**IBM PROJECT**

**TEAM LEADER : KARTHIGA .S**

**TEAM MEMBERS :**

**1. SASDIKA.E**

**2. TANNU**

**3. SHUNMUGA SUNDARI.M**

**4. SANTHIYA JESSICA .B**

**PROJECT NAME : FLOOD MONITORING AND EARLY WARNING SYSTEM**

**PHASE 5 : THE FULL PROJECT**

**INTRODUCTION:**

Floods are one of the most devastating and frequent natural disasters, causing significant damage to property, infrastructure, and, unfortunately, loss of lives. The ability to monitor and predict floods is crucial for mitigating their impact and providing timely warnings to at-risk communities. In response to this need, an IoT-based Flood Monitoring and Early Warning System is proposed. This project leverages the power of the Internet of Things (IoT) to provide real-time flood data collection, analysis, and alerting to help save lives and protect assets.

**PROJECT COMPONENTS :**

1. ESP8266 Node MCU
2. Ultrasonic HC-SR04 Sensor
3. LED –(Red,Green)
4. Buzzer
5. ThinkSpeak IoT Cloud Platform
6. For programming –Arduino IDE

**ABSTRACT:**

Floods, as natural disasters, pose significant threats to both human life and property. The need for effective flood monitoring and early warning systems has never been more pressing. This project introduces an innovative solution: the IoT-Based Flood Monitoring and Early Warning System. By harnessing the capabilities of the Internet of Things (IoT), this project aims to provide real-time flood data collection, analysis, and timely alerts, ensuring the safety of at-risk communities and the protection of critical assets.

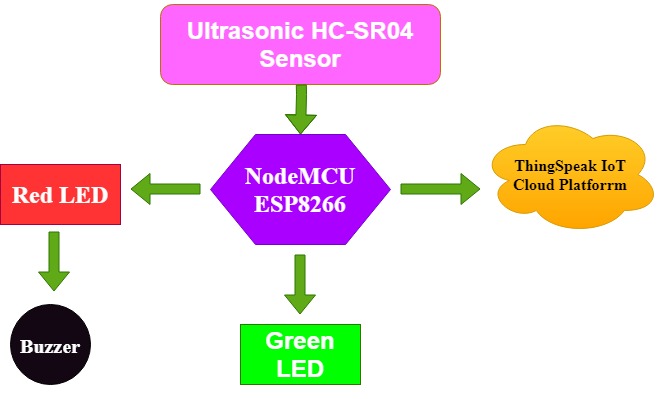
**OBJECTIVE:**

The project's primary objectives include the development of a robust IoT infrastructure, continuous data collection from a network of sensors, real-time data analysis using data analytics and machine learning techniques, the creation of an integrated early warning system, and the provision of a user-friendly interface for data access and alert reception.

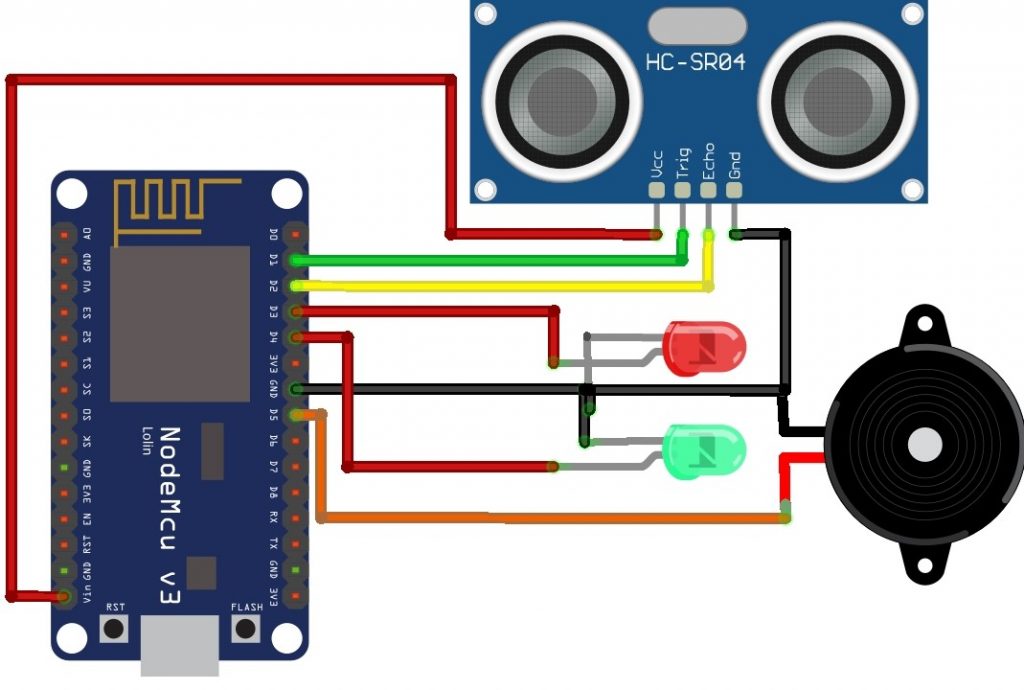
**BENEFITS :**

The benefits of this IoT-based system are profound, including enhanced safety through timely and accurate flood warnings, protection of critical infrastructure, data-driven decision making for flood risk management, cost savings by reducing flood-related damages, and scalability for broader geographic coverage.

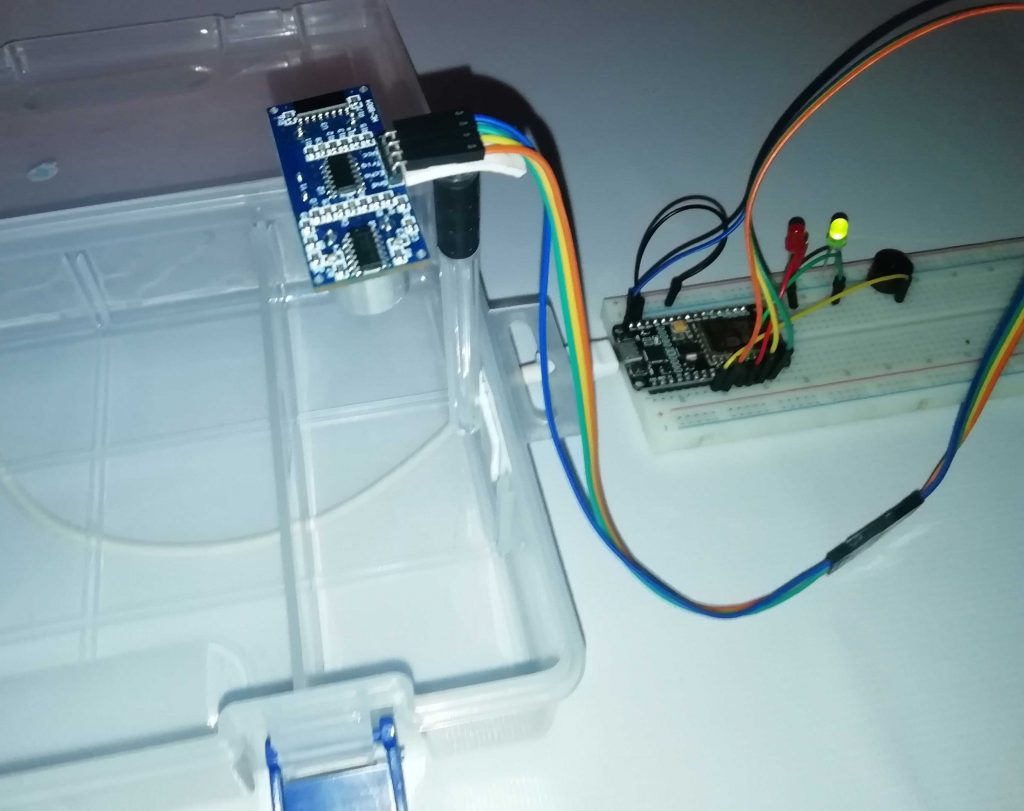
**BLOCK DIAGRAM :**



**CIRCUIT DIAGRAM :**



**OUR HARDWARE CONNECTION DIAGRAM :**



**PROGRAM CODE EXPLANATION :**

Begin the code by including all the necessary library files in the code for ESP8266 boards. The ThingSpeak.h library is used for the ThingSpeak platform. It can be added to the Arduino IDE using the following steps:

* ***In the Arduino IDE, choose Sketch/Include Library/Manage Libraries.***
* ***Click the ThingSpeak Library from the list, and click the Install button.***

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>

Next, define the pins which are used for the Ultrasonic sensor, Buzzer and LEDs.

#define redled D3

#define grnled D4

#define BUZZER D5 //buzzer pin

Now, Enter the network credentials- i.e. SSID and password of your WiFi Network to connect the NodeMCU with the internet. Then the ThingSpeak account credentials: channel number, API Key, and Author Key. These all credentials were recorded while setting ThingSpeak IoT Platform. Hence, make sure, you have edited these credentials in place of these variables

unsigned long ch\_no = 1053193;//Replace with Thingspeak Channel number

const char \* write\_api = "1WGTOHK9622G57JI";//Replace with Thingspeak write API

char auth[] = "mwa0000018384149";

char ssid[] = "Alsan Air WiFi 4"; char pass[] = "11122235122@kap1";

The variables are defined for timing purposes.

unsigned long startMillis;

unsigned long currentMillis;

const unsigned long period = 10000;

WiFiClient client;

long duration1;

int distance1;

Basically, to connect NodeMCU to the internet, we call **WiFi.begin**function. To Check for the successful network connection ***WiFi.status()*** is used. Finally, after a successful connection, we print a message on the Serial Monitor with the NodeMCU IP address.

Serial.begin(115200);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500):

Serial.print(".");

}

Serial.println("WiFi connected");

Serial.println(WiFi.localIP());

Now we connect to the **ThingSpeak IoT Platform**using Provided credentials. For this, we need to use *ThingSpeak.begin*function.

ThingSpeak.begin(client);

For calculating the distance, an input pulse is given to the sensor through the trig pin of the HC-SR04 sensor. Here, a 2-microsecond pulse is given, then from the echo pin, the output pulse of the sensor is read. Finally, the distance is calculated in cm.

digitalWrite(trigPin1, LOW);

delayMicroseconds(2);

digitalWrite(trigPin1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

duration1 = pulseIn(echoPin1, HIGH);

distance1 = duration1 \* 0.034 / 2;

Then, an if-else condition is defined for the LED indications and Buzzer state for both Normal and Flood conditions. Here as per my setup, I have taken 4 cm as reference. But, you can change it as per your requirements.

if (distance1 <= 4)

{

digitalWrite(D3, HIGH);

tone(BUZZER, 300);

digitalWrite(D4, LOW);

delay(1500);

noTone(BUZZER);

}

Else

{

digitalWrite(D4, HIGH);

digitalWrite(D3, LOW);

}

Finally, the river water level is uploaded to the ThingSpeak channel in interval of 10 seconds.

if (currentMillis - startMillis >= period)

{

ThingSpeak.setField(1, distance1);

ThingSpeak.writeFields(ch\_no, write\_api);

startMillis = currentMillis;

}

**SET UP TO THINKSPEAK ACCOUNT FOR FLOOD MONITORING AND EARLY WARNING SYSTEM:**

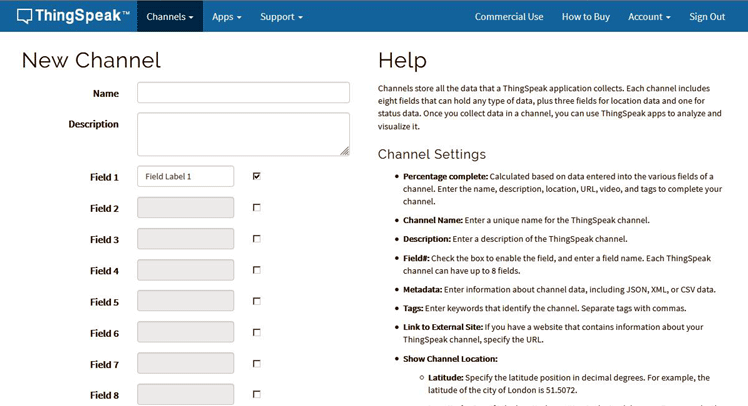
After the successful interface of the hardware parts according to the circuit diagram above. Now its time to set up the IoT platform, where data can be stored for online monitoring. Here we are using ThingSpeak to store data. ThingSpeak is a very popular IoT cloud platform that is used to store, monitor, and process data online.

**Step 1: Sign up for ThingSpeak**

First go to [ThingSpeak](https://thingspeak.com/) and create a new free MathWorks account if you don’t already have a MathWorks account.

**Step 2: Sign in to**[**ThingSpeak**](https://thingspeak.com/)

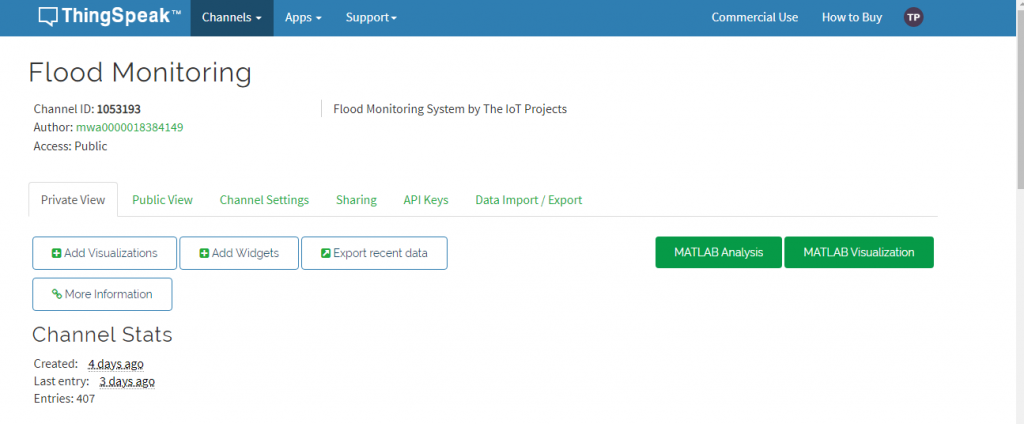
Sign in to ThingSpeak using your credentials and create “New Channel“. Now fill the project details like name, field names, etc. Here we need to create three field area names such as Flood Live Monitoring, and Flood Status. Then click “Save Channel”.



**Step 3: Record the credentials**

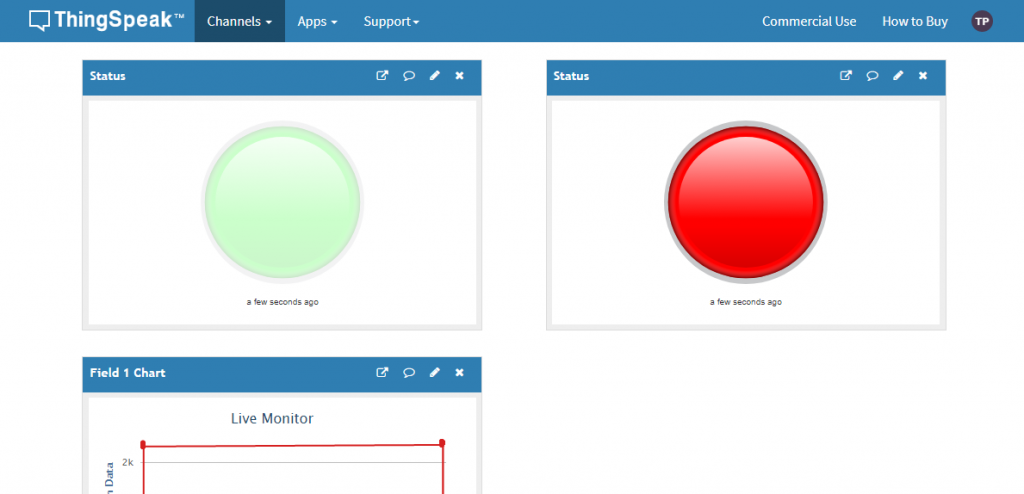
Select the created channel and record the following credentials.

Channel ID, which is at the top of the channel view.  
API key, which can be found in the API Key tab of your channel view.



**Step 4: Add widgets to your GUI**

Click “Add Widgets” and add two appropriate Indicator widgets. In my case, I have taken an indicator of flooding. Choose the appropriate field names for each widget.

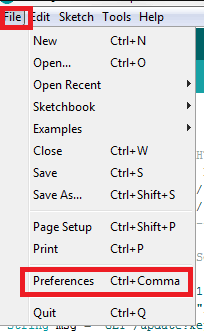


**SETTING UP ARDUINO IDE NODE MCU BOARD :**

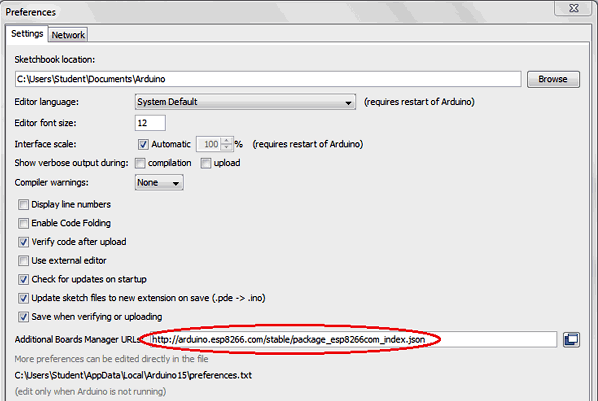
After the successful completion of the hardware setup. Now its time to program ESP8266 NodeMCU.

To upload the code to NodeMCU using the Arduino IDE, follow the steps below:

1. Open the Ardino IDE, then go to File> Preferences> Settings



1. Paste URL: **https://arduino.esp8266.com/stable/package\_esp8266com\_index.json** in the ‘Additional Board Manager URL‘ field and click ‘Ok’.



Now. Now go to **Tools> Board> Board Manager.** In the Boards Manager window, type **ESP8266 in the search box**, select the new version of the board and click Install.



After successful installation, go to Tools -> Board -> and select NodeMCU 1.0 (ESP-12E Module). Now you can program NodeMCU with Ardino IDE.

After the above setup for programming NodeMCU. You can upload the complete code to ESP8266 NodeMCU. The step-by-step explanation of the full code is provided below.

**OUR OUTPUT VIDEO LINK :**

[**https://www.kapwing.com/videos/654133441ed568c03dea85ff**](https://www.kapwing.com/videos/654133441ed568c03dea85ff)

**ADVANTAGES :**

The IoT-Based Flood Monitoring and Early Warning System offers several advantages that contribute to its effectiveness in mitigating the impact of floods and enhancing overall disaster preparedness. Here are some of the key advantages of the project:

1. Real-Time Data: The system provides real-time data on various flood-related parameters, including water levels, rainfall, weather conditions, and river flow, allowing for up-to-the-minute monitoring of flood-prone areas.
2. Early Warning: The system can issue timely and accurate flood warnings, enabling residents, emergency services, and local authorities to take proactive measures and evacuate areas at risk, reducing the potential for loss of life and property damage.
3. Asset Protection: By providing early warnings and actionable information, the project helps protect critical infrastructure, homes, and businesses from flood-related damages, minimizing economic losses.
4. Data-Driven Decision Making: Decision-makers can access historical and real-time flood data to inform disaster response planning, resource allocation, and flood risk management, improving overall decision-making processes.
5. Cost Savings: The system's ability to reduce the economic burden caused by flood damage and recovery costs results in substantial cost savings for governments, insurance providers, and affected communities.
6. Scalability: The project can be easily expanded to cover larger geographic areas and integrate additional sensors, making it adaptable to different regions and flood-prone zones.
7. Improved Response Time: The IoT-based system automates the process of data collection and analysis, reducing response time and ensuring that flood alerts reach relevant authorities and communities faster than traditional methods.
8. Enhanced Data Accuracy: IoT sensors and data analytics provide accurate and reliable flood data, minimizing the chances of false alarms and ensuring that alerts are based on scientific evidence.
9. User-Friendly Interface: The user interface, such as a mobile app or web application, offers a convenient and accessible platform for users to monitor real-time data, receive alerts, and make informed decisions.
10. Increased Resilience: By providing advanced warning and better flood management capabilities, the project enhances community resilience and the ability to recover quickly from flood-related disasters.
11. Environmental Monitoring: In addition to flood warnings, the system can monitor environmental conditions, helping to assess and protect the natural environment in flood-prone areas.
12. Public Awareness: By raising awareness about flood risks and providing educational resources, the project encourages communities to take proactive measures and be better prepared for flood events.

In summary, the IoT-Based Flood Monitoring and Early Warning System offers a comprehensive and technologically advanced approach to flood risk mitigation, with the potential to save lives, protect assets, and improve overall disaster management strategies.

**CONCLUSION :**

In conclusion, the IoT-Based Flood Monitoring and Early Warning System represents a cutting-edge approach to flood risk mitigation and disaster preparedness. By utilizing technology and data-driven solutions, this project seeks to make communities more resilient in the face of a growing threat.