*A MINI- PROJECT REPORT ON*

***“PATIENT HEALTH MONITORING SYSTEM”***

*SUBMITTED BY*

***Anant Shinde Roll No.-17IT2004***

***Ishan Kotian Roll No.-17IT2013***

***Yash Patil Roll No.-17IT2006***

*Supervisor*

# Sumedha Bhagwat



*Department of Information Technology*

*Dr. D. Y. Patil Group’s*

# Ramrao Adik Institute of Technology

*Nerul, Navi Mumbai*

*(Affiliated to University of Mumbai****)***

*( 2020)*

# CERTIFICATE

*This is to certify that the project entitled* ***` PATIENT HEALTH MONITORING SYSTEM*** *' being submitted by* ***Anant Shinde Roll No.-17IT2004, Ishan Kotian Roll No.-17IT2013, Yash Patil Roll No.-17IT2006****to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of* ***‘T. E. I. T’*** *in* ***“IOT (Mini Project) Lab”****.*

|  |  |  |
| --- | --- | --- |
| ***Project Guide*** | ***External Examiner*** | ***Head of Department*** |
| *(Sumedha Bhagwat)* | *( )* | *(Dr. Ashish Jadhav)* |

# DECLARATION

*We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.*

## Name and Roll No. of Students Signature

1. *Anant Shinde (17IT2004) ( )*

1. *Ishan Kotian (17IT2013) ( )*

1. *Yash Patil (17IT2006) ( )*

*Date:*

*Place:*

*ACKNOWLEDGEMENT*

*The project “* ***Patient Health Monitoring System*** *” is creative work of many minds. A proper synchronization between individual is must for any project to be completed successfully. One cannot imagine the power of the force that guides us all and neither can we succeed without acknowledging it.*

*We would like to express our gratitude to Principal* ***Dr. Mukesh D. Patil*** *and* ***Dr. Ashish Jadhav,*** *our Head of the department, Information Technology Engineering for encouraging and inspiring us to carry out the project in the department lab.*

*We would also like to thank our Guide* ***Sumedha Bhagwat****, Department of the Information Technology Engineering for her expert guidance, encouragement and valuable suggestions at every step.*

*We also would like to thank all the staff members Department of the Information Technology Engineering for providing us with the required facilities and support towards the completion of the project.*

*Last but not the least we are thankful to our parents and friends for their constant Inspiration, encouragement and well wishes by which we have made a challenging project.*

***Anant Shinde Roll No.-17IT2004***

***Ishan Kotian Roll No.-17IT2013***

***Yash Patil Roll No.-17IT2006***

# PREFACE

*We take great opportunity to present this Mini Project report on “****PATIENT HEALTH MONITORING SYSYTEM”*** *and put before readers some useful information regarding our project.*

*We have made sincere attempts and taken every care to present this matter in precise and compact form, the language being as simple as possible. We are sure that the information contained in this volume certainly prove useful for better insight in the scope and dimension of this project in it true perspective.*

*The task of the completion of the project though being difficult was made quite simple, interesting and successful due to deep involvement and complete dedication of our group members.*

***TABLE OF CONTENTS***

***Declaration …………….…….………………………………………… I***

***Acknowledgement II***

***Preface ............................................................. III***

***Table of Contents ……………………………………………………… IV***

***Table of figures .......................................................... V***

***Abstract ................................................................. VI***

# TABLE OF CONTENTS

***Sr. No. Topic Page No.***

## 1. INTRODUCTION .............................................................. 8

***1.1 IOT INTRODUCTION, CHARACTERISTICS AND 6A CONNECTIVITY ................................................. 8 1.2 PROBLEM STATEMENT ................................................ 12 1.3 OBJECTIVES .............................................................. 13***

## 2. LITERATURE SURVEY ........................................................ 14

***2.1 MOTIVATION ........................................................ 15***

## 3. PROPOSED SYSTEM ......................................................... 16

***3.1 INTRODUCTION OF PROPOSED SYSTEM AND ARCHITECTURE .......................................................... 16 3.2 HARDWARE AND SOFTWARE REQUIREMENTS ..................... 17***

## 4. IMPLEMENTATION .................................................... 21

***4.1 CIRCUIT DIAGRAM USING SIMULATOR ........................ 21 4.2 CONNECTION DETAILS ............................................ 22 4.3 CODE .................................................................. 26***

***5. RESULT ................................................................... 36***

## 6. CONCLUSION AND FUTURE SCOPE ................................ 37

***6.1 CONCLUSION ....................................................... 37 6.2 FUTURE SCOPE .................................................... 37 6.3 BENEFITS TO SOCIETY ........................................... 38***

***7. REFERENCES .......................................................... 47***

***ABSTRACT***

Patient Health Monitoring System using ESP8266 & Arduino:

With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry. In this project, we have designed the IoT Based Patient Health Monitoring System using ESP8266 & Arduino. The IoT platform used in this project is ThingSpeak. ThingSpeak 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. This IoT device could read the pulse rate and measure the surrounding temperature. It continuously monitors the pulse rate and surrounding temperature and updates them to an IoT platform.

The Arduino Sketch running over the device implements the various functionalities of the project like reading sensor data, converting them into strings, passing them to the IoT platform, and displaying measured pulse rate and temperature on character LCD.

***Introduction to IoT***

***1 IOT INTRODUCTION, CHARACTERISTICS***

Today, Internet application development demand is very high. So IoT is a major technology by which we can produce various useful internet applications.

Basically, IoT is a network in which all physical objects are connected to the internet through network devices or routers and exchange data. IoT allows objects to be controlled remotely across existing network infrastructure. IoT is a very good and intelligent technique which reduces human effort as well as easy access to physical devices. This technique also has autonomous control feature by which any device can control without any human interaction.

“Things” in the IoT sense, is a mixture of hardware, software, data, and services. “Things” can refer to a wide variety of devices such as DNA analysis devices for environmental monitoring, electric clamps in coastal waters, Arduino chips in home automation and many others. These devices gather useful data with the help of various existing technologies and share that data between other devices. Examples include Home Automation System which uses Wi-Fi or Bluetooth for exchange data between various devices of home.

**CHARACTERISTICS OF IOT**

### INTELLIGENCE

Together algorithms and compute (i.e. software & hardware) provide the “intelligent spark” that makes a product experience smart. Consider Misfit Shine, a fitness tracker, compared to Nest’s intelligent thermostat. The Shine experience distributes compute tasks between a smartphone and the cloud. The Nest thermostat has more compute horsepower for the AI that makes them smart.

### CONNECTIVITY

Connectivity in the IoT is more than slapping on a Wi-Fi module and calling it a day. Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data. If this sounds familiar, that’s because it is Metcalfe’s Law and it rings true for IoT.

### SENSING

We tend to take for granted our senses and ability to understand the physical world and people around us. Sensing technologies provide us with the means to create experiences that reflect a true awareness of the physical world and the people in it. This is simply the analogy input from the physical world, but it can provide a rich understanding of our complex world.

### EXPRESSING

Expressing enables interactivity with people and the physical world. Whether it is a smart home or a farm with smart agriculture technology, expressing provides us with a means to create products that interact intelligently with the real world. This means more than just rendering beautiful UIs to a screen. Expressing allows us to output into the real world and directly interact with people and the environment.

### ENERGY

Without energy we can’t bring our creations to life. The problem is we can’t create billions of things that all run on batteries. Energy harvesting, power efficiency, and charging infrastructure are necessary parts a power intelligent ecosystem that we must design. Today, it is woefully inadequate and lacks the focus of many product teams.

### SAFETY

As we gain efficiencies, novel experiences, and other benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints, networks, and the data moving across all of it means creating a security paradigm that will scale.

***Problem Statement***

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.

In this Project, We will make an IoT based Health Monitoring System which records the patient pulse rate and body temperature and also send a data over the internet whenever those readings goes beyond critical values. Pulse rate and body temperature readings are recorded over ThingSpeak so that patient health can be monitored from anywhere in the world over internet.

***Objectives***

## 1. To design a system to reduce manual efforts.

## 2. Develop fast sensing mechanism.

## 3. Make use of knowledge of IOT sensing in solving health care problems.

## 4. To have an easy to use, time saving approach.

## 5.To make knowledge of the current healthcare facilities.

## 6. To monitor the patient using sensors.

***Literature Survey***

# 1. An IoT based patient monitoring system using raspberry Pi

Author: [R. Kumar](https://ieeexplore.ieee.org/author/37085897962); [M. Pallikonda Rajasekaran](https://ieeexplore.ieee.org/author/38667247500)

In this paper discuss about, monitoring patient's body temperature, respiration rate,heart beat and body movement using Raspberry Pi board.

2. An IOT based Human healthcare system using Arduino Uno board

Author: [S. Jayapradha](https://ieeexplore.ieee.org/author/37086370662); P. M. Durai Raj Vincent

This paper contains various IOT applications and the role of IOT in the healthcare system, challenges in the healthcare system using IOT. Also, introduced a secured surveillance monitoring system for reading and storing patient's details using low power for transmitting the data.

# 3. Health Monitoring & Management using IoT devices in a Cloud Based Framework

Author: [Anirvin Sharma](https://ieeexplore.ieee.org/author/37086431207); [Tanupriya Choudhury](https://ieeexplore.ieee.org/author/37085794967); [Praveen Kumar](https://ieeexplore.ieee.org/author/37085348077)

The framework this paper proposes is aptly called the Internet of Medical Things (IoMT) which opens a whole new avenue for the Patient-HealthCare provider Interface (PHI) and Wearable Health Technology (WHT).

# 4. An IoT based Patient Health Monitoring System

Author: [D. Shiva Rama Krishnan](https://ieeexplore.ieee.org/author/37086441692); [Subhash Chand Gupta](https://ieeexplore.ieee.org/author/37086011462); [Tanupriya Choudhury](https://ieeexplore.ieee.org/author/37085794967)

This system uses Temperature and heartbeat sensor for tracking patients health. Both the sensors are connected to the Arduino-uno. To track the patient health micro-controller is in turn interfaced to a LcD display and wi-fi connection to send the data to the web-server(wireless sensing node).

# 5. IoT based low-cost distant patient ECG monitoring system

Author: [Parmveer Singh](https://ieeexplore.ieee.org/author/37086332969); [Ashish Jasuja](https://ieeexplore.ieee.org/author/37086286270)

This paper studies the application of IoT in health care domain and a system is proposed to monitor the ECG of the distant patient.

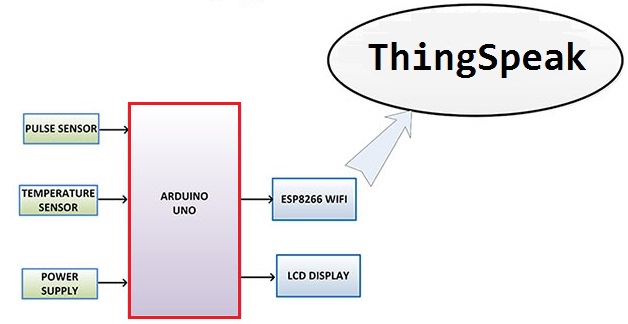
***Motivation***

In the Pandemic situation the monitoring of the patient for heart rate and bod temperature constantly was a necessity. Due this situation some critical patients also couldn’t be monitored because of the risk of going to the hospitals was greater for them. So we thought of a system that could help the Doctors monitor the patients from their convince on the internet. This system will be at the patient’s home where they can keep on giving the readings and the doctors can monitor them at their clinic through internet.

***PROPOSED SYSTEM***

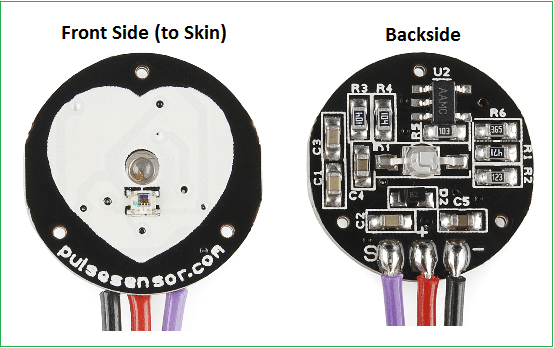
***Introduction to proposed system and architecture***

This is a simple block diagram that explains the **IoT Based Patient Health Monitoring System using ESP8266 & Arduino**. Pulse Sensor and LM35 Temperature Sensors measure BPM & Environmental Temperature respectively. The Arduino processes the code and displays it to 16\*2 LCD Display. **ESP8266 Wi-Fi module** connects to Wi-Fi and sends the data to IoT device server. The IoT server used here is Thingspeak. Finally, the data can be monitored from any part of the world by logging into the Thingspeak channel.



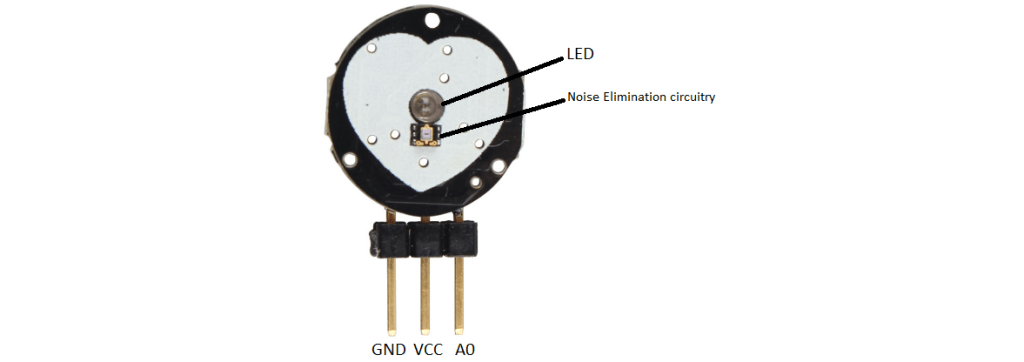
***HARDWARE AND SOFTWARE REQUIREMENTS***

### 1. **Pulse Sensor:**



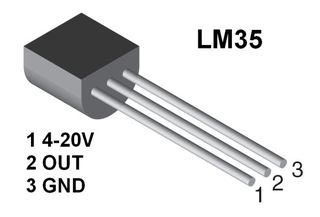
The **Pulse Sensor** is a plug-and-play **heart-rate sensor for Arduino**. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects.

The pulse sensor has three pins: VCC, GND & Analog Pin.



There is also a LED in the center of this sensor module which helps in detecting the **heartbeat**. Below the LED, there is a noise elimination circuitry that is supposed to keep away the noise from affecting the readings.

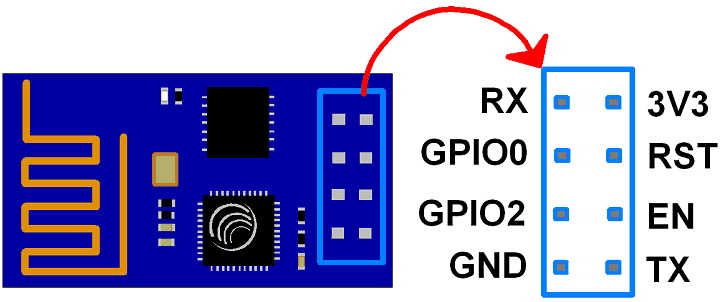
### 2. **LM35 Temperature Sensor:**



The **LM35** series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range.

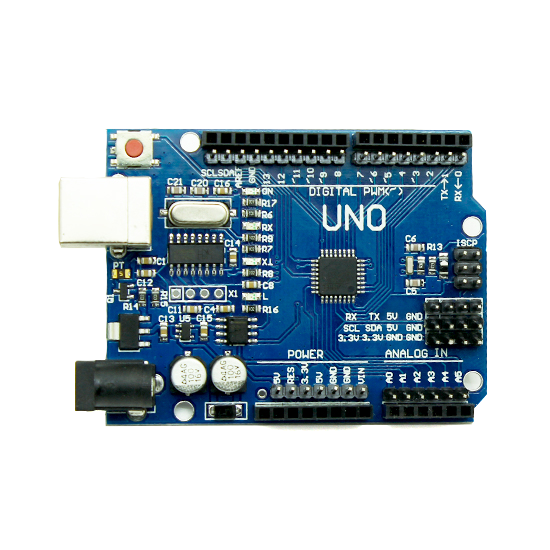
3**.** **ESP8266:**

The **ESP8266** is a very user-friendly and low-cost device to provide internet connectivity to your projects. The module can work both as an 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 the Internet of Things as easy as possible. It can also fetch data from the internet using API’s hence your project could access any information that is available on 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.



**Pin 1: Ground:** Connected to the ground of the circuit  
**Pin 2: Tx/GPIO – 1:** Connected to Rx pin of programmer/uC to upload program  
**Pin 3: GPIO – 2:** General purpose Input/output pin  
**Pin 4 : CH\_EN:** Chip Enable/Active high  
**Pin 5: Flash/GPIO – 0:** General purpose Input/output pin  
**Pin 6 : Reset:** Resets the module  
**Pin 7: RX/GPIO – 3:** General purpose Input/output pin  
**Pin 8: Vcc:** Connect to +3.3V only

4.Arduino Uno:



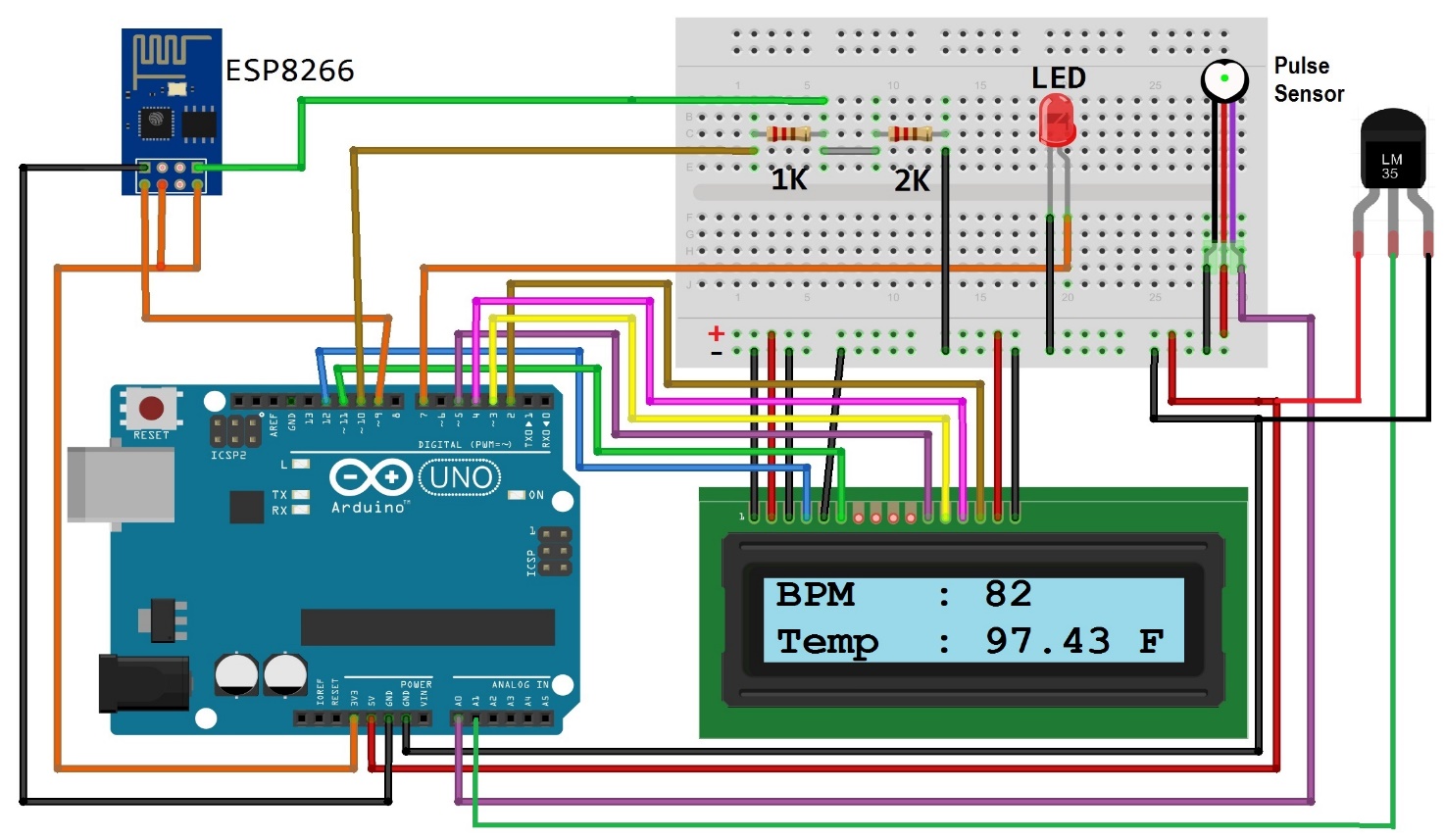
### General pin functions

* **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
* **VIN**: The input voltage to the Arduino/Genuino board when it is 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 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the 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/Genuino 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 to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields that block the one on the board.

**IMPLEMENTATION**

***CIRCUIT DIAGRAM USING SIMULATOR:***

For designing IoT Based Patient Health Monitoring System using ESP8266 & Arduino, assemble the circuit as shown in the figure below.



***CONNECTION DETAILS***

1. Connect Pulse Sensor output pin to A0 of Arduino and other two pins to VCC & GND.
2. Connect LM35 Temperature Sensor output pin to A1 of Arduino and other two pins to VCC & GND.
3. Connect the LED to Digital Pin 7 of Arduino via a 220-ohm resistor.
4. Connect Pin 1,3,5,16 of LCD to GND.
5. Connect Pin 2,15 of LCD to VCC.
6. Connect Pin 4,6,11,12,13,14 of LCD to Digital Pin12,11,5,4,3,2 of Arduino.
7. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting the 2.2K & 1K resistor. Thus the RX pin of the ESP8266 is connected to pin 10 of Arduino through the resistors.
8. Connect the TX pin of the ESP8266 to pin 9 of the Arduino.

***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 = "I5Y51SPJBTPLAZEW";

int pulsePin = A0;

int blinkPin = 7 ;

int fadePin = 13;

int fadeRate = 0;

volatile int BPM;

volatile int Signal;

volatile int IBI = 600;

volatile boolean Pulse = false;

volatile boolean QS = false;

static boolean serialVisual = true;

volatile int rate[10];

volatile unsigned long sampleCounter = 0;

volatile unsigned long lastBeatTime = 0;

volatile int P = 512;

volatile int T = 512;

volatile int thresh = 525;

volatile int amp = 100;

volatile boolean firstBeat = true;

volatile boolean secondBeat = false;

void setup()

{

lcd.begin(16, 2);

pinMode(blinkPin,OUTPUT);

pinMode(fadePin,OUTPUT);

Serial.begin(115200);

interruptSetup();

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(115200);

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=\"iPhone\",\"yashpatil\"";

ser.println(cmd);

delay(1000);

ser.println("AT+CIFSR");

delay(1000);

}

void loop()

{

serialOutput();

if (QS == true)

{

fadeRate = 255;

serialOutputWhenBeatHappens();

QS = false;

}

ledFadeToBeat();

delay(20);

read\_temp();

esp\_8266();

}

void ledFadeToBeat()

{

fadeRate -= 15;

fadeRate = constrain(fadeRate,0,255);

analogWrite(fadePin,fadeRate);

}

void interruptSetup()

{

TCCR2A = 0x02;

TCCR2B = 0x06;

OCR2A = 0X7C;

TIMSK2 = 0x02;

sei(); //

}

void serialOutput()

{

if (serialVisual == true)

{

arduinoSerialMonitorVisual('-', Signal);

}

else

{

sendDataToSerial('S', Signal);

}

}

void serialOutputWhenBeatHappens()

{

if (serialVisual == true)

{

Serial.print("\* Heart-Beat Happened \* ");

Serial.print("BPM: ");

Serial.println(BPM);

}

else

{

sendDataToSerial('B',BPM);

sendDataToSerial('Q',IBI);

}

}

void arduinoSerialMonitorVisual(char symbol, int data )

{

const int sensorMin = 0;

const int sensorMax = 1024;

int sensorReading = data;

int range = map(sensorReading, sensorMin, sensorMax, 0, 11);

switch (range)

{

case 0:

Serial.println("");

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)

{

cli();

Signal = analogRead(pulsePin);

sampleCounter += 2;

int N = sampleCounter - lastBeatTime;

if(Signal < thresh && N > (IBI/5)\*3)

{

if (Signal < T)

{

T = Signal;

}

}

if(Signal > thresh && Signal > P)

{

P = Signal;

}

if (N > 250)

{

if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)\*3) )

{

Pulse = true;

digitalWrite(blinkPin,HIGH);

IBI = sampleCounter - lastBeatTime;

lastBeatTime = sampleCounter;

if(secondBeat)

{

secondBeat = false;

for(int i=0; i<=9; i++)

{

rate[i] = IBI;

}

}

if(firstBeat)

{

firstBeat = false;

secondBeat = true;

sei();

return;

}

word runningTotal = 0;

for(int i=0; i<=8; i++)

{

rate[i] = rate[i+1];

runningTotal += rate[i];

}

rate[9] = IBI;

runningTotal += rate[9];

runningTotal /= 10;

BPM = 60000/runningTotal;

QS = true;

pulse = BPM;

}

}

if (Signal < thresh && Pulse == true)

{

digitalWrite(blinkPin,LOW);

Pulse = false;

amp = P - T;

thresh = amp/2 + T;

P = thresh;

T = thresh;

}

if (N > 2500)

{

thresh = 512;

P = 512;

T = 512;

lastBeatTime = sampleCounter;

firstBeat = true;

secondBeat = false;

}

sei();

}

void esp\_8266()

{

String cmd = "AT+CIPSTART=4,\"TCP\",\"";

cmd += "184.106.153.149";

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);

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");

}

***RESULT***

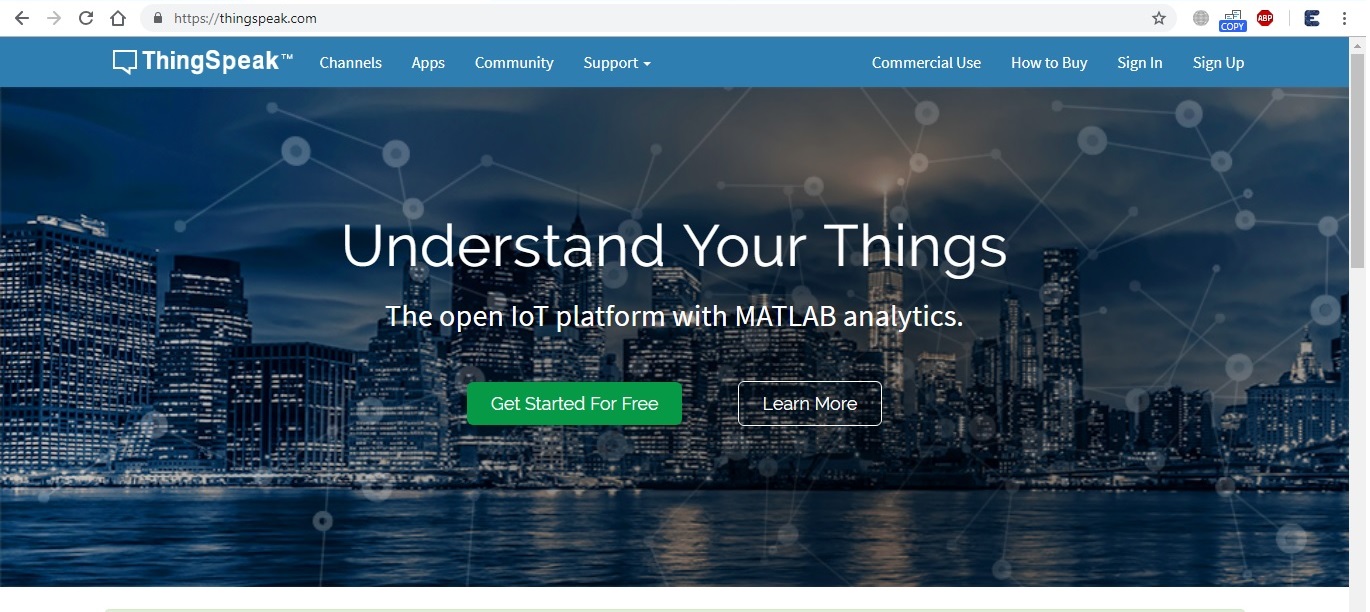
***A circuit board

Description automatically generated***

***A close up of a computer

Description automatically generated***

***Graphical user interface, application

Description automatically generated***

***CONCLUSION AND FUTURE SCOPE***

**CONCLUSION**

The system we have created will help the doctors to monitor their patients from anywhere is the world through the website known as Thingspeak. The patient’s body temperature and heart rate will be received from our device and the data will be sent to the Thingspeak website where the Doctor will monitor it.

**FUTURE SCOPE**

Our device has the scope of monitoring blood oxygen level, breath analysis. This will can also be modified to sent emergency signal to the hospital and the emergency contact of the patient.

***REFERENCES***

[1] R. Kumar; M. Pallikonda Rajasekaran. An IoT based patient monitoring system using raspberry Pi. Assistant Professor, Dept. of CSE, N.I.T.Nagpur, R.T.M. Nagpur University, Nagpur, Maharashtra, India.

[2] S. Jayapradha; P.M Durai Raj Vincent. An IOT based Human healthcare system using Arduino Uno board. Associate Professor, SITE, VIT University, Vellore, Tamilnadu.

[3] <https://ieeexplore.ieee.org/abstract/document/8250529>

[4] <https://ieeexplore.ieee.org/abstract/document/8441752>

[5] <https://ieeexplore.ieee.org/abstract/document/8441708>

[6] <https://www.ijareeie.com/upload/2017/june/37_E60606069.pdf>

[7] <https://acadpubl.eu/jsi/2017-117-16/articles/17/32.pdf>