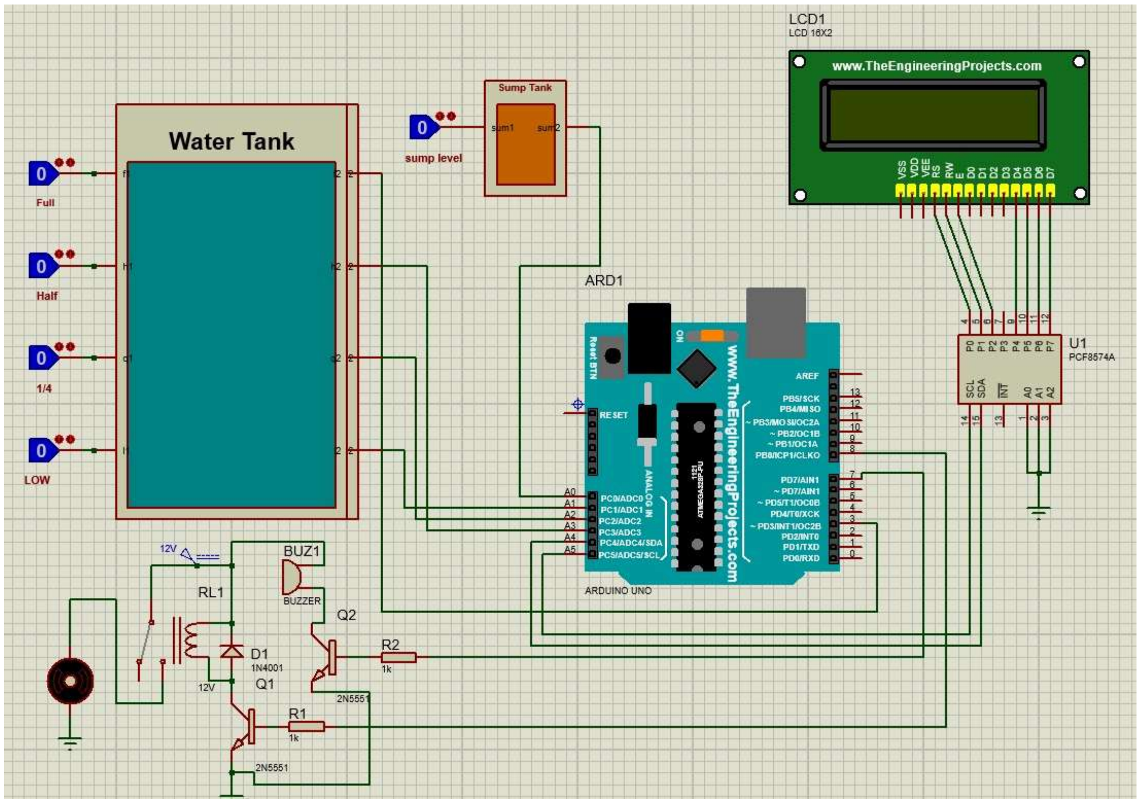
FLOOD MONITORING AND EARLY WARNING SYSTEM

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 respectiveorga nizations 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 ca nnot 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 th einstantaneous water level at any instant of time by means of wireless sensor networksand provides GSM modem and then sends the notifications through t he 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

The circuit diagram you sent shows a water tank with an Arduino board and a 16x2 LCD display. The Arduino board is used to read the water level sensor and control the water pump. The LCD display is used to display the water level.
The circuit works as follows:

- The water level sensor is connected to analog pin A0 of the Arduino board.
- The water pump is connected to digital pin D9 of the Arduino board through a relay.
- The LCD display is connected to I2C pins A4 and A5 of the Arduino board.

When the water level is low, the water level sensor will send a low signal to the Arduino board. The Arduino board will then turn on the water pump to fill the tank. When the water level reaches the desired level, the water level sensor will send a high signal to the Arduino board. The Arduino board will then turn off the water pump.The LCD display shows the water level in the tank as a percentage. The Arduino board calculates the water level percentage by reading the voltage from the water level sensor.

3.1.Water level sensor:

The water level sensor is a device that measures the water level in a tank. The sensor typically consists of a series of electrodes that are submerged in the water. The resistance between the electrodes changes depending on the water level.

3.2.Arduino board:

The Arduino board is a microcontroller board that can be used to control a variety of electronic devices. The Arduino board has a number of analog and digital pins that can be used to read and write signals to other devices.

3.3.Relay:

A relay is a switch that is controlled by an electrical signal. Relays are often used to control high-power devices, such as motors and pumps.

3.4.LCD display:

An LCD display is a type of display that uses liquid crystals to produce images. LCD displays are often used in electronic devices, such as calculators and watches.

The circuit diagram you sent is a relatively simple example of how to use an Arduino board to control a water tank. More complex circuits can be created to implement additional features, such as alarms and automatic shutoff.

4.Program

```
#include 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);
}
  #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'
```

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.