DEPARTMENT OF INFORMATION TECHNOLOGY FACULTY OF ENGINEERING & TECHNOLOGY

IOT PROJECT REPORT

SUBJECT TITLE : INTERNET OF THINGS

SUBJECT CODE: 15IT422E

SUBMITTED TO: Prof Kayalvizhi Jayavel

FALL DETECTION SYSTEM

PRACHIKA AGARWAL

RA1611008010249

Department of Information Technology

SRM University, SRM Nagar, Kattankulathur-603203 Kanchipuram District, Tamil Nadu

LINKS TO GITHUB AND YOUTUBE:

YouTube:

https://youtu.be/uot7ELnhGO8

Github:

https://github.com/prachikaag/Smart-Hospital-Management-System/edit/master/README.md

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to our IoT Professor Mrs Kayalvizhi Jayavel for being out there at every step of my course and guiding us all along the way to be capable of moulding our ideas into smart projects and helping us enhance our skills.

I would also like to thank my parents, without whom I wouldn't be able to anything in any regard.

ABSTRACT

HARDWARE REQUIRED:

- Nodemcu ESP8266 module.
- LM35 Temp Sensor
- KY039 Heartbeat Sensor
- Jumper Wires
- A Smartphone
- A Computer.
- Micro USB cable

SOFTWARE REQUIRED :

- Arduino IDE.
- ESP8266 library.
- LM35 Library.
- KY039 Library
- ThingSpeak MQTT Library.

MISCELLANEOUS :

- Wifi Internet Connection
- Soldering tool
- Soldering wires

TOTAL COST OF COMPONENTS :- Rs.500 - Rs.600.

ESP8266- Rs.380

LM35- Rs.60

KY039-Rs.60

Jumper wires- Rs.20

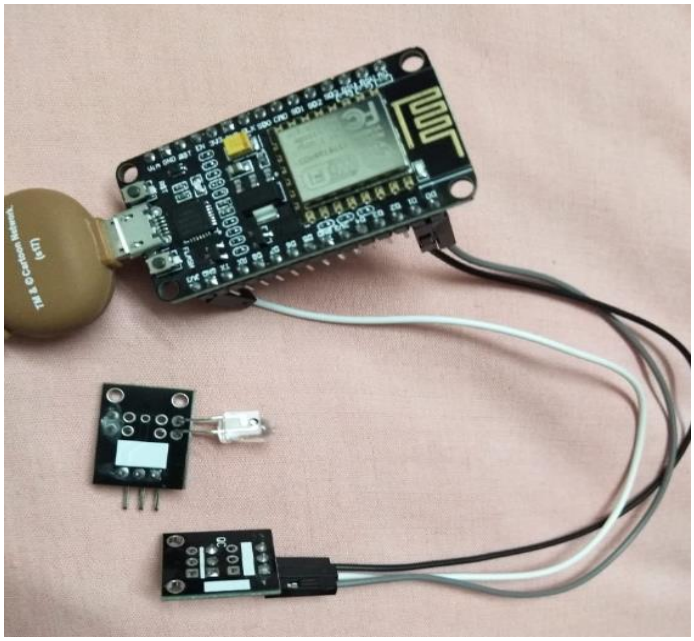
SYSTEM OVERVIEW

The main component of the setup is the Nodemcu ESP8266 module. All the other hardware components are connected to the Nodemcu. The board is programmed in Arduino IDE and uses the ESP8266,LM35,KY039 and ThingSpeak MQTT libraries. These libraries have been added to the Arduino IDE.

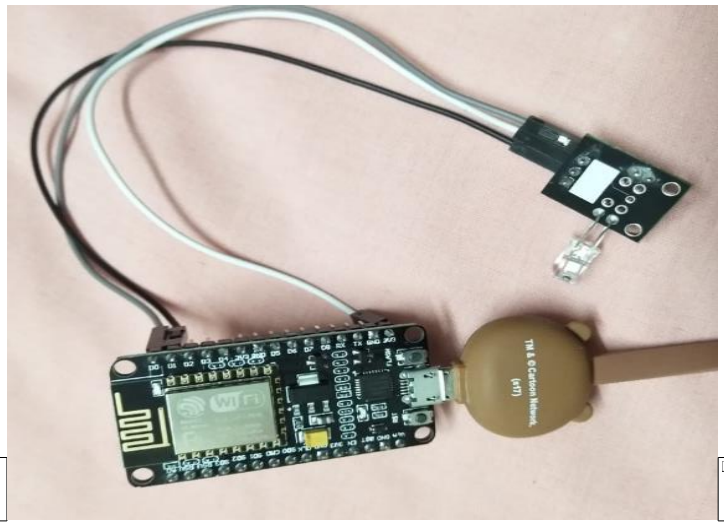
The LM35 temperature sensor along with KY039 heartbeat sensor is directly connected to the NodeMCU using a micro USB cable. This is used to detect the body temperature and heartbeat and is signalled by the programmed NodeMCU ESP8266 with a trigger as an Email using ThingSpeak.

Circuit diagrams:

Circuit for NodeMCU and LM35



Circuit for NodeMCU and KY039



The image part with relationship ID rId8 was not found in the file.

The image part with relationship ID rId8 was not found in the file.

Programming NodeMCU and setting up ThingSpeak Dashboard :

- Make all the necessary circuit connections.
- Import ESP8266WiFi.h Library to the Arduino IDE
- Fill your THINGSPEAK WRITEAPI key , wifi ssid, and wifi password to the code.
- Upload temperature sensor and heartbeat detection code to your NodeMCU.
- Setup your Thingspeak dashboard, add a new channel and two visualization and give it name.
- Run your code in aurdino ide.
- Once the code is running open COM3 port where all your readings are shown.
- These readings are reflected on thingspeak.

CODE
Temperature sensor

```
#include <ESP8266WiFi.h>
String apiWritekey = "YAFEPH5003CIZPB6"; // replace with your THINGSPEAK WRITEAPI key here
const char* ssid = "G-207"; // your wifi SSID name
const char* password = "superwoman" ;// wifi pasword
const char* server = "api.thingspeak.com";
float resolution=3.3/2307;// 3.3 is the supply volt & 1023 is max analog read value
WiFiClient client;
void setup() {
Serial.begin(115200);
WiFi.disconnect();
delay(10);
WiFi.begin(ssid, password);
Serial.println();//prints over the serial port
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.print("NodeMcu connected to wifi...");
Serial.println(ssid);
Serial.println();
}
void loop() {
float temp = (analogRead(A0) * resolution) * 100;
if (client.connect(server,80))
{
String tsData = apiWritekey;
tsData +="&field1=";
tsData += String(temp);
tsData += "\r\n\r\n";
client.print("POST /update HTTP/1.1\n");//here the post is been done over the //think speak
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiWritekey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(tsData.length());
```

```
client.print("\n\n"); // the 2 carriage returns indicate closing of Header fields & starting of data
client.print(tsData);
Serial.print("Temperature: ");
Serial.print(temp);
Serial.println("uploaded to Thingspeak server....");
}
client.stop();
Serial.println("Waiting to upload next reading...");
Serial.println();
// thingspeak needs minimum 15 sec delay between updates
delay(15000);
}
```

Heartbeat sensor

```
#include <ESP8266WiFi.h>
String apiWritekey = "4QDNCV5V95FN1LMR"; // replace with your THINGSPEAK WRITEAPI key here
const char* ssid = "G-207"; // your wifi SSID name
const char* password = "superwoman" ;// wifi pasword
const char* server = "api.thingspeak.com";
WiFiClient client;
int led_Pin = 13;                          // initializing the led pin
int output_Pin = A0;                        // initializing the sensor output pin
//initializng other variables
double alpha = 0.75;
int period = 2000;
double change = 0.0;
void setup ( )                    // Code written in it will only run once.
{
  pinMode (led_Pin, OUTPUT);              // declaring led pin as output
 Serial.begin(115200);
WiFi.disconnect();
delay(10);
WiFi.begin(ssid, password);
Serial.println();//prints over the serial port
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.print("NodeMcu connected to wifi...");
Serial.println(ssid);
```

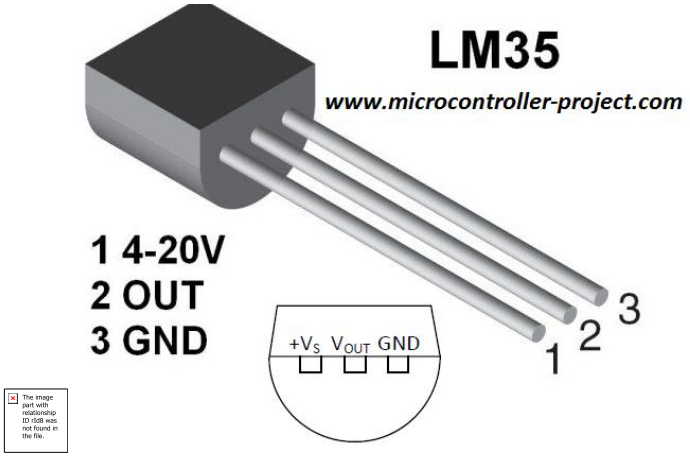
```
Serial.println();// setting baud rate at 9600
}
void loop ()
{
   // initializing other variables

   static double oldValue = 0;
   static double oldChange = 0;
   int rawValue = analogRead (output_Pin);                                    // Reading the sensors values
   double value = alpha * oldValue + (1 - alpha) * rawValue;       // calculating values using the formula
   //Serial.print (rawValue);            // printing the sensor output value on the screen
   //Serial.print (",");
   //Serial.println (value);           // printing the heart beat value on the screen
   //oldValue = value;


   //float temp = (analogRead(A0) * resolution) * 100;
if (client.connect(server,80))
{
String tsData = apiWritekey;
tsData +="&field1=";
tsData += String(rawValue);
tsData += "\r\n\r\n";
client.print("POST /update HTTP/1.1\n");//here the post is been done over the //think speak
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiWritekey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(tsData.length());
client.print("\n\n"); // the 2 carriage returns indicate closing of Header fields & starting of data
client.print(tsData);
Serial.print("Heartbeat: ");
Serial.print(value);
Serial.println("uploaded to Thingspeak server....");
}
client.stop();
Serial.println("Waiting to upload next reading...");
Serial.println();
// thingspeak needs minimum 15 sec delay between updates
delay(period);
}
```

ThinkSpeak Dashboard, Widgets and Triggers

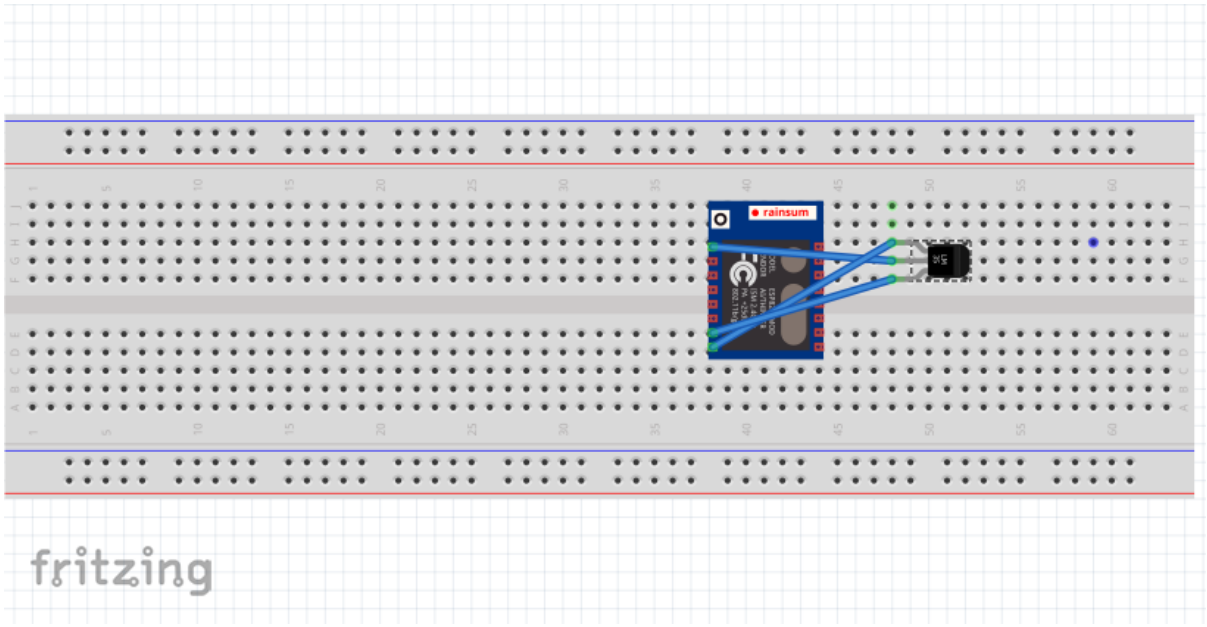
We give our WiFi SSID and password inside the code before compiling and uploading it to the Nodemcu. Also in order to connect to the ThinkSpeak database, we give the ThinkSpeak username, ThinkSpeak password and ThinkSpeak client ID so that it can connect via MQTT.

We then create the two state widget to make triggers in it. The trigger states that when a rise or fall in temperate is signalled by the temperature sensor and if there is an increase/decrease in heartbeat , it send a detected information to NodeMCU board connected to the ThinkSpeak network which then provides a Triggered notification to the mail given on the trigger.

Lm35 connection diagram:

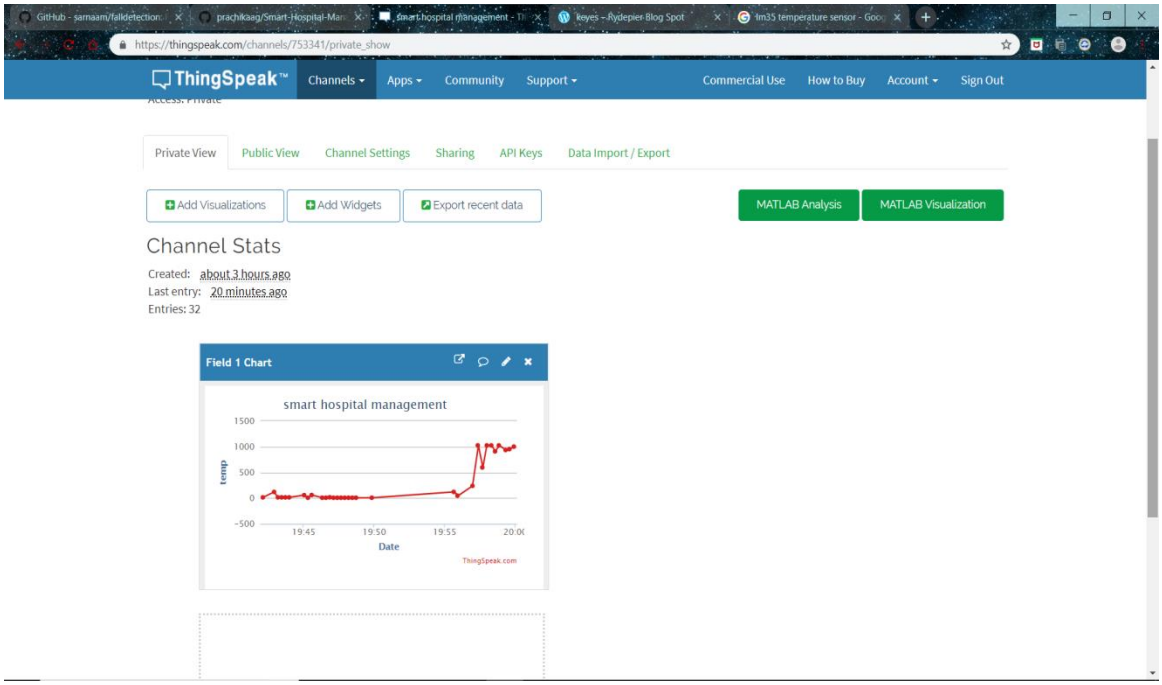


The image part with relationship ID rId8 was not found in the file.

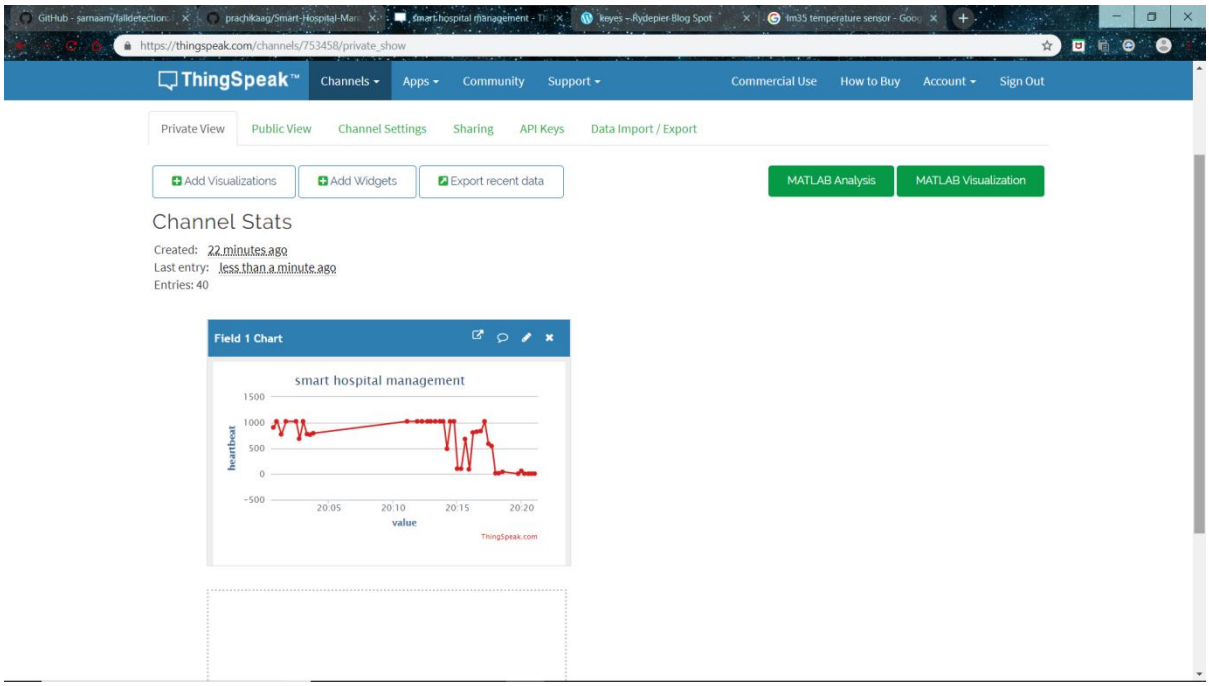


Ky039 connection diagram:

Temp sensor:



Heartbeat sensor



Result :

Smart hospital management system using ESP8266 NodeMCU, LM35 for temperature and KY039 for heartbeat sensor has been successfully developed and implemented.