PROJECT STATEMENT : FLOOD MONITORING SYSTEM

TEAM MEMBERS:

KRISHNA PRIYA.S

KALAIMATHI.S

PABITHA.J

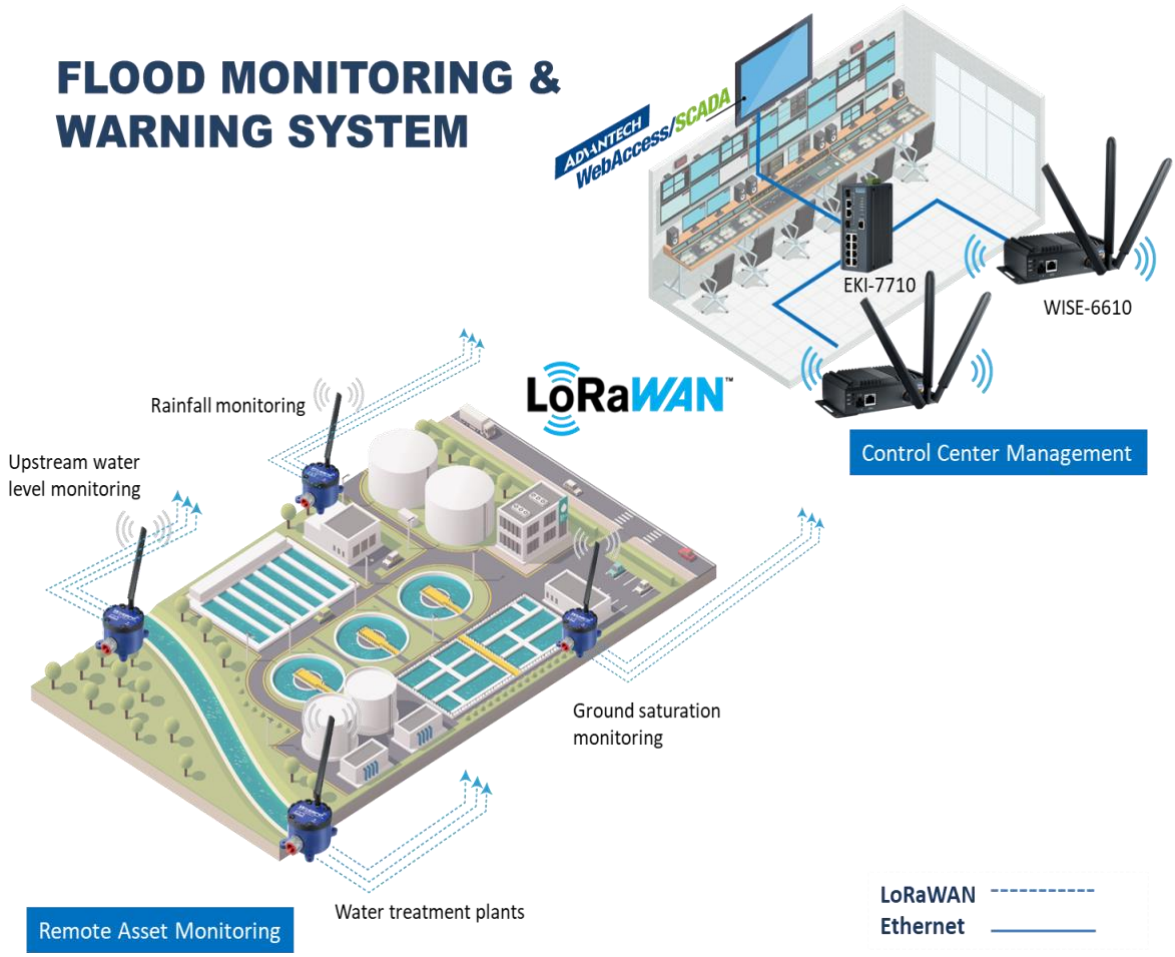
MADHAVI.M

ABINAYA.S

PHASE-2-PROJECT :

INTRODUCTION :

A flood monitoring system is used to monitor a rise in water levels. The system comprises sensors that are deployed in cities or any area of interest. The sensors can be connected to either the main electricity or can be solar-powered.



- The project involves deploying IoT sensors near water bodies and flood-prone areas to monitor water levels and provide early flood warnings through a public platform.

INNOVATION:

A smart computer system for the exploitation of hydro-meteorological and weather data captured to generate warnings and notifications for events that may involve a flood risk situation.

DEFINING OBJECTIVES :

A flood warning system is designed to warn and alert the owners. of the vehicle about the flood immediately. The system will detect water level through an. ultrasonic sensor and will automatically send a notification via IoT technology to the owner's. smartphone through the Blynk Application.

DESIGNING THE IoT SENSOR NETWORK:

- The main goal was to assess the optimal sites needed for flood monitoring and prediction and assess the representativeness of the selected locations following the previous historical floods.

DESIGN DETAILS:

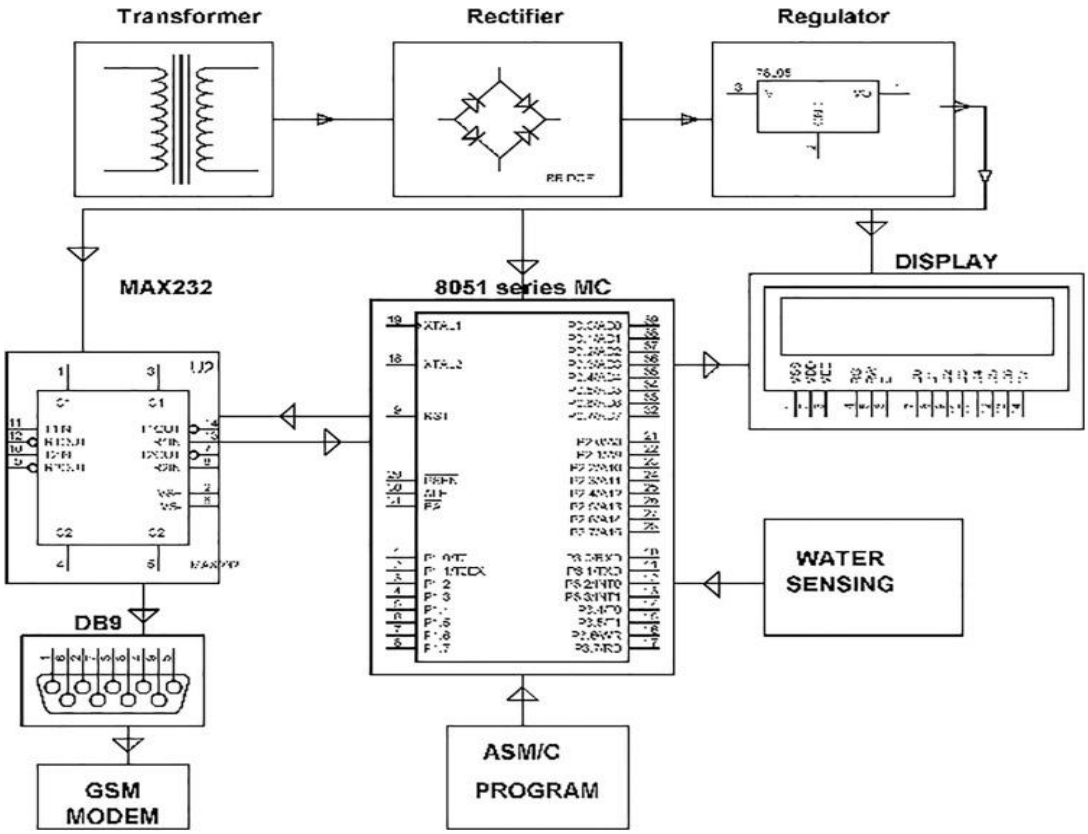
- The sensors collect data on water levels, rainfall, and other environmental factors.
- The communication devices transmit the data from the sensors to the central location.
- The central location collects and analyzes the data.
- If the data indicates that there is a risk of flooding, the central location generates an early warning.
- The early warning is then transmitted to residents and authorities through a variety of channels, such as text messages, email, or social media.

Innovative Real-Time Transit Information Platform:

- **AI-powered flood prediction**: AI could be used to analyze data from a variety of sources, including sensors, weather forecasts, and social media, to predict the likelihood of flooding in a given area. This information could then be used to generate more accurate and timely flood warnings.
- **Real-time transit information**: The platform could provide real-time information on the status of public transportation, roads,

and other transportation infrastructure during a flood event. This would help people to plan their travel and avoid flooded areas.

- **Social media integration**: The platform could be integrated with social media platforms to allow people to share flood warnings and other information with their friends and family. This would help to spread the word about flood warnings and get people to safety more quickly.



In addition to these specific features, an innovative real-time transit information **platform for a flood monitoring system should also be designed to be**:

- **Easy to use**: The platform should be easy to use for people of all ages and technical abilities.
- **Accessible**: The platform should be accessible to people with disabilities.

- **Multilingual**: The platform should be available in multiple languages to reach as many people as possible.

  Integration features on a flood monitoring system can allow it to interact with other systems and devices to provide more comprehensive and accurate information. Some examples of integration features include:

- **Integration with weather forecasting systems**: This would allow the flood monitoring system to access and process real-time weather data, such as rainfall forecasts and radar imagery. This information could then be used to generate more accurate and timely flood warnings.
- **Integration with hydrological models**: Hydrological models simulate the movement of water through the environment. Integrating a flood monitoring system with hydrological models would allow it to predict how a flood will develop over time and to identify areas that are at risk.
- **Integration with social media**: This would allow the flood monitoring system to share flood warnings and other information with the public through social media platforms. It would also allow the system to collect information from social media, such as reports of flooding, to improve its accuracy.
- **Integration with emergency management systems**: This would allow the flood monitoring system to share information with emergency management personnel, such as the location and severity of flooding. Emergency management personnel could then use this information to coordinate their response efforts.

❖ FLOOD MONITORING SYSTEM FEATURES:

Sensors: Sensors are deployed in areas at risk of flooding to measure water levels, rainfall, and other relevant data. Sensors can be wired or wireless, and they can be powered by solar panels or batteries.

Data transmission: The sensors transmit data to a central server or cloud-based platform. This can be done using a variety of communication technologies, such as cellular, satellite, or radio.

Data analysis: The central server or cloud-based platform analyzes the data from the sensors to detect rising water levels and other potential flood hazards.

Alerts: The system generates alerts and warnings when flood hazards are detected. Alerts can be sent to emergency responders, government officials, and the public via a variety of channels, such as text messages, email, social media, and sirens.

Integration with other systems: Flood monitoring systems can be integrated with other systems, such as early warning systems and emergency management systems. This allows for a more coordinated response to flood hazards.

Tittle: Innovative Design For Flood Monitoring System

```
Import time

Import board

Import busio

Import adafruit_ultrasonic

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import smtplib


# Create an ultrasonic sensor object

Sensor = adafruit_ultrasonic.Ultrasonic(board.D4, board.D3)
```

```
# Set the threshold for the water level
Threshold = 10

# Create a list to store the water level data
Water_level_data = []

# Create a function to send an email alert
Def send_email_alert(message):
    Sender = flood.monitoring.system@example.com
    Receiver = user@example.com
    Subject = “Flood alert!”
    Body = message

    Msg = f”Subject: {subject}\n\n{body}”

    Smtp_server = smtplib.SMTP(“smtp.example.com”, 587)
    Smtp_server.starttls()
    Smtp_server.login(sender, “password”)
    Smtp_server.sendmail(sender, receiver, msg)
    Smtp_server.quit()
```

```
# Start monitoring the water level
While True:

    # Get the current water level
    Current_water_level = sensor.range

    # Add the current water level to the list
    Water_level_data.append(current_water_level)

    # Check if the water level has crossed the threshold
    If current_water_level < threshold and len(water_level_data) >= 2:

        # Calculate the change in water level
        Change_in_water_level = water_level_data[-1] –
water_level_data[-2]

        # If the change in water level is positive and greater than a certain
threshold, send an alert
        If change_in_water_level > 0.1:

            # Send an email alert
```

```
        Send_email_alert("The water level is rising rapidly!")


    # Plot the water level data
    Plt.plot(water_level_data)
    Plt.xlabel("Time (seconds)")
    Plt.ylabel("Water level (cm)")
    Plt.title("Flood Monitoring System")
    Plt.draw()
    Plt.pause(0.1)


    # Save the water level data to a file
    Df = pd.DataFrame(water_level_data, columns=["Water level"])
    Df.to_csv("water_level_data.csv", index=False)


    # Forecast the water level for the next hour
    # This can be done using a variety of forecasting methods, such as
ARIMA or exponential smoothing
    # Here, we are using a simple linear regression model
    Forecast = water_level_data[-1] + 0.1 * time.time()


    # Print the forecast water level
```

```
Print("Forecast water level for the next hour:", forecast)


# Wait for 1 second before checking the water level again

Time.sleep(1)
```

It stores the water level data to a file, which can be used for further analysis.

It forecasts the water level for the next hour, which can be used to identify areas that are at risk of flooding.

The output of this code will vary depending on the actual water level data that is collected. However, the following is an example of some possible results:

The water level data can be analyzed to identify trends and patterns, which can be used to develop flood risk models.

The forecast water level can be used to identify areas that are at risk of flooding in the next hour, and to evacuate people from these areas.

This code can be customized to meet the specific needs of a flood monitoring system. For example, the email alert can be customized to include additional information, such as the current water level, the forecast water level, and the location of the sensor. Additionally, the forecasting method can be changed to use a more sophisticated forecasting model, such as ARIMA or exponential smoothing.

Thank you