Department Of Information Technology

Academic Term Jan-May 2021

Class: TE IT (Sem VI)

Subject: Wireless Sensor Network Lab Project

Title of the Project	**Temperature Monitoring System.**
Date Of Performance	
Date Of Submission	
Roll No (Group members)	8652 8691 8773
Name Of The Student (Group members)	Mahesh Dattatraya Babar (8652) Rahul Pujari (8691) Chaitanya Chandrakant Mohane (8773)

Evaluation:

Sr. No	**Rubric**	**Grade**
1	Timeline(2)	
2	Completeness(5)	
3	Project specific Features (9)	
4	Total (10)	

Signature of Teacher:

Title: Temperature Monitoring System.

Abstract:

Places where it's difficult to go and get temperature readings on daily basis, this project solves this problem with very minimal components and gets accurate readings remotely.

Requirements:

1. Computer which can run Arduino IDE.
2. Nodemcu Board.
3. LM35 (Temperature Sensor).
4. Neo 6m (GPS sensor)
5. Jumpers (male-male, female-female, male-female).
6. Breadboard.
7. Hotspot having active Internet Connection.

Features:

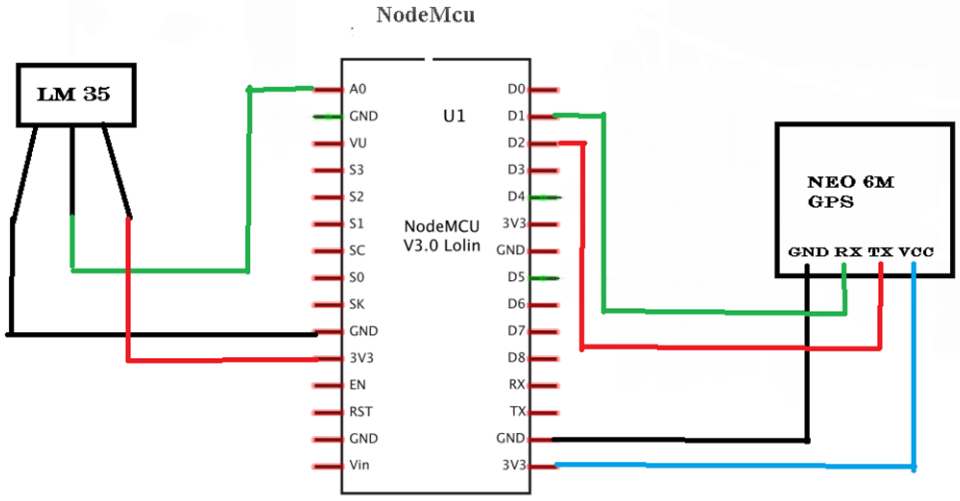
1. **Detects GPS location in latitude and longitude.**
2. **Accurate Temperature Readings.**
3. **Sensor reading are updated to cloud database in Realtime.**
4. **Once program loaded in nodemcu can be fixed at any location and taken readings remotely.**

Description of the Project

1. Proposed methodology:

- **LM35 and neo-6m are connected to nodeMCU, this way readings are taken.**
- **Firebase Realtime database is used to store data.**
- **Arduino IDE is used to configure sensors and get readings.**
- **Libraries are added as**
  #include <TinyGPS++.h> for GPS connection and reading.
  #include <ESP8266WiFi.h> for Wi-Fi connection.
  #include <SoftwareSerial.h> for serial communication on digital pins
  #include <FirebaseESP8266.h> for firebase database.

2. Proposed diagram:



3. Working:

- ## LM35 (Temperature Sensor):

  Connections:
  GND to GND on board.
  VCC to 3V3 on board.
  Data to A0 on Board.

  LM35 is a **temperature measuring** device having an analog output voltage proportional to the **temperature**.
  It provides output voltage in Centigrade (Celsius).
  It does not require any external calibration circuitry. The sensitivity of **LM35** is 10 mV/degree Celsius.

- ## Neo 6m (GPS Sensor):

  The **NEO-6M** GPS module is a well-performing complete GPS receiver with a built-in 25 x 25 x 4mm ceramic antenna, which provides a strong satellite search capability. With the power and signal indicators, you can monitor the status of the module

  Connections:
  GND to GND on board.
  VCC to 3V3 on board.
  Rx to D1 on board.
  Tx to D2 on board.

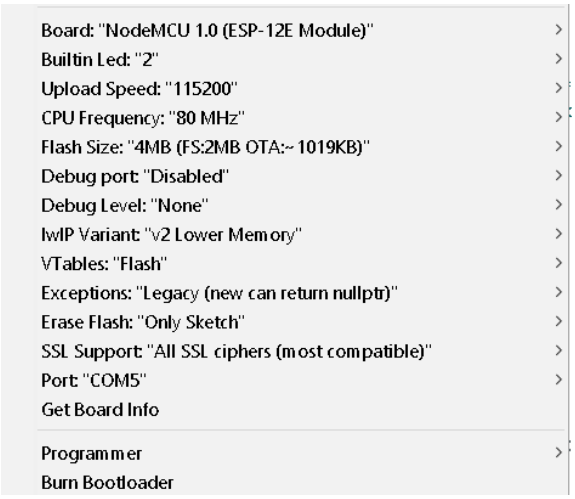
- The readings are taken and stored in variables; all the readings are uploaded to Firebase cloud Realtime database using
  #include <FirebaseESP8266.h> Library.

- As the readings get updated, also the database gets updated in no time.

- Once WIFI connection is successful , data will keep updating on changing.

4. Steps to execute the project:

- Check whether all the connections are done correct.
- Select port COM5.
- Select Board: NodeMcu.
- Crosscheck all the details shown in below.
- Compile the code written in Arduino IDE.
- On successful compilation of code, upload the code to NodeMcu board.



Result Analysis:

Code Snippets:

```
#include <TinyGPS++.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <FirebaseESP8266.h>



#define FIREBASE_HOST "temperature-monitoring-s-affc4-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "cC5iA7Knar1Vd1Yublk4lAbTT1jYhx5Zb15vC9F9"

FirebaseData firebaseData;

float vref = 3.3;
float resolution = vref/1023;

const char *ssid =  "Chaitanya";
const char *pass =  "123456879";
WiFiClient client;

static const int RXPin = 4, TXPin = 5;   // GPIO 4=D2(conneect Tx of GPS) and GPIO 5=D1(Connect Rx of GPS)
static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps; // The TinyGPS++ object
SoftwareSerial ss(RXPin, TXPin);  // The serial connection to the GPS device
```

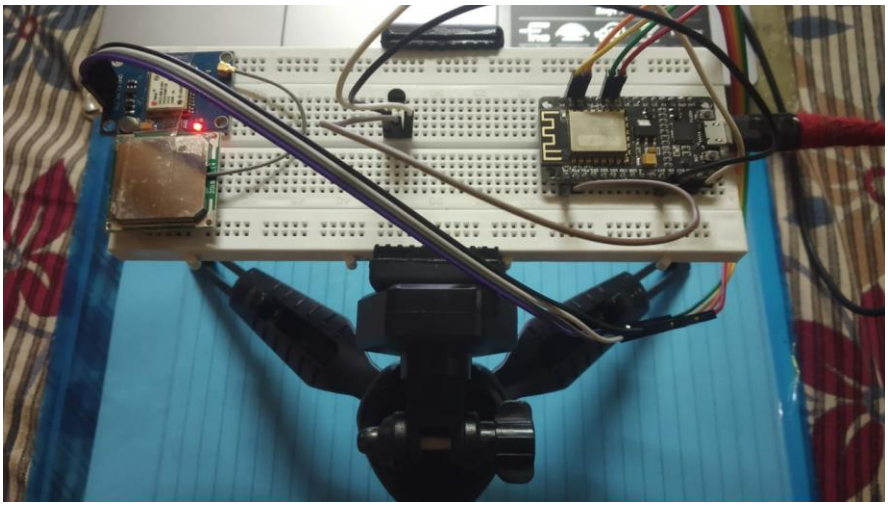
```
void setup()
{
      Serial.begin(9600);
      delay(1000);

      Serial.println("Connecting to ");
      Serial.println(ssid);

      WiFi.begin(ssid, pass);
      while (WiFi.status() != WL_CONNECTED)
        {
          delay(500);
          Serial.print(".");
        }
     Serial.println("");
     Serial.println("WiFi connected");

     ss.begin(GPSBaud);

     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

}
void loop() {

   while (ss.available() > 0)
```

```
    {
      // sketch displays information every time a new sentence is correctly encoded.
      if (gps.encode(ss.read()))
        displayInfo();

}

}
void displayInfo()
{
  float temperature = analogRead(A0);
  if (gps.location.isValid() )
  {
    float latitude = (gps.location.lat());     //Storing the Lat. and Lon.
    float longitude = (gps.location.lng());

    Serial.print("LAT:  ");
    Serial.println(latitude, 6);  // float to x decimal places
    Serial.print("LONG: ");
    Serial.println(longitude, 6);


     temperature = (temperature*resolution);
     temperature = temperature*100;
     Serial.print("TEMP: ");
     Serial.println(temperature);
```

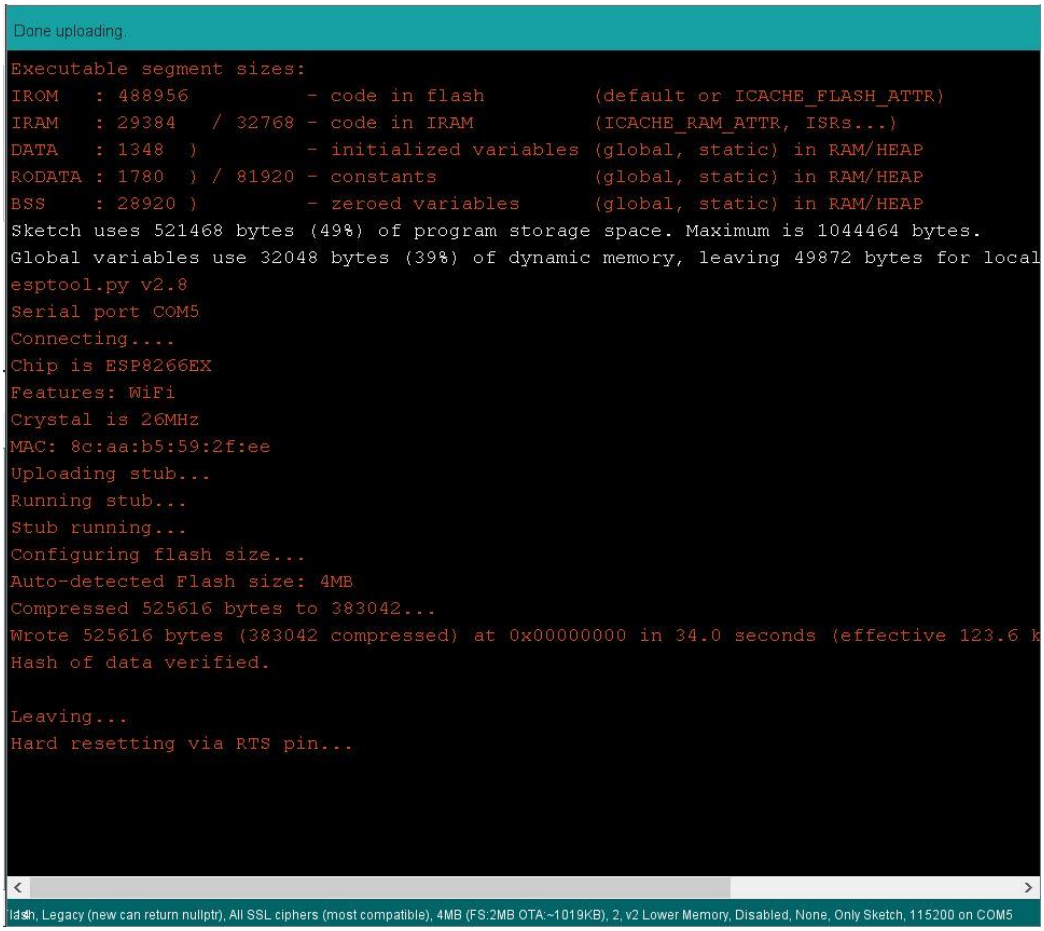
```
    delay(2000);

    Firebase.setFloat(firebaseData,"/Temperature",temperature);
    Firebase.setFloat(firebaseData,"/Latitude",latitude);
    Firebase.setFloat(firebaseData,"/longitude",longitude);
  }

  Serial.println();
}
```

Output screenshots (Model and Cloud Data):

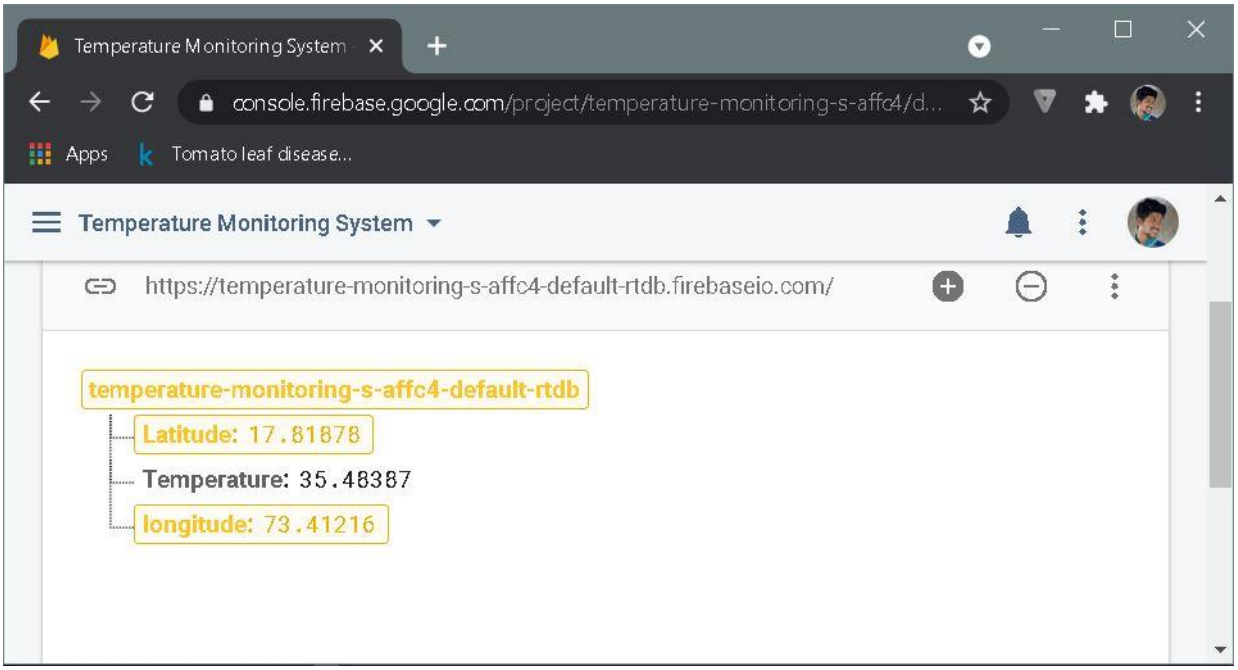
1.Model:

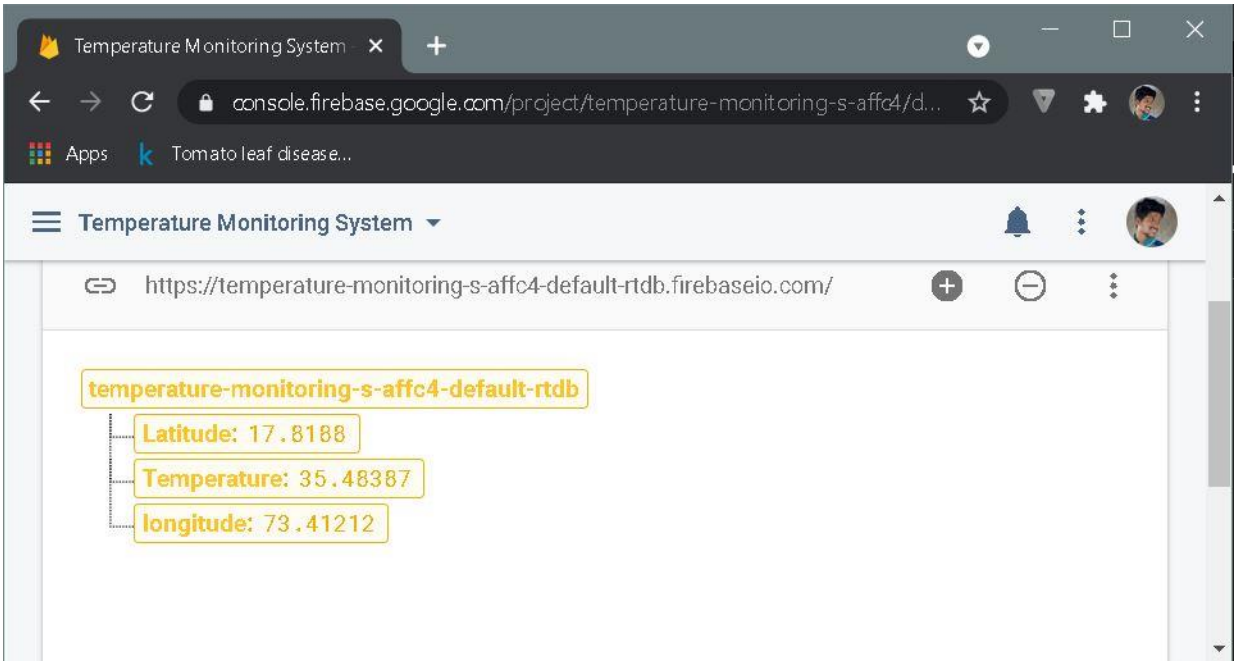


2.Successful Compilation and code uploaded to NodeMcu:



Firebase Data getting updated:





Radings are printed on serial monitor before updating;

Conclusion:

In this way, we worked on a project which is very helpful in areas where its not possible to go daily and check for temperature readings.

Future Scope:

- **Can be mounted on Train or vehicle.**
- **A GUI interface can be developed to make it more user friendly.**
- **Will be very useful to monitor places remotely.**

References:

https://firebase.google.com/docs/database

https://www.javatpoint.com/iot-project-google-firebase-nodemcu

https://www.nodemcu.com/index_en.html#fr_5475f7667976d8501100000f

Project Video Link :

https://drive.google.com/file/d/1vix30UqP6ESd4KYhRKPXJQzCUwANl6Q1/view?usp=sharing