FLOOD LEVEL MONITORING USING THINGSPEAK CLOUD

A Project Report for Industrial Training and Internship

submitted by

Maharghya Mukherjee
Ritankar Roy
Santam Banik
Debankur Sen
Saswata Haldar

In the partial fulfillment of the award of the degree of

B.Tech

in the

Electronics and Communication Engineering

Of

Budge Budge Institute of Technology

At

Ardent Computech Pvt. Ltd.

CERTIFICATE FROM SUPERVISOR

This is to certify that **“Debankur Sen, 27600323066**” have completed the project titled "Flood Level Monitoring using “ThingSpeak Cloud” under my supervision during the period from “27/08/2025” to “27/10/2025” which is in partial fulfillment of requirements for the award of the **B.Tech** degree and submitted to the Department of “**Electronics and Communication Engineering**” of **“Budge Budge Institute of Technology”**.

_____

Signature of the Supervisor

Date: 27/10/25

Name of the Project Supervisor:

Ardent Computech Pvt. Ltd.
Drives you to the Industry
Module 132, SDF Building, Sector V, Salt Lake, Pin - 700091
www.ardentcollaborations.com

BONAFIDE CERTIFICATE

Certified that this project work was carried out under my supervision

"Flood Level Monitoring Using ThingSpeak Cloud" is the bonafide work of

Name of the student: Maharghya Mukherjee ***Signature:***

Name of the student: Ritankar Roy ***Signature:***

Name of the student: Santam Banik ***Signature:***

Name of the student: Debankur Sen ***Signature:***

Name of the student: Saswata Haldar ***Signature:***

SIGNATURE

Name : Anamika Banerjee

PROJECT MENTOR

SIGNATURE
Name:
EXAMINERS
Ardent Original Seal

ACKNOWLEDGEMENT

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor, **Ms. *Anamika Banerjee*** for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of **Ardent Computech Pvt. Ltd.** for their support.

CONTENTS:

INTRODUCTION

Introduction to IoT (Internet of Things):

The Internet of Things, or IoT, refers to the network of physical objects or "things" embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet. These connected devices can range from simple household items like smart thermostats and light bulbs to complex industrial machines and vehicles.

The concept of IoT revolves around the idea of enabling everyday objects to communicate and interact with each other autonomously, creating a smart, interconnected ecosystem. This connectivity allows for the collection, exchange, and analysis of data, leading to improved efficiency, accuracy, and convenience in various aspects of life and business.

IOT Flood Level Monitoring System:

1. **Background and Motivation:**

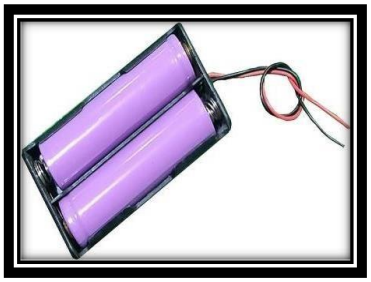
Flooding stands as one of the most destructive and frequent natural disasters globally, causing extensive damage to infrastructure, property, and, most critically, human life. Effective flood mitigation heavily relies on timely and accurate early warning systems. Traditional monitoring methods, which often involve manual inspection or expensive fixed telemetry stations, can be slow, geographically limited, and prone to failure during severe weather events. The rise of the Internet of Things (IoT) provides an ideal solution to these challenges, enabling the creation of inexpensive, decentralized, and real-time monitoring networks. By leveraging microcontrollers with integrated Wi-Fi capabilities, it is possible to deploy smart sensing devices in remote or high-risk areas, allowing continuous data collection and instant alerting.

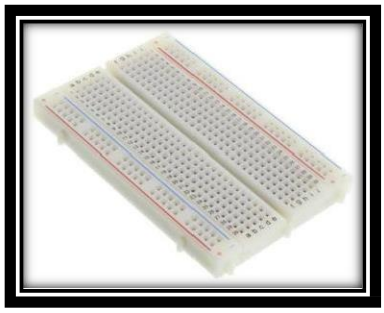
2. **Project Objectives:**

This project is dedicated to designing and implementing a low-cost, real-time flood level monitoring system. The primary goal is to provide continuous, high-precision measurement of water depth and to communicate potential flood threats to users wirelessly. The system utilizes the ESP8266 Wi-Fi enabled microcontroller as the core processing unit, paired with an ultrasonic sensor for non-contact distance measurement. The measured data is processed to determine the current flood threat level and visually communicated using integrated LED indicators before being transmitted over a network. This report details the design, implementation, and performance of this critical early warning system.

COMPONENTS USED IN FLOOD LEVEL MONITORING

1. NODEMCU (ESP8266)
2. ULTRASONIC SENSOR
3. LED
4. BATTERY
5. BREADBOARD
6. JUMPER WIRE



DESCRIPTION

- **ESP - 8266**

The ESP8266 is a popular and versatile microcontroller developed by Espressif Systems. It is widely used in IoT (Internet of Things) applications due to its low cost, low power consumption, and built-in Wi-Fi connectivity. Here are some key features of the ESP8266:

Microcontroller: The ESP8266 is based on a 32-bit Tensilica L106 microcontroller, which runs at clock speeds up to 80 MHz.

Wi-Fi Connectivity: One of the standout features of the ESP8266 is its integrated Wi-Fi module, which allows it to connect to Wi-Fi networks and communicate with other devices over the internet.

GPIO Pins: The ESP8266 features a number of General Purpose Input/Output (GPIO) pins, which can be used to connect to external sensors, actuators, and other peripheral devices.

Memory: The ESP8266 typically comes with 512KB to 4MB of flash memory for program storage, as well as 64KB of instruction RAM and 96KB of data RAM.

Programming: The ESP8266 can be programmed using the Arduino IDE, as well as other development environments such as MicroPython and Lua.

Community Support: The ESP8266 has a large and active community of developers, which has contributed to its popularity and the availability of libraries and resources.

Overall, the ESP8266 is a powerful and cost-effective microcontroller that is well-suited for a wide range of IoT applications, from home automation to industrial monitoring and control.

➢ FEATURES OF THE ESP-8266:

The ESP8266 is a popular and versatile microcontroller with integrated Wi-Fi capability. Here are some of its key features:

Microcontroller: The ESP8266 is based on a 32-bit Tensilica L106 microcontroller, running at clock speeds up to 80 MHz.

Wi-Fi Connectivity: It features integrated 802.11 b/g/n Wi-Fi connectivity, allowing it to connect to Wi-Fi networks and communicate with other devices over the internet.

GPIO Pins: The ESP8266 has a number of General-Purpose Input/Output (GPIO) pins, which can be used to interface with external sensors, actuators, and other devices.

Memory: It typically comes with 512KB to 4MB of flash memory for program storage, as well as 64KB of instruction RAM and 96KB of data RAM.

Programming: The ESP8266 can be programmed using the Arduino IDE, as well as other development environments such as MicroPython and Lua.

Low Power Consumption: The ESP8266 is designed for low power consumption, making it suitable for battery-powered applications.

Community Support: The ESP8266 has a large and active community of developers, which has contributed to its popularity and the availability of libraries and resources.

Cost-Effective: The ESP8266 is relatively inexpensive, making it an attractive option for projects with budget constraints.

Overall, the ESP8266 is a versatile and cost-effective microcontroller with integrated Wi-Fi, making it well-suited for a wide range of IoT applications

➢ ESP-8266 HARDWARE PART:

The ESP8266 module includes several key hardware components that enable its functionality:

Microcontroller: The heart of the ESP8266 is a 32-bit Tensilica L106 microcontroller running at speeds up to 80 MHz. This microcontroller is responsible for executing program instructions, managing GPIO pins, and interfacing with other hardware components.

Wi-Fi Module: The ESP8266 features an integrated 802.11 b/g/n Wi-Fi module, which allows it to connect to Wi-Fi networks and communicate with other devices over the internet.

Memory: The ESP8266 typically comes with 512KB to 4MB of flash memory for program storage. It also has 64KB of instruction RAM and 96KB of data RAM.

GPIO Pins: The ESP8266 module has several GPIO pins that can be used for digital input/output, analog input, and other purposes. These pins allow the ESP8266 to interface with external sensors, actuators, and other devices.

Power Supply: The ESP8266 can be powered using a 3.3V power supply. It is important to note that the ESP8266 is not 5V tolerant, so care must be taken when interfacing it with 5V systems.

Antenna: Some ESP8266 modules come with an onboard antenna for Wi-Fi communication. Others may require an external antenna for improved signal strength and range.

Serial Interface: The ESP8266 can communicate with other devices using a serial interface (UART). This interface is commonly used for programming the ESP8266 and for serial communication with other devices.

Overall, the ESP8266 hardware is designed to be compact, cost-effective, and easy to use, making it a popular choice for IoT projects and embedded systems.

➢ POWER REQUIREMENT OF ESP8266:

The ESP8266 typically requires a power supply voltage of 3.3 volts (V). It is important to provide a stable and clean power supply to the ESP8266 to ensure proper operation.

The current consumption of the ESP8266 can vary depending on its operating mode and the specific task it is performing. In general, the ESP8266 consumes more current when transmitting data over Wi-Fi compared to when it is idle or in sleep mode.

When designing a power supply for the ESP8266, it is recommended to provide sufficient current capacity to accommodate the peak current demands of the module,especially during Wi-Fi transmission. Using a voltage regulator to ensure a stable 3.3V supply voltage is also advisable to prevent damage to the ESP8266.

The ESP8266 has different power modes that affect its power consumption and performance. Here are the main power modes of the ESP8266:

Active Mode: In active mode, the ESP8266 is fully powered on and executing instructions. This mode is used when the module is actively performing tasks, such as transmitting or receiving data over Wi-Fi. The power consumption in active mode can vary depending on the specific task and the operating conditions but is generally higher than in other modes.

Modem Sleep Mode: In modem sleep mode, the ESP8266 turns off the Wi-Fi modem while keeping the CPU and other peripherals active. This mode reduces power consumption compared to active mode but still allows the module to quickly wake up and resume normal operation when needed.

Light Sleep Mode: In light sleep mode, the CPU is paused, and the system clock is gated off, while the Wi-Fi modem and other peripherals remain active.

This mode reduces power consumption further than modem sleep mode but still allows for quick wake-up times.

Deep Sleep Mode: In deep sleep mode, the ESP8266 shuts down most of its internal components, including the CPU, system clock, and Wi-Fi modem.

Only a real-time clock (RTC) and some low-power RAM are kept powered.

➢ PERIPHERALS AND I/O OF ESP-8266

The ESP8266 microcontroller module features a variety of peripherals and I/O (Input/Output) pins, making it suitable for a wide range of IoT (Internet of Things) applications. Here are some of the key peripherals and I/O features of the ESP8266:

GPIO (General Purpose Input/Output) Pins: The ESP8266 typically includes several GPIO pins that can be used for digital input/output. These pins can be used to interface with sensors, LEDs, relays, and other external devices.

UART (Universal Asynchronous Receiver/Transmitter): The ESP8266 includes one or more UART interfaces, which can be used for serial communication with other devices. UART is commonly used for debugging and programming the ESP8266.

SPI (Serial Peripheral Interface): The ESP8266 includes one or more SPI interfaces, which can be used to communicate with SPI-compatible devices such as sensors, displays, and SD cards.

I2C (Inter-Integrated Circuit): The ESP8266 includes one or more I2C interfaces, which can be used to communicate with I2C-compatible devices such as sensors, EEPROMs, and real-time clocks.

ADC (Analog-to-Digital Converter): The ESP8266 includes one or more ADC channels, which can be used to measure analog voltages from sensors or other external devices.

PWM (Pulse-Width Modulation): The ESP8266 includes one or more PWM channels, which can be used to generate PWM signals for controlling things like LEDs, motors, and servos.

I2S (Inter-IC Sound): Some variants of the ESP8266 include an I2S interface, which can be used for digital audio input and output.

Deep Sleep Wake-Up GPIOs: The ESP8266 includes several GPIO pins that can be used to wake up the device from deep sleep mode.

These peripherals and I/O features make the ESP8266 a versatile microcontroller for a wide range of IoT applications, allowing it to interface with various sensors, actuators, and other devices.

➢ ON-BOARD SWITCHES & LED INDICATOR

The ESP8266 module itself does not typically include on-board switches or LED indicators. However, development boards and modules based on the ESP8266 often include these components for easier prototyping and debugging. Here are some common features you might find on such boards:

Reset Switch: A reset switch is often included to reset the ESP8266 module without having to power cycle it. This can be useful during development and testing.

Flash Button: Some development boards include a flash button that puts the ESP8266 into programming mode, allowing you to easily upload new firmware to the module.

User Button: A user button can be included for general-purpose use in your application. You can program the ESP8266 to respond to button presses in a variety of ways.

LED Indicators: Development boards often include LEDs that can be controlled by the ESP8266. These LEDs can be used for status indicators, debugging, or as general-purpose indicators in your application.

These features can vary depending on the specific development board or module you are using. It's a good idea to consult the documentation for your specific board to see what features are included and how to use them.

➢ SWITCHES & INDICATORS

The ESP8266 itself does not have built-in switches or indicators. However, when working with ESP8266 development boards or modules, you might find the following components:

Reset Switch: Allows you to reset the ESP8266 module without having to power cycle it, which is useful during development and testing.

Flash Button: Puts the ESP8266 into programming mode, facilitating the uploading of new firmware to the module.

User Button: A general-purpose button that you can program to trigger specific actions in your application.

LED Indicators: LEDs that can be controlled by the ESP8266, typically used for status indication, debugging, or as general-purpose indicators in your application.

These components vary depending on the specific ESP8266 development board or module you are using. It's a good idea to refer to the documentation of your board for details on the available switches and indicators and how to use them in your projects.

SERIAL COMMUNICATION:-

The ESP8266 supports serial communication, which allows it to communicate with a devices using the UART (Universal Asynchronous Receiver/Transmitter interface. Here's other brief overview of how serial communication works with ESP8266:

UART PINS:- The ESP8266 has several GPIO pins that can be used for serial communication. These pins are typically labeled TX (transmit) and RX (receive).

Serial Library:- To use serial communication with the ESP8266, you can use the Serial library in the Arduino IDE of other development environments. This library provides functions for sending and receiving serial data.

Baud Rate:- When using serial communication , you need to specify the baud rate, which determines the speed of communication. The ESP8266 supports baud rates ranging from 300 to 115200 baud.

Serial Begin:- To initialize serial communication, you use the Serial.begin() function , specifying the desired baud rate. For example – Serial.begin(9600): initializes serial communication at a baud rate of 9600.

Serial.print() and Serial.println():- These functions are used to send data over the serial connection. Serial,print() sends data as is, while Serial.println() adds a newline character at the end of the data.

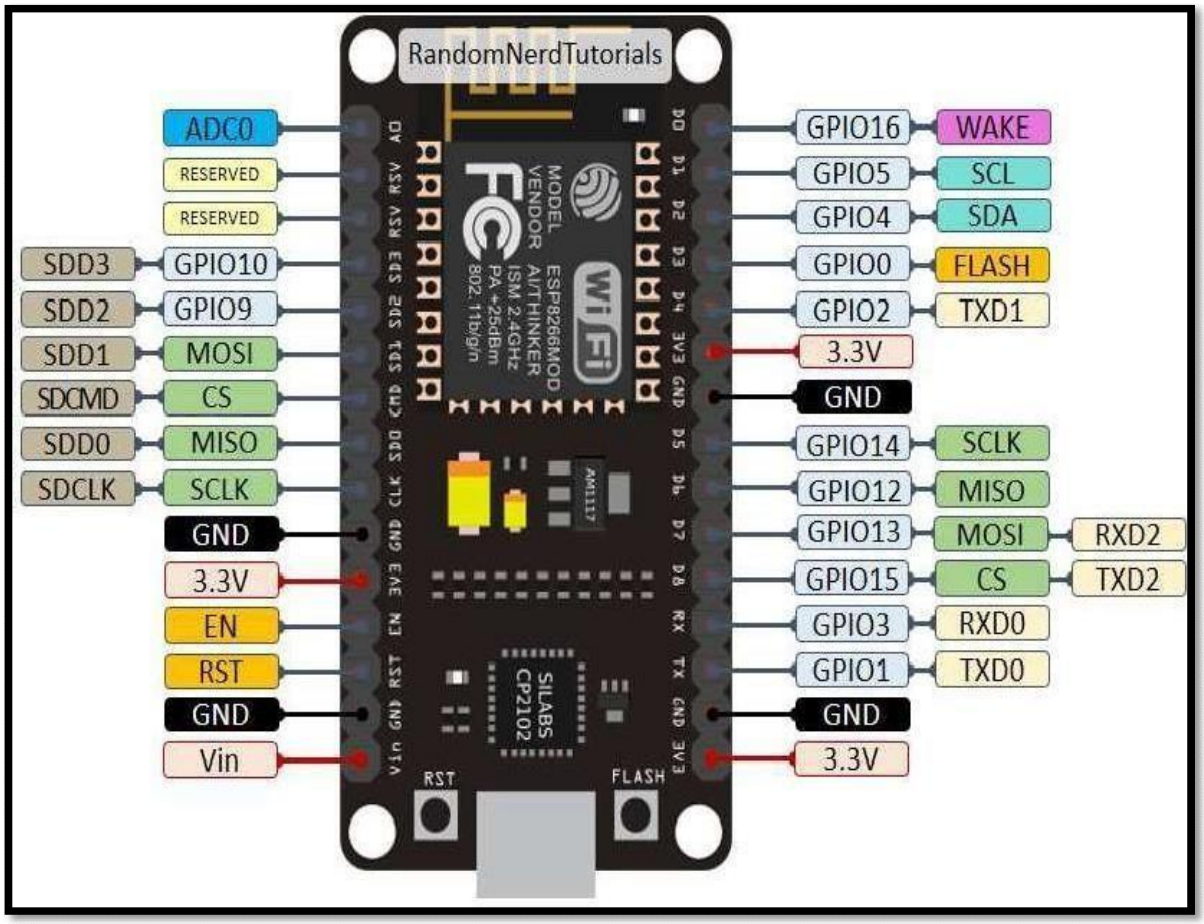
Serial.read():- This function reads incoming serial data. It returns the next byte of incoming data, or -1 if no data is available.

Serial.available():- This function returns the number of bytes available for reading from the serial buffer.

Here's the basic example of how to use serial communication with the ESP8266:-

```
cpp
void setup() {
Serial.begin(9600);
}
void loop() {
if (Serial.available()>0) {
   char incomingByte = Serial.read();
Serial.print()("Received:");
Serial.println(incomingByte);
 }
}
```

ESP 8266 PINOUT:



The ESP8266 module typically has several GPIO (General Purpose Input/Output) pins that can be used for various purposes. The exact pinout can vary depending on the specific module or development board you are using, but here is a general pinout for the ESP8266-12 module, which is one of the commonly used variants:

VCC: Power supply pin. Typically requires 3.3V.
GND: Ground pin.
TXD: Transmit pin for serial communication.
RXD: Receive pin for serial communication.
GPIO0: General-purpose input/output pin. Can also be used to put the ESP8266 into programming mode.
GPIO2: General-purpose input/output pin.
CH_PD: Chip enable pin. Pulling this pin high enables the ESP8266.
RESET: Reset pin. Pulled low to reset the ESP8266.
ADC: Analog-to-digital converter pin. Can be used to measure analog voltages.
GPIO15: General-purpose input/output pin. Used to select boot mode.
Please note that the pinout can vary between different ESP8266 modules and development boards, so it's important to refer to the documentation for your specific module or board to determine the correct pinout.

➢ SOFTWARE USED:

The ESP8266 can be programmed using various software development tools and environments. Here are some of the commonly used options:

Arduino IDE: The Arduino IDE supports programming the ESP8266 using the Arduino programming language. You can use the ESP8266 Arduino core, which provides libraries and examples specifically for the ESP8266.

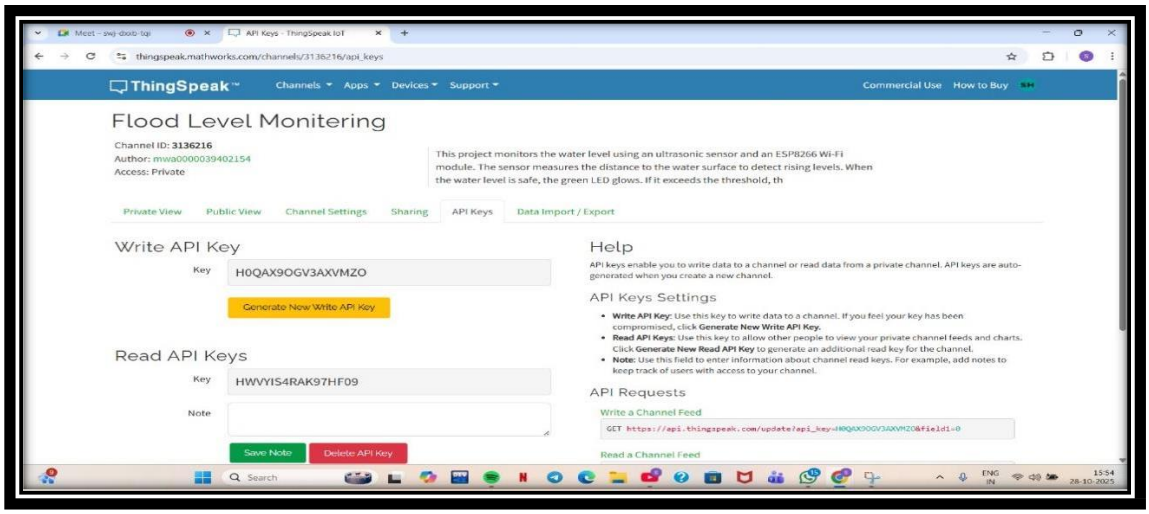
PlatformIO: PlatformIO is an open-source ecosystem for IoT development that supports the ESP8266. It provides a unified development platform that integrates with popular IDEs such as Visual Studio Code, Atom, and Eclipse.

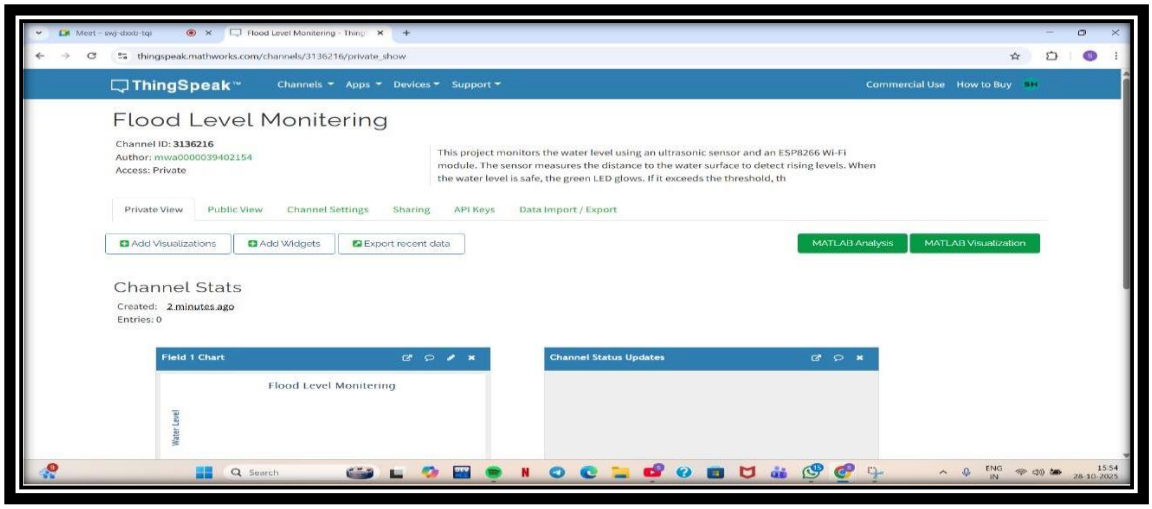
Espressif IDF: Espressif IoT Development Framework (ESP-IDF) is the official development framework for the ESP8266 and ESP32. It provides a set of libraries and tools for developing applications for the ESP8266.

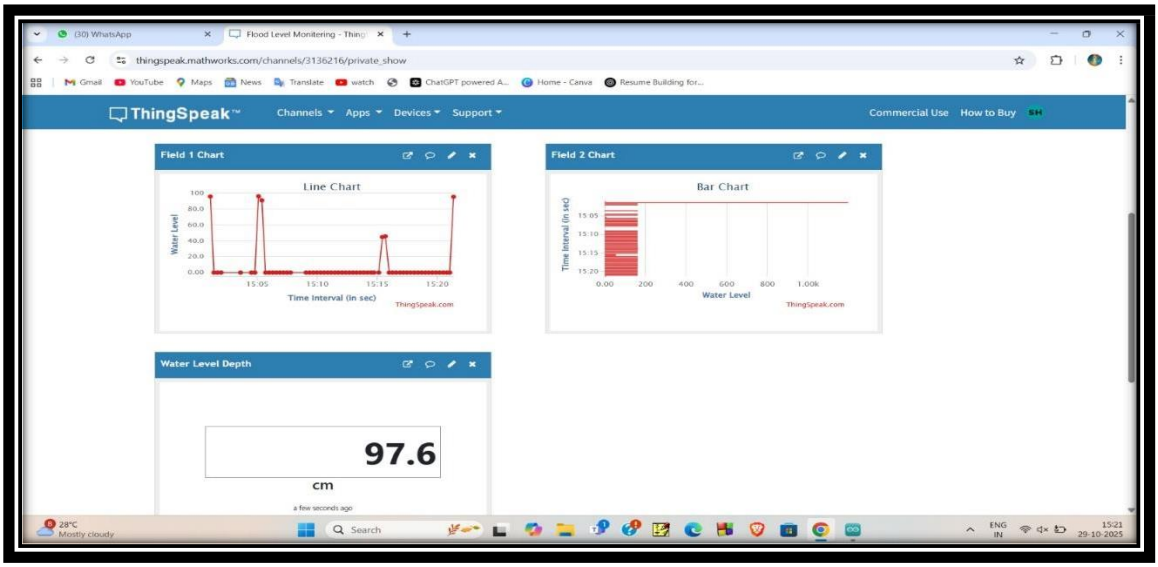
Lua/NodeMCU: Lua is a scripting language that can be used to program the ESP8266. NodeMCU is a firmware that provides an environment for running Lua scripts on the ESP8266.

These are just a few examples of the software tools that can be used to program the ESP8266. The choice of tool depends on your programming experience, the requirements of your project, and your personal preference.

ThingSpeak IOT: Data Transmission and Cloud Integration The system utilizes ThingSpeak as the primary cloud platform for data management. The ESP8266 securely publishes the water depth measurements via the MQTT or HTTP protocol to a dedicated ThingSpeak channel. This enables real-time data visualization through customizable graphs, historical logging, and the implementation of web-based alert triggers based on predefined flood threshold


ThingSpeak
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Flood Level Monitering
Channel ID: 3136216
Author: mwa0000039402154
Access: Private
This project monitors the water level using an ultrasonic sensor and an ESP8266 Wi-Fi module. The sensor measures the distance to the water surface to detect rising levels. When the water level is safe, the green LED glows. If it exceeds the threshold, th
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Write API Key
Key
H0QAX9OGV3AXVMZO
Generate New Write API Key
Read API Keys
Key
HWVYIS4RAK97HF09
Note
Save Note
Delete API Key
Help
API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.
API Keys Settings
Write API Key: Use this key to write data to a channel. If you feel your key has been compromised, click Generate New Write API Key.
Read API Keys: Use this key to allow other people to view your private channel feeds and charts. Click Generate New Read API Key to generate an additional read key for the channel.
Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.
API Requests
Write a Channel Feed
GET https://api.thingspeak.com/update?api_key=H0QAX9OGV3AXVMZO&field1=0
Read a Channel Feed


ThingSpeak
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Flood Level Monitering
Channel ID: 3136216
Author: mwa0000039402154
Access: Private
This project monitors the water level using an ultrasonic sensor and an ESP8266 Wi-Fi module. The sensor measures the distance to the water surface to detect rising levels. When the water level is safe, the green LED glows. If it exceeds the threshold, th
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Add Visualizations
Add Widgets
Export recent data
MATLAB Analysis
MATLAB Visualization
Channel Stats
Created: 2 minutes ago
Entries: 0
Field 1 Chart
Flood Level Monitering
Water Level
Channel Status Updates


ThingSpeak
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Field 1 Chart
Line Chart
Water Level
Time Interval (in sec)
ThingSpeak.com
Field 2 Chart
Bar Chart
Time Interval (in sec)
Water Level
ThingSpeak.com
Water Level Depth
97.6
cm
a few seconds ago

Sensing Component: The Ultrasonic Sensor

3.1 What is an Ultrasonic Sensor?

An ultrasonic sensor, typically the HC-SR04 model, is an electronic device designed to measure the distance to a target object using high-frequency sound waves. This sensor is crucial for the flood monitoring project because it enables non-contact measurement, allowing the system to determine the water level by calculating the distance between the sensor (mounted above the water) and the water's surface.

3.2 Principles of Operation (How it Works)

The fundamental operation of the sensor is based on the speed of sound and the principle of echolocation. The measurement process involves two core steps: emitting a sound pulse and timing the echo return. The sensor determines the distance (D) by measuring the time interval (Time) between sending the ultrasonic pulse (typically 40 kHz) and receiving the reflected echo from the water's surface. The distance is calculated using the following mathematical relationship, where the result is divided by two to account for the sound traveling both to and from the water:

3.3 Components of an Ultrasonic Sensor (HC-SR04)

The HC-SR04 module is a compact unit consisting of two primary transducers and supporting electronics: the Transmitter (T), which is a piezoelectric crystal responsible for converting an electrical trigger signal into an ultrasonic sound burst; and the Receiver (R), a second piezoelectric crystal that detects the returning sound echo. All these are managed by integrated Control Circuitry which handles the precise timing and signal generation.

3.4 Pinout and Interfacing

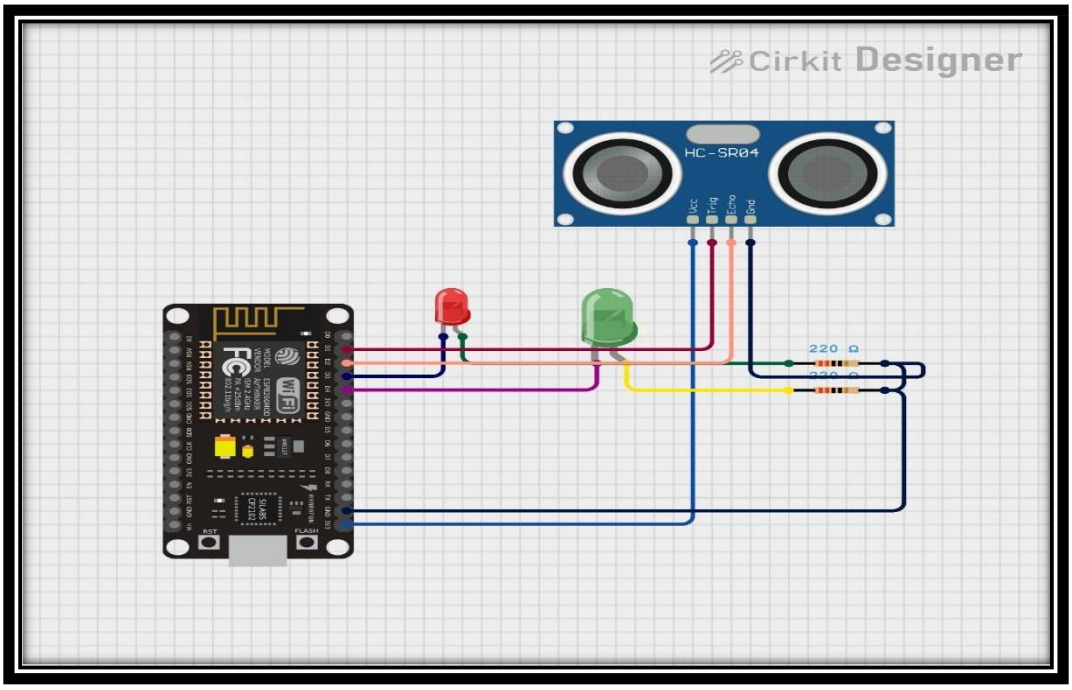
The standard interface of the HC-SR04 utilizes four pins for connectivity with the ESP8266 microcontroller. VCC provides the power supply (typically 3.3V or 5V). GND serves as the common ground reference. The Trig pin is an input that must be connected to a digital output pin on the ESP8266; this pin receives the initial pulse that instructs the sensor to fire the sound wave. Finally, the Echo pin is an output that connects to a digital input pin on the ESP8266; this pin generates a variable-width pulse whose duration is directly proportional to the distance measured.

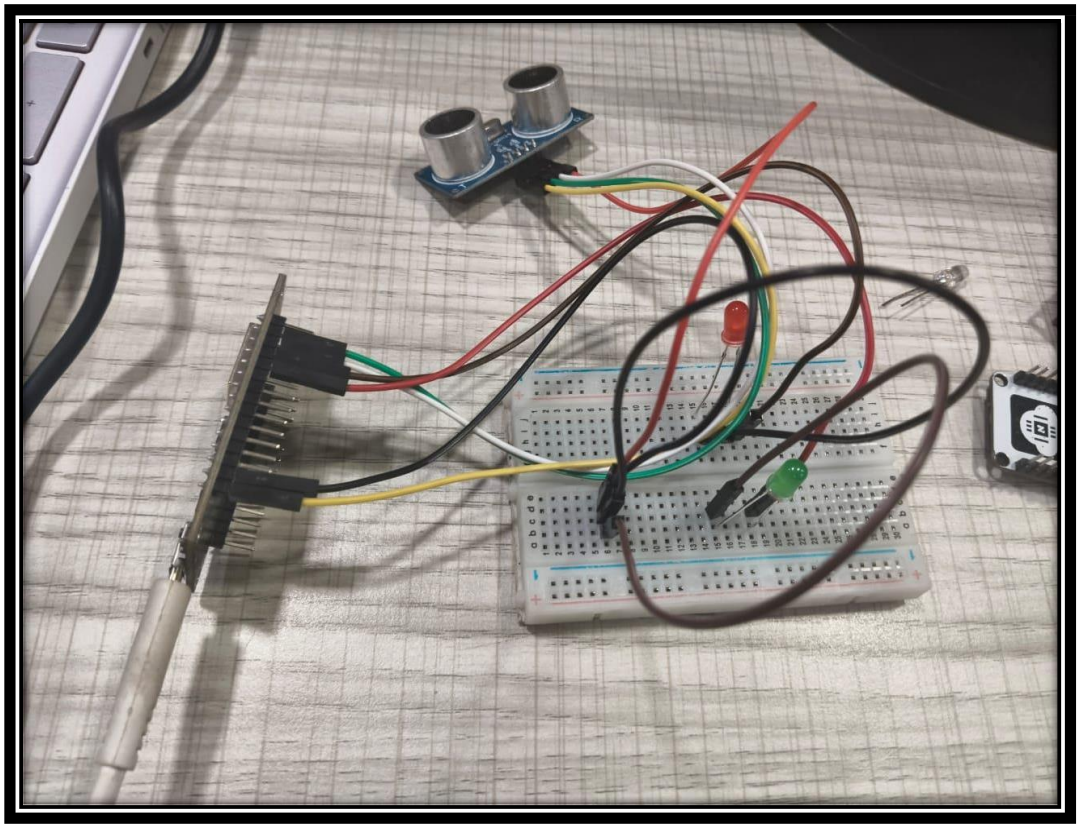
3.5 How to Use the Sensor

To acquire a distance measurement, the ESP8266 executes a precise sequence of steps. First, the microcontroller sends a brief (10 \mu s) HIGH pulse to the Trig pin to activate the sensor. This action causes the sensor to emit an 8-cycle ultrasonic burst, and simultaneously, the Echo pin goes HIGH. The ESP8266 then measures the length of time (duration) that the Echo pin remains HIGH. Once the returning sound wave is detected by the receiver, the Echo pin returns to LOW. The recorded duration is then used in the distance formula to compute the distance to the water surface, providing the crucial data point for flood monitoring.

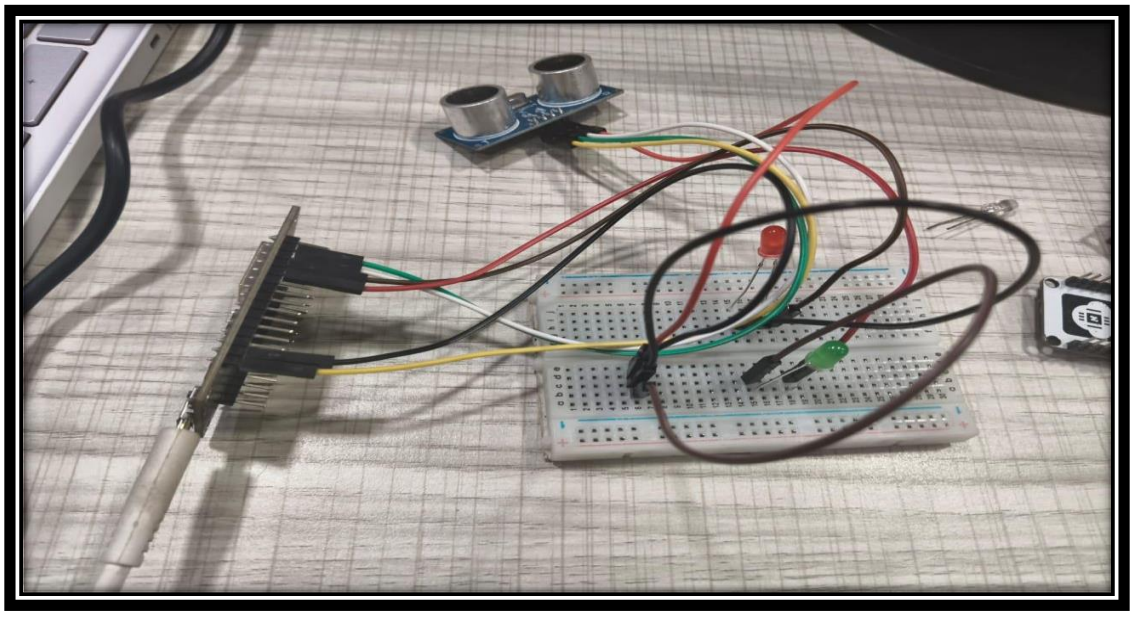

HC-SR04 Pinout
HC-SR04
VCC
Trig
Echo
GND
T
R
VCC
Trig
GND
Echo

CIRCUIT DIAGRAM

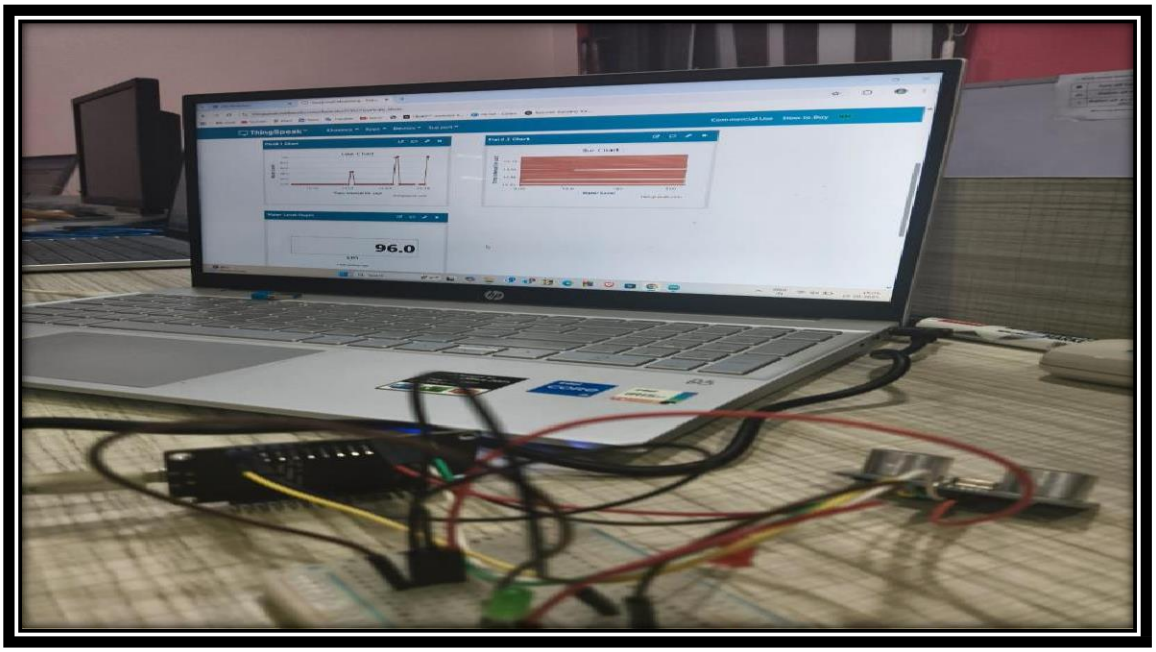




Condition Before Working Of Project



Condition After Working Of Project



CODE:-

```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"

const char* ssid = “Airtel_Ardent”;
const char* password = "Ardent@1234";
unsigned long channelID = 3136216;
const char* writeAPIKey = "H0QAX9OGV3AXVMZO";

WiFiClient client;

const int trigPin = D1;
const int echoPin = D2;
const int greenLED = D3;
const int redLED = D4;
const int tankHeight = 100;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(redLED, OUTPUT);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) delay(500);
  ThingSpeak.begin(client);
}

float getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  long duration = pulseIn(echoPin, HIGH);
  float distance = duration * 0.034 / 2;
  return distance;
}

void loop() {
  float distance = getDistance();
  float waterLevel = tankHeight - distance;
  if (waterLevel < 0) waterLevel = 0;

  if (waterLevel < 50) {
    digitalWrite(greenLED, HIGH);
    digitalWrite(redLED, LOW);
  } else {
    digitalWrite(greenLED, LOW);
    digitalWrite(redLED, HIGH);
  }

  ThingSpeak.setField(1, waterLevel);
  ThingSpeak.setField(2, distance);
  ThingSpeak.writeFields(channelID, writeAPIKey);
  delay(15000);
}
```

- **OTHER APPLICATIONS OF IOT**

Home Is Where the Smart Is:

EVM Machine-to-machine communication, and you understand you're not the most tech-savvy consumer, it's impossible that you've missed the abundance of home automation products filling the shelves and ads of every home improvement store. Suddenly an ordinary errand for light bulbs will leave you wondering if your lamp could send you a message alerting you that the light bulb needs to be replaced. Furthermore, if your lamp is talking to you, could your refrigerator and sprinkler system be too? Experts say: Yes, the possibilities are endless. If that's the case, where do you begin?

Any day-to-day, repeatable process is automatable with smart home applications. The greater the control and flexibility of these processes, the more energy and cost savings the resident experiences, which are factors anyone who pays utilities strives to moderate. The smart home revolution is likely to be more of an evolution, with the incorporation of one or two home systems at a time, gradually automating our households through smart mobile devices.

However, with these elements of efficiency comes the question of ease of use. Will it bring you enjoyment or exasperation? With so many brands and models already available in an ever- growing market, how do you knowwhich is best for you?

- **Lighting Control: Leaving the Dark Ages and Stepping Into the Light**

Smart lighting allows you to control wall switches, blinds, and lamps, but how intuitive is a lighting control system? Itturns out, quite; its capabilities are extensive. You're able to schedule the times lights should turn on and off, decide which specific rooms should be illuminated at certain times, select the level of light which should be emitted, and choose how particular lights react through motion sensitivity, as seen with Belkin's WeMo Switch + Motion, which isboth affordable and easy to use with its plug-and-play simplicity.

- **HVAC Regulation: No Longer Burned by Your Heating Bill:**

As fuel costs rise and the availability and sustainability of our resources becomes a greater concern, heating/cooling our homes efficiently is less of a budgetary bonus and more of a necessity. Over the past year, smart thermostats andautomated home heating systems have become more readily available and easily incorporated into any home.

Heating and cooling our homes consume an average of 50% of energy costs yearly, making daily HVAC regulation progressively rewarding. Maintaining a substantial lead among the nearly non-existent competition, the Nest Learning Thermostat, learns your heating and cooling preferences over time, eliminating the need for programming and is accessible from your smartphone app. With automated HVAC you are able to reduce the heat when a room is unoccupied, and increase or decrease it at specific times based on your schedule and occupancy.

Lawn Irrigation Systems: The Grass is Always Greener

A lush and healthy lawn is a source of pride for most homeowners, but the weather doesn't always cooperate and provide the adequate elements for a flourishing landscape. For decades we've relied on sprinkler systems to keep our yards at peak presentation, but at what cost? The average American home spends approximately 30% of their daily water usage on lawn and garden maintenance. Nearly half of that amount is wasted due to inefficiency. If you apply that statistic to the national average, up to 4.5 billion gallons of water is wasted per day through ineffective watering methods. If we reflect upon the monetary impact of this, it results in Americans spending over a thousand dollars a year in water, with a portion of that being wasted. The global effects are even greater when you consider the growing

concern over climate change and the dramatic decrease in agricultural natural resources. However, sprinkler control systems, like Skydrop, are providing water regulation through real-time communication with local weather data. If a rainstorm develops and deposits two inches of rainwater on your lawn, the automated sprinkler detects the saturation and disables its scheduled watering. Conversely, the system will be alerted to dry conditions and supply the necessary amount of nourishment, without over-watering.

- **Smart Appliances: What's for Dinner?**

Will smart kitchen appliances actually make you a better cook? Maybe. Smart refrigerators, such as LG's Smart ThinQ, allow you to scan grocery store receipts and keep an inventory of your items, and alerts you if an item is about to expire. More impressively, it suggests recipes based on your refrigerator's contents and lets you know when you need to replace items. Smart ovens sync with your smartphone and automatically preheat to the correct temperature based on a recipe selected from your database. While these appliance options seem a bit superficial and convenience based, there is a conservation factor as well. By automating your kitchen appliances and making them accessible from your smart device, you're able to sever the electricity supplied to unused appliances and reduce your energy consumption and costs.

- **Security Systems: Knock, Knock…**

Who's there? The Internet of Things. While efficiency and conservation are certainly IoT benefits, its potential to have improved control over home security is a primary focus. Smart locks, like Kwikset's Kevo, a Bluetooth enabled electronic deadbolt, and various connected home security systems, such as iSmartAlarm, offer a variety of features including door and window sensors, motion detectors, video cameras and recording mechanisms. All of which are connected to a mobile device and accessible via the cloud, thus enabling you to access real-time information on the security status of your home. Naturally, there is a great deal of scrutiny regarding the level of trust in controlling your home's security system via a mobile device, but it begs earnest exploration when weighing the potential benefits and peace of mind it provides homeowners.

Future Scope of this Project:-

1. **Future Scope and Potential Enhancements**: While the current system successfully provides real-time flood level data and visualization via ThingSpeak, the project has significant potential for expansion and optimization. Incorporating future enhancements would transform the prototype into a more robust and comprehensive commercial or public safety solution

1.1. **Advanced Alert and Notification Systems**: The current system relies on basic ThingSpeak alerts. The future scope includes integrating more versatile notification channels to ensure timely response: SMS and Voice Alerts: Integrating with services like Twilio to send immediate text messages and automated voice calls to local authorities or homeowners when the high-alert threshold is breached. Mobile Application: Developing a dedicated mobile app (Android/iOS) to receive push notifications and display a more user-friendly interface for monitoring multiple deployment sites.

1.2. **Power Management and Reliability**: To transition from a prototype to a permanent outdoor solution, power efficiency and reliability must be addressed: Solar Power Integration: Replacing the current USB power source with a small solar panel and rechargeable battery pack, enabling autonomous operation in remote locations without access to the electrical grid. Low-Power Deep Sleep Mode: Optimizing the ESP8266 code to utilize deep sleep cycles, waking up only to take a measurement and transmit data, drastically reducing power consumption.

1.3. **Data Processing and Analysis**: Enhancing the data analysis capabilities will improve predictive power: Rainfall Integration: Incorporating a digital rain gauge to measure current precipitation levels alongside the water level. This allows for correlation analysis to better predict rapid level increases. Machine Learning for Prediction: Implementing a simple machine learning model (run either in the cloud or on the ESP32—a more powerful alternative to ESP8266) to analyze historical data and current rainfall to provide short-term flood forecasts, moving the system from reactive to predictive monitoring.

CONCLUSION

In summary, the IoT Flood Level Monitoring System successfully integrates low-cost, readily available hardware and cloud services to address the critical need for localized flood detection. The ESP8266 microcontroller provides the essential Wi-Fi connectivity, while the HC-SR04 ultrasonic sensor delivers continuous, precise distance measurements, effectively translating to real-time water level data. This data is reliably streamed to the ThingSpeak platform, which offers immediate visualization and the capacity to trigger both LED warnings and remote alerts based on predefined flood thresholds. The working prototype validates the project's central objective: to establish an accessible, real-time warning system. Future development efforts should focus on transitioning the system to solar power and implementing predictive analytics to enhance its reliability and contribution to public safety.

REFERENCES

- Subramaniam, S. K., Gannapathy, V. R., Subramonian, S., & Hamidon, A. H. (2010). Flood level indicator and risk warning system for remote location monitoring using Flood Observatory System. WSEAS Transactions on Systems and Control, 5(3), 153-163.
- Lo, S. W., Wu, J. H., Lin, F. P., & Hsu, C. H. (2015). Visual sensing for urban flood monitoring. Sensors, 15(8), 20006-20029.
- Sunkpho, J., & Ootamakorn, C. (2011). Real-time flood monitoring and warning system. Songklanakarin Journal of Science & Technology, 33(2).
- Natividad, J. G., & Mendez, J. M. (2018, March). Flood monitoring and early warning system using ultrasonic sensor. In IOP conference series: materials science and engineering (Vol. 325, No. 1, p. 012020). IOP Publishing.
- Garcia, F. C. C., Retamar, A. E., & Javier, J. C. (2015, November). A real time urban flood monitoring system for metro Manila. In TENCON 2015-2015 IEEE region 10 conference (pp. 1-5). IEEE.