

# Project Report ECN-351

## Temperature Monitoring and Remote Display System

*Raj Vardhan, 21116079*

*Prateek Verma, 21116074*

### AIM:

- The objective of this project is to design and build a complete sensor data acquisition system that captures data from sensor, process it through signal conditioning and digital conversion, and displays the data on a user interface such as mobile or laptop.

### Components used for the project

- LM35 sensor, IC324(op amp), resistors, diodes, capacitors, ESP32(uC)
- Tech-stack: Arduino ide, VS code, Google Firebase

### Designed Circuit

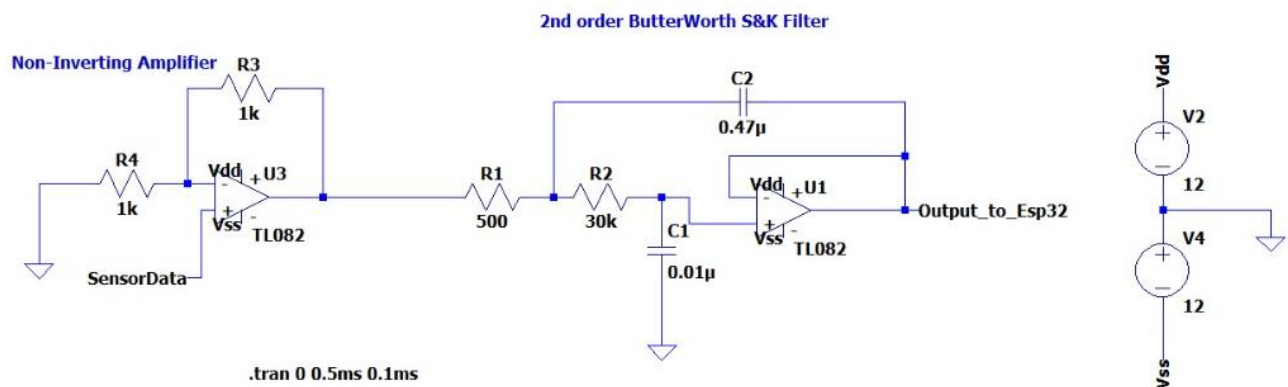
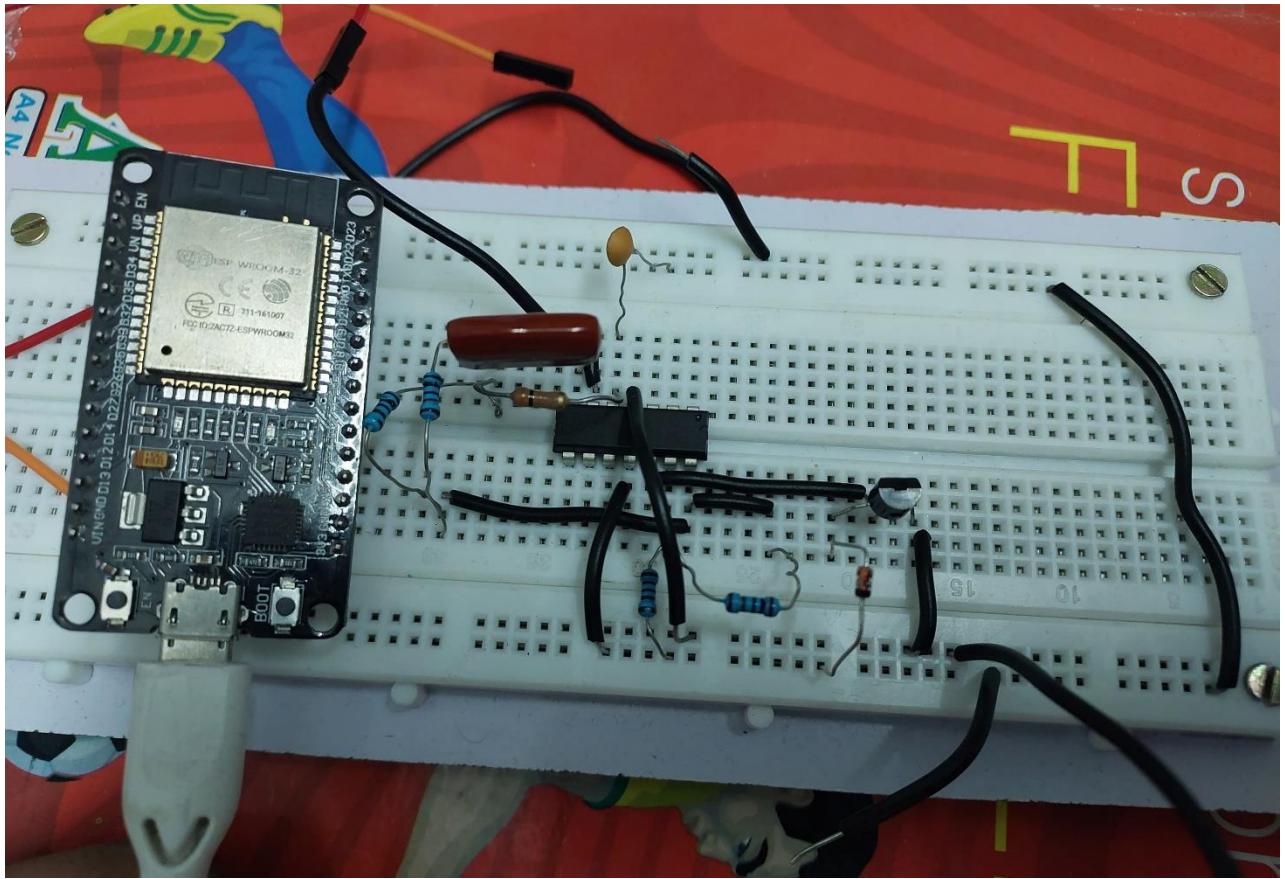
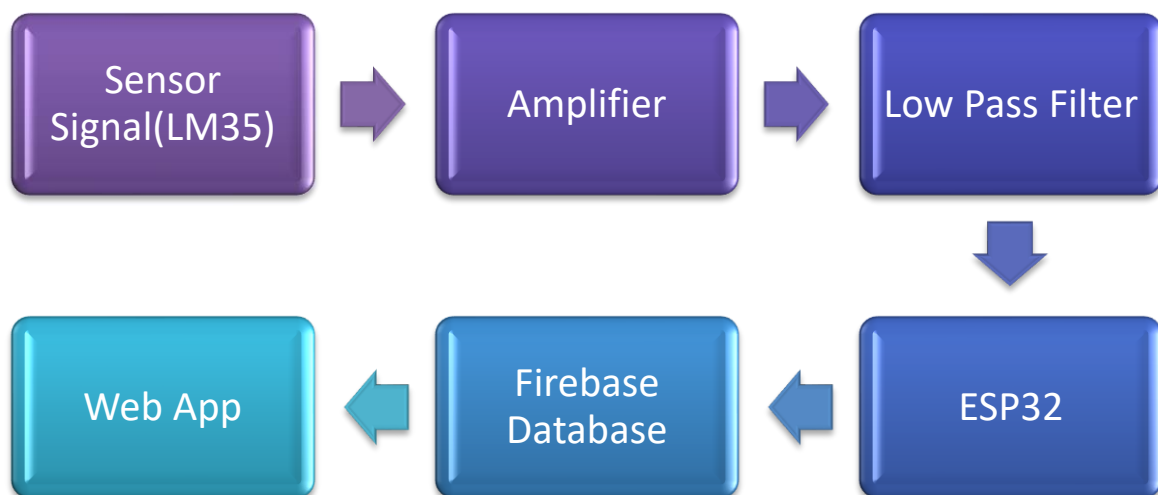


Fig.1. Lt spice Simulation of design



**Final Implemented Circuit**

## Signal Flow Graph



## System Architecture Breakup

**Signal Conditioning :** It include Amplifier and Low Pass Filter

**Amplifier :** used for amplification of the signal from the LM35 IC. The obtained gain is 2 with the taken resistances shown in fig.1.

**Low Pass Filter :** We used Sallen Key Low pass filter with 3dB bandwidth of 700Hz, with the used resistances and capacitances are shown in fig.1.

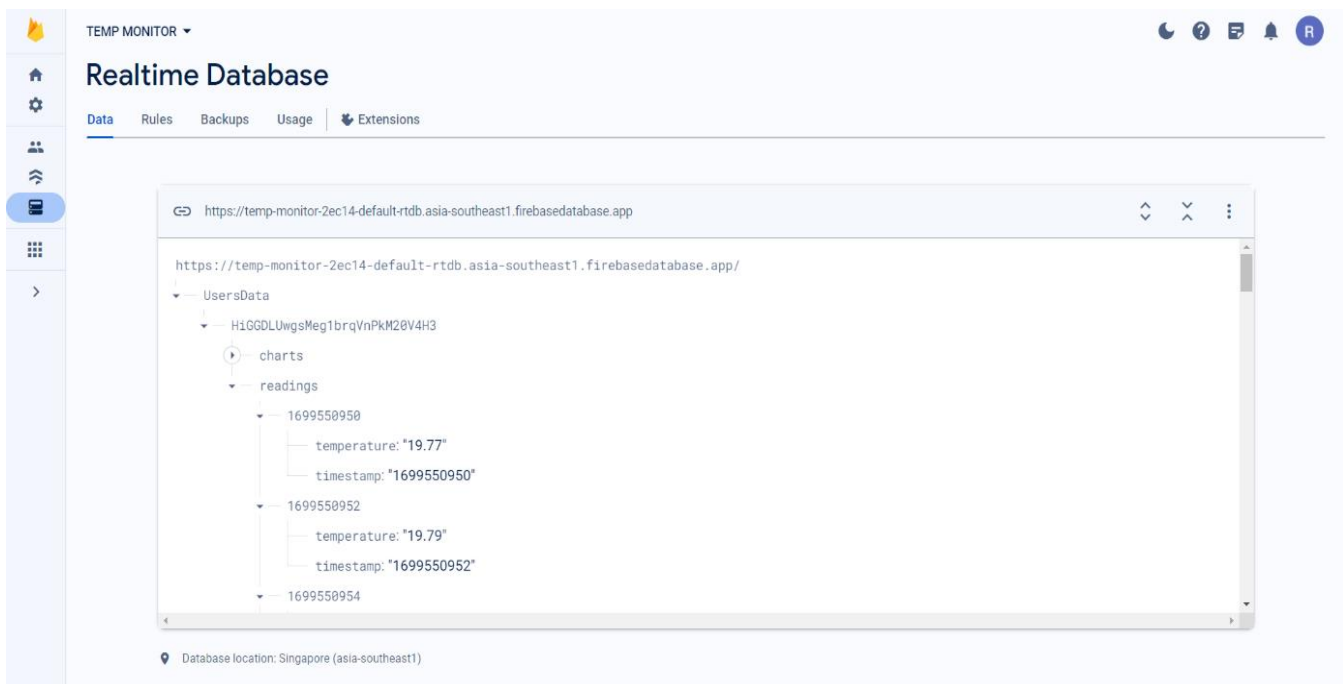
**Analog-to-Digital Conversion(ADC) :** This is done using ESP32. Every analog data is encoded into 12 bits.

**Sensor Interface :** Filter output is connected to D36 pin of ESP32 which then convert it to Digital signal. We upload the software code enabled with formulas to do necessary calculation and manipulate data to send it to database.

## What we have implemented

**Internet Connectivity :** we provide SSID and password of the network in the ESP32 code using which it gets connected to the internet.

**Data storing on Web App database :** After internet connectivity, ESP32 use email and password of the system user to get its UID(user id). Now it encodes sensor data with this UID and sends this data to the database.

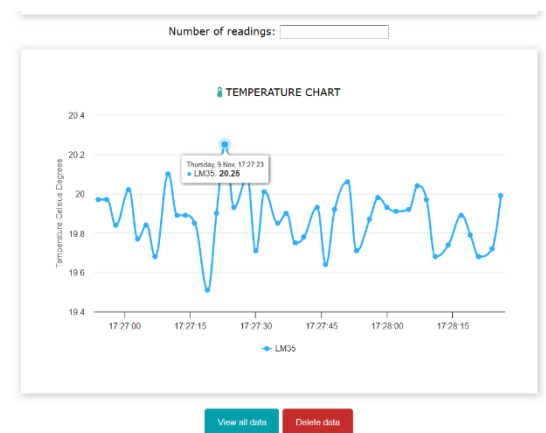
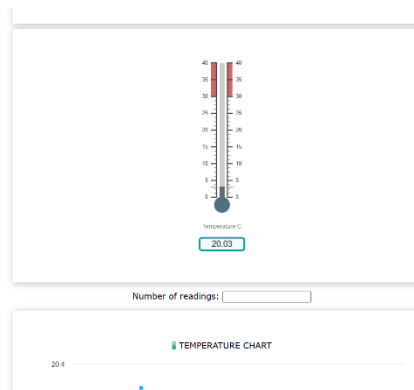
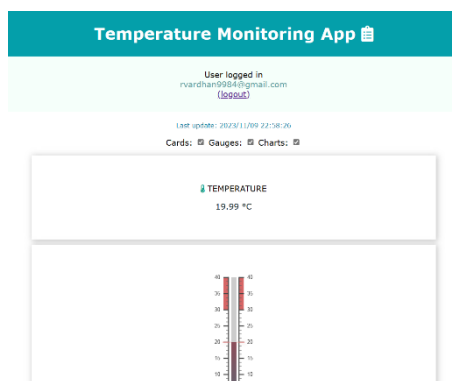


Temperature data with time Stamp

**Data Displaying on Web App:** we created a web app onto which only a registered user can log in. The web app gets data from its database and we represent it using graphs, gauge and current temperature value.

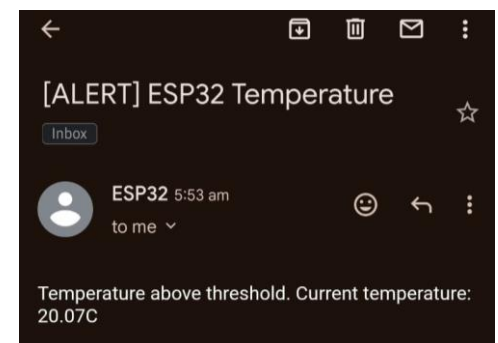
### Additional functionality

- we can set number of data points to be plotted on the graph.
- All saved data points can be seen at the bottom by clicking on view all data.
- At any point all previous data by selecting delete data



## Alert Notification on Email

- Alert notifications are sent to email when the temperature exceeds predefined threshold.



## Remote access and control of the system through the internet.

- We can set the threshold level for the temperature from the webpage.

## Key points Addressed:

1. **Data uploading:** set correct API key of the firebase project, database URL, user login credentials in ESP32 code. In Firebase project Realtime Database rules section set read and write to True.
2. In the web app html code use the project firebase configuration.

## Challenges faced:

### 1. Firebase Integration Challenges:

**Issue:** Encountered authentication errors and data upload failures while to send sensor data to the Firebase Realtime Database

**Solution:** Verified the Firebase project configuration and security rules, ensuring that the necessary permissions were granted for data transmission. Implemented secure token-based authentication and error handling to manage the authentication process effectively.

## 2. Web Page Development Challenges:

**Issue:** Faced difficulties in creating a dynamic web page for data visualization and real-time monitoring.

**Solution:** Employed a combination of HTML, CSS, and JavaScript to design an intuitive web interface. Utilized Firebase's JavaScript SDK to establish a connection with the Realtime Database, enabling seamless data retrieval and visualization. Implemented responsive design principles to ensure the web page's compatibility across various devices and screen sizes.

## Application

The developed system can be used for real time monitoring of temperature of any place remotely. It can be used in Laboratories, Healthcare, Data Centers, and Cold storage warehouses.

## References:

<https://randomnerdtutorials.com/esp32-email-alert-temperature-threshold/>

<https://www.electronicwings.com/esp32/lm35-sensor-interfacing-with-esp32>

<https://randomnerdtutorials.com/esp32-data-logging-firebase-realtime-database/>