IOT DOMAIN ANALYST

ECE3502

Final Report Batch-3

Remote health monitoring of Infectious Patient

Team Members:

- Mukund-19BEE0122
- Shant Rakshit-19BEE0154
- Anmol Khantaal-19BEE0153
- Moiz K-19BEE0165

Description:

The core objective of this project is the design and implementation of a smart patient health tracking system that uses sensors like temperature sensor etc to track patient health and other essential information. The objective of developing monitoring systems is to reduce health care costs by reducing physician office visits, hospitalisations, and diagnostic testing procedure. Each of our bodies utilises temperature if framework finds any sudden changes in understanding body temperature and any strange movement, the framework consequently alarms the client about the patient's status over IOT and furthermore indicates subtle elements like temperature of patient live in the web.

Existing Model:

In general, IoT has been broadly used to interconnect the serious clinical assets and to offer keen and successful medical care administrations to the individuals. The serious sensors can be either worn or be inserted into the body of the patients, in order to consistently screen their wellbeing. The data gathered in such way, can be examined, collected and mined to do the early expectation of illnesses.

Literature Survey:

Sharma, Nonita, et al. [1] have proposed an IoT based framework which is an alarm enable bio-wearable sensor for early detection of COVID-19 based on ontology methodology using biosignals in the form of ECG,PPG,temperature etc.

Honey Pandey et. al [2] have proposed a system for predicting heart ailments using sensor values like pulse, ECG and Spo2 and then using IFTTT transfer protocol for making csv file and then using Machine Learning algorithms most notably Decision Tree and SVM for prediction purposes.

D.S. R.Krishnan et. al [3] proposed an all connected to internet approach of health monitoring. Any abnormalities in patients health(temperature, heart rate, etc) is monitored and any alert is sent through wireless servers.

Gonçalo M et. al [4] briefs about the concept of cloud computing and edge computing and its resources that helps in prediction of health of a monitored patient via the network.

Poongodi, M. et al.[5] put forwards a device enabled to carry out a quick 30 second diagnosis using heartbeat, temperature, breath rate sensors and transmit the vitals to the hospital and doctor database. It employs the GSM model for quick ambulance alert and live tracking to reduce the time elapsed.

A. Athira et .al [6] proposed work implements an IoT based MPM system to monitor and analyze the vitals using corresponding sensors and sets the database for reference. It further implements SVM for the performance improvement for the given MPM system.

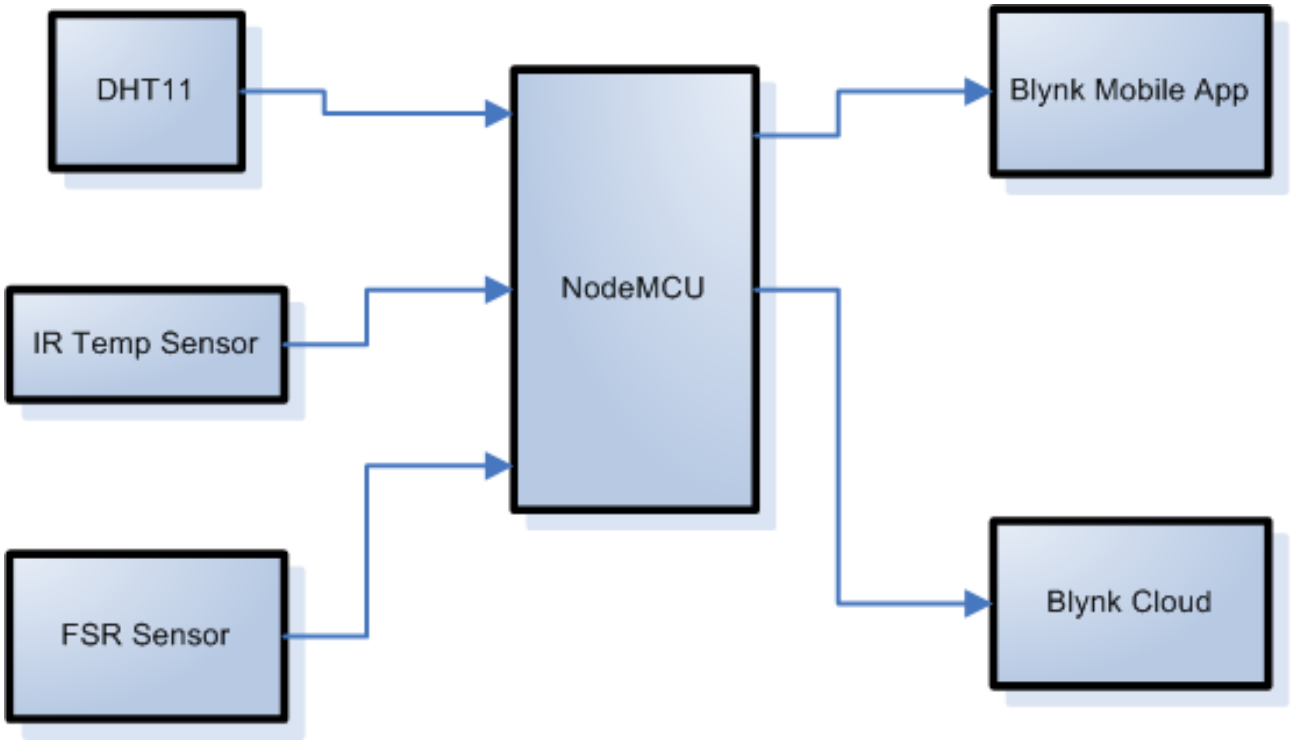
Tamilselvi, V. et. al [7] has utilized GSM and IoT to perceive the status or state of the patient. This proposed strategy comprises of various savvy sensors like Temperature, Heartbeat, Eye flicker and SPO2 sensors for getting the patient's internal heat level, coronary pulse, eye development and oxygen immersion level of the patient

Yeri, V. et .al[8] proposed framework comprises of the web and portable application in light of persistent remote checking of patients. The goal is paper is to carry out a minimal expense framework and send the patient essential signs in crisis circumstances. Sensors are being utilized for estimating the patient indispensable signs by utilizing the remote network.

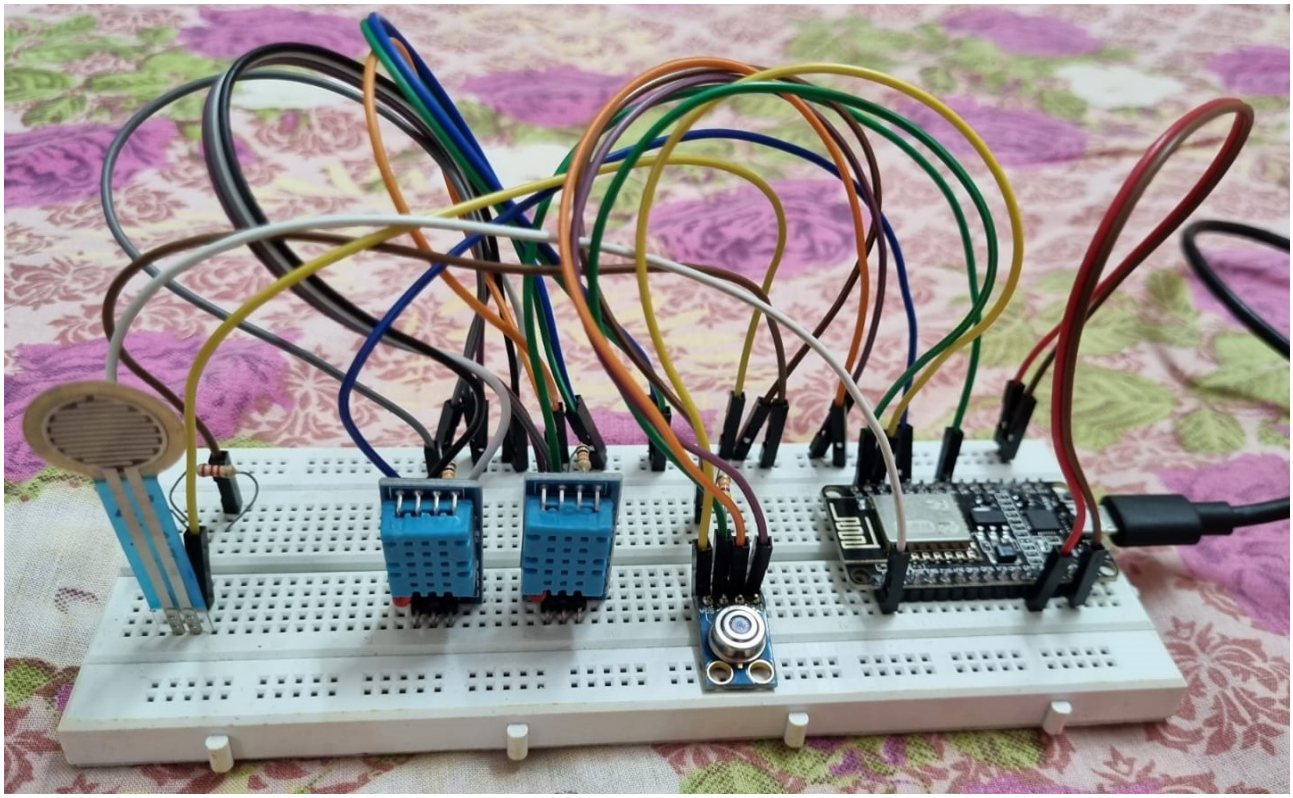
Components Required:

- Jumper Wires
- Breadboard
- 1k and 2k resistors.
- FSR Force Sensor.
- DHT11 Temperature and Humidity Sensor.
- MLX90614 IR Temperature Sensor.
- NodeMCU Wifi Module
- Blynk Cloud.

Block Diagram:



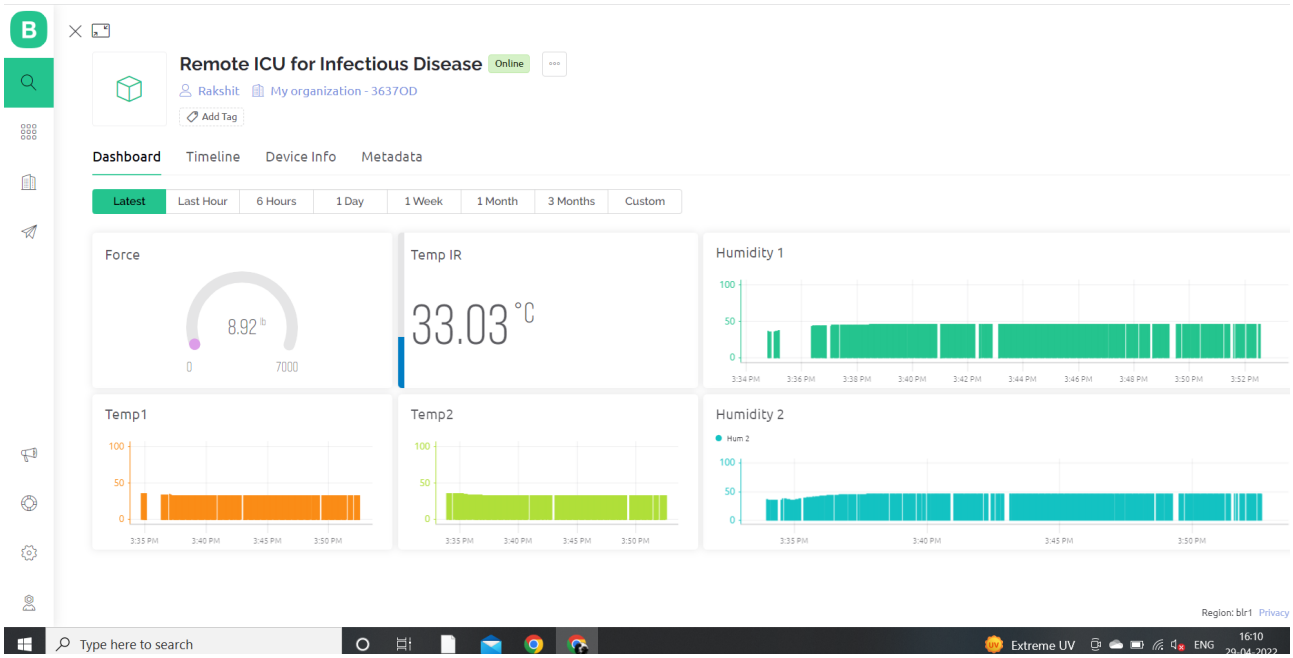
Circuit Diagram:



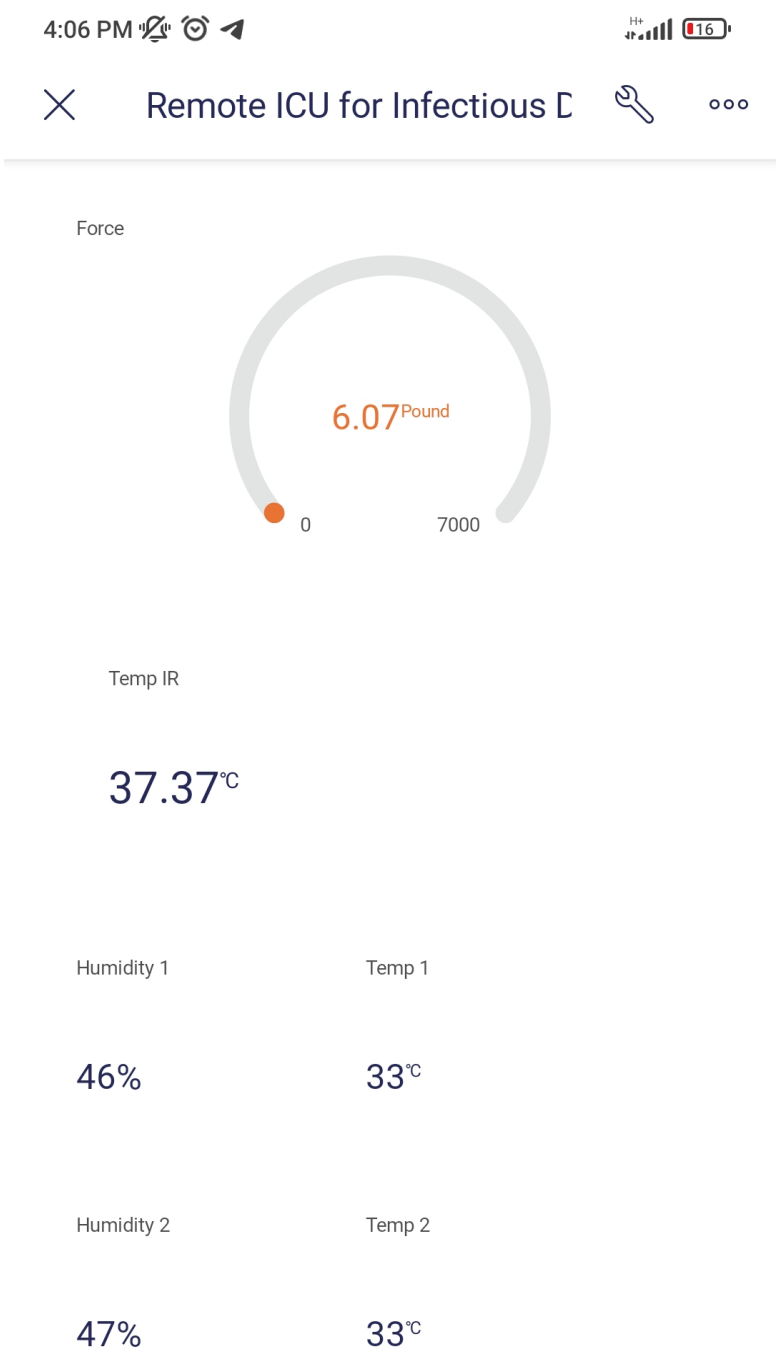
Working:

The DHT11 sensor will record the environmental conditions around the patient's bedrest as well as the room while the IR sensor will keep monitoring the patient's temperature. If the patient needs any medical assistance then he or she will put pressure on FSR sensor which will record the amount of force being exerted and if it is above the specified threshold it will alert the medical staff that the patient needs some kind of assistance. The data will be sent to Blynk Cloud with help of NodeMCU Wifi Module and it will show the real time data without any delay. Baud Rate should be kept at 115200 ms so as to reduce latency. Thus on dashboard the doctor can keep a check on the patient from his cabin and thus no requirement to visit patient's room thus reducing the risk of infection. Data will also be transferred to a mobile app thus can be used by patient's family members to keep a check on his condition.

Blynk-Dashboard:



Blynk-App:


4:06 PM
Remote ICU for Infectious D
Force
6.07Pound
0
7000
Temp IR
37.37℃
Humidity 1
46%
Temp 1
33℃
Humidity 2
47%
Temp 2
33℃

Result and Inference:

Thus we can see that real time data is sent over to blynk and can be viewed simultaneously on web based cloud as well as app. Since Blynk is open-source it can be further utilised for adding vizualization and can also generate CSV file, and furthermore we can perform data analysis on the csv data and can predict the future trends regarding the patient's condition. Since data is stored in cloud it can be used for further references by other doctors and other medical institutions and it can give an insight into the patient's medical history.

Code:

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLYVYfR7wz"
#define BLYNK_DEVICE_NAME "Diabeties"
#define BLYNK_AUTH_TOKEN "gb5pVrDVrDVLkKrFOuIsSE01C7Il5cWr"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include "FSR.h"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <Adafruit_MLX90614.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "ShantR";
char pass[] = "rakshitS";

#define DHTPIN1 D4
#define DHTPIN2 D5
#define FSR_PIN_1  A0// What digital pin we're connected to
```

```
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

// Uncomment whatever type you're using!
#define DHTTYPE DHT11     // DHT 11
//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321
//#define DHTTYPE DHT21   // DHT 21, AM2301

DHT dht1(DHTPIN1, DHTTYPE);
DHT dht2(DHTPIN2, DHTTYPE);
FSR fsr(FSR_PIN_1);
BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual
Pin (5).
// In the app, Widget's reading frequency should be set to PUSH.
This means
// that you define how often to send data to Blynk App.
void sendSensor()
{
  float h1 = dht1.readHumidity();
  float t1 = dht1.readTemperature(); // or dht.readTemperature(t
  rue) for Fahrenheit
  float h2 = dht2.readHumidity();
  float t2 = dht2.readTemperature();
  float fsrForce = fsr.getForce();
  float temp_ir = mlx.readObjectTempC();

    Serial.print(F("force: "));
    Serial.print(fsrForce);
    Serial.print(F(" [g]"));

    Serial.print(F("Temp IR "));
    Serial.print(temp_ir);
    Serial.print((" *C"));

    Serial.print(F("Humidity_1: "));
    Serial.print(h1);
    Serial.print(F(" [%]\t"));
```

```
    Serial.print(F("Temp_1: "));
    Serial.print(t1);
    Serial.println(F(" [C]"));

    Serial.print(F("Humidity_2: "));
    Serial.print(h2);
    Serial.print(F(" [%]\t"));

    Serial.print(F("Temp_2: "));
    Serial.print(t2);
    Serial.println(F(" [C]"));


  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  //Blynk.virtualWrite(V5, h);
  //Blynk.virtualWrite(V6, t);
  Blynk.virtualWrite(V5,fsrForce);
   Blynk.virtualWrite(V4,temp_ir);
   Blynk.virtualWrite(V6,h1);
   Blynk.virtualWrite(V7,t1);
   Blynk.virtualWrite(V2,h2);
   Blynk.virtualWrite(V3,t2);
}

void setup()
{
  // Debug console
  Serial.begin(115200);

  mlx.begin();

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080
  );
```

```
  dht1.begin();
  dht2.begin();

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

References

[1] Sharma, Nonita, et al. "A smart ontology-based IoT framework for remote patient monitoring." Biomedical Signal Processing and Control 68 (2021): 102717.

[2] Pandey, Honey, and S. Prabha. "Smart health monitoring system using IOT and machine learning techniques." 2020 sixth international conference on bio signals, images, and instrumentation (ICBSII). IEEE, 2020.

[3] Krishnan, D. Shiva Rama, Subhash Chand Gupta, and Tanupriya Choudhury. "An IoT based patient health monitoring system." 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, 2018.

[4] Victor Hugo, C., and K. S. Hareesha. IoT in healthcare and ambient assisted living. Eds. Gonçalo Marques, and Akash Kumar Bhoi. Springer, 2021.

[5] Poongodi, M., et al. "Smart healthcare in smart cities: wireless patient monitoring system using IoT." The Journal of Supercomputing 77.11 (2021): 12230-12255.

[6] Athira, A., T. D. Devika, and K. R. Varsha. "Design and development of IOT based multi-parameter patient monitoring system." 2020 6th Interna-

tional Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020.

[7] Tamilselvi, V., et al. "IoT based health monitoring system." 2020 6th International conference on advanced computing and communication systems (ICACCS). IEEE, 2020.

[8] Yeri, Vani, and D. C. Shubhangi. "IoT based real time health monitoring." 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2020.