

IOT BASED HEART ATTACK DETECTOR

Project Report



Team Details:-

Taha Junaid 18BCD7072

Mohd Taquiddin 18BCN7050

Darrain Saa-eyd 18BCN7009

Jivitesh Yadav 18BME7016

Kotha Kowshik Kumar 18MIS7204

Shoaeb Nawab Shaik 18BEC7067

Guided by,

Prof. Saroj Kumar Panigrahy

Dept. of CSE

**Abstract**

* Heart rate monitoring is a vital aspect of maintaining heart health. People from different age groups have different ranges for maximum and minimum values of heart rate, the monitoring system must be compatible enough to tackle this scenario.
* In this paper, an IoT based system has been implemented that can monitor the heartbeat from the output given by a hardware system pulse sensor. Further, an alert system is added which is executed if the heartbeat goes below or above the permissible level given in the devised algorithm.
* The alert message is received by the doctor through a mobile phone application. By using this prototype, the doctors can access the heartbeat data of the patient from any location. The nurses or the duty doctor available at the hospital can monitor the heart rate of the patient in the serial monitor through the real-time monitoring system.
* The heartbeat data and other personal details of the patient are stored in the cloud, this can be utilized for future studies on the health condition of the patient.

**Index**

* **Introduction…………………………………………………...**
* **Background……………………………………………………**
* **Methods………………………………………………………...**
* **Objective……………………………………………………….**
* **Results and Discussion………………………………………...**
* **Conclusion and Future scope………………………………….**
* **References………………………………………………………**
* **Codes in Appendix……………………………………………...**

**Introduction**

The heart is one of the most important organs in the human body. It acts as a pump for circulating oxygen and blood throughout the body, thus keeping the functionality of the body intact. A heartbeat can be defined as a two-part pumping action of the heart which occurs for almost a second. It is produced due to the contraction of the heart. When blood collects in upper chambers, the SA(Sinoatrial) node sends out an electrical signal which in turn causes the atria to contract. This contraction then pushes the blood through tricuspid and the mitral valves; this phase of the pumping system is called diastole. The next phase begins when the ventricles are completely filled with blood. The electrical signals generating from SA node reach the ventricle and cause them to contract. This phase of the pumping system is called systole. The tricuspid and mitral valves are closed tightly to prevent the backflow of blood; the pulmonary and aortic valves are opened. Once the blood moves from the pulmonary artery and aorta the ventricles relax and the pulmonary and aortic valves close. Tricuspid and mitral valves open because of the lower pressure from the ventricles leading to the start of another cycle. In today's scenario, health problems related to heart are very common. Heart diseases are one of the most important causes of death among men and women; it claims approximately 1 million deaths every year. Heart rate is a critical parameter in the functioning of the heart. Therefore, heart rate monitoring is crucial in the study of heart performance and thereby maintaining heart health.

This paper proposes a heart rate monitoring and abnormality detection system using IoT. Nowadays treatment of most of the heart-related diseases requires continuous as well as long term monitoring. IoT is very useful in this aspect as it replaces the conventional monitoring systems with a more efficient scheme, by providing critical information regarding the condition of the patient accessible by the doctor in any remote place, at any time through the internet. In addition, the nurses or the duty doctor available at the hospital can monitor the heart rate of the patient in the serial monitor through the real-time monitoring system. Also, a warning system is incorporated in which if the patient’s heartbeat goes below or exceeds a particular value the doctor receives an alert message through a mobile application. Early recognition of the disease is very vital in preventing more complications in the future.

The suggested prototype consists of both hardware and software components. The hardware consists of pulse sensor, and LCD display. The software consists of two IoT platform along with a mobile application. The system is based on a portable heart rate monitoring system designed in a cost-efficient manner. The prototype is also easy to use and access the data. And also can be used by people of different age groups. The real-time data can be viewed as well as stored for future studies with respect to the heart condition of the patient.

**Background**

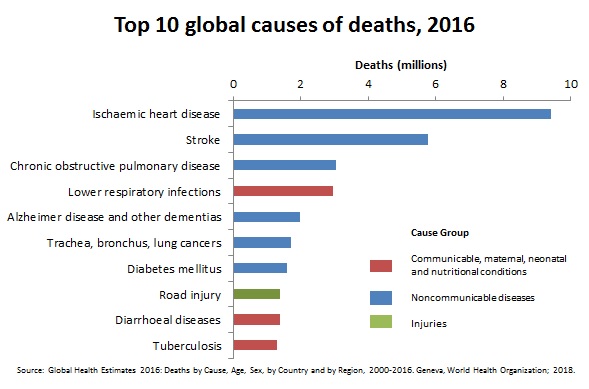
As the amount of elderly people and chronic diseases patients grow rapidly, drawbacks of traditional healthcare service are increasingly prominent. The Internet of Things (IoT) is intercommunication of embedded devices using networking technologies. Elderly people need to make regular visit to the doctor for their health signs test results. Observing on regular basis of essential signs is compulsory as they are main signs of well-being of one’s individual health. These vital signs include, Pulse rate and Body temperature. The goal is to develop a low power, more reliable, non-intrusive, are the essential signs monitor which gather information on the body and send the parameters through wireless technology. With the assistance of some equipment units, different sensors and gadgets with web association. The system functionality is divided into major three modules; they are:

1) Sensing module

2) The Main module

3) Interaction module.

The Pulse Sensor and Temperature Sensor of sensing module should extract the accurate readings and should be able to send the data to the Arduino. The Wi-Fi module which is also a part of the sensing module must send the values to the server without any delay and without any data loss. The Server must store all the data sent by the Wi-Fi module and display the same on the Web Server. Developed a system that measures and detect Human-Heartbeat and body temperature of the patient, sends the data to user or server end by using microcontroller with reasonable cost and great effect. Use two different sensors and these are mainly under the control of microcontroller. For Human-Heartbeat measurement use fingertip, it’s in bpm (beats per minute). These calculated rates will have stored in server by transferring through Wi-Fi module via internet. liquid crystal display (LCD) has been used to display the calculated human-heart beat rate. Finally, the stored data in server will be displayed for further analysis by physician or specialist to provide better aid. Heart attacks stand as a leading cause of deaths since years. Most of the times, patients are late to reach the emergency room during heart attacks since detection takes time.



**Methods**

**Sensing Module:**

Sensing module consists of some sub-modules called sensors node and brain node. Pulse sensor is attached to the patient’s body for perceiving health data from the patient. In other words, these sensors collect the readings from the patient. The Sensors which are attached to the patient’s body are then interfaced to the Arduino. And readings are transmitted to brain node through gate-way. This Gateway acts as the intermediate between the sensors and the brain node. The sensors which collect the readings store the values in their flash memory. This stored data is transmitted to the brain node at regular intervals, the data is sent only at regular intervals because during this time intervals the values obtained which can either analog or digital will be converted into best suitable for human understanding. All collected data are stored into brain node. The data stored in the brain node is then sent to the main module by using the Wi-Fi module interfaced to the Arduino After some time the brain node transmits a set of data to main module. In the sensing module, all the sensors are interfaced to the Arduino. The sensors being

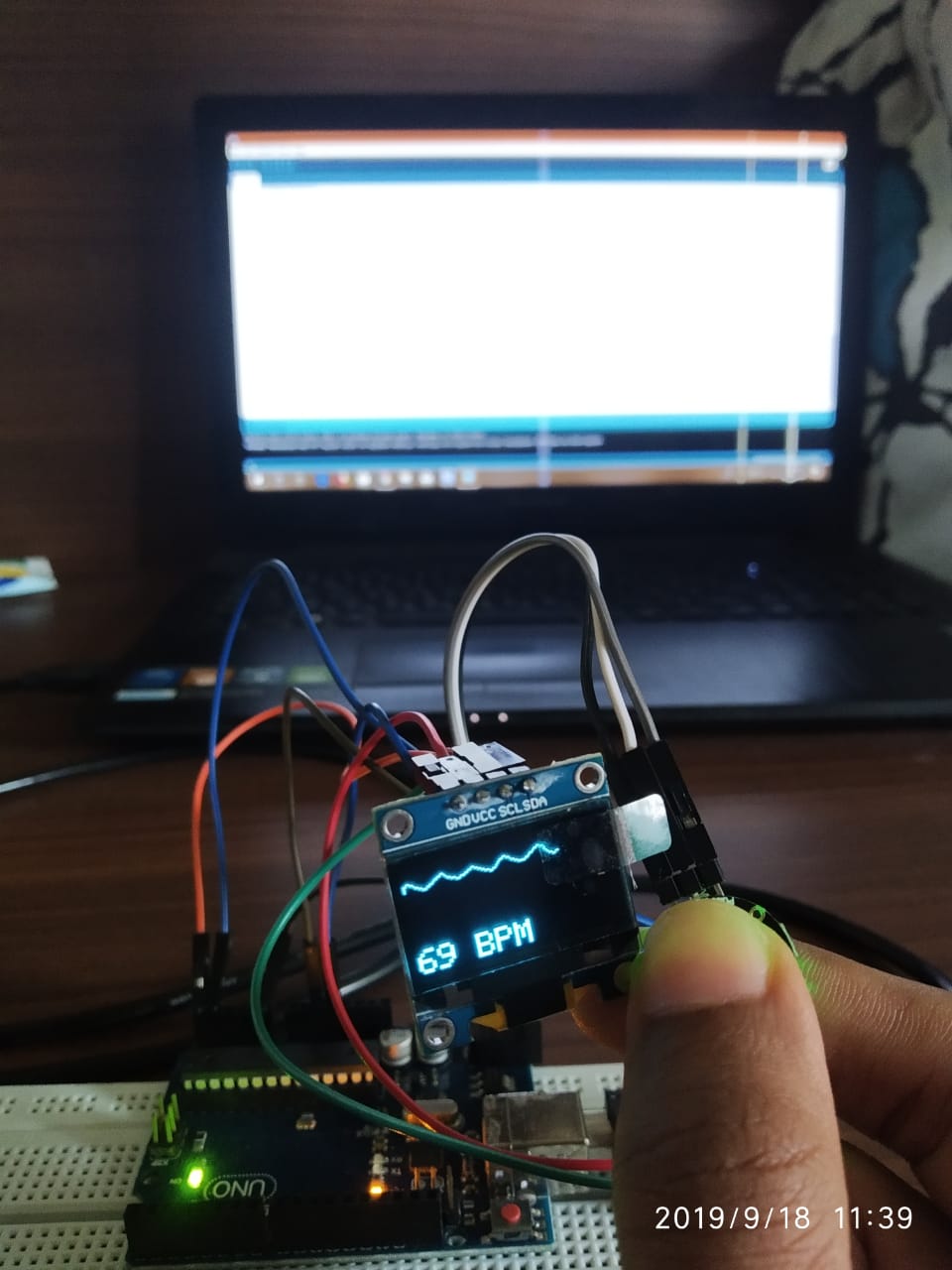
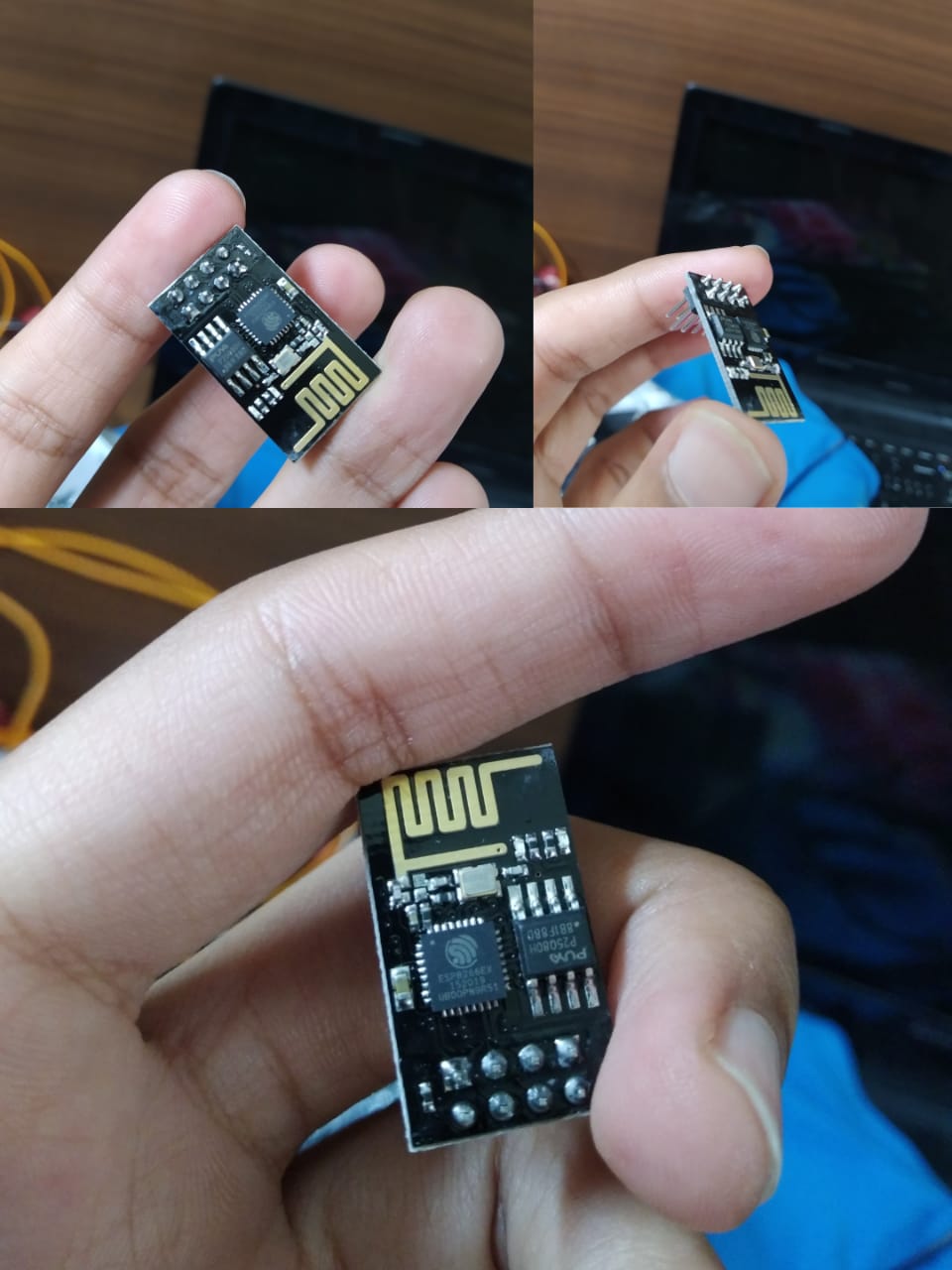
• Pulse Sensor

• Wi-Fi Module

Pulse Sensor Amped is a plug-and-play heart-rate sensor for Arduino compatibles. Pulse Sensor Amped works with either a 3V or 5V. There are 4 steps involved in interfacing the pulse sensor to the Arduino and to the computer.

**Wi-Fi Module**

Connect ESP8266 module with Arduino. ESP8266 runs on 3.3V, so need to power from the 3.3V output of Arduino. Connect VCC and CH\_PD of ESP8266 module to 3.3V output of Arduino and the ground of ESP8266 to the ground of Arduino. The RX pin of the ESP8266 is not 5V tolerant so, need to reduce the 5V TX output of Arduino to 3.3V using voltage dividing resistors. Here, using three 1KΩ resistors connected in series for that. So, connect pin 10 (TX) of Arduino to RX of ESP8266 module via voltage dividing resistors. Can directly connect TX pin of ESP8266 to 9th (RX) of Arduino as it will detect 3.3V as logic HIGH according to TTL Voltage specification



**Pulse sensor**

Heart rate data can be really useful whether you’re designing an exercise routine, studying your activity or anxiety levels or just want your shirt to blink with your heart beat. The problem is that heart rate can be difficult to measure. Luckily, the Pulse Sensor Amped can solve that problem The Pulse Sensor Amped is a plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects. It essentially combines a simple optical heart rate sensor with amplification and noise cancellation circuitry making it fast and easy to get reliable pulse readings. Also, it sips power with just 4mA current draw at 5V so it’s great for mobile applications. Simply clip the Pulse Sensor to your earlobe or fingertip and plug it into your 3 or 5 Volt Arduino and you’re ready to read heart rate! The 24" cable on the Pulse Sensor is terminated with standard male headers so there’s no soldering required. Of course, Arduino example code is available as well as a Processing sketch for visualizing heart rate data

**Objective**

In this project we are implementing a heartbeat monitoring and heart attack detection system using the Internet of things. These days we have an increased number of heart diseases including increased risk of heart attacks. The sensor is then interfaced t o a microcontroller that allows checking heart rate readings and transmitting them over internet. The user may set the high as well as low levels of heart beat limit. After setting these limits, the system starts monitoring and as soon as patient heart beat go es above a certain limit, the system sends an alert to the controller which then transmits this over the internet and alerts the doctors as well as concerned users. Also, the system alerts for lower heartbeats. Whenever the user logs on for monitoring, the system also displays the live heart rate of the patient. Thus concerned ones may monitor heart rate as well get an alert of heart attack to the patient immediately from anywhere and the person can be saved on time.

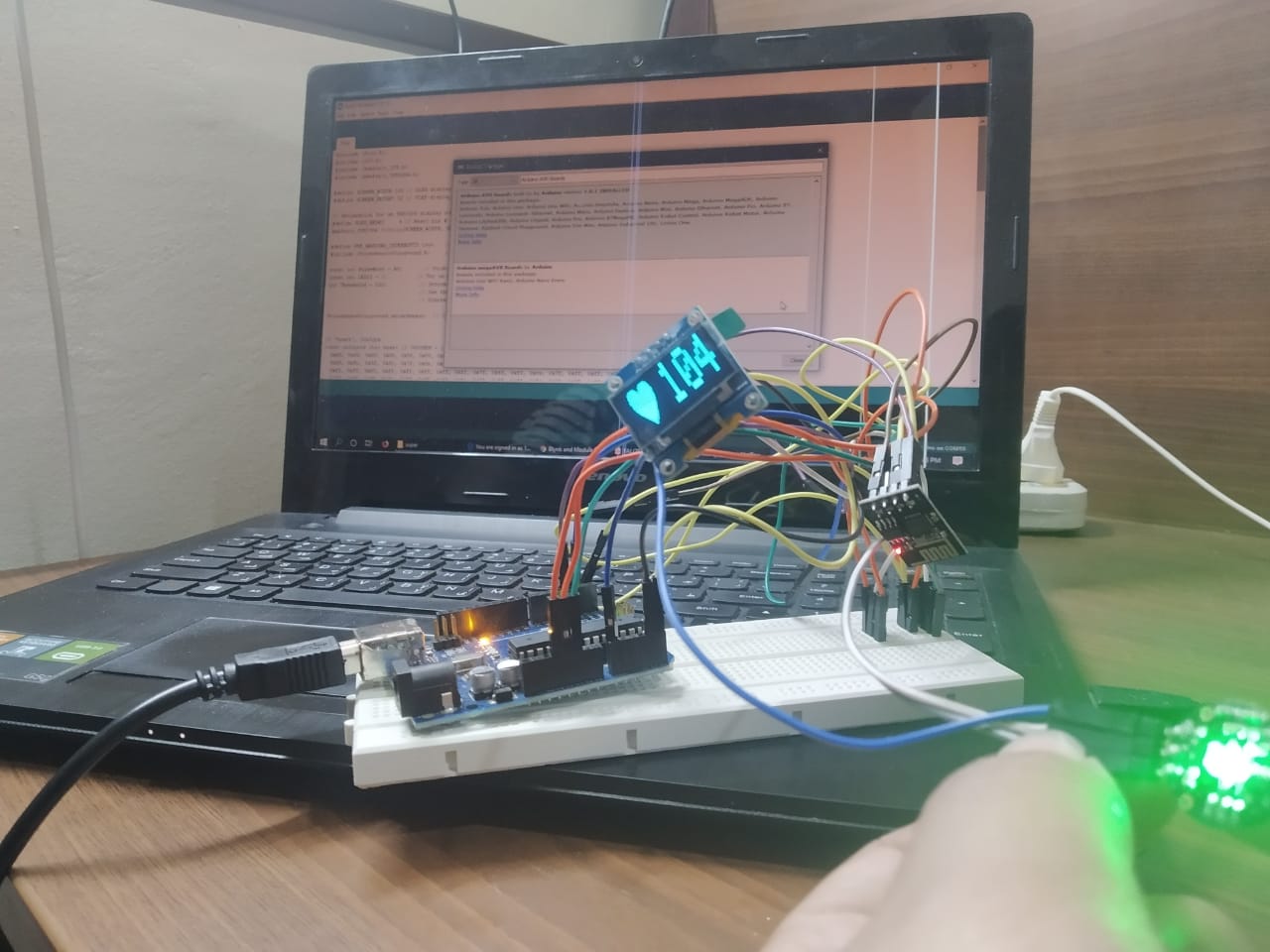
**Results and Discussion**

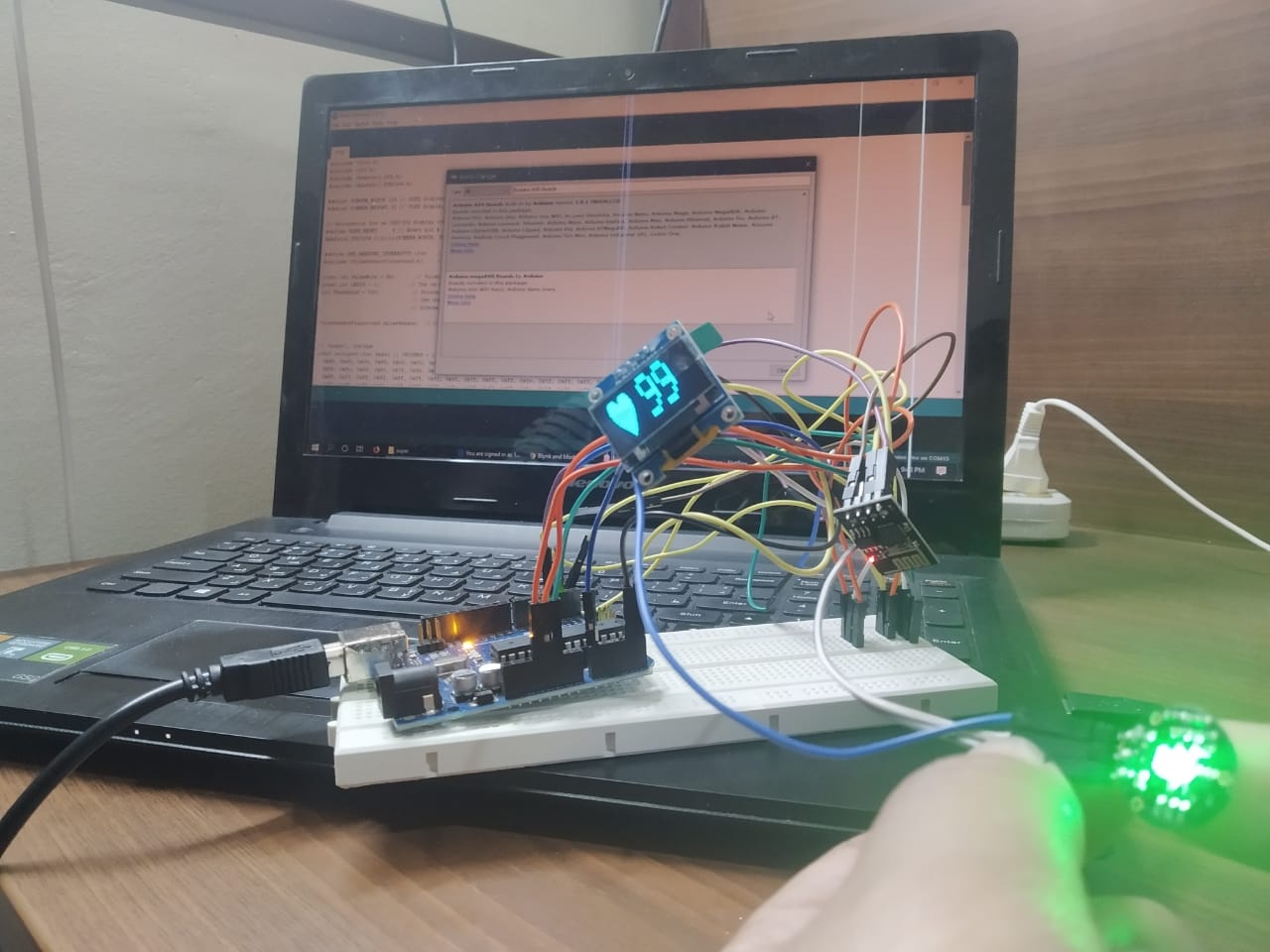
The frequency of the cardiac cycle is described by the heart rate, which is typically expressed as beats per minute. Each beat of the heart involves five major stages. The first two stages, often considered together as the "ventricular filling" stage, involve the movement of blood from the atria into the ventricles. The next three stages involve the movement of blood from the ventricles to the pulmonary artery (in the case of the right ventricle) and the aorta (in the case of the left ventricle). The first stage, "diastole," is when the semilunar valves (the pulmonary valve and the aortic valve) close, the atrioventricular (AV) valves (the mitral valve and the tricuspid valve) open, and the whole heart is relaxed. The second stage, "atrial systole," is when the atrium contracts and blood flows from atrium to the ventricle.

This works presents a lot of considerations and improvements that were incorporated in to the functionality of the device so as to reflect desired features such as cost, design complexity, size, software development, weight, lack of portability etc. This design uses a miniaturized pulse sensor (IC sensor) which has been optimized for very accurate sensing and measurement of changes in the heartbeat rate. The system calculates the heartbeat rate in beat per minute (BPM) with the help of the microcontroller, displays the measured heart rate on a and sends the result to the pc which is interfaced with the Arduino board by using the compiler, each time the heart rate goes above or below a fixed threshold, the result is displayed in the pc and from the pc the result can be trans mitted to the android devices via internet

**Conclusion and future scope**

In These days we have an increased number of heart diseases including increased risk of heart attacks. Our proposed system uses sensors that allow to detect heart rate of a person using heartbeat sensing even if the person is at home. The sensor is then interfaced to a microcontroller that allows checking heart rate readings and transmitting them over internet. The user may set the high as well as low levels of heart beat limit. After setting these limits, the system starts monitoring and as soon as patient heart beat goes above a certain limit, the system sends an alert to the controller which then transmits this over the internet and alerts the doctors as well as concerned users. Also, the system alerts for lower heartbeats. Whenever the user logs on for monitoring, the system also displays the live heart rate of the patient. Thus, concerned ones may monitor heart rate as well get an alert of heart attack to the patient immediately from anywhere and the person can be saved on time. In our proposed research, we tried to propose a complete paper for detecting heart attack using two ways. However, we have some plan about this research. Time of India, a leading newspaper in India published that “Researchers in the United States, within the next decade Heart Microeconomic Microchip will be set in blood vessel of human body. The smart phone will collect data and send the information to us”. Researchers are trying to implement the requirements of Microchip for uses of the technology in smart phone. We will try to use this technology in future. If this technology will be developed then we can detect heart blockage through this technology by our project.

****

****

**References**

<http://academicscience.co.in/admin/resources/project/paper/f201805011525176215.pdf>

<https://www.ijrte.org/wp-content/uploads/papers/v7i6s5/F13020476S519.pdf>

<https://www.hindawi.com/journals/ahci/2019/1507465/>

<http://ijsrd.com/Article.php?manuscript=IJSRDV6I20354>

**Codes in appendix**

#include <Wire.h>  
#include <SPI.h>  
#include <Adafruit\_GFX.h>  
#include <Adafruit\_SSD1306.h>  
  
#define SCREEN\_WIDTH 128 // OLED display width, in pixels  
#define SCREEN\_HEIGHT 32 // OLED display height, in pixels  
  
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)  
#define OLED\_RESET     4 // Reset pin # (or -1 if sharing Arduino reset pin)  
Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);  
  
#define USE\_ARDUINO\_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.  
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.    
  
const int PulseWire = A0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 1  
const int LED13 = 1;          // The on-board Arduino LED, close to PIN 13.  
int Threshold = 520;           // Determine which Signal to "count as a beat" and which to ignore.  
                               // Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.  
                               // Otherwise leave the default "550" value.  
                                 
PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called "pulseSensor"  
  
  
// 'heart', 32x32px  
const unsigned char heart [] PROGMEM = {  
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xf0, 0x0f, 0xf0, 0x3f, 0xf8, 0x3f, 0xfc,  
  0x3f, 0xfc, 0x3f, 0xfc, 0x7f, 0xfe, 0x7f, 0xfe, 0x7f, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xfe,  
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xfe,  
  0x7f, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xfe, 0x3f, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xfc,  
  0x1f, 0xff, 0xff, 0xf8, 0x0f, 0xff, 0xff, 0xf0, 0x0f, 0xff, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xe0,  
  0x03, 0xff, 0xff, 0xc0, 0x01, 0xff, 0xff, 0x80, 0x00, 0xff, 0xff, 0x00, 0x00, 0x7f, 0xff, 0x00,  
  0x00, 0x3f, 0xfe, 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x0f, 0xf8, 0x00, 0x00, 0x07, 0xf0, 0x00,  
  0x00, 0x03, 0xc0, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00  
};  
  
int Signal;  
bool NoSignal = false;  
int cycle = 0;  
  
void setup() {  
  Serial.begin(9600);  
  
  // SSD1306\_SWITCHCAPVCC = generate display voltage from 3.3V internally  
  if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x32  
    Serial.println(F("SSD1306 allocation failed"));  
    for(;;); // Don't proceed, loop forever  
  }  
  
  // Configure the PulseSensor object, by assigning our variables to it.  
  pulseSensor.analogInput(PulseWire);    
  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.  
  pulseSensor.setThreshold(Threshold);    
  
  // Double-check the "pulseSensor" object was created and "began" seeing a signal.  
   if (!pulseSensor.begin()) {  
    Serial.println("We failed to start the pulseSensor Object !");  
    for(;;); // Don't proceed, loop forever  
  }  
  
  // Clear the buffer  
  display.clearDisplay();  
   
  display.display();  
  delay(3000);  
}  
  
void loop() {  
  int bpm = pulseSensor.getBeatsPerMinute();  
  if (pulseSensor.sawStartOfBeat()) {  
    cycle++;  
    if(cycle > 5) {  
      NoSignal = false;  
      drawBPM(bpm);  
      cycle = 5;  
    }  
  } else {  
    if (!NoSignal) {  
      cycle = 0;  
      NoSignal = true;  
      drawBPM(bpm);     //i have added this  
      //drawNoInput();  
    }  
  }  
  delay(20);  
}  
  
  
void drawBPM(int bpm) {  
  display.clearDisplay();  
  
  display.drawBitmap(10, 0, heart, 32, 32, 1);  
  display.setTextSize(4);  
  display.setTextColor(WHITE);  
  display.setCursor(32,0);  
  display.print(' ');  
  //bpm=bpm-120;  
  display.println(bpm);  
  
  display.display();  
  delay(1000);  
}  
  
void drawNoInput(void) {  
  display.clearDisplay();  
  display.setTextSize(2);             // Normal 1:1 pixel scale  
  display.setTextColor(WHITE);        // Draw white text  
  display.setCursor(0,0);             // Start at top-left corner  
  display.print(F("NO SIGNAL"));  
  display.display();  
}