***FLOOD MONITORING AND EARLY WARNING***

***ABSTRACT:***

1.INTRODUCTION

2.COMPONENTS

3.COMPONENTS DETAILS

4.CODING

5.REAL TIME APPLICATION

6.CONCLUSION

***INTRODUCTION:***

Flood is a major known natural disaster that causes a huge amount of loss to the environment and living beings. So in these conditions, it is most crucial to get the emergency alerts of water level status at river beds in different conditions. In this project, the objective is to sense the water levels at river beds and check whether they are at a normal condition or not. If they reach beyond the limit, then it alerts the people through LED indications as well as through internet application. Here we are using an ultrasonic sensor to sense the river levels and a Node MCU ESP8266 to process these data. The data will be uploaded to Thing Speak IoT cloud, using which the river levels can be graphically monitored from anywhere in the world.

***COMPONENTS :***

***HARDWARE COMPONENTS:***

1.Bolt-IoT wi-fi module

2.ESP8266 nodemcu

3.Breadboard- 400 tie points

4.5mm LED:(Green, Red, Orange) and Buzzer

5.16×2 LCD Display

6.LM35 Temperature Sensor

7.HC-SR04 Ultrasonic Sensor

8.Some Jumper Wires

Male to Female Jumper Wires- 15 pcs

Male to Male Jumper Wires- 10 pcs

Female to Female Jumper Wires- 5 pcs

9. 9v Battery and Snap Connector

10.USB Cable Type B

***SOFTWARE COMPONENTS:***

1.Arduino IDE

2.Python 3.7 IDLE

3.Bolt IoT Cloud

4.Bolt IoT Android App

5. TWILLO SMS Messaging API

6. MAILGUN EMAIL Messaging API Software component

***COMPONENTS DETAILS:***

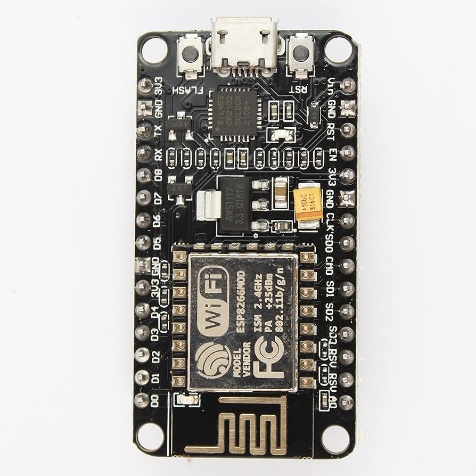
***HARDWARE COMPONENTS:***

**1. Bolt-IoT wi-fi module:**

****

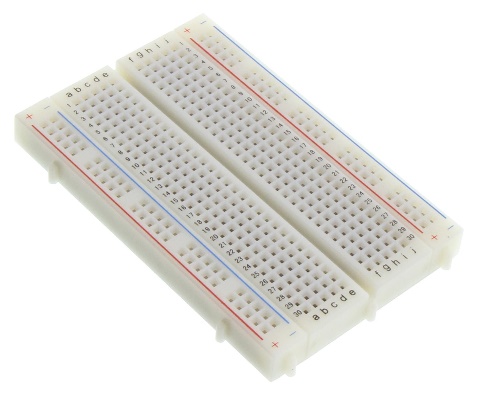
Used to ground and complete any circuit connection

**2.ESP8266 nodemcu:**

****

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

**3. Bread board**:



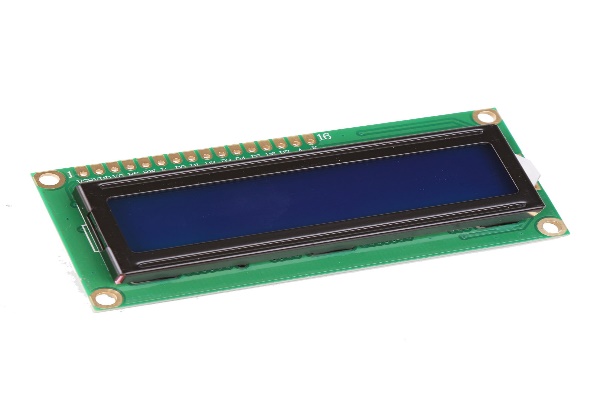
Used to create circuit.

**4**. **Light Emitted Diode:**

****

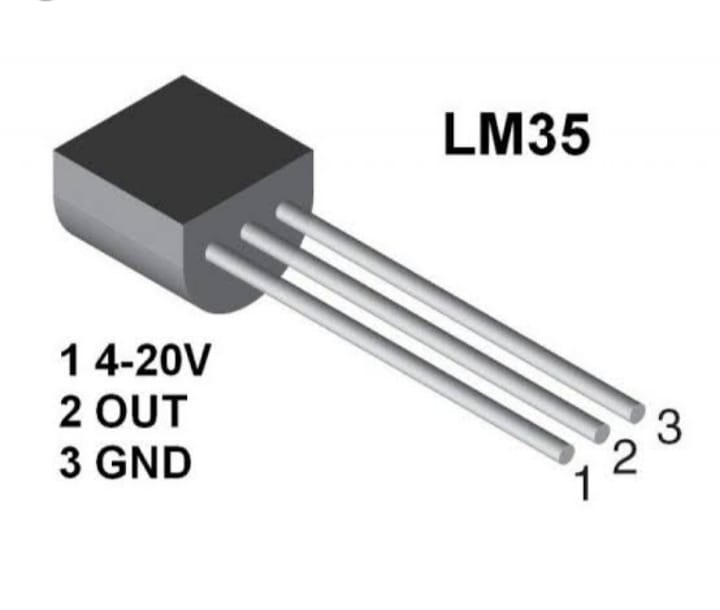
A device that produce a light on electrical and electronic equipment.

**5. LCD display:**

****

A type of flat panel display which uses liquid crystal in its primary form of operation.

**6. Temperature sensor:**

****

A device used to measure temperature.

**7. Ultrasonic sensor:**

****

It measure the distance to an object using ultrasonic sound waves.

**8. Jumper wires:**

****

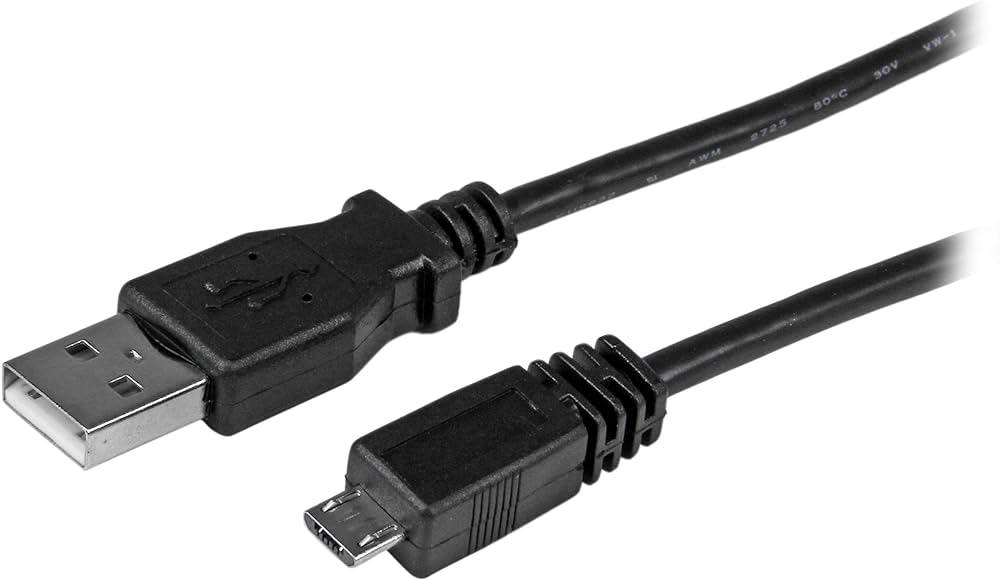
It connect remote electric circuit used for print circuit board.

**9. Battery and snap connector:**

****

For connecting 9v batteries.

**10. USB cable type B:**

****

It is used to connect large devices to your computer such as printers and scanner

***SOFTWARE COMPONENTS:***

**1.Arduino IDE:**

Arduino IDE (Integrated development environment) is used to write the computer code and upload this code to the physical board.

**2. Python:**

Developing website and software, task automation, data analysis and data visualization.

**3. Bolt IOT cloud:**

Gives you the capability to control your device and select data from IOT device safely and securely no matter where you are.

**4. Bolt IOT android app:**

Used to connect your actuators and sensors to the internet.

**5. TWILLO SMS messaging API:**

You can quickly integrate text messaging capability into your web, mobile, desktop application.

**6. MAILGUN email messaging API:**

Provide access to your numerous API end points that allows you to view and modify many aspects of your accounts and domain.

***CODING:***

* //IOT Based Flood Monitoring And Alerting System.
* #include<LiquidCrystal.h>
* LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
* const int in = 8;
* const int out = 9;
* const int green = 10;
* const int orange = 11;
* const int red = 12;
* const int buzz = 13;
* void setup() {
* Serial.begin(9600);
* lcd.begin(16, 2);
* pinMode( in , INPUT);
* pinMode(out, OUTPUT);
* pinMode(green, OUTPUT);
* pinMode(orange, OUTPUT);
* pinMode(red, OUTPUT);
* pinMode(buzz, OUTPUT);
* digitalWrite(green, LOW);
* digitalWrite(orange, LOW);
* digitalWrite(red, LOW);
* digitalWrite(buzz, LOW);
* lcd.setCursor(0, 0);
* lcd.print("Flood Monitoring");
* lcd.setCursor(0, 1);
* lcd.print("Alerting System");
* delay(5000);
* lcd.clear();
* }
* void loop() {
* long dur;
* long dist;
* long per;
* digitalWrite(out, LOW);
* delayMicroseconds(2);
* digitalWrite(out, HIGH);
* delayMicroseconds(10);
* digitalWrite(out, LOW);
* dur = pulseIn( in , HIGH);
* dist = (dur \* 0.034) / 2;
* per = map(dist, 10.5, 2, 0, 100);
* #map
* function is used to convert the distance into percentage.
* if(per < 0) {
* per = 0;
* }
* if (per > 100) {
* per = 100;
* }
* Serial.println(String(per));
* lcd.setCursor(0, 0);
* lcd.print("Water Level:");
* lcd.print(String(per));
* lcd.print("% ");
* if (per >= 80) #MAX Level of Water--Red Alert!{
* lcd.setCursor(0, 1);
* lcd.print("Red Alert! ");
* digitalWrite(red, HIGH);
* digitalWrite(green, LOW);
* digitalWrite(orange, LOW);
* digitalWrite(buzz, HIGH);
* delay(2000);
* digitalWrite(buzz, LOW);
* delay(2000);
* digitalWrite(buzz, HIGH);
* delay(2000);
* digitalWrite(buzz, LOW);
* delay(2000);
* }
* else if (per >= 55) #Intermedite Level of Water--Orange Alert!{
* lcd.setCursor(0, 1);
* lcd.print("Orange Alert! ");
* digitalWrite(orange, HIGH);
* digitalWrite(red, LOW);
* digitalWrite(green, LOW);
* digitalWrite(buzz, HIGH);
* delay(3000);
* digitalWrite(buzz, LOW);
* delay(3000);
* }
* else #MIN / NORMAL level of Water--Green Alert!{
* lcd.setCursor(0, 1);
* lcd.print("Green Alert! ");
* digitalWrite(green, HIGH);
* digitalWrite(orange, LOW);
* digitalWrite(red, LOW);
* digitalWrite(buzz, LOW);
* }
* delay(15000);
* }

**The above code is given to the Arduino to run the circuit.**

***CODING:***

#twillo details for sending alert sms

SID = 'You can find SID in your Twilio Dashboard'

AUTH\_TOKEN = 'You can find on your Twilio Dashboard'

FROM\_NUMBER = 'This is the no. generated by Twilio. You can find this on your Twilio Dashboard'

TO\_NUMBER = 'This is your number. Make sure you are adding +91 in beginning'

#bolt iot details

API\_KEY = 'XXXXXXXXX'

#This is your Bolt cloud API

Key.

DEVICE\_ID = 'BOLTXXXXXXXXX' #This is the ID of your Bolt device.

#mailgun details for sending alert E-mails

MAILGUN\_API\_KEY = 'This is the private API key which you can find on your Mailgun Dashboard'

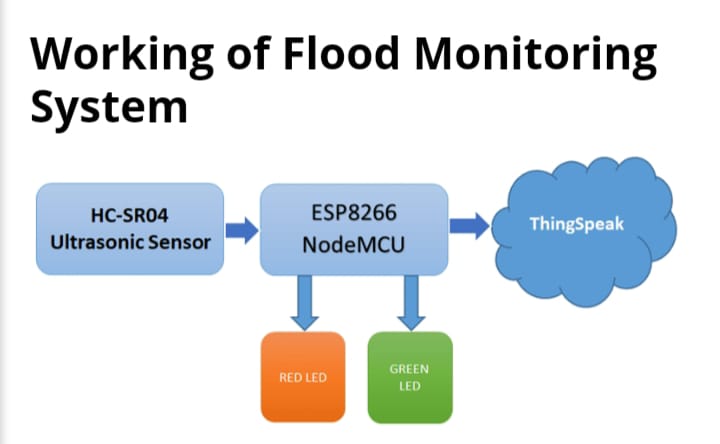
SANDBOX\_URL= 'You can find this on your Mailgun Dashboard'

SENDER\_EMAIL = 'test@ + SANDBOX\_URL' # No need to modify this. The sandbox URL is of the format test@YOUR\_SANDBOX\_URL

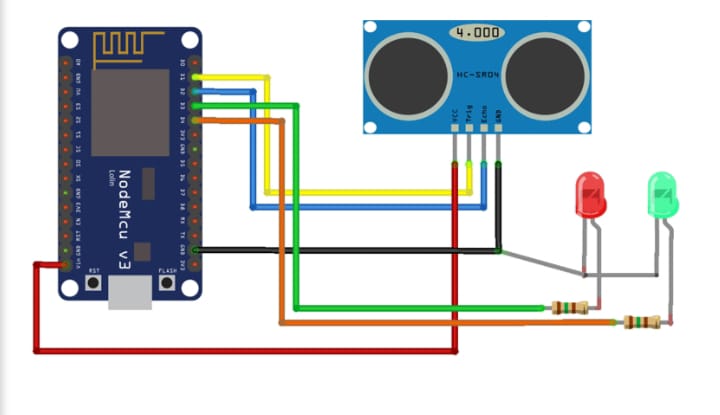
RECIPIENT\_EMAIL = 'Enter your Email ID Here'

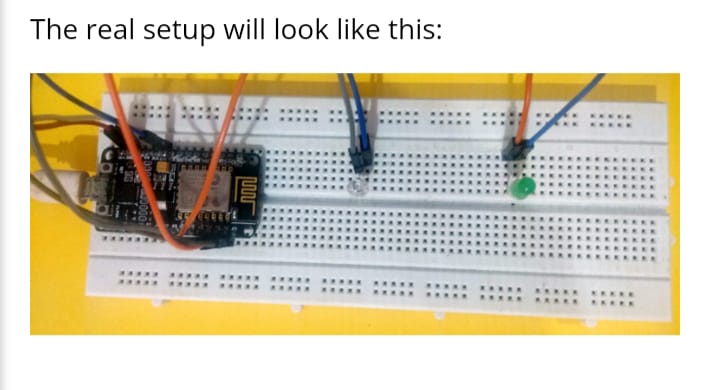
**This code will be helpful to send sms and email alert when the water level crosses threshold .**

***WORKING OF FLOOD MONITORING SYSTEM:***

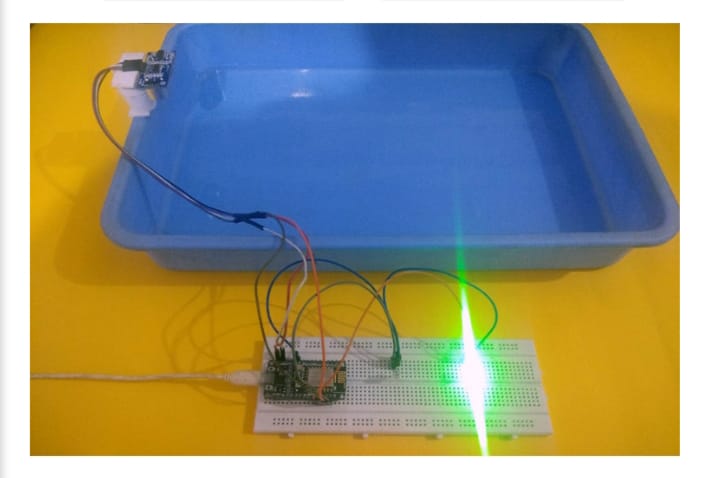
****

**CONNECTION OF ARDUINO TO LED AND ULTRASONIC SENSOR:**

****

****

***PROJECT OUTCOME:***

****

***REALTIME APPLICATION:***

**1. \*Sensor Networks:\***

Deploying sensors in flood-prone areas to monitor water levels, rainfall, and weather conditions in real time.

**2. \*Data Analytics Platforms:\***

Using advanced analytics to process data from various sources, such as river gauges, weather forecasts, and satellite imagery, to predict and identify potential flood events.

**3. \*Mobile Apps and Alerts:\***

Developing mobile applications that provide real-time alerts and warnings to residents in at-risk areas, allowing them to take necessary precautions**.**

**4. \*GIS and Mapping Tools:\***

Utilizing Geographic Information System (GIS) technology to create flood risk maps, track flood extents, and plan evacuation routes.

**5. \*Community Engagement:\***

Involving communities through social media, local networks, and community outreach to disseminate information and gather real-time updates from affected areas.

**6. \*Early Warning Systems:\***

Implementing automated systems that trigger alarms or notifications based on predefined thresholds of water levels, rainfall intensity, or weather forecasts.

**7. \*Collaboration with Authorities:\***

Integrating with local authorities, emergency response teams, and relevant stakeholders to ensure a coordinated and effective response to flood warnings and emergencies.

**8. \*Remote Sensing and Drones:\***

Employing satellite data and drone technology to monitor flood-prone areas and gather real-time imagery for assessing and responding to flood situations.

These applications work together to enhance the monitoring, prediction, and communication aspects of flood early warning systems, helping to reduce the impact of flooding on communities.

***CONCLUSION:***

*In conclusion, flood monitoring and early warning systems are critical in mitigating the devastating impact of floods on communities. Leveraging a combination of sensor networks, data analytics, mobile apps, GIS technology, community engagement, early warning systems, collaboration with authorities, and remote sensing can significantly improve preparedness and response. Real-time applications facilitate timely alerts, informed decision-making, and coordinated efforts, ultimately saving lives, reducing property damage, and enhancing overall resilience in the face of flood events. The continuous advancement and integration of these technologies and strategies are imperative in building more robust and effective flood monitoring and early warning systems.*

**Submitted by:**

1. PRIYA. L
2. SANJANA PANDEY. D
3. PRIYA DHARSHINI. S