected by both age and sex. During graded exercise, the majority of studies show that HRV decreases progressively up to moderate intensities, after which it stabilizes. There is abundant evidence from cross-sectional studies that trained individuals have higher HRV than untrained individuals. The results from longitudinal studies are equivocal, with some showing increased HRV after training but an equal number of studies showing no differences. The duration of the training programs might be one of the factors responsible for the versatility of the results. HRMs are mainly used to determine the exercise intensity of a training session or race. Compared with other indications of exercise intensity, HR is easy to monitor, is relatively cheap and can be used in most situations.

**CHAPTER THREE**

**THEORETICAL ANALYSIS**

## 

## 3.1 Introduction

Previously this Heart Rate Monitor project has been done by other student. The different with this project it is embedded system where the heart rate device itself interfacing with computer to receive data through serial, display the result per minute and heart beat pulse signal or graph and also heart rate status information with data logging capabilities.

In order to gain a better understanding of this project, a research was done on the topic of this project. The research was found from journals, articles, books and web sites from the internet.

## 3.2 Heart Rate

Heart rate is the number of [heartbeats](http://en.wikipedia.org/wiki/Cardiac_cycle) per unit of [time](http://en.wikipedia.org/wiki/Time), typically expressed as *beats per minute* (BPM). Heart rate can vary as the body's need to absorb oxygen and excrete carbon dioxide changes, such as during [physical exercise](http://en.wikipedia.org/wiki/Physical_exercise) or sleep.

The measurement of heart rate is used by [medical professionals](http://en.wikipedia.org/wiki/Medical_professional) to assist in the [diagnosis](http://en.wikipedia.org/wiki/Diagnosis) and tracking of medical conditions. It is also used by individuals, such as [athletes,](http://en.wikipedia.org/wiki/Sportsperson) who are interested in monitoring their heart rate to gain maximum efficiency from their training. The [*R wave*](http://en.wikipedia.org/wiki/R_wave) *to R wave interval* (*RR interval*) is the inverse of the heart rate.

## 3.3 Types of Heart Rate Monitors

Modern heart rate monitors usually comprise two elements: a chest strap transmitter and a wrist receiver or mobile phone (which usually doubles as a [watch](http://en.wikipedia.org/wiki/Watch) or phone). In early plastic straps water or liquid was required to get good performance. Later units have used conductive smart fabric with built-in microprocessors which analyze the EKG signal to determine heart rate.

Strapless heart rate monitors now allow the user to just touch two sensors on a wristwatch display for a few seconds to view their heart rate. These are popular for their comfort and ease of use though they don't give as much detail as monitors which use a chest strap.

More advanced models will offer measurements of heart rate variability, activity, and breathing rate to assess parameters relating to a subject's fitness. Sensor fusion algorithms allow these monitors to detect core temperature and dehydration.

Another style of heart rate monitor replaces the plastic around-the-chest strap with fabric sensors - the most common of these is a sports bra for women which include sensors in the fabric.

In old versions, when a heartbeat is detected a radio signal is transmitted which the receiver uses to determine the current heart rate. This signal can be a simple radio pulse or a unique coded signal from the chest strap (such as [Bluetooth](http://en.wikipedia.org/wiki/Bluetooth), [ANT](http://en.wikipedia.org/wiki/ANT_%28network%29) or other low-power radio link) the latter prevents one user's receiver from using signals from other nearby transmitters (known as cross-talk interference).

Newer versions include a microprocessor which is continuously monitoring the EKG and calculating the heart rate, and other parameters. These may include accelerometers which can detect speed and distance eliminating the need for foot worn devices.

There are a wide number of receiver designs, with various features. These include average heart rate over exercise period, time in a specific heart rate zone, calories burned, breathing rate, built-in speed and distance and detailed logging that can be downloaded to a computer.

## 3.4 Introduction to Heart Rate Monitor concept

The pulse rate will be detected by using heart beat sensor. The data from this module will be sent to the Analog Digital Converter to convert a [continuous quantity](http://en.wikipedia.org/wiki/Continuous_signal) to a [discrete time](http://en.wikipedia.org/wiki/Discrete_signal) representation in [digital](http://en.wikipedia.org/wiki/Digital) form and will be sent to microcontroller and use LCD for quick display before transfer data to computer via ESP8266. Finally Arduino Basic program will view the output from the devices connection.

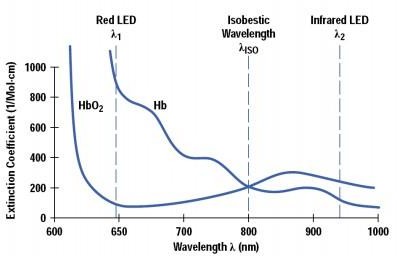
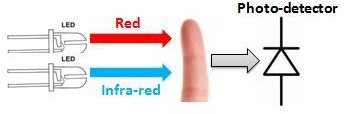


Figure 3.1:Two different wavelength

The optical sensor is consisted of two different wavelength LEDs and a photo-detector for receiving the light coming from the finger. In Figure 2, the probe structure has been shown.

 Figure3.2:Photo Detector Reaction

The small amplitude analog current coming from the photo-detector needs to be amplified by the transimpedance amplifier (TIA) and then processed by a filter. The microcontroller is one of the most important parts in our design. We programmed it in order to do the necessary calculations to measure the heart rate and oxygen saturation level.

## 3.5 Hardware and Software technology background

# **3.5.1 The Arduino Uno Board**

# The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. It's an open-source physical computing platform based on a simple microcontroller board, and a development

environment for writing software for the board.

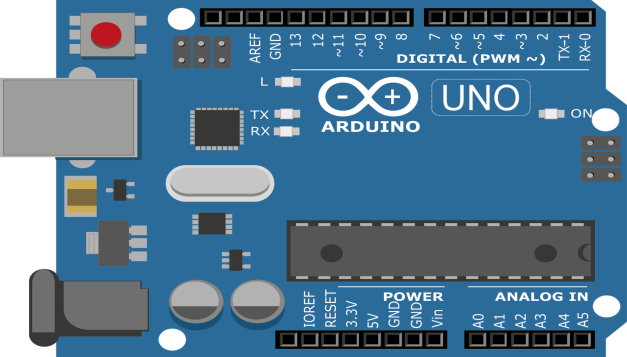


Figure3.3: Pin out diagram of the Arduino Board

## 3.5.2 Heart Rate Sensor

Heart beat sensor is designed to give digital output of heat beat when a finger is placed on it. When the heart beat detector is working, the beat LED flashes in unison with each heart beat. This digital output can be connected to microcontroller directly to measure the Beats per Minute (BPM) rate. It works on the principle of light modulation by blood flow through Fig 3. At each pulse [13].

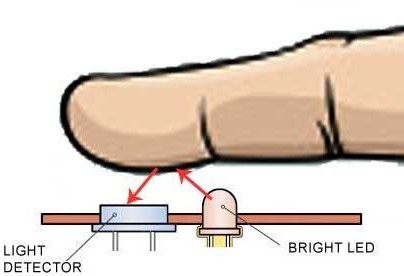


Figure3.3:How Sensor works

**3.5.3 The Pulse Sensor Unit**

A Heartbeat sensor is a monitoring device that allows one to measure his or her heart rate in real time or record the heart rate for later study. It provides a simple way to study the heart function. This sensor monitors the flow of blood through the finger and is designed to give digital output of the heartbeat when a finger is placed on it. When the sensor is working, the beat LED flashes in unison with each heartbeat. This digital output can be connected to the microcontroller directly to measure the Beats per Minute (BPM) rate. It works on the principle of light modulation by blood flow through finger at each pulse [7]. The Pulse Sensor is a well-designed plug- and-play heart-rate sensor for Arduino. It also includes an open-source monitoring app that graphs your pulse in real time.

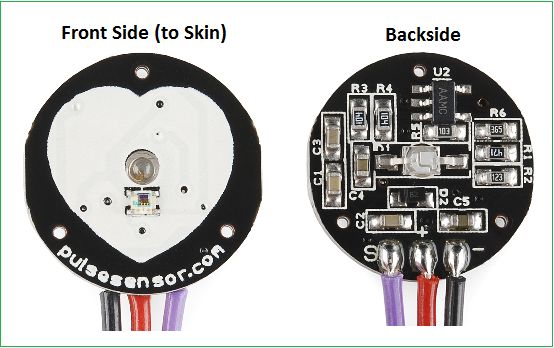


Figure3.4: Pin configuration of pulse sensor

**3.5.4 LCD Display Unit**

Liquid Crystal Display (LCD) modules that display characters such as text and numbers are the most cheapest and simplest to use of all LCDs. They can be purchased in various Sizes, which are measured by the number of rows and columns of characters they can display. Any LCD with an HD44780- or KS0066-compatible interface is compatible with Arduino. A 16x2 LCD display is very basic electronic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments](http://www.engineersgarage.com/content/seven-segment-display) and other multi segment [LED](http://www.engineersgarage.com/content/led)s because they are economical, easily programmable, has no limitation of displaying special and even [custom](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) [characters](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](http://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so on . A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines.

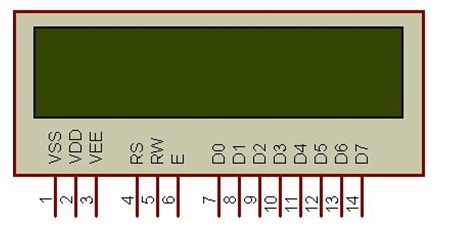


FIGURE 3.5: Pin Out of the LCD

**3.5.5 ESP8266:**

 ESP8266 has 8 pins, 4 in the row of 2. The first pin on the top left is GND. The two pins right from the GND are GPIO 2 and 0. I'm not going to use these pins, as they are not important for the operation. The pin on the top right side is the RX pin and the pin on the lower left is TX. These are the pins for communication. The middle pins on the bottom are CH\_PD (chip power-down) and RST (reset).

**The main thing to remember is, that this device works with 3.3V; even the RX and TX pins. Arduino or many USB to serial converters work with 5V. The solution for this project is in the next step.**

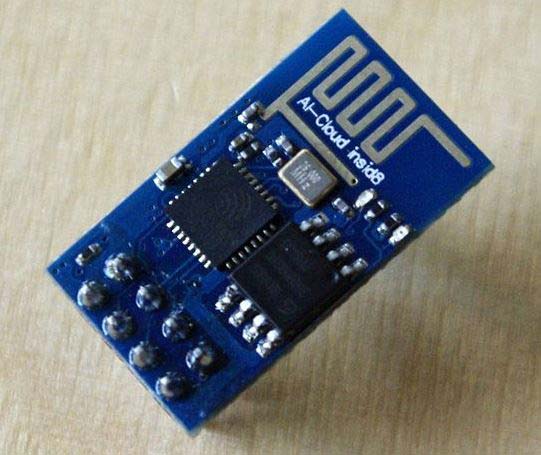


Figure3.6:ESP8266

## 3.6 Summary

After going throughout all this data collections, and by referring to the previous project that have been done, it’s decided to proceed with the project. Based on the author research and study, this HRM project benefits from other previous project also, the idea to enhance the project can be obtained from the input from the supervisor and the reader.

**CHAPTER THREE**

**RESEARCH METHODOLOGY**

## 4.1 Introduction

In this research methodology, researcher must select the appropriate methodology for this project. The methodology chosen will guide the researcher in finishing the project successfully and achieving its objectives.

Once the topic is chosen, a lot of study will be made how to solve possible problem such as this research will need a lot of testing and analysis in order to gather as much as possible information about how to develop a Heart Rate Monitor system. In this design phase, a single approach has been decided to be used, which is implementation.

**4.2 Design**

In this phase, student needs to design and choose the appropriate hardware and software to build the Heart Rate Monitor. The circuits must be built with a right schematic so that it can connect smoothly with the computer to display the signal output in beats per minutes. A schematic shows how electronic components are connected in an electronic circuit.

Like a map, it uses symbols to represent components, such as resistors and lines to show how those components are connected to one another.

The student needs to prepare with all related component and devices such as PCB (printed circuit board), ARDUINO microcontroller, power supply, heart beat sensor module, computer with coding and others. The construction is an important phase because it has to meet the requirement and must be function. The development of the system is discussed in next chapter

**4.3 Block Diagram**

PC

LCD DISPLAY (2x16)

Reset Button

Selector Switch MODE

ARDUINO UNO R3

Heart Beat sensor with OP-AMP

LM 358

|  |  |  |
| --- | --- | --- |
|  | |  |
| ESP8266( Wifi  module) |  |
|  |
|  | |  |

Figure 4.1: HRM block diagram

## 4.4 Hardware Requirements

The hardware part will be explained first as this project comprise of several important hardware and the fact that the hardware part are designed and built first before anything else. Below show the hardware that was used to build this project:

* ARDUINO UNO
* LDR sensor
* Potentiometer
* LED
* Wi-Fi Module
* DB9 male/female
* Resonator 4Mhz
* PCB board
* LCD 2x16
* Resistor/Variable Resistor

**4.5 Introduction to Sensor**

A sensor is a device that produces a measurable response to a change in a physical condition, such as temperature or thermal conductivity, or to a change in chemical concentration. Sensors are particularly useful for making in measurements such as in industrial process control.

Sensors are an important part to any measurement and automation application. The sensor is responsible for converting some type of physical phenomenon into a quantity measurable by a data acquisition (DAQ) system.

Factors to consider when choosing a sensor:

* Accuracy - The statistical variance about the exact reading.
* Calibration - Required for most measuring systems since their readings will drift over time.
* Environmental - Sensors typically have temperature and/or humidity limits.
* Range - Limits of measurement or the sensor.
* Repeatability - The variance in a sensor's reading when a single condition is repeatedly measured.
* Resolution - The smallest increment the sensor can detect.

**4.6 Arduino Uno**

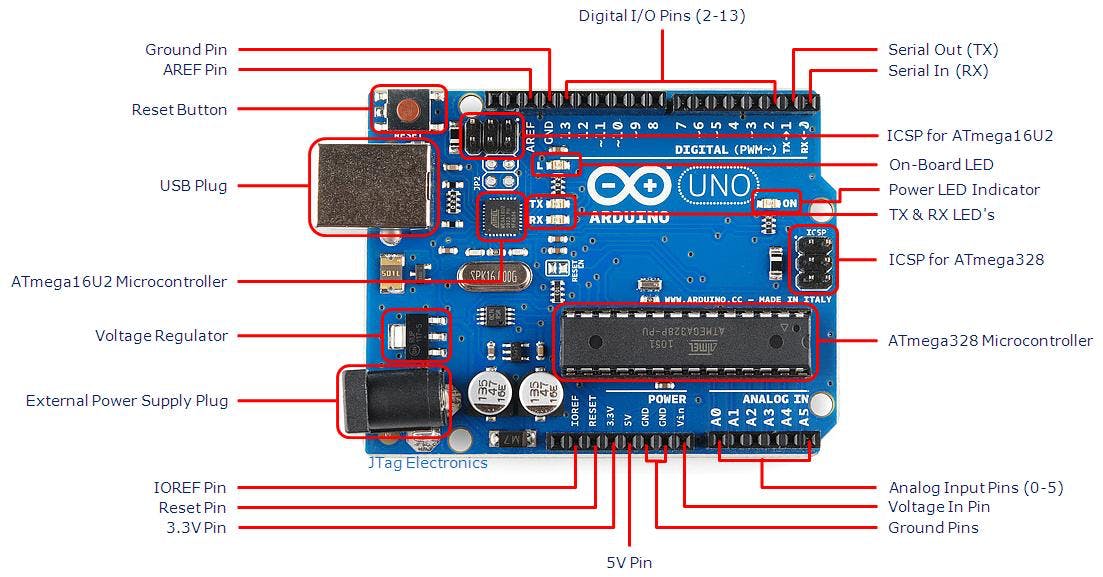


Figure 4.2: Pin out diagram of ARDUINO UNO

The Arduino Uno is programmed using the Arduino Software (IDE) our Integrated Development Environment common to all our boards and running both online and offline.

For programming this project use Arduino Uno. The reason why choose to use this type is because of the low cost and wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.

Heart Beat Sensor

* Capable to detect heart pulse base on illumination change during blood pressure. Heart beat sensor works on a very basic principle of optoelectronics. All it takes to measure you heart rate is a pair of LED and LDR and a microcontroller.

**LED:** A Light emitting diode (LED) is a two lead semiconductor light source.it is an p-n junction diode that emits light when activated. When a suitable current is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

Here we can use two LED .One is Red and one is Green for counting the heartbeat. When Heartbeat is counting those LED are continuing their light in phases on and off. So we can understand easily to counting the patient heartbeat.

**Resistor:** A Resistor is a passive two terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to device voltages, bias active elements, and terminate transmission lines among other uses.

Here we can use two Resistors. One is 220ohm resistors and another is 1k resistors. Those resistors are in need of circuit under circumstances.

**220ohm Resistors:** 220ohm are used to limit current flow, adjust signal levels, to device voltages, bias active elements, and terminate transmission lines among other uses.

**1k ohm Resistors:** 1k ohm Resistors are used to same thing like as a 220ohm resistor. Which also used to limit current flow, adjust signal levels, to device voltages, bias active elements, and terminate transmission lines among other uses.

**PCB Board:** A printed circuit board mechanically supports and electrically connects electronic components or electrical components using conductive tracks, pads and other features etched from one or more sheet layers of copper laminated onto and/or between sheet layers of a nonconductive substrate.

Here we can use a PCB board to design our heart beat monitoring circuit.

**Potentiometer:** A potentiometer is a three terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

Her we can use a potentiometer for lightening the LCD brightness.

**4. 7 ESp8266**:

The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware.

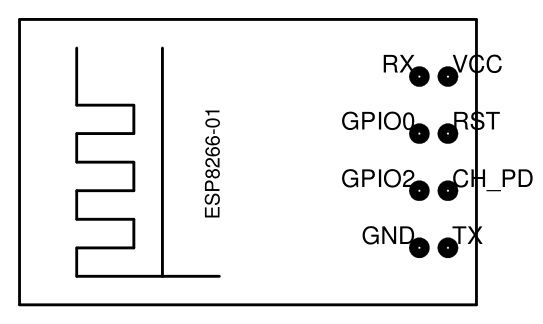


Figure4.3: ESP8266

For this project the ESP8266 module functions as an interfacing unit between the Arduino microcontroller and the computer. By using ESP8266, it enables communication between the computer and the microcontroller via serial port.

## 4.8Budget and Costing

Table 4.3 below shows the estimated budget to buy all the equipment’s needed in order to initiate the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Item** | **Price (RM)** | **Quantity** | **Total (RM)** |
| **HARDWARE** | | | | |
| **1** | Laptop Asus | - | 1 | - |
| **2** | LDR sensor | 450.00 | 1 | 450.00 |
| **3** | Bread Board | 120.00 | 1 | 120.00 |
| **4** | Potentiometer | 120.00 | 1 | 120.00 |
| **5** | ESP8266 Interface IC | 220.00 | 1 | 220.00 |
| **6** | DB9 male/female | 60.00 | 1 | 60.00 |
| **7** | Resistors 220 ohm | 1.00 | 5 | 5.00 |
| **8** | Arduino Uno R3 | 450.00 | 1 | 450.00 |
| **9** | LCD 2x16 | 120.00 | 1 | 120.00 |
| **10** | Resistor 10k | 1.00 | 5 | 5.00 |
| **SOFTWARE** | | | | |
| **11** | Windows 10 | - | 1 | - |
| **12** | Proteus software | Freeware | 1 | - |
| **13** | Arduino IDE | Freeware | 1 | - |
| **TOTAL** | | | | 1550.00 |

Table 4.3: Budget and Costing

## CHAPTER FIVE

## PROTOTYPE/PRODUCT DEVELOPMENT

**5.1 Introduction**

Based on the methodology explained in Chapter III, it is shown that are many aspects of the project which need to be carefully considered before moving to the development stage. This chapter explains the prototype development for Heart Rate Monitor software and hardware. It explains the prototype modules and system interface used in this development.

**5.2 Hardware Design**

Circuit Diagram:

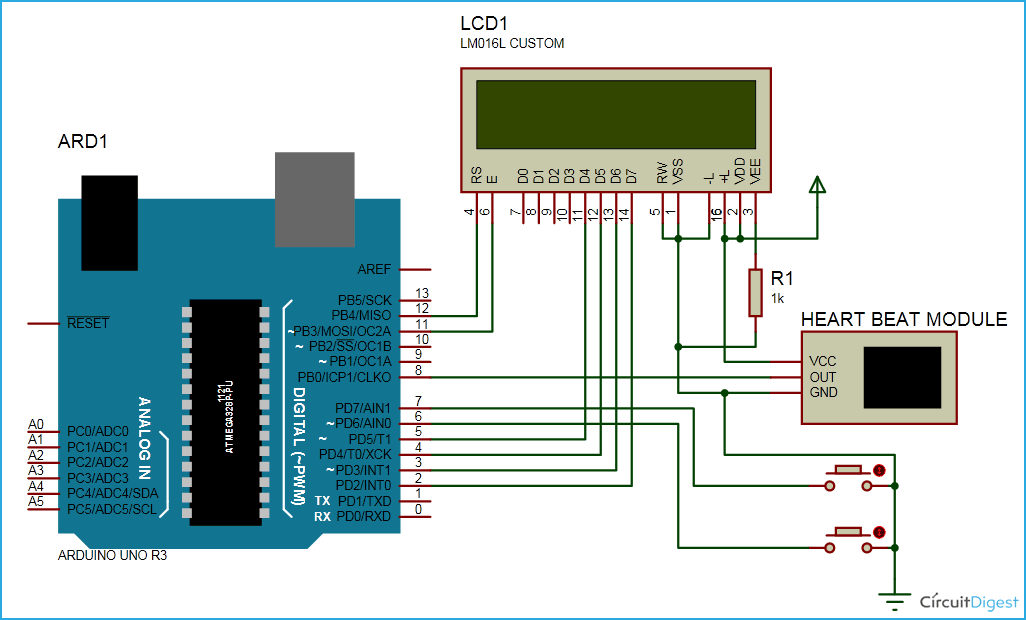


Figure5.1:Circuit Diagram of HRM

# First of all we will connect the Pulse Sensor with the Arduino. Pulse sensor has three pins. Connect 5V and the ground pin of the pulse sensor to the 5V and the ground of the Arduino and the signal pin to the 8 of Arduino.

Now we will connect LCD with the Arduino.

The connections of the LCD are as follows

Connect pin 1 (VEE) to the ground

Connect pin 2 (VDD or VCC) to the 5V.

Connect pin 4 (RS) to the pin 12 of the Arduino.

Connect pin 5 (Read/Write) to the ground of Arduino. This pin is not often used so we will connect it to the ground.

Connect pin 6 (E) to the pin 11 of the Arduino. The RS and E pin are the control pins which are used to send data and characters.

The following four pins are data pins which are used to communicate with the Arduino.

Connect pin 11 (D4) to pin 5 of Arduino.

Connect pin 12 (D5) to pin 4 of Arduino.

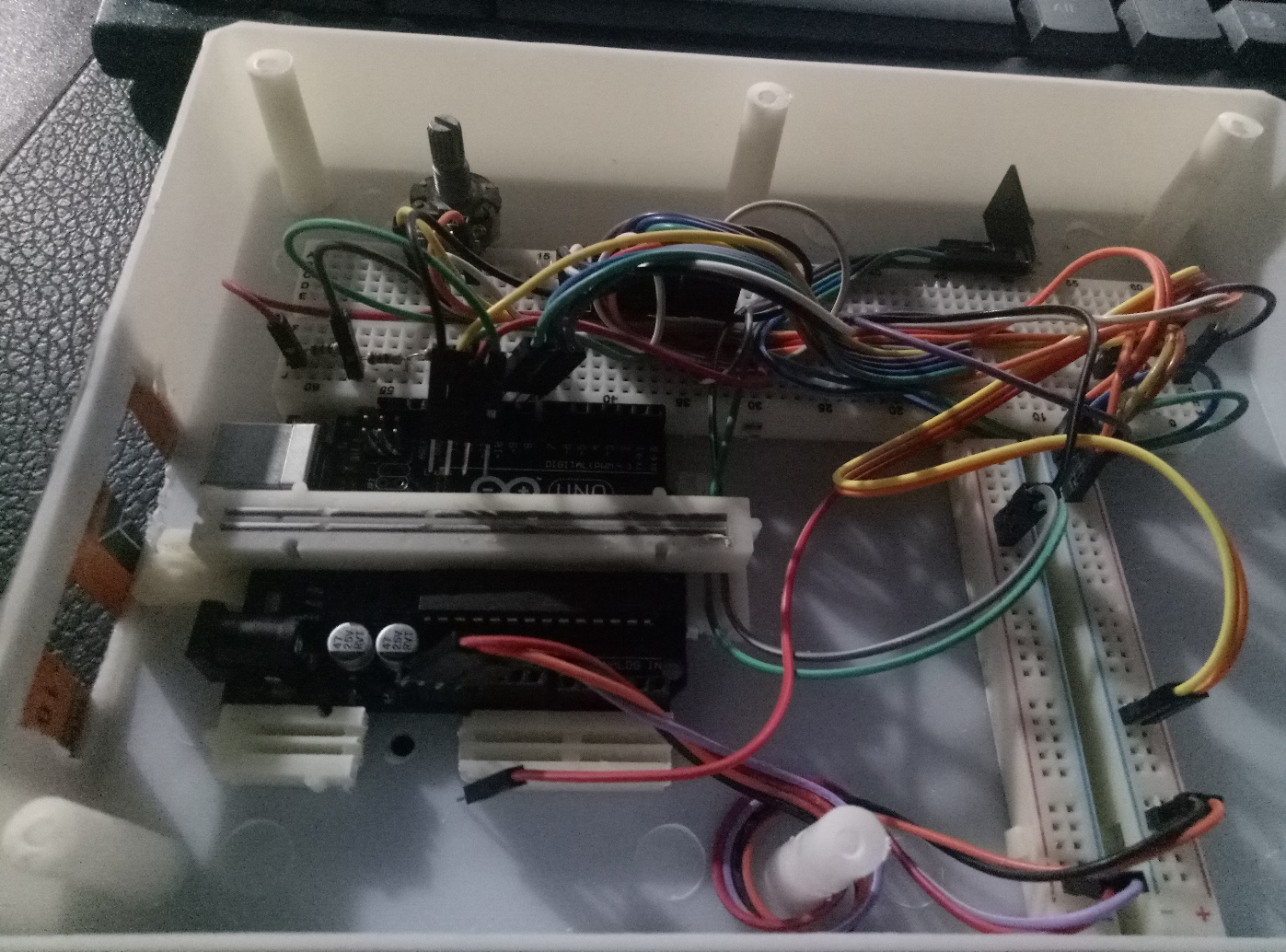
Connect pin 13 (D6) to pin 3 of Arduino.

Connect pin 14 (D7) to pin 2 of Arduino.

Connect pin 3 to the VCC through the 1k ohm resistor.

Connect pin 16 to the VCC

**5.3 Project Set Up:**

****

****

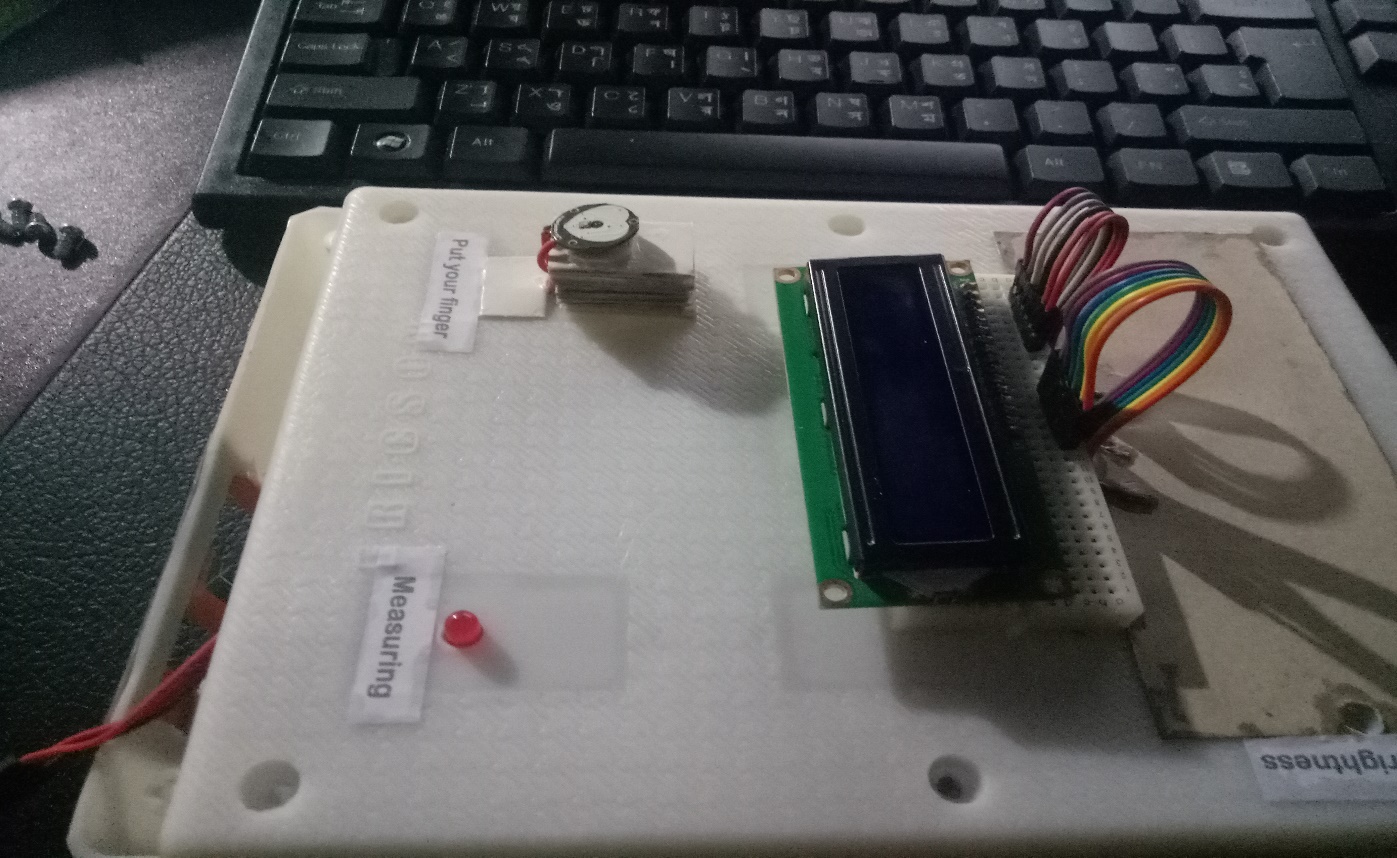
****

Figure 5.2:Final project setup

**5.3 Thing Speak Setup:**

Thing Speak provides very good tool for IoT based projects by using Thing Speak site, we can monitor our data and control our system over the Internet, using the Channels and webpages provided by Thing Speak. Thing Speak **‘Collects’** the data from the sensors, **‘Analyze and visualize’** the data and **‘Acts’** by triggering a reaction. First of all, user needs to create a Account on ThingSpeak.com, then **Sign In and click on Get Started**.

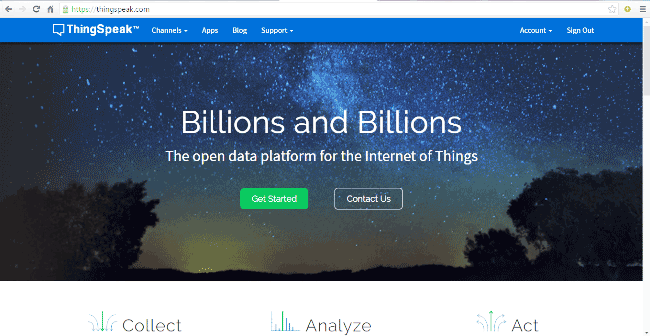


Figure5.3: Thing speak website

After creating an account, go to channels and **create a new channel** Now write the name of the Channel and name of the Fields. Also tick the check box for ‘Make Public’ option below in the form and finally Save the Channel. Now new channel has been created.

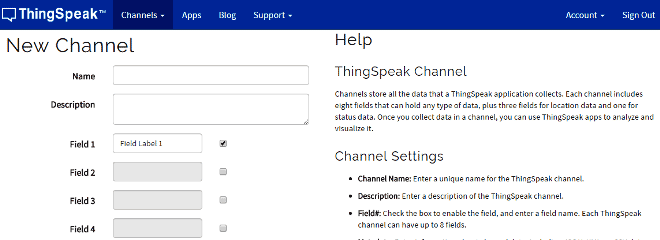


Figure5.4: Thing speak channel setup

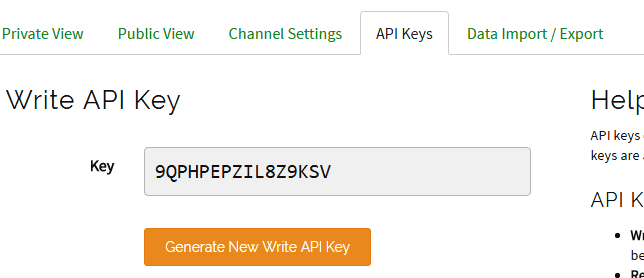


Figure 5.5: Thing speak key generate

After this go to API keys and copy of Write API key. We will need this in the code.

**5.4 Working Explanation:**

First we need to attach the Pulse Sensor to any organ of body where it can detect the pulse easily like finger, check the video below. Then the Pulse Sensor will measure the change in volume of blood, which occurs when every time heart pumps blood in the body. This **change in volume of blood causes a change in the light intensity** through that organ. The Arduino will then convert this change into the heart beat per minute (BPM). The LED connected at pin 13 will also blink according the Heart Beat.

The ESP8266 will then communicate with the Arduino and will send the data to ThingSpeak. The ESP8266 will connect the network of your router that you will provide in the code and will send the data of the sensor online. This data on the ThingSpeak will be shown in a Graph form showing the past readings too and can be accessed from anywhere over internet. The LCD connected will also show you the BPM.

**5.5 Software Design**

**Proteus Design:**

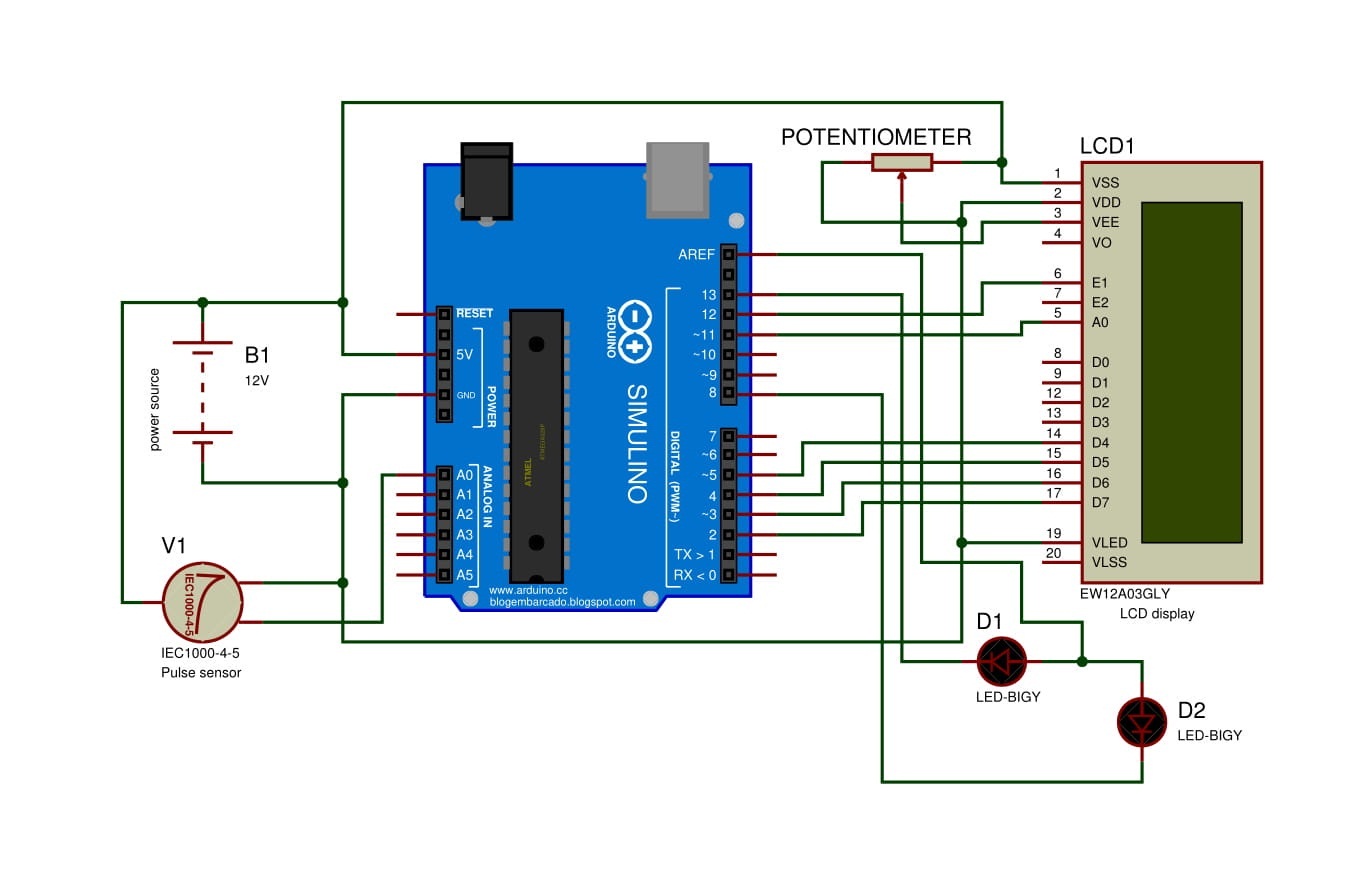


Figure5.6: Proteus design for Heart beat Monitoring

In this Section, The Proteus software is involved .Here First of all we design our total project circuit is in proteus for testing purpose. After complete the circuit design we are connecting the wire and other things. Such as power supply, switch etc. After all the connection setup correctly we can go to for test the whole circuit. When we go to for test the result or the heart rate shown by LCD display. And we see a value in LCD.

**Arduino IDE:**

In this section, the software that involved in the project is discussed. Arduino IDE used to program the Arduino microcontroller and Wi-Fi module to transfer data from the terminal receiver to the terminal receiver. The connection both of them should be programmed first by using Arduino IDE software.



Figure5.:Arduino IDE

Thing speak is used to display the result obtained from the project. Finally, the data will be displayed in Thing speak web which can be monitored continuously by the doctors, nurses or caregivers.

## 4.6 Coding:

The coding part is most important part for us. Because of, this code is communicate with microcontroller and LCD. Which can control all the things of processing data, sending data to LCD etc. So we can write CODE for this project. Here we write crystal programming, microcontroller programming, LED programming and of all things which we need.

**Code:**

#include <SoftwareSerial.h>  
#define DEBUG true  
SoftwareSerial esp8266(9,10);   
#include <LiquidCrystal.h>  
#include <stdlib.h>  
LiquidCrystal lcd(12,11,5,4,3,2);

#define SSID "Your Wifi Name"     // "SSID-WiFiname"   
#define PASS "Your Wifi Password"       // "password"  
#define IP "184.106.153.149"// thingspeak.com ip  
String msg = "GET /update?key=9YS21NU0HY5YS1IKU"; //change it with your api key like "GET /update?key=Your Api Key"

//Variables  
float temp;  
int hum;  
String tempC;  
int error;  
int pulsePin = 0;                 // Pulse Sensor purple wire connected to analog pin 0  
int blinkPin = 13;                // pin to blink led at each beat  
int fadePin = 5;  
int fadeRate = 0;

// Volatile Variables, used in the interrupt service routine!  
volatile int BPM;                   // int that holds raw Analog in 0. updated every 2mS  
volatile int Signal;                // holds the incoming raw data  
volatile int IBI = 600;             // int that holds the time interval between beats! Must be seeded!   
volatile boolean Pulse = false;     // "True" when heartbeat is detected. "False" when not a "live beat".   
volatile boolean QS = false;        // becomes true when Arduino finds a beat.

// Regards Serial OutPut  -- Set This Up to your needs  
static boolean serialVisual = true;   // Set to 'false' by Default.  Re-set to 'true' to see Arduino Serial Monitor ASCII Visual Pulse   
volatile int rate[10];                    // array to hold last ten IBI values  
volatile unsigned long sampleCounter = 0;          // used to determine pulse timing  
volatile unsigned long lastBeatTime = 0;           // used to find IBI  
volatile int P =512;                      // used to find peak in pulse wave, seeded  
volatile int T = 512;                     // used to find trough in pulse wave, seeded  
volatile int thresh = 525;                // used to find instant moment of heart beat, seeded  
volatile int amp = 100;                   // used to hold amplitude of pulse waveform, seeded  
volatile boolean firstBeat = true;        // used to seed rate array so we startup with reasonable BPM  
volatile boolean secondBeat = false;      // used to seed rate array so we startup with reasonable BPM

void setup()  
{  
  lcd.begin(16, 2);  
  lcd.print("circuitdigest.com");  
  delay(100);  
  lcd.setCursor(0,1);  
  lcd.print("Connecting...");  
  Serial.begin(9600); //or use default 115200.  
  esp8266.begin(9600);  
  Serial.println("AT");  
  esp8266.println("AT");  
  delay(5000);  
  if(esp8266.find("OK")){  
    connectWiFi();  
  }  
  interruptSetup();   
}

void loop(){  
  lcd.clear();  
  start: //label   
  error=0;  
  lcd.setCursor(0, 0);  
  lcd.print("BPM = ");  
  lcd.print(BPM);  
  delay (100);  
  lcd.setCursor(0, 1); // set the cursor to column 0, line 2  
  delay(1000);  
  updatebeat();  
  //Resend if transmission is not completed   
  if (error==1){  
    goto start; //go to label "start"  
  }  
    
  delay(1000);   
}

void updatebeat(){  
  String cmd = "AT+CIPSTART=\"TCP\",\"";  
  cmd += IP;  
  cmd += "\",80";  
  Serial.println(cmd);  
  esp8266.println(cmd);  
  delay(2000);  
  if(esp8266.find("Error")){  
    return;  
  }  
  cmd = msg ;  
  cmd += "&field1=";     
  cmd += BPM;  
  cmd += "\r\n";  
  Serial.print("AT+CIPSEND=");  
  esp8266.print("AT+CIPSEND=");  
  Serial.println(cmd.length());  
  esp8266.println(cmd.length());  
  if(esp8266.find(">")){  
    Serial.print(cmd);  
    esp8266.print(cmd);  
  }  
  else{  
   Serial.println("AT+CIPCLOSE");  
   esp8266.println("AT+CIPCLOSE");  
    //Resend...  
    error=1;  
  }  
}

boolean connectWiFi(){  
  Serial.println("AT+CWMODE=1");  
  esp8266.println("AT+CWMODE=1");  
  delay(2000);  
  String cmd="AT+CWJAP=\"";  
  cmd+=SSID;  
  cmd+="\",\"";  
  cmd+=PASS;  
  cmd+="\"";  
  Serial.println(cmd);  
  esp8266.println(cmd);  
  delay(5000);  
  if(esp8266.find("OK")){  
    Serial.println("OK");  
    return true;      
  }else{  
    return false;  
  }  
}

void interruptSetup(){       
  TCCR2A = 0x02;     // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE  
  TCCR2B = 0x06;     // DON'T FORCE COMPARE, 256 PRESCALER   
  OCR2A = 0X7C;      // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE  
  TIMSK2 = 0x02;     // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A  
  sei();             // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED        
}

ISR(TIMER2\_COMPA\_vect){                       // triggered when Timer2 counts to 124  
  cli();                                      // disable interrupts while we do this  
  Signal = analogRead(pulsePin);              // read the Pulse Sensor   
  sampleCounter += 2;                         // keep track of the time in mS  
  int N = sampleCounter - lastBeatTime;       // monitor the time since the last beat to avoid noise

    //  find the peak and trough of the pulse wave  
  if(Signal < thresh && N > (IBI/5)\*3){      // avoid dichrotic noise by waiting 3/5 of last IBI  
    if (Signal < T){                         // T is the trough  
      T = Signal;                            // keep track of lowest point in pulse wave   
    }  
  }

  if(Signal > thresh && Signal > P){        // thresh condition helps avoid noise  
    P = Signal;                             // P is the peak  
  }                                         // keep track of highest point in pulse wave

  //  NOW IT'S TIME TO LOOK FOR THE HEART BEAT  
  // signal surges up in value every time there is a pulse  
  if (N > 250){                                   // avoid high frequency noise  
    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)\*3) ){          
      Pulse = true;                               // set the Pulse flag when there is a pulse  
      digitalWrite(blinkPin,HIGH);                // turn on pin 13 LED  
      IBI = sampleCounter - lastBeatTime;         // time between beats in mS  
      lastBeatTime = sampleCounter;               // keep track of time for next pulse

      if(secondBeat){                        // if this is the second beat  
        secondBeat = false;                  // clear secondBeat flag  
        for(int i=0; i<=9; i++){             // seed the running total to get a realistic BPM at startup  
          rate[i] = IBI;                        
        }  
      }

      if(firstBeat){                         // if it's the first time beat is found  
        firstBeat = false;                   // clear firstBeat flag  
        secondBeat = true;                   // set the second beat flag  
        sei();                               // enable interrupts again  
        return;                              // IBI value is unreliable so discard it  
      }     
      word runningTotal = 0;                  // clear the runningTotal variable

      for(int i=0; i<=8; i++){                // shift data in the rate array  
        rate[i] = rate[i+1];                  // and drop the oldest IBI value   
        runningTotal += rate[i];              // add up the 9 oldest IBI values  
      }

      rate[9] = IBI;                          // add the latest IBI to the rate array  
      runningTotal += rate[9];                // add the latest IBI to runningTotal  
      runningTotal /= 10;                     // average the last 10 IBI values   
      BPM = 60000/runningTotal;               // how many beats can fit into a minute? that's BPM!  
      QS = true;                              // set Quantified Self flag   
      // QS FLAG IS NOT CLEARED INSIDE THIS ISR  
    }                         
  }

  if (Signal < thresh && Pulse == true){   // when the values are going down, the beat is over  
    digitalWrite(blinkPin,LOW);            // turn off pin 13 LED  
    Pulse = false;                         // reset the Pulse flag so we can do it again  
    amp = P - T;                           // get amplitude of the pulse wave  
    thresh = amp/2 + T;                    // set thresh at 50% of the amplitude  
    P = thresh;                            // reset these for next time  
    T = thresh;  
  }

  if (N > 2500){                           // if 2.5 seconds go by without a beat  
    thresh = 512;                          // set thresh default  
    P = 512;                               // set P default  
    T = 512;                               // set T default  
    lastBeatTime = sampleCounter;          // bring the lastBeatTime up to date          
    firstBeat = true;                      // set these to avoid noise  
    secondBeat = false;                    // when we get the heartbeat back  
  }

  sei();       
  // enable interrupts when youre done!  
}// end isr

## 

## CHAPTER SIX

## TESTING ANALYSIS

**6.1 Introduction**

This chapter discusses the testing method employed in the development of Heart Rate Monitor system. Various testing method relevant to the project are discussed. Result analysis where the analysis includes all the selected guidelines considered relevant to the prototype quality. Testing is important since the primary purpose is to detect software and hardware failures such that the defects may be uncovered and corrected.

## 6.2 Heart Beat Sensor

Heart beat sensor is design to give digital output of heart beat when finger is placed on it. The IR LED is use to illuminate a human finger with infrared light. The light intensity is then modulated by blood pressure changes within the finger before striking the photo resistor or LDR. The sensor then converts the changing light intensity into a proportional voltage containing two components below:

* A large DC offset corresponding to the average light intensity
* A small varying signal caused by changing blood pressure

When the heart beat detector is working, the beat LED flashes in synchronization to the human heart beat. It works on the principle of light modulation by blood flow through finger at each pulse. In order to make sure that the heart beat sensor was successfully design and developed, 2 method of testing have been done as shown in the picture below.

## 6.3 Arduino UNO Controller

Before continue to the software development of Arduino UNO source code, the

Hardware functionality of Arduino UNO controller must be working in perfect condition otherwise we will facing a lot of problem during software testing and development stages. There are several important testing that can be used to prove that the hardware is working in accordance to the design.

## 6.4 Software Tools to Write Program.

Download the Arduino Software (IDE)

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

When the download finishes, proceed with the installation

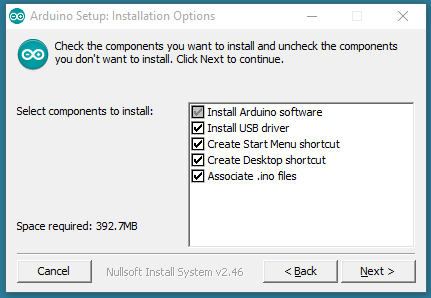


Figure6.1:Installing options

than choosing the components to install

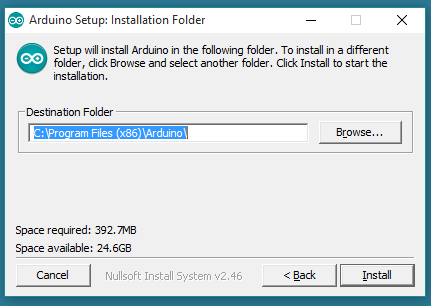


Figure6.2:Inslattion folder

now choosing the installation directory

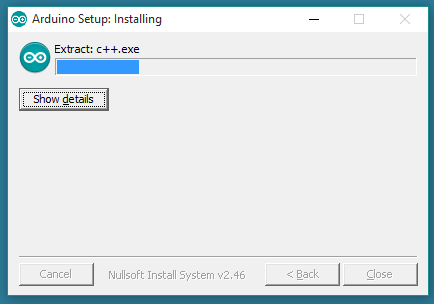


Figure6.3: Installing

The process will extract and installing all the required files to execute properly the Arduino Software (IDE)

Now open the LED blink example sketch: **File > Examples >01.Basics > Blink**.

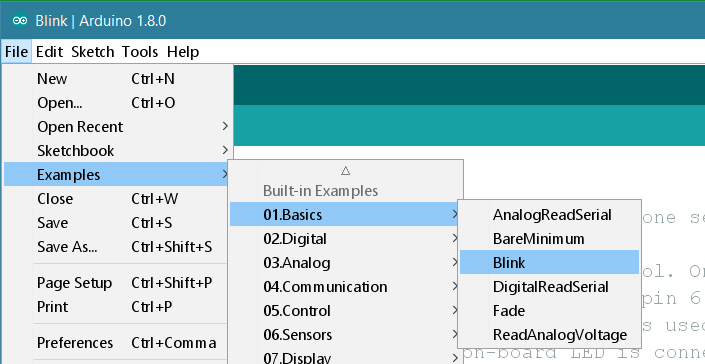


Figure6.4: Basics Blink

Now it will be need to select the entry in the **Tools > Board** menu that corresponds to our Arduino or Genuino board.

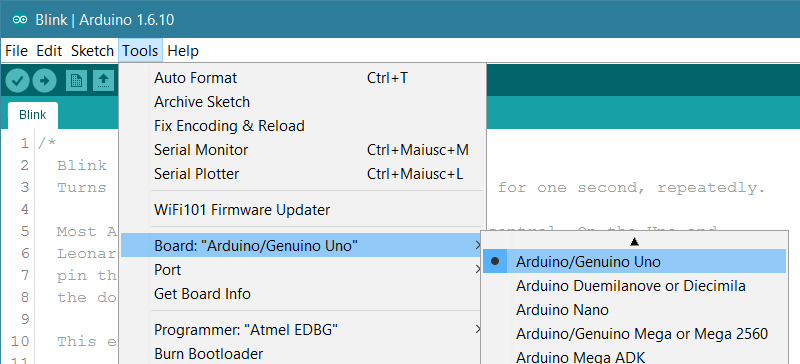


Figure6.5: Basic Tools Board

Selecting the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

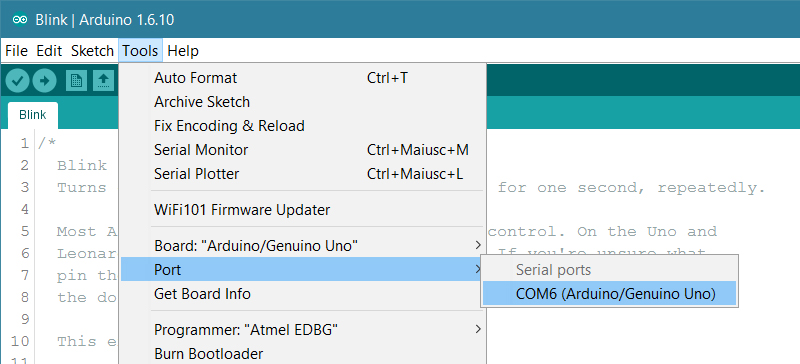


Figure6.6: Basics Port setting

This time to upload the program. Now we click the "Upload" button in the environment. Waiting a few seconds – we could see the RX and TX lads on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

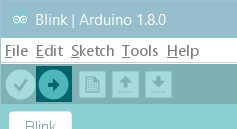


Figure6.7: Uploading Program

A few seconds after the upload finishes, we should see the pin 13 (L) LED on the board start to blink (in orange). If it does, we have successfully completed all the steps.

## 6.5 Uploading and Testing

The main function of the Arduino Uno controller are to detect the pulse rate signal generated by the heart beat sensor , counting the pulse in order to get the BPM result and transmit the data to the PC through ESP8266 serial interfacing circuit. Then, the reply status from PC will be display to LCD unit.

* Wrap the strap around the finger or wrist.
* Now, upload the program for testing the module for the first time.
* Open serial plotter and wait for 5 seconds and stay calm. We will see the graph of the heart beats.
* Try to block the blood flow through using another hand.
* The graph line will become straight for some time and will come to normal stage as the blood finds way for circulation if all the capillaries aren't blocked properly.
* Now, upload the main code to the Arduino.
* Remember to change the API key, SSID (Wi-Fi name) and PASS (Password) for our project. Use the heart rate sensor on self again. The ESP8266 will then communicate with the Arduino and will send the data to Thing Speak.
* The ESP8266 will connect the network of our router that we will provide in the code and will send the data of the sensor online. This data on the Thing Speak will be shown in a Graph form showing the past readings too and can be accessed from anywhere over internet. The LCD connected will also show the BPM.

**** 

Figure6.8: BPM show on web and LCD

**CHAPTER SEVEN**

**CONCLUSION AND SUGGESTION**

**7.1 Introduction**

This chapter discusses the suggestion of future work for the project and conclusion will be made according to the project development. This thesis has discussed the development of the sensor module, main circuit and interfacing with computer.

## 7.2 Conclusion

The design of a low-cost microcontroller based device for measuring the heart pulse rate has been described. The device has the advantage that it can be used by non- professional at home to measure the heart rate easily and safely. With this system the patient will be more proactive with their doctor because it appears prevention is better than cure. Here I can say that this Heart Rate Monitor project have achieved all of its objectives.

An important factor to finish this project is the overall processes that have been considered. Enough knowledge is required along the process. Earlier preparation is needed for the components that are limited. More components should also be ready as a precaution from unpredictable situation such as components burned and not in good condition. Some suggestion has been made for the limitation of heart beat sensor accuracy due to the LDR mounting and light exposure (sensitive to light).

Lastly, taking into account issues related to the completion of the project, it is hereby suggested that the project be continued and further developed in the area of software and hardware enhancement.

## 7.3 Discussion

In order to improve the project function and implementation in the future, several

suggestions are proposed:

* + 1. Use of mobile phone or computer. Research should be done to find new method in order make this system functions using mobile phone or computer. If you use this the system will be more convenient and practical.
    2. Use of different types of techniques to detect the heart rate. There are number of methods and technique to detect the heart rate, for example ECG and optical sensor. This kind of method is more accurate than this project method. But the main problem is that, this method is more difficult to design and implement.

**7.4 Future works:**

* ECG, EEG and other health parameters can also be monitored.
* More than a single patient at a different places can monitored using the single system,
* Continuous monitoring future diagnosis can be performed via the same system.

**REFERENCES**

[1] [Evangelos Petroutsos,](http://www.goodreads.com/author/show/39876.Evangelos_Petroutsos)(1998), Mastering Visual Basic 6.0, In Sybex (Ed). 0782122728

[2] Peter Spasov (2005), Microcontroller Technology, Fifth edition, Sir Sandford Fleming College: Prentice Hall.

[3] Pulse Detector Through Difference Amplifier by Norwahidah Mohd Noor Fakulti Kejuruteraan Elektronik & Kejuruteraan Komputer Kolej Universiti Teknikal Kebangsaan Malaysia 2006

[4] [Juul Achten](http://link.springer.com/search?facet-author=%22Juul%2BAchten%22)., & [Asker E. Jeukendrup](http://link.springer.com/search?facet-author=%22Asker%2BE.%2BJeukendrup%22). (2003). Heart Rate Monitoring. Vol 33, Issue 7, pp 517-538.

[5] John, E. S ., Stephan , A. and David, C. (1993) Pulse Oximetry Through Spectral Analysis, University of South Florida Department of Anesthesiology, University of South Florida Center for Microelectronics Research.

[6] M. Malik, "Guidelines: Heart rate variability Standards of measurement, physiological interpretation, and clinical use", Eur Heart J., vol. 17, pp354–38, 1996.

[7] Microsoft Visual Basic 6 Deluxe Learning Edition Michael Halvorson (Author), Microsoft Press (Publisher)

[8] "The 6 Best Computer-Linked Pedometers to Buy in 2018", Verywell Fit, 2018. [Online]. Available: http://walking.about.com/od/prpedometer/fr/dashtrak.htm. [Accessed: 13- Jul- 2018].

[9]Final Year Project "Home Patient Monitor"", Idriah-safriza.blogspot.com, 2018. [Online]. Available: http://idriah-safriza.blogspot.com. [Accessed: 13- Jul- 2018].

[10] "Home | Jack Baskin School of Engineering", Classes.soe.ucsc.edu, 2018. [Online]. Available: http://classes.soe.ucsc.edu. [Accessed: 13- Jul- 2018].

[11] \2018. [Online]. Available: http://wwwnskelectronic.in/heart\_beat.html. [Accessed: 13- Jul- 2018].

[12] Sana M. Ceesay., Adrew M. Prentise., & Keeneth C.Day. (2002). The use of heart rate monitoring in the estimation of energy expenditure a validation study using indirect whole-body calorimetry, 175-186

[13] Sharief F. Babiker,. Liena Elrayah Abdel-Khair,. & Samah M. Elbasheer. (2011). Microcontroller Based Heart Rate Monitor using Fingertip Sensors.

UofKEJ Vol. 1 Issue 2 pp. 47-51

[14] Ww1.microchip.com, 2018. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/51290a.pdf. [Accessed: 13- Jul- 2018].

[15] V.Ryan., (2000-2008). Light Dependant Resistor(LDR).World Association Technology Teachers, (1-3), from

[16]"Project Report on Heartbeat Monitoring | Flash Memory | Monitoring (Medicine)", Scribd, 2018. [Online]. Available: https://www.scribd.com/doc/53795698/Project-Report-on-Heartbeat-Monitoring. [Accessed: 13- Jul- 2018].

[17] "Arduino - Home", Arduino.cc, 2018. [Online]. Available: https://www.arduino.cc/. [Accessed: 13- Jul- 2018].

[18] "Circuit Digest", Circuitdigest.com, 2018. [Online]. Available: https://circuitdigest.com/. [Accessed: 13- Jul- 2018].

[19] ESP8266", En.wikipedia.org, 2018. [Online]. Available:

https://en.wikipedia.org/wiki/ESP8266. [Accessed: 13- Jul- 2018].