IBM NAANMUDHALVAN

INTERNET OF THINGS-PHASE 5

FLOOD MONITORING AND EARLY WARNING SYSTEM

ABSTRACT:

The objective of this project is to create an Internet of Things (IoT) powered flood monitoring system near water bodies and flood-prone zones. This system will primarily concentrate on tracking water levels and issuing early flood warnings through a publicly accessible platform. The main goal is to enhance flood preparedness and response by guaranteeing that timely alerts are provided to both the general public and emergency response teams.

DESIGN THINKING:

1. **Project Objectives:**
    - A flood monitoring system for IoT (Internet of Things) devices is a critical application that can help detect and respond to flooding events in real-time.
    - Such a system typically involves a network of IoT sensors and devices that collect data related to water levels, weather conditions, and other relevant parameters.
    - IoT devices collect data continuously from the sensors that includes water level measurements and weather data.
    - Data transmission is through wireless communication protocols like WIFI and processing the collected data in real-time to detect potential flooding events.
    - Real Monitoring System trigger alerts to relevant stakeholders that can help mitigate the impact of flooding events, improve response times, and save lives and property.
2. **IoT Sensor Network Design:**

- The deployment of IoT sensor Network includes the use ultrasonic sensors, pressure sensors, or float sensors to measure the water levels in rivers, lakes, or flood-prone areas.
- Incorporating weather sensors to monitor rainfall, temperature, humidity, wind speed, and direction, as these factors can contribute to flooding.
- - Besides the selection of sensors, it also includes that to ensure that IoT devices have backup power sources (e.g., batteries) to operate during power outages.

3. **Early Warning Platform**:
    - Processing and analyzation the collected data id done in real-time to detect potential flooding events and implementation of algorithms and thresholds to trigger alerts when critical conditions are met.
    - When a flood event is detected, the system triggers alert to relevant stakeholders that can be sent via SMS, email, push notifications, or automated phone calls.
    - Finally, a user-friendly dashboard or interface for monitoring and visualization of data in real-time is developed and is used for the generation of reports and historical data for analysis and decision-making.
4. **Integration Approach:**
    - The sensors monitor, collect and transmit data to the microcontroller Esp32 through WIFI communication module.
    - The Esp32 can be configured and programmed in Arduino IDE through embedded C.
    - Deep Learning techniques and Data Analytics are used to improve flood prediction accuracy.
    - The data can be stored and accessed in Cloud by implementing strong security measures to protect data and prevent unauthorized access to the system and collaborating with local emergency services to ensure they receive flood alerts in real-time.

- Additionally, we can also incorporate the capability to remotely control certain actions such as opening or closing floodgates, activating pumps, or diverting traffic when necessary.

INNOVATION:

Step 1- Analyzation of flood prone areas

- Analysing flood-prone areas is a crucial aspect of flood monitoring systems, enabling the identification of regions at elevated risk of flooding and facilitating improved preparedness and response.
- Gathering of relevant data sources such as historical flood data, topographic data, land cover data, hydrological data, and rainfall data.
- Topographic Data includes Obtain Digital Elevation Models (DEMs) or topographic maps that provide elevation information for the study area.
- Collection of data on river and stream networks, including river gauges and streamflow measurements.
- Gathering of historical rainfall data, including records of intense rainfall events and past flood events in that area.

Step 2- Parameters Detection

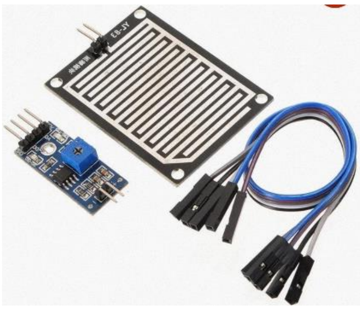
- Flood monitoring systems incorporate various parameters and data sources to accurately assess flood risks and provide early warnings.
- The parameters include weather conditions, rainfall, humidity, temperature, direction and speed of air flow, intensity and velocity of rainfall.

Step 3- Selection of Sensors

- Suitable sensors are selected according to their specifications and for calculating various parameters in a flood monitoring system is essential to ensure accurate data collection.
- The following sensors are employed for the flooding monitoring and early warning systems:
- Parameters:

1. Temperature – DHT22
2. Humidity – DHT22
3. Rainfall – YL83
4. Wind Speed and direction – ELA1877
5. Water level Indicator – Ultrasonic sensor

DHT22

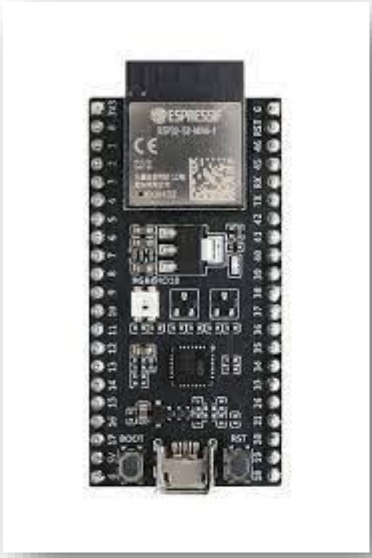


YL83

Ultrasonic sensor

Step 4 - Selection of Microcontroller

- For the integration of all these sensors to oversee environmental parameters, WIFI based the ESP8266 microcontroller is chosen.
    1. Advantages of utilizing the ESP8266 for Flood monitoring system provides a cost-effective solution.
    2. Built-in wireless Wi-Fi connectivity for remote monitoring over the flood prone areas.
    3. Low power consumption, suitable for battery-powered applications.
    4. Integration with a variety of sensors and peripherals.
    5. Capable of data logging and remote data transmission.



ESP32

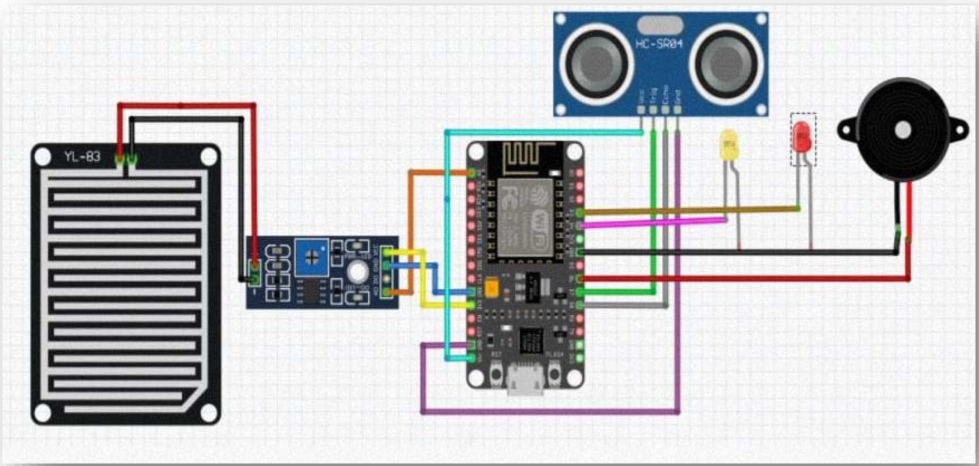


Fig 1. Circuit Connection of Rainfall sensor and Ultrasonic sensor with ESP32

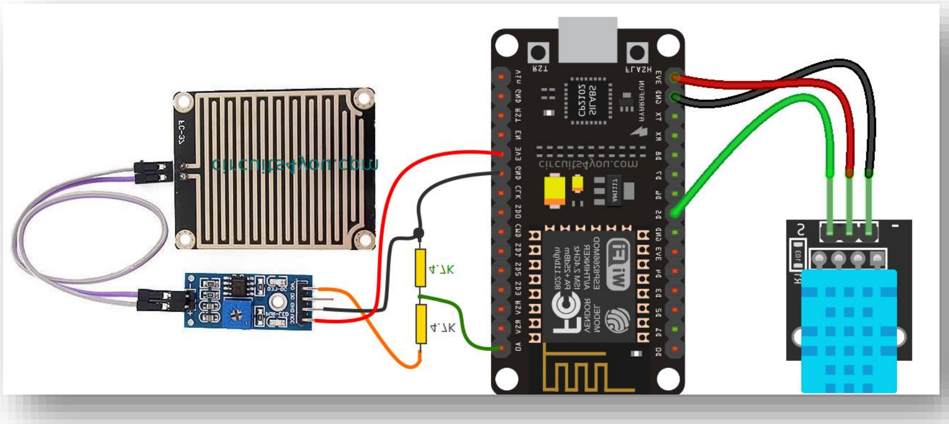


Fig 2. Circuit Connection of Rainfall sensor and Temperature sensor with ESP32

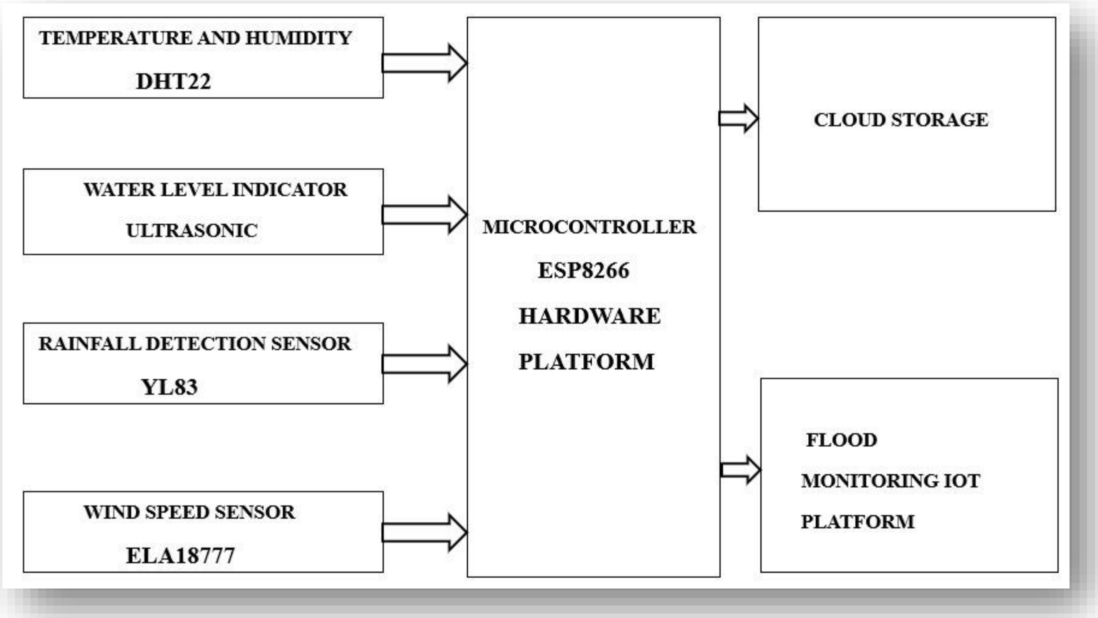
Step 5- Selection of Monitoring Platform

- A web-based platform has been chosen for monitoring and maintaining records of environmental parameters. This platform is accessible to the public for real-time updates and the generation of alert system is offered a setting a threshold for the parameters.
- We employ the Blynk 2.0 platform, known for its user-friendly display interface and a wide array of data visualization features, enhancing user experience and data comprehension.
- It helps to display the intensity of rainfall, water level and flood predictions.
- When exceeding the thresholds, it generates an alert to the control station via email notifications or text messages.

Step 6-Data Storage

- Data is initially collected from sensors and undergoes pre-processing.
- The processed data is transmitted to the cloud via WIFI protocol.
- This data is accessible remotely, eliminating the necessity for physical storage.
- Remote access facilitates the retention of historical flood records.

BLOCK DIAGRAM:



PREPROCESSING OF DATASET:

Steps Involved:

For the building the IoT based flood monitoring and early warning system,

- Deployment of IoT sensors (e.g., Ultrasonic Sensors) in flood-prone areas and configure them to measure water levels.
- Develop a Python script on the IoT sensors to send collected water level data to the early warning platform.

Sensor and Microcontroller:

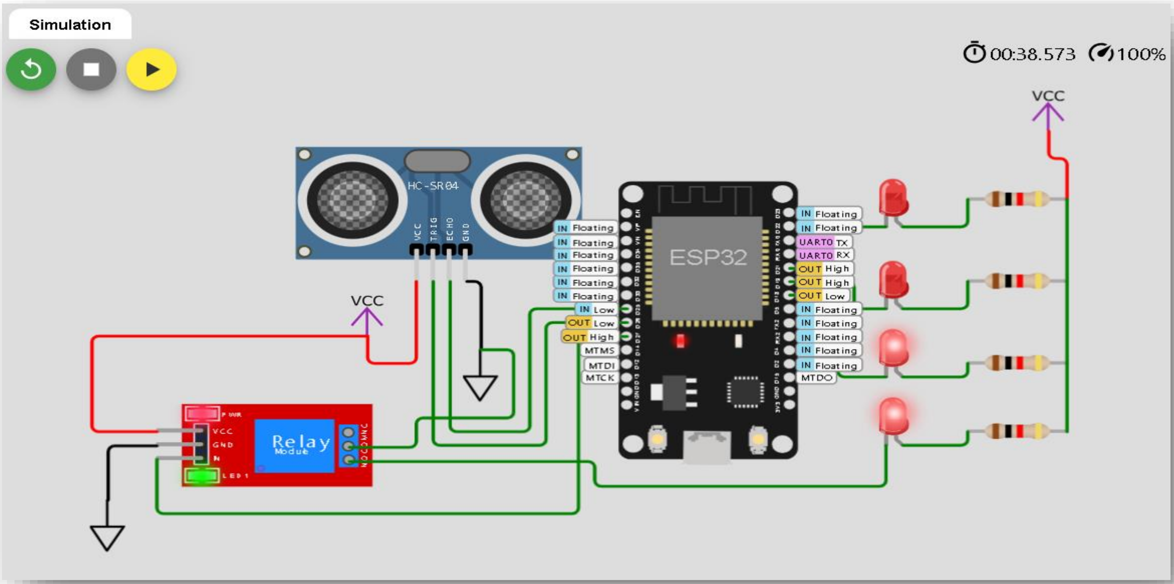
- Ultrasonic Sensor is connected with WIFI based ESP32 controller to measure the water levels.

- The alerts are generated through buzzer, LED and the data are transferred to the Firebase Cloud platform and are displayed in real-time database.

Working Principle of Ultrasonic sensor:

- The ultrasonic sensor contains a transducer that can both emit and receive ultrasonic sound waves. To measure the water level, the sensor emits a short burst of high-frequency ultrasonic sound waves (typically in the ultrasonic range, around 40 kHz).
- These sound waves travel through the air or any other medium until they encounter the water surface. When the emitted sound waves hit the water surface, they are partially reflected back toward the sensor. The time it takes for the sound wave to travel from the sensor to the water surface and back is measured.
- Once the distance to the water surface is known, you can calculate the water level by subtracting this distance from the sensor's mounting height above the water surface.
- Generation of alerts is for water level monitoring is based on setting specific thresholds. If the alert is generated when the water level rises beyond a certain point by programming the microcontroller (such as an ESP32) connected to the ultrasonic sensor to compare the calculated water level with the threshold.
- If the water level is below the threshold, no alert is generated.
- If the water level crosses the threshold, the system triggers an alert, which could be a visual indicator (LED), audible alarm, or a message is displayed in the firebase.

Circuit Connection in Wokwi simulation environment:



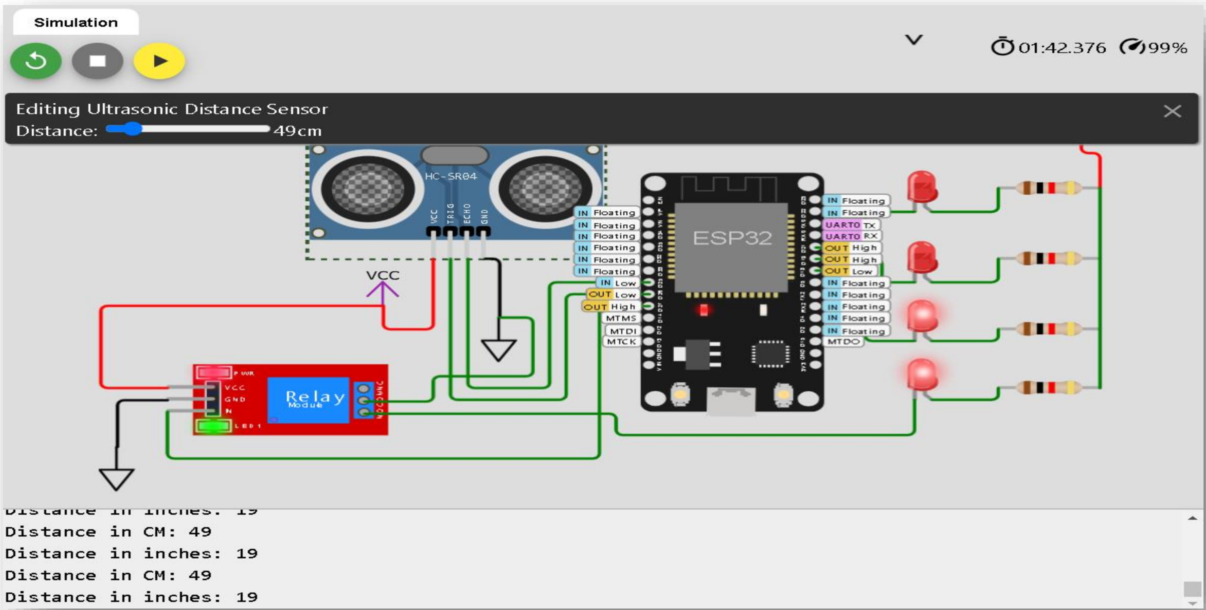


Fig 3. Water level below threshold

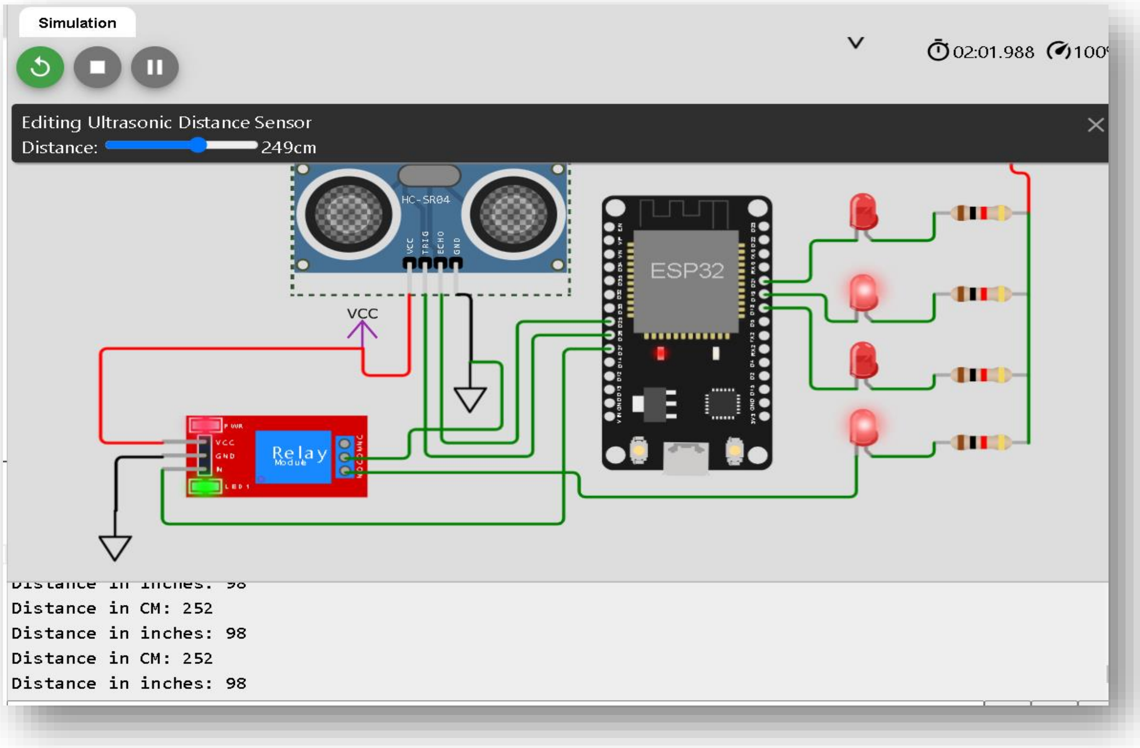
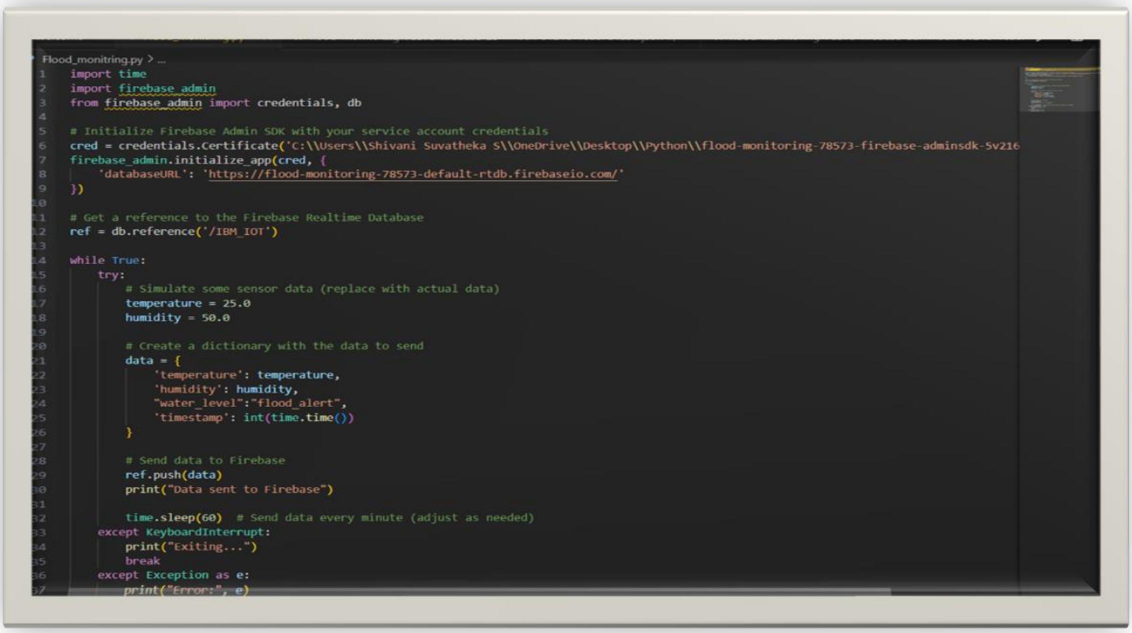


Fig 4. Water level above threshold

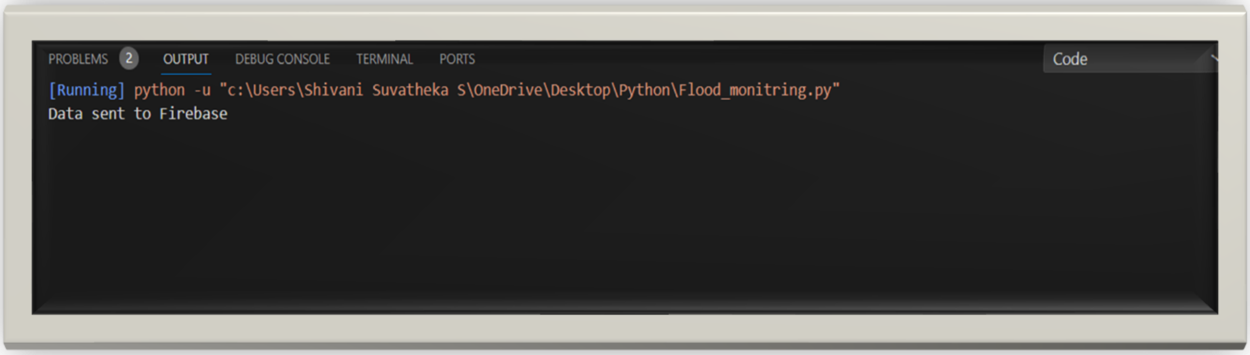
Python Code for Water level Measurement:



Flood_monitring.py > ...

```python
import time
import firebase_admin
from firebase_admin import credentials, db

# Initialize Firebase Admin SDK with your service account credentials
cred = credentials.Certificate('C:\\Users\\Shivani Suvatheka S\\OneDrive\\Desktop\\Python\\flood-monitoring-78573-firebase-adminsdk-5v216
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://flood-monitoring-78573-default-rtdb.firebaseio.com/'
})

# Get a reference to the Firebase Realtime Database
ref = db.reference('/IBM_IOT')

while True:
    try:
        # Simulate some sensor data (replace with actual data)
        temperature = 25.0
        humidity = 50.0

        # Create a dictionary with the data to send
        data = {
            'temperature': temperature,
            'humidity': humidity,
            "water_level":"flood_alert",
            'timestamp': int(time.time())
        }

        # Send data to Firebase
        ref.push(data)
        print("Data sent to Firebase")

        time.sleep(60)  # Send data every minute (adjust as needed)
    except KeyboardInterrupt:
        print("Exiting...")
        break
    except Exception as e:
        print("Error:", e)
```

Output:



MODEL TRAINING AND EVALUATION:

Data Storage and model training:

- The data collected from various sensors collecting the parameters that includes temperature, rain intensity and water level indication pays a way for predicting the possibility of occurrence of flood and the generation of alerting system during the response of the disaster.
- NEO 6M GPS module is used for location tracking and alerting systems. In the event of a flood onset, the device expeditiously transmits the gathered data to ESP32 (a WIFI based NodeMCU central microcontroller). Within the microcontroller, the data is subjected to validation and processing procedures to predict the possibility of the flood occurrence.

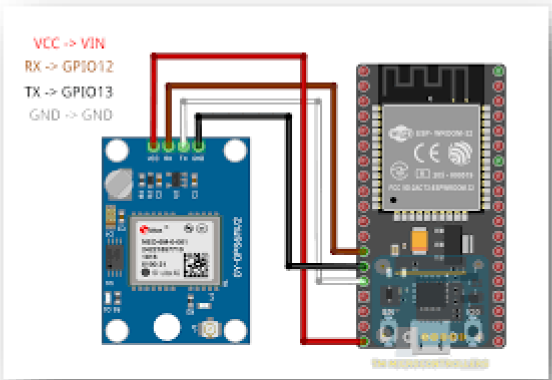


Fig 5. Connection of ESP32 with Neo 6M GPS

- The model is trained and the data are stored in the cloud platform called Firebase for the ease of later access of data.

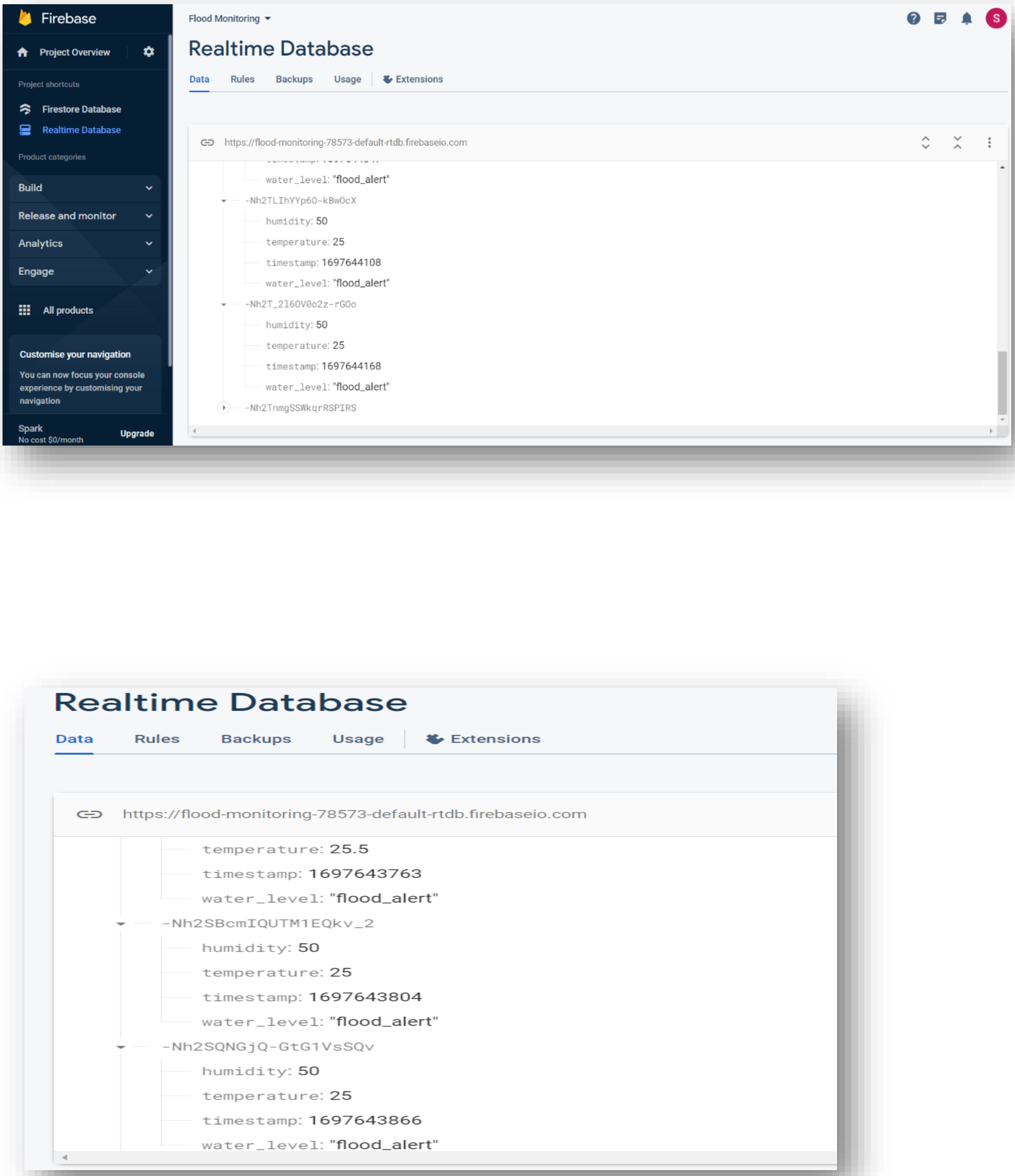


Fig 6. Model Training and Data storage in Firebase

Concept Validation and real-time implementation:

- Following the verification, an alert is promptly generated and conveyed to individuals residing in disaster-prone regions through user friendly mobile application via a variety of communication channels, thereby ensuring the timely dissemination of life-saving notifications via methods such as SMS, email, and other communication means.
- Blynk is a IOT based open-source application which has a user-friendly interface that displays the indication of various parameters such as water intensity and rainfall rate when flood occurs and the alerts are generated when the parameters exceed the thresholds provided.

Fig 7. Indication of water level and rain intensity

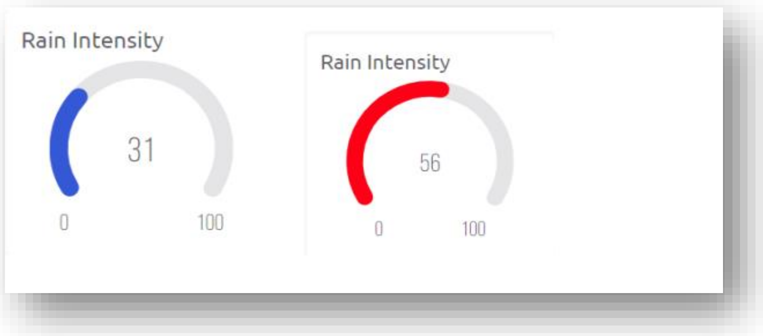


Fig 8. Indication of normal and critical level rain intensity

Fig 9. Indication of normal and critical water level

Fig 10. Alert generation and early warning notification through Blynk IOT

- Thus the project aims in developing a cost-effective and efficient smart flood monitoring system using NodeMCU and the Blynk application. This system leverages ultrasonic sensors to provide accurate data for flash flood detection and early warning.
- The wireless sensor node, built on the Blynk platform, offers flexibility and cost-effectiveness. By connecting ultrasonic sensors and a rain sensor to the NodeMCU, the system can accurately detect and monitor flooding conditions. It serves as a reliable solution for detecting, monitoring, and issuing community alerts in flood-prone areas.

- Moreover, the IoT system includes an accompanying application that enhances the overall functionality by offering real-time monitoring of fundamental environmental conditions. This element empowers users to monitor shifts in local conditions and receive periodic updates, thereby promoting flood preparedness and augmenting overall situational awareness.

THANK YOU