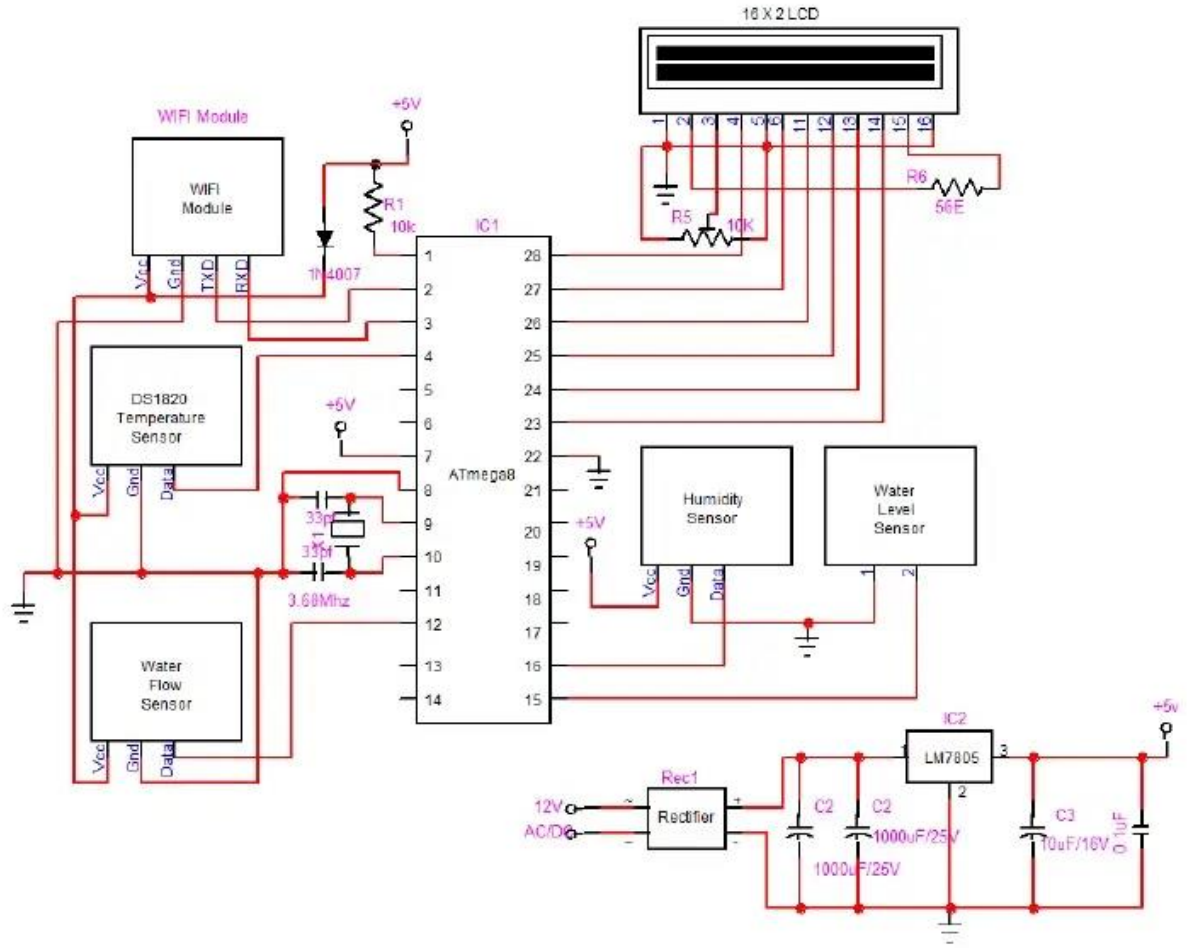
# FLOOD MONITORING AND EARLY WARNING SYSTEM

## 1.INTRODUCTION

In Recent years flooding became one of the major natural disasters occurringin India. India is among the top 10 in the world's most food-threatened country.There are many effects of floods where the material, human, economic and sociallosses are considered as some of the main effects of floods. Heavy rains are also oneof the major aspects for the causes of flash floods. In order to reduce the human andeconomic losses there are some necessary steps to be followed. One of the most andthe preliminary step is to alert the people before the occurrence of the disaster. Thereare some places with early flood alert systems but most of them are not most efficient as they can usually send the information to only some respectiveorganizations with limiting distances.

So, in case of floods it is taking more time for passing the message to the people living in the nearby areas so that the people could not save most of their belongings as water rises rapidly within less time. Usually, the flooding cannot beabandoned but the early detections can be made i.e., early alerting system with helpof continuous monitoring can be used to reduce the losses faced by the society. Inthis advanced technology there are some projects related to early flood monitoringsystem. At the initial stage a project to indicate the level of water and to alert thesurrounding people in remote areas using flood observatory system is bought upwhere the observatory system communicates with the monitoring system via GSMmodem in order to send the information of flow rate and to retrieve commands fromthe monitoring system. Secondly, the flood detections which estimates theinstantaneous water level at any instant of time by means of wireless sensor networksand provides GSM modem and then sends the notifications through the socialnetworks such as the Facebook and Twitter. Thirdly, the real-time flood monitoringsystem using wireless sensor networks are introduced which monitors the alteringand real-time data of river conditions.

# 2.CIRCUIT DIAGRAM




# 3.CIRCUIT DIAGRAM DESCRIPTION

For measuring the water level, we are using level sensor. This sensor usesmagnetic sensor to send the level of water. For calculating the water flow rate, weare using water flow sensor, its operating flow rate is0L/min to 50L/min. So, thissensor is sufficient to calculate the flowrate of a river. For processing of the data, weare using Atmega8 microcontroller. The humidity sensor produce dc voltage withrespect to the humidity. The atmega8 has inbuilt ADC so that it can easily read thesensor values.DS1820 is a temperature sensor IC which can sense upto 120 degree Celsius.It has inbuilt adc and so that microcontroller can read the values digitally. All thestatus are display over the LCD display. The flow sensor is a device for observingthe rate of fluid flow. Regularly a flow sensor is the sensing element used in a flowmeter for the footage of the flow rate.For web access we use ESP8266 wifi module which connects to

the internetthrough wifi. The fetched datas are updated to the web for every minute.

# 4.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);
}
```

- After writing the code. Verify the code and then upload the code to the specific Arduino using USB Cable type A. Remember while uploading select specific board you want to upload.

# Step 4.2: Writing the code in Python IDE.

- For writing python code we will be using python IDE.
- In this project we will be making two python files. One will be saved in the name of conf.py and other will be main.py.
- The purpose of making two files is to make the code understandable. Also this both python files will be usefull in sending sms and emails alerts to users.
- Now the most important part is arrived writing code in Python IDE. The full code is divided into two parts. The detailed code is given below.
- Open Python 3.7 IDE(Downloaded from the above section).
- Click on new file. Save the file in the name conf.py.

- **conf.py:** The file consists of important Api keys, Device id of Bolt IoT WiFi Module. Also it consists of important keys of Twillo and Mailgun respectively which will be further usefull in this project.
- Below is the complete structure of conf.py file. Make sure that you add the updated Bolt API key, device id and Mailgun and Twillo details respectively:

```
#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'
```

- After writing the conf.py now the last part is to write the main.py code. This code will be helpfull to send sms and email alerts when the water level crosses the threshold.
- Open the Python IDE.
- Click on new file. Save the file in the name main.py. Save the file in the same path where conf.py is saved.
- **main.py:** This file consists of the main coding facility. Discussed earlier it will be used to send sms and emails alerts. It will be also helpfull to keep close monitor on water level to send alerts whenever required.
- Below is the complete code of main.py.

```
import conf
from boltiot import Sms, Email, Bolt
import json, time
intermediate_value = 55
max_value = 80
mybolt = Bolt(conf.API_KEY, conf.DEVICE_ID)
sms = Sms(conf.SID, conf.AUTH_TOKEN, conf.TO_NUMBER, conf.FROM_NUMBER)
mailer = Email(conf.MAILGUN_API_KEY, conf.SANDBOX_URL, conf.SENDER_EMAIL,
conf.RECIPIENT_EMAIL)
def twillo_message(message):
```

```
  try:
     print("Making request to Twilio to send a SMS")
     response = sms.send_sms(message)
     print("Response received from Twilio is: " + str(response))
     print("Status of SMS at Twilio is :" + str(response.status))
  except Exception as e:
     print("Below are the details")
     print(e)
def mailgun_message(head,message_1):
  try:
     print("Making request to Mailgun to send an email")
     response = mailer.send_email(head,message_1)
     print("Response received from Mailgun is: " + response.text)
  except Exception as e:
     print("Below are the details")
     print(e)
     while True:
    print ("Reading Water-Level Value")
    response_1 = mybolt.serialRead('10')
    response = mybolt.analogRead('A0')
    data_1 = json.loads(response_1)
    data = json.loads(response)
    Water_level = data_1['value'].rstrip()
    print("Water Level value is: " + str(Water_level) + "%")
    sensor_value = int(data['value'])
    temp = (100*sensor_value)/1024
    temp_value = round(temp,2)
    print("Temperature is: " + str(temp_value) + "°C")
    try:
         if int(Water_level) >= intermediate_value:
            message ="Orange Alert!. Water level is increased by "
+str(Water_level) + "% at your place. Please be Safe. The current
Temperature is " + str(temp_value) + "°C."
            head="Orange Alert"
            message_1="Water level is increased by " + str(Water_level) +
"% at your place. Please be Safe. The current Temperature is " +
str(temp_value) + "°C."
            twillo_message(message)
            mailgun_message(head,message_1)
        if int(Water_level) >= max_value:
           message ="Red Alert!. Water level is increased by " +
str(Water_level) + "% at your place. Please Don't move out of the house.
The Current Temperature is " + str(temp_value) + "°C"
           head="Red Alert!"
           message_1="Water level is increased by " + str(Water_level) + "%
at your place. Please Don't move out of the house. The Current Temperature
is " + str(temp_value) + "°C."
           twillo_message(message)
           mailgun_message(head,message_1)
    except Exception as e:
        print ("Error occured: Below are the details")
        print (e)
    time.sleep(15)
```

# 5. Conclusion

Nowadays the Internet Of things (IoT) is broadly used in worldwide, this system will display the data of the water level measured on lcd display. This project can be very helpful to the Meteorological Department to continuously monitor the dams and river beds water level. With this project it can save many people lives by giving alerts when the water level crosses beyond the limit. This project is very cost-effective, flexible and productive in areas where flood conditions happens everytime.