Sudan University of Science and Technology

College of Postgraduate Studies

Design of Real -Time Patient Health Monitoring and Alarming System using Internet of Things

تصميم نظام مراقبة صحة المريض والإنذار في الوقت الحقيقي بإستخدام إنترنت الأشياء

A Thesis Submitted in Partial Fulfillment for the Requirements of the Degree of M.Sc. in Computer and Network Engineering

Prepared By:

Tasneem Salah El-Deen Osman Hussein

Supervised By:

Dr. Hisham Ahmed

April 2021

الآيـــة

قال تعالى في محكم تنزيله:

{ قُل لَّوْ كَانَ الْبَحْرُ مِدَادًا لِّكَلِمَاتِ رَبِّي لَنَفِدَ الْبَحْرُ قَبْلَ أَن تَنفَدَ

كَلِمَاتُ رَبِّي وَلَوْ جِئْنَا بِمِثْلِهِ مَدَدًا }

(الكهف: 109)

صَدَقَ اللهُ العَظِيم

DEDICATION

To my family, with all due love......

ACKNOWLEDMENT

I would like to express my appreciation and thank my supervisor, Dr. Hisham Ahmed, for his support and guidance throughout my research studies. His continuous encouragement and advice were greatly appreciated. Thanked for his comments improving the final version of this thesis. I would like also to thank my friends who provided me with the facilities being required and conductive conditions for my project. Finally, last but not least thank my parents for their endless support and encouragement. I could not have done it without their support.

ABSTRACT

Internet of Things makes all objects interconnected and it has been recognized as the next technical revolution. One such application is in healthcare to monitor the patient health status. This system is developed for the home use by patients in rural areas that do not have healthcare professionals. Real time monitoring of the patient's health status is displayed in the web application so that the doctors can monitor and access the patient information from anywhere of the world at any time. In this project sensors are used to measure the temperature and heart rate of the patient. The raspberry pi board is used to collect data from the sensors and then it sends wirelessly to a web server using web socket protocol. The sensor data processing and Patient's record will be stored on the cloud. According to the relative error calculated it can be said that the developed system less expensive and accurate.

المستخلص

إنترنت الأشياء يجعل كل الأشياء مترابطة عبر الانترنت وقد تم الاعتراف بها على أنها الثورة التقنية التالية. أحد هذه التطبيقات هو في مجال الرعاية الصحية لمراقبة حالة المريض الصحية. تم تطوير هذا النظام للاستخدام المنزلي من قبل المرضى في المناطق الريفية التي ليس لديها أخصائيي الرعاية الصحية. يتم عرض المراقبة في الوقت الحقيقي للحالة الصحية للمريض في تطبيق الشبكة بحيث يمكن للأطباء مراقبة معلومات المريض والوصول إليها من أي مكان في العالم في أي وقت. في هذا المشروع يتم استخدام أجهزة الاستشعار لقياس درجة الحرارة ومعدل ضربات القلب. يتم استخدام لوحة الراسبري باي لتجميع البيانات من المستشعرات ثم ارسالها لاسلكيا الى خادم الشبكة باستخدام نظام اتفاقيات مقبس الشبكة. يتم تخزين معالجة بيانات الاستشعار وسجل المريض على السحابة الالكترونية.وفقا للخطأ النسبي المحسوب يمكن القول أن النظام المتقدم أقل تكلفة ودقته جيدة.

Contents

List of Tables

List of Figures

List of Symbols

◦	Centigrade
∑	Sum of
µ	Mean of data set
σ	Standard Deviation
V	Volt
mA	milliAmpere
W	Watt
C	Celsius
x	Value in the data set
N	Number of data points

Abbreviations

ADC	Analog to Digital Convertor
ARM	Acorn Risk Machines
AJAX	Asynchronous JavaScript and XM
BPM	Beats Per Minutes
BSN	Body Sensor Network
CPU	Central Processing Unit
DVD	Digital Versatile Disc
ECG	Electro Cardio Gram
GUI	Graphical User Interface
GPU	Graphics Processing Unit
GPIO	General Purpose Input/output
GND	Ground

GSM	Global System for Mobile
HDMI	High Definition Multimedia Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IoT	Internet of Things
IP	Internet Protocol
LCD	Liquid Cristal Display
LED	Light Emitting Diode
OpAmP	Operational Amplifier
RDBMS	Relational Data Base Management System
SDRAM	Synchronous Dynamic Random Access Memory
SQL	Structured Query Language
SSE	Server Sent Events

TCP	Transmission Control Protocol
USB	Universal Serial Bus
WBSN	Wireless Body Sensor Network

CHAPTER ONE

INTRODUCTION

Chapter One
Introduction

1.1 Preface

Nowadays, Internet has become one of the imperative parts of our daily life. It has changed how people live, work, play and learn. Internet serves for numerous ideas such as in education, finance, business, industries, entertainment, social networking, shopping, e-commerce etc. The next innovative mega trend of Internet is Internet of Things (IoT).

The IoT connects smart objects to the Internet. It can facilitate an exchange of data never available before, and bring users information in a more secured way. Medical care and health care represent one of the most attractive application areas of the IoT. The IoT has the potential to give rise to many medical applications such as remote health monitoring, fitness programs, and elderly care. In the field of health monitoring the current most important user groups are those aged 40 and more [1]. In a hospital either the nurse or the doctor has to move physically from one person to another for health check, which may not be possible to monitor their conditions continuously. Thus any critical situations cannot be found easily unless the nurse or doctor checks the person's health at that moment. This may be a strain for the doctors who have to take care of a lot number of people in the hospital. In order to keep in track of critical health conditions, a real time health monitoring system of patient based on internet of things will be designed and developed in this project. This finds vast application in the remote places where the people are out of reach from the experienced doctors. Hence the entire project can be broadly divided into four sections firstly, the parameters measured from the patient and transmitted, secondly the signal processing and conversion to digital form; thirdly decision making with the help of an algorithm where they obtained signal values are compared with the standard values and finally the transmission of the condition of the patient to the doctor. IoT-based health care services are foreseen to reduce costs, increase the quality of life, and enrich the user's experience [2].

1.2 Problem Statement

Everyday many lives are affected by heart attacks and more importantly because the patients did not get timely and proper help. Care of critically ill patient, requires instantaneous and accurate decisions so that lifesaving therapy can be properly applied. In rural areas, health care delivery systems struggle to maintain adequate numbers of clinical staff to serve their patient populations. Lack of access to medical specialists is not just a rural problem. Many urban areas also do not have enough specialists to provide proper health care.

1.3 Proposed Solution

The problems can be solve by using internet of things to ensure the patients can be monitor continuously by doctors, nurses or caregivers anywhere and anytime even though the patients stay at home. Besides, the costs for wiring and installation might be reducing as well.

1.4 Objectives

The objectives of the project are:

i. To design a health system for monitoring temperature and heart beat of the patient.

ii. To further transmitting this information on the cloud server through the internet.

iii. Also in case of emergency, an alarm will alert care takers of the patient.

1.5 Methodology

- The proposed method of patient monitoring system is to monitor patient's body temperature and heart rate by connecting sensors (LM35 for temperature, and pulse sensor for the heartbeat) to the patient body. These sensors output is then received by the Raspberry Pi.
- The raspberry pi will connect to the internet and then send the data of the sensors to a web server using web socket protocol.
- The sensor's data will store in the data base using MySQL to make it easy for doctors to monitor the status of the patient at real time or even for a time before.

1.6 Thesis Outlines

The thesis will be organized as follows. Chapter 2 contains two parts, background and the literature review. In the first part of this chapter the components of the system and tools will be discussed. The second part presents some scientific papers and literature review of the real time health system. Chapter 3 discusses methodology that will be used on the project design includes the programming, the software and the hardware. Chapter 4 presents the implementation results and discussion. Finally chapter 5 contains explanations about the conclusion and recommendation that can make for future improvements

CHAPTER TWO

BACKGROUND AND RELATED WORK

2 Background and related work

2.1 Background

Internet of Things is a network of devices that is built with embedded systems, electronic things, actuators, sensors and network connectivity and which let these objectives to exchange and gather the data information. Using internet of things objects can be sensed and controlled from the present network. The increasing generation needs empowered instruments by wireless technology which involves Bluetooth, Radio frequency identification, embedded sensors and many more. The normal human life can be change to smart life using the new technologies of IOT. Internet of Things is used to monitor all patients at any situation. Health plays an important role in our life. The patients who are suffering from chronicle diseases they need to occupy check-up daily. Manually it is very hard to keep track on the heart beat abnormalities of the patient. The normal heart rate for adult's ranges from 60 to 100 beats per minute while for an old person the heartbeat range is between 54 to 91 beats per minutes. The above or below this range of heart beat leads to heart attacks and the normal body temperature is ranges from 97°F(36.1°C) to 99°F to (37.2°C). A temperature over 100.4°F (38°C) it means the person is having fever caused by an infection or illness [1]. There are varied devices available in the market in order to keep track on the inner changes of the body. But there are many restrictions in the maintenance due to the size of the device, high cost, and the portability of the device. So patient health monitoring system is design in a small size, low cost and a portable device so as to provide a continuous monitoring of the patient health. The proposed method of patient health monitoring system is to monitor patient's body temperature, heart rate using Raspberry Pi3. The temperature sensor senses the temperature from the patient's body and forwards the information to the raspberry pi3. The pulse sensor collects the heart beat from the patient, the information obtained from the sensor output sends to the raspberry pi3. After connecting internet to the Raspberry Pi3 it works as a server. Then the server is automatically sends information to the website. Using IP address anybody can monitor the patient's health case anywhere in the world.

2.1.1 The Raspberry pi

A group of computer scientists led by Eben Upton at the University of Cambridge's Computer Laboratory in 2006 struck upon the idea of producing a cheap educational micro-computer geared towards the amateur computer enthusiast, budding students, and children. The aim was to help to provide the skills to future Computer Science undergraduate applicants that many of those applying in the '90s possessed, thanks to the home computers of the '80s. However it would be another two years before the project became viable, and not until 2012 before the Raspberry Pi was being shipped out to the public. The 2000s saw a huge growth in mobile computing technologies, a large segment of this being driven by the mobile phone industry. By 2005, ARM–a British manufacturer of CPU core components and a by-product of the '80s home computer company Acorn, had grown to where 98 percent of mobile phones were using their technology. This translated into around 1 billion CPU cores. During the same period, Ebon Upton designed several concepts for the Raspberry Pi and by 2008, thanks to a by-product of the increasing penetration of mobile phone technology, the cost of building a miniature, portable microcomputer with many of the multimedia functions that the public were accustomed to becoming viable. Thus the Raspberry Pi foundation was formed and set about the task of developing and manufacturing the Raspberry Pi computer. By 2011, the first Alpha models were being produced and tested, and the public finally got to see what the Raspberry Pi was capable of. Finally in 2012, the Raspberry Pi was ready for public consumption. Two versions of the Raspberry pi were scheduled to be manufactured, namely models A and B, with B being released first. The model A board which will not include an Ethernet port and will consume considerably less power than the model B was given a price tag of $25. The model B that includes an Ethernet port was given a target price of $35 USD and manufacturing in China started. This would later be moved to the UK with Sony taking over the process [13].

- **Raspberry pi Hardware Specification**

The Raspberry Pi is built off the back of the Broadcom BCM2835. The BCM2835 is a multimedia application processor geared towards mobile and embedded devices. On top of this, several other components have been included to support USB, RCA, and SD card storage. The following figure highlights some of these specifications with a description of each provided:

Figure 2.1: Raspberry Pi3 board

- Dimension

  The Raspberry Pi is a small device coming in at 85.60mm x 53.98mm x 17mm and weighing only 45g. This makes it perfect for home automation, where a small device can be placed in a case and mounted inside an electrical box, or replace an existing thermostat device on a wall.

- 3.5mm analog audio jack

The 3.5mm analog audio jack allows you to connect headphones and speakers to the Raspberry Pi. This is especially useful for audio and media player based projects.

- Composite RCA port

You are probably familiar with the composite cables used to hook up your DVD player to the TV. They usually come in the red, white, and yellow plug variety. The Raspberry Pi has a port for attaching the yellow video cable from your TV to it, allowing you to use your TV as a monitor.

- Two USB 2.0 ports plus one micro USB

USB is one of the most common methods for connecting peripherals and storage devices to a computer. The Raspberry Pi comes equipped with two of them, allowing you to hook up a keyboard and mouse when you get started and a micro USB port for powering your device.

• High Definition Multi-media Interface port

The High Definition Multi-media Interface (HDMI) port allows the Raspberry Pi to be hooked up to high-definition televisions and monitors that support the technology. This provides an additional option to the composite RCA port for video and additionally supports audio. Should you wish to stream video and audio from the web to your TV, this is the port you would want to use.

• SD card port

The main storage mechanism of the Raspberry Pi is via the SD card port. The SD card will be where we install our operating system and will act as our basic hard disk. Of course, this storage can be expanded upon using the USB ports.

• 256 MB/512 MB SDRAM shared with GPU

The Raspberry Pi comes equipped with 256 MB of SDRAM on older versions of the model B and 512 MB on the newer revisions. This isn't a huge amount, and much less than you would expect on a PC, where RAM is available in gigabytes. However, for the type of applications we will be building, 256 MB or 512 MB of RAM will be more than enough.

• Central processor unit (CPU)

Early in this chapter we touched upon ARM – the British manufacturers of central processor unit (CPU) cores. The Raspberry Pi comes equipped with a 700 MHz, ARM1176JZF-S core – part of the ARM 11 32-bit multi-processor core family. The CPU is the main component of the Raspberry Pi, responsible for carrying out the instructions of a computer program via mathematical and logical operations. The Raspberry Pi is in good company using the ARM 11 series and has joined the ranks of the iPhone, Amazon Kindle, and Samsung Galaxy.

• Graphics-processing unit (GPU)

The graphics-processing unit (GPU) is a specialized chip designed to speed up the manipulation of image calculations. In the case of our Raspberry Pi, it comes equipped with a Broadcom Video Core IV capable of hardware accelerated playback and support for OpenGL. This is especially useful if you want to run games or video via your Raspberry Pi, or work on 3D graphics in an open source application such as Blender.

• Ethernet port

The Ethernet port is the Raspberry Pi's main gateway to communicating with other devices and the Internet. You will be able to use the Ethernet port to plug your Raspberry Pi into a home router such as the one you currently use to access the Internet, or a network switch if you have one set up.

• General Purpose Input/output (GPIO) pins

The General Purpose Input/output (GPIO) pins on the Raspberry Pi are the main way of connecting with other electronic boards such as the Arduino. As the name suggests, the GPIO pins can accept input and output commands and thus can be programmed on the Raspberry Pi. With the arrival of the Raspberry Pi , a set of open source technologies now exist that combine the power of the PC, the communication and multimedia technologies of the web, the ability to interact with the environment of a microcontroller, and the portability of a mobile device. This provides the perfect set of factors allowing us to build cheap devices for our homes that can interface with commercial devices, but can be tailored for our own needs while also providing a great tool for learning about technology.

2.1.2 Analog-To-Digital Converter (ADS1115):

The ADS1115 is a 4-channel breakout board as shown in the figure below; it is perfect for adding high resolution analog to digital conversion to the raspberry pi 3. It is also a 4-channel analog to digital converter and it utilizes the I2C protocol with selectable addresses. It has a wide supply ranges from 2V to 5.5V.The current consumption is low 150µA (Continuous Conversion Mode), Operating Temperature Range: –40°C to+125°C. The connection of this ADC with the raspberry pi3 board runs at the Raspbian operating system.

Figure 2.2: Analog to Digital Converter ADS1115

2.1.3 Pulse Sensor:

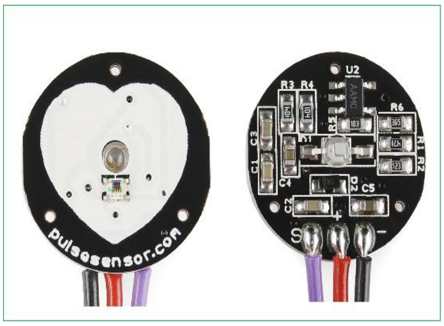

Figure 2.3: pulse sensor

A person's heartbeat is the sound of the valves in his/her's heart contracting or expanding as they force blood from one region to another. The number of times the heart beats per minute (BPM), is the heart beat rate and the beat of the heart that can be felt in any artery that lies close to the skin is the pulse .

Manual way: Heart beat can be checked manually by checking one's pulses at two locations – wrist (the radial pulse) and the neck (carotid pulse). The procedure is to place the two fingers (index and the middle finger) on the wrist (or neck below the windpipe) and count the number of pulses for 30 seconds and then multiplying that number by 2 to get the heart beat rate. Pressure should be applied minimum and also fingers should be moved up and down till the pulse is felt.

Practical way: Designed to give digital output of heart beat when figure is placed on it. When the heart beat detector is working, the beat LED flashes the unison with each heart beat. This digital output can be connected to Raspberry pi to measure the beats per minute rate.

Principle of pulse Sensor:

The sensor consists of super bright red LED and light detector. The LED needs to be super bright as the maximum light must pass spread in figure and detected by detector. Now, when the heart pumps a pulse of blood through the blood vessels the figure becomes slightly more opaque and so less light reached the detector with each heart pulse the detector signal varies. This variation converted to electrical pulse. This signal is amplified and triggered through an amplifier which output +5V logic level signal. The output signal is also indicated by LED which blinks on each heart beat.

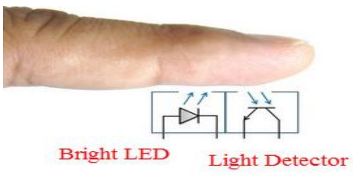

Figure 2.4: sensor principle

Sensor connected regulated DC power supply of 5 volts. Purple wire is ground, next middle wire is Blue which is positive supply and the Green is signal. To test sensor you only need power the sensor by connect two wires +5v and GND can leave the output wire as it is.

When beat LED is off the output is at 0V. Put finger on the marked position and you can view the beat LED blinking on each heartbeat. The output is active high for each beat and can be given to Raspberry pi for interfacing applications.

2.1.4 Temperature (LM35) sensor:

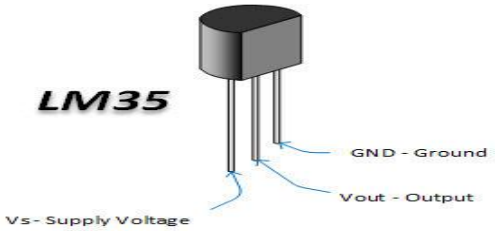

Figure 2.5: temperature lm35 sensors

The LM35 series are precision integrated circuit temperature sensors. Whose output voltage is linearly proportional to the temperature in Celsius (Centigrade). The LM35 sensor thus has an advantage over linear temperature sensors, calibrated in Kelvin, as the use is not required to subtract a large constant voltage from its output to obtain the convenient centigrade scaling. The LM35 sensor does not require any external calibration or trimming to provide typical accuracies of C at room temperature and C over a full -55 to +150C temperature range. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy.

Features of LM35

- Calibrated directly in Celsius (Centigrade)
- Linear +10.0mV/◦C scale factor
- 0.5◦C accuracy guarantee able (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
- Low impedance output, 0.1 W for 1 mA load

2.1.5 Electric Buzzer:

The electric buzzer is an electronic device used to produce a sound. This buzzer is prepared by incorporating a piezo electric vibration plate in a plastic case (resonator). In this project, this buzzer is used to alert the caretaker during the critical condition of the patient and the sound obtained by this buzzer indicates that the patient health is in a risk.

Figure 2.6: buzzer

2.1.6 Web Socket Protocol:

A WebSocket is a standard protocol for two-way data transfer between a client and server. The WebSocket protocol does not run over HTTP, instead it is a separate implementation on top of TCP. A WebSocket connection allows full-duplex communication between a client and server so that either side can push data to the other through an established connection. The reason why WebSockets, along with the related technologies of Server-sent Events (SSE) and WebRTC data channels, are important is that HTTP is not meant for keeping open a connection for the server to frequently push data to a web browser. Previously, most web applications would implement long polling via frequent Asynchronous JavaScript and XML (AJAX) requests as shown in the below diagram.

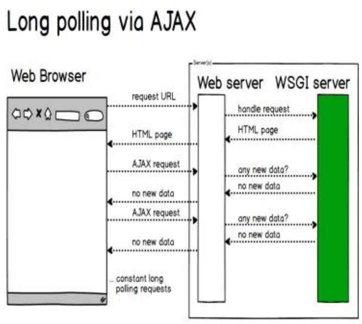

Figure 2.7: Long polling via AJAX

Server push is more efficient and scalable than long polling because the web browser does not have to constantly ask for updates through a stream of AJAX requests.

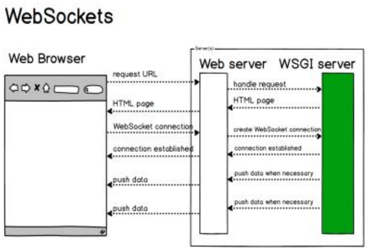

Figure 2.8: Web Socket flow

While the above diagram shows a server pushing data to the client, WebSockets is a full-duplex connection so the client can also push data to the server as shown in the diagram below.

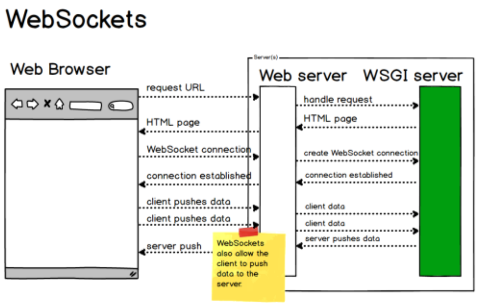

Figure 2.9: Web Socket flow with client push

The WebSockets approach for server- and client-pushed updates works well for certain categories of web applications such as chat room, which is why that's often an example application for a WebSocket library.

Implementing Web Sockets

Both the web browser and the server must implement the WebSockets protocol to establish and maintain the connection. There are important implications for servers since WebSockets connections are long lived, unlike typical HTTP connections.

A multi-threaded or multi-process based server cannot scale appropriately for WebSockets because it is designed to open a connection, handle a request as quickly as possible and then close the connection. An asynchronous server such as Tornado or Green Unicorn monkey patched with gevent is necessary for any practical WebSockets server-side implementation.

On the client side, it is not necessary to use a JavaScript library for WebSockets. Web browsers that implement WebSockets will expose all necessary client-side functionality through the Web Sockets object.

2.1.7 Structured Query Language

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. In MySQL the beginning my of the name comes from the daughter of the Finnish developer [11]. SQL was initially developed to operate on data in databases that follow the relational model. It is a programming language for querying, modifying and managing data. MySQL is the most common open source database tool. It is considered an easy and reliable program compared to other database software. MySQL offers various different programs that are database related. The most famous one is MySQL Enterprise.

Defining databases

Database is a set of tables that can be manipulated in accordance with the relational model of data. It is a collection of integrated records. A record is a representation of some conceptual object that has multiple attributes like name, address and telephone number of a customer. A database consists of both data and metadata. Metadata is data that describes the structure of the data within a database. They store metadata in an area called the data dictionary, which describes the tables, columns, index constraints, and other items that make up the database [1]. Databases come in all sizes, from a simple collection of a few records to millions of records.

2.2 Literature Review

Prathiba et al. [3] proposes a cooperative approach for an automated Healthcare Monitoring System which uses sensors to measure various parameters of the patient like temperature, pulse and body movement using wireless communications. The patient’s status will be recorded and stored on the cloud. This proposed system process the sensor data and provides a real time monitoring information about physiological conditions of a patient to the doctor , so that the Healthcare professionals can monitor and access patient’s data from anywhere of the world at any time. It enables personalization of treatment and management.

R.Kumar and Pallikonda [4] discussed a paper about, monitoring patient's body temperature, respiration rate; heart beat and body movement using Raspberry Pi board. These sensors signals send to the Raspberry pi as well as monitoring through anywhere in the world using internet source.

In [5] the paper provides an overview of the Internet of Things (IoT) with emphasis on enabling technologies, protocols, and application issues. This paper starts by providing a horizontal overview of the IoT. Then, give an overview of some technical details that pertain to the IoT enabling technologies, protocols, and applications. Compared to other survey papers in the field, the objective is to provide a more thorough summary of the most relevant protocols and application issues to enable researchers and application developers to get up to speed quickly on how the different protocols fit together to deliver desired functionalities without having to go through RFCs and the standards specifications. Also provide an overview of some of the key IoT challenges presented in the recent literature and provide a summary of related research work. Finally, present detailed service use-cases to illustrate how the different protocols presented in the paper fit together to deliver desired IoT services.

Breivold and Sandström [6] discuss a paper that describe the specific quality attribute constraints within industrial automation, present specific industrial IoT challenges due to these constraints, and discuss the potentials of utilizing some technical solutions to cope with these challenges. Finally, find the remaining challenges within long lived industrial automation systems with huge complexity and proprietary solutions. By introducing industrial IoT into these systems, connecting different long-lived systems, and creating new services based on fast-paced technologies, additional challenges in integrating these long lived systems and innovative IoT solutions will arise– this has so far been largely unaddressed by the Industrial IoT community, and remains to be a future research direction.

Jin et al. [7] proposed a generic framework for creating IoT implementations, including smart healthcare. According to this article, there are three main viewpoints that guide the building of an IoT implementation: network-centric IoT, cloud-centric IoT and data-centric IoT. Each viewpoint has several building blocks, and each building block in a viewpoint corresponds to one or more blocks in the other viewpoints. Therefore, the three viewpoints have strong relation with each other. All these three viewpoints have influenced the implementation of different IoT applications.

Khairnar et al. [8] designed system to be used in hospitals for measuring and monitoring various parameters like temperature, ECG, heart beat etc. The results can be recorded using Raspberry Pi displayed on a LCD display. Also the results can be sent to server using GSM module. Doctors can login to a website and view those results.

Mangrulkar et al. [9] proposed system aims to cover an end-to-end smart, efficient and innovative health application that can be built up from two functional building blocks. However the main function of the first building block is to gather all sensory data that are related to the monitoring of the patients, whereas the second block functions is to store, process and present the resulted information at the server where the doctors can access health reports following the case of the monitored patients.

Bhaskar and Prabhakar [10] discussed method of patient health monitoring system to monitor patient's body temperature, heart rate using Raspberry Pi3. The temperature sensor (DHT11) senses the temperature from the patient's body and sends the information to the raspberry pi3. The heart Rate sensor collects the heart beat from the patient and sends it to raspberry pi. The output obtained from raspberry pi3 is displayed at the HDMI display which is again sent to ubidots through Wi-Fi. After connecting internet to the Raspberry Pi 3 it acts as a server. Then the server is automatically sends data to the website. Using IP address anybody can monitor the patient's health status anywhere in the world using laptops, tablets and smart phones.

Dighem et al. [11] this system presents a monitoring system that has the capability to monitor physiological parameters from patient body at every 10 seconds. A sensor node has attached on patient body to collect all the signals from the wireless sensors and sends them to the BSN care node. The attached sensors on patient's body form a wireless body sensor network (WBSN) and they are able to sense the heart rate, Temperature of surrounding. This system is mainly to detect abnormal conditions in human body and abnormal physiological parameters. The main advantage of this system in comparison to previous systems is to reduce the energy consumption to prolong the network lifetime, speed up and extend the communication coverage to increase patient quality of life.

Al shaikh[12] presents a paper based on monitoring of remote patients, after he is discharged from hospital. The system designed and developed a reliable, energy efficient remote patient monitoring system. It is able to send parameters of patient in real time. It enables the doctors to monitor patient's parameters (temp, heartbeat, ECG) in real time. Here the parameters of patient are measured continuously (temp, heartbeat, ECG) and wirelessly transmitted using Zigbee.

Chapter Three

SYSTEM DESIGN and IMLPEMENTATION

3.1 System design and implementation

The project revolves around developing a remote patient monitoring system, which allows doctors to monitor patient through a web application. Currently it is focused on monitor hear rate and temperature of the patient, alarm when the system detects abnormal values. In this chapter discusses the basic design of the measuring sensor. The complete system design will be highlighted.

3.2 The structure of the system

The structure of the system consists of web application connects to cloud server through the internet to Raspberry Pi. The Raspberry Pi interacts with server to perform controlling and monitoring function by software program written in python programming language. Pulse and temperature sensors are connected to raspberry pi via analog to digital convertor ADS1115. Figure 3.1 below describes the structure of the system.

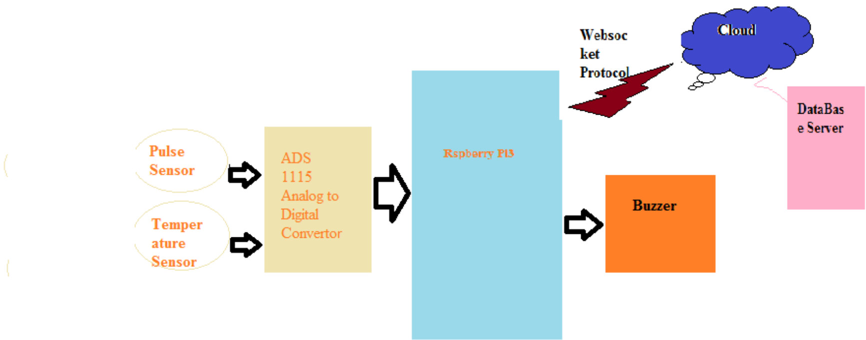

Figure 3.1: the structure of the system

3.2.1 ADS1115 with raspberry pi interface

The ADS1115 are great analog to digital converters that are easy to use with the Raspberry Pi using its I2C communication bus, which is a higher precision 16-bit ADC with 4 channels. The 16-bit signed integer value can represent decimal number from 0 to 65536.

First step is to install the Adafruit ADS1x15 Python library from the source on Github. The library is installed by running the following commands in the terminal:

Figure 3.2 ADS1115 commands

To use the ADS1115 in Python program, import the Adafruit_ADS1x15 and create an ADS1115 instance

adc = Adafruit_ADS1x15.ADS1115().

Then data can be read from the selected channel x,

data = adc.read_adc(x, gain=GAIN)

Where x is an integer from 0 to 4.

In this system, the output represents the pulse per minutes is read from A0, and the temperature is read from A1.

3.2.2 Temperature measurement

Now for reading the temperature of the patient, LM35 temperature sensor is used in this design. Temperature is usually measured in "Centigrade" or "Fahrenheit". "LM35" sensor provides output in degree Centigrade.

As mentioned in the previous chapter, LM35 has three pins:

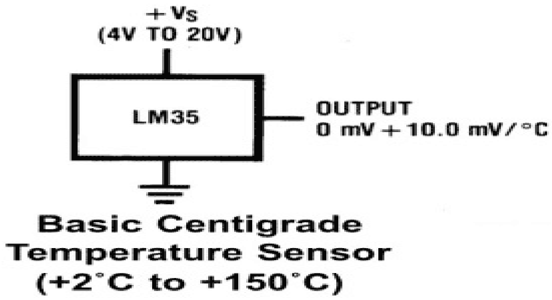

Figure 3.3 lm35 sensor output pins

Pin1= VCC power (connected to +5V from the raspberry pi GPIO)

Pin2= signal or output (connected to A1 from the ADS1115)

Pin3= ground (connected to the ground from the raspberry pi GPIO)

This sensor provides variable voltage at the output. For every +1 centigrade rise in the temperature there will be +10mv higher voltage at the output pin. So if the temperature is 0◦ centigrade the output of the sensor will be 0v, if the temperature is 10◦ centigrade the output of the sensor will be 100mv.

Reverse the equation get the following:

$$100 * x\,volts = y°C \qquad (3\text{-}1)$$

3.2.3 Heart rate measurement:

The Pulse Sensor is a plug-and-play heart-rate sensor. It essentially combines a simple optical heart rate sensor with amplification and noise cancellation circuitry making it fast and easy to get reliable pulse readings. The sensor unit consists of an infrared light emitting diode (IR LED) and a photodiode, placed side by side. The IR diode transmits an infrared light into the finger (placed over the sensor unit), and the photo diode senses the portion of the light that is reflected back. The intensity of the reflected light depends upon the blood volume inside the fingertip. So each heart beat slightly alters the amount of reflected infrared light that can be detected by photo diode. With a proper signal conditioning, this little change of the amplitude of the reflected light can be converted into a pulse. The pulses can be later counted by the raspberry pi to determine the heart rate. The signal conditioning circuit consists of two identical active low pass filters. The operational amplifier used in this circuit is MCP602, a dual opAmp chip. It operates at a single power supply and provides rail-to-rail output swing. The filtering is necessary to block any higher frequency noises presents in the signal.

In this circuit the device calculates the time between two pulses within a period of maximum 2 seconds and calculates the pulses per seconds where inverted time and hit that rate in the 60 to give the rate per unit time in Minutes (described in appendices B).

Pulse sensor has three pins that connect as shown:

Pin1= ground (connected to the ground from raspberry pi GPIO)

Pin2= VCC (connected to +5v from raspberry pi GPIO)

Pin3= signal (connected to ADS1115 from A0)

3.3 Software Design

Software design for develop an IOT health system will be discussed in this section. The Raspberry Pi is programmed using python programming language. The following figures show the flow charts for functions that used to control, and to monitor process.

3.3.1 Raspberry pi

In order to use Raspberry Pi device it's required to start by installing an operating system onto an SD card. The Raspberry Pi operates on a LUNIX based open source operating system called Raspbian OS. This allows more control and flexibility in the software therefore making it easy to program the Pi. The Raspberry Pi communicates with the attached devices and sensors through python codes with addition library to control their functions. The Raspbian operating system was installed onto Raspberry Pi, which was obtained by downloading Raspbian onto the SD card from the manufacturer's website [13].

The Raspberry Pi Model B+ board contains a single 40-pin expansion header labeled as 'J8' providing access to 26 GPIO pins.

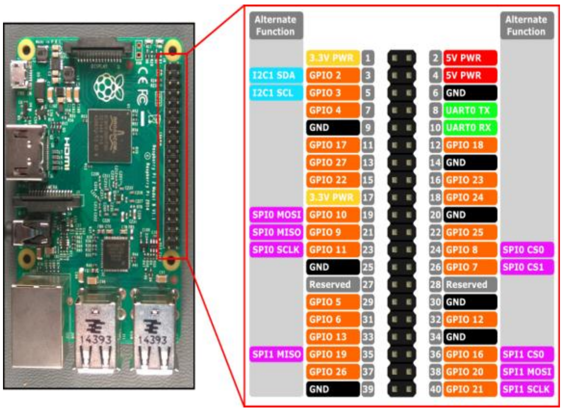

Figure 3.4 Raspberry Pi Model B+ GPIO

After interfacing bio sensors with ads1115 and raspberry pi, the raspberry pi will connect to the internet so raspberry pi can access the server to perform the following algorithm

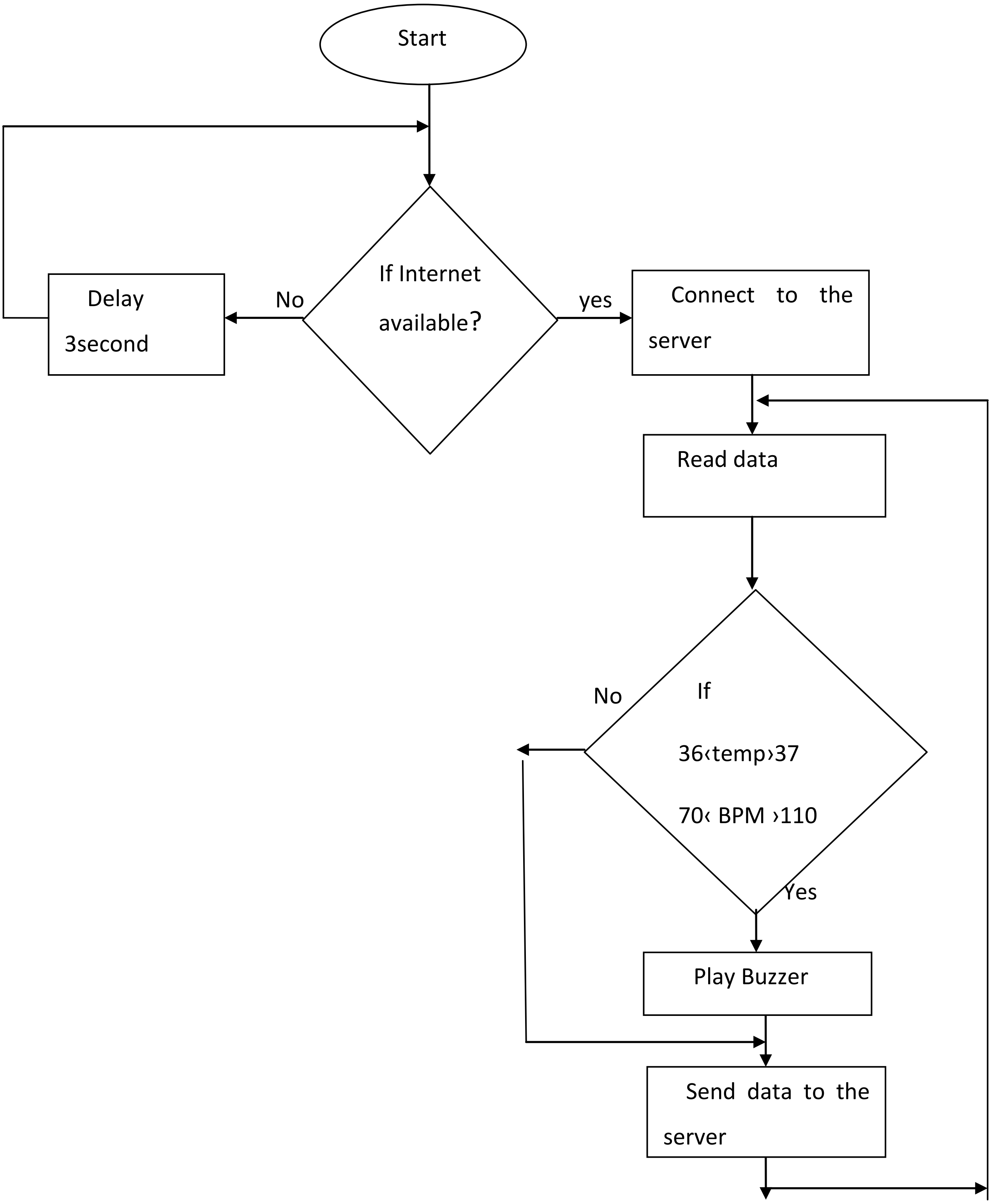

Figure 3.5 raspberry pi flow chart

3.3.2 Flask development server

Flask is a micro web framework written in python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for form validation; upload handling, various open authentication technologies and several common framework related tools.

Because flask has no data base layer, we need to install MySQL data base in order to store the patient's parameters.

This flask server uses web socket protocol to enable communication between raspberry pi (client1) and web browser (client2). After raspberry pi read and process patient parameters from the sensors, then these parameters is send to the flask server.

The following flowchart describes the working algorithm of the flask server.

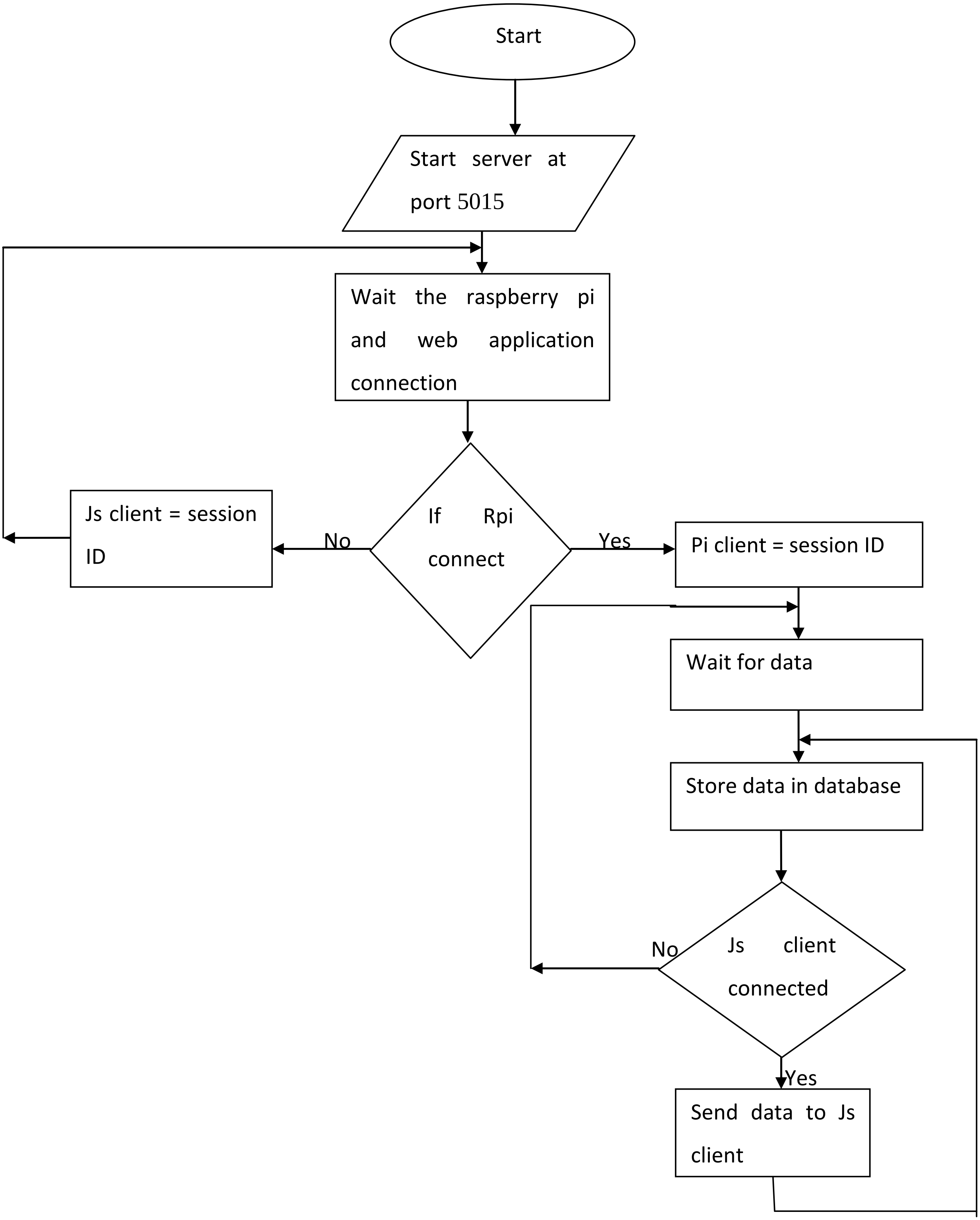

Figure 3.6 Flask server flow chart

3.3.3 Web application

The IOT health web page is designed using html and java script programming language.

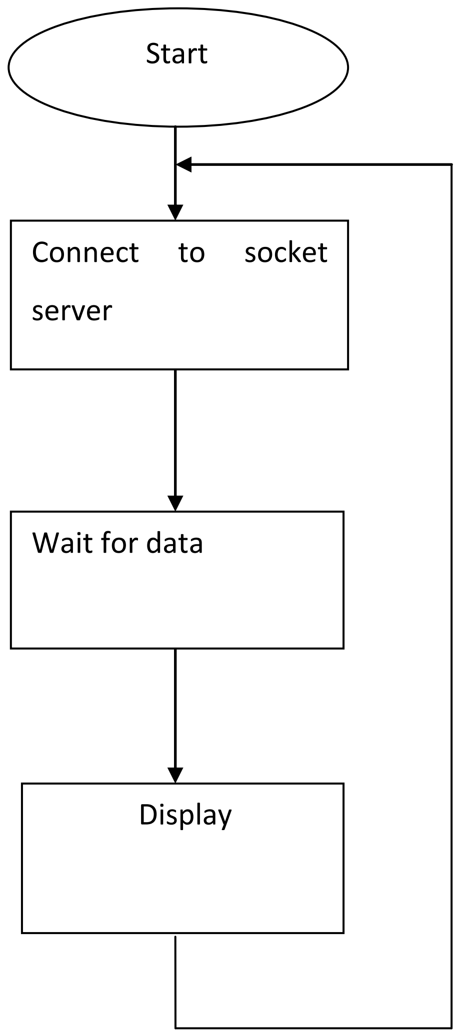

Figure 3.7Web Application flow chart

3.4 System Integration and Testing

The hardware and software design needed to be integrated as a complete system. Testing of the complete system is needed to ensure that system integration is functioning as required. Each part of the development of hardware and software designs should be tested before combining all parts into a complete system. Each part must pass the testing process to make sure that the system on that part is functioning correctly before moving to another part. If the part has errors, troubleshooting should be made to correct the system.

Chapter Four

RESULTS AND DISCUSSIO

4 Result and Discussion

The proposed system aims to cover an end-to-end efficient health application that can be implemented from two functional modules. The main function of the first module is to gather all the sensory data of the patient, whereas the second block functions is to store, process and display the resulted readings at the sever where the doctor can access health report following the case of the monitored patient.

4.1 Sensing the patients Physiological parameters

The objective of this module is to sense the patient's temperature and pulse. These two sensors are connected to the raspberry pi using ADS1115 analog to digital convertor. Raspberry pi is used to process the data gathered from the sensors and send this data to the server.

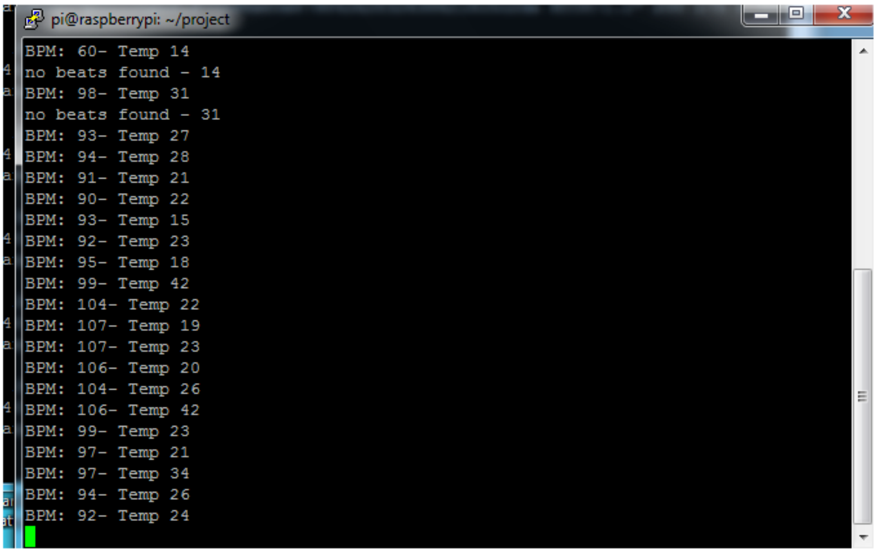

Figure 4.1 the sensor's data read by Raspberry pi

4.1.1 IoT health Web application

When the raspberry pi is connected to the internet, the sensor's signals will be shown in the website at real time.

The IOT health monitoring web page was developed using HTML, Java script and bootstrap.

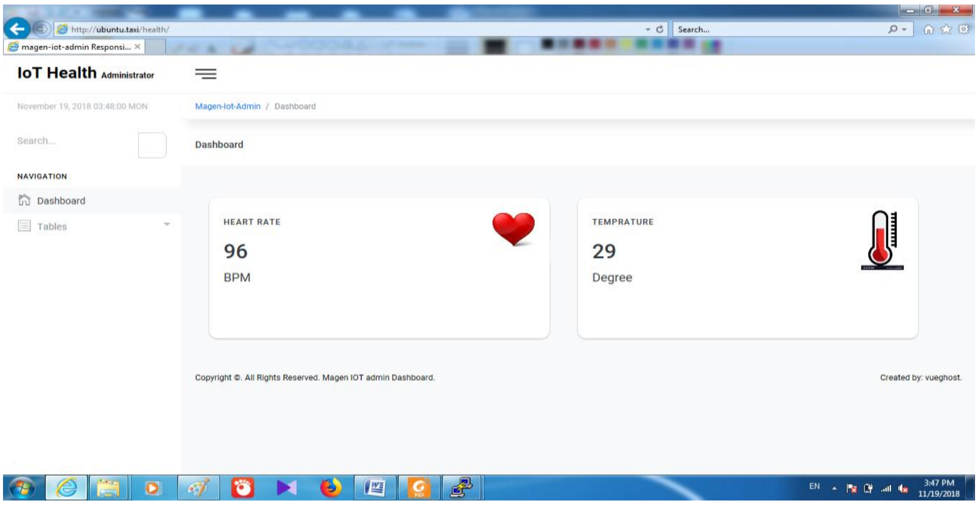

Figure 4.2 IOT health webpage

4.1.2 Cloud Data Storage

The patient's status will be recorded and stored on the cloud using MySQL. MySQL database is a client/server system consists of a multi-threaded SQL server that supports different background. If the doctor needed to revise or analyze the history of the patient for better diagnosis he can easily access the database tables in webpage so the Healthcare professionals can monitor and access patient's data from anywhere of the world at any time.

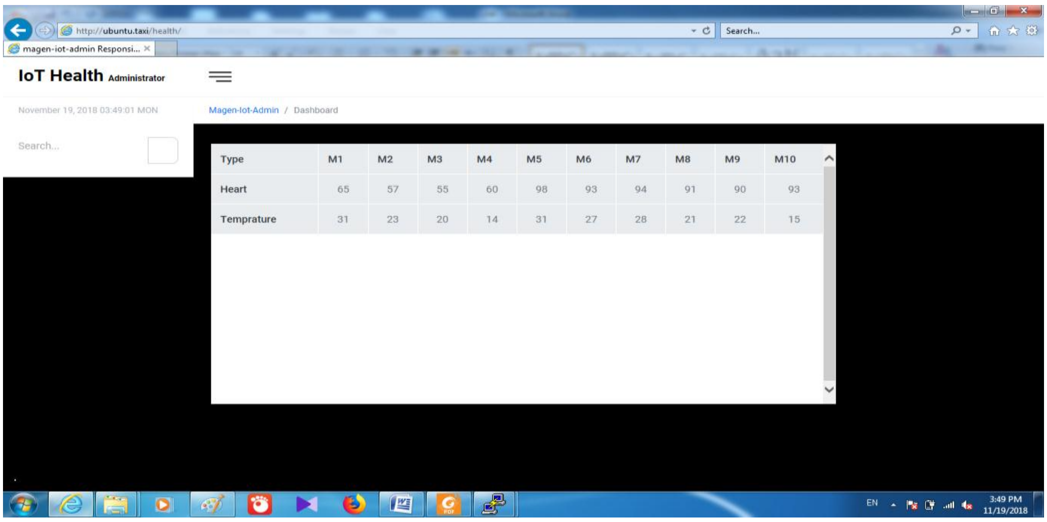

Figure 4.3 patient's history stored in the web application

4.2 Discussions

This system has been compared to the people's education hospital system by taking ten readings from both systems for the same patient at the same time. Mean value and the standard deviation have been calculated as shown below.

4.2.1 Comparison Result of the heart rate:

Table 4-1 Heart rate result comparison

Readings No.	Readings from designed System	Readings from reference system
1	80	70
2	79	76
3	80	80
4	83	92
5	90	85
6	88	81
7	86	82
8	87	87
9	75	87
10	81	87

The formula of stander deviation (SD) is

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (4\text{-}1)$$

Where:

$\sum$ = sum of

x = value in the data set

μ = the mean of the data set

N = the number of data points

Table 4.2 Mean and Stander Deviation

	Designed System	**Reference System**
Mean	82.9	82.7
Stander Deviation	4.49	6.04

$Relative\ Error = measured\ value - \ expected\ value /\ expected\ value\ \%$ (4-2)

Calculating the error according to the equation above, the result is:

Relative error = -0.0025%

Chapter Five

CONCLUSION AND RECOMMENDATIONS

5 Conclusion and future work

5.1 Conclusion

From the above design IoT based healthcare system using raspberry pi is able to transmit the data which is sensed from the patient to the doctor's PC using Inter of Things. The proposed system is able to monitor the body temperature and heart rate with enough accuracy [relative error equal to -0.0025%] so reduce the cost.

This system insures that the patient receives medical attention in the opportunely before it is too late. The different physiological variables to be monitored remotely are sensed and converted from analog to digital form then stored and displayed online in the web application page.

5.2 Recommendations and Future Work

There is always a chance to improve any system as research to development in an endless process. Based on threats assessments a set of security requirements has been identified and recommend have been suggested for the overall patient monitoring systems.

Flexibility of this system allow additional of sensors to measure other vital signs such as blood pressure and respiration.

References

[1]Laxmi Bhaskar, 2 Prof. Prabhakar Manage "IOT based Patient Health Monitoring System using Raspberry pi 3" International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 07 | July -2017.

[2] Dae-Hyeok Mun, Minh Le Dinh, and Young-Woo Kwon. An Assessment of Internetof Šings Protocols for Resource-Constrained Applications.In IEEE COMPSAC, 2016.

[3] Chengathir Selvi. M, T.D.Rajeeve, A.John Paul Antony and Prathiba. T, "Wireless Sensor Based Healthcare Monitoring System Using Cloud," International Conference on Inventive Systems and Control (ICISC-2017), IEEE Journal, 2017.

[4] R.Kumar, and Dr.M.Pallikonda Rajasekaran, "AN IOT BASED PATIENT MONITORING SYSTEM USING RASPBERRY PI," International Conference on Current Research in Engineering Science and Technology (ICCREST-2016), IEEE Journal, 2016.

[5] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari and Moussa Ayyash. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, Fourth Quarter 2015.

[6] Hongyu Pei Breivold, and Kristian Sandström. "Internet of Things for Industrial Automation – Challenges and Technical Solutions". IEEE International Conference on Data Science and Data Intensive Systems, 2015.

[7] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things,"

IEEE Internet of Things Journal, vol. 1, no. 2, pp. 112–121, 2014.

[8] Pooja Navdeti, Sumita Parte, Prachi Talashilkar, Jagruti Patil, Dr. Vaishali Khairnar, "Patient Parameter Monitoring System using Raspberry Pi," International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume – 5 Issue -03 March, 2016.

[9] Durga Amarnath M. Budida, Dr. R. S. Mangrulkar, "A Review Paper on Design and Implementation of Smart Health Care System using IoT," International Conference on Emanations in Modern Engineering Science and Management (ICEMESM-2017), Volume: 5 Issue: 3 March, 2017.

[10] Laxmi Bhaskar, 2 Prof. Prabhakar Manage "IOT based Patient Health Monitoring System using Raspberry pi 3" International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 07 | July -2017.

[11] Sangle Sagar D, Deshpande Niranjan R, Vadane Pandurang M, Dighe M. S, "IoT Based Health-Care System Using Raspberry Pi," International Research Journal of Engineering and Technology (IRJET),Volume: 04 Issue: 04 April, 2017.

[12] Rubina.a.shaikh, "Real time health monitoring system of remote patient using ARM7," International Journal of Instrumentation, Control and Automation (IJICA) ISSN: 2231-1890, Vol-1 Iss-3,4, 2012.

[13] https://www.raspberrypi.org/ (18 August 2018 at 5:20:13)

Appendix A

LM35 Temperature Sensors

A.1 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 ±1⁄4˚C at room temperature and ±3⁄4˚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.

Typical Applications

Figure A-1: Basic Centigrade Temperature Sensor

(+2˚C to +150˚C)

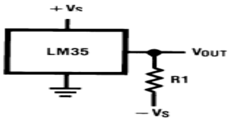

Figure A-2: Full-Range Centigrade Temperature Sensor

Choose R1 = -VS/50 µA

V OUT=+1,500 mV at +150˚C

= +250 mV at +25˚C

= -550 mV at -55˚C

Appendix B

Analog-to-Digital Converter ADS1115

B.1 Description

The ADS1113, ADS1114, and ADS1115 are precision analog to digital converter (ADC) with 16 bits of resolution offered in an ultra small, leadless QNF-10 package. The ADS1113/4/5 are designed with precision power and ease of implementation. The ADS1113/4/5 features an onboard reference and oscillator. Data are transferred via I2C compatible serial interface. The ADS1113/4/5 operates from a single power supply ranging from 2.0v to 5.5v.

The ADS111/3/4/5 can perform conversions at rate 860 samples per seconds. An onboard PGA is available on the ADS1114/5 that offers input ranges from supply to as low as ±256mv, allowing both ranges and small signals to be measured with high resolution. The ADS1115 also features an input multiplexer that provides four single ended inputs. The ADS1115 operate either in continuous conversion mode or a single shot mode that automatically powers down after conversion and greatly reduce current consumption during idle periods.

B.2 Applications

- Portable instrumentation
- Consumer goods
- Battery monitoring
- Temperature measurement
- Factory automation and process control

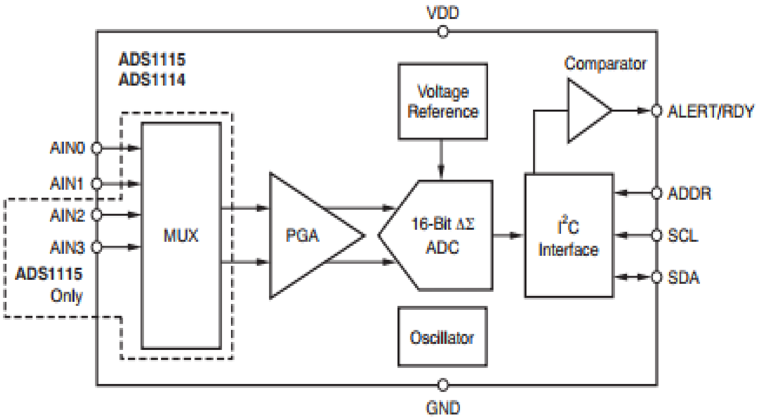

Figure B-1: ADS1115 block diagram

Appendix C

Programming Codes

C.1 Server code:

```
from gevent import monkey

monkey.patch_all()

from flask import *

from flask_socketio import SocketIO,send, emit

from flask_cors  import CORS

import MySQLdb

from time import *

import json

app = Flask(__name__)

@app.after_request

def after_request(response):

    response.headers.add('Access-Control-Allow-Origin', '*')

    response.headers.add('Access-Control-Allow-Headers', 'Content-
Type,Authorization,X-Auth')


    response.headers.add('Access-Control-Allow-Methods',
'GET,PUT,POST,DELETE')

    return response


socketio = SocketIO(app)

global jsClient
```

```
jsClient=''
global piClient
piClient=''
global Time
Time=''
global started
started=True
global start
start=1
global rpiConnected
rpiConnected=False
def new_measure():
    db = MySQLdb.connect("localhost","root","Abdullah1225","test" )
    cursor = db.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("insert into history(t_name) values('a')")
    db.commit()
    cursor.execute("SELECT time FROM history ORDER BY id DESC LIMIT 1")
    global Time
    Time=cursor.fetchone()['time']
    cursor.execute("create table `{}`(id int primary key auto_increment,value int,tmp int)".format(Time))
    db.commit()
def insert_value(value,tmp):
    db = MySQLdb.connect("localhost","root","Abdullah1225","test" )
```

```
        cursor = db.cursor(MySQLdb.cursors.DictCursor)
        global Time
        cursor.execute("insert into `{}`(value,tmp)
values({},{})".format(Time,value,tmp))
        db.commit()
    @socketio.on('connect')
    def test_connect():
        print 'Client connected '+request.sid
    @socketio.on('jsClientConnect')
    def test_connect(data):
        print 'JS Client connected '+request.sid
        global jsClient
        jsClient=request.sid
    @socketio.on('rpiConnect')
    def test_connect(data):
        print 'RPi Client connected '+request.sid
        global rpiConnected
        global piClient
        global start
        global started
        started=True
        start=1
        rpiConnected=True
        piClient=request.sid
```

```
@socketio.on('checkPi')
def test_connect(data):
    global rpiConnected
    global jsClient
    emit('piStatus',{'connected':rpiConnected},room=jsClient)
@socketio.on('reading')
def test_connect(data):
    global started
    global start
    if started:
        new_measure()
        started=False
    print 'temp',data['temp'],'heart',data['heart']
    global jsClient
    if jsClient:
        emit('reading',data,room=jsClient)
    if start<=10:
        insert_value(data['heart'],data['temp'])
    start+=1
@app.route('/getHistory', methods=['GET'])
def getHistory():
    db = MySQLdb.connect("localhost","root","Abdullah1225","test" )
    cursor = db.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("SELECT * FROM history")
```

```
    Times=cursor.fetchall()
    for res in Times:
        res['time']=res['time'].isoformat()
    print Times
    return json.dumps(Times)
@app.route('/getReading', methods=['POST'])
def getReading():
    db = MySQLdb.connect("localhost","root","Abdullah1225","test" )
    cursor = db.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("SELECT * FROM
`{}`".format(request.json['time'].replace("T"," ")))
    Times=cursor.fetchall()


    print Times
    return json.dumps(Times)
@socketio.on('disconnect')
def test_disconnect():
    print 'Client disconnected '+request.sid
    global rpiConnected
    global piClient
    if piClient==request.sid:
        rpiConnected=False
if __name__=='__main__':
    socketio.run(app,host='0.0.0.0',port=5015,debug=True)
```