INTRODUCTION:

As we all know that Flood is one of the major well known Natural Disasters. When water level suddenly rises in dams, river beds etc. Alot of Destruction happens at surrounding places. It causes a huge amount of loss to our environment and living beings as well. So in these case, it is very important to get emergency alerts of the water level situation in different conditions in the river bed.

The purpose of this project is to sense the water level in river beds and check if they are in normal condition. If they reach beyond the limit, then it alerts people through LED signals and buzzer sound. Also it alerts people through Sms and Emails alerts when the water level reaches beyond the limit.

HARDWARE COMPONENTS :

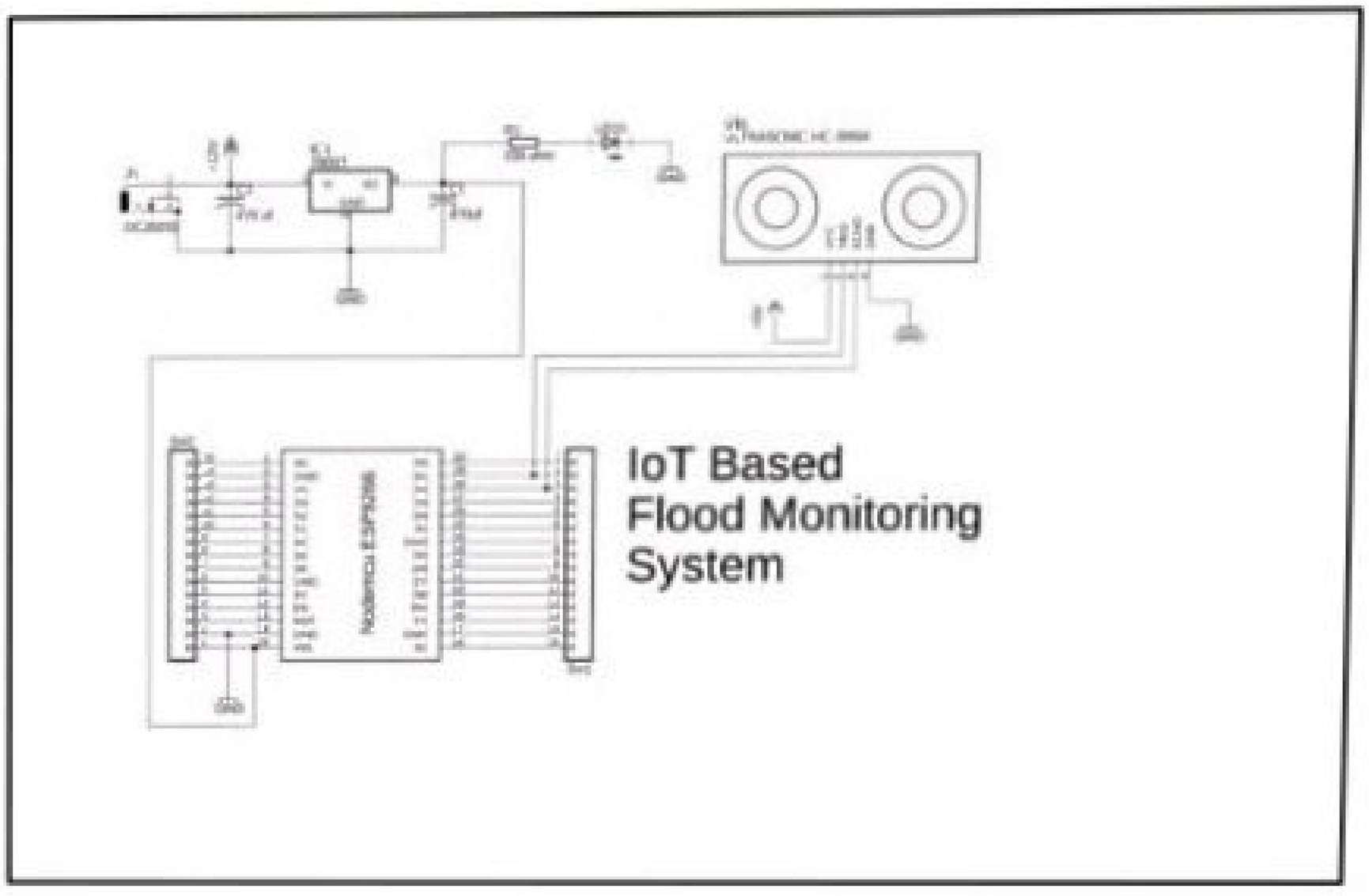
- Bolt-IoT wifi module
- Arduino uno
- Breadboard- 400 tie points
- 5mm LED:(Green, Red, Orange) and Buzzer
- 16×2 LCD Display
- LM35 Temperature Sensor
- HC-SR04 Ultrasonic Sensor
- Some Jumper Wires
- Male to Female Jumper Wires- 15 pcs
- Male to Male Jumper Wires- 10 pcs
- Female to Female Jumper Wires- 5 pcs
- 9v Battery and Snap Connector
- USB Cable Type B

SOFTWARE COMPONENTS:-

- Arduino IDE
- Python 3.7 IDLE
- Bolt IoT Cloud
- Bolt IoT Android App

- Twillo SMS Messaging API
- Mailgun EMAIL Messaging APISoftware components

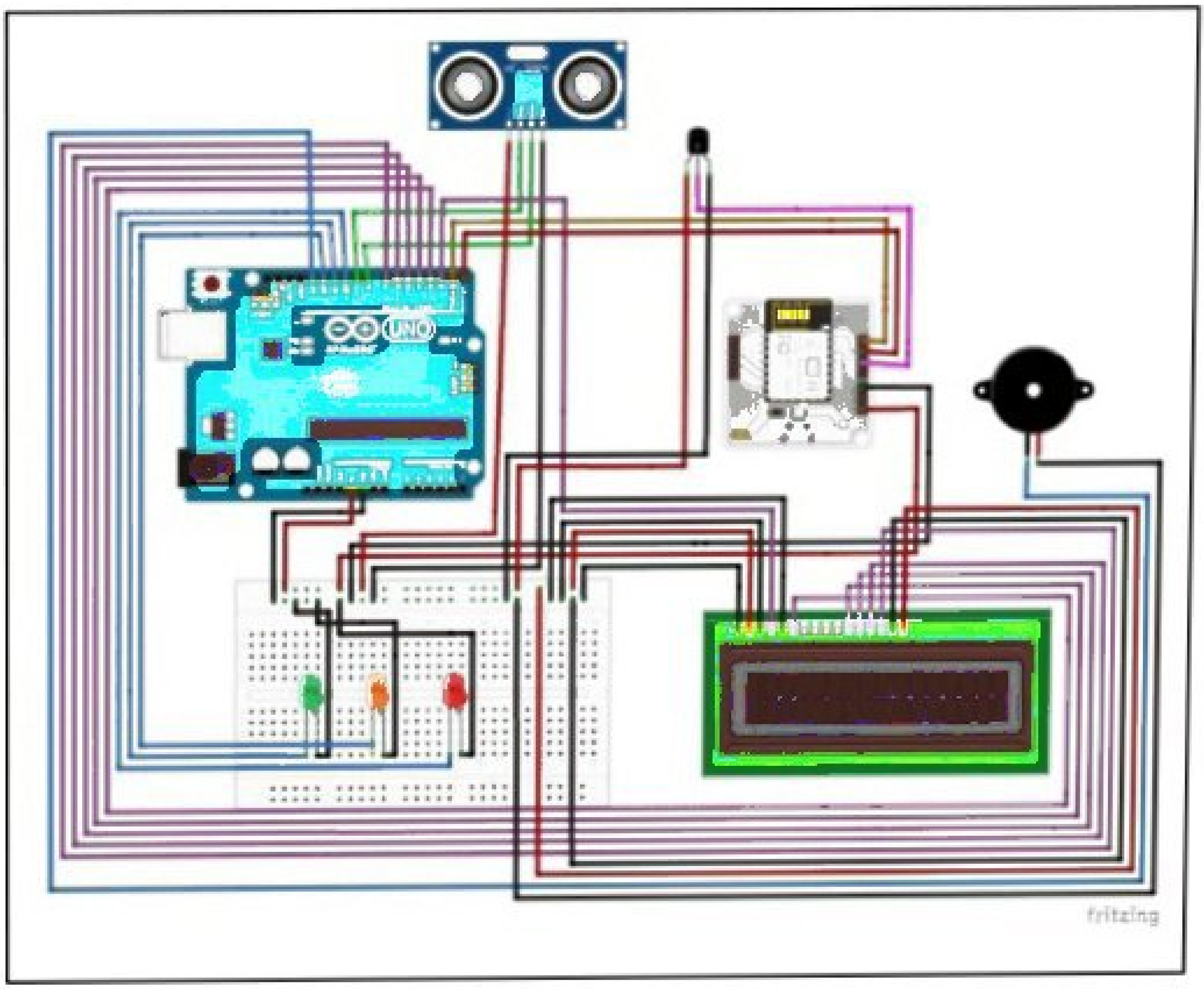
BLOCK DIAGRAM:



The blockt diagram of the IoT based Flood Monitoring System is very simple. The Nodemcu ESP8266 WiFi module is powered up using the 5V regulated power supply based on the LM7805 Voltage regulator. As the LM7805 Voltage Regulator accepts a wide range of input voltages (7 to 25 Volts), this project can be powered up using a 12V battery, a solar Panel, etc. Make sure the Input voltage does not exceed the maximum input voltage of the LM7805 Voltage Regulator.

Two resistor R2 "4.7K" and R3 "10K" are connected in series which makes the voltage divider circuit. This simply works as the voltage level converter; it converts 5 volts from the Ultrasonic Sensor into 3.3Volts.

SYSTEMATIC DIAGRAM:

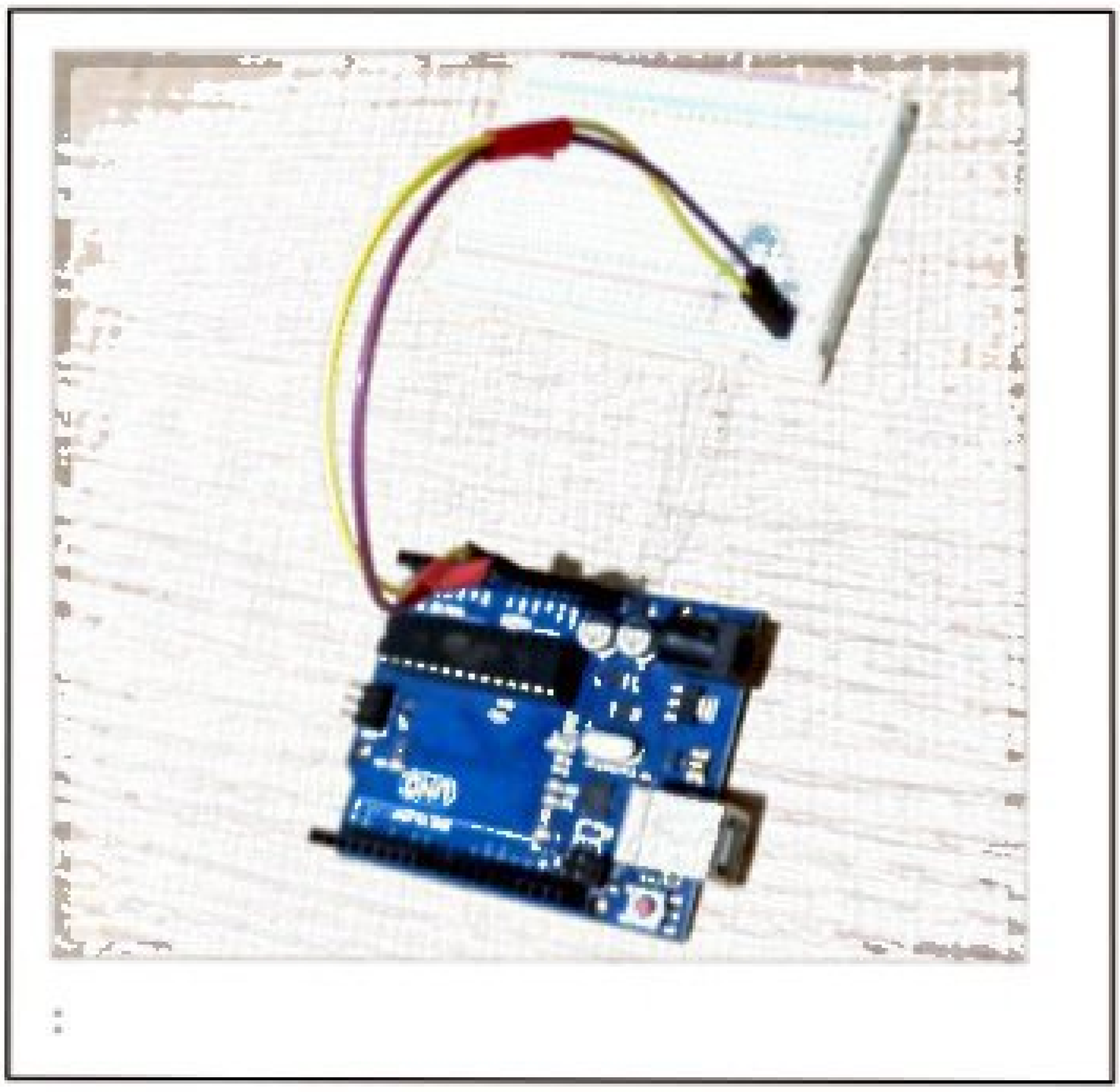


J1 is the female power jack and this is where we connect the external power supply used to power up the entire circuit. The external supply can be a 12V battery, an adaptor, a solar panel, etc. The voltage and ground pins of the female power jack J1 are connected with the Input and ground legs of the LM7805 voltage regulator. Two 470uF decoupling capacitors are connected at the input and output sides of the voltage regulator.

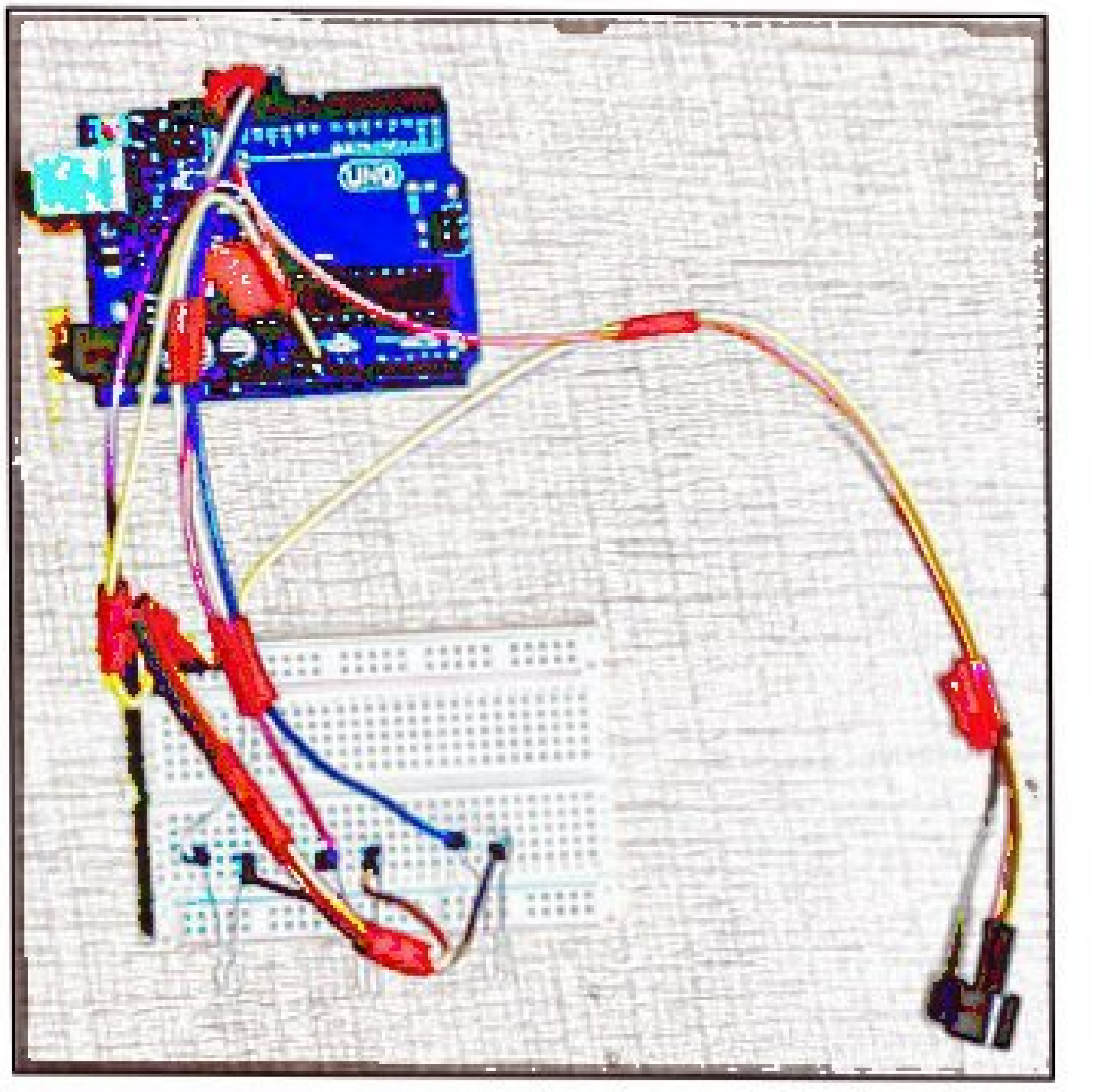
A 330-ohm resistor is connected in series with a 2.5v LED. This is a current limiting resistor. This is the circuit power ON LED. It has no role in this project, except to let you know that the circuit is powered up.

HARDWARE SETUP:

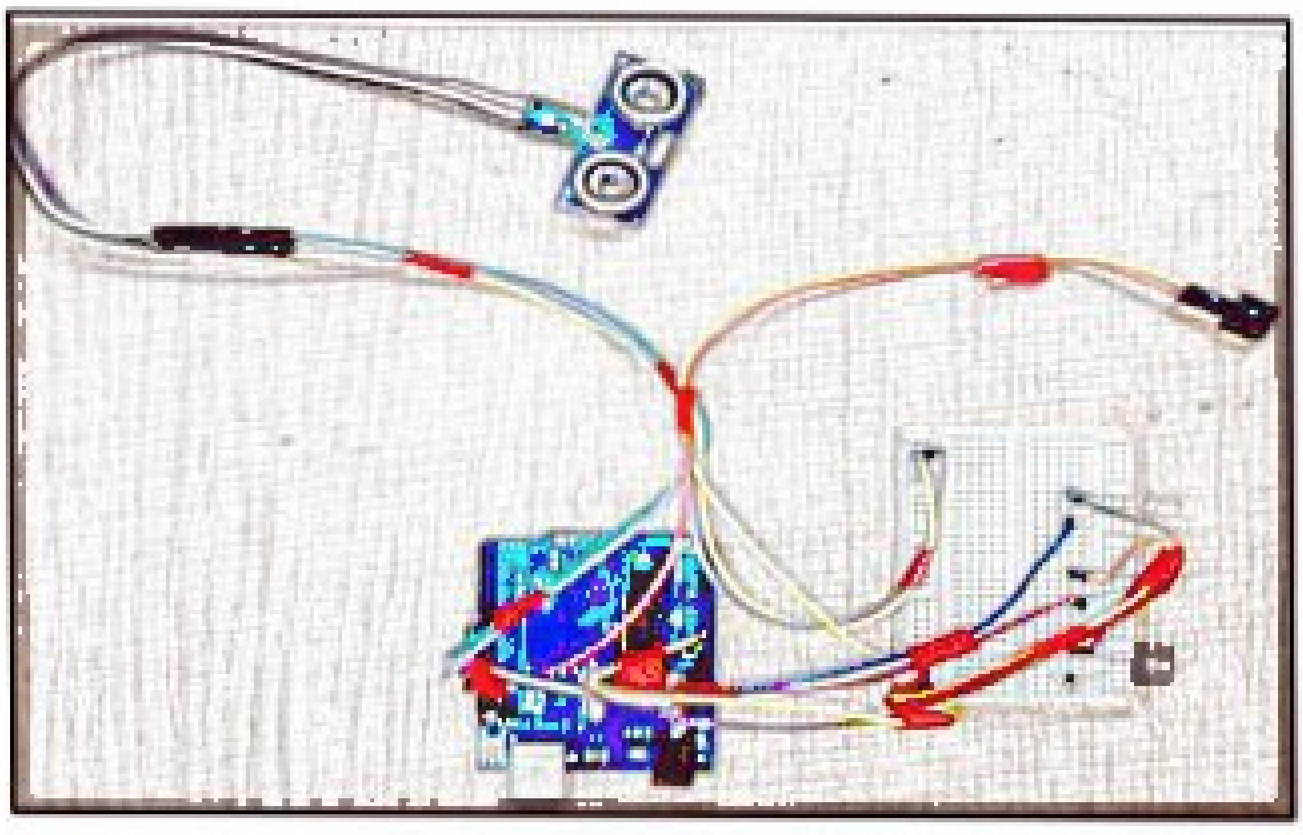
For Building this project we first configure the hardware connections. Then later on moving to the software part.



- VCC of Green Colour LED to Digital Pin ‘10’ of the Arduino.
GND of Green Colour LED to the GND of Arduino.
VCC of Orange Colour LED to Digital Pin ‘11’ of the Arduino.
GND of Orange Colour LED to the GND of Arduino.
VCC of Red Colour LED to Digital Pin ‘12’ of the Arduino.
GND of Red Colour LED to the GND of Arduino

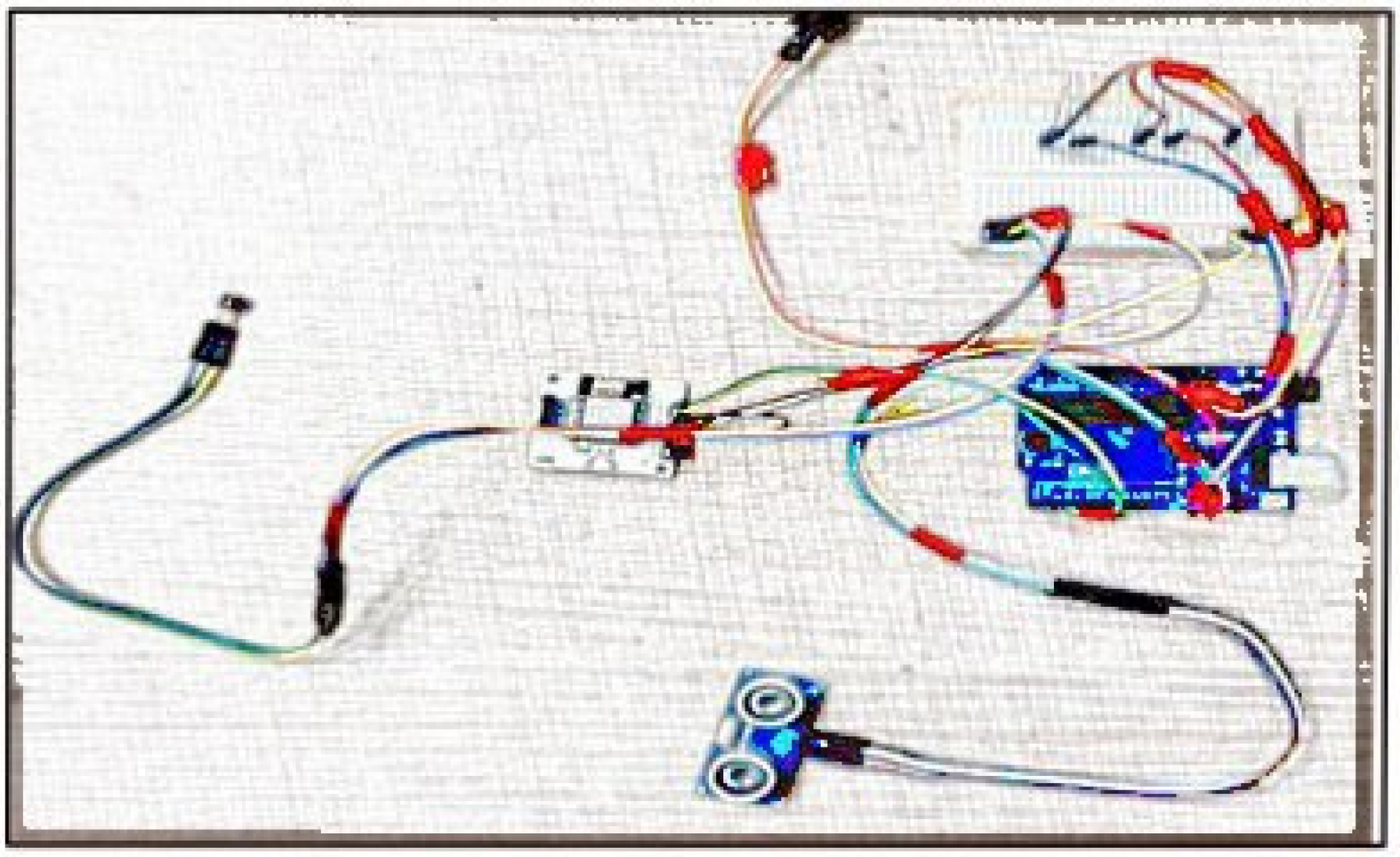


VCC of Ultrasonic Sensor to 5v of Arduino.
GND of Ultrasonic Sensor to GND of Arduino.
Echo of Ultrasonic Sensor to Digital Pin '8' of Arduino.
Trig of Ultrasonic Sensor to Digital Pin '9' of Arduino.



5v of Bolt WiFi Module to 5v of Arduino.
GND of Bolt WiFi Module to GND of Arduino.
TX of Bolt WiFi Module to RX of Arduino.

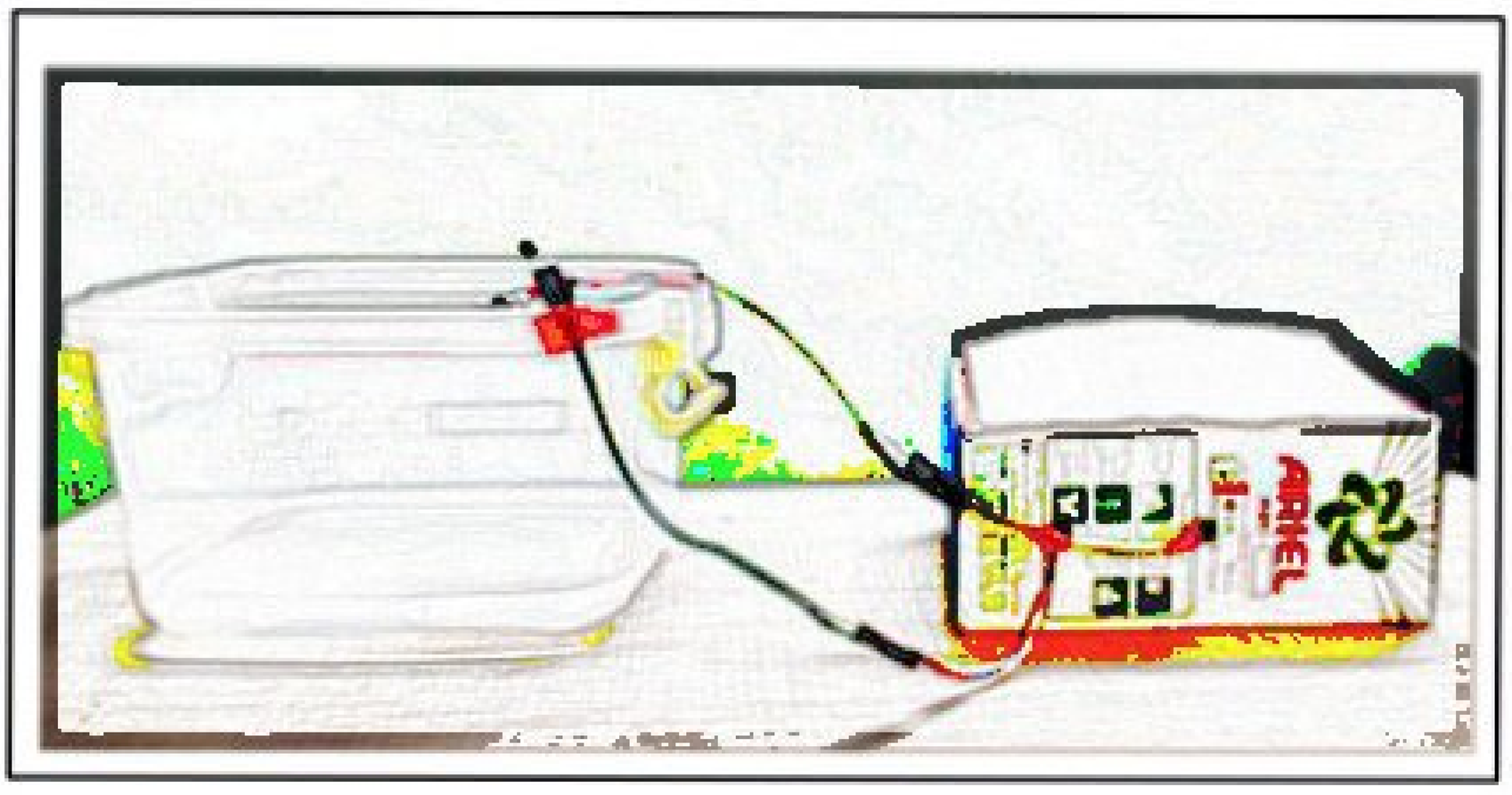
RX of Bolt WiFi Module to TX of Arduino.

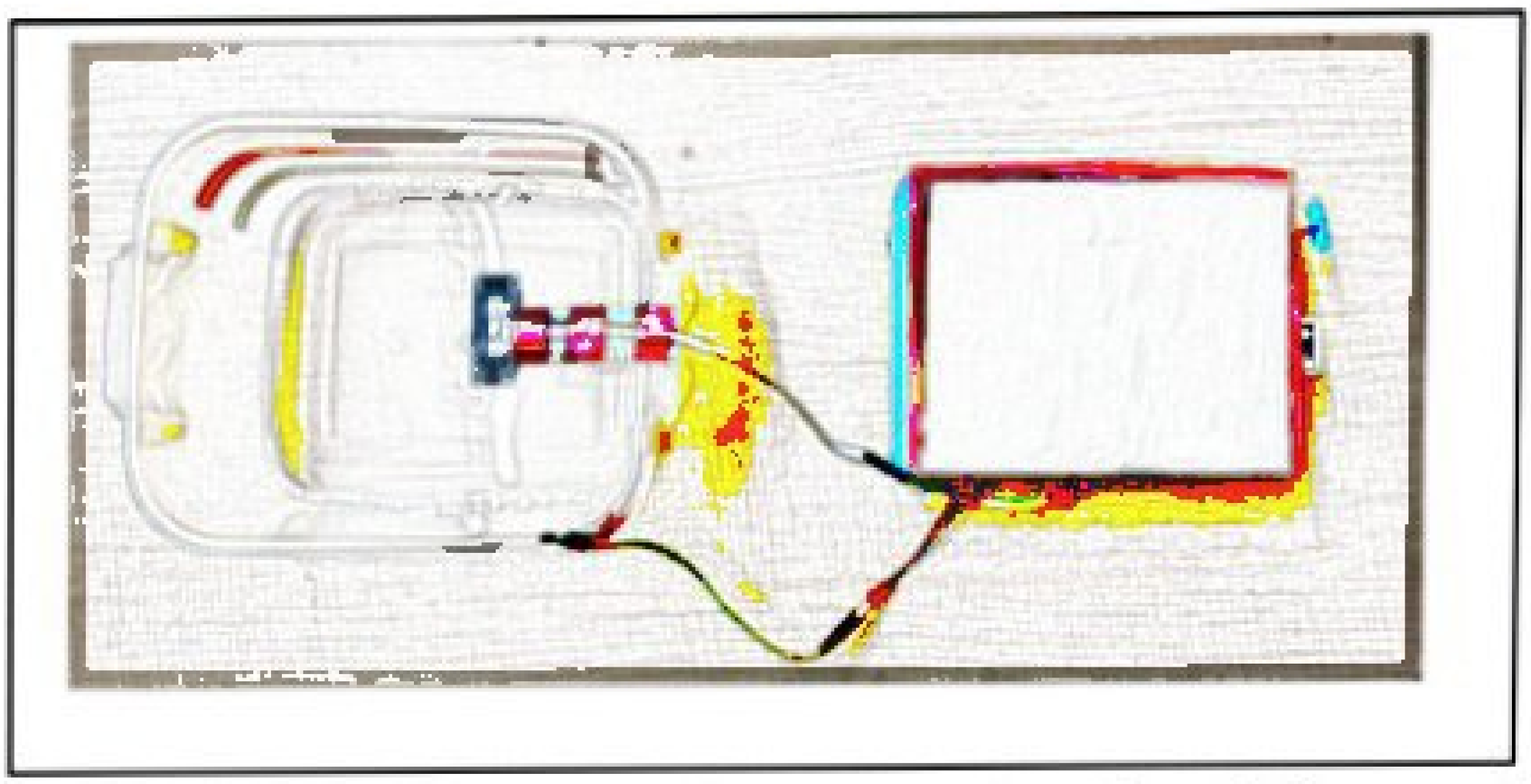


- Pin 1,3,5,16 of 16×2 LCD to GND of Arduino.
- Pin 2,15 of 16×2 LCD to 5v of Arduino.
- Pin 4 of 16×2 LCD to Digital Pin '2' of Arduino.
- Pin 6 of 16×2 LCD to Digital Pin '3' of Arduino.
- Pin 11 of 16×2 LCD to Digital Pin '4' of Arduino.
- Pin 12 of 16×2 LCD to Digital Pin '5' of Arduino.
- Pin 13 of 16×2 LCD to Digital Pin '6' of Arduino.

Pin 14 of 16×2 LCD to Digital Pin '7' of Arduino

After doing the hardware connection put all the hardware components in one box.





Also attach LM35 Temperature Sensor on the side of the container.Also attach Ultrasonic sensor on the top of the container.

SOFTWARE PROGRAM:

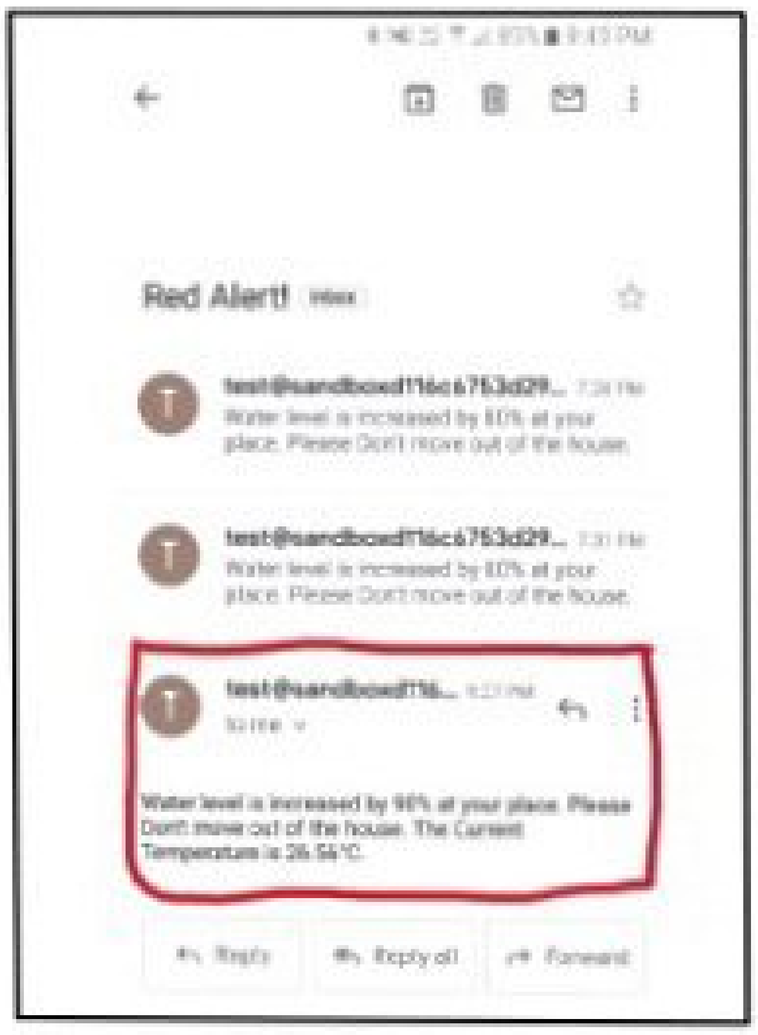
After writing the code. Verify the code and then upload the code to the specific Arduino using USB Cable type A. Remember while uploading select specific board you want to upload

```
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).
```

Also Sms and Email is send to registered user with proper message and current temperature of that place



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

After Successfully writing code for Arduino and Python. Now it is the time to test and demonstrate the project. Move to next section for demonstration of the project.