A

PROJECT REPORT ON:

HEARTBEAT AND TEMPERATURE MONITOR BASED ON IOT

DEPARTMENT OF-

ELECTRONICS AND COMMUNICATION ENGINEERING.

Submitted by

ECE- BATCH-2

Under the valuable guidance of

Mr. J. Mallesh MTECH

GOVT. POLYTECHNIC, WARANGAL.

Near MGM Circle, Warangal,

TELANGANA, 506007.

A

PROJECT REPORT ON

HEARTBEAT AND TEMPERATURE MONITOR BASED ON IOT

CERTIFICATE:

This is to certify that the project entitled “HEARTBEAT AND TEMPERATURE MONITOR VIA SMS GATEWAY” is a bonafide work carried out by ECE- BATCH-2 under the guidance of Mr. J.Mallesh and is submitted in partial fulfillment for leading to the award of diploma in "ELECTRONICS AND COMMUNICATION ENGINEERING", by State board of Technical & training, Hyderabad, Telangana during the academic year 2019-2022

PROJECT GUIDE	HEAD OF DEPT.	PRINCIPAL
Mr. j. Mallesh MTECH	Mr. E. Mallikarjun MTECH	Dr. B. Prabhakar PHD, M.T echB.E(ECE),MISTE,FIE

ACKNOWLEDGEMENT

We would like to express our special thanks to our lecturer J.MALLESH SIR as well as our head of department E. MALLIKARJUN SIR and our principal B. PRABHAKAR SIR who gave us the golden opportunity to do this wonderful project on the topic HEARTBEAT AND TEMPERATURE MONITOR USING ARDUINO WITH INTERNET OF THINGS(ioT), which also helped us in doing a lot of Research and we came to know about so many new things.

We are really thankful to them. Secondly, we would also like to thank our friends who helped us a lot in finishing this project within the limited time.

We are making this project not only for marks but to also increase our knowledge.

THANKING YOU

<u>Submitted by:</u>

19004-EC-219	K. SURAJ
19004-EC-206	V. PRANATHI
19004-EC-213	D. PRASANTH
19004-EC-258	J. VENKATESH
19004-EC-229	G. NIKHIL
19004-EC-224	TASLEEM
190004-EC-259	CH. DEEPAK

TABLE OF CONTENT

ABSTRACT

This study is to develop a new system that monitors heartbeat rate and also body temperature at the same time via mobile phone. The heartbeat sensor was initially designed to measure internal temperature and heartbeat rate of the human body which is highly related to heat stroke and heart attack access. The cognition behind this current study is to determine the impact for the implementation of this monitoring device towards the children's body temperature state. Heat stroke conditions can be caused by body overheating where usually as a result of prolonged exposure to or physical exertion in high temperature. This is the most serious form of heat injury; heatstroke and it can occur if body temperature rises to 104 F (40 C) or higher. The results obtained undeniably imply that the body temperature increases significantly when children are exposed to the environment and heart attack cases cause the person's heartbeat to be at a dangerous rate at the wrong moment. Thus, the importance of this device provides information about when the temperature and BPM goes out of range.

KEYWORDS: HEARTBEAT SENSOR, LM35, ESP8266, ARDUINO.

INTRODUCTION:

Heatstroke is a condition caused by your body overheating, usually as a result of prolonged exposure to or physical exertion in high temperatures. This most serious form of heat injury, heatstroke, can occur if your body temperature rises to 104 F (40 C) or higher. The condition is most common in the summer months. Heatstroke Requires emergency treatment. Untreated heat stroke can quickly damage your brain, heart, kidneys and muscles. The damage worsens the longer treatment is delayed, increasing your risk of serious complications or death. Heat stroke symptoms are high body temperature, altered mental state or behavior. Confusion, agitation, slurred speech, irritability, delirium, seizures and coma can all result from heatstroke. Alteration in sweating. In heat stroke brought on by hot weather, skin will feel hot and dry to the touch. However, in heat stroke brought on by strenuous exercise, skin may feel dry or slightly moist. Nausea and vomiting are also one of the symptoms. That person may feel sick to their stomach. Heart rate and breathing rate also increases. Cases of heat stroke spike at the end of June into July each year and continue through August. Troy Smurawa, M.D., Director of Pediatric Sports Medicine at the Children's Health Andrews Institute for Orthopedics and Sports Medicine, says that heat stroke in children can be extremely serious. The Health Ministry recorded 14 heat-related illnesses from March 1 until today due to the recent heat wave phenomenon. Health Minister Datuk Seri Dr S. Subramaniam said the cases comprised 11 heat exhaustion and 3 heat stroke cases. The medical records of three children who were entrapped inside vehicles are reviewed and their outcome following the incidents were assessed in this report. The children developed heat stroke following the incidents and survived after several days in coma but with severe cognitive functions impairment. Two of the children were left with hyperactivity and attention deficit, while the third had active epilepsy. Vehicular entrapment heat stroke is one of the preventable brain injuries in children. Several children get entrapped in cars or other vehicles yearly and survivors are left with significant brain damage. Now-a-days health problems like cardiac failure, lung failures & heart related diseases are arising day by day at a very high rate. Due to these problems, time to time health monitoring is very essential. A modern concept is health monitoring of a patient wirelessly. It is a major development in the medical arena. Health professionals have developed a brilliant and inexpensive health monitoring system for providing more comfortable living to the people suffering from various diseases using leading technologies like wireless communications, wearable and portable remote health monitoring devices. As visits of doctors to the patients constantly are decreased as the information regarding the patient's health directly reaches the doctor's monitor screen from anywhere the patient resides . Also, based on this doctors can save many lives by imparting them a quick & valuable service. According to recent statistics, nearly two million people suffer from heart attacks every year and one person dies every 33 seconds in India. The World Health Organization (WHO) reports that the heart disease rate might increase to 23.3% worldwide by the year 2030. The treatment of such chronic disease requires continuous and Long Term monitoring to have proper control on it. IoT helps to move from manual heart rate monitoring systems to remote heart rate monitoring systems A doctor may not be present all the time to provide medication or treatment to the patients or a guardian may not be present all the time to take the patient to the hospital. Hence, our proposed system is the right solution for this problem. The remote heart rate monitoring system is used to monitor physical parameters like heartbeat and send the measured heart rate directly to a doctor through Email or SMS. In today's era, health problems are increasing day-by-day at a high pace. The death rate of 55.3 million people dying each year or 151,600 people dying each day or 6316 people dying each hour is a big issue for all over the world. Hence it is the need of hour to overcome such problems. We, therefore, propose a change in wireless sensors technology by designing a system which includes different wireless sensors to receive information with respective human body temperature, blood pressure, saline level, heart rate etc. that will be undoubtedly further transmitted on an IoT platform which is accessible by the user via the internet. An accessible database is created about a patient's health history which can be further monitored & analyzed by the doctor if necessary. This paper proposes a health monitoring system which is capable of detecting multiple parameters of our body such as blood pressure, temperature, heart rate. A continuous record of body health parameters can be used to detect the disease in a more efficient manner. Now-a-days, people pay more attention towards prevention & early recognition of disease. In addition to it, new generation mobile phones technologies. Proposed system consists of a pulse rate sensor, Arduino Nano and Raspberry Pi 3. This system is able to measure the heart rate of an infant to an elderly person. The low cost of the device helps to provide an appropriate portable remote based effective heart rate monitoring system. The system is based on advanced wireless and wearable sensor technology. The rapid growth in technology has remarkably enhanced the scope of remote health monitoring systems. Thus, in such an environment the proposed system serves to be of effective cost with ease of application.

LITERATURE RESEARCH:

First of all, monitoring heart rate is very important for athletes, patients as it determines the condition of the heart (just heart rate). There are many ways to measure heart rate and the most precise one is using an Electrocardiography but the easier way to monitor the heart rate is to use a Heartbeat Sensor. It comes in different shapes and sizes and allows an instant way to measure the heartbeat. Heartbeat Sensors are available in Wrist Watches (Smart Watches), Smart Phones, chest straps, etc. The heartbeat is measured in beats per minute or bpm, which indicates the number of times the heart is contracting or expanding in a minute.

CONCEPT OF THEORY:

The principle behind the working of the Heartbeat Sensor is Photoplethysmograph. According to this principle, the changes in the volume of blood in an organ is measured by the changes in the intensity of the light passing through that organ. Usually, the source of light in a heartbeat sensor would be an IR LED and the detector would be any Photodetector like a Photodiode, an LDR (Light Dependent Resistor) or a Phototransistor. With these two i.e., a light source and a detector, we can arrange them in two ways: A Transmissive Sensor and a Reflective Sensor.

LITERATURE REVIEW:

The circuit design of the Arduino based Heart rate monitor system using Heart beat Sensor is very simple. First, in order to display the heartbeat readings in bpm, we have to connect a 16×2 LCD Display to the Arduino UNO. The 4 data pins of the LCD Module (D4, D5, D6 and D7) are connected to Pins 1, 1, 1 and 1 of the Arduino UNO. Also, a 10KΩ Potentiometer is connected to Pin 3 of LCD (contrast adjust pin). The RS and E (Pins 3 and 5) of the LCD are connected to Pins 1 and 1 of the Arduino UNO.

TOPIC SUMMARY:

The working operation of the heartbeat and temperature sensor was illustrated in the form of a basic flowchart and block diagram. Those mechanisms were described as well as it's theoretical statements. The previous research or inventions found on the market were shown as an example for reference purposes.

METHODOLOGY OF RESEARCH:

Heartbeat and Temperature sensors will help reduce disease in the community. The data collected by many highly known doctors were used as our data for programming. The users will be able to monitor their heart rate so that they can keep track of any unexpected disease that might hit or get to them.

HARDWARE:

Nowadays, people face problems measuring temperature and heart beat frequently. People need to use a thermometer to measure temperature which takes time and for heart beat we have to go to the clinic or hospital. If you have one device to measure the temperature and heart rate frequently, it will be easier to keep the body hydrated and stable. Thus, this is one key to prevent heatstroke and to know about heart rate easily.

INTERFACE:

People are facing problems monitoring their body temperature and heart rate frequently. Most of the people are now busy with their work. As proven by articles in newspapers, states that most of the parents are not concerned about their children's body temperature and heart rate. Furthermore, it is also very hard for them to monitor their body temperature and heart beat when they are at the workplace and their children at school. So, people need a device that can monitor and record body temperature and heart beat at a distance. This will reduce time and cost.

SOFTWARE:

Nowadays, people facing problem to measure heart rate and temperature frequently. People need to go to a nearby clinic for a measure of heart rate and temperature which takes time. Thus, hydration is one of the keys to prevent heat stroke. Everyone needs to monitor their body temperature and heart rate from time to time so that, can prevent heat stroke. If they got one device to measure the temperature and heart rate from time to time it would be easier to keep their body hydrated from anywhere based on the Internet of Things (IoT).

OBJECTIVES

HARDWARE:

Fabrication of the heat stroke prevention wearable based in IOT Aim of this project is to conserve energy through the use of green technology.it can be used to store energy. which can be used later when sun is not available on a wet or cloudy day. Furthermore, to make a person's life easier by monitoring the temperature and heart beat by just looking at it will save time, cost.

INTERFACE:

IoT system for thermoregulatory homeostasis in the human body. Enhancing productivity, reducing costs and the automation of internal processes are also the aim of this project.This technology could replace humans who are in charge of monitoring and maintaining supplies.Moreover, the smart phone keeps a track both on the quality and the viability of things. It will reduce the number of heat stroke and cardiovascular diseases related patients.

SOFTWARE:

Alert system for heat stroke based on IoT. The objective is to prevent anyone from heat stroke from their smartphone. Besides that, it is portable and we no need to worry about the power source. However, there is also no limit to energy. Moreover, the connection between smartphone and user to monitor their body temperature is based on Internet of Things(IoT) which is known as interface connection. There is no body temperature wearable based on IoT in our country yet.

Benefits of temperature and heart beat sensor:

The increase of temperature on earth's surface in recent years has significantly affected the health of humans, where the concept of heat stroke has become a disturbing situation, especially if we consider the increase in deaths caused by this condition. The effects caused by problems related to high temperatures have been of interest for studies where the technology can have an important role for the solution of the same ones. The present work shows the design of a system employing temperature sensors using the IoT concept for detecting heat stroke early in everyone. The results show that the system proposed is efficient and practically usable in real life. You can find the Principle of Heartbeat Sensor, working of the Heartbeat & Temperature Sensor and Arduino based Heart Rate Monitoring System B using a practical heartbeat, Sensor. Monitoring heart rate is very important for athletes, patients as it determines the condition of the heart (just heart rate). There are many ways to measure heart rate and the most precise one is using Electrocardiography. But the easier way to monitor the heart rate is to use a Heartbeat Sensor. It comes in different shapes and sizes and allows an instant way to measure the heartbeat. The heartbeat is measured in beats per minute or bpm, which indicates the number of times the heart is contracting or expanding in a minute.

Concept of Internet of Things:

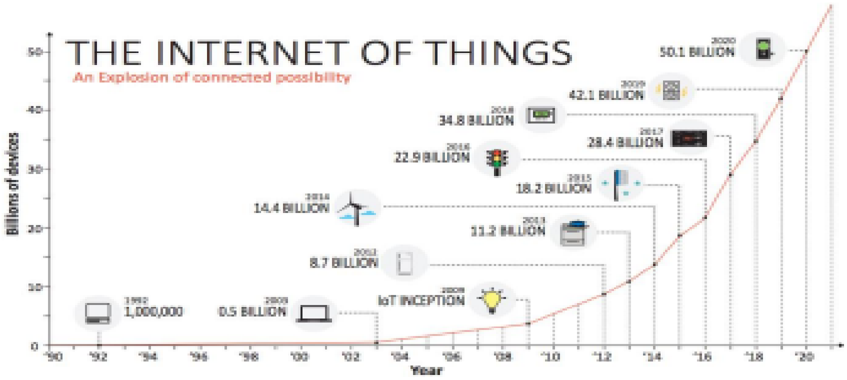
A recent study reported that the Internet of Things links people, devices and services [8]. Furthermore, the Internet of Things is more likely to be described as a system or framework. The Internet of Things is known as the connection between a network of things or objects and the Internet [5]. Internet of Things consist
of two words. That is the Internet and Things. The Internet depicts networks that are linked worldwide via some standard protocols. Moreover, The Things shows that any physical objects which have connectivity [5]. Based on Cisco statistics nearly 50 billion devices will be communicated to the Internet by 2020[2]. This turns IoT into reality [2]. The Internet of Things is gathering in importance with rising access to the Internet [1].

Impact of Internet of Things:

The Internet of Things has a higher effect on certain sectors. That are education, business, science, communication, humanity and government [10].It also proven that this service helps the community by reducing costs , increasing efficiency and enhancing the usability of existing systems [8].Some evidence proves that by using Internet of Things objects identify themselves and gain intelligence behavior by allowing related decisions things to fact that they can communicate information about themselves [10].Market Analyst determined that IoT devices would reach 25 billion by 2020[5].Corresponding to the report by marketresearch.com, market of Internet of Things in health care was estimated to reach 117 billion by 2020[5]. IoT devices will collect a very big amount of data on the way people communicate between each other.

<u>INTERNET OF THINGS STANDARDIZATIONS AND PROTOCOLS:</u>

It was estimated that by 2020 around 50 to 100 billion things will be connected electronically to the Internet [10]. The victory of IoT based on standardization, which provides interoperability, compatibility, reliability, and effective operations on a global scale whereas today more than 60 companies for leading technology, in communications and energy, working with standards, such as IETF, IEEE and ITU to specify new IP based technologies for the Internet of Things [10].



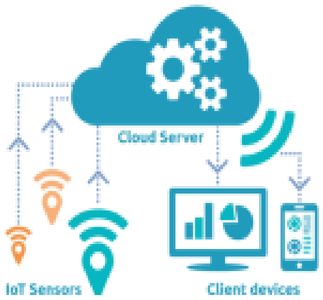
IoT and Healthcare Monitoring:

According to Forbes magazine, the IoT market will be more than 117 billion by 2020 [5]. Health care is one of the biggest sectors in Internet of Things technology. ioT wearable device monitor patient health related blood

pressure and body temperature [5]. Health care sector uses smart health sensors to analyze and collect data using gateways and analyzes through clouds and stores in clouds [10].

Cloud Computing:

It was certified that cloud computing can analyze and store data effectively . Cloud is the most convenient and cost-effective solution to deal with data produced by IoT and, in this respect, it generates new opportunities for data aggregation, integration, and sharing with third parties. Based on findings, the Internet of Things platform needs to be contributed by powerful computing sources. Therefore, cloud computing is suitable for the development of the Internet of Things platform .



Benefits of Using Internet of Things in Health Care System:

Data can be collected very accurately and precisely. This was because of usage of sensors

Early Detection of Chronic Disease Using Big data analytics it is possible to predict chronic disorder in the early stage and treatment can be done within a short period.

Concept of Alert System:

According to R. Wipfli and C. Lovis about Alert System . As already introduced, an alert can also be viewed as an interaction between user and the system. In fact, an alerting system can be reduced to the physician's interaction with the display, keyboard and mouse. In order to find a way to reduce errors on this cognitive level, we would like to refer to Rasmussen's model of decision making. According to his model, human decision making can be modeled as bottom-up problem identification and a top-down process of problem solving. In the context of this work, we would like to limit the problem identification. From a user's perspective the first step is to be alerted of the occurrence of an abnormal situation. This level of decision making addresses the skill based level of decision making, as it concerns automated processes.

Obstacles without Alert System for Heat stroke and Heart rate:

The impacts from work stress are one of the reasons why most people tend to forget . This clarifies that an individual's day by day life could influence their conduct and contribute to one's decision because of distressing and workload.

Visual Alert:

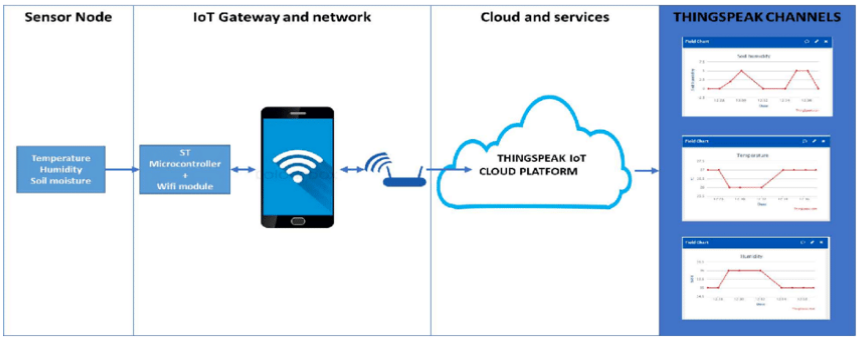
Based on the International Journal of Engineering and Computer Science in visual alert . ODAS provides visual alerts to the user by blinking LED lights and buzzers. Similar to audible alert, the frequency of blinking depends upon vehicle speed above threshold limit. But in our project research we are using Blynk Software which is an Android app for monitoring sensors or controlling electrical devices via Bluetooth, local WIFI or Internet.

Android Application Based Internet of Things (IOT).

There are separate login credentials for doctors and patients [13]. Here the Doctors can monitor health status of all the patients assigned to them while patients can only view their own health status by logging in using their respective login credentials while humidity status is accessible to both doctors as well as patients via their Android app based on IoT in smart phones.

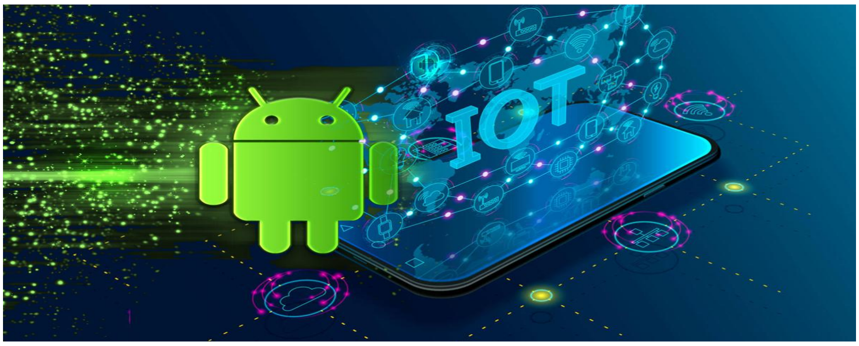
Thingspeak platform:

Thingspeak is not an app that works only with a particular shield. Instead, it's been designed to support the boards and shields you are already using. And it works on iOS and Android. UPD: Thingspeak also works over USB. Thingspeak Server is an Open-Source Netty based Java server, responsible for forwarding messages between Thingspeak mobile application and various microcontroller boards and SBCs (I.e. Arduino, Raspberry Pi)



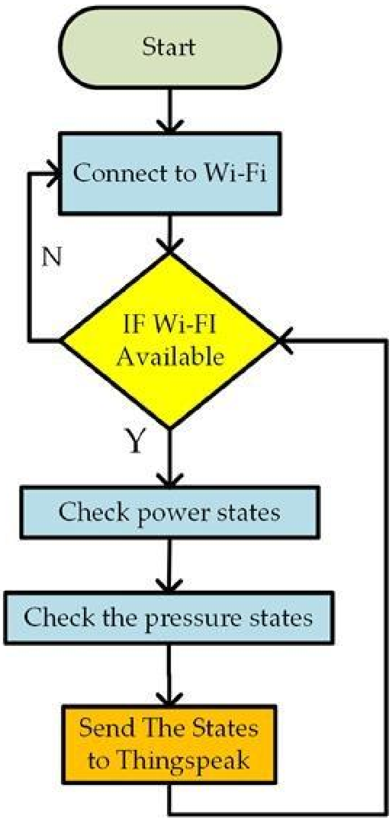
Android App Based on IoT:

There are separate login credentials for doctors and patients . Here the Doctors can monitor health status of all the patients assigned to them while patients can only view their own health status by logging in using their respective login credentials while humidity status is accessible to both doctors as well as patients via their Android app based on IoT in smart phones.



INTERFACE:

The methodology adopted for this sub project is by using IOT system.These devices, or things, connect to the network to provide information they gather from the environment through sensors, or to allow other systems to reach out and act on the world through actuators.A gateway enables devices that are not directly connected to the Internet to reach cloud services.Although the term gateway has a specific function in networking, it is also used to describe a class of device that processes data on behalf of a group or cluster of devices.

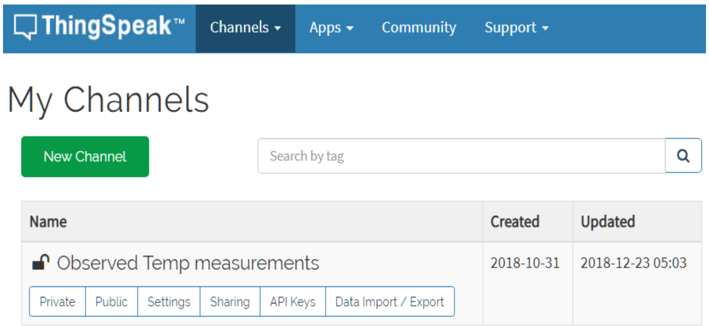


The data from each device is sent to Cloud Platform, where it is processed and combined with data from other devices.A gateway manages traffic between networks that use different protocols. A gateway is responsible for protocol translation and other interoperability tasks. An IOT gateway device is sometimes employed to provide the connection and translation between devices and the cloud. Because some devices don't contain the network stack required for Internet connectivity, a gateway device acts as a proxy, receiving data from devices and packaging it for transmission over TCP/IP.devices and packaging it for transmission over TCP/IP. According to its developers, Blynk is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. Blynk enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.Blynk is an Internet of Things (IoT) platform that lets you collect and store sensor data in the cloud and develop IoT applications. The Blynk™ IoT platform provides apps that let you analyze and visualize your data in MATLAB, and then act on the data.

Sensor data can be sent to Blynk from ArduinoFrom the software perspective, the application code running on the device maintains the source of truth.

THINGSPEAK:

Thingspeak was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.



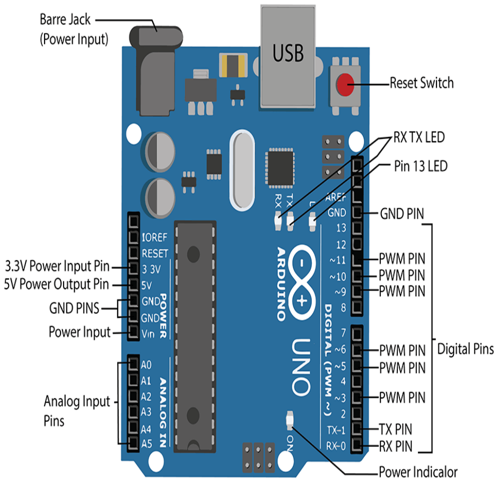
My Channels

Name	Created	Updated
Observed Temp measurements	2018-10-31	2018-12-23 05:03

COMPONENTS REQUIRED.

- **Arduino UNO - 1**
- **16 x 2 LCD display - 1**
- **330Ω Resistor (Optional – for LCD backlight)**
- **Heartbeat Sensor Module with Probe (finger based)**
- **Temperature sensor**
- **ESP8266 wifi module**
- **LED**
- **Breadboard**
- **Connectings wires**

1)Arduino UNO x 1



Defining Arduino:

An Arduino is actually a microcontroller-based kit which can be either used directly by purchasing from the vendor or can be made at home using the components, owing to its open source hardware feature. It is basically used in communications and in controlling or operating many devices.

- **Digital pins:** 14 (These pins have only 2 states i.e. high or low or in simple words either 5 V or 0 V no in between values. These pins are mostly used to sense the voltage presence when switch is open or close)
- **Analog pins:** 6 (A0 to A5 and they come up with a resolution of 10 bits and they provide flexibility of connecting any external device via these pins. These pins are configured from 0 V to 5 V but they can be configured to high range by using AREF pin or analogReference () function. ADC (analog to digital convertor) is used to sample these pins. These pins take analog signal and by using ADC convertor they convert this analog signal to number between 0 – 1023)

- 16 MHz crystal oscillator
- Out of 14 digital pins, 6 can be used for PWM (pulse width modulation)
- USB port
- TX and RX pins (for serial communication)
- Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its products are licensed under the

GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone.

- Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers can be programmed using C and C++ programming languages. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

How to use an Arduino Board?

The 14 digital input/output pins can be used as input or output pins by using pin Mode(), digital Read( ) and digital Write() functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current and has an internal pull-up resistor of 20- 50 K Ohms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite() function.

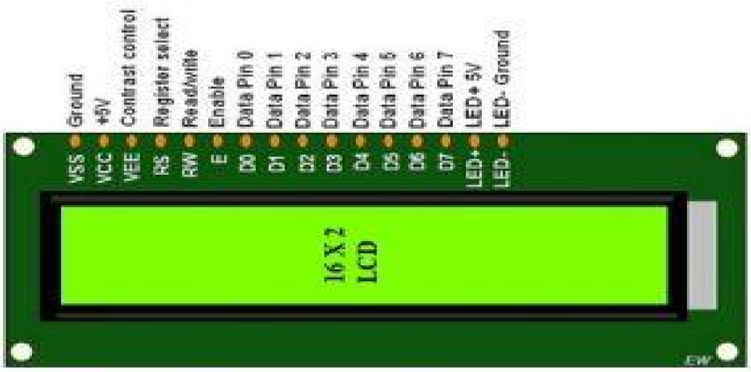
SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

In-built LED Pin 13: This pin is connected with a built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, it's off. Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts, but this limit can be increased by using AREF pin with analog Reference () function. Analog pin 4 (SDA) and pin 5 (SCA) are also used for TWI communication using the Wire library. Arduino Uno has a couple of other pins as explained below:

AREF: Used to provide reference voltage for analog inputs with analogReference() function.

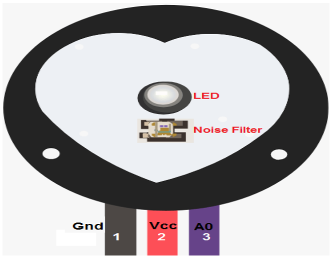
Reset Pin: Making this pin LOW, resets the microcontroller.

2)16 x 2 LCD Display x 1



A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly,[1] instead using a backlight or reflector to produce images in color or monochrome. [2] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and seven segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made from a matrix of small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the color of the backlight, and a character negative LCD will have a black background with the letters being of the same color as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance. A 16x2 LCD display is a very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means **it can display 16 characters per line and there are 2 such lines**. In this LCD each character is displayed in a 5x7 pixel matrix.

3)Heartbeat Sensor Module with Probe (finger based)

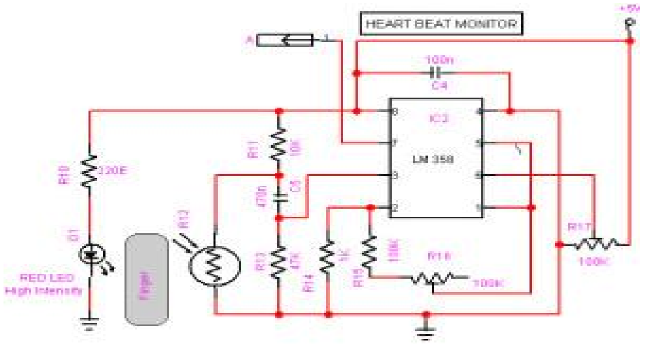


A heart rate monitor (HRM) is a personal monitoring device that allows one to measure/display heart rate in real time or record the heart rate for later study. It is largely used to gather heart rate data while performing various types of physical exercise. Measuring electrical heart information is referred to as Electrocardiography (ECG or EKG). Medical heart rate monitoring used in hospitals is usually wired and usually multiple sensors are used. Portable medical units are referred to as a Holter monitor. Consumer heart rate monitors are designed for everyday use and do not use wires to connect.

Principle of Heartbeat Sensor

The principle behind the working of the Heartbeat Sensor is Photoplethysmograph. According to this principle, the changes in the volume of blood in an organ is measured by the changes in the intensity of the light passing through that organ. Usually, the source of light in a heartbeat sensor would be an IR LED and the detector would be any Photodetector like a Photodiode, an LDR (Light Dependent Resistor) or a Phototransistor. With these two i.e. a light source and a detector, we can arrange them in two ways: A Transmissive Sensor and a Reflective Sensor. In a Transmissive Sensor, the light source and the detector are placed facing each other and the finger of the person must be placed in between the transmitter and receiver. Reflective Sensor, on the other hand, has the light source and the detector adjacent to each other and the finger of the person must be placed in front of the sensor.

PIN DIAGRAM:



4) TEMPERATURE SENSOR

definition:

The Temperature Sensor LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. A temperature sensor is a device, typically, a thermocouple or RTD, that provides for temperature measurement through an electrical signal. A thermocouple (T/C) is made from two dissimilar metals that generate electrical voltage in direct proportion to changes in temperature. An RTD (Resistance Temperature Detector) is a variable resistor that will change its electrical resistance in direct proportion to changes in temperature in a precise, repeatable and nearly linear manner.

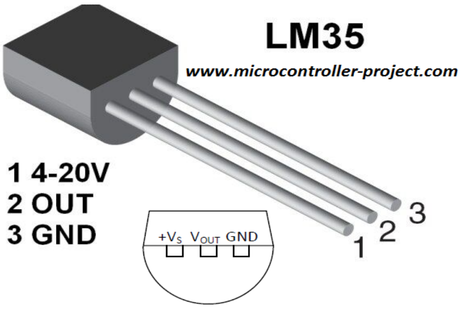
Principle of operation:

thermocouples:

A thermocouple is made from two dissimilar metal wires. The wires are joined together at one end to form a measuring (hot) junction. The other end, known as the reference (cold) junction, is connected across an electronic measurement device (controller or digital indicator). A thermocouple will generate a measurement signal not in response to actual temperature, but in response to a difference in temperature between the measuring and reference junctions. A small ambient temperature sensor is built into the electronic measuring device near the point where the reference junction is attached. The ambient temperature is then added to the thermocouple differential temperature by the measuring device in order to determine and display the actual measured temperature. Only two wires are necessary to connect a thermocouple to an electrical circuit; however, these connecting wires must be made from the same metals as the thermocouple itself. Adding wire made from other materials will create new measuring junctions that will result in incorrect readings.

RTDs

To greater or lesser degrees, all electrical conducting materials have some amount of resistance to the flow of electricity. When a known electric voltage passes through a conductor, the resistance varies based on the temperature of the conductor. This resistance can be measured and will correspond to a specific temperature. While various elements are affected by temperature in different ways, platinum is commonly used in an RTD due to its purity, linearity and stability over a wide range of temperatures. An electronic readout device, such as a controller or digital indicator designed to measure resistance, is required for use with RTD sensors.



5)ESP8266 WIFI MODULE:

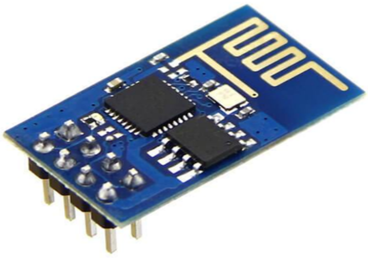
DEFINITION:

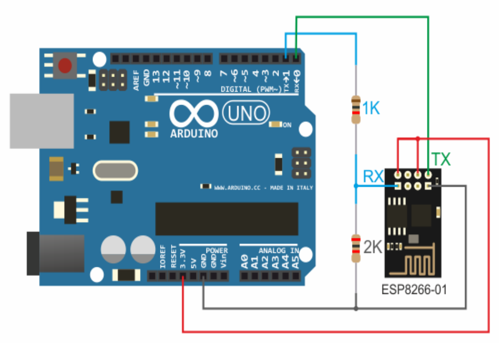
An ESP8266 Wi-Fi module is a SOC microchip mainly used for the development of end-point IoT (Internet of things) applications. It is referred to as a standalone wireless transceiver, available at a very low price. It is used to enable the internet connection to various applications of embedded systems.

ABOUT ESP8266:

Espressif systems designed the ESP8266 Wi-Fi module to support both the TCP/IP capability and the microcontroller access to any Wi-Fi network. It provides the solutions to meet the requirements of industries of IoT such as cost, power, performance, and design.It can work as either a slave or a standalone application. If the ESP8266 Wi-Fi runs as a slave to a microcontroller host, then it can be used as a Wi-Fi adaptor to any type of microcontroller using UART or SPI. If the module is used as a standalone application, then it provides the functions of the microcontroller and Wi-Fi network.

The ESP8266 Wi-Fi module is highly integrated with RF balun, power modules, RF transmitter and receiver, analog transmitter and receiver, amplifiers, filters, digital baseband, power modules, external circuitry, and other necessary components. The ESP8266 Wi-Fi module is a microchip shown in the figure below.



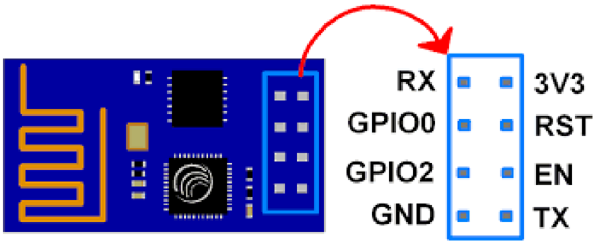


A set of AT commands are needed by the microcontroller to communicate with the ESP8266 Wi-Fi module. Hence it is developed with AT commands software to allow the Arduino Wi-Fi functionalities, and also allows loading various software to design the own application on the memory and processor of the module.

PIN CONFIGURATION AND DIAGRAM:

The processor of this module is based on the Tensilica Xtensa Diamond Standard 106 micro and operates easily at 80 MHz. There are different types of ESP modules designed by third-party manufacturers. They are,

- ESP8266-01 designed with 8 pins (GPIO pins -2)
- ESP8266-02 designed with 8 pins (GPIO pins -3)
- ESP8266-03 designed with 14 pins (GPIO pins- 7)
- ESP8266-04 designed with 14 pins (GPIO pins- 7)



The **ESP8266 Wi-Fi module pin configuration/pin diagram** is shown in the figure below. The ESP8266-01 Wi-Fi module runs in two modes. They are;

Flash Mode: When GPIO-0 and GPIO-1 pins are active high, then the module runs the program, which is uploaded into it.

UART Mode: When the GPIO-0 is active low and GPIO-1 is active high, then the module works in programming mode with the help of either serial communication or Arduino board.

ESP8266 Wi-Fi Module Specifications:

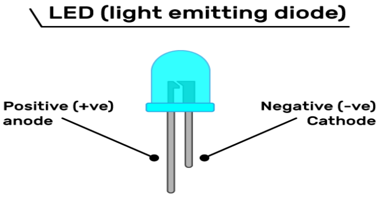
The **ESP8266 Wi-Fi module specifications or features** are given below.

- It is a powerful Wi-Fi module available in a compact size at a very low price.
- It is based on the L106 RISC 32-bit microprocessor core and runs at 80 MHz
- It requires only 3.3 Volts power supply
- The current consumption is 100 m Amps
- The maximum Input/Output (I/O) voltage is 3.6 Volts.
- It consumes 100 mA current
- The maximum Input/Output source current is 12 mA
- The frequency of built-in low power 32-bit MCU is 80 MHz
- The size of flash memory is 513 kb
- It is used as either an access point or station or both
- It supports less than 10 microAmps deep sleep
- It supports serial communication to be compatible with several developmental platforms such as Arduino
- It is programmed using either AT commands, Arduino IDE, or Lua script
- It is a 2.4 GHz Wi-Fi module and supports WPA/WPA2, WEP authentication, and open networks.
- It uses two serial communication protocols like I2C (Inter-Integrated Circuit) and SPI (Serial Peripheral Interface).
- It provides 10- bit analog to digital conversion
- The type of modulation is PWM (Pulse Width Modulation)
- UART is enabled on dedicated pins and for only transmission, it can be enabled on GPIO2.
- It is an IEEE 802.11 b/g/n Wi-Fi module with LNA, power amplifier, balun, integrated TR switch, and matching networks.
- Memory Size of instruction RAM – 32 KB
- The memory size of instruction cache RAM – 32 KB
- Size of User-data RAM- 80 KB

6)LED : (Light-emitting diode)

LEDs (Light Emitting Diodes) are the latest development in the lighting industry. Made popular by their efficiency, range of color, and long lifespan, LED lights are ideal for numerous applications including night lighting, art lighting, and outdoor lighting.

LED lighting products produce light up to 90% more efficiently than incandescent light bulbs. How do they work? **An electrical current passes through a microchip**, which illuminates the tiny light sources we call LEDs and the result is visible light.

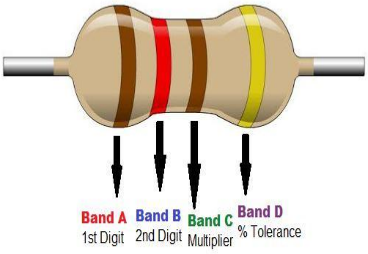


7)RESISTORS:

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

A resistor is a passive electrical component with the primary function **to limit the flow of electric current.**

The resistor's ability to reduce the current is called resistance and is measured in units of ohms (symbol: Ω). If we make an analogy to water flow through pipes, the resistor is a thin pipe that reduces the water flow.



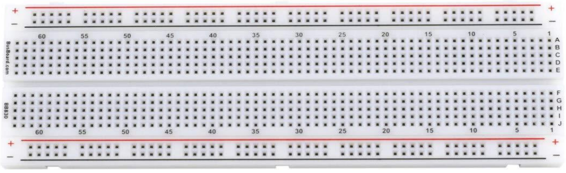
8)BREADBOARD:

DEFINITION:

The breadboard is the bread-and-butter of DIY electronics. Breadboards allow beginners to get acquainted with circuits without the need for soldering, and even seasoned tinkerers use breadboards as starting points for large-scale projects.

HOW IT IS WORKS:

If you are taking your first steps in the world of DIY or microcontrollers, you might have received a breadboard in your **Arduino starter kit** or **Raspberry Pi starter kit**. Let's look at what a breadboard actually is, where they came from, and how you can make use of them.



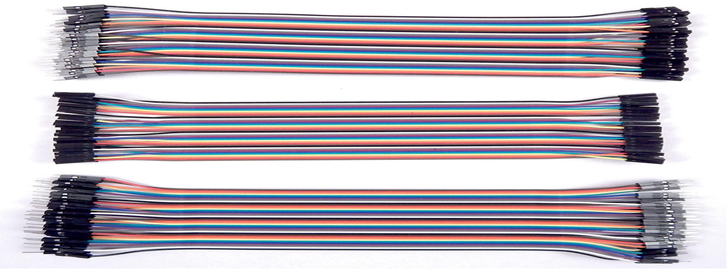
9)CONNECTING WIRES(JUMPER WIRES)

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.[1]

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

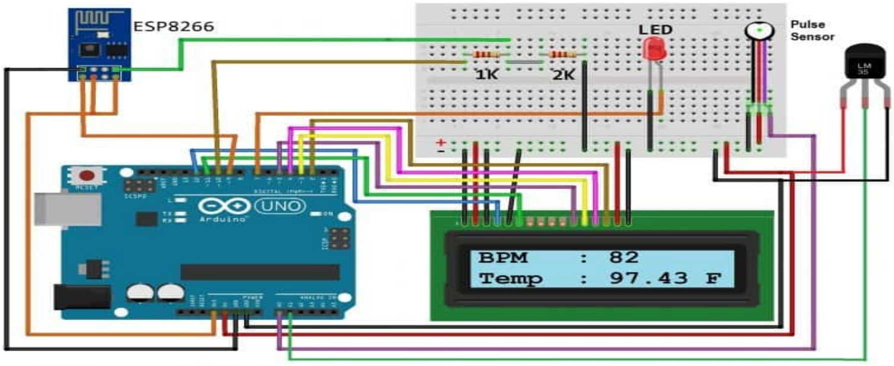
Types of Jumper Wires:

Jumper wires typically come in three versions: **male-to-male, male-to-female and female-to-female**. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into.



ARCHITECTURE OF PROJECT:

The block diagram of overall health monitoring system which is divided into two blocks i.e. Sender system consists of two sensors controlled by a microcontroller Arduino UNO, equipped with an SD Card module, LCD screen, ESP8266 module and a receiver system that consists of a database server and smartphones based on Android. Data of measuring results are sent to the server using wireless network (Wi-Fi) that operates at a frequency of 5 GHz. Data processing starts from sensors that are controlled by the microcontroller which then is sent to the Receiver system and displayed to the LCD screen and saved to the SD Card in the form of a text file that can be opened using Notepad ++ applications.



Pulse Rate Measuring Instrument

This system uses a Pulse Rate Finger Sensor using photoplethysmography technique to detect a pulse. This sensor detects the pulse rate signal using light as a pulse detector of blood volume changes in the network that is synchronized with the heartbeat. In other words, the pulse sensor is used to measure subtle changes in light of the expansion of capillaries to feel the heartbeat. This sensor is placed on the fingertip to measure pulse rate.

Body Temperature Measuring Instrument

LM35 is a temperature sensor that is used in this project. Its waterproof nature is suitable for the measurement of human body temperature and has a higher accuracy. This sensor was developed by Maxim Integrated, has an accuracy of ± 0.5 ◦C from -10 C to + 85◦C at room temperature, works at a variety of temperatures (−55◦C to + 125◦C) or (-67 ◦F to + 257◦F). LM35 has a 64-bit serial code that is unique, allowing multiple LM35 functions on the same 1-Wire bus. It also has a query time of less than 750ms and can be used at 3.0V.

Microcontroller Arduino

A Arduino UNO Microcontroller was selected as one of the main components of this system. ArduinoUNO is using ATmega328P chip, has a number of I/O pins i.e. 14 pieces of I/O digital pin (6 pins of which are PWM), 6 analog input pins, 2 pins UART. Arduino UNO is equipped with a 8 MHz oscillator.

LCD Screen

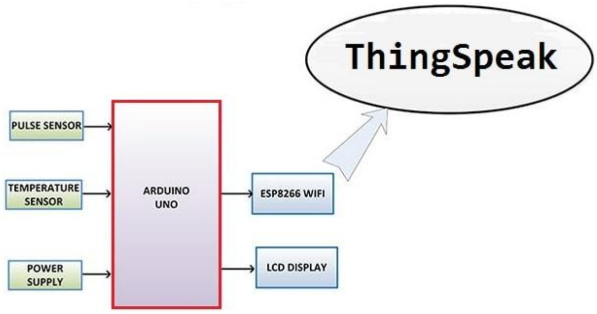
The display module is lightweight, low power, small and easy to use via I2C. It can operate in the 3.3 to 5.0 V power supply. It has a screen resolution of 128 x 64 pixels, I2C interface, dimensions 0.96 x 0.75.

ESP8266 Module

The ESP8266 Wi-Fi Module is used to connect the Arduino board with a Wi-Fi router so that it can access the cloud. It is a self-contained SOC with an integrated **TCP/IP protocol** stack that can access to a Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all **Wi-Fi** networking functions from another application processor. Each ESP8266 module comes pre-programmed with an **AT command** set firmware. The module comes available in two models – ESP-01 and ESP-12. ESP-12 has 16 pins available for interfacing while ESP-01 has only 8 pins available for use.

Schematic

Schematic is an interconnection illustration of circuits between electronic components with Arduino Microcontroller pins. The Components connected to the Arduino are two sensors, module SD Card, LCD and Wireless Modules as shown in Figure



WORKING:

WORKING OF HEARTBEAT SENSOR:

A simple Heartbeat Sensor consists of a sensor and a control circuit. The sensor part of the Heartbeat Sensor consists of an IR LED and a Photo Diode placed in a clip. The Control Circuit consists of an Op-Amp IC and few other components that help in connecting the signal to a Microcontroller. The working of the Heartbeat Sensor can be understood better if we take a look at its circuit diagram.

The above circuit shows the finger type heartbeat sensor, which works by detecting the pulses. Every heartbeat will alter the amount of blood in the finger and the light from the IR LED passing through the finger and thus detected by the Photodiode will also vary.

The output of the photodiode is given to the non – inverting input of the first op – amp through a capacitor, which blocks the DC Components of the signal. The first op – amp acts as a non – inverting amplifier with an amplification factor of 1001.

Working on this project is quite easy but a little calculation for calculating heart rate is required. There are several methods for calculating heart rate, but here we have read only five pulses. Then we have calculated total heart beat in a minute by applying the below formula:

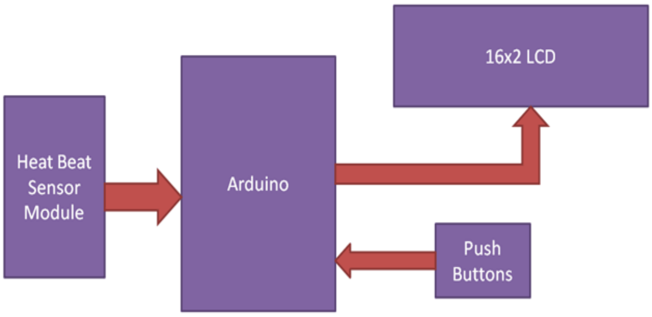
Five_pusle_time=time2-time1;

Single_pulse_time= Five_pusle_time /5;

rate=60000/ Single_pulse_time;

where time1 is first pulse counter value time2 is list pulse counter value rate is final heart rate.

When the first pulse comes, we start the counter by using a timer counter function in the arduino that is millis();. And take the first pulse counter value form millis();. Then we wait for five pulses. After getting five pulses we again take counter value in time2 and then we subtract time 1 from time to take the original time taken by five pulses. And then divide this time by 5 times for getting single pulse time. Now we have time for a single pulse and we can easily find the pulse in one minute, dividing 600000 ms by single pulse time. Rate= 600000/single pulse time.



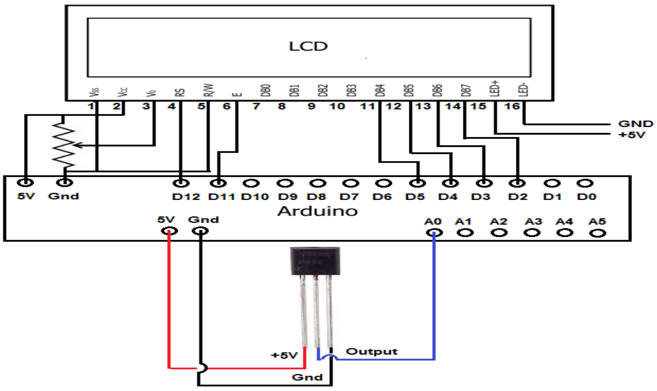
In this project we have used the Heart beat sensor module to detect Heart Beat. This sensor module contains an IR pair which actually detects heartbeat from blood. Heart pumps the blood in the body which is called heartbeat, when it happens the blood concentration in the body changes. And we use this change to make a voltage or pulse electrically.

WORKING OF TEMPERATURE SENSOR:

The working of the project starts with the LM35 sensor that senses the change in temperature of the surrounding, and uses that temperature difference to produce a voltage signal which is processed by the Arduino to give a digital output displaying the temperature of the given surrounding**.**

LM35 is an analog, linear temperature sensor whose output voltage varies linearly with change in temperature. LM35 is a three terminal linear temperature sensor from National semiconductors. It can measure temperature from*-55 degree celsius to +150 degree celsius.* The voltage output of the LM35 increases 10mV per degree Celsius rise in temperature. LM35 can be operated from a 5V supply and the standby current is less than 60uA. So that's all the info you need about LM35 for this particular temperature display project using arduino uno. So let's get to LM35 temperature sensor interfacing with arduino!

We are using Arduino Uno as our board and LM35 can be connected to arduino as shown in circuit diagram.



This is a prototype model for an IoT based pulse rate monitor. It can be designed as a wearable watch or earplug. In a wearable design, the character LCD could be removed and the entire circuit can be shifted to a small controller board or SOC.

When the circuit is powered by the battery, the Arduino starts reading the pulse rate from the pulse sensor and the ambient temperature from the LM-35 temperature sensor. The pulse sensor has an infrared LED and a phototransistor which help detect the pulse at the tip of the finger or earlobe. Whenever it detects a pulse, its IR LED flashes. The flash of the IR LED is detected by the phototransistor and its resistance changes when the pulse is changed. The heartbeat of a normal adult ranges from 60 to 100 per minute. For detecting beats per minute (BPM), first, an interrupt is set which triggers in every 2 Milliseconds. So, the sampling rate by the Arduino to detect pulse is 500 Hz. This sampling rate is sufficient to detect any pulse rate. So, at every 2 Milliseconds, the Arduino reads analog voltage output from the pulse sensor. The analog output from the pulse sensor is converted to a digital value using an in-built ADC channel. The Arduino has a 10-bit long ADC channel, so the digitized value can range from 0 to 1024. The middle value for this range is 512. Initially, the first beat is set to true and the second beat is counted when the condition that analog output from the pulse sensor is greater than the middle point i.e. 512 is satisfied. Then, onwards, every next beat is counted when the analog output from the pulse sensor is greater than the middle point i.e. 512 and 3/5 of the time between the beats recorded in the previous cycle has passed. Every time the beat is detected, a variable representing BPM is updated. This value in this variable is pushed to an array every minute and is used to represent the actual Beats Per Minute or Heart Rate. The Arduino code also uses a function to provide an LED fading effect on every beat. The pulse sensor can also detect body temperature. The LM-35 is used to detect the surrounding temperature here. The operating temperature range of LM-35 is from -55 °C to 150 °C. The output voltage varies by 10 mV in response to every oC rise/fall in ambient temperature, i.e., its scale factor is 0.01 V/ oC. The LM-35 IC does not require any external calibration or trimming to provide typical accuracies of ±0.25 °C at room temperature and ±0.75 °C over the temperature range from −55 °C to 150 °C. Under normal conditions, the temperature measured by the sensor won't exceed or recede the operational range of the sensor. Typically in the temperature range from −55 °C to 150 °C, the voltage output of the sensor increases by 10 mV per degree Celsius. The voltage output of the sensor is given by the following formulae –

Vout = 10 mV/°C*T

where,

Vout = Voltage output of the sensor

T = Temperature in degree Celsius

So, T (in °C) = Vout/10 mV

T (in °C) = Vout(in V)*100

If VCC is assumed to be 5 V, the analog reading is related to the sensed voltage over 10-bit range by the following formulae –

Vout = (5/1024)*Analog-Reading

So, the temperature in degree Celsius can be given by the following formulae –

T (in °C) = Vout(in V)*100

T (in °C) = (5/1024)*Analog-Reading *100

So, the temperature can be measured directly by sensing the analog voltage output from the sensor. The analogRead() function is used to read analog voltage at the controller pin.

The Arduino collects data from both the sensors and converts the values to strings. The heartbeat is graphically represented on the character LCD along with the measured pulse rate and time between pulses as text. The temperature is also displayed on the LCD module.

The ESP8266 Wi-Fi module connected to the Arduino uploads the same data to ThingSpeak Server as it finds the Wi-Fi Access Point. For displaying and monitoring data uploaded to the ThingSpeak server, either a digital dashboard or a data broker is needed. In this project, a digital dashboard called Freeboard.io is used to monitor the sensor data visually online. Freeboard.io uses the JASON file to visualize ThingSpeak data.

CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7 , 6 , 5, 4, 3, 2);
#include <SoftwareSerial.h>
float pulse = 0;
float temp = 0;
SoftwareSerial ser(10,11);
String apiKey = "M7F1BFHIQE7JCKS8";
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 not a "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,
ser.println("AT");
delay(1000);
ser.println("AT+GMR");
delay(1000);
ser.println("AT+CWMODE=1");
delay(1000);
ser.println("AT+RST");
delay(5000);
ser.println("AT+CIPMUX=1");
delay(1000);
String cmd="AT+CWJAP=\"*SSID*\",\"*PASSWORD*\"";
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
{
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);
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
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 realistic 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
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 you're 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 + 62;
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("C");
}
```

RESULTS AND ANALYSIS:

In preparing the final project, analysis and discussion are carried out to ensure the project results are in the desired state without causing any problems. Project analysis is performed to determine, select and produce an optimal design in terms of material usage, cost and perfect manufacturing methods. The analysis is more focused on project characteristics, problems encountered and the cost of materials used such as the price of raw materials used and experiments conducted and other items involved.

1. Sensor work and detect the heartbeat of the user.

2. Wifi modulator able to connect with phone via THINGSPEAK

3. temperature sensor working

4. Arduino has a good performance.

5. LCD can display the reading very well.

6. Reading was detected every 60 sec.

Through this methodological research we were able to understand the demand or the hype for the system and the potential of the heartbeat and temperature sensor among today's society.

CONCLUSION:

From this project, it was to encourage the self-expression of thoughts and feelings related to illness/hospitalization. And also, help patients process and work through traumatic experiences associated with hospitalization. Next to facilitate positive self esteem and positive body image, and promote a sense of independence and feelings of control. Encourage the development of healthy strategies for coping with hospitalization. The main result of this project is to help to measure the temperature to prevent heat stroke among the people.

Most of the rehab process depends on the feeling of the patient and judgement by the doctor. Sometimes the judgement could be wrong. With this project it got to resolve this problem and give accurate rehab level things to patients. The resulting system was also low in power and cost, non - invasive, and provided real time monitoring. It is also easy to use and provides accurate measurements.