(IOT\_PHASE 3)

**TO DEVELOP A PYTHON SCRIPT FOR SENDING COLLECTED WATER LEVEL DATATL TO AN EARTHLY WARNING PLATFORM IN AN IOT(INTERNET OF THINGS) AND DEVELOPING IOT SENSOR IN FLOOD -PRONE AREAS TO MEASURE WATER LEVEL S:-**

**TO DEVELOP A PYTHON SCRIPT FOR SENDING COLLECTED WATER LEVEL DATATL TO AN EARTHLY WARNING PLATFORM IN AN IOT(INTERNET OF THINGS) :**

1. Setup Hardware:

- Connect your water level sensor to your IoT device (e.g., Raspberry Pi or Arduino) and ensure that it’s properly calibrated and collecting data.

2. Install Required Libraries:

- Install any necessary libraries for your hardware and communication protocols. For example, you might need libraries for working with sensors (e.g., RPi.GPIO or Adafruit CircuitPython) and communication protocols (e.g., MQTT for IoT communication).

3. Collect Data:

- Read the water level data from the sensor at regular intervals.

4. Format the Data:

- Prepare the collected data in a format suitable for transmission to the Earthly warning platform. This format may vary depending on the platform’s requirements.

5. Send Data to Earthly Warning Platform:

- Use the appropriate protocol and APIs to transmit the data to the Earthly warning platform. This could be through HTTP, MQTT, or other methods depending on the platform’s supported protocols.

6. Handle Errors and Exceptions:

- Implement error handling to deal with issues like network connectivity problems or platform API errors.

python

Import time

Import requests # For HTTP communication with the platform

Import your\_sensor\_library # Import the library for your water level sensor

# Configuration for your sensor and platform

Sensor\_pin = 17

Earthly\_platform\_url = <https://earthly-warning-platform-api-url.com>

Api\_key = “your\_api\_key”

# Initialize the sensor (replace with your specific sensor code)

Sensor = your\_sensor\_library.setup(sensor\_pin)

While True:

# Read water level data from the sensor

Water\_level = sensor.read\_data()

# Prepare data in the desired format

Data\_to\_send = {

“water\_level”: water\_level,

“timestamp”: time.time(),

}

Try:

# Send data to Earthly Warning Platform (replace with appropriate API call)

Response = requests.post(

F”{earthly\_platform\_url}/send\_data”,

Json=data\_to\_send,

Headers={“Authorization”: f”Bearer {api\_key}”}

)

If response.status\_code == 200:

Print(“Data sent successfully.”)

Else:

Print(f”Error: {response.status\_code} – {response.text}”)

Except Exception as e:

Print(f”Error: {str€}”)

Time.sleep(60) # Adjust the interval as needed

DEVELOPING IOT SENSOR IN FLOOD -PRONE AREAS TO MEASURE WATER LEVELS:

1. Select IoT Sensors:

- Choose appropriate water level sensors that are suitable for the environmental conditions in flood-prone areas. These sensors should be designed to withstand exposure to water and provide accurate measurements.

2. Hardware Setup:

- Set up the IoT sensors in the selected flood-prone areas. Ensure they are securely mounted to withstand flooding and other environmental factors. Power the sensors using a reliable source, such as solar panels and batteries.

3. Connect to IoT Devices:

- Connect the water level sensors to IoT devices like Raspberry Pi, Arduino, or specialized IoT gateways. These devices will collect data from the sensors.

4. Install Communication Modules:

- Equip the IoT devices with communication modules, such as cellular modems, Wi-Fi, or LoRa, to transmit data to a central server or platform. Ensure that these devices have internet connectivity.

5. Data Collection and Processing:

- Develop or configure software on the IoT devices to collect and process data from the water level sensors. This may involve calibration, data filtering, and formatting.

6. Data Transmission:

- Set up protocols (e.g., MQTT, HTTP) for transmitting data from the IoT devices to a central server or cloud platform. Implement security measures to protect data during transmission.

7. Central Data Platform:

- Create a central data platform or use an existing IoT platform to receive, store, and analyze the collected data. This platform should be capable of handling data from multiple sensors.

8. Data Visualization and Analysis:

- Develop a dashboard or interface for monitoring and analyzing the water level data in real-time. Implement algorithms for flood prediction and early warning systems.

9. Alerting System:

- Integrate alerting mechanisms into your platform to notify authorities and residents when water levels reach critical thresholds. Alerts can be sent via SMS, email, or other communication channels.

10. Maintenance and Monitoring:

- Regularly maintain and monitor the deployed IoT sensors to ensure they are functioning correctly. Replace batteries and sensors as needed.

11. Community Engagement:

- Engage with the local community and authorities to ensure they are aware of the monitoring system and understand how to respond to alerts.

12. Regulatory Compliance:

- Ensure compliance with any relevant regulations or permits for deploying sensors in flood-prone areas.