

# **FLOOD MONITORING USING IOT**

By,  
E.R.HARISH

## Writing the code in the Arduino IDE

- Open the Arduino IDE(Downloaded from the above section).
- Click on new file. Choose the correct file path to save the file. Give appropriate name to the file and add .cpp extension to the file and save the file.
- Now the core part of the project is writing code for Arduino Uno. Below this line complete code is given. You can refer the below code.
- 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.

```
//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);
}|
```

# 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 useful 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 Twilio 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 Twilio details respectively

- 
- After writing the `conf.py` now the last part is to write the `main.py` code. This code will be helpful 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 helpful 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)

```

- After Successfully writing code for Arduino and Python. Now it is the time to test and demonstrate the project