DESIGN AND ANALYSIS OF WIRELESS HEALTHCARE MONITORING SYSTEM BASED ON IOT

SYAHMI AL-ZUBAIR BIN ABDULLAH

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DESIGN AND ANALYSIS OF WIRELESS HEALTHCARE MONITORING SYSTEM BASED ON IOT

SYAHMI AL-ZUBAIR BIN ABDULLAH

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia

2020

DECLARATION

I declare that this report entitled "Design and Analysis of Wireless Healthcare Monitoring System Based on IoT" is the result of my own work except for quotes as cited in the references.

Signature : ....

Author : SYAHMI AL-ZUBAIR BIN ABDULLAH

Date : 2 JULY 2020

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : ....

Supervisor Name : MOHD SHAHRIL IZUAN BIN MOHD ZIN

Date : 30 JUNE 2020

DEDICATION

Specially dedicated to my family, supervisor and my fellow friends that help me to finish my final year project report.

ABSTRACT

Technology plays an important role in the healthcare system today, not for measuring health parameters, but also for communication, monitoring and storage. This system's main objective is to continually track patient health status and use the internet connection to display information through the cloud. The research is using NodeMCU board to gather information on patient’s health. It uses temperature and pulse sensor to provide real-time monitoring for the doctor and family members. The sensors will be attached with controller board that needs to be worn by the patient for continuous monitoring. These data will be sent to the server using the internet wirelessly. The information can be accessed by the physician via the web page from the IoT system. If the system detects any abrupt changes in the temperature and pulse of the patient, the diagnosis will be sent via notification to the doctor. At the end of the project, analysis was conducted to test the performance of the system that covers sensor data accuracy and wireless transmission strength.

ABSTRAK

Teknologi memainkan peranan penting dalam sistem penjagaan kesihatan hari ini, bukan untuk mengukur parameter kesihatan, tetapi juga untuk komunikasi, pemantauan dan penyimpanan. Objektif utama sistem ini adalah untuk terus mengetahui status kesihatan pesakit dan menggunakan platform internet untuk memaparkan maklumat melalui awan. Penyelidikan ini menggunakan papan NodeMCU untuk mengumpulkan maklumat mengenai kesihatan pesakit. Ia menggunakan sensor suhu dan nadi untuk memberikan pemantauan masa nyata untuk doktor dan ahli keluarga. Sensor akan dipasang dengan papan kawalan yang perlu dipakai oleh pesakit untuk pemantauan berterusan. Data ini akan dihantar ke pelayan menggunakan internet tanpa wayar. Maklumat tersebut dapat diakses oleh doktor melalui laman web dari sistem IoT. Sekiranya sistem mengesan adanya perubahan suhu dan nadi pesakit secara tiba-tiba, diagnosis akan dihantar melalui pemberitahuan kepada doktor. Pada akhir proyek, analisis dilakukan untuk menguji prestasi sistem yang meliputi ketepatan data sensor dan kekuatan transmisi tanpa wayar

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all those directly or indirectly involved in helping me to complete the Final Year Project (FYP), which is a compulsory subject for all Universiti Technical Malaysia Melaka (UTeM) students to complete their degree.

First of all, I would like to thank my supervisor, En Mohd Shahril Izuan Bin Mohd Zin, for directing me continuously throughout this venture. Thank you for your expertise and spiritual advice that will bring success to this project. I am also encouraged by his planning actions to always prepare before I do something.

Next, special gratitude goes to my parents, who also gave words of encouragement and optimism followed by prayer and hope that I would go forward. Lastly, my friends who helped to provide inspiration, whether directly or indirectly, besides having to face the struggle together to complete these tasks that are entrusted to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

IoT	:	Internet of Things
ECG	:	Electrocardiogram
IFTTT	:	If This Then That
ADC	:	Analog-to-Digital Converter
DAC	:	Digital-to- Analog Converter
IP	:	Internet Protocol
LCD	:	Liquid-crystal display
PDA	:	personal digital assistant
GSM	:	Global System for Mobile Communications
BPM	:	Beat Per Minute
IC	:	Integrated Circuit
WLAN	:	Wireless Local Area Network
HTML	:	Hypertext Markup Language
PHP	:	Hypertext Preprocessor
MySQL	:	My Structured Query Language
PCB	:	Printed Circuit Board
RF	:	Radio Frequency

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Nowadays, technology becomes an important part of the healthcare system that can be used to monitor parameters of health. Continuous monitoring of the health of patients is necessary in a hospital to achieve their health conditions [1]. In addition, studies show that up to 90% of all hospital alarm systems for patients in a critical situation cannot be used [2]. This is due to the beeping of medical devices that fade into the background and sound of many alerts that may cause medical errors. Technology uses can improve the medical system with sensory devices where data can be stored in the cloud system and processed.

The Internet of Things (IoT) has an extensive applicability in numerous areas, including healthcare. IoT refers to the number of physical devices connected to the

Internet and is capable of sharing information with other entities. IoT healthcare device able to get health parameters such as glucose level, blood pressure, heartbeat rate, and body temperature using sensor devices [3]. These information can be stored in the cloud and can be exchanged with an authorized person, family members or physicians to allow them to track the collected health parameters with the aid of the IoT platform, regardless of place and time.

Recently, using wireless communication technology, the Internet has become a major impact in healthcare and other industries. These technologies can be expanded to include the hospital system properly [4]. For example, Wi-Fi network connectivity ensure data collected from each patient with the proper database system stored in the cloud system. In addition, the advanced sensors as well as wireless technologies are available to enable effective wearable medical devices that collect and transmit data to support remote patient monitoring. Hence, this wireless development therefore has a big impact on the healthcare industry, as it can provide the latest evidence of treatment.

1.2 Problem Statement

Healthcare resources in exposed areas are under extreme stress during a pandemic like COVID-19. As a result, certain treatments need to be postponed for patients with chronic conditions. This pandemic also cause restrictions and limitations for people including their regular healthcare facility visits [5]. Thus, this will affect their regular monitoring system.

1.3 Project Objective

The objectives of this project are:

i. To develop and design a system that can detect health data and transmit data wirelessly integrated with Internet of Things implementation for monitoring and notification system.

ii. To analyze the performance of the system for sensors accuracy and wireless transmission strength.

1.4 Scope of Work

The scope of this project is to develop a system that able to detect the health status of the patients and enable the user to monitor the status through webpage. The project is using NodeMCU board that integrates Wi-Fi connection to the Internet and connected sensors to detect health biometric parameters. The project will use two sensors, which are temperature sensor and pulse sensor. Development of each hardware module connected with the controller board for input and output operation will be created.

On software development, the project will cover C language coding to be uploaded into NodeMCU using Arduino IDE. A database platform known as MySQL will be created using PHP language to collect health data from the sensors and store the data through the database system. For IoT implementation, an Android application or webpage will be created using Android Studio for monitoring system through smartphone. Then, an integration of IFTTT platform with Gmail is created for push notification system. Next, analysis will be made to test the efficiency of the hardware system. A comparison for both sensors to check the percentage of error with real measurement devices. Finally, another analysis will be conducted to test the wireless signal strength by distance is measured.

1.5 Project Outline

Chapter 1 offers a brief explanation of the project and healthcare system background. Each section contains explanation of concerns, project goals and scope of research to complete the task.

Chapter 2 addresses the literature review in this section, where all the research and technical papers related to this project are included. Result analysis, circuit designs, type of sensors and IoT platforms included in previous research.

Chapter 3 explains the full methods used in the implementation of the project. The methodology chapter covered the programming language to communicate with the controller board and the setup as well as measurement of the sensor.

Chapter 4 presents all analytical data and results. This chapter will analyze and observe all the tests and conclusions. In this chapter, the results for evaluated system performance using an acceptable diagram will be presented.

The suggestions and future work will be described in Chapter 5 based on the completed project. This will summarize all the process conclusion that occurs during the completion of this study document.

CHAPTER 2

BACKGROUND STUDY

2.1 Introduction

The literature review is one of the ways in which field research has been fully understood. The first portion consists of a critical review of the previous researcher's project. In addition, this part consisted of other sources to endorse research and analysis claim. It needs the reader to return to this section if there is ambiguity about some of the words used for the entire research process in another paragraph.

2.2 Motivation of previous work

The previous author identified the IoT-based healthcare system. I find five articles from many sources that linked to this scope of the project. It was therefore selected in this section for a critical review and evaluation. The analysis includes all of the researcher's part of the project.

2.2.1 Healthcare monitoring system by Utilizing BLE-Based Sensors and Real Time Data Processing

From the study, the author suggests a personalized healthcare monitoring system that uses a BLE-based sensor array, real-time data processing and machine-based learning algorithms to help diabetic patients better manage their chronic condition [6]. BLEs were used to collect vital signs from sensor nodes to users' smartphones, while data processing in real-time was used to manage the large volume of continuously generated sensor data.

As a streaming network, Apache Kafka used the proposed real-time data storage, and MongoDB to store the sensor data of the patient. The findings show that commercial BLE-based sensor models and the proposed real-time data processing are adequately efficient for monitoring the vital signs of the data from diabetic patients. Additionally, classification methods based on machine learning were tested on a diabetes dataset and showed that provided the user's sensor data as input, a multilayer perceptron can provide prediction of early diabetes.

The results also show that Long Short-Term Memory can accurately predict the future BG level, based on the current sensor data. Hence, the current definition of diabetes and BG calculation should be paired with guidelines for tailor-made diet and physical activity to improve patient health safety and prevent potential critical conditions.

2.2.2 Wireless Patient Health Monitoring System

In this paper the researcher describes a wireless monitoring device that uses a microcontroller to achieve heart rate and temperature. The machine uses the Zigbee modem to wirelessly transfer all of the data to the computer system. It is mainly to be used at home, and it must be monitored continuously. The author used an operational amplifier, an LED, a picture and other components to construct the heartbeat sensor [7].

The development of the project was split into two parts that were transmitter and receiver. The sensor stored safety parameters in the transmitter and translated them into digital data for wireless transmission using ZigBee module It uses a microcontroller for this device to act as a brain. In eagle code, each part and hardware was built and undergoes the process of PCB creation. The author uses VB6 software in the receiver section to collect and store data from the section of the transmitter. Here, the USB-connected Zigbee module was used for data transmission to the PC. The doctor can adjust the limits of health parameters from the software design that can activate the alert system.

.

2.2.3 IoT Based Health Monitoring System Using Blynk App

The researchers propose this project to develop a healthcare system which will give body temperature, heart rate pulse using DS18b20 and pulse sensor as well. In addition, the ESP32 Dev. board controller is connected to the sensors and wireless transmission of information is handled through the Wi-Fi module using Arduino. The ESP32 controller is used for wireless IOT data transmission using the Blynk android app [8]. Digitizing the information visually on an Android app from Blynk and the data record of the patient will be kept for a period of time. To monitor the computer using the app, the information is stored using an Android app.

Controller ESP32 is developed and designed for functions such as wearable electronics, mobile phones and the Internet of Things. This lists all the modern features of low-consumption chips, including fine-grained clock gating, multiple power modes, adaptive energy scaling, etc. For example, low duty cycle is used to reduce the energy consumption dissipated by the chip. The power amplifier turn-out can also be changed. It therefore offers an appropriate trade mark in the mid-range of information, communication range and energy consumption. The ESP32 microcontroller is a combination of an attached Bluetooth chip and a single 2.4 GHz Wi-Fi built with a 40 nm TSMC ultra-low consumption process. It is basically designed to achieve improved power and RF output and performance robustness, power consumption, reliability on a huge acquisition basis, power scenes and flexibility.

Blynk is an open source android app designed and developed to monitor the hardware through the Internet of Things (IoT). It displays sensor data electronically, it can capture and show data. Additionally, it can also do other parameters including Blynk app which create amazing interfaces for a project using multiple widgets, Blynk server which is responsible for communication between smartphone and hardware, Blynk libraries that supports communication with the server and complete progression of incoming and outgoing instructions.

2.2.4 IoT-based Healthcare Monitoring System for War Soldiers using Machine Learning

The author creates a health care monitoring system in this paper that soldiers can use to obtain their health status, track location and presence of nearby explosive compounds. Each soldier's device wear is integrated with the Zigbee module for 10 squadron leader wireless transmission. The leader system has a module from LoRaWAN that uses long-range communication to send to the cloud [9].

The author uses K-Means Clustering data analytics methodology to forecast different types of soldier situation by means of data collected from the web portal. This uses the ThingSpeak database for the cloud system as it offers free processing and analysis of data and can provide visualization and analysis of data from MATLAB. To obtain health status of soldiers, it uses smart biomedical sensor attach on soldier jacket that can provide communication to the base station and soldier mobility.

2.2.5 Secured Smart Healthcare Monitoring System Based on IoT

This uses a microcontroller as an intermediate in this project to link sensors with the implementation of IoT [10]. Since security was the main problem due to unauthorized access to the system, the author encrypted a password-protected Wi-Fi module that only allows doctors or caregivers to access the data. . In a critical situation, the project also added GSM module to alert caretaker or doctor with SMS. Therefore, due to its low power consumption, fast response time and easy set-up, the proposed project will benefit more from medical service.

The author uses the ESP8266 Wi-Fi module for Wi-Fi protocol that allows doctors to access data via internet browser by entering the IP address. In addition, to track the patient and caregiver system. The healthcare variable was presented with an LCD display. On the human body there are two measurements that were body temperature and heartbeat. A digital thermometer was used to measure up to 9-bit resolution and 0.5 ° C precision. As it monitors heart rate based on blood flow rate, it uses pulse oximeter sensor for heartbeat. The research includes the C language for the microcontroller in software configuration.

2.3 Health parameters

In accordance with their health factor, the health system is critical for everyone to improve health. The health status, illness, and physical body structure will affect a person's quality of life. To know the health status of the patient, a calculation can be performed using the health parameter to test the symptoms and signs of the disease. These parameters include body temperature, blood pressure, heart rate and air frequency. Table 2.1 below demonstrate measurement of each health parameter with analyze parameters and type of measured value [11].

Table 2.1 Basic health parameter measurement

Health parameters	**Analyze parameter**	**Measured signal**
Body temperature	Temperature	Numerical value
Blood pressure	Systolic and diastolic	Time function and numerical value
Pulse rate	Heartbeat rate	Mean numerical rate (beat per minute)
Respiratory rate	Breathing rate	Numerical value (Breath time)
Electrocardiogram (ECG)	Electrical heart rate	Time function

However, most important parameters are heart rate and body temperature since both can define a person health state.

2.3.1 Body temperature

Body temperature is one of the key vital signs to be controlled in order to ensure safe and effective treatment [12]. To ensure that temperature is measured accurately, it is essential to use the most appropriate technique. Inaccurate results can influence diagnosis and treatment, lead to failure to identify deterioration of the patient and compromise the safety of the patient. The normal human body temperature range is typically stated as 36.5–37.5 °C (97.7–99.5 °F). Because of their nervous system that can control metabolism, their temperature can only switch about ±0.4 ° C for a healthy person. It may increase body temperature in extreme conditions such as changing environmental parameters or doing physical activity. The surrounding condition, age, activity level, and hormones are some intrinsic factors that can affect normal body temperature. Body temperature may be unstable due to 2 symptoms which is fever for high temperature and hypothermia for low temperature. To measure body temperature, there are several ways which are:

1. Orally. The temperature is measured by a person's mouth using a glass thermometer to achieve a reading.
2. Rectally. This method appears to provide more accurate reading for core temperature measurements, but the measurement through the rectum takes longer.
3. Axillary. Use a glass thermometer to take the temperature under the arm near to the wall of the chest.
4. Tympanic. It's a temperature measurement through the ear that can get the temperature of the ear. Operation is faster and can get the core temperature of the body directly.

The temperature of the human body varies over time from deep visceral tissues and the central nervous system to the outer body layer, as seen in Figure 2.1 [13]. The core temperature contracts less in a cold place while the core temperature increases in a warm place. Yellow areas in A, B, and C are "acral" regions that help regulate body temperature by narrowing the blood flow as body temperature decreases.

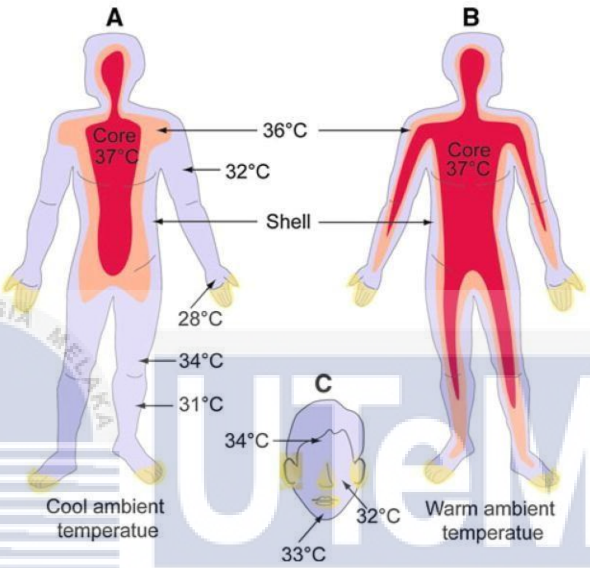


Figure 2.1 Core body temperature in cold and warm places

2.3.2 Heartbeat rate

Heartbeat speed, or so-called pulse rate, can be described as how many times a person's heart beast takes in one minute. The heart rate depends on the person's age, body size, heart condition, activity level, state of health and emotion. A normal pulse rate for a reading human ranges from 60 to100 beats per minute.

Based on Figure 2.2, chart of pulse rate varies by age and gender which is different and important for adults.

WOMEN'S RESTING HEART RATE CHART						
AGE	18 - 25	26 - 35	36 - 45	46 - 55	56 - 65	65+
ATHLETE	54-60	54-59	54-59	54-60	54-59	54-59
EXCELLENT	61-65	60-64	60-64	61-65	60-64	60-64
GOOD	66-69	65-68	65-69	66-69	65-68	65-68
ABOVE AV	70-73	69-72	70-73	70-73	69-73	69-72
AVERAGE	74-78	73-76	74-78	74-77	74-77	73-76
BELOW AV	79-84	77-82	79-84	78-83	78-83	77-84
POOR	85+	83+	85+	84+	84+	84+

MEN'S RESTING HEART RATE CHART						
AGE	18 - 25	26 - 35	36 - 45	46 - 55	56 - 65	65+
ATHLETE	49-55	49-54	50-56	50-57	51-56	50-55
EXCELLENT	56-61	55-61	57-62	58-63	57-61	56-61
GOOD	62-65	62-65	63-66	64-67	62-67	62-65
ABOVE AV	66-69	66-70	67-70	68-71	68-71	66-69
AVERAGE	70-73	71-74	71-75	72-76	72-75	70-73
BELOW AV	74-81	75-81	76-82	77-83	76-81	74-79
POOR	82+	82+	83+	84+	82+	80+

Figure 2.2 Chart of pulse rates

The heart rate can be determined in multiple locations of the human body at ankles, inside an elbow, chest hand, and foot edge. A person's pulse simply indicates the opening and closing rhythm of the valve in the heart which causes blood to flow from one position to another.

To monitor bpm, it can use two forms of ECG signal and optical measurement. For the ECG method, it uses two electrodes placed on the chest to record electrical activity on the heart. In the optical measurement, it measures variation of light scattered or absorbed through blood flow. Based on Figure 2.3, it consists of the light emitting diode (LED) and a photodiode as a detector used for measurement on a person's fingertip. The LED transmits light to the skin's vascular tissue and light intensity variability occurs due to the sensor measuring blood volume [14].

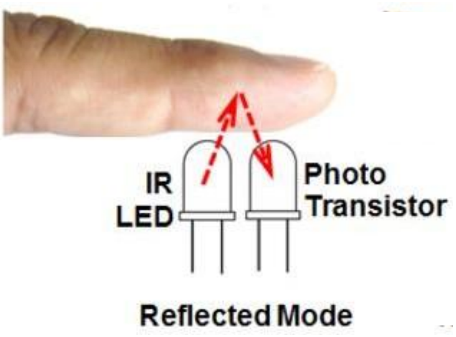


Figure 2.3 Heartbeat variation of light measurement

2.4 Microcontroller

The microcontroller is a standalone computer IC capable of programming and interacting with hardware devices with a set of instructions. It is primarily used for receiving input from sensors, storing and processing received information, and sending processed data to output for additional control of operation. There is a central processing unit (CPU) inside the microcontroller that is responsible for all input and output operations [15].

Microcontroller development began in 1970 and 1971 when Texas Instruments ' Gary Boone developed a TMS-1802-NC microcontroller chip with all the necessary circuits and capable of performing a specific task. In addition, Intel also launched its significant 8048 and 8051 microcontroller. This chip was used by IBM in PC and was still considered by usability to be the most long-lived microcontrollers. Figure 2.4 shows the evolution of microcontroller from 1970 until 2010 with uses of the transistor on the IC increasing due to uses of higher storage and processing data.

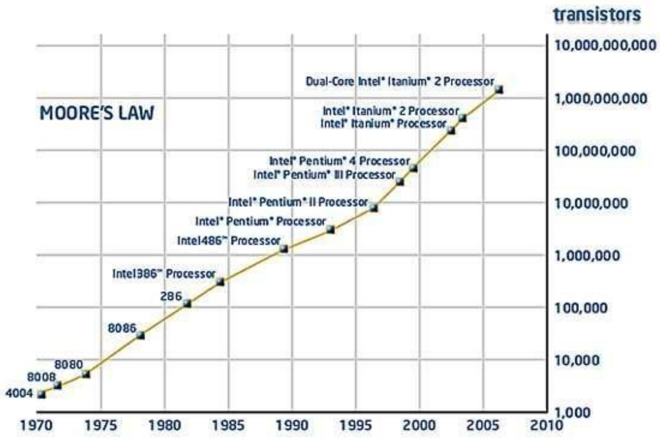


Figure 2.4 Evolution of microcontroller

2.4.1 Type of microcontroller

The manufacturers is developing many types of microcontrollers to cater for specific function. Therefore, most of these four microcontrollers have been used commonly and will be covered.

1. 8051 microcontrollers

It is an Intel-developed 8-bit microcontroller and one of the world's most popular microcontrollers used. The chip pin can be operated from Figure 2.5, for the cycle of input-output, serial communication, timer input and control indicators.

Figure 2.5 Intel 8051 microcontroller

2. PIC Microcontroller

To develop the programming language, programming in this microcontroller is fast and easy. The architecture consists of two programming and information storage memories. Some of the great features of the PIC microcontroller are timers, analog to digital converters and modulation of the output pulse width mode. Figure 2.6 show PIC microcontroller with better memory management capability to perform different tasks.

Figure 2.6 **Microchip PIC microcontroller**

3. AVR microcontroller

Atmel Corporation has developed the microcontroller since 1996. It has special features including 6-sleep mode, ADC and DAC, serial communication. For each microcontroller, it uses C language to program with predefined IDE. The AVR available in three tinyAVR, MegaAVR and XmegaAVR categories as shown in Figure 2.7. The larger size of the controller with more integrated peripherals and circuits can handle larger memory sizes.

Figure 2.7 Three Categories of AVR microcontroller

4. ARM Processor

A 32-bit or 64-bit ARM processor is used to execute machine instruction at a faster speed. It is primarily used in electronic devices such as smartphones, consoles and tablets due to its advanced system. From Figure 2.8, the ARM processors are designed with less size, less power and better speed performance.

Figure 2.8 ARM processor

It can be inferred that the object of each microcontroller is to perform a given task. From Table 2.2 based on its speed, memory size, architecture and cost, each microcontroller is compared. Such microcontrollers are now much smaller and cheaper to use with more efficient applications such as lighting, robotics, communication and consumer applications.

Table 2.2 Comparison of each type of microcontroller

	8051	PIC	AVR	ARM
Bus width	8-bit	8/16/32-bit	8/32-bit	32-bit.64-bit
Speed	12 clock cycle	4 clock cycle	1 clock cycle	1 clock cycle
Memory	ROM, SRAM, FL ASH	SRAM, FLASH	Flash, SRAM, EEPROM	Flash, SDRAM, EEPROM
Memory Architecture	Von Neumann architecture	Harvard architecture	Modified	Modified Harvard architecture
Families	8051 variants	PIC16, 17, 18, 24, 32	Tiny, Atmega, Xmega,	ARMv4,5,6, 7 and series
Cost	Very Low	Average	Average	Low
Popular Microcontro llers	AT89C51, P89v51, etc.	PIC18fXX8 PIC16f88X, PIC32MXX	Atmega8, 16, 32, Arduino.	LPC2148, ARM Cortex-M0 to M7,

2.5 Wireless communication system

Wireless communication is the exchange of data or energy between two or more points that an electrical conductor does not link. With the rapid development of wireless systems, most telephone and television communications have vanished using wired network. This situation triggers ignoring wired infrastructure and replacing it with a wireless system. An antenna with transmitter and receiver end will be required to transmit data through the air. These antennas act in the form of electromagnetic waves that propagate through space to convert electrical signals to radio signals. Thanks to a broad range of frequencies, radio waves are mostly used for short or long communication. Wireless communication uses include mobile phones, information transmission, peripheral devices, wireless energy transfer and medical technology. Wireless communication systems can be categorized into various types based on distance range, data form and type devices that are radio frequency communication, Wi-Fi communication, and mobile communication [16].

Wireless communication is a transfer of information over the air from one location to another without the use of any physical medium. An antenna with transmitter and receiver end is needed to transmit data through the air. Such antennas operate in the form of electromagnetic waves that travel through space to transform electrical signals to radio signals. Because of a wide range of frequencies, radio waves are mostly used for short or long communication. Wireless communication uses include mobile phones, data communications, linking peripheral devices, wireless energy transfer, and medical technology. Wireless communication uses include mobile phones, data communications, linking peripheral devices, wireless energy transfer, and medical technology.

There are many benefits of cost-effective, mobile wireless communication, ease of installation and reliability. The cost of installing wires can be reduced by using wireless communication, thereby reducing the cost of a communication system. In fact, it is quite easy to set up a wireless network. With the aid of code and manual, the setup time using the wireless system is also much quicker. Ultimately, there is a secure wireless network as there are no physical cables that may malfunction due to environmental factors. [17]

Free-space path loss (FSPL) in telecommunications is the reduction of signal intensity of an electromagnetic wave that would result from a line-of-sight path through free space, with no surrounding barriers to cause reflection or diffraction.

Path loss is also the decrease in power density which occurs when a radio wave propagates over a distance. The primary factor in the loss of direction is the reduction of signal intensity over the reach of the radio waves themselves. In free space, the sole factor affecting range is the inverse square law. In the real world, however, other factors will weaken the selection to [18]

• Obstacles such as walls, trees, and hills can cause significant signal loss.

• Water in the air (humidity) can absorb RF energy.

• Metal objects can reflect radio waves, creating new versions of the signal. These multiple waves reach the receiver at different times and destructively interfere with themselves. This is called a multipath.

2.6 Internet of Things (IoT)

The Internet of Things (IoT) is a network of interrelated computer equipment, mechanical and electronic devices, artifacts, animals or individuals with unique identifiers and the freedom to transfer data across a network without human-to-human or computer-to-human interference. In 1999, Kevin Ashton, co-founder of the Auto-ID Center, introduced the term IoT for the first time. He wants to bring radio frequency Identification to the attention of an organization with the name of the presentation as the Internet of Things which enables computers to handle individual things. Each machine's IoT type interacts with other objects without human interaction through a network. It then evolves to the next stage that is embedded in a sensor network linking the data exchange and collection process.

The IoT ecosystem consists of web-enabled smart devices that capture, send and act on data they obtain from their environments using embedded processors, sensors and communication hardware. Currently, IoT application getting bigger with more than 8.4 billion IoT devices were used in 2017[21]. Furthermore, application that mostly uses by consumer that contribute to IoT were smart Television and digital set-top boxes. From Figure 2.9 it demonstrates how the IoT system works. The system is a complete networking solution that integrates controllers, sensors and networking devices capable of receiving, transmitting and processing data via the IoT platform. The IoT devices collect data using sensor and antenna uses that share data with an IoT gateway. Using networking tools, these data will be sent to the cloud for further review.

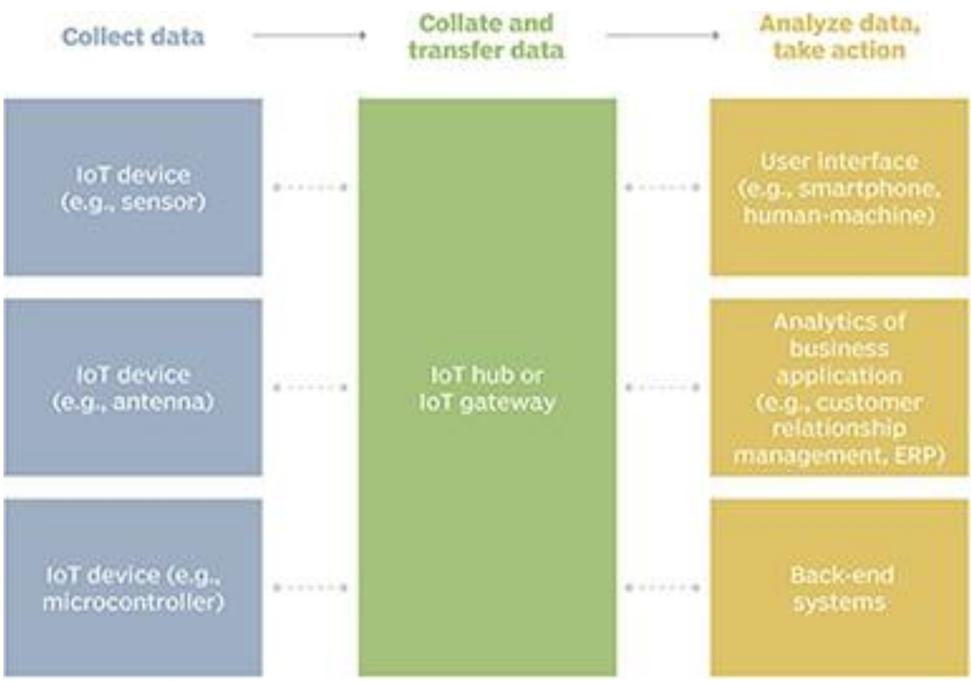


Figure 2.9 IoT process

2.6.1 Architecture of an IoT system

The system consists of 4 stages that are integrated and deliver actionable business insights. From Figure 2.10, phase 1 consists primarily of analog signal reading sensors and actuators. In phase 2, the data of the sensor are aggregated and transferred through the ADC phase to digital form. Phase 3 includes edge IT, which processes the data before sending it to the cloud. Lastly, in stage 4 the data is being store managed and analyzed through the cloud system. Eventually, the data is stored and analyzed via the cloud system in phase 4. The first two stages of technology are Process Engineering, while the last two phases are Information Technology.

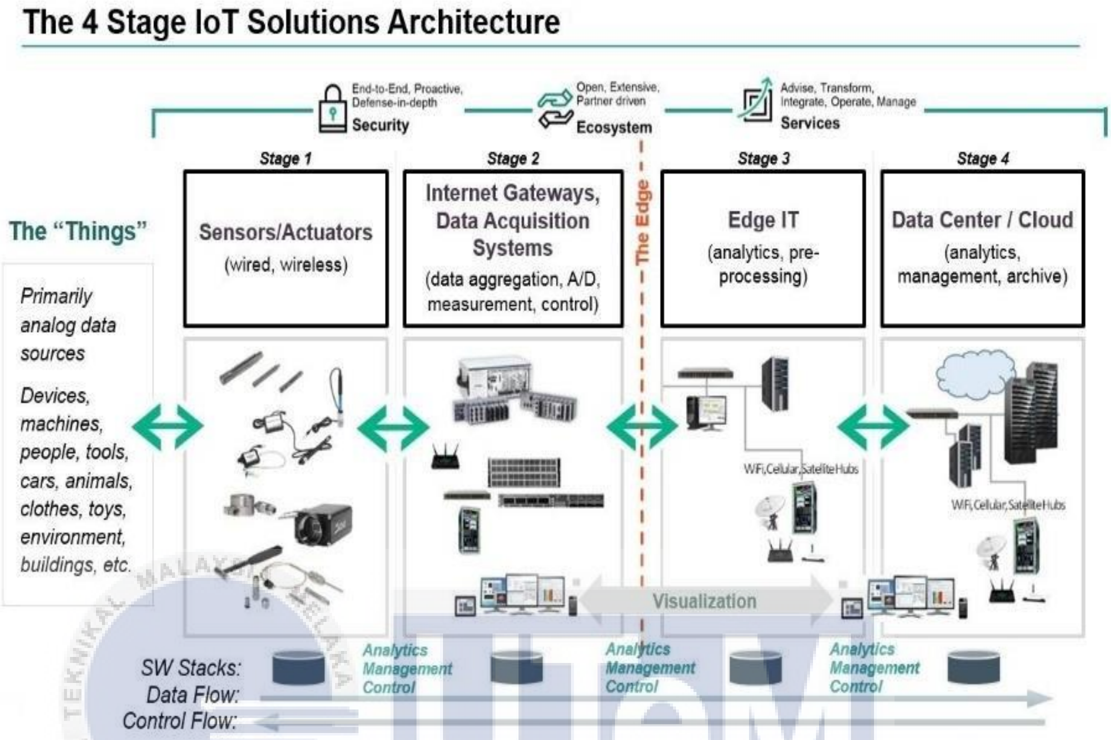


Figure 2.10 IoT architecture

CHAPTER 3

METHODOLOGY

3.1 Overview

This section describes the approaches used to complete this task and achieve the goal of the project. This part shows the project's flow from the start to the end of the project. The plan will be divided into two parts, the design of hardware and software. The first component consists of attaching sensors with integrated Wi-Fi development in NodeMCU controller board.

For the software development part, it comprises the uses of Fritzing software for circuit design and simulation, and Arduino IDE for programming language. Lastly, in project development, it will cover database development using PHP and MySQL language. Next, application-based is created using Android Studio for monitoring system and IFTTT platform for notification process.

3.2 Project planning

There are 3 major parts that will be used to complete the preparation, execution and evaluation of this project. Project planning is carried out at the project's earliest stage. Project planning becomes the most vital part of this project in order to select the hardware and software that meet the requirement. First of all, the planning includes making a work schedule that needs to be done within a timeframe using the Gantt chart to list tasks. Next, do some research from previous work from different sources to obtain more information about this project. Then, finding proper hardware and software to create a complete circuit that meets the requirements of the design. The design will then be developed on the basis of the results and checked using code designed using specific software. In addition, the completed circuit is calculated and documented using an appropriate method to achieve the outcome of the design. Lastly. After the work and completion of the project circuit, on the last section, a full evaluation will be done to test its performance of system and wireless transmission strength.

3.2.1 Research methodology flowchart

To complete the project, there are 3 part which are software development, hardware development which is design plan as shown in Figure 3.1. The flowchart below is a step by step overview of the project from start to finish. It can only be done for the development of a project after completing the first two-part. It consists of three parts consisting of software development, hardware development and project development.

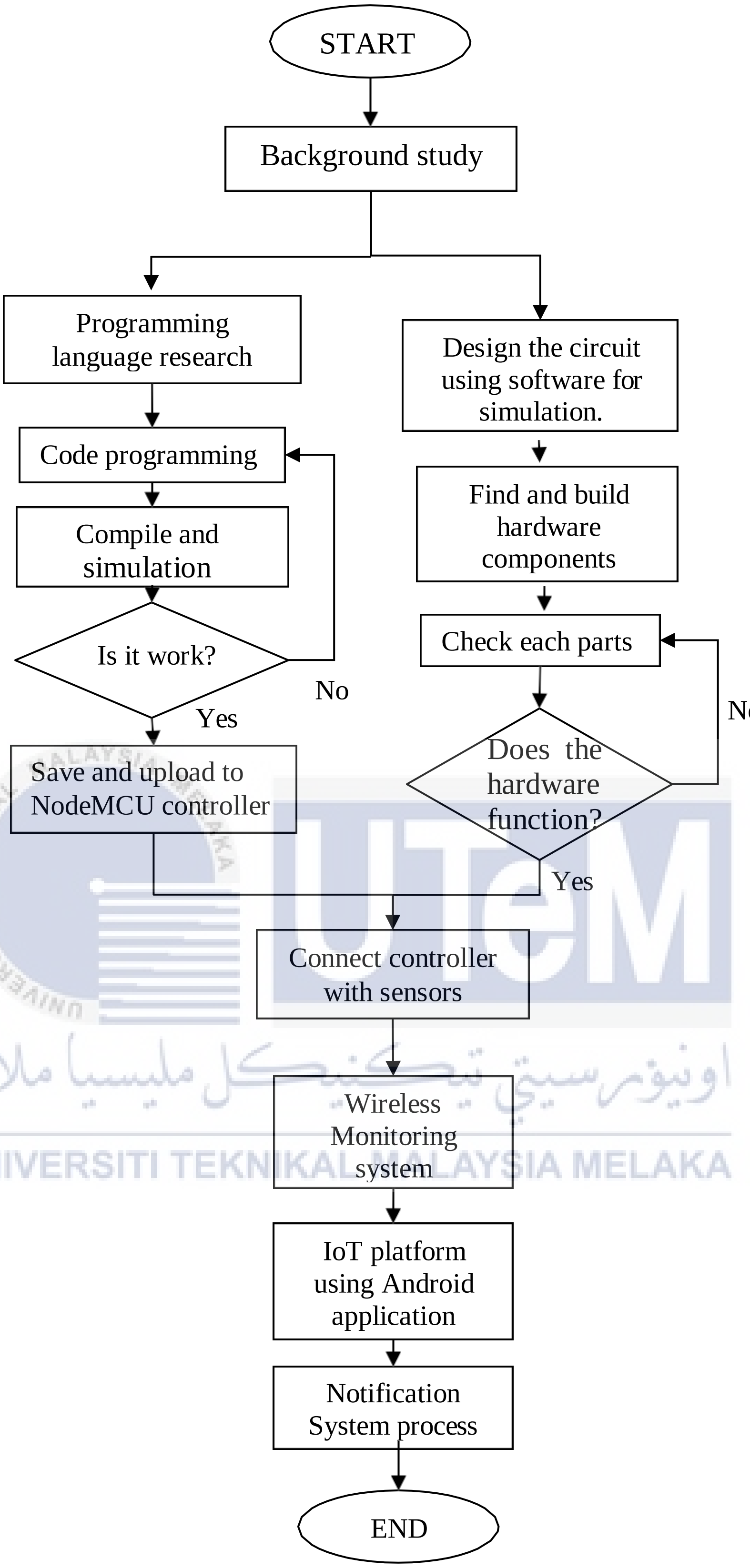


Figure 3.1 Flowchart of project methodology

A detailed explanation for the project methodology are as following:

1. Software development

For this part, I will carry out some research from previous work and find the code used to communicate with the controller board. Next, I will write the code program using preferred software and verify the coding to avoid errors. Then, I will upload the proper code program to be uploaded to the NodeMCU board.

2. Hardware development

.For this part, I will design the circuit and run some simulation for the simulation program to prevent any hardware connection errors. Then, I will find each components for the hardware that will be used in the project after the simulation results. The controller board will be configured with two health parameter sensors with Wi-Fi access point. Next, checking each hardware circuit and transmit data between controller board and sensors. I will configure open source database platform to display the data collected from sensors through IoT implementations with internet connection.

3. Project development

For project development, I will do some test on the controller board to make sure it can transmit data through the database platform with Wi-Fi connection. Then, for IoT-based monitoring system, I will create an android application to monitor with smartphone. Next, notification system will be implemented using email. Then, I will do the analysis on the performance of system whether the results stored is accurate and can be compared with actual health devices. Finally, another analysis for wireless transmission strength by distance will be conducted.

3.2.2 Project schedule

To complete this project, it will take a schedule to indicate the time span for every task that needs to be done. Project scheduling can be simpler and quicker with the use of a Gantt chart, which can show what work to do on a specific day**.** For Figure 3.2 Show project preparation where project software and hardware development is to be completed by the first semester. For semester two, I will build a hardware prototype with integrated IoT as the project's final task. After that, I will start evaluating the output performance of the project and obtain results.

	SEM I																			**SEM II**															
Aktiviti Projek *Project Activities*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**	**1**	**1**	**1**	**14**	**15**	**16**
PSM project title and proposal preparation		X	X	X																															
Proposal submission				X																															
Past studies research				X	X	X																													
Programming language research					X	X	X	X	X																										
Hardware construction								X	X	X		X	X	X																			**SEMINAR PSM II**		
Software development:														X	X	X				X	X	X	X												
Hardware development																							X	X	X	X	X								
Project prototype																											X	X	X	X	X	X			
PSM report submission																																		X	

Figure 3.2 Project Gantt chart

3.2.3 Data collection

This is part of the phase which focuses on the collection of previous studies. At this point I planned to collect project assets and information, context analyses, and hardware and software specifications. Such tools are gathered across a variety of media consisting of internet blogs, journals, academic papers and encyclopedia.

During data collection, there are many ways to create a healthcare monitoring system by using different type of sensors, controller board and wireless transmission system. For example, there are particular type of temperature sensor used has its own sensitivity and precision that can influence the measurement of body temperature. Other than that, the uses of LoRa, Wi-Fi module, ESP8266 module, and Arduino Wi-Fi Shield which used for sending data wirelessly in a reliability of data collected.

In the method, I review each part feature relevant to the projects, including the comprehensive specification and its data sheets. In addition, the processing of data is included in the context and limitation of the project.

3.3 Project Implementation

This section will cover the requirements for hardware and software and how to use it on the project.

3.3.1 Hardware development

This section will cover hardware and software specifications and ways to apply them to the project. The design part of the project will explain in detail about the operation of the circuit, the project block diagram and its flow chart. Lastly, in the development of projects, the monitoring and notification system used in this project will be covered are:

1) Temperature sensor LM35

LM35 is an analog linear temperature sensor. Its output is proportional to the temperature in degree Celsius. The operating temperature range is from -55°C to 150°C. It also have low self-heating at 0.08°C and output voltage varies by 10mV in response to every rise or fall in temperature of 1 ° C. It can be operated from a 5V as well as 3.3 V supply and the stand by current is less than 60uA. Table 3.1 below shows specification of the sensor that can obtain accuracy up to ±0.5℃ which are suitable for body temperature measurement.

Table 3.1 Specification of LM35

Parameter	**Value**
Supply voltage	4V to 30V
Accuracy	±0.5℃
Temperature range	-55℃ to 150℃
Sensitivity	10mV/℃
Output maximum current	10mA

The sensor consists of p-n junction diode that employs CMOS technology that has its output voltage vary with resistance in the LM35 IC as shown in Figure 3.3 can provide changes to every degree rise. Using equation 3.1, the temperature measure can be obtained.

$$Temperature = {}^{Vout}/_{10mV/℃} \qquad (3.1)$$

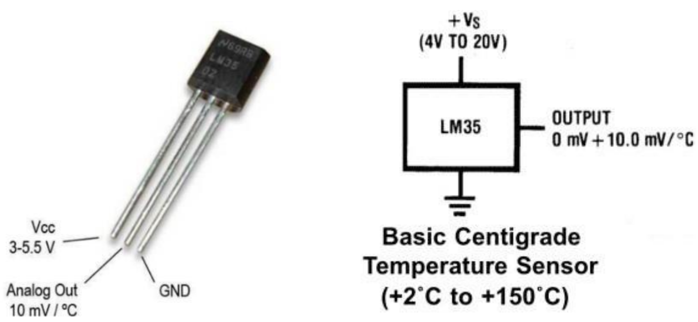


Figure 3.3 LM35 pinout

2) Pulse Rate Sensor

This sensor can easily interface with any controller using analog input pin. It also includes an open-source monitoring app that graphs your pulse in real time. The sensor front is the Heart shape logo covered. This is the side of the skin making contact.

For heartbeat measurement, the sensor takes reading through the tip of the finger based on blood flow rate. The sensor circuit designed using principle of photoplethysmography (PPG) that simply responds to the relative changes of light intensity that reflected during each pulse. The LED lights up the fingertip or earlobe or other capillary tissue, and the detector reads the amount of light bouncing back. This is how the heart rate is calculated. If there is no light reflected, the output remains constant while if the photodiode detects more amount of light reflected it will provide output voltage. From Figure 3.4, the pulse sensor shown consists of PPG detector circuit, amplification circuit, and noise cancellation circuitry.

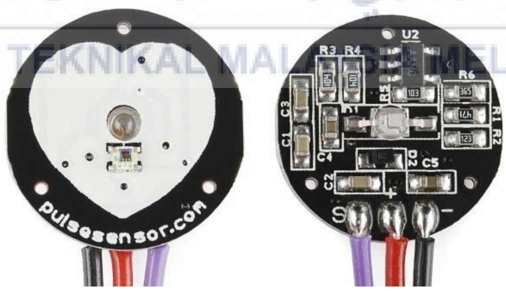


Figure 3.4 pulse sensor circuitry and PPG detector

3) NodeMCU ESP-12E WiFi ESP8266

NodeMCU is an open source development board and firmware based on the widely used WiFi module ESP8266-12E. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The term "NodeMCU" applies specifically to the firmware rather than the related development kits. It allows to program the ESP8266 WiFi module with the simple and powerful LUA programming language or Arduino IDE. The firmware is based on the eLua project and is based on the Espressif Non-OS SDK for ESP8266. It uses a number of open source projects, such as Lua-cjson and SPIFFS. The design was initially based on the ESP-12 module of the ESP8266, a Wi-Fi SoC integrated with the Tensilica Xtensa LX106 core, and makes it widely used in IoT applications.

With its USB-TTL, the NodeMCU Dev board help flashing directly from the USB port as shown in Figure 3.5. Combines the functionality of the WIFI access point and the station and microcontroller. These features make the NodeMCU extremely powerful Wifi networking tool. It can be used as an access point and/or station, hosting a web server, or connecting to the internet to collect or upload data.

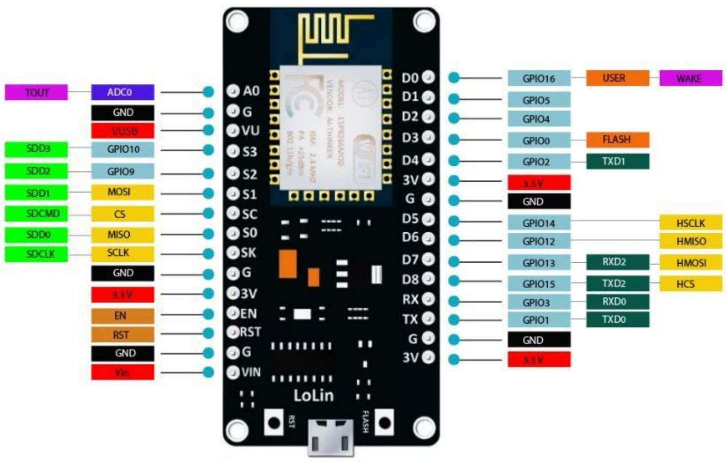


Figure 3.5 NodeMCU

4) CD-4051 IC Multiplexer with 8 Channel

CD 4051 is essentially an analog multiplexer and a demultiplexer. It has very low leakage current OFF and low ON impedance. The CD-4051 dissipates very low power over full voltage ranges. The CD-4051 is an 8 channel multiplexer and has three control inputs named A, B and C. These inputs bind only 1 out of 8 channels to the output to obtain the desired output. Channel I / O terminals have become outputs and common O / I input terminals when CD 4051 is used as a demultiplexer as Figure 3.6 shown.

CD-4051 has many impressive features, binary address encoding on a chip, split before switching removes overlapping channels, wide range of analog signals as well as digital signal rates. CD-4051 can be used for signal gating, analog to digital conversion, digital to analog conversion, analog and digital multiplexing and demultilplexing. Since NodeMCU has only one analog input, the multiplexer is used in order to connect multiple analog sensors which are enough for both analog sensors LM35 temperature sensor and pulse rate sensor.

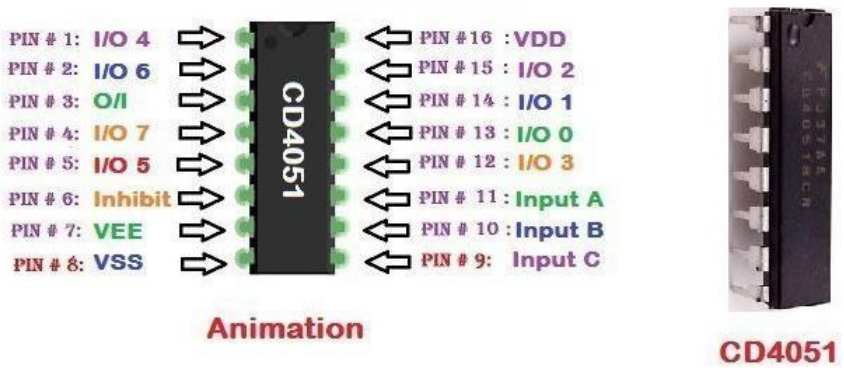


Figure 3.6 CD-4051 IC Multiplexer

3.3.2 Software development

In this section, Fritzing software is used to design any circuit diagram that preferred and run simulation to check pin connection. The Fritzing software is a free tool that allows to design any electronic hardware. It support designers ready to move from experimenting with a prototype to build a more permanent circuit. In this software, a complete circuit was created which consist of sensors and each module before moving on to real hardware connection. From Figure 3.7, circuit design with pin connection to the board was created. To avoid any error, design rule checking was performed through simulation process.

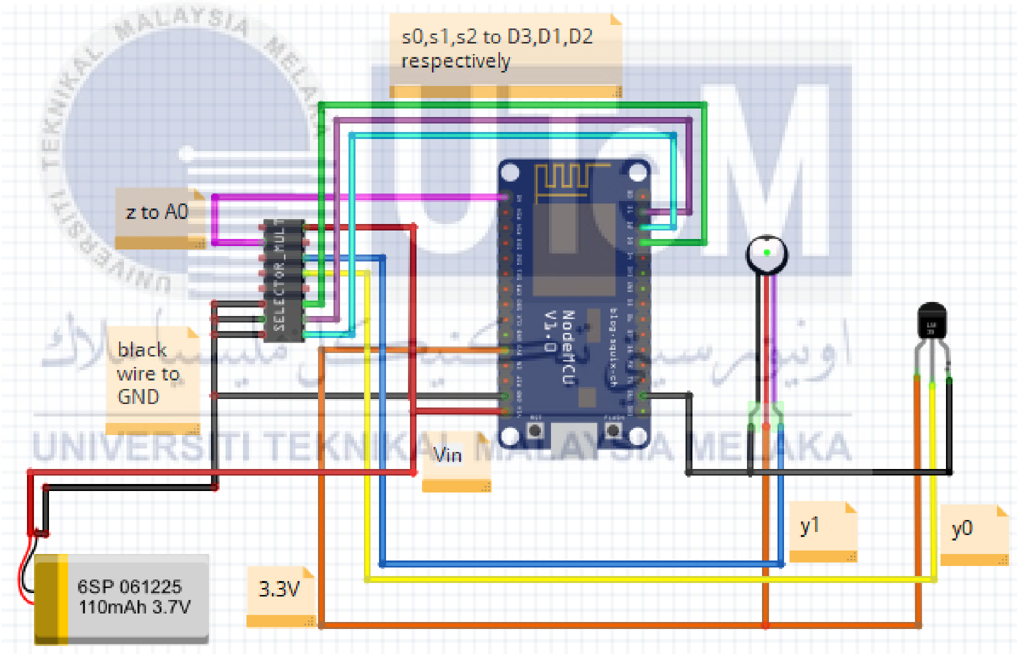


Figure 3.7 Circuit Design in Fritzing software

Next, Arduino Integrated Development Environment (IDE) is used to write code program and upload into the controller board. Arduino IDE uses serial transmission on the USB port to upload completed program sketch into the controller board.

Besides, the software also has extra features call serial monitor as shown in Figure 3.8. It allows displaying data from NodeMCU board into the serial monitor using Serial.PrintIn command. The IDE software is used throughout the project to write code and compile the code to display data result from sensors when connected to controller board. I also used this software to write code for the project and compile the code to check program code error and upload it into controller once it is done.

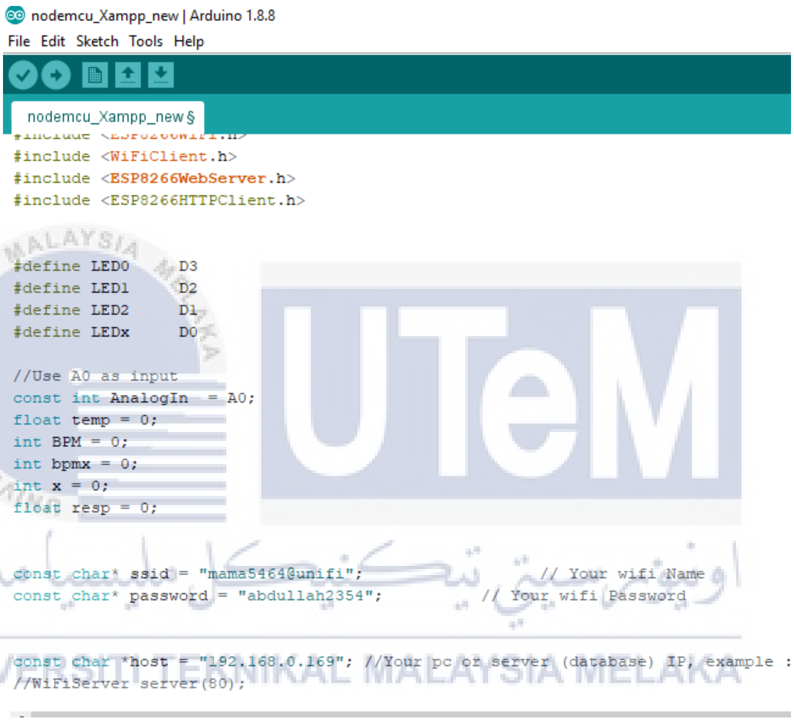
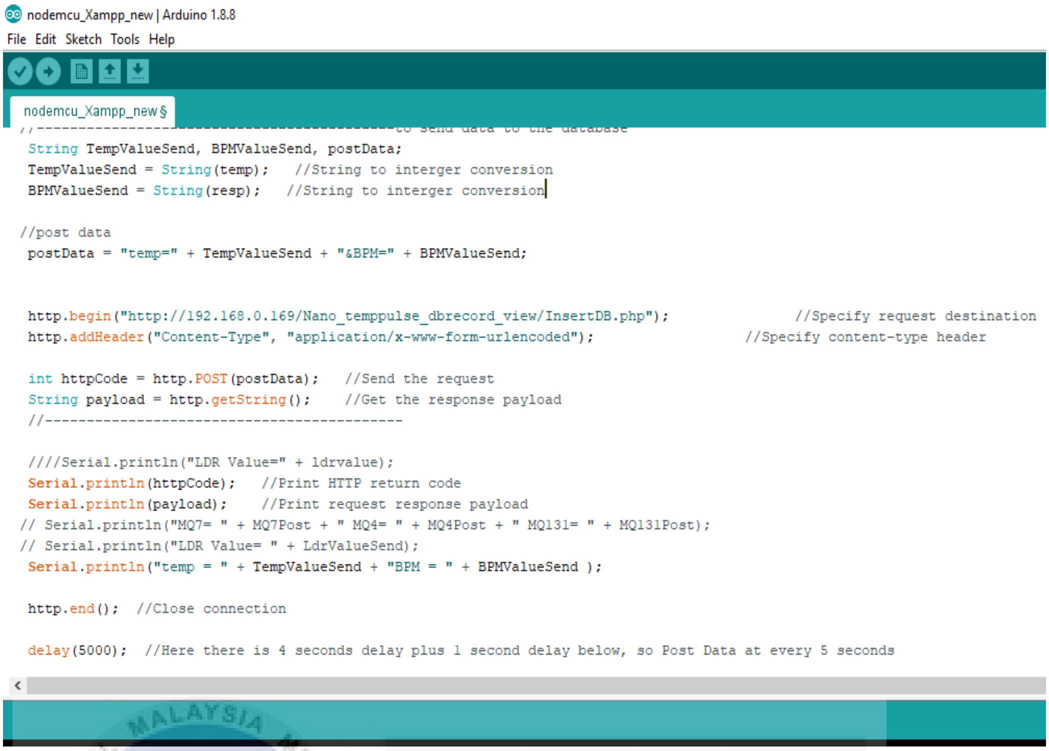


Figure 3.8 Arduino IDE sketch

The coding on NodeMCU uses Wi-Fi access point to send sensor's data wirelessly. Other than that, Analog to digital conversion occurs for temperature and pulse sensor which convert its voltage output to 10 bit of binary digit that will be processed by microcontroller through IC multiplexer. Next, the data obtained is sent to database platform by using POST method and HTTP request before actual IoT platform is implemented. In order to perform this request, a complete HTTP code needs to be applied in Arduino IDE sketch as shown in Figure 3.9.




```
//-----------------------------------------------------to send data to the database
 String TempValueSend, BPMValueSend, postData;
 TempValueSend = String(temp);   //String to interger conversion
 BPMValueSend = String(resp);   //String to interger conversion

//post data
 postData = "temp=" + TempValueSend + "&BPM=" + BPMValueSend;


 http.begin("http://192.168.0.169/Nano_temppulse_dbrecord_view/InsertDB.php");              //Specify request destination
 http.addHeader("Content-Type", "application/x-www-form-urlencoded");              //Specify content-type header

 int httpCode = http.POST(postData);   //Send the request
 String payload = http.getString();    //Get the response payload
 //-------------------------------------------

 ////Serial.println("LDR Value=" + ldrvalue);
 Serial.println(httpCode);   //Print HTTP return code
 Serial.println(payload);    //Print request response payload
// Serial.println("MQ7= " + MQ7Post + " MQ4= " + MQ4Post + " MQ131= " + MQ131Post);
// Serial.println("LDR Value= " + LdrValueSend);
 Serial.println("temp = " + TempValueSend + "BPM = " + BPMValueSend );

 http.end();  //Close connection

 delay(5000);  //Here there is 4 seconds delay plus 1 second delay below, so Post Data at every 5 seconds
```

Figure 3.9 Coding function to send HTTP request

3.3.3 Project design

After finding all the hardware requirements and create a program code, project build hardware is conducted. In order to take the next step, each component module was purchased through an online shop and electronic shop. The faculty of university also provides components needed for this final year project. For the first stage of circuit development, jumper wires and protoboard is used to perform connection between each hardware module as shown in Figure 3.10.

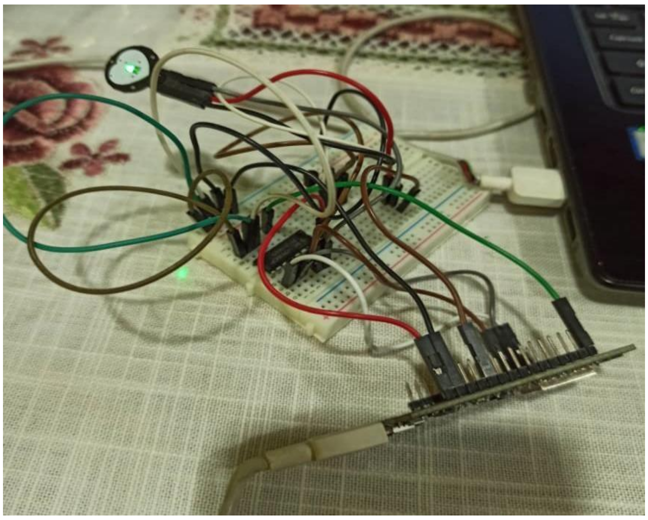


Figure 3.10 Project with hardware development

Figure 3.11 shows the flow of diagram consists of input, data processing, and output part. First, the temperature sensor and pulse sensor are mounted on each tip of the finger from any patient to detect health parameters which are heartbeat rate and body temperature. These data were sent to the NodeMCU controller through multiplexer for ADC and DAC process. The received data were processed and send to database platform using ESP8266Wi-Fi module in NodeMCU and configure with database platform using MySQL. This application allows wearable monitoring system to be achieved where the NodeMCU controller, sensors and power source are mounted on wrist of the hand. Figure 3.14 shows the diagram of the proposed system project.

Figure 3.11 Diagram of the proposed project

Figure 3.12 also shows a flowchart of the project for monitoring and alert notification. From the flowchart, it shows that when the temperature or heartbeat not within a normal range, the system will send push notification to the doctor or family member. If the parameters within an acceptable range, the data will be displayed through database using Wi-Fi for monitoring system by doctors or any authorized person.

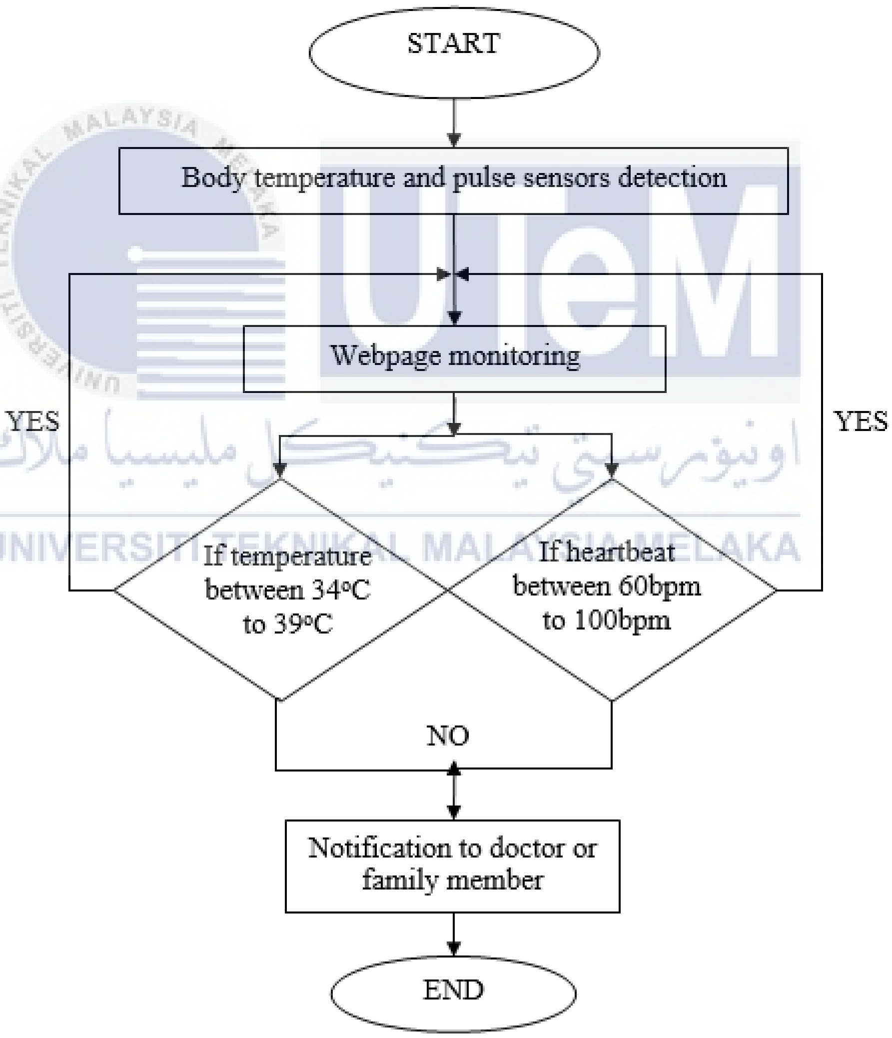


Figure 3.12 Flowchart of the project for monitoring and alert process

3.3.4 Project Development

This is the last stage of completing the project. It consists of design each hardware module connection in a PCB blank board using soldering tools. Next, create a database platform using PHP and MySQL programming language. Lastly, build an application for Android smartphone using Android Studio that covers Java and Kotlin script for IoT implementation. Finally, a push notification will be made using IFTTT to trigger Gmail to send message when results is not within the range.

3.3.4.1 Soldering Circuit Board Design

First, the hardware is constructed on breadboard in order to test each component and get wanted results. In this experiment, jumper wires and temporary power source are used to test the system's functionality. Once each of the test completed with zero error, the circuit is designed and soldered on blank printed circuit board (PCB). Each hole of blank PCB breadboard is assign in any path and soldered properly based on circuit diagram using soldering tools provided as shown in Figure 3.13. After completing the soldering part, each connection was tested using Multimeter to avoid short circuit by applying continuity test.

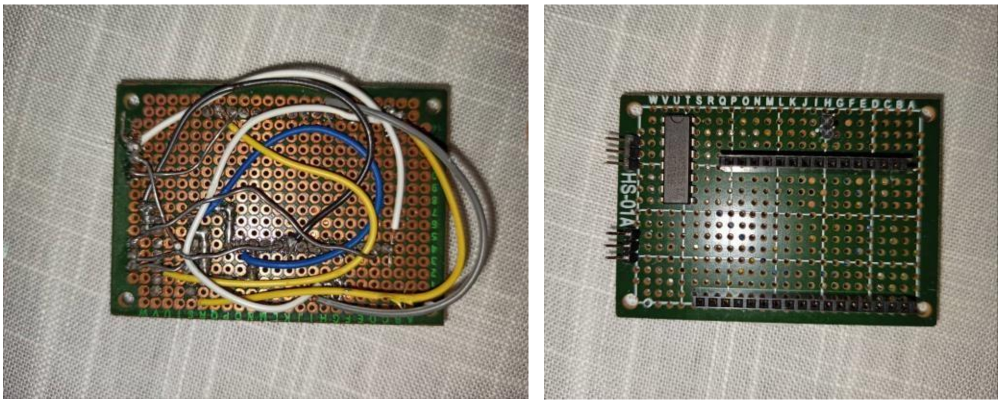


Figure 3.13 Soldering process back and front

3.3.4.2 Database Platform

In order to achieve real-time monitoring system in this project, a database platform is created using PHP language. The database is created through phpMyAdmin to display temperature and pulse value in table form that shows time and date of each received data. From Figure 3.14, the table consists of five columns which are created using MySQL. Besides, the data display in table were fetched from the database using MySQL command. Each time the database receives POST request from NodeMCU controller, it will store the data from both sensors into the database system.

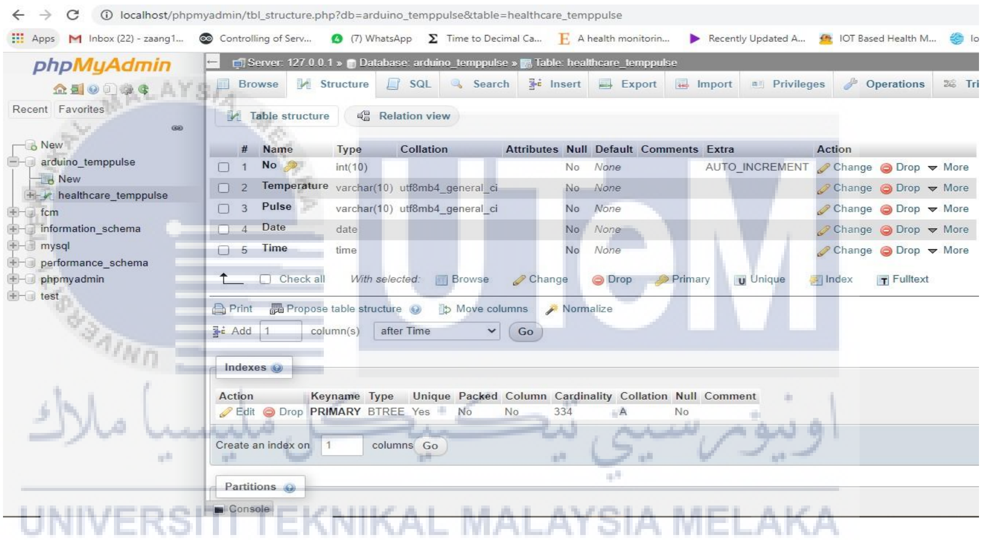


Figure 3.14 Database Platform for Monitoring System

Other than that, the database has “InsertDB.php” file which used to obtain data send from esp8266 Wi-Fi inside NodeMCU board using POST method and store into database.

Once each of PHP file finish created, the database platform is hosted online through free hosting website with database server. The project uses 000webhost as the website server to keep the monitoring system online in real time. From Figure 3.15 show list of PHP file stored into the server file manager used in processing the database. From this process, the monitoring system use “https://healthdroid.000webhostapp.com” as the main URL to monitor the data.

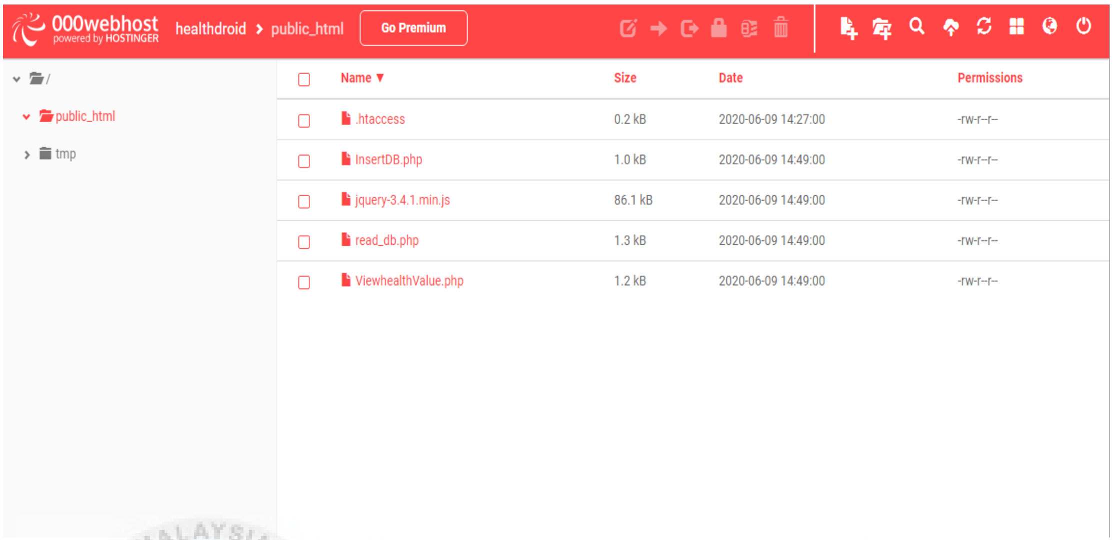


Figure 3.15 Online web hosting file manager

3.3.4.3 Android application design

The IoT platform is a part of the project to be developed by creating an Android application and displays database instantly from the website that has been created through any smartphone.

For integration of a website provided in the application, the WebView tool is used in Android Studio. WebView is an extension of the android layout view that allows any webpage to be displayed as an activity in the application. The layout of the application consists of two main activity which are Main Activity and WebView Activity. From Figure 3.16, The Main Activity shall display information about the project. Then, the button will be used to navigate to the next website interface to view the data.

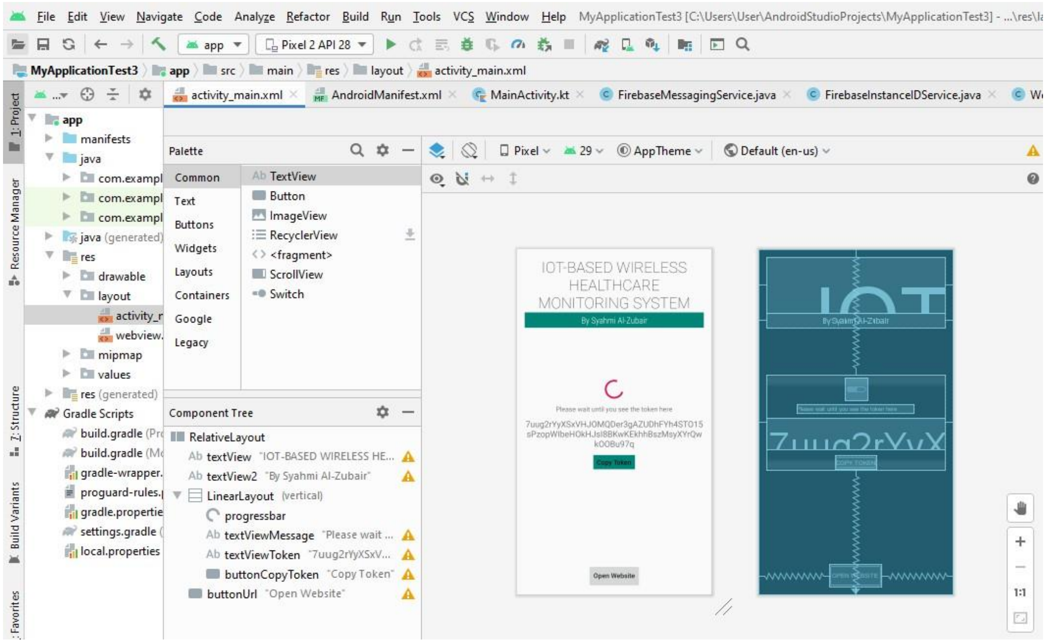


Figure 3.16 Main Activity layout of the application

Lastly, WebView is designed on layout using tools in Android Studio. From Figure 3.17, the WebView loads the webpage using the 'loadUrl' command and displays the contents inside the layout.

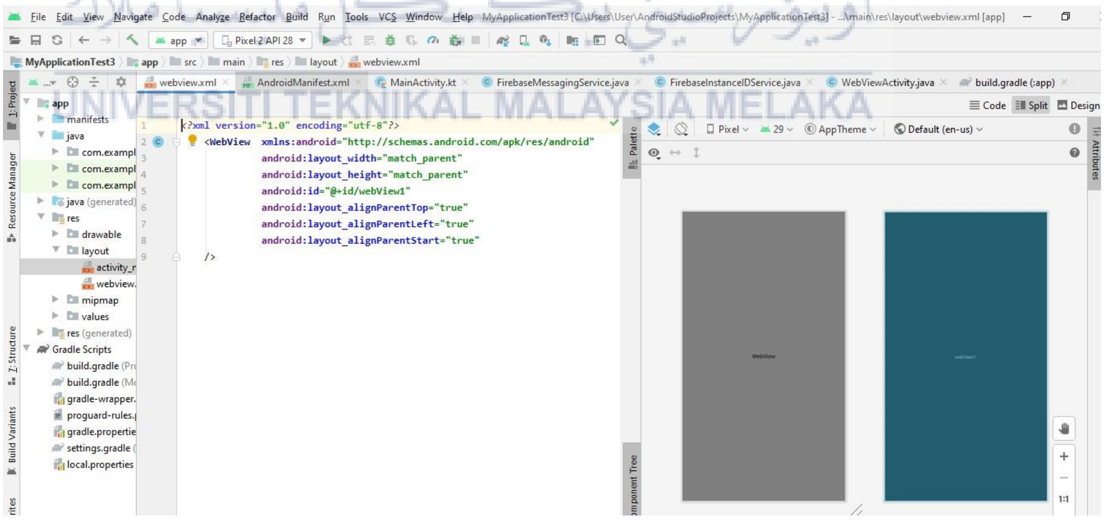


Figure 3.17 WebView activity layout of the application

Once the Android application was designed using the Android Studio software, the USB debugging setting was installed on the smartphone. From Figure 3.18, the application consists of two layout that allow monitoring through database platform at a given time. Once ‘open website’ button is pressed, it will load up webpage created and display the data in database form.

Figure 3.18 Layout of Android application installed

Finally, a notification system which is also important for the monitoring system to alert any user if any abrupt changes in database system occur. A push notification is created using implementation of IFTTT. IFTTT is a web-based freeware service that creates simple conditional statement chains, called applets. An applet is triggered by changes occurring within other web services such as Gmail, Facebook, Telegram, Google Sheet and other applications. For this project, Gmail is integrated with IFTTT to send notification in form of email messages if else condition applied to the data. When the value for temperature or pulse is not within a normal range for a patient, it will send push notification to email through implementation of IFTTT as shown in Figure 3.19.

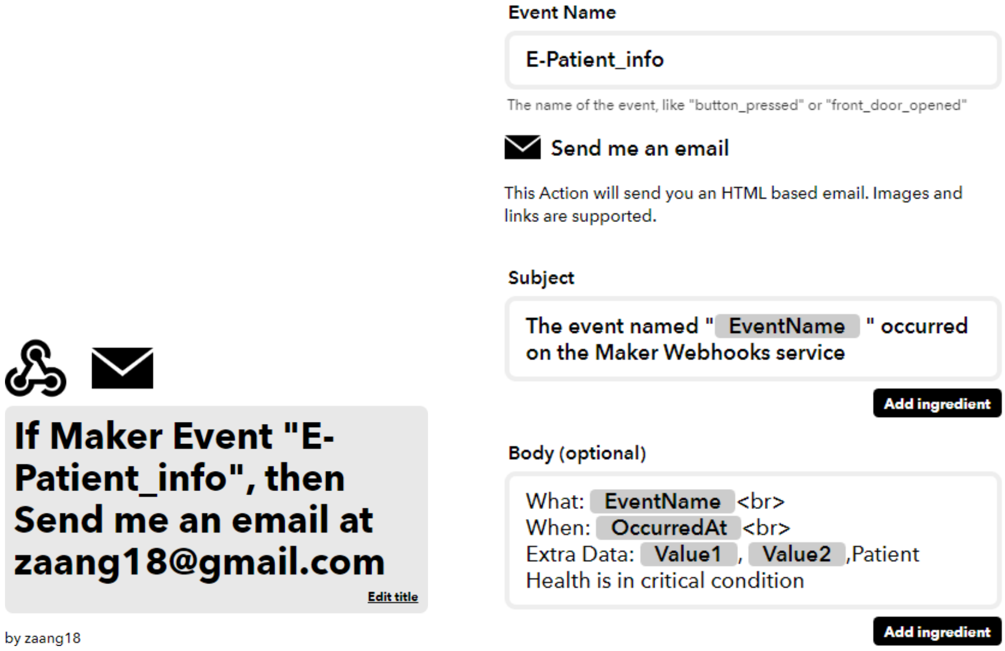


Figure 3.19 Notification system using integrated IFTTT's Gmail

3.4 Project Analysis

This is the final approach used to complete the project. This method is also intended to achieve the second objective. The first analysis covers analysis for project performance based on result accuracy from project sensors. To obtain accuracy of the sensors from system used, medical devices used by medical center will be needed to compare with measured values using sensors in this project. Then, samples of data values from both devices recorded simultaneously are calculated with percentage of error formula. Big amount of data samples obtained would be a great deal in order to achieve accurate analysis. Despite the reference of each sensor's datasheet, this analysis should be carried out to ensure the system project measured correctly whether the project can be used to test patient's health status remotely from hospitals. An actual pulse reading device which is Omron Automatic Blood Pressure Monitor in **Figure**

3.20 is used to compare with pulse rate sensor. The setup is important to ensure accurate data received when comparing pulse reading values from both devices as shown in **Figure 3.21** and **Figure 3.22**.

Figure 3.20 Omron Automatic Blood Pressure Monitor

Figure 3.21 Strap Pulse sensor on fingertip properly

Figure 3.22 Applying the arm cuff of blood pressure monitor properly

As for the temperature reading comparison, a Digital Thermometer is used as shown in **Figure 3.23**. For this case, the digital thermometer is measured by a person's mouth to achieve a reading while strapping LM35 sensor to one of fingertips as shown in **Figure 3.24** and **Figure 3.25**.

Figure 3.23 Digital Thermometer

Figure 3.24 Strap LM35 sensor with a fingertip properly

Figure 3.25 Use digital thermometer orally and under the tongue

Next, another analysis carried out is on wireless transmission signal strength measured by distance. The aim of this analysis is to find out how far the wearable system can travelled while transmitting data through Wi-Fi connection. For this, a measuring tape is used to measure distances on each data received from WIFI router from home. The distances measured can be used for free-space path loss calculation. The calculation can be proved to be theoretical for signal strength loss with different frequencies.

3.5 Project cost

Most components and hardware received from faculty and bought through online shop due to unavailability in normal shop as well as cause of pandemic season. The component received from faculty of university are NodeMCU ESP-12E WiFi ESP8266 and Lipo ion Battery 7.4V. Table 3.2 show cost of each hardware and component bought which was around RM67.74.

Table 3.2 Total project cost

Component	Quantities	Price	Total
Pin header	1	RM2	RM2
CD-4051 IC Multiplexer	5	RM1.10	RM5.50
LM35 temperature sensor	1	RM8	RM8
Jumping wires	3	RM7	RM21
Pulse sensor	1	RM11.24	RM11.24
Li-ion Battery 7.4V	1	RM20	RM20
			RM67.74

3.6 Expected result

From the experiment, the project had been completed from 3 stages mentioned. By using hardware and software created, the reading values can be displayed through the webpage monitoring system with notification system implemented and connected by Wi-Fi. Any doctor or family members are able to monitor the database platform of patient's health status through IoT-based application through smartphone. In order to wear the controller board mounted with sensors on hand, a casing is created to keep the project hardware intact as a prototype as shown in **Figure 3.26**.

Figure 3.26 Hardware Prototype

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Overview

This chapter will discuss results obtained from the constructed project. In this chapter, each of the data collected from the experimental and simulation process will be analyzed. Besides, the calculation used to obtain pulse sensor beat per minute will be proven by the application of general formula. In analysis section, the system performance is analyzed by comparing values of sensors with real medical devices through data samples and calculation. Next, another analysis is conducted for wireless transmission signal strength by measured distances.

4.2 Results Obtained

The webpage displays the data from the project device in every 50 milliseconds delay and update in the database platform connected to the server. The data obtained from sensors are displayed in table form with date and time on each data received. Figure 4.1 show result obtains from the webpage for pulse rate and temperature value. The parameter values are within an acceptable range. Due to the limited functionality of NodeMCU controller, the system can send readings from both sensors per time. The pulse sensor using an interrupt and timer function in order to get beat interval up to fifty seconds. Hence, this somewhat cause some inaccurate reading obtained from temperature sensor due to the delay in getting the measurement. To solve this issue, ensure the system collects pulse rate and temperature with correct connection as well as the correct placement of fingers for each sensor.

localhost/Nano_temppulse_dbrecord_view/ViewhealthValue.php

Healthcare Database

Temperature (°C)	Pulse Rate (BPM)	Date	Time
36.77	99.00	2020-06-05	14:55:22
35.48	100.00	2020-06-05	16:26:59
36.77	65.00	2020-06-05	16:27:47
34.19	93.00	2020-06-05	16:28:35
37.10	93.00	2020-06-05	16:30:22
34.84	97.00	2020-06-05	16:31:45
36.77	91.00	2020-06-05	16:34:32
36.13	85.00	2020-06-05	16:36:08
39.68	99.00	2020-06-05	16:37:19
37.42	96.00	2020-06-05	16:39:42
37.42	98.00	2020-06-05	16:40:23
36.13	90.00	2020-06-05	16:40:59
35.81	100.00	2020-06-05	16:41:11
37.74	99.00	2020-06-05	16:41:35
67.74	90.00	2020-06-05	16:42:22
36.13	96.00	2020-06-05	16:43:10
40.00	91.00	2020-06-05	16:43:34
40.00	99.00	2020-06-05	16:44:27
39.68	100.00	2020-06-05	16:44:51
38.71	99.00	2020-06-05	16:45:02

Figure 4.1 Results obtain for temperature and pulse rate

From the result obtained in Figure 4.1, the monitoring system for the project is achievable.

Next, for IoT platform, the project uses Android application as monitoring webpage through smartphone. The layout for webpage from Android application can be done using Android Studio software tool as shown in **Figure 4.2.** Once the button 'open website' is pressed, the created webpage will be loaded and the data will be displayed in database form. An IoT-based monitoring system has achieved.

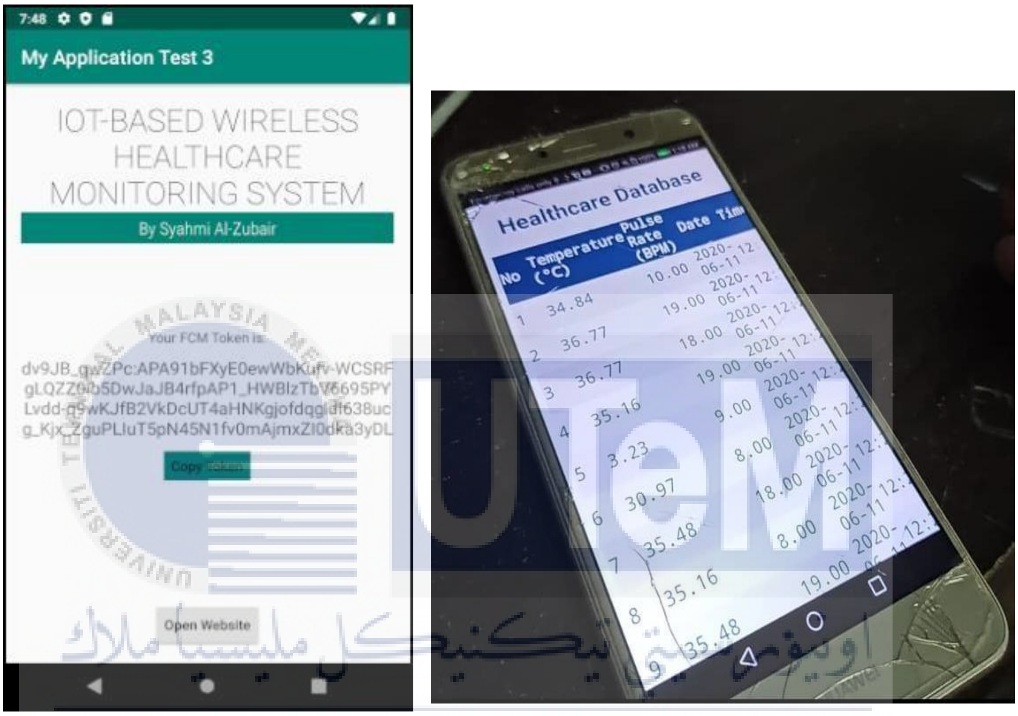


Figure 4.2 Layout of Android application installed

The project then demanded a push notification to be sent to the doctor's phone when the parameter of the patient's health is in the critical range. Each individual has different pulse rate and body temperature depending on size, age, sex and metabolism. So, in normal condition only doctors or doctors know acceptable range for a person's health parameter. However, in this experiment, the normal range for the temperature is set between 34 to 39 and pulse rate between 60 to 100 bpm. Therefore, where the health parameter is not within the threshold, the smartphone installed with the Android application will be notified via IoT implementation. From Figure 4.3, the project receives emails when the temperature and pulse rate of the patient reaches an abnormal reading.

Figure 4.3 Application of push notification through email from both sensors

4.3 Pulse and temperature sensor mechanism

The pulse sensor uses a light reflection and heart rate detection system based on **Figure 4.4.** the LED emits green light to the skin of fingertip, and the APDS-9008 light sensor tests the amount of light reflected for each heartbeat depending on the blood flow saturation. This input signal from the light sensor is transmitted via a low pass filter to clear undesirable high frequency signals. The signal is then amplified to get a better output signal using the MCP-6001 Op-amp circuit. Finally, the output voltage between 0.2V and 1.5V is transferred to the NodeMCU controller board through IC multiplexer using the ADC converter.

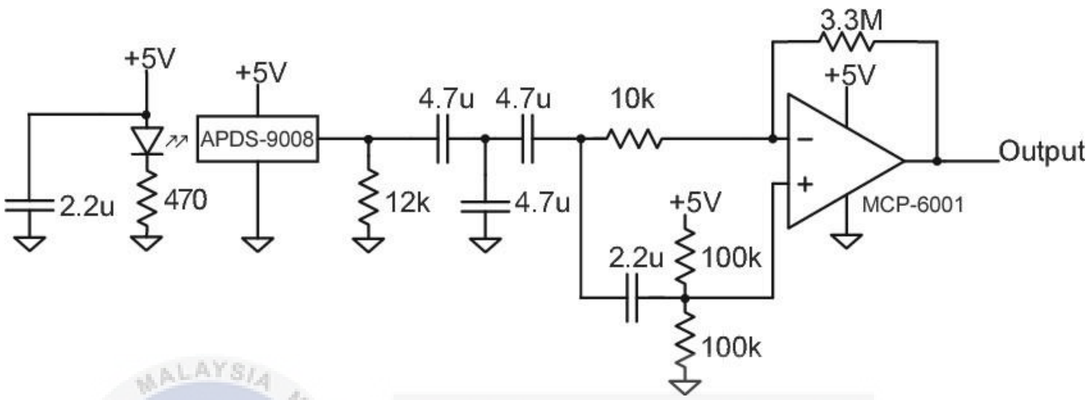


Figure 4.4 Circuit Diagram of pulse sensor

From **Figure 4.5**, the displayed output was in PPG signal form obtained from the serial plotter of Arduino IDE. Each time a heartbeat rate is calculated, it requires the interrupt function that measures the pulse signal interval (IBI). This approach permitted the use of a timer to track each signal pulse inside the IBI. A threshold value is set within the code program to 516 to get 50 percent of the amplitude of the pulse to obtain IBI values.

Figure 4.5 PPG signal from Arduino serial plotter

The computer program uses the library PulseSensorPlayground which already provides a set of pulse rate methods. From **Figure 4.6**, the IBI is obtained and stored in sequence of 10 elements. The equations 4.1 and 4.2 are used to get Beat per Minute (BPM). Then, equation 4.2 is used to calculate 60 seconds of average BPM where the IBI collected is in millisecond.

$$IB(average) = \frac{IBI_1+IBI_2+IBI_3+\cdots+IBI_{10}}{10} \quad (4.1)$$

$$BPM = \frac{60000}{IBI(average)} \quad (4.2)$$

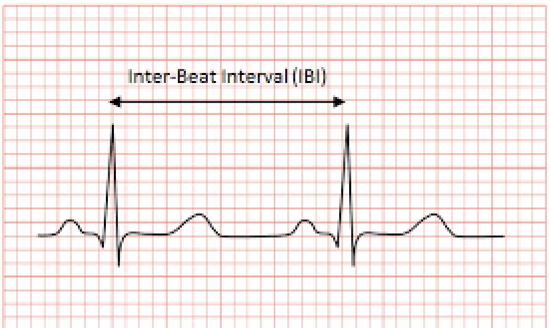


Figure 4.6 IBI measurement

The LM35 temperature sensor used can measure temperature from 0 °C up to 100 °C. The code sketch use equation 4.3 and equation 4.4 to calculate the value of the temperature in degree Celsius. The LM35 works at an input voltage of 4 to 20V and has analog output voltage. With each degree Celsius means the sensitivity is 10mV, for the output range 0.3V is equal to 30 ° C. The analog pin in NodeMCU converts the analog voltage input from 0 to 1023 through the IC multiplexer to digital output range. This value is converted to voltage output by splitting 1024 and multiplying by reference voltage from equation 4.3. Lastly, by using equation 4.4, the output voltage needs to be converted to temperature value by divide with the sensitivity of the LM35.

$$Vout = \frac{\text{analogRead(tempPin)}\times 5}{1024} \tag{4.3}$$

$$Temperature = \frac{Vout}{0.01} \tag{4.4}$$

4.4 Analysis on Project System

4.4.1 System performance through data comparison between two devices

1) Pulse Rate Sensor

The pulse sensor is being tested for its accuracy and reliability for checking the patient's heart beat. From **Figure 4.7**, BPM reading values of pulse rate sensor is compared with actual pulse reading device which is Omron Automatic Blood Pressure Monitor. The heartbeat detector in blood pressure monitor is achievable when provide upper arm blood pressure measurements. This medical device proves to be accurate with +5% of pulse reading. The BPM obtained from pulse sensor have slightly difference between actual measurements. From **Table 4.1**, the average BPM obtained using pulse senor is 64.92 BPM and actual sensor is 63.75 BPM. Therefore, from the measure and actual data obtained, the error of reading is around 1.84 % and give accuracy of 98.16% for the pulse sensor

$$\%error = \frac{|Avg\ measured - Avg\ actual|}{Avg\ actual} \times 100\%$$

$$\%error = \frac{|64.92 - 63.75|}{63.75} \times 100\% = 1.84\%$$

$$\%accuracy = 100\% - 1.84\% = 98.16\%$$

Time (min)	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51
Measured	73	88	73	70	65	78	72	56	60	67	65	66	64	65	53	57	62	58
Actual	74	78	72	65	68	75	66	58	61	62	56	56	62	60	57	58	63	64

Time (min)	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99	102	105
Measured	53	66	56	68	65	60	62	75	66	72	56	63	61	73	70	59	60	60
Actual	54	64	58	61	52	64	64	71	69	80	64	51	65	68	65	61	61	68

Table 4.1 Table of measured and actual BPM value

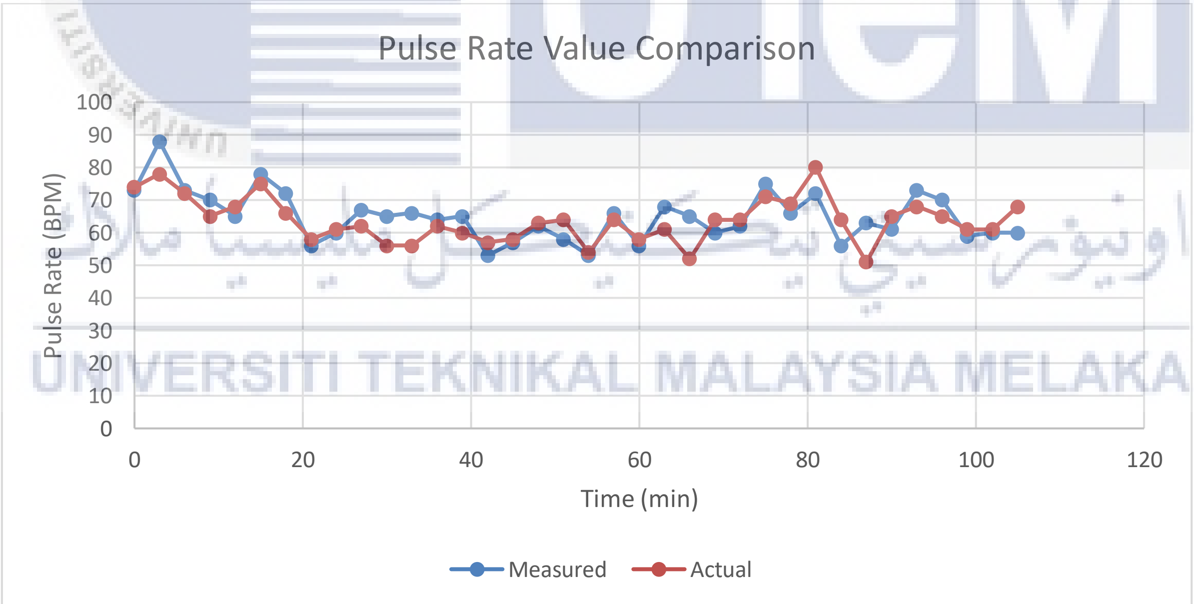


Figure 4.7 Line graph of measured and actual BPM

2) LM35 Temperature Sensor

The LM35 temperature sensor used in this project is a low-cost sensor capable of providing a sensitivity of 0.5 ° C that is suitable for body temperature. The temperature sensor had undergone several tests and compared with Digital Thermometer device. From Figure 4.8, the measured temperature using LM35 has varied value due to its accuracy as expected from datasheet. The average measured temperature obtained is around 35.12 ℃ and compared with actual temperature which is 36.34 ℃, the percentage of error is 3.36%. Hence, the accuracy of the sensor for this project is around 96.64%.

$$\%error = \frac{|Avg\ measured - Avg\ actual|}{Avg\ actual} \times 100\%$$

$$\%error = \frac{|35.12 - 36.34|}{36.34} \times 100\% = 3.36\%$$

$$\%accuracy = 100\% - 3.36\% = 96.64\%$$

Time	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69
Measured	36.77	36.45	36.45	36.77	36.7	36.45	35.16	36.77	36.45	36.45	36.77	36.45	36.13	36.13	36.13	36.45	36.13	36.13	36.13	36.13	36.45	36.45	36.43	36.45
Actual	36.6	36.5	36.5	36.6	36.45	36.3	35.5	36.6	36.5	36.1	36.5	36.6	36.2	36.2	36.2	36.6	36.2	36.2	36.3	36.2	36.3	36.3	36.3	36.4

Table 4.2 Table of measured and actual temperature value

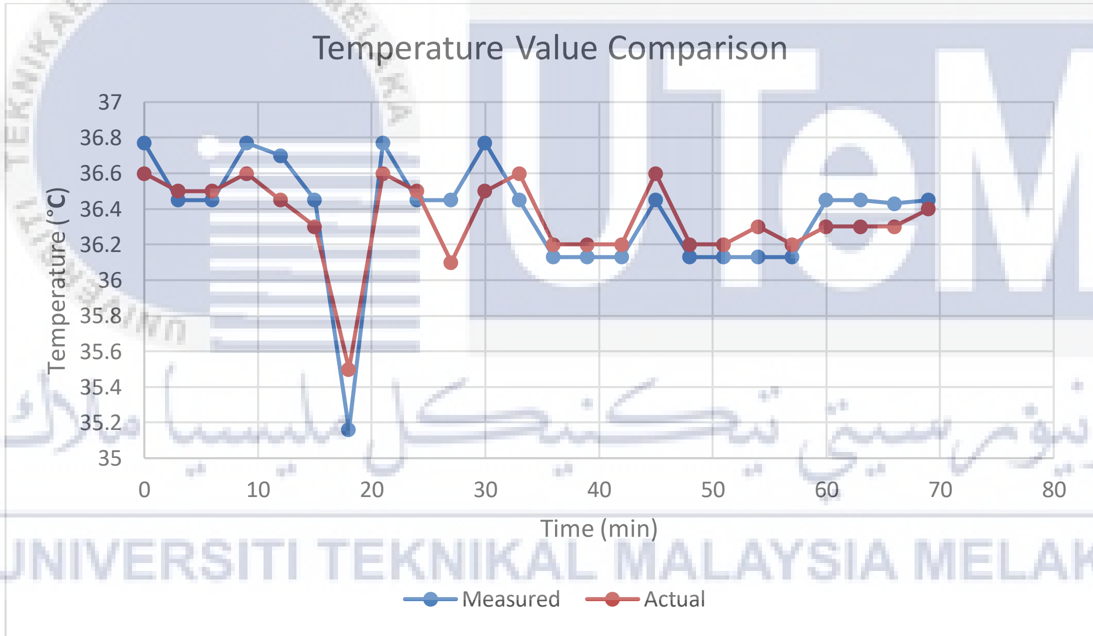


Figure 4.8 Line graph of measured and actual temperature

4.4.2 Wireless transmission through distance measurement

Theoretically, Free-space path loss (FSPL) is the loss of signal strength of an electromagnetic wave resulting from a line-of-sight path through free space, with no obstacles nearby causing reflection or diffraction. In a Wi-Fi environment, FSPL refers to the amount of power a Wi-Fi signal losses as it travels away from the transmitter. For this analysis, several samples of distance are recorded to find out the wireless signal strength of each distance when wearable device is away from the WiFi router. To record the distances, a measuring tape was used to measure each distance where data can stored from sensors for monitoring system.

The maximum distance where the device received the last data stored to database is around 78.75ft which equals to 24 m. Wifi router's bandwidth is around 300Mbps and household coverage around 1500sq.ft.

In terms of Wavelength: $\mathrm{FSPL} = \left(\frac{4\pi d}{\lambda}\right)^2$

In terms of Frequency: $\mathrm{FSPL} = \left(\frac{4\pi d f}{c}\right)^2$

- λ : Wavelength of signal
- c: Speed of light = 3×10^8 **ms-¹**

In decibels, this would be equivalent to

Free Space Path Loss (dB) = 20 log (d) + 20 log (f) – 27.55

Where: d= distance from the transmitter (m), f = signal frequency (MHz)

For my application I used the recommended constant -27.55, which treats frequency in MHz and distance in meters (m).

Then, FSPL are calculated with 2.4GHz and 5GHz provided from Wi-Fi router as shown in Table and graph shown in Figure 4.89.

Distance (m)	Loss signal strength(dB) for 2432MHz	Loss signal strength(dB) for 5745MHz
3	49.71	57.18
6	55.73	63.20
9	59.25	66.72
12	61.75	69.22
15	63.69	71.16
18	65.27	72.74
21	66.61	74.08
24	67.77	75.24

Table 4.3 Table of both frequencies and distances measured

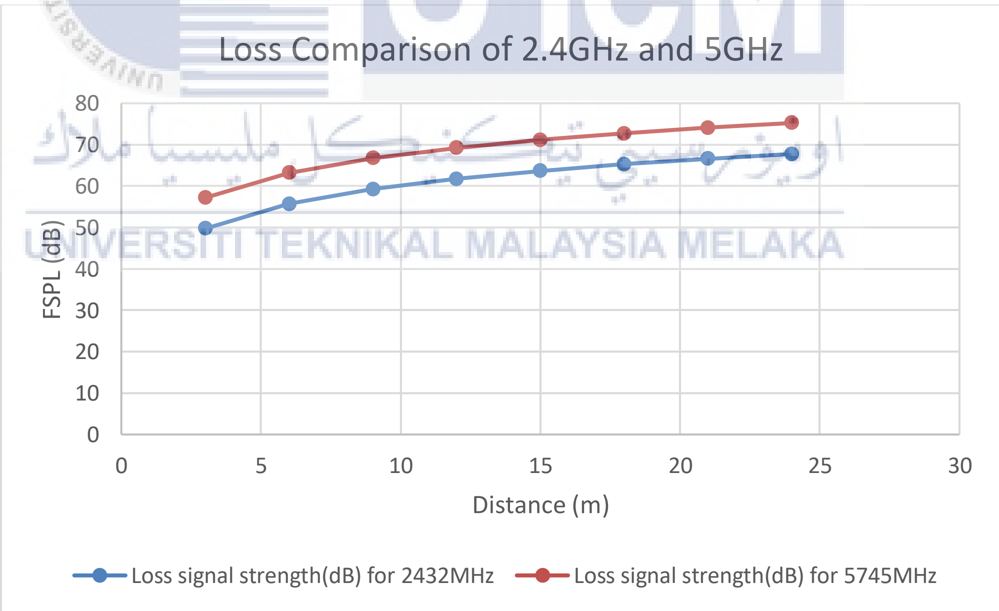


Figure 4.9 Line graph of loss comparison of 2.4GHz and 5GHz

Based on Table 4.3 below, -49.71dB is max signal strength while -67.77db is reliable signal strength for 2.4GHz compare to 5GHz which have higher signal loss. Hence, the wearable system can be monitored afar from WIFI source up to 24m.

SIGNAL STRENGTH	EXPECTED QUALITY	REQUIRED FOR
-30 dBm	Maximum signal strength, you are probably standing right next to the access point.	
-50 dBm	Anything down to this level can be considered excellent signal strength.	
-60 dBm	Good, reliable signal strength.	
-67 dBm	Reliable signal strength.	The minimum for any service depending on a reliable connection and signal strength, such as voice over Wi-Fi and non-HD video streaming.
-70 dBm	Not a strong signal.	Light browsing and email.
-80 dBm	Unreliable signal strength, will not suffice for most services.	Connecting to the network.
-90 dBm	The chances of even connecting are very low at this level.	

Table 4.4 Signal Strength quality

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

This project is a low-cost with efficient monitoring system which collects health parameters using cheap sensors. Therefore, the result was not fully accurate due to the difference between the measured and actual values collected. The main goal is to design wearable healthcare monitoring system for the patients and to ensure that the user can keep track of the status of the patient through the webpage even if they are far from home. The fully integrated monitoring system with IoT implementations consist of monitoring webpage and push notification to smartphone of any user including doctor. The healthcare system are also able to send data wirelessly through cloud server and can be monitored anytime using integrated ESP8266 Wi-Fi development in NodeMCU board. This will allow wearable system to be placed on patient’s wrist with continuous database monitoring.

Owing to some technical difficulties the first objective was accomplished partially. Around the same time, the uses of both pulse and temperature sensor triggered slightly inaccurate reading. This is because of limitation of analog inputs of NodeMCU board that cause two analog sensors to connect IC multiplexer. This connection might cause interruption on both sensors' reading. To solve this, controller board with multiple analog and digital inputs can be used to attach many sensors with one board. For monitoring system, the data sent is displayed on webserver. By using Android application, an IoT-based platform will be created. A push notification also can be sent to email with integration of IFTTT during abnormal condition.

Next, the second objective is achievable which consists of analyzing system performance and its efficiency. Sensor reading is evaluated and determined based on measured value and actual value. It is known that the LM35 temperature sensor can produce accurate results while the pulse sensor has a slightly lower accuracy due to the pulse sensor circuitry and NodeMCU controller response delay interrupt configuration.

Moreover, this project has high commercial value because of total cost of project is around RM67.74 which is cheaper than existing medical devices on the market. Besides, the system is easy to use where patient can wear the device on the wrist properly and the health status automatically displayed on database platform. Furthermore, this system device are able read two readings compare to most medical devices which can have one reading only. Lastly, the system uses rechargeable battery and non-harmful substance which proves that it is environmentally friendly.

5.2 Future work

For the future work, this project will focus more on the development of the webpage, with some additional medication and lifestyle notification provided by the webpage that will enable patient to be more cautious remotely through the website more effectively.

Furthermore, this project could be designed by adding more sensors that detect other health parameters. These sensors can be used by consumers by selecting which parameter they need to use. Besides, the web UI can be upgraded to perform many activities such as displaying real-time graph, controlling sensors and observe anomalies.

Nowadays, cyber criminals can hack data could cause data. Hence, the uses of security system that encrypt the IoT system could solve this. This security algorithm will help users avoid any type of security attack in their network and improve data privacy.

Lastly, the project can be added with extra features such as linking it with ambulance service that automatically called the ambulance if patient is in critical condition. Next feature should add a GPS tracker in the system, so the doctor or caretaker can know patient location at all time with the location display all the time on the webpage.

REFERENCES

[1] M. Aminian, “A Hospital Healthcare Monitoring System Using Wireless Sensor Networks,” *J. Heal. Med. Informatics*, vol. 04, no. 02, pp. 4–9, 2013.

[2] Maiken Scott, “Beep, beep, beep – hospital alarms sound mostly without real cause,” *the pulse*, 2015. [Online]. Available: https://whyy.org/segments/beep-beep-beep-hospital-alarms-sound-mostly-without-real-cause/. [Accessed: 21-Oct-2018].

[3] Nasrullah Patel, “Internet of things in healthcare: applications, benefits, and challenges,” 2018. [Online]. Available: https://www.peerbits.com/blog/internet-of-things-healthcare-applications-benefits-and-challenges.html. [Accessed: 21-Oct-2018].

[4] Medgadget, “Wireless Communications Technologies In Healthcare Market – Global Industry Analysis, Trends And Forecast 2015 – 2023 | Center for Wireless Communications,” *Global Wireless News*, 2015. [Online]. Available: http://cwc.ucsd.edu/content/wireless-communications-technologies-healthcare-market-—-global-industry-analysis-trends-and. [Accessed: 22-Oct-2018].

[5] Andrew Baker, "COVID-19 Pandamic Highlights Importance of Remote Patient Monitoring Technologies, 2020. [Online]. Available: https://semiengineering.com/covid-19-pandemic-highlights-importance-of-remote-patient-monitoring-technologies/. [Accessed: 13-June-2020].

[6] Alfian, G., Syafrudin, M., Ijaz, M. F., Syaekhoni, M. A., Fitriyani, N. L., & Rhee, J. (2018). A Personalized Healthcare Monitoring System for Diabetic Patients by Utilizing BLE-Based Sensors and Real-Time Data Processing. *Sensors (Basel, Switzerland)*, *18*(7), 2183. https://doi.org/10.3390/s18072183.

[7] Shelar, M., Singh, J., & Tiwari, M. (2013). Wireless patient health monitoring system. *International Journal of Computer Applications*, *62*(6).

[8] Priyanka, R., & Reji, M. (2019). IOT Based Health Monitoring System Using Blynk App. *International Journal of Engineering and advanced Technology*, *8*(6).

[9] Gondalia, A., Dixit, D., Parashar, S., Raghava, V., Sengupta, A., & Sarobin, V. R. (2018). IoT-based healthcare monitoring system for war soldiers using machine learning. *Procedia computer science*, *133*, 1005-1013.

[10] C. S. Krishna and N. Sampath, "Secured Smart Healthcare Monitoring System Based on IoT," *2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2017*, pp. 4958–4961, 2018.

[11] M. Shelar, "Wireless Patient Health Monitoring System," vol. 62, no. 6, pp. 2–6, 2013.

[12] A. Gondalia, D. Dixit, S. Parashar, V. Raghava, A. Sengupta, and V. R. Sarobin, "IoT-based Healthcare Monitoring System for War Soldiers using Machine Learning," *Procedia Comput. Sci.*, vol. 133, pp. 1005–1013, 2018.

[13] A. Sagahyroon, H. Raddy, A. Ghazy, and U. Suleman, "Design and implementation of a wearable healthcare monitoring system," *Int. J. Electron. Healthc.*, vol. 5, no. 1, p. 68, 2009.

[14] A. Dosinas and M. Vaitk, "Measurement of Human Physiological Parameters in the Systems of Active Clothing and Wearable Technologies," vol. 7, no. 7, pp. 77–82, 2006.

[15] Louise McCallum, "Measuring body temperature | Practice | Nursing Times," vol. 108, no. 45, pp. 20–23, 2016.

[16] "Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure)," *Johns Hopkins Medicine.* [Online]. Available: https://www.hopkinsmedicine.org/healthlibrary/conditions/cardiovascular_diseases/vital_signs_body_temperature_pulse_rate_respiration_rate_blood_pressure_85,p00866. [Accessed: 12-Nov-2018].

[17] F. E. N. Matthew D. White, Catharine M. Bosio, Barry N. Duplantis, "Human body stemperature and new approaches to constructing temperature-sensitive bacterial vaccines," *Cell. Mol. Life Sci.*, vol. 68, pp. 3019–3031, 2011.

[18] M. Shah, H. Solanki, K. Engineer, H. Sheth, and A. Consultants, "IOT BASED PATIENT HEALTH MONITORING SYSTEM," *Int. J. Adv. Eng. Res. Dev.*, pp. 681–689, 2017.

[19] O. Ugweje, *Radio Frequency and Wireless Communications*, no. April 2004. 2017.

[20] S. Joseph, S. Akhil, A. K. Narayanankutty, P. M. Amalraj, and A. P. K, “RF Transceiver Based Traffic Alert System for Automobiles,” pp. 7623–7627, 2014.

[21] Egham, “Gartner Says 8.4 Billion Connected ‘Things’ Will Be in Use in 2017, Up 31 Percent From 2016,” *gartner*, U.K., 07-Feb-2017.

[22] R. H. Sudhan, M. G. Kumar, A. U. Prakash, S. A. R. Devi, and S. P., “Arduino Atmega-328 Microcontroller,” *Ijireeice*, vol. 3, no. 4, pp. 27–29, 2015.

[23] L. Joseph, M. Sonia, P. D. Riya, S. Tom, S. P. Andrews, and M. John, “Rf Chat,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 4, pp. 1889–1893, 2017.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

1. Syahmi Al-Zubair Bin Abdullah, En. Mohd Shahril Izuan bin Mohd Zin, "Design and Analysis of Wireless Healthcare Monitoring System Based on IoT", 2019, Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia, will be publish on IEEE-publication.

APPENDICES

Appendix A: LM35 datasheet

National Semiconductor

November 2000

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

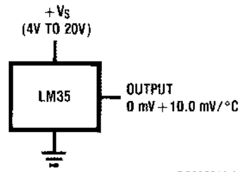
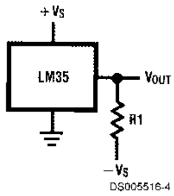


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)

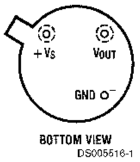


Choose $R_1 = -V_S/50\ \mu A$
V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

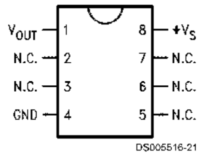
TO-46
Metal Can Package*



*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

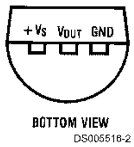
SO-8
Small Outline Molded Package



N.C. = No Connection

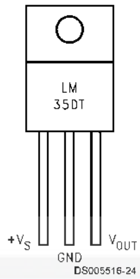
Top View
Order Number LM35DM
See NS Package Number M08A

TO-92
Plastic Package



Order Number LM35CZ, LM35CAZ or LM35DZ
See NS Package Number Z03A

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).
Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.;	
TO-46 Package,	−60°C to +180°C
TO-92 Package,	−60°C to +150°C
SO-8 Package,	−65°C to +150°C
TO-220 Package,	−65°C to +150°C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300°C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55°C to +150°C
LM35C, LM35CA	−40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T_A=+25°C	±0.2	±0.5		±0.2	±0.5		°C
	T_A=−10°C	±0.3			±0.3		±1.0	°C
	T_A=T_{MAX}	±0.4	±1.0		±0.4	±1.0		°C
	T_A=T_{MIN}	±0.4	±1.0		±0.4		±1.5	°C
Nonlinearity (Note 8)	$T_{MIN}\leq T_A\leq T_{MAX}$	**±0.18**		**±0.35**	**±0.15**		**±0.3**	°C
Sensor Gain (Average Slope)	$T_{MIN}\leq T_A\leq T_{MAX}$	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/°C
Load Regulation (Note 3) $0\leq I_L\leq 1$ mA	T_A=+25°C	±0.4	±1.0		±0.4	±1.0		mV/mA
	$T_{MIN}\leq T_A\leq T_{MAX}$	**±0.5**		**±3.0**	**±0.5**		**±3.0**	mV/mA
Line Regulation (Note 3)	T_A=+25°C	±0.01	±0.05		±0.01	±0.05		mV/V
	$4V\leq V_S\leq 30V$	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25°C	56	67		56	67		μA
	V_S=+5V	**105**		**131**	**91**		**114**	μA
	V_S=+30V, +25°C	56.2	68		56.2	68		μA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	μA
Change of Quiescent Current (Note 3)	$4V\leq V_S\leq 30V$, +25°C	0.2	1.0		0.2	1.0		μA
	$4V\leq V_S\leq 30V$	**0.5**		**2.0**	**0.5**		**2.0**	μA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	μA/°C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	T_A=+25°C	±0.4	±1.0		±0.4	±1.0		°C
	T_A=−10°C	±0.5			±0.5		±1.5	°C
	$T_A=T_{MAX}$	±0.8	±1.5		±0.8		±1.5	°C
	$T_A=T_{MIN}$	±0.8		±1.5	±0.8		±2.0	°C
Accuracy, LM35D (Note 7)	T_A=+25°C				±0.6	±1.5		°C
	$T_A=T_{MAX}$				±0.9		±2.0	°C
	$T_A=T_{MIN}$				±0.9		±2.0	°C
Nonlinearity (Note 8)	$T_{MIN} \le T_A \le T_{MAX}$	**±0.3**		**±0.5**	**±0.2**		**±0.5**	°C
Sensor Gain (Average Slope)	$T_{MIN} \le T_A \le T_{MAX}$	**+10.0**	**+9.8, +10.2**		**+10.0**		**+9.8, +10.2**	mV/°C
Load Regulation (Note 3) $0 \le I_L \le 1$ mA	T_A=+25°C	±0.4	±2.0		±0.4	±2.0		mV/mA
	$T_{MIN} \le T_A \le T_{MAX}$	**±0.5**		**±5.0**	**±0.5**		**±5.0**	mV/mA
Line Regulation (Note 3)	T_A=+25°C	±0.01	±0.1		±0.01	±0.1		mV/V
	$4V \le V_S \le 30V$	**±0.02**		**±0.2**	**±0.02**		**±0.2**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25°C	56	80		56	80		μA
	V_S=+5V	**105**		**158**	**91**		**138**	μA
	V_S=+30V, +25°C	56.2	82		56.2	82		μA
	V_S=+30V	**105.5**		**161**	**91.5**		**141**	μA
Change of Quiescent Current (Note 3)	$4V \le V_S \le 30V$, +25°C	0.2	2.0		0.2	2.0		μA
	$4V \le V_S \le 30V$	**0.5**		**3.0**	**0.5**		**3.0**	μA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.7**	**+0.39**		**+0.7**	μA/°C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	$T_J=T_{MAX}$, for 1000 hours	±0.08			±0.08			°C

Note 1: Unless otherwise noted, these specifications apply: $-55°C \le T_J \le +150°C$ for the LM35 and LM35A; $-40° \le T_J \le +110°C$ for the LM35C and LM35CA; and $0° \le T_J \le +100°C$ for the LM35D. V_S=+5Vdc and I_{LOAD}=50 μA, in the circuit of *Figure 2*. These specifications also apply from +2°C to T_{MAX} in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W, junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and 10mv/°C times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in °C).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of *Figure 1*.

Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

Appendix B: AT command Wi-Fi and TCP/IP

4. Wi-Fi AT Commands

4.1. Overview

Commands	Description
AT+CWMODE_CUR	Sets the Wi-Fi mode (Station/AP/Station+AP); configuration not saved in the flash.
AT+CWMODE_DEF	Sets the default Wi-Fi mode (Station/AP/Station+AP); configuration saved in the flash.
AT+CWJAP_CUR	Connects to an AP; configuration not saved in the flash.
AT+CWJAP_DEF	Connects to an AP; configuration saved in the flash.
AT+CWLAPOPT	Sets the configuration of command AT+CWLAP.
AT+CWLAP	Lists available APs.
AT+CWQAP	Disconnects from an AP.
AT+CWSAP_CUR	Sets the current configuration of the ESP8266 SoftAP; configuration not saved in the flash.
AT+CWSAP_DEF	Sets the configuration of the ESP8266 SoftAP; configuration saved in the flash.
AT+CWLIF	Gets the Station IP to which the ESP8266 SoftAP is connected.
AT+CWDHCP_CUR	Enables/Disables DHCP; configuration not saved in the flash.
AT+CWDHCP_DEF	Enable/Disable DHCP; configuration saved in the flash.
AT+CWDHCPS_CUR	Sets the IP range of the DHCP server; configuration not saved in the flash.
AT+CWDHCPS_DEF	Sets the IP range of the DHCP server; configuration saved in the flash.
AT+CWAUTOCONN	Connects to an AP automatically on power-up.
AT+CIPSTAMAC_CUR	Sets the MAC address of the ESP8266 Station; configuration not saved in the flash.
AT+CIPSTAMAC_DEF	Sets the MAC address of ESP8266 station; configuration saved in the flash.
AT+CIPAPMAC_CUR	Sets the MAC address of the ESP8266 SoftAP; configuration not saved in the flash.
AT+CIPAPMAC_DEF	Sets the MAC address of the ESP8266 SoftAP; configuration saved in the flash.
AT+CIPSTA_CUR	Sets the IP address of the ESP8266 Station; configuration not saved in the flash.
AT+CIPSTA_DEF	Sets the IP address of the ESP8266 Station; configuration saved in the flash.
AT+CIPAP_CUR	Sets the IP address of ESP8266 SoftAP; configuration not saved in the flash.

5. TCP/IP-Related AT Commands

5.1. Overview

Command	Description
AT+CIPSTATUS	Gets the connection status
AT+CIPDOMAIN	DNS function
AT+CIPSTART	Establishes TCP connection, UDP transmission or SSL connection
AT+CIPSSLSIZE	Sets the size of SSL buffer
AT+CIPSSLCONF	Set configuration of ESP SSL client
AT+CIPSEND	Sends data
AT+CIPSENDEX	Sends data when length of data is <length>, or when \0 appears in the data
AT+CIPSENDBUF	Writes data into TCP-send-buffer
AT+CIPBUFRESET	Resets the segment ID count
AT+CIPBUFSTATUS	Checks the status of TCP-send-buffer
AT+CIPCHECKSEQ	Checks if a specific segment is sent or not
AT+CIPCLOSE	Closes TCP/UDP/SSL connection
AT+CIFSR	Gets the local IP address
AT+CIPMUX	Configures the multiple connections mode
AT+CIPSERVER	Deletes/Creates a TCP server
AT+CIPSERVERMAXCONN	Set the maximum connections that server allows
AT+CIPMODE	Configures the transmission mode
AT+SAVETRANSLINK	Saves the transparent transmission link in the flash
AT+CIPSTO	Sets timeout when ESP8266 runs as TCP server
AT+PING	Ping packets
AT+CIUPDATE	Upgrades the software through network
AT+CIPDINFO	Shows remote IP and remote port with +IPD
+IPD	ESP receives network data
AT+CIPRECVMODE	Set TCP Receive Mode
AT+CIPRECVDATA	Get TCP Data in Passive Receive Mode
AT+CIPRECVLEN	Get TCP Data Length in Passive Receive Mode
AT+CIPSNTPCFG	Configures the time domain and SNTP server.
AT+CIPSNTPTIME	Queries the SNTP time.