SMART PATIENT HEALTH MONITORING SYSTEM USING IOT

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in

Computer Science and Engineering

By

SUMEET DHALI (Roll no-17CS0445)

SURAJ PRAKASH (Roll no-17CS8094)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited WITH Grade "A" by NAAC

JEPPIAR NAGAR,RAJIV GANDHI SALAI,CHENNAI-600119

MARCH 2021

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 www.sathyabama.ac.in

---

DEPARTMENT OF COMPUTER SCIENCE AND ENGINNERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SUMEET DHALI (Roll no-17CS0445) and SURAJ PRAKASH (Roll no-17CS8094)** who carried out the project entitled "**SMART PATIENT HEALTH MONITORING SYSTEM USING IOT**".under our supervision from Jan 2021 to April 2021.

Internal Guide

Dr. T. JUDGI,M.E.,Ph.D

Head of the Department

Dr.L.LAKSHMANAN ,M.E.,Ph.D

Dr.S.VIGNESHWARI,M.E.,Ph.D.

Submitted For Viva Voice Examination held on ____________________

Internal Examiner **External Examiner**

DECLARATION

I **SUMEET DHALI (Roll no-17CS0445) and SURAJ PRAKASH (Roll no-17CS8094)** hereby Declare that the project Report entitled "**SMART PATIENT HEALTH MONITORING SYSTEM USING IOT**" done by under the guidance of **Dr.T.JUDGI, M.E,,Ph.D** is submitted in partial fulfillment of the requirement for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.SASIKALA ,M.E.,Ph.D.,Dean,School of Computing** and **Dr.S.VIGNESHWARI ,M.E.,Ph.D. Head of the Department,Dept. of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.T. JUDGI. M.E.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTRACT

The fundamental element of people’s needs is health .Humans face a haul of surprising death and plenty of diseases because of varied diseases that are a result of lack of treatment to the patients at right time. The main objective of this project is to develop a reliable sensible patient health observance system victimization IoT so the attention professionals will monitor their patients. The sensors will be either worn or be embedded into the body of the patients, to unendingly monitor their health. the knowledge collected in such a fashion will behold on, analyzed, and well-mined to try and do the first prediction of diseases. A mobile device-based attention observance system is developed which may offer period on-line data regarding physiological conditions of a patient primarily consists of sensors, the information acquisition unit, Arduino, and programmed with code. The patient’s temperature, heartbeat rate, pressure level, graph knowledge square measure monitored, displayed, and hold on by the system and sent to the doctor’s and patient's mobile containing the appliance. The sensible Health observance System monitors health standing and saves it on the online page.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Internet of things (IoT) describes the network of physical objects “things” that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. There are a number of serious concerns about dangers in the growth of IoT, especially in the areas of privacy and security, and consequently industry and governmental moves to address these concerns have begun including the development of international standards. The Internet of Medical Things (IoMT) is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. The IoMT has been referenced as "Smart Healthcare" as the technology for creating a digitized healthcare system, connecting available medical resources and healthcare services.

The IOT is generally considered as connecting objects to the Internet and using that connection for control of those objects or remote monitoring. But this definition was referred only to part of IOT evolution considering the machine to machine market today. But actual definition of IOT is creating a brilliant, invisible network which can be sensed, controlled and programmed. The products developed based on IOT include embedded technology which allows them to exchange information, with each other or the Internet and it is assessed that about 8 to 50 billion devices will be connected by 2020. Since these devices come online, they provide better life style, create safer and more engaged communities and revolutionized healthcare. The entire concept of IOT stands on sensors, gateway and wireless network which enable users to communicate and access the application/information.

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well.These sensors create a network of intelligent sensors that are able to collect, process, transfer, and analyze valuable information in different environments, such as connecting in-home monitoring devices to hospital-based systems. Other consumer devices to encourage healthy living, such as connected scales or wearable heart monitors, are also a possibility with the IoT.

The application of the IoT in healthcare plays a fundamental role in managing chronic diseases and in disease prevention and control. Remote monitoring is made possible through the connection of powerful wireless solutions. The connectivity enables health practitioners to capture patient's data and applying complex algorithms in health data analysis.

Health is a fundamental element of people's need for a better life. Unfortunately, the global health problem has created a dilemma because of certain factors, such as poor health services, the presence of large gaps between rural and urban areas, physicians, and nurses unavailability during the hardest timeThe Healthcare industry remains among the fastest to adopt the Internet of Things. The reason for this trend is that integrating IoT features into medical devices greatly improves the quality and effectiveness of service, bringing especially high value for the elderly, patients with chronic conditions, and those requiring constant supervision. According to some estimates, spending on the Healthcare IoT solutions will reach a staggering $1 trillion by 2025 and, hopefully, will set the stage for highly personalized, accessible, and on-time Healthcare services for everyone. Networked sensors, either worn on the body or embedded in our living environments, make possible the gathering of rich information indicative of our physical and mental health. Captured on a continual basis, aggregated, and effectively mined, such information can bring about a positive transformative change in the health care landscape. The IoT is used by clinical care to monitor physiological statuses of patients through sensors by collecting and analyzing their information and then sending analyzed patient's data remotely to processing centers to make suitable actions. Not only for patients, it also useful for normal people to check the health status by using wearable devices with sensors.

Health has prime importance in everyone's life. currently, attention and eudaimonia management is one every of the foremost promising applications of knowledge technology. Among the applications that the Internet of Things (IoT) beyond any doubt reworking the attention trade, In general, IoT has been wide accustomed to interconnect advanced medical resources and to supply sensible and effective attention services to the individuals. In recent years the Internet of Things(IoT) plays a key role in the healthcare industry. The world population is increasing continuously. In many parts of our country, people are not getting medical facilities at right time. Due to covid 19, many people unable to go to the hospital because of that people are unable to do their routine check-ups for their blood pressure and body temperature. Also, many peoples are not going because of a lot of lengthy processes or because of not availability of doctors. So, we are doing this project for reducing time consumption. In recent years the application of IoT in healthcare is increasing. so, having a smart patient health monitoring system is observed that it will reduce time, cost, and reduce efficiency. People can easily monitor themselves and can get the report at the same time. Because of that, it is easy for the early prediction of diseases. The body temperature, heartbeat rate, blood pressure are the main factors or parameters to diagnose the disease. This project gives temperature, pulse rate, and Ecg data.

1.1 Background

The Future Internet goal is to provide an infrastructure to have an immediate access to information about the physical world and its objects. Physical objects can be applicable to different application domains, such as e-health, warehouse management, etc. Each application domain may have different types of physical devices. Each physical device can have its own specifications, which is required to use in order to interact with it. To achieve the future Internet goal, a layered vision is required that can facilitate data access. Internet of Things (IoT) is a vision that aims to integrate the virtual world of information to the real world of devices through a layered architecture.

The term „Internet of Things" consists of two words, namely *Internet* and *Things*. *Internet* refers to the global network infrastructure with scalable, configurable capabilities based on interoperable and standard communication protocols. *Things* are physical objects or devices, or virtual objects, devices or information, which have identities, physical attributes and virtual personalities, and use intelligent interfaces . For instance, a virtual object can represent an abstract unit of sensor nodes that contains metadata to identify and discover its corresponding sensor nodes. Therefore, IoT refers to the *things* that can provide information from the physical environment through the Internet.

Middleware is as an interface between the hardware layer and the application layer, which is responsible for interacting with devices and information management . The role of a middleware is to present a unified programming model to interact with devices. A middleware is in charge of masking the heterogeneity and distribution problems that we face when interacting with devices .

1.2 Motivation

IoT-based system is in charge of providing knowledge from an environment to an non-expert user. IoT-based system can be used in different environments, so it needs to be able to address many heterogeneous devices. Thus, a major concern within developing an IoT-based system is how to handle the interaction with the heterogeneous devices for non-expert users. This concern can be addressed by a middleware layer between devices and non-expert users. This layer is responsible to hide the diversity of devices from the user perspective, and provides access transparency to the devices for the end users.
The idea of creating abstractions of devices been addressed in the literature. The middleware we found in the literature can provide satisfaction by facilitating the interaction with devices, but they do not support low-level device configuration .
Modern health care system introduces new technologies like wearable devices or cloud of things. It provides flexibility in terms of recording patients monitored data and send it remotely via IOT. In storage stage, data is stored, updated for future use. In data retrieving stage, retrieve data from cloud.

1.3. IoT definition

In this section, we explain some of the IoT definitions. Also, we explain the layered architecture for IoT.

Internet of Things (IoT) has increasingly gained attention in industry to interact with different types of devices. IoT can have influence on industry and society by integrating physical devices into information networks . IoT impacts can be on different perspectives, namely for private and business users. From the perspective of a private user, IoT has effect on both working and personal fields, such as smart homes and offices, e-health and assisted living. From the aspect of a business user the impacts would be in fields such as automation and industrial manufacturing, logistics, business process management, intelligent transportation of people and goods.

IoT integrates physical things into information networks. IoT covers the overall infrastructure, including software, hardware and services, which is used to support these information networks. The integrated physical things can exchange data about the physical properties and information that they sense in their environment. To identify devices, we can use identification technologies like for example RFID, which allow each device be uniquely identified

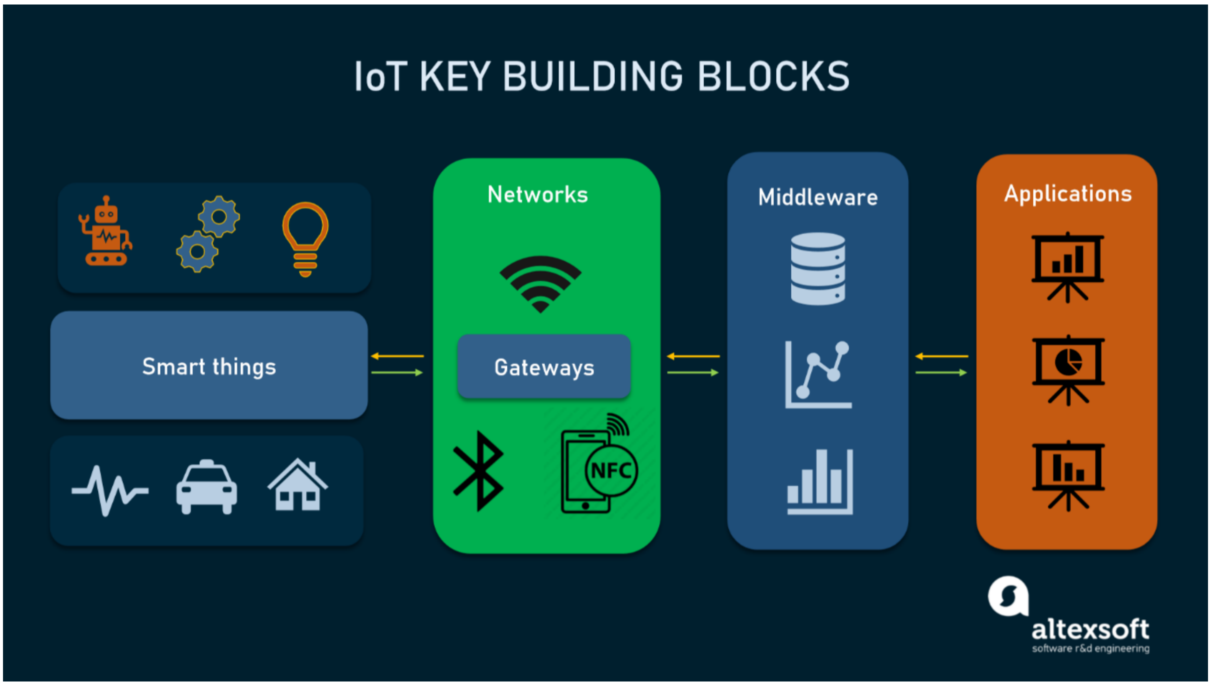

FIG 1.3 IOT ARCHITECTURE

International Telecommunication Union (ITU) defines IoT as "*A global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies*".

IoT has a layered architecture designed to answer the demands of various industries, enterprises and society. It shows a generic layered architecture for IoT that consist of five layers, which are discussed, in the following:

- Edge Technology layer

This is a hardware layer that consists of embedded systems, RFID tags, sensor networks and all of the other sensors in different forms. This hardware layer can perform several functions, such as collecting information from a system or an environment, processing information and supporting communication.

- Access Gateway layer

This layer is concerned with data handling, and is responsible for publishing and subscribing the services that are provided by the *Things*, message routing, and hovelling the communication between platforms.

- Middleware layer

This layer has some critical functionalities, such as aggregating and filtering the received data from the hardware devices, performing information discovery and providing access control to the devices for applications.

- Application layer

This layer is responsible for delivering various application services. These services are provided through the middleware layer to different applications and users in IoT-based

CHAPTER 2
AIM AND SCOPE

2.1 AIM OF PROJECT

The primary goal of this project is to develop a smart patient health monitoring system in such a way that we can get all the necessary and detailed information of the disease. The proposed system measures the body temperature, pulse rate and ECG data. Health is always a major concern in every growth the human race is advancing in terms of technology. Like the recent corona virus attack that has ruined the economy of China to an extent is an example how health care has become of major importance. In such areas where the epidemic is spread, it is always a better idea to monitor these patients using remote health monitoring technology. Remote Patient Monitoring arrangement empowers observation of patients outside of customary clinical settings (e.g. at home), which expands access to human services offices at bring down expenses. The fundamental element of people's needs is health .Humans face a haul of surprising death and plenty of diseases because of varied diseases that are a result of lack of treatment to the patients at right time. The main objective of this project is to develop a reliable sensible patient health observance system victimization IoT so the attention professionals will monitor their patients. The sensors will be either worn or be embedded into the body of the patients, to unendingly monitor their health. the knowledge collected in such a fashion will behold on, analyzed, and well-mined to try and do the first prediction of diseases. The concept of Internet of things is recent and is defined as the integration of all devices that connect to the network, which can be managed from the web and in turn provide information in real time, to allow interaction with people they use it5 . Another concept of IoT "is the general idea of things, especially everyday objects, which are readable, recognizable, locatable, addressable and controllable via the Internet - either through RFID, wireless LAN, wide area network, or by other means "6 . IoT The term itself was first mentioned by Kevin Ashton in 1998 and aims at the exchange of information7 . On the other hand8 , the Internet of things can be seen from three paradigms, which are Internet-oriented middleware, things sensors oriented and knowledge-oriented semantics. Therefore, it is appropriate, such delimitation because the interdisciplinary nature of the subject. However the usefulness of the IoT is reflected when crossing between the three paradigms in the development of applications9 . The Internet of Things has a number challenges that are still working. IoT driven Fog Computing is developed in the healthcare industry that can expedite facilities and services among the mass population and help in saving billions of lives. The new computing platform, founded as fog computing paradigm may help to ease latency while transmitting and communicating signals with remote servers, which can accelerate medical services in spatial-temporal dimensions. The latency reduction is one of the necessary features of computing platforms which can enable completing the healthcare operations, especially in large-size medical projects and in relation to providing sensitive and intensive services. Reducing the cost of delivering data to the cloud is one of the research objectives.

2.1.1 **OBJECTIVES**

- To develop a reliable patient health monitoring system.
- To measure the body temperature, heartbeat rate and ecg.
- To design a system to store patient data.
- To do analysis of collected data of sensors.

- To get health related information in understandable format.

2.2 SCOPE

This project will help in monitoring the patient's health which will be helpful for doctors and patients both. It will help in reducing and early prediction of disease. The core objective of this project is the design and implementation of a smart patient health tracking system that uses Sensors to track patient health and uses internet to inform their loved ones in case of any issues. The objective of developing monitoring systems is to reduce health care costs by reducing physician office visits, hospitalizations, and diagnostic testing procedure Each of our bodies utilizes temperature and also pulse acknowledging to peruse understanding wellbeing. The sensors are linked to a microcontroller to track the status which is thus interfaced to a LCD screen and additionally remote association with have the capacity to exchange alarms. If framework finds any sudden changes in understanding heart beat or body temperature, the framework consequently alarms the client about the patients status over IOT and furthermore indicates subtle elements of pulse and temperature of patient live in the web. In this manner IOT set up tolerant wellbeing following framework viably utilizes web to screen quiet wellbeing measurements and spare persists time. The increased use of mobile technologies and smart devices in the area of health has caused great impact on the world. Health experts are increasingly taking advantage of the benefits these technologies bring, thus generating a significant improvement in health care in clinical settings and out of them. Likewise, countless ordinary users are being served from the advantages of the MHealth (Mobile Health) applications and E-Health (health care supported by ICT) to improve, help and assist their health. Applications that have had a major refuge for these users, so intuitive environment. The Internet of things is increasingly allowing to integrate devices capable of connecting to the Internet and provide information on the state of health of patients and provide information in real time to doctors who assist. It is clear that chronic diseases such as diabetes, heart and pressure among others, are remarkable in the world economic and social level problem. The aim of this article is to develop an architecture based on an ontology capable of monitoring the health and workout routine recommendations to patients with chronic diseases. Through connected devices, it becomes easy for doctors and physicians to monitor patients' health. Also, real-time monitoring can save lives in a medical emergency like diabetic attacks, heart failure, asthma attacks, etc.

By means of a smart medical device connected to the smartphone app, collecting medical and other required health data will not be challenging. IoT devices collect and transfer health data like- blood pressure, oxygen, and blood sugar levels, weight, and ECGs.

Data collected from these devices are stored in the cloud and can be used by an authorized person, who could be a physician, insurance company, a participating health firm or an external consultant, regardless of their place, time, or device. Another objective of IoT in healthcare enables operability, machine-to-machine communication, information exchange, and data movement that ultimately makes the healthcare facility delivery effective and efficient.

Through connectivity protocols like Bluetooth LE, Wi-Fi, Z-wave, ZigBee, and other modern protocols, healthcare personnel can change the way they spot illnesses and ailments in patients and also innovating the ways of treatment.

Consequently, a technology-driven setup can cut regular visits to the health personnel while lowering the cost factor. Without the cloud, it is impossible to store a vast amount of data collected from healthcare mobile applications and devices. Also, for healthcare personnel, it is quite tough to acquire data originating from various devices and sources.

In such a situation, IoT devices can collect, report and analyze the data in real-time while cutting the need to store the raw data. In life-threatening circumstances, on-time alerts become quite critical.

To combat such situations, medical IoT devices and applications can gather vital data and transfer it to doctors and health personnel for real-time tracking. Also, these mobile applications and IoT devices can also send notifications regarding a patient's critical conditions irrespective of place, time. By using IoT enabled devices, doctors can monitor patients in real-time. Thus, the process of real-time monitoring at distinct places can help patients cut down not-so-necessary visits to doctors, hospital stays and re-admissions. Devices like Audemix reduces manual work which a doctor has to do during patient charting.

The device is powered by voice commands and also captures the patient's data.

While doing so, it makes the patient's data accessible for review.

CHAPTER 3

SYSTEM DESIGN

3.1 EXISTING SYSTEM

- The system used before in health observance is that the fastened observance system, which might be detected only the patient is within the hospital or bed. It takes abundant time for doctors additionally as patients. within the existing system, the patient has to get hospitalized for normal observance or routine medical.
- The systems are mensuration the health parameter of the patient and send by it

through totally different platform like Bluetooth protocol many more.

- These are used for under short-range communication to transfer the information. The doctor cannot fetch all the small print in the slightest degree times.

3.2 PROBLEM STATEMENT

- In rural hospitals, the facilities for health caring are limited. The poor quality of health management enables issues in health care system.

- In developing countries there is lack of resources and management to reach out the problems of individuals.

- . A common man cannot afford the expensive and daily check up for his health.

3.3 PROPOSED SYSTEM

- In our proposed system,we are using the Arduino Uno, Temperature Sensor, Pulse Sensor, ThingSpeak IoT platform, wifi Module, Power supply.

- An Smart patient health Monitoring System will not only help in maintaining health but also reducing the work of doctors and saving the time of patients.

- The proposed method of patient monitoring system monitors patient's health parameters using Arduino Uno. After connecting internet to the Arduino uno, it is connected to cloud database system which acts as a server. Then the server automatically sends data to the receiver system. Hence, it enables continuous monitoring of the patient's health parameters by the doctor. Any abrupt increase or decrease in these parameter values can be detected at the earliest and hence necessary medications can be implemented by the doctor immediately.

- Various varieties of sensors are interfaced with the microcontroller Arduino Uno to create the system smart. The info will display on both LCD and in their webpage.

- The most ideal of the system is to transmit the knowledge through the webpage to continuous monitoring of the patient over the internet. Such a system would constantly detect the important body parameters like temperature, vital signs and would compare it against a predetermined range set and if these values cross the particular limit, it will immediately alert the doctor, during this system, a microcontroller is employed to transmit the info.

- The doctor will simply access the patients health anytime from anyplace. An LCD is additionally connected to the microcontroller for the patients to look at their health status live.

3.4 ADVANTAGES

- Simple system to monitor the health parameters
- Response time is too good
- Immediately update in the IOT cloud web server
- Easy to implement
- Cost effective system

3.5 BLOCK DIAGRAM

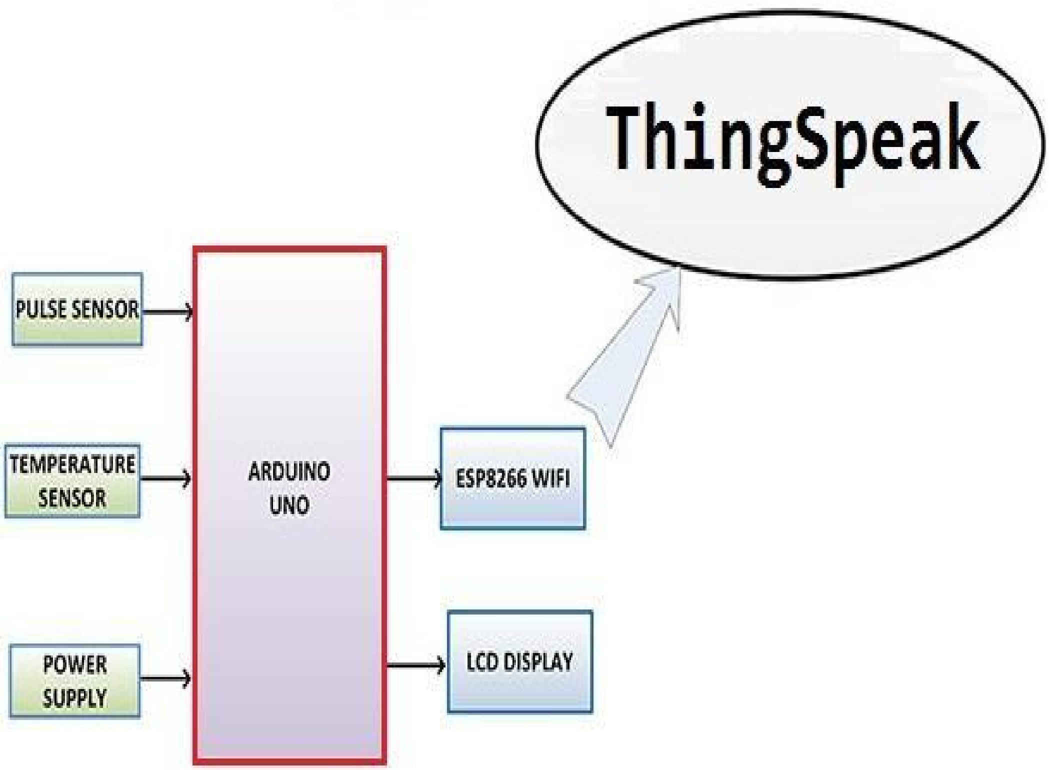

FIG 3.5

BLOCK DIAGRAM

3.6 Project Implementation

The system is implemented using the combination of hardware components. The smart patient health monitoring system will have sensors to detect body temperature, pulse rate and ECG data. The health monitoring sensors are used to collect health related data i.e. for data acquisition. Communication can be done by controller for sending data on internet wirelessly. Data processing has been done at server. All data collected and aggregated at server point. To get health related information in understandable format it can be shown on web page using Thing Speak IOT. All these data will be accessible in real time scenario for continuous monitoring. Health monitoring is the major problem in today's world. Due to lack of proper health monitoring, patient suffer from serious health issues. There are lots of IoT devices now days to monitor the health of patient over internet. Health experts are also taking advantage of these smart devices to keep an eye on their patients. With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry.

Here in this project, we will make an **IoT based Health Monitoring System** which records the patient heart beat rate and body temperature and also send an email/SMS alert whenever those readings goes beyond critical values. Pulse rate and body temperature readings are recorded over ThingSpeak and Google sheets so that patient health can be monitored from anywhere in the world over internet. A panic will also be attached so that patient can press it on emergency to send email/sms to their relatives.

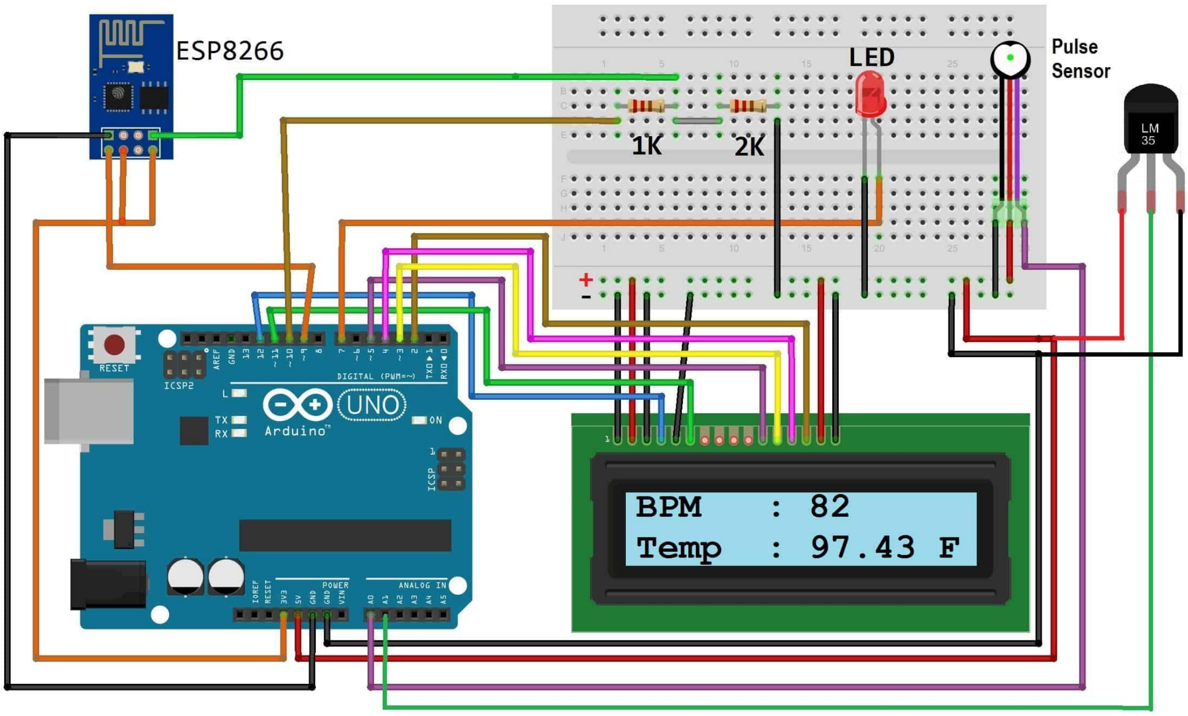

FIG-3.6 CIRCUIT DIAGRAM

3.7 HARDWARE REQUIREMENTS

- Arduino
- Temperature Sensor
- Pulse Sensor
- WIfi Module
- IOT Module
- Power Supply
- Connecting Cable

3.8 SOFTWARE REQUIREMENTS

- Arduino IDE
- Language: C
- ThingSpeak (IOT Cloud Server)

CHAPTER 4

HARDWARE DESIGN

4.1 ARDUINO UNO AND ITS PROGRAMMING

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer.

The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free. The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment. The USB connection with the PC is necessary to program the board and not just to power it up. The Uno automatically draw power from either the USB or an external power supply. Connect the board to your computer using the USB cable. The green power LED (labelled **PWR**) should go on. The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package. Arduino is a great tool for people of all skill levels. However, you will have a much better time learning along side your Arduino if you understand some basic fundamental electronics beforehand. We recommend that you have at least a decent understanding of these concepts before you dive in to the wonderful world of Arduino.

Arduino Uno:

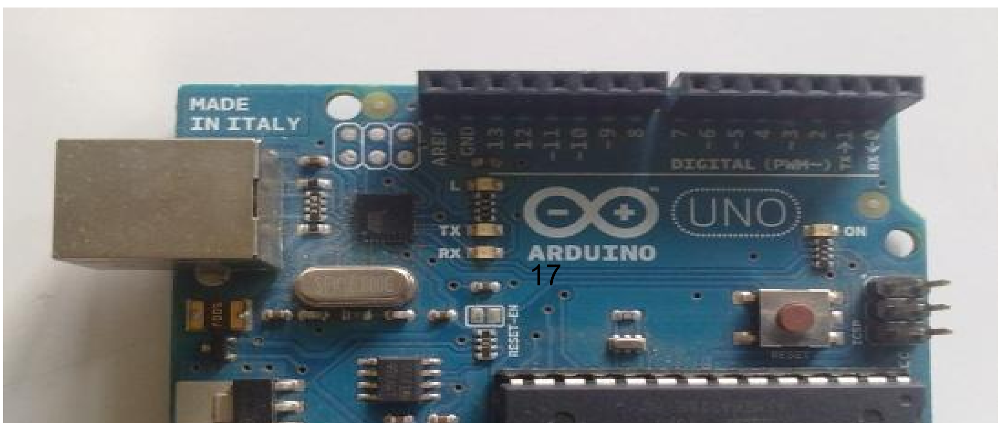

Fig 4.1 ARDUINO UNO

4.2 Overview

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

The Arduino Uno is a microcontroller board based on the ATmega328.It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

4.3 Summary:

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

4.4 Pin Configuration

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall- wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V**. this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.

• **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

• **GND**. Ground pins.

• **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

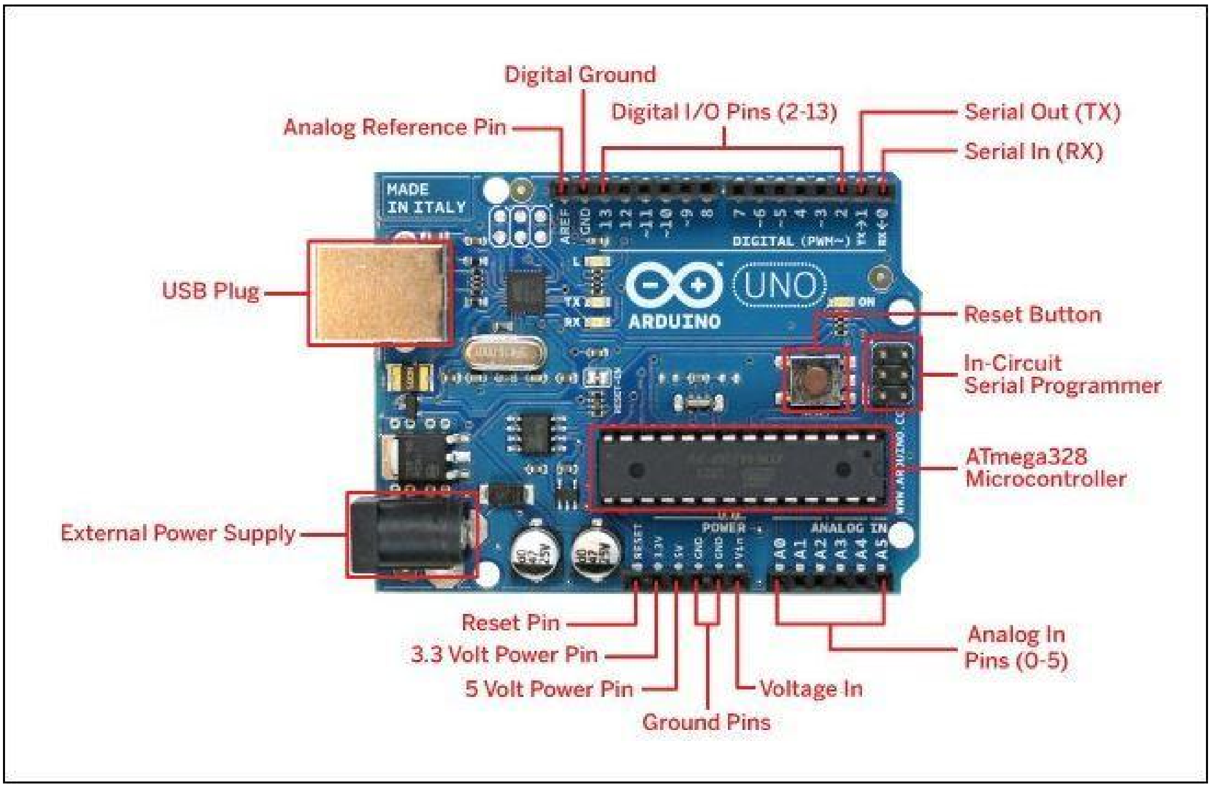

FIG-4.4 PIN DIAGRAM

4.5 Temperature Sensor

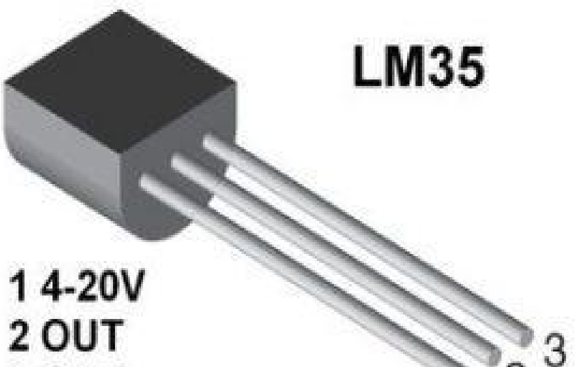

FIG-4.5 TEMPERATURE SENSOR PIN DIAGRAM

The LM35 is one kind of commonly used temperature sensor that can be used to measure temperature with an electrical o/p comparative to the temperature (in °C). It can measure temperature more correctly compare with a thermistor. This sensor generates a high output voltage than thermocouples and may not need that the output voltage is amplified. The LM35 has an output voltage that is proportional to the Celsius temperature. The scale factor is .01V/°C.

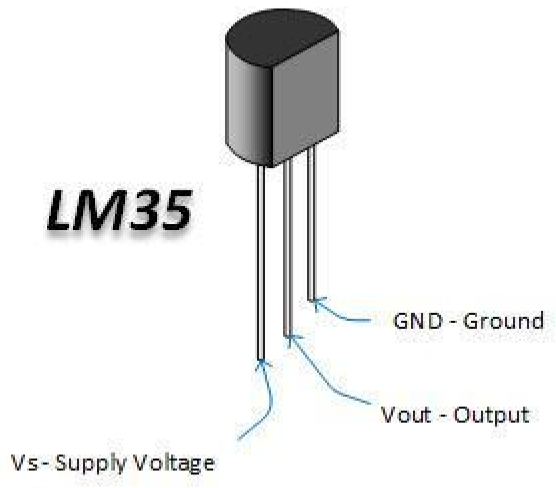

FIG 4.5 LM35 Temperature Sensor

The LM35 does not need any exterior calibration and maintains an exactness of +/-0.4°C at room temperature and +/-0.8°C over a range of 0°C to +100°C.One more significant characteristic of this sensor is that it draws just 60 microamps from its supply and acquires a low self-heating capacity. The LM35 temperature sensor available in many different packages like T0-46 metal can transistor-like package, TO-92 plastic transistor-like package, 8-lead surface mount SO-8 small outline package.

Pin Configuration

	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be the increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Connected to ground terminal of the circuit

LM35 Temperature Sensor Circuit Diagram

The LM35 temperature sensor is used to detect precise centigrade temperature. The output of this sensor changes describes the linearity. The o/p voltage of this IC sensor is linearly comparative to the Celsius temperature. The operating voltage range of this LM35 ranges from-55˚ to +150˚C and it has low-self heating. This is operated under 4 to 30 volts. The most extensively used electronic devices are operational amplifiers, which are certain kind of differential amplifiers. Temperature sensor circuit has terminals such as two inputs like non-inverting (+) and inverting (-) and only output pin. IC741 is used as a non-inverting amplifier. The variation between the i/p terminals amplifies the circuit.

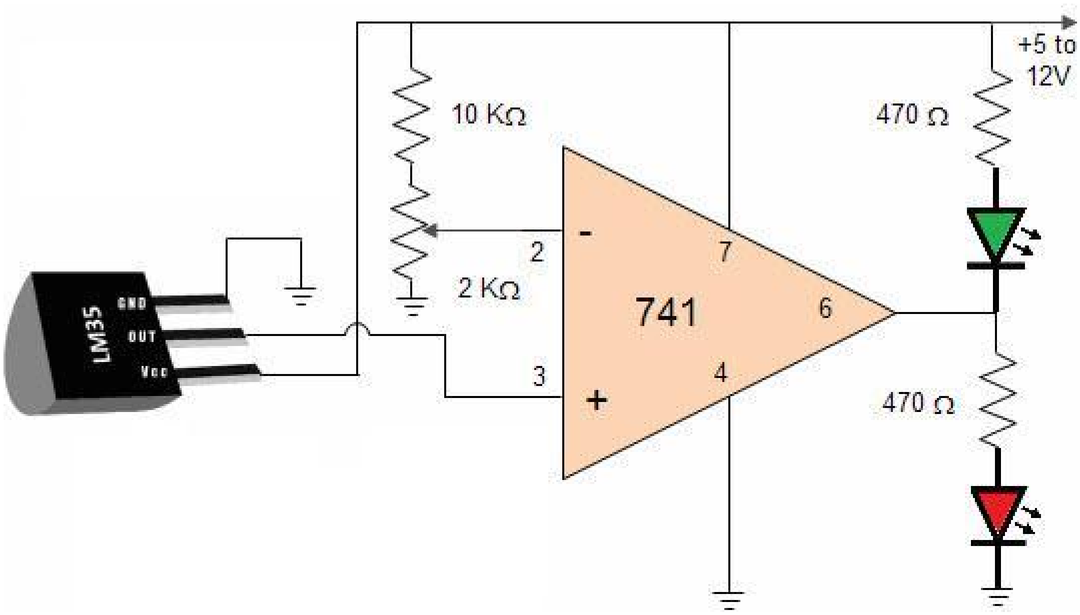

FIG.4.6 LM35 Circuit Diagram

The amount produced by IC2 amplifies in an amount to the temperature by 10 mV per degree. This unstable voltage is supply to a comparator IC 741. OP Amplifier is the most generally used electronic devices today. The IC 741 op-amp is one sort of differential amplifier. We have used IC741 as a non-inverting amplifier which means pin-3 is the input and the output is not inverted. This LM35 temperature sensor circuit amplifies the difference between its input terminals. The advantages of temperature sensor include It has no effect on the medium, more accurate, It has an easily conditioned output and It responds instantly.

4.6 Pulse Sensor

The sensor clips onto a fingertip and plugs right into Arduino with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time. 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.

Pulse Sensor is a well designed plug and play hear rate sensor for Arduino. It can be used by students, artists, athletes, makers and game & mobile developers who want

FIG.4.6 PULSE SENSOR

to easily incorporate live heart rate data into their projects. The sensor clips onto a fingertip and plugs right into Arduino with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time. 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.

Features

- Biometric Pulse Rate or Heart Rate detecting sensor
- Plug and Play type sensor
- Operating Voltage: +5V or +3.3V
- Current Consumption: 4mA
- Inbuilt Amplification and Noise cancellation circuit.
- Diameter: 0.625"
- Thickness: 0.125" Thick

Pin Number	Pin Name	Wire Colour	Description
1	Ground	Black	Connected to the ground of the system
2	Vcc	Red	Connect to +5V or +3.3V supply voltage
3	Signal	Purple	Pulsating output signal.

Pin Configuration:

Two Ways to Measure a Heartbeat

- 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 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. However pressure should be applied minimum and also fingers should be moved up and down till the pulse is felt.
- Using a sensor: Heart Beat can be measured based on optical power variation as light is scattered or absorbed during its path through the blood as the heart beat changes.

Working

The working of the Pulse/Heart beat sensor is very simple. The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side we have some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be your Finger tip or you ear tips, but it should be placed directly on top of a vein.
Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so if we monitor the flow of blood we can monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up more light since they will be reflect ted by the blood, this minor change in received light is analysed over time to determine our heart beats.

Applications

- Sleep Tracking
- Anxiety monitoring
- Remote patient monitoring/alarm system
- Health bands
- Advanced gaming consoles

4.7 Wifi Module

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability. The chip first came to the attention of Western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first there was almost no English-language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, the chip, and the software on it, as well as to translate the Chinese documentation. The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing the building of single chip devices capable of connecting to Wi-Fi. These microcontroller chips have been succeeded by the ESP32 family of devices, including the pin-compatible ESP32-C3. The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

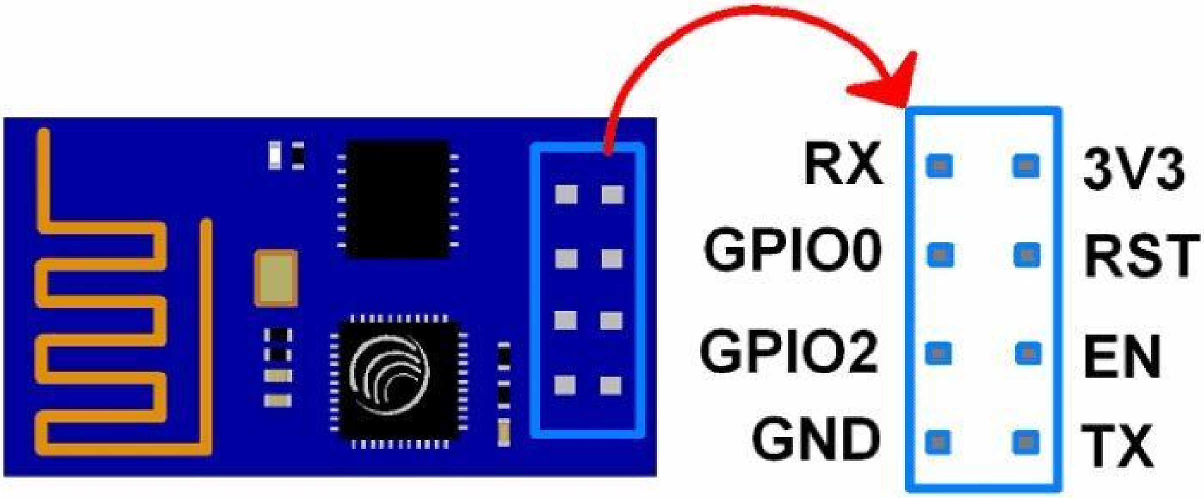

FIG.4.7 Wifi Module

ESP8266-01 Module Pin Description

3V3: - 3.3 V Power Pin.

GND: - Ground Pin.

RST: - Active Low Reset Pin.

EN: - Active High Enable Pin.

TX: - Serial Transmit Pin of UART.

RX: - Serial Receive Pin of UART.

GPIO0 & GPIO2: - General Purpose I/O Pins. These pins decide what mode (boot or normal) the module starts up in. It also decides whether the TX/RX pins are used for Programming the module or for serial I/O purpose.

To program the module using UART, Connect GPIO0 to ground and GPIO2 to VCC or leave it open. To use UART for normal Serial I/O leave both the pins open (neither VCC norGround).

ESP8266 comes with capabilities of

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),

- general-purpose input/output (16 GPIO),
- Inter-Integrated Circuit (I²C) serial communication protocol,
- analog-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I²S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and
- pulse-width modulation (PWM).

It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI.

ESP8266 module is low cost standalone wireless transceiver that can be used for end-point IoT developments.

To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate.

Working

The ESP8266 is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making Internet of Things as easy as possible. It can also fetch data from internet using API's hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. However this version of the module has only 2 GPIO pins (you can hack it to use upto 4) so you have to use it along with another microcontroller like Arduino, else you can look onto the more standalone ESP-12 or

ESP-32 versions. So if you are looking for a module to get started with IOT or to provide internet connectivity to your project then this module is the right choice for you. The **ESP8266** is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making **Internet of Things** as easy as possible. It can also fetch data from internet using API's hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. However this version of the module has only 2 GPIO pins (you can hack it to use upto 4) so you have to use it along with another microcontroller like Arduino, else you can look onto the more standalone **ESP-12** or **ESP-32** versions. So if you are looking for a **module to get started with IOT** or to provide internet connectivity to your project then this module is the right choice for you. There are so many methods and IDEs available to with ESP modules, but the most commonly used on is the Arduino IDE. So let us discuss only about that further below.

The **ESP8266 module** works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with your circuits. The best way to program an **ESP-01** is by using the FTDI board that supports 3.3V programming. If you don't have one it is recommended to buy one or for time being you can also use an Arduino board. One commonly problem that every one faces with ESP-01 is the powering up problem. The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider. So it is important to make a small voltage regulator for 3.31v that could supply a minimum of 500mA. One recommended regulator is the LM317 which could handle the job easily

Applications

- IOT Projects
- Access Point Portals

- Wireless Data logging
- Smart Home Automation
- Learn basics of networking
- Portable Electronics
- Smart bulbs and Sockets

4.8 Thing Speak

ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates".ThingSpeak was originally launched by ioT Bridge in 2010 as a service in support of IoT applications.ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks,allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.ThingSpeak has a close relationship with Mathworks, Inc. In fact, all of the ThingSpeak documentation is incorporated into the Mathworks' Matlab documentation site and even enabling registered Mathworks user accounts as valid login credentials on the ThingSpeak website.

FIG.4.8 ARCHITECTURE OF THINGSPEAK

ThingSpeak enables sensors, instruments, and websites to send data to the cloud where it is stored in either a private or a public channel. ... Once data is in a **ThingSpeak** channel, you can analyze and visualize it, calculate new data, or interact with social media, web services, and other devices. ThingSpeak™ is an IoT analytics platform service from MathWorks®, the makers of MATLAB® and Simulink®. ThingSpeak allows you to aggregate, visualize, and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices or equipment. Execute MATLAB code in ThingSpeak, and perform online analysis and processing of the data as it comes in. ThingSpeak accelerates the development of proof-of-concept IoT systems, especially those that require analytics. You can build IoT systems without setting up servers or developing web software. For small- to medium-sized IoT systems, ThingSpeak provides a hosted solution that can be used in production.

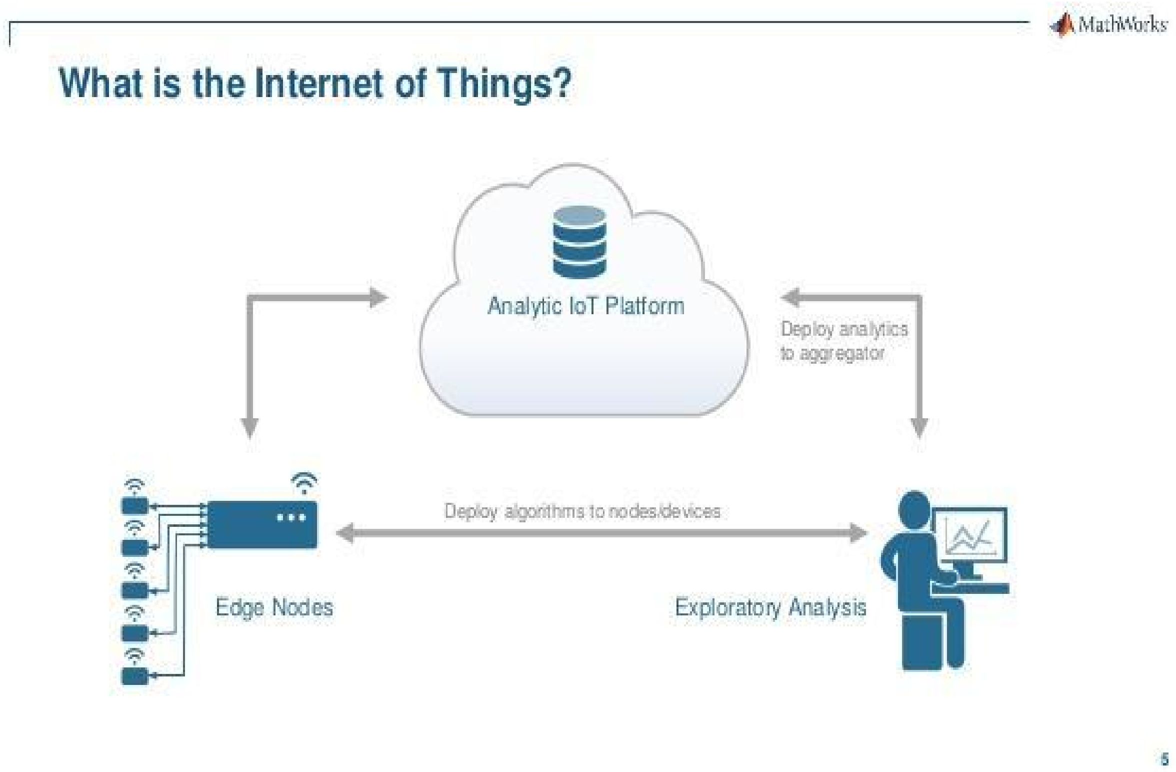

FIG.4.8 BLOCK DIAGRAM

4.9 LCD Display

Flat panel LCDs can be of various types, so depending on your project needs, you will have to go with the right screen.

The most basic version uses “seven-segment displays” commonly seen in calculators, digital alarms and gas station pump displays. These simply consist of vanes which can be lit in various combinations to produce a numerical reading.

On the other hand, there are “thin-film transistor (TFT) LCDs” used in television sets, laptops, smartphones and complex electronic devices. They utilize pixel algorithms to provide fast response times, wide viewing angles, high contrast and accurate color schemes.

“Dot-matrix LCD” lies somewhere in the middle of the basic and the most advanced LCD technologies. They are useful in displaying alphabets, numbers and upper and lower cases of letters.

One of the most common dot matrix LCD displays is the 2×16 module which refers to two rows and sixteen columns of LCD display pins. This is what we will be using in our connections with the Arduino board. The Arduino website offers a detailed example of an assembly. It is compatible with the built-in library example available for Arduino IDE.

While the circuit may look complex at first glance, here is a simple step-by-step explanation of the various stages.

First, all sixteen pins of the LCD should be connected to the bottom positive rail of the breadboard.

The first and last pins of LCD are then connected to the negative rail on the top. Arduino's GND and 5 V pins on the power-source side are connected to the positive rail contrasting with the LCD connections.

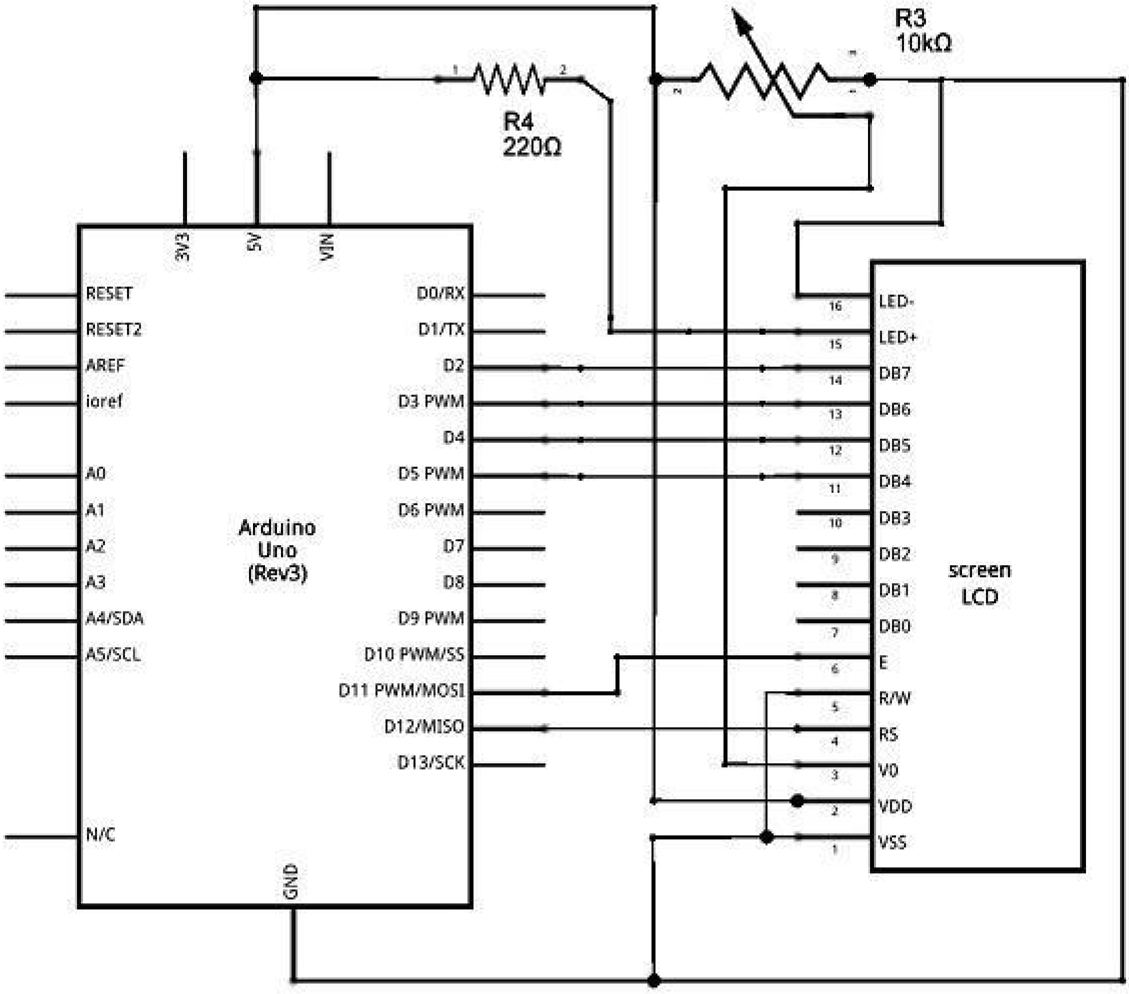

FIG.4.9 PIN DIAGRAM OF LCD DISPLAY

LCD screens are one of the most useful applications of Arduino smart device projects. As shown in this tutorial, it is possible to interface an LCD screen using a standard example from Arduino library. IOT 16×2 LCD can be used for the remote data monitoring using Nodemcu Esp8266 wifi Module and Blynk Application. I have been using 16×2 LCD for displaying the sensors data. 16×2 LCD Physical interfacing with Arduino or Mega or Nodemcu esp8266 wifi module is really a hectic job, as it needs a lot of soldering and needs many IO pins to control a 16×2 LCD.

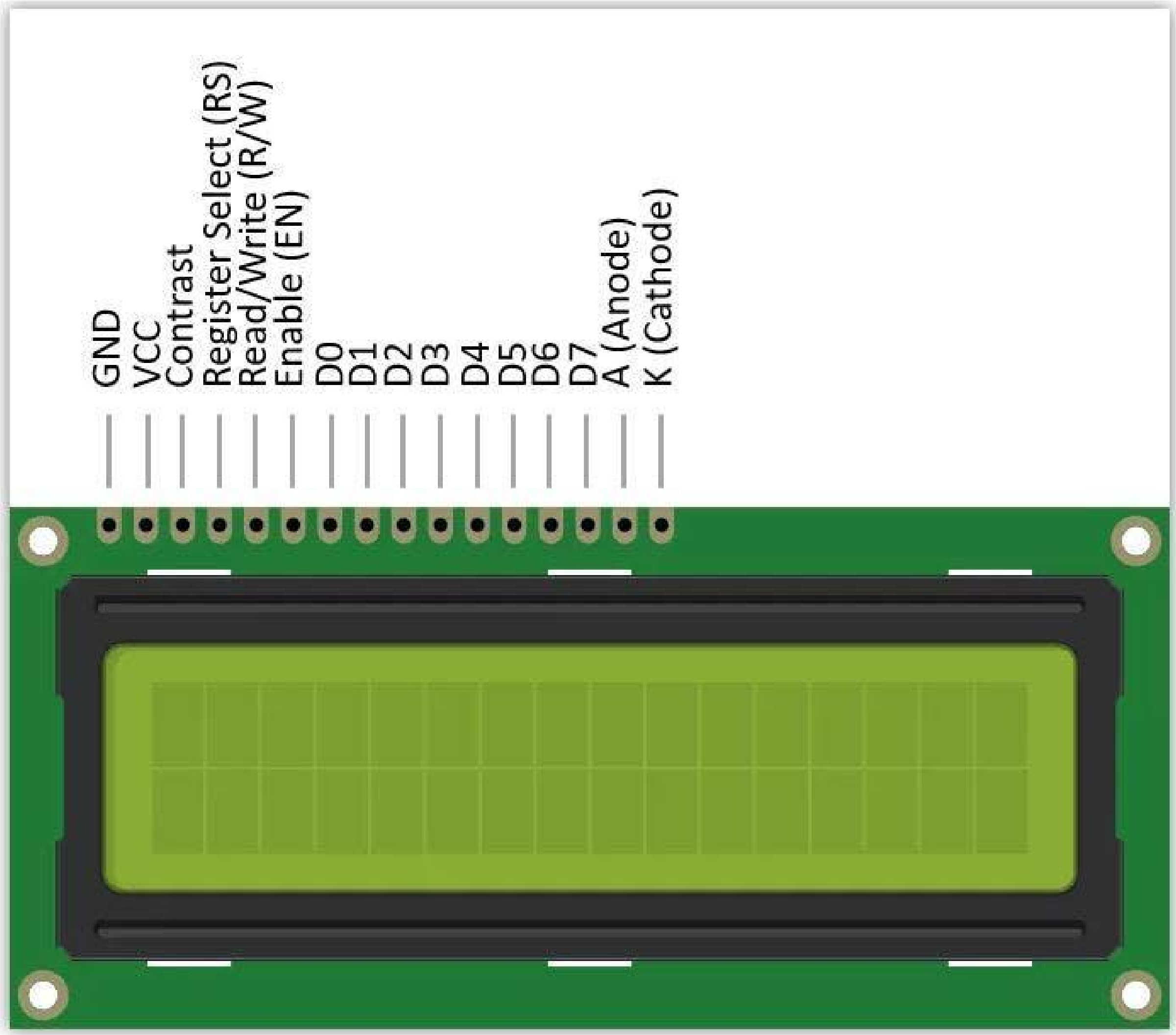

FIG.4.9- LCD DISPLAY

CHAPTER 5
RESULT AND CONCLUSION

Our goal is to implement a smart patient health monitoring system that can monitor the heartbeat and body temperature. The smart patient health care monitoring system developed by us has numerous applications. These types of healthcare systems can be implemented in hospitals as well as at home places where a person needs to have immediate medical attention whenever his/her health goes unstable. As we are using the Thing Speak IoT platform with the help of Thing Speak we can easily capture sensor data. This allows to keep a track of patients' heartbeat and body temperature value with change in time. This would give the doctor a more wide perspective of treating the patient in a much effective way within less time. The system developed patient monitoring based on Internet of things, is an alternative that can be used to help patients with chronic diseases. Likewise with this set of solutions the aim is to improve the quality of life of patients, not just monitoring them, but also to enable direct them to improve their eating habits and workout routines. The context model developed for the system proved to be efficient when making inferences related to the context, such as recommendations for taking measures through sensors, as well as recommendations and workout routines tips to improve the eating habits of patients.

5.1 Conclusion

The Internet of Things is considered now as one of the feasible solutions for any remote value tracking especially in the field of health monitoring. It facilitates that the individual prosperity parameter data is secured inside the cloud, stays in the hospital are reduced for conventional routine examinations and most important that the health can be monitored and disease diagnosed by any doctor at any distance. In this paper, an IoT based health monitoring system was developed. The system monitored body temperature, pulse rate and room humidity and temperature using sensors, which are also displayed on a LCD. These sensor values are then sent to a medical server using wireless communication. These data are then received in an authorized personals smart phone with IoT platform. With the values received the doctor then diagnose the disease and the state of health of the patient. The main objective of the experiment was successfully achieved. All the individual modules like Heartbeat detection module, fall detection module etc. and remote viewing module gave out the intended results. 55 The designed system modules can further be optimized and produced to a final single circuit. More important fact that came up during project design is that all the circuit components used in the remote health detection system are available easily.

In this paper, we found the importance and fruitful benefits of implementation of IoT in remote health monitoring systems.The compact sensors with IoT will make a huge impact on every patient's life, that even though they are away from home and physician, this helps them to reduce the fear of danger. The sensory data can be acquired in home or work environments. Also, the challenges in sensing, analytics and prediction of the disease are also highlighted and those can be addressed to provide a seamless integraton.. The fundamental element of people's needs is health .Humans face a haul of surprising death and plenty of diseases because of varied diseases that are a result of lack of treatment to the patients at right time. The main objective of this project is to develop a reliable sensible patient health observance system victimization IoT so the attention professionals will monitor their patients. The sensors will be either worn or be embedded into the body of the patients, to unendingly monitor their health. the knowledge collected in such a fashion will behold on, analyzed, and well-mined to try and do the first prediction of diseases

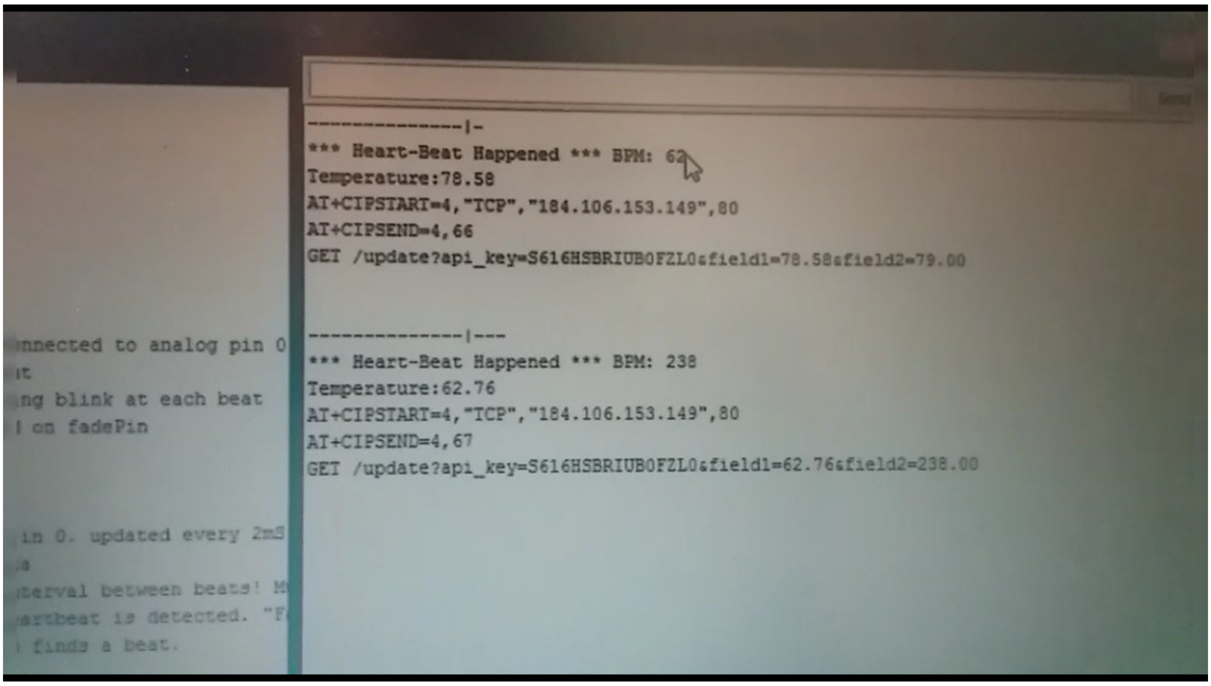

FIG.5.1 SERIAL MONITOR

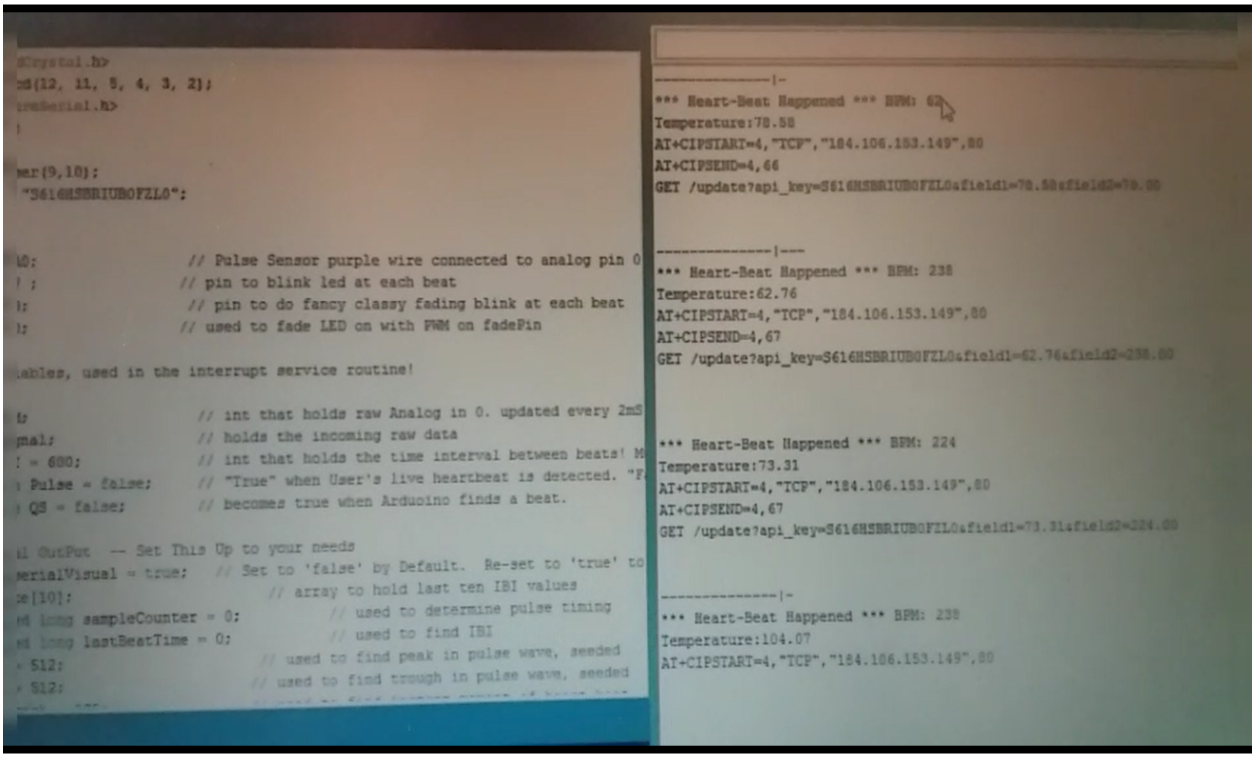

FIG.5.1 SERIAL MONITOR

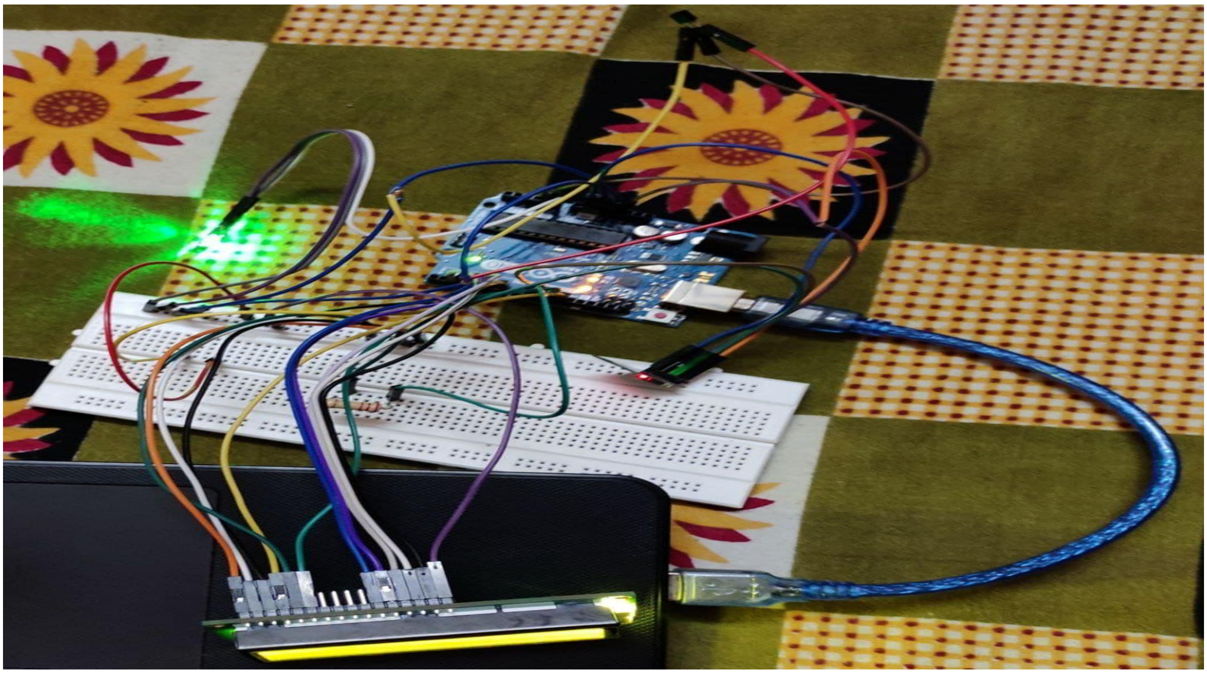

FIG.5.2 STRUCTURE OF PROJECT

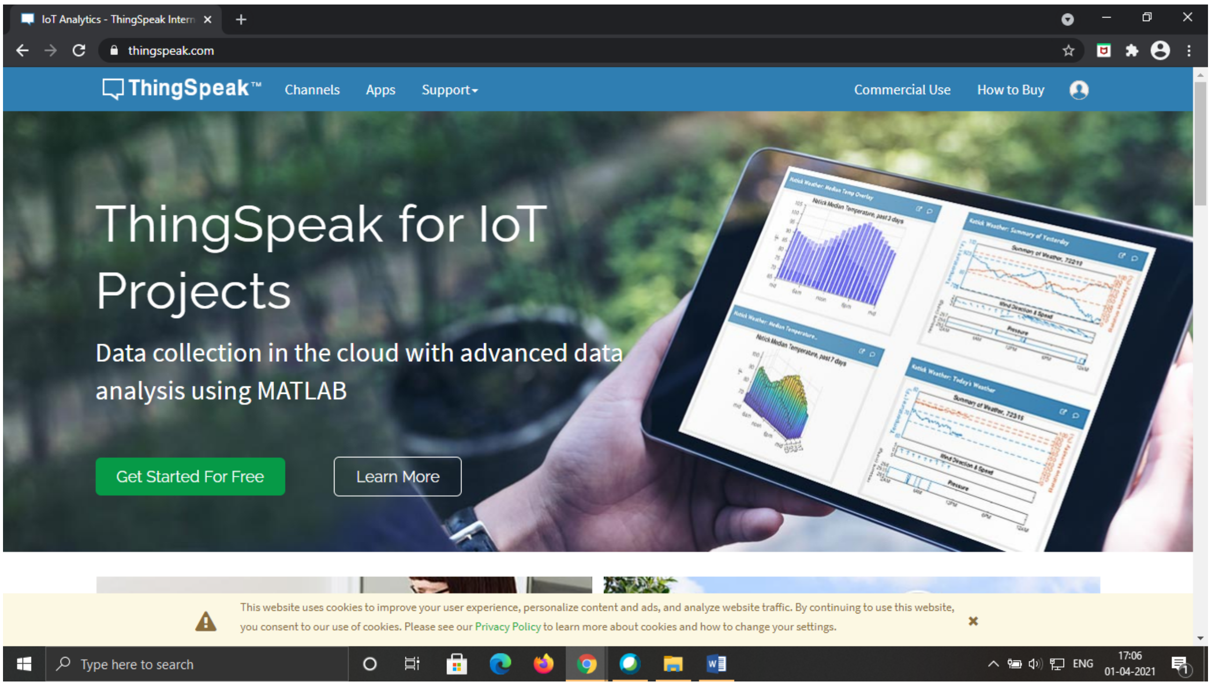

FIG.5.3 THING SPEAK PLATFORM

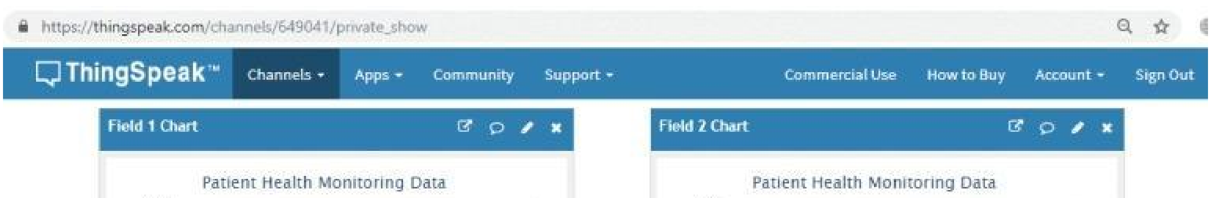

FIG.5.4 OUTPUT CONSOLE

REFERENCE PAPERS:

- Mohammad M. Masud, Mohamed Adel Serhani, and Alramzana Nujum Navaz "Resource-Aware MobileBased Health Monitoring", 2168-2194 (c) 2015 IEEE.

- M.A. Miah et al., "Continuous heart beat and body temperature monitoring system using Arduino UNO and Android device, International Conference on Electrical Information and Communication Technology (EICT), pp. 189-194, 2015.

- Maradugu Anil Kumar, Y.Ravi Sekhar, "Android Based Health Care Monitoring System" IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIIECS'1.

- Hamid Al-Hamadi and Ing-Ray Chen, "Trust-Based Decision Making for Health IoT Systems" DOI 10.1109/JIOT.2017.2736446, IEEE Internet of Things Journal.

- Gulraiz J. Joyia, Rao M. Liaqat, Aftab Farooq, and Saad Rehman, Internet of Medical Things (IOMT): Applications, Benefits and Future Challenges in Healthcare Domain, Journal of Communications Vol. 12, No. 4, April 2017.

- . Shubham Banka, Isha Madan and S.S. Saranya, Smart Healthcare Monitoring using IoT. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15, pp. 11984-11989, 2018. 4.

- K. Perumal, M. Manohar, A Survey on Internet of Things: Case Studies, Applications, and Future Directions, In Internet of Things: Novel Advances and Envisioned Applications, Springer International Publishing, (2017) 281-297.

SOURCE CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#include <SoftwareSerial.h>
float pulse = 0;
float temp = 0;
SoftwareSerial ser(9,10);
String apiKey = "OO707TGA1BLUNN12";

// Variables
int pulsePin = A0; // Pulse Sensor purple wire connected to analog pin 0
int blinkPin = 7 ; // pin to blink led at each beat
int fadePin = 13; // pin to do fancy classy fading blink at each beat
int fadeRate = 0; // used to fade LED on with PWM on fadePin

// Volatile Variables, used in the interrupt service routine!

volatile int BPM; // int that holds raw Analog in 0. updated every 2mS
volatile int Signal; // holds the incoming raw data
volatile int IBI = 600; // int that holds the time interval between beats! Must be seeded!
volatile boolean Pulse = false; // "True" when User's live heartbeat is detected. "False" when nota "live beat".
volatile boolean QS = false; // becomes true when Arduoino finds a beat.

// Regards Serial OutPut -- Set This Up to your needs
static boolean serialVisual = true; // Set to 'false' by Default. Re-set to 'true' to see Arduino Serial Monitor ASCII Visual Pulse
volatile int rate[10]; // array to hold last ten IBI values
```

```
volatile unsigned long sampleCounter = 0; // used to determine pulse timing
volatile unsigned long lastBeatTime = 0; // used to find IBI
volatile int P = 512; // used to find peak in pulse wave, seeded
volatile int T = 512; // used to find trough in pulse wave, seeded
volatile int thresh = 525; // used to find instant moment of heart beat, seeded
volatile int amp = 100; // used to hold amplitude of pulse waveform, seeded
volatile boolean firstBeat = true; // used to seed rate array so we startup with reasonable BPM
volatile boolean secondBeat = false; // used to seed rate array so we startup with reasonable BPM

void setup()
{
lcd.begin(16, 2);
pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat!
pinMode(fadePin,OUTPUT); // pin that will fade to your heartbeat!
Serial.begin(115200); // we agree to talk fast!
interruptSetup(); // sets up to read Pulse Sensor signal every 2mS

// IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN THE BOARD VOLTAGE,

// UN-COMMENT THE NEXT LINE AND APPLY THAT VOLTAGE TO THE A-REF PIN

// analogReference(EXTERNAL);

lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Patient Health");
lcd.setCursor(0,1);
lcd.print(" Monitoring ");
```

```
delay(4000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Initializing....");
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Getting Data....");
ser.begin(9600);
ser.println("AT");
delay(1000);
ser.println("AT+GMR");
delay(1000);
ser.println("AT+CWMODE=3");
delay(1000);
ser.println("AT+RST");
delay(5000);
ser.println("AT+CIPMUX=1");
delay(1000);

String cmd="AT+CWJAP=\"Alexahome\",\"98765432\"";
ser.println(cmd);
delay(1000);
ser.println("AT+CIFSR");
delay(1000);
}

// Where the Magic Happens
void loop()
{
serialOutput();
if (QS == true) // A Heartbeat Was Found
```

```
{

// BPM and IBI have been Determined
// Quantified Self "QS" true when arduino finds a heartbeat
fadeRate = 255; // Makes the LED Fade Effect Happen, Set 'fadeRate' Variable to
255 to fade LED with pulse
serialOutputWhenBeatHappens(); // A Beat Happened, Output that to serial.
QS = false; // reset the Quantified Self flag for next time
}
ledFadeToBeat(); // Makes the LED Fade Effect Happen
delay(20); // take a break
read_temp();
esp_8266();
}
void ledFadeToBeat()
{
fadeRate -= 15; // set LED fade value
fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into
negative numbers!
analogWrite(fadePin,fadeRate); // fade LED
}
void interruptSetup()
{
// Initializes Timer2 to throw an interrupt every 2mS.
TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO
CTC MODE
TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER
OCR2A = 0X7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE
RATE
TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND
OCR2A
sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
```

```
}
void serialOutput()
{ // Decide How To Output Serial.
if (serialVisual == true)
{
arduinoSerialMonitorVisual('-', Signal); // goes to function that makes Serial Monitor Visualizer
}
else
{
sendDataToSerial('S', Signal); // goes to sendDataToSerial function
}
}
void serialOutputWhenBeatHappens()
{
if (serialVisual == true) // Code to Make the Serial Monitor Visualizer Work
{
Serial.print("*** Heart-Beat Happened *** "); //ASCII Art Madness
Serial.print("BPM: ");
Serial.println(BPM);
}
else
{
sendDataToSerial('B',BPM); // send heart rate with a 'B' prefix
sendDataToSerial('Q',IBI); // send time between beats with a 'Q' prefix
}
}
void arduinoSerialMonitorVisual(char symbol, int data )
{
const int sensorMin = 0; // sensor minimum, discovered through experiment
const int sensorMax = 1024; // sensor maximum, discovered through experiment
int sensorReading = data; // map the sensor range to a range of 12 options:
```

```
int range = map(sensorReading, sensorMin, sensorMax, 0, 11);
// do something different depending on the
// range value:
switch (range)
{
case 0:
Serial.println(""); /////ASCII Art Madness
break;
case 1:
Serial.println("---");
break;
case 2:
Serial.println("------");
break;
case 3:
Serial.println("---------");
break;
case 4:
Serial.println("------------");
break;
case 5:
Serial.println("--------------|-");
break;
case 6:
Serial.println("--------------|---");
break;
case 7:
Serial.println("--------------|-------");
break;
case 8:
Serial.println("--------------|----------");
break;
```

```
case 9:
Serial.println("--------------|----------------");
break;
case 10:
Serial.println("--------------|-------------------");
break;
case 11:
Serial.println("--------------|-----------------------");
break;
}
}

void sendDataToSerial(char symbol, int data )
{
Serial.print(symbol);
Serial.println(data);
}
ISR(TIMER2_COMPA_vect) //triggered when Timer2 counts to 124
{
cli(); // disable interrupts while we do this
Signal = analogRead(pulsePin); // read the Pulse Sensor
sampleCounter += 2; // keep track of the time in mS with this variable
int N = sampleCounter - lastBeatTime; // monitor the time since the last beat to avoid
noise
// find the peak and trough of the pulse wave

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

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

if(secondBeat)
{ // if this is the second beat, if secondBeat == TRUE
secondBeat = false; // clear secondBeat flag
for(int i=0; i<=9; i++) // seed the running total to get a realisitic BPM at startup
{
rate[i] = IBI;
}
}
if(firstBeat) // if it's the first time we found a beat, if firstBeat == TRUE
{
firstBeat = false; // clear firstBeat flag
secondBeat = true; // set the second beat flag
sei(); // enable interrupts again
return; // IBI value is unreliable so discard it
}
// keep a running total of the last 10 IBI values
```

```
word runningTotal = 0; // clear the runningTotal variable
for(int i=0; i<=8; i++)
{ // shift data in the rate array
rate[i] = rate[i+1]; // and drop the oldest IBI value
runningTotal += rate[i]; // add up the 9 oldest IBI values
}
rate[9] = IBI; // add the latest IBI to the rate array
runningTotal += rate[9]; // add the latest IBI to runningTotal
runningTotal /= 10; // average the last 10 IBI values
BPM = 60000/runningTotal; // how many beats can fit into a minute? that's BPM!
QS = true; // set Quantified Self flag
// QS FLAG IS NOT CLEARED INSIDE THIS ISR
pulse = BPM;
}
}
if (Signal < thresh && Pulse == true)
{ // when the values are going down, the beat is over
digitalWrite(blinkPin,LOW); // turn off pin 13 LED
Pulse = false; // reset the Pulse flag so we can do it again
amp = P - T; // get amplitude of the pulse wave
thresh = amp/2 + T; // set thresh at 50% of the amplitude
P = thresh; // reset these for next time
T = thresh;
}
if (N > 2500)
{ // if 2.5 seconds go by without a beat
thresh = 512; // set thresh default
P = 512; // set P default
T = 512; // set T default
lastBeatTime = sampleCounter; // bring the lastBeatTime up to date
firstBeat = true; // set these to avoid noise
secondBeat = false; // when we get the heartbeat back
```

```
}
sei(); // enable interrupts when youre done!
}// end isr
void esp_8266()
{
// TCP connection AT+CIPSTART=4,"TCP","184.106.153.149",80
String cmd = "AT+CIPSTART=4,\"TCP\",\"";
cmd += "184.106.153.149"; // api.thingspeak.com
cmd += "\",80";
ser.println(cmd);
Serial.println(cmd);
if(ser.find("Error"))
{
Serial.println("AT+CIPSTART error");
return;
}
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr +="&field1=";
getStr +=String(temp);
getStr +="&field2=";
getStr +=String(pulse);
getStr += "\r\n\r\n";
// send data length
cmd = "AT+CIPSEND=4,";
cmd += String(getStr.length());
ser.println(cmd);
Serial.println(cmd);
delay(1000);
ser.print(getStr);
Serial.println(getStr); //thingspeak needs 15 sec delay between updates
delay(3000);
```

```
}
void read_temp()
{
int temp_val = analogRead(A1);
float mv = (temp_val/1024.0)*5000;
float cel = mv/10;
temp = (cel*9)/5 + 32;
Serial.print("Temperature:");
Serial.println(temp);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("BPM :");
lcd.setCursor(7,0);
lcd.print(BPM);
lcd.setCursor(0,1);
lcd.print("Temp.:");
lcd.setCursor(7,1);
lcd.print(temp);
lcd.setCursor(13,1);
lcd.print("F");
}
```