Flood monitoring and early warning system

Introduction:

- The existing flood monitoring system consists of two microcontrollers and one sensor. The microcontroller used here is node MCU and the sensor used is an ultrasonic sensor which senses the level. The ultrasonic sensor continuously monitors the level of water each time it reaches the certain defined level. The flood warning system utilizes computer technology, database technology, communication technology, and sensor technology. Powered by IoT technology, rainfall and water levels are monitored and floods are predicted. Early warning of impending flooding can save lives and reduce extensive property damage.

Software components:

1. Arduino IdE
2. Python 3.7IDLE
3. Bolt iot cloud app
4. Bolt iot android app
5. Twillo sms messaging app

Proposed system

- The Proposed system consists of Rain Sensor, Ultrasonic Sensor, Power Source, Node MCU ESP8266, Buzzer and LEDs and finally Blank App. This Wireless Sensor Node is Kept in desired location Like dam, Bridge. etc. and Blynk App is downloaded by victims near the flooding area. The Schematic Diagram of the proposed flood forecasting and Monitoring System **fig 1.1.**

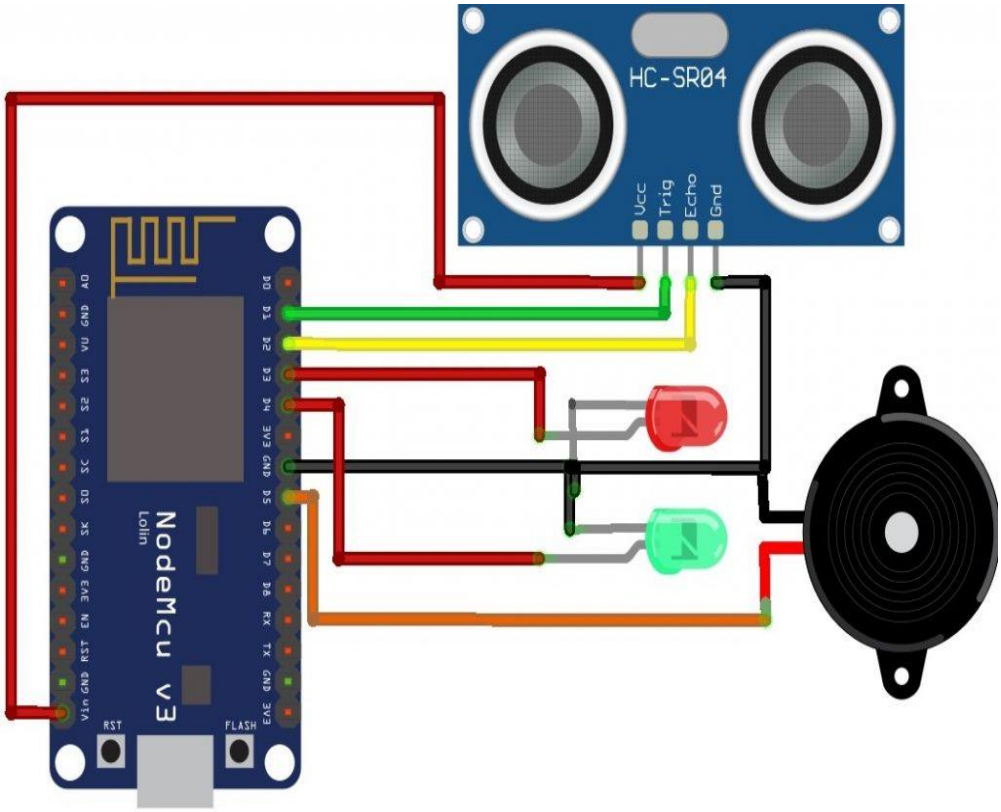


Fig 1.1 Circuit Diagram of Proposed System

ESP8266 Wi-Fi MCU:

- The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

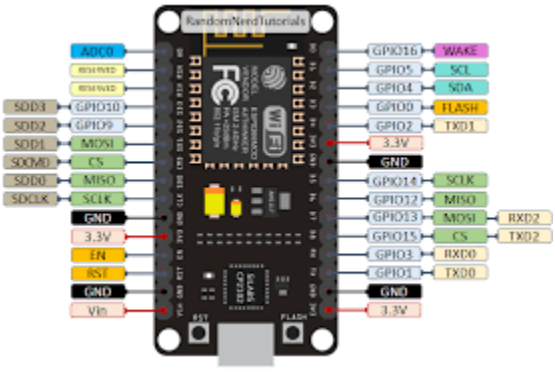


Fig1.2 **ESP8266 WiFi Module**

Program:

```
#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);
}
```

Connection to mobile app:

- The prototype works accordingly, an experiment was conducted to test the measurement of water detected by wireless sensor node. Buzzer and LED started to trigger when the water level reached 40 until it reaches critical level (62) in the gauge, a notification sent to victim through Blynk and email. Rain sensor detects the rain intensity and sends an alert when rain heavily started.

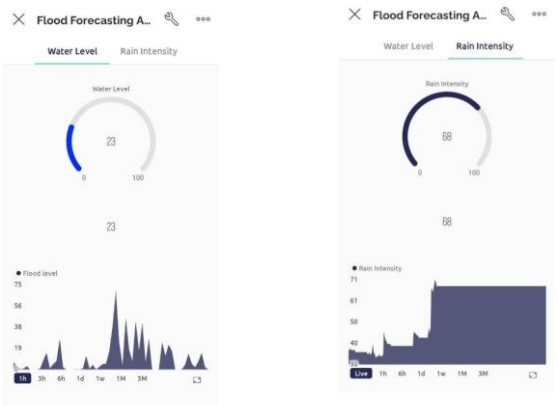
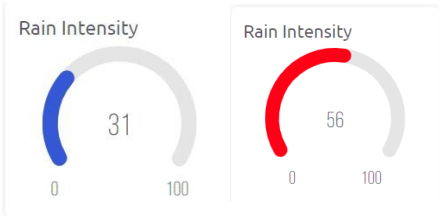


Fig 1.3: **Interface of Water Level and Rain Intensity**

- The level of rain intensity which is in Blue colour shows that the rain just started to fall. This indicates that the people who live nearby should alert as they know their place will get a very disastrous disaster if the rain started heavily. The user receives a "Rain Warning!!" warning in order to alert them



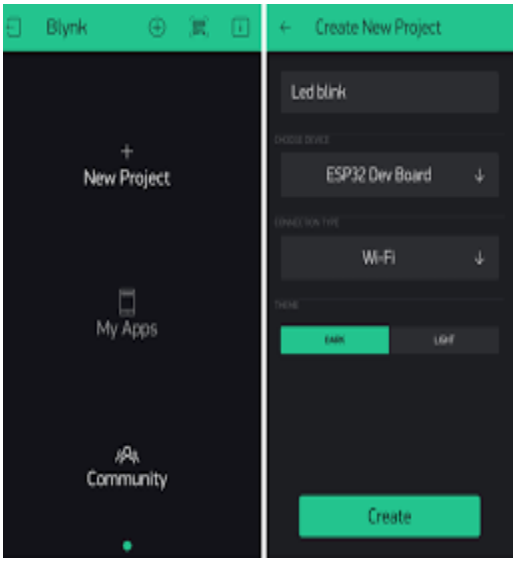
- Fig 1.**4 Low & High Rain Intensity**

fig1.5Medium & Critical levels of water

Fig1.6S: **Notifications in Blynk app**

Bylnk application:

- Blynk platform powers low-batch manufacturers of smart home products, complex HVAC systems, agricultural equipment, and everyone in between. These companies build branded apps with no code and get the full back-end IoT infrastructure through one subscription.



Conclusion:

•This project is built on creating a smart flood monitoring system with NodeMCU and Blynk application utilizing ultrasonic sensors. Flexibility, efficiency, and cheap cost are provided by the outcomes. A suitable platform for monitoring flash floods and issuing early warnings is a wireless sensor node based on the Blynk platform. In order to detect and give precise sensing data for monitoring and alerting purposes, ultrasonic sensors and a rain sensor connected with NodeMCU are able to work. Hence, the system shows that it may be utilized for flooding area detection, monitoring, and community warning.