LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5068NI– Cloud Computing & IoT

<<Flood Detection System>>

Assessment Type

50% Group Report
Semester
2024 Spring

Group Members

London Met ID	Student Name
23050262	Sujal Parajuli (Leader)
23049184	Shrine Ghimire
23049338	Sakshyam Kafle
23047617	Niran Bhatta
23049029	Sarthak Baniya
22067652	Paras Kumar Yadav

Assignment Due Date: January 20, Monday

Assignment Submission Date: January 20, Monday

Submitted to: Mr. Sugat Man Shakya

Word Count:3287

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

23050262 Sujal Parajuli.docx

Islington College,Nepal

Document Details

Submission ID
trn:oid:::3618:79603011

Submission Date
Jan 20, 2025, 1:27 AM GMT+5:45

Download Date
Jan 20, 2025, 1:29 AM GMT+5:45

File Name
23050262 Sujal Parajuli.docx

File Size
20.7 KB

27 Pages

3,241 Words

17,488 Characters

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **20** Not Cited or Quoted 12%
  Matches with neither in-text citation nor quotation marks
- **2** Missing Quotations 2%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

12% Internet sources
2% Publications
12% Submitted works (Student Papers)

Acknowledgement

We would like to express our heartfelt thanks to our teacher, **Mr. Sugat Man Shakya**, for his amazing support and guidance while we worked on our flood detection system. His knowledge and encouragement helped us understand the project better and improve our skills.

We also would like to thank our other tutors and lecturers for their encouraging advice and support. Your feedback has enriched our learning experience so much and has inspired us to perform the best. We would like to extend special thanks to our team members. Their hard work, teamwork, and dedication were the major keys to the success of this project. It was a pleasure working together, and we appreciate each one of your works and contributions.

Abstract

As we know, flooding is one of the natural disasters which can damage different homes, places also the whole city. We cannot cure this disaster but if we find floods early it can help to reduce the damage. So, we are planning to make such a device which uses Arduino Uno, breadboard, an ultrasonic distance sensor and a piezo speaker which will show or warn us before it happens. Ultrasonic sensor will measure the level of water when it reaches a certain point then with the help of Arduino triggers the piezo speaker will create a sound which will warn us before it happens. Code for this device will be written on Arduino IDE which makes it functional. This device will show how modern technology can help peoples make safe from flood and its damage.

Table of Contents

Table of Figure

Table Of Table

1. Introduction:

The Internet of Things is modifying how different problems in the world are being tackled; hence, this project tackles using IoT for developing a flood detection system. The "Flood Detection and Monitoring System" is made to track and keep an eye on the water levels in a specific home. The threat created by neighboring rivers, dams, and other possible flooding sources is what this project attempts to tackle. It enables people and families to take prompt action, protecting lives and property, by giving them real-time warnings. The protection of vulnerable populations, such as children, the elderly, and families residing in high-risk locations, is given particular attention by this system. Such a natural phenomenon may result in huge losses including loss of lives, injuries to people, and infrastructure devastation. Considering all these challenges in mind, it becomes important that a flood detection system provides timely warnings to allow the most effective responses. Thus, this project uses IOT technology to monitor the water levels and can predict floods which will help to reduce disaster impacts (ORACLE, 2024).

1.1 Current Scenario:

For many places that experience flooding and excessive rain, flood detection has been a major concern. Some clever ideas have been developed to assist in solving this issue. The Flood Detection System is one of them which monitors rising water levels using water level sensors positioned in strategic locations. The system notifies local officials and citizens via a dedicated mobile notification IOT system, when the water level rises too high. As an early warning, this alert allows people to take precautions or, if needed, leave the area. To guarantee prompt reaction and keep people safe during possible floods, in future, the system can also collaborate with nearby emergency services for timelier rescue.

1) **Using Different Types of Sensors**: Instead of completely relying on water level sensors, it would be helpful to use a variety of sensors. They can be helpful to measure the correct water level and alert the residents about the disaster immediately
2) **Automatic Security Alerts:** Enhancing the IoT alert system to deliver personalized notifications could really help. For instance, if a flood is approaching, people could receive SMS or app notifications based on their specific locations and preferences.

1.2 Problem Statement and Project as a Solution:

Heavy rainfall from 27 to 28 September brought about floods in Nepal that caused unprecedented destruction, affecting thousands of children and their families. According to government reports, at least 217 people, including 35 children (11 girls and 24 boys), lost their lives, while many others were injured or went missing. Hundreds of homes were destroyed in the floods, leaving thousands of people displaced and exposed to harsh weather (Unicef, 2024).

Figure 1 Flood in Kathmandu

City	Date	Deaths	Injured	Casualties
Kathmandu	27 September	217	147	700+
Kathmandu, Sindupalchowk, Dhading	2 October	228	158	1000+
Sunsari	28 September	47	95	355+
Udaypur	12 June	38	78	300+

Table 1 Problem Statement

The Flood Detection System tracks water bodies in real time using Internet of Things devices like water level sensors. These sensors are positioned thoughtfully in flood-prone locations, like lakes, rivers, and drainage networks. In our case we need to thoughtfully position the sensors in the water

level prone areas like in the walls of a house where we can detect the flood if the water levels are higher than usual and can alert the individual to flooding.

Key Features:

- **IoT Sensor Water Level Monitoring:** Real-time water level tracking using the sensors of the IoT and showing real-time information in terms of flood risk. Automatic notification of users, as well as authorities and emergency responders when water reaches critical level.
- **Simple Data Insights:** Supports improved decision-making by predicting floods using previous water level data.

Benefits:

1. **Enhanced Safety:** Timely warnings can save lives and reduce injuries during flooding events.
2. **Infrastructure Protection:** Early warnings can protect homes, schools, hospitals, and critical infrastructure from flood damage.
3. **Support for Vulnerable Populations:** Protection of children and families who are most vulnerable in cases of flooding by guaranteeing safe evacuation and access to emergency services.

1.3 Aims and Objectives:

1.3.1 Aim

The aim of this work is to design an IoT-based flood detection system that will provide real-time monitoring and prediction of flooding events to reduce the impact of floods on human lives and properties through timely warnings and notifications to the community, first responders, and local government. The system will also be used to improve the preparedness and response capabilities in communities that are prone to flooding.

1.3.2 Objective

The objectives of this project are:

- **Design and Implementation:** Implement a continuous IoT-based mechanism for monitoring water levels within certain households with regards to accurate and timely data.
- **Real-Time Warning System:** A real-time warning mechanism that would keep the local communities, authorities, and even emergency teams updated regarding possible flood hazards; thereby prompt actions can be considered accordingly.
- **Scalability and Adaptability:** This system shall be scalable and adaptable to other flood-prone areas of the world to enable quick deployment as a solution to similarly vulnerable communities.

2. Background

We plan to conduct the test to determine the level of water when it is flooded. The plan includes the design of a flood alarm system which gives early water level detection warning. The water level categories include safety and emergency warnings. The residential areas and other sectors which are near the risk area of unexpected flood become more cautious of flood disasters because of this system prototype. Therefore, by using this system, we can solve the issue and monitor the unexpected flood.

2.1 System Overview

The project is to develop a flood detection system by using NodeMCU microcontroller, Breadboard, Raindrop Sensor Module, Piezo, jumper wires to conduct the system. The raindrop sensor measures the water level and provides early warning through the Piezo buzzer.

Its Mechanism:

- Sensors: Water level sensors, such as raindrop sensors, measure the rising water levels.
- Microcontroller (ESP8266): Acts as the system's brain, processing the sensor inputs and triggering alerts.
- Alert Mechanism: A piezo buzzer sounds an alarm when the water level crosses a specified limit in terms of safety.

2.2 Design Diagram

2.2.1 Block Diagram

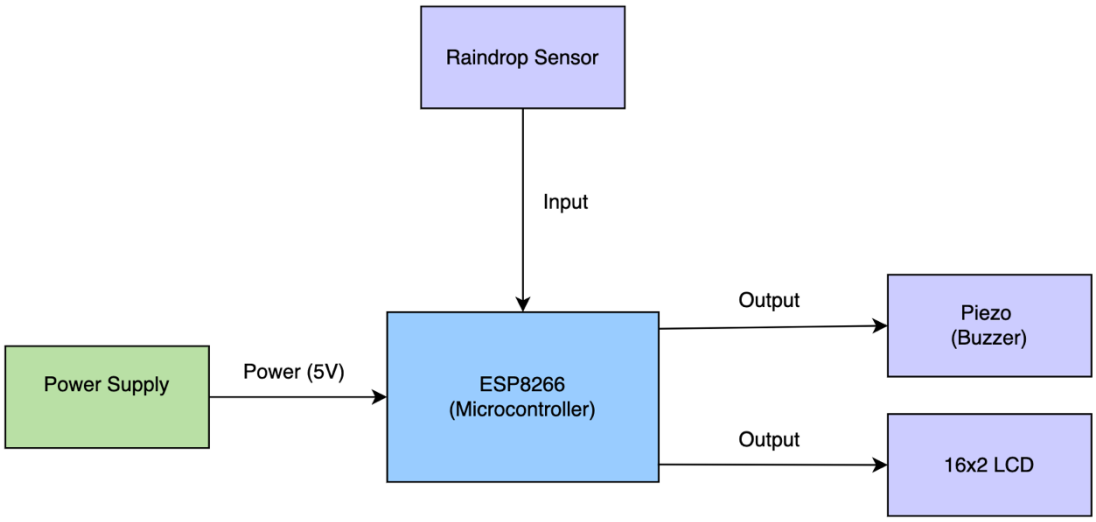


Figure 2 Block Diagram

2.2.2 System Architecture

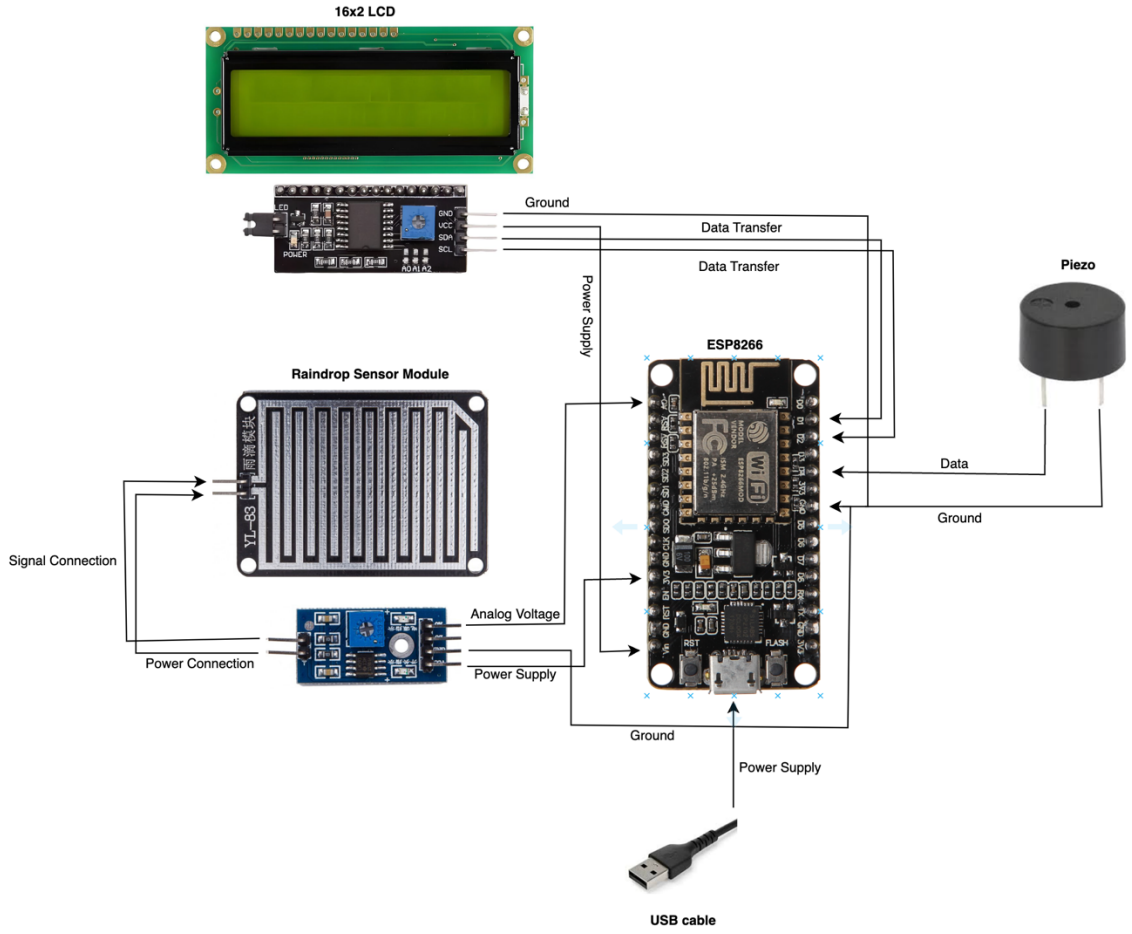


Figure 3 System Architecture

2.2.3 Schematic Diagram

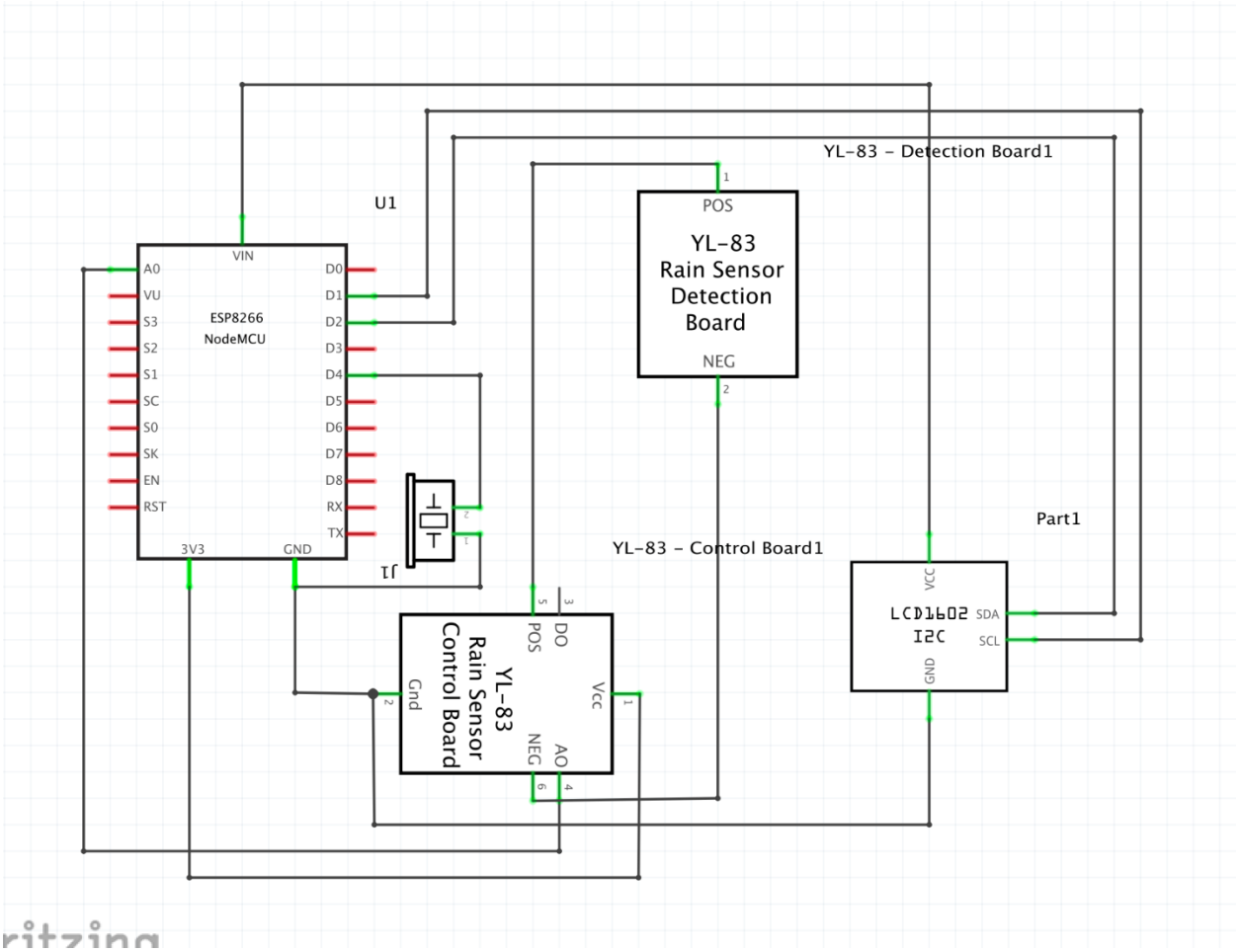


Figure 4 Schematic Diagram

2.2.4 Circuit Diagram

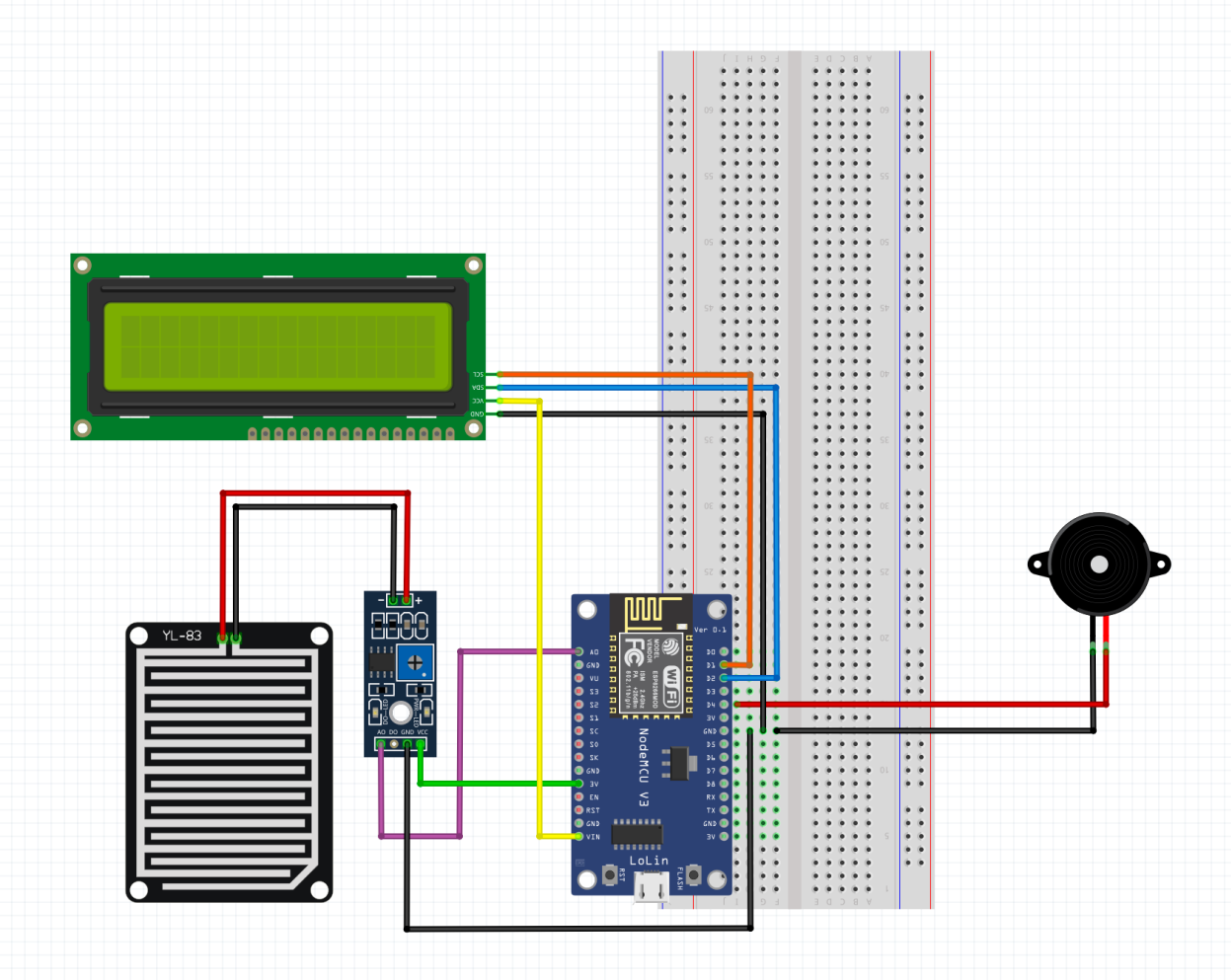


Figure 5 3D Circuit Diagram

2.2.5 Flow Chart

1. **System Flow:**

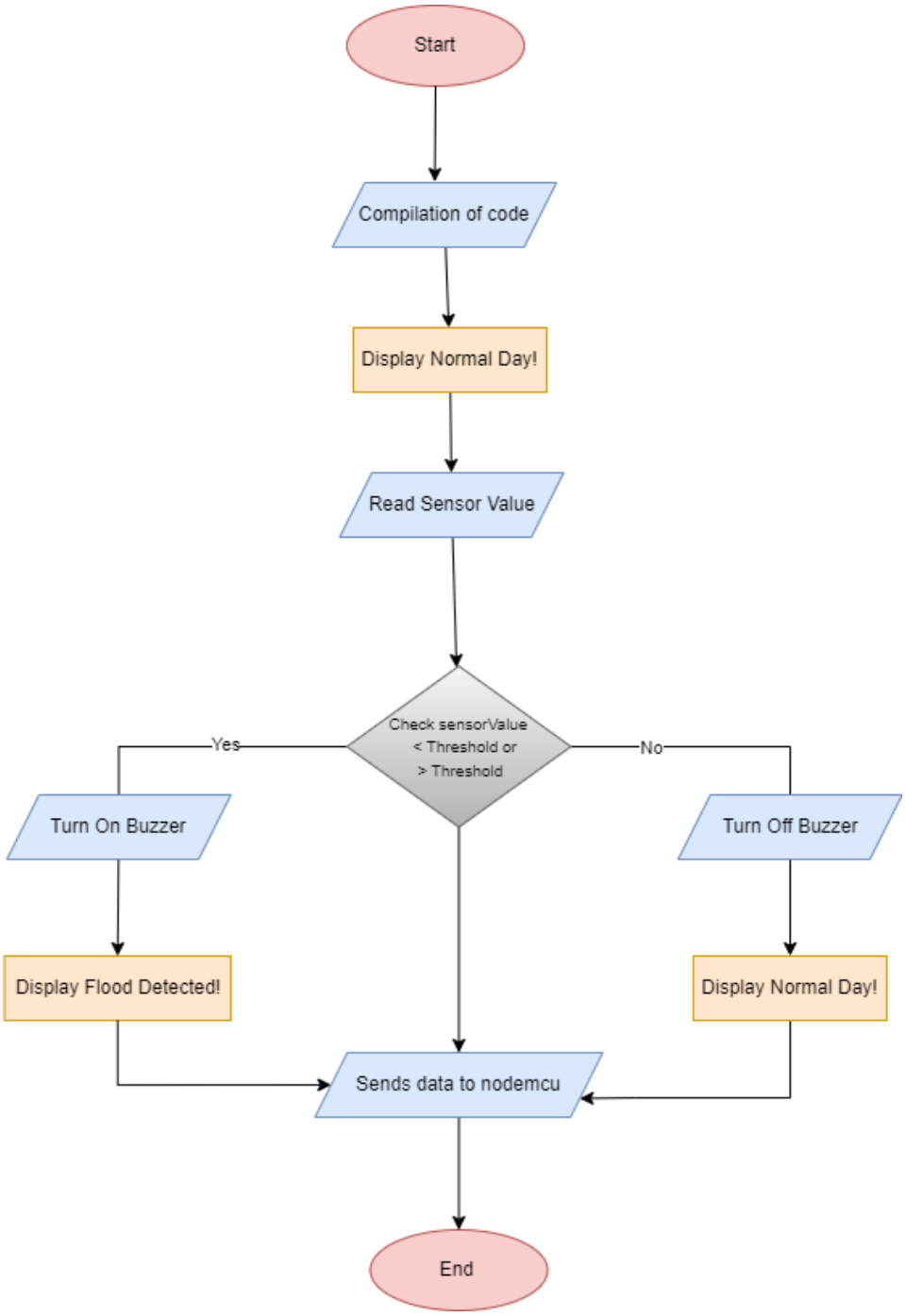


Figure 6 System Flow

2. Code Flow:

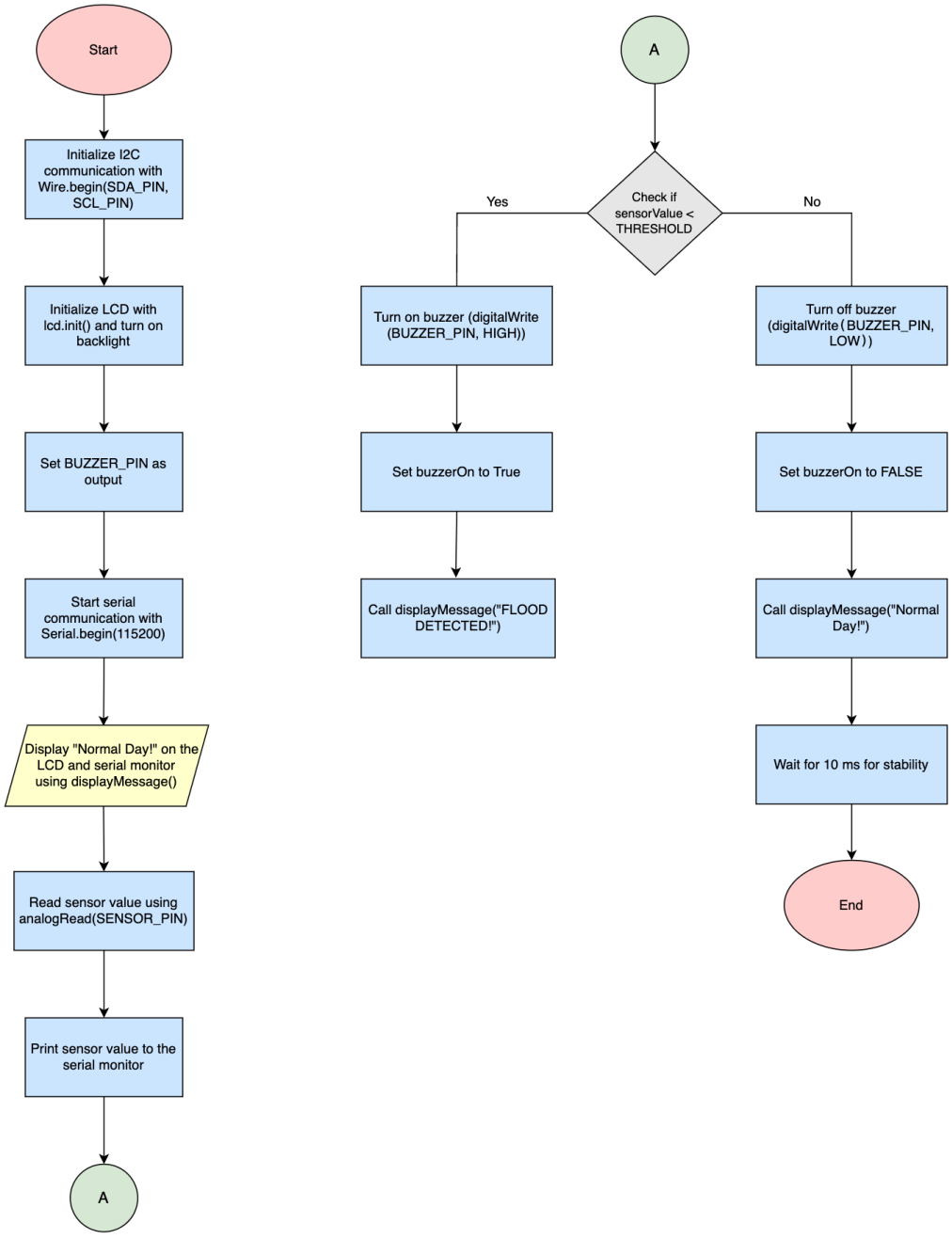


Figure 7 Code Flow

2.3 Requirements Analysis

2.3.1 Hardware Components

The project focuses on developing early detection and warning system for which the project requires:

- **NodeMcu:** NodeMCU is a firmware for ESP8266 developed using C Programming Language, Espressif NON-OS SDK and Lua Scripting Language. NodeMCU Firmware will interpret the bytecode and execute the commands. (Teja, 2024)

Figure 8 NodeMCU

- **16x2 LCD**: An LCD screen is an electronic display module that uses liquid crystal to produce a visible image. The 16x2 LCD display is a very basic module commonly used in DIYs and circuits. The 16x2 translated a display of 16 characters per line in 2 such lines. In this LCD, each character is displayed in a 5x7 pixel matrix. (CD-Team, 2023)

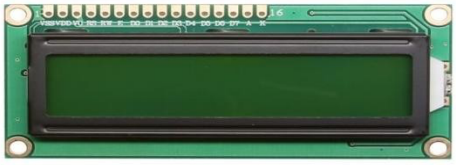


Figure 9 16x2 LCD

- **Raindrop Sensor**: Raindrop sensor is simple sensor that gives digital output commonly used to detect rain (water). It can also measure rainfall intensity as well as for weather monitoring (robocraze, n.d.).

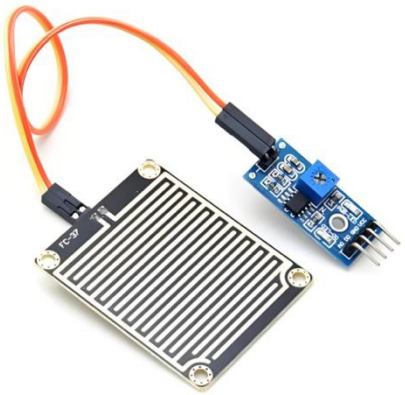


Figure 10 Raindrop Sensor

- **Breadboard**: A breadboard is simply a board for prototyping or building circuits on. It allows one to place components and connections on the board to make circuits without soldering. The holes in the breadboard take care of the connections by physically holding onto parts or wires where one can put them and electrically connecting the inside the board. (Wagner, 2024)

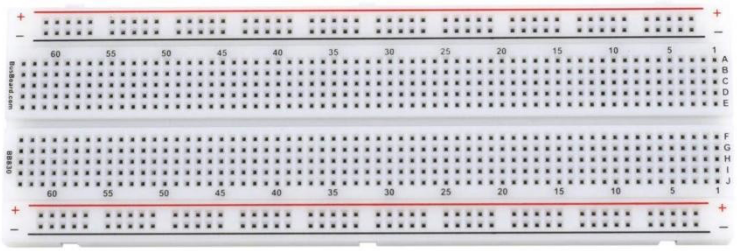


Figure 11 Breadboard

- **Buzzer**: A buzzer is in the mechanical form of a small rectangular or cylindrical housing, with electrical connection for direct mounting on rigid printed circuit, or with electrical connection consisting of flexible electrical son. The buzzer has two small brackets. (k, 2012)

Figure 12 Buzzer

- **Jumper Wires**: Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in other to make it easy to change circuit as needed. (Hemmings, 2018)

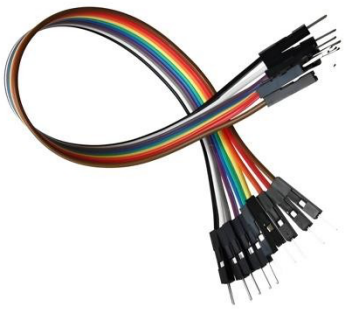


Figure 13 Jumper Wires

- **Micro-USB:** A micro-USB is a miniaturized version of the Universal Serial Bus interface developed for connecting compact and mobile devices, such as smartphones, MP3 players, Global Positioning System devices, printers and digital cameras. (Hanna, 2022)

Figure 14 Micro-USB cable

2.3.2 Software Components

- **Arduino IDE:** Arduino IDE is an open-source software which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It also supports the programming languages C and C++. IDE stands for Integrated Development Environment. (Ardubots, 2024)

Figure 15 Arduino IDE

- **Fritzing:** Fritzing is an electronic design automation program, designed to help designers and artists transition from prototypes to final products. Fritzing was created based on the principles of Processing an Arduino, allowing designers, artists and researchers and hobbyists to register their Arduino-based prototypes and create printed circuit diagrams for further fabrication. (Perera, 2021)

Figure 16 Fritzing

3. Development

The development part of the report shows the step-by-step process of the development of the project. This part of the project helps to ensure that the project meets its objectives. Following is the overview of the development process:

3.1 Planning and Design

The project's main aim is to create a flood detection device based on IoT and to alert people with the help of the device before flooding to reduce the damage caused by it. The idea of this project was to make such a device where when the level of water reaches certain points it starts the buzzer which gives alert to the people before flood occurs.

To make it more effective we used LCD where it displays the rising level of water. Device will save or reduce many damages which can be caused by flood. Main idea behind the project was same, to minimize the damage which can be caused in different part of the world by flood. For the completion of this process devices such as NodeMCU which helps to process the data, raindrop sensor, a 16x2LCD to display and a buzzer for alerts are used. All these are connected using breadboard and jumper wires to each other.

Our main aim was to create an affordable solution where it warns the people in high flood area. We used different devices to make a proper working flood detection device. All those devices help to finalize our main device which works reliably and meets the criteria to detect the flood.
Key Features:

- LCD for Water Levels
- Alerts using Buzzer
- User-friendly components
- Scalable and low cost

3.2 Resource Collection

Different devices were used to complete our project. Every device was chosen carefully by seeing their role in the project also should be matched with our design. Almost every device was managed by the resource department of Islington College itself but for some devices we just bought them. All the members of the group made equal contributions to collect the resources which we need in our project. List of devices which we found from resource department at Islington College are:

College Resources:

- NodeMCU for processing and communication
- 16x2 LCD to display messages
- Breadboard to create a base for components
- Buzzer to generate sound as alert system
- USB cables and jumper wires to provide power and connectivity.

External Purchases:

- Raindrop sensor which detects water level.

Selection of proper resources is a crucial step to build any device or to make any device more effective.

3.3 System Development

To make the flood detection device functional, it is necessary to assemble the components in proper steps. Different code is also used to make this device which alerts the damage caused by flood. All the process is carried out in multiple phases which help to ensure its functionality and reliability of the device.

3.3.1 Phase 1: Initial Setup

The microcontroller (ESP8266) is connected to the computer device with the help of USB cable to:

- Provide power supply to the microcontroller.
- Used to upload the initial code which is the backbone of the system functionality.

Figure 17 Power Supply from PC to Nodemcu

3.3.2 Phase 2: Component Assembly

I. Power Distribution:

By the help of ESP8266, power is provided in breadboard then it provides power to every device by the help the wires which are connected in breadboard. Breadboard helps in this project by making connections between different devices using various wires. The breadboard acts like a central hub which distributes power to other components using jumper wires.

- The microcontroller provides power to the breadboard.
- Breadboard distributes the power to other components which acts like a central hub.
- Jumper wires are used to distribute the power.

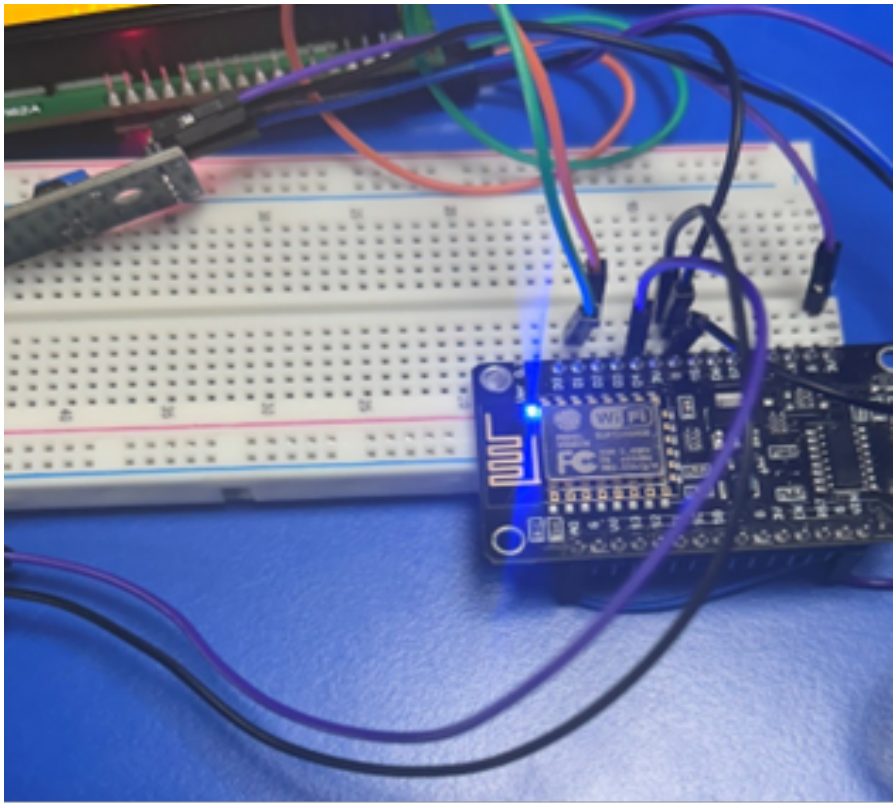


Figure 18 Breadboard connected with ESP8266

II. Buzzer Integration:

When power is provided to breadboard by wires, we connected buzzer to the breadboard which send alert sound before any damage caused by flood. The buzzer was connected to breadboard where NodeMCU is used to provide power supply. NodeMCU also helps to provide input and output results.

- Buzzer is connected to the breadboard.
- Buzzer is controlled by NodeMCU to send sound alerts when water levels crossed the sensor.

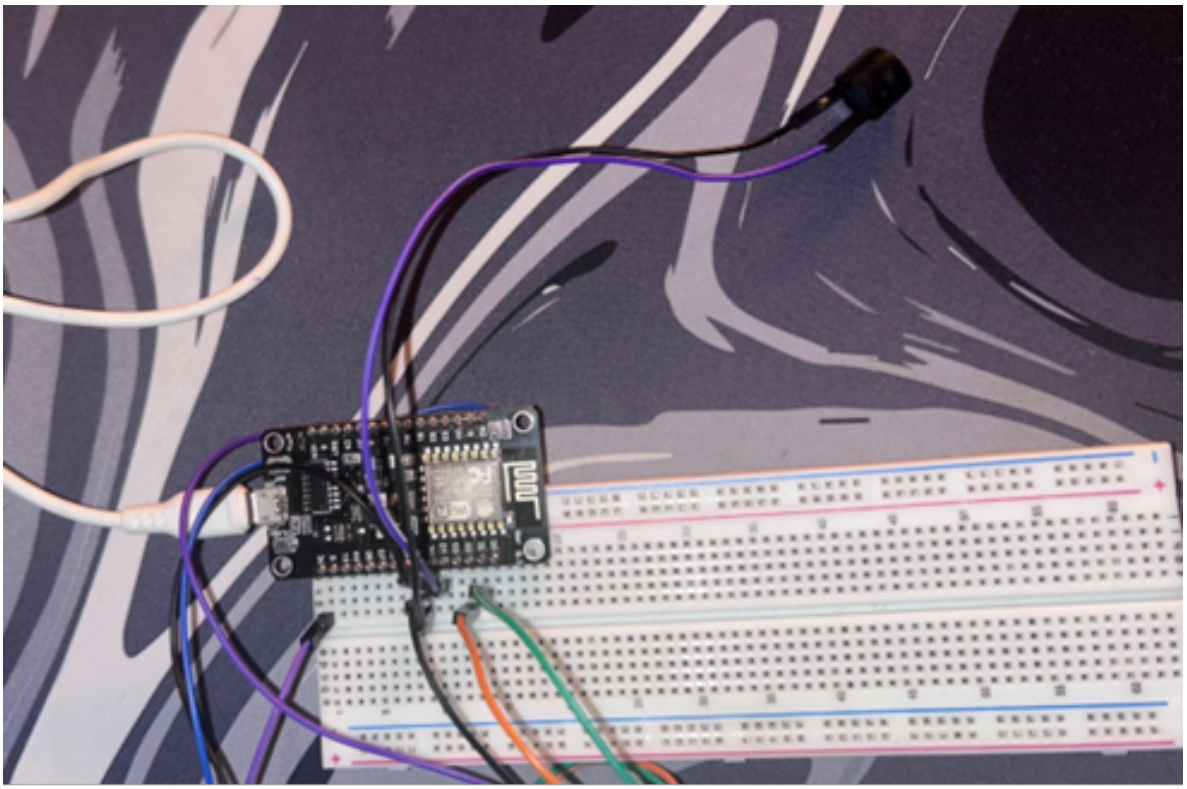


Figure 19 Buzzer connected to the breadboard

III. Raindrop Sensor Connection:

After the connection of buzzer, we connected raindrop sensor which senses the level of water. It is also connected with breadboard for supplying power. Jumper wires are used to provide connection between them. It ensures the level of water so that it helps to know the amount of water which can cause floods.

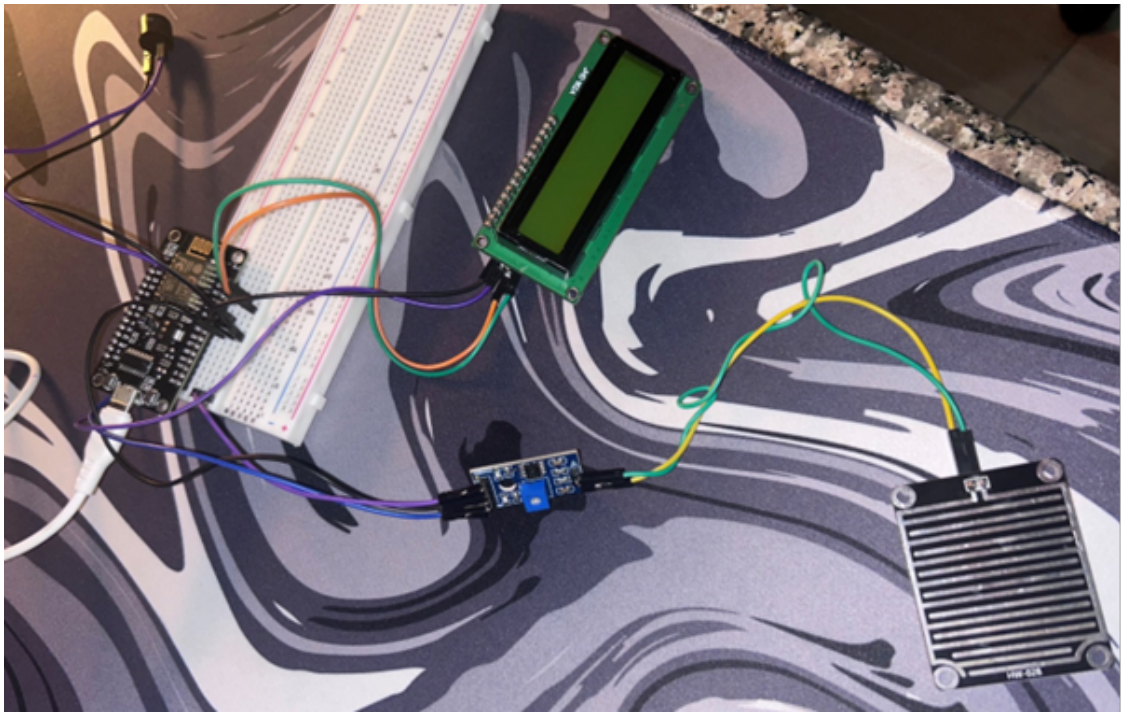


Figure 20 Connecting raindrop sensor to the breadboard

IV. LCD Installation:

Then LCD is connected into breadboard and microcontroller (NodeMCU) using jumper wires. Breadboard is used to connect multiple components which also act like a central hub. NodeMCU shows direct connection with LCD which helps to send commands and data to the LCD.

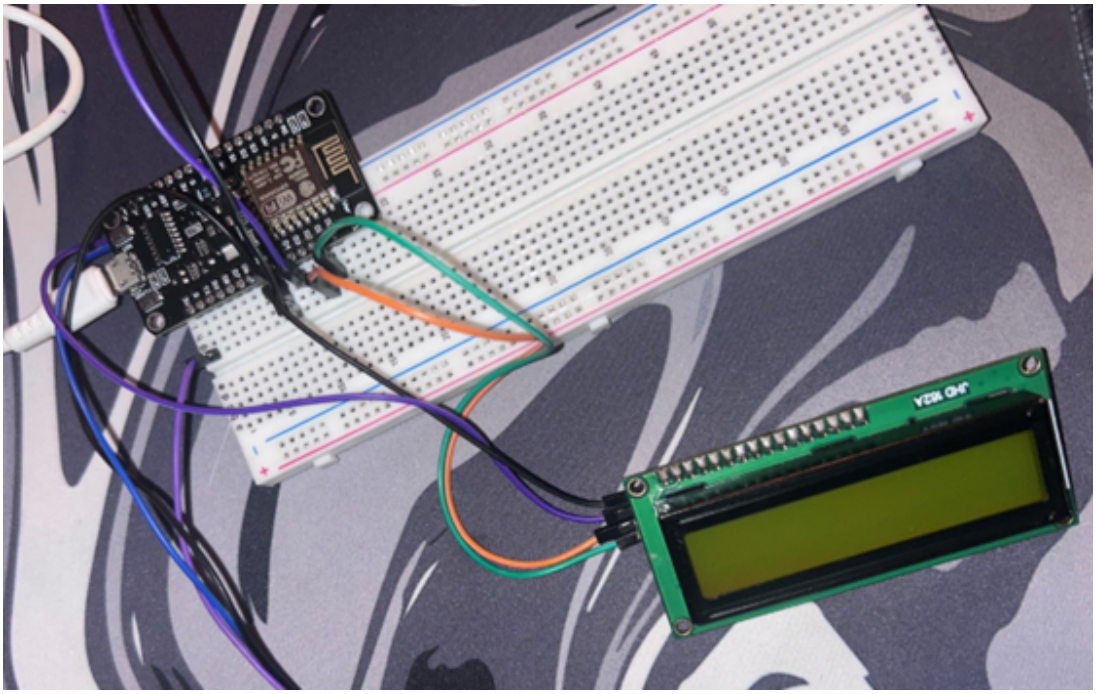


Figure 21 Connecting LCD to the breadboard

3.3.3 Phase 3: Code Implementation

The code is then run through Arduino IDE after,

- selecting the specific port the USB is connected to, and,
- installing the necessary libraries.

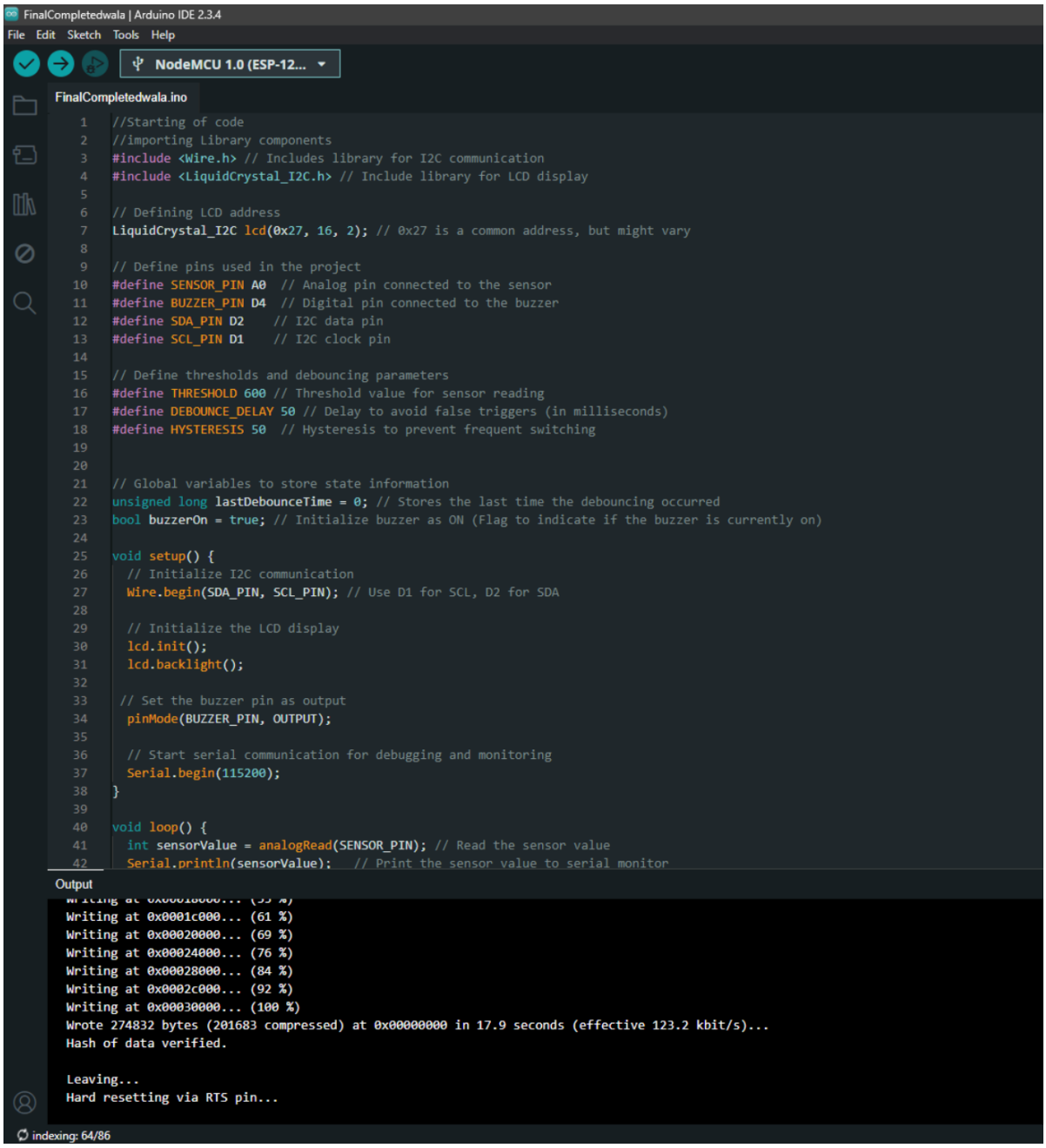


Figure 22 Running code on Arduino IDE-1

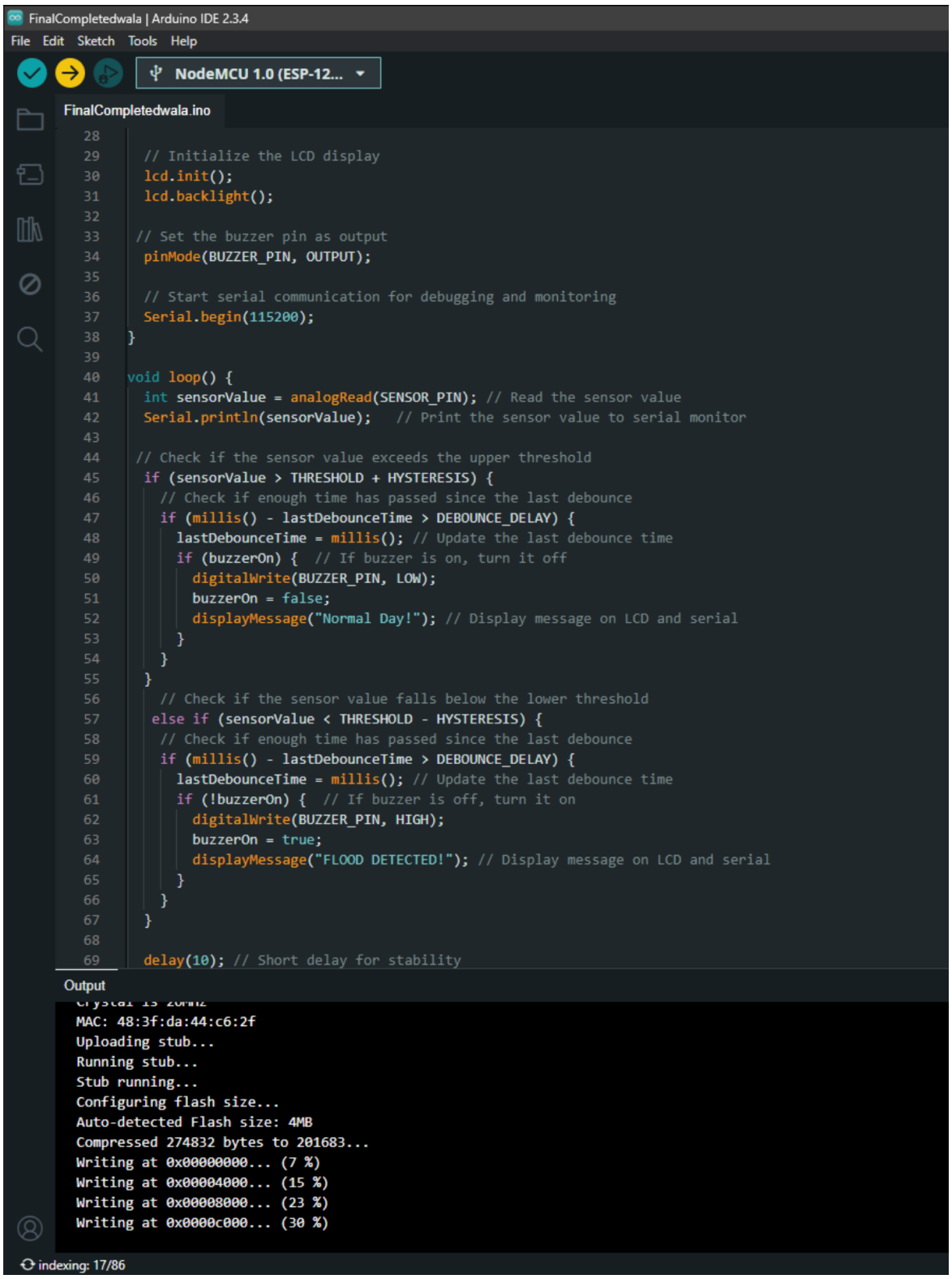
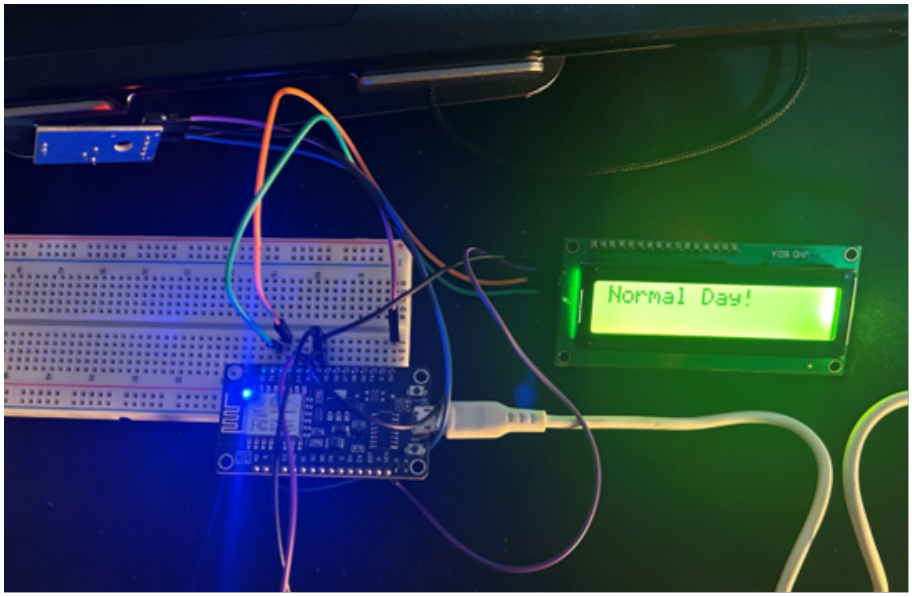


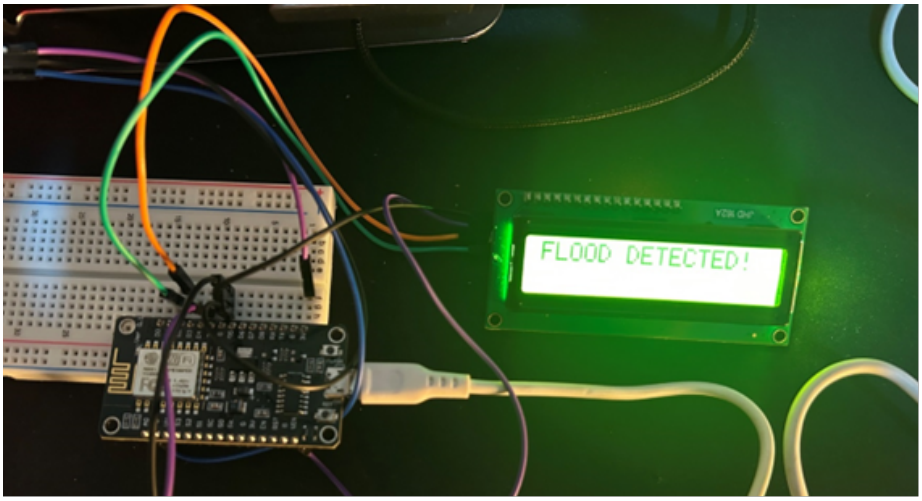
Figure 23 Running code on Arduino IDE-2

3.3.4 Phase 4: Functional Testing

LCD helps us to show alert message whenever the level of water reaches a certain point. For normal water levels, the system will display “Normal Day!” to indicate safe conditions. In this situation, buzzer will not produce sound as an alert.



Whenever the level of water reaches a higher point, with the help of raindrop sensor and according to the code, it shows “FLOOD DETECTED!” and simultaneously the buzzer rings to alert of the incoming flood.



4. Result and Findings

After the completion of development phase, the functions of the device are checked which includes the script to run successfully as well as the device to be working as intended. The flood detection system provides an alert around the area when it detects water building up.

After the script is ran successfully, the raindrop sensor module, when water is detected, changes its sensor value exceeding normal threshold, which then in return, tips the buzzer to start playing sound and LCD displays “FLOOD DETECTED!”.

Testing:

4.1 Test 1: To run the script

Test	1
Objective	To run the script.
Activity	After connecting the microcontroller to PC, the code was compiled.
Expected Result	The script would run without errors.
Actual Result	The script is compiled successfully.
Conclusion	The test was successful.

Table 2 Testing the execution of code

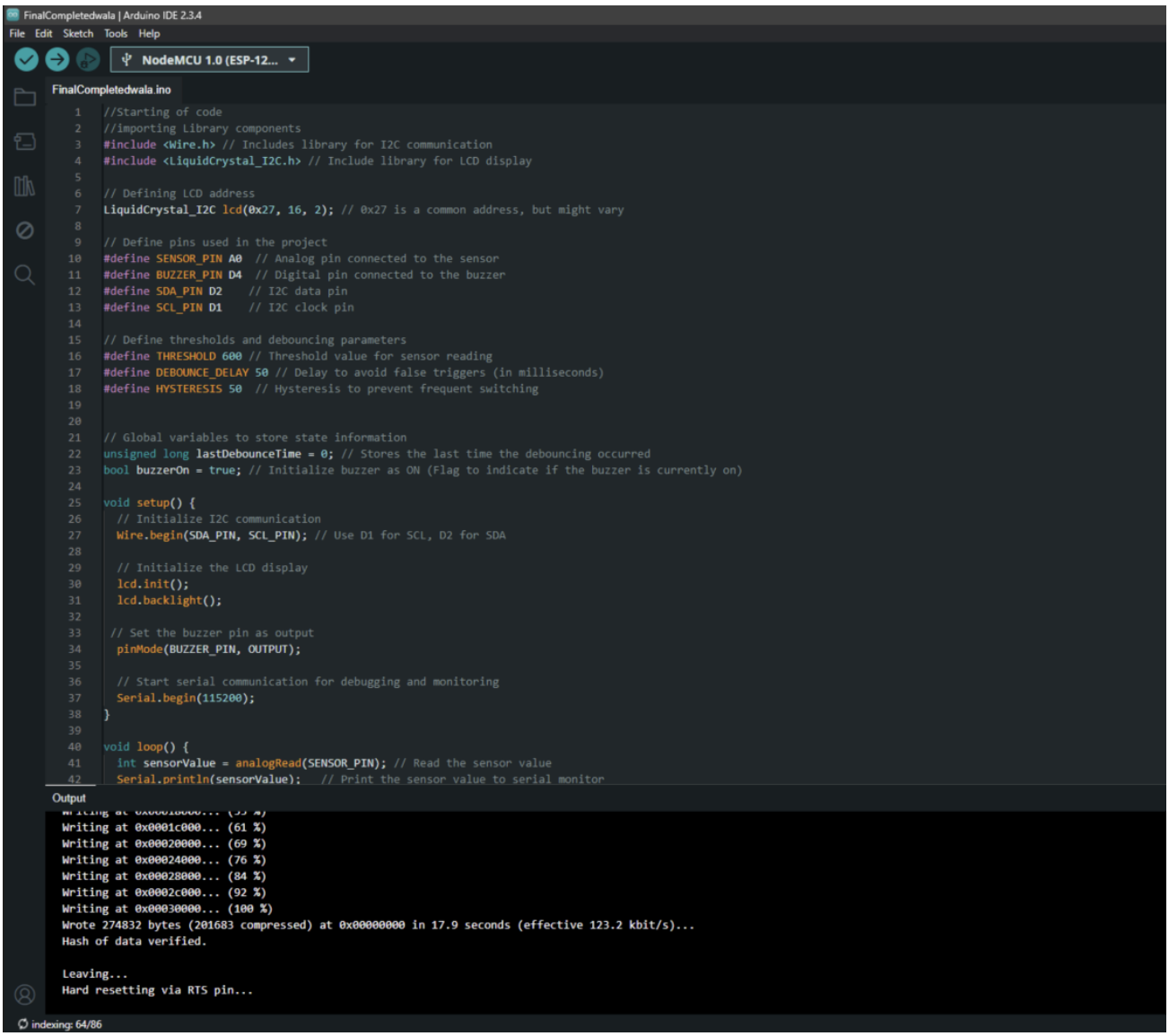


Figure 24 Test-1 Code compiles without error

4.2 Test 2: To check if the raindrop sensor detects water

Test	2
Objective	To check if the raindrop sensor detects water.
Activity	The raindrop sensor was put on a water reserve.
Expected Result	The raindrop sensor would detect water.
Actual Result	The raindrop sensor detects water.
Conclusion	The test was successful.

Table 3 Testing if raindrop sensor detects water

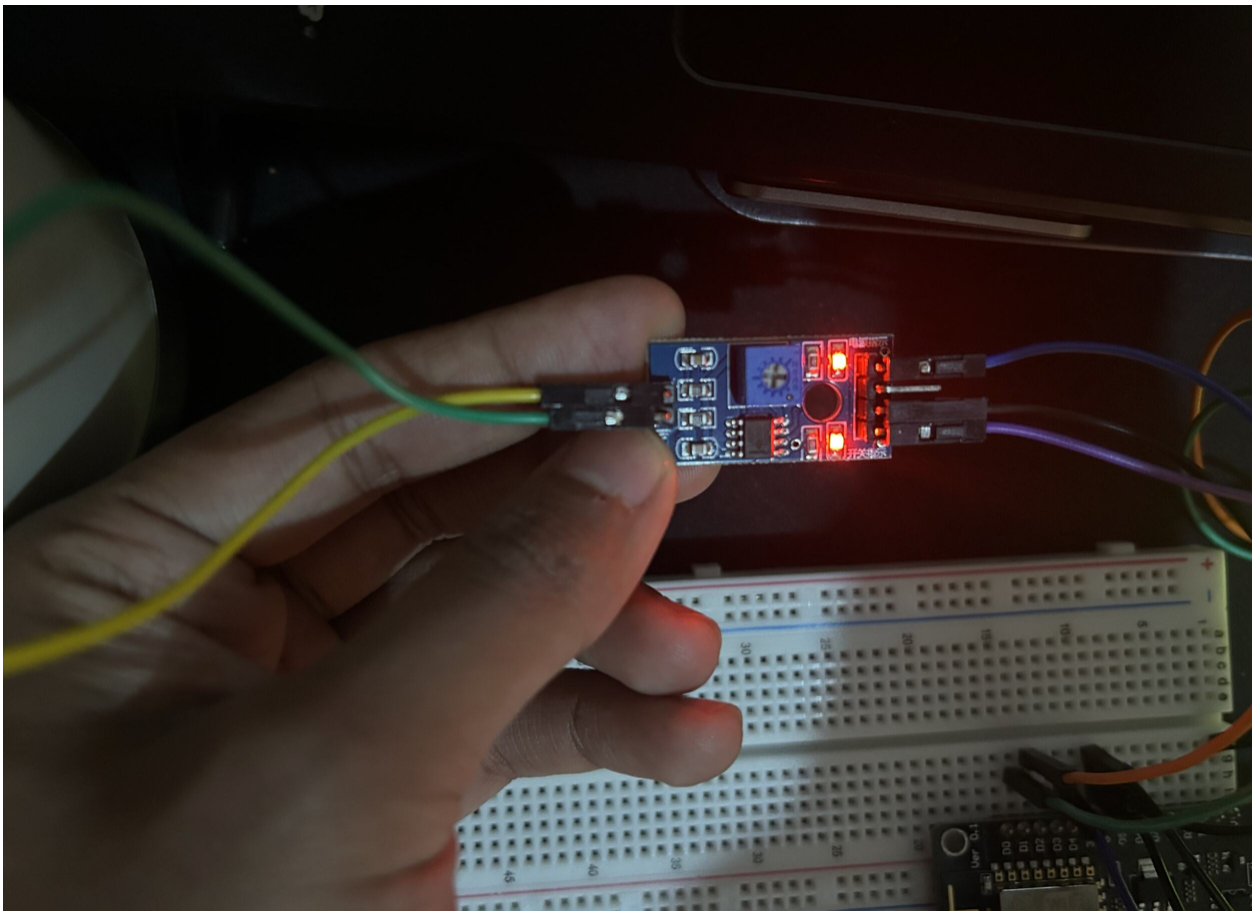


Figure 25 Raindrop sensor active

4.3 Test 3: To check if the piezo makes sound when necessary

Test	3
Objective	The check if the piezo makes sound when necessary.
Activity	The raindrop sensor was put on a water reserve.
Expected Result	The piezo would buzz.
Actual Result	The piezo did not buzzes.
Conclusion	The test was unsuccessful.

Table 4 Testing if the piezo buzzes

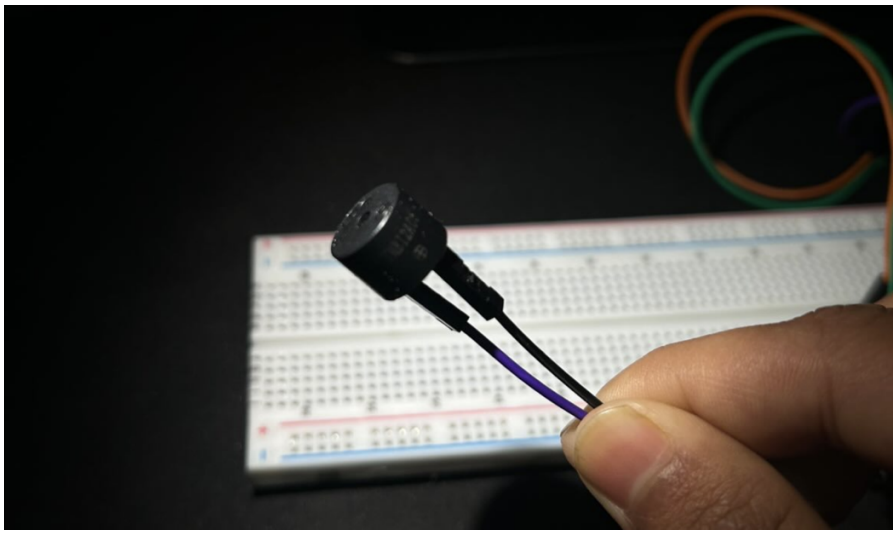


Figure 26 Wire ends on wrong hole (Problem)

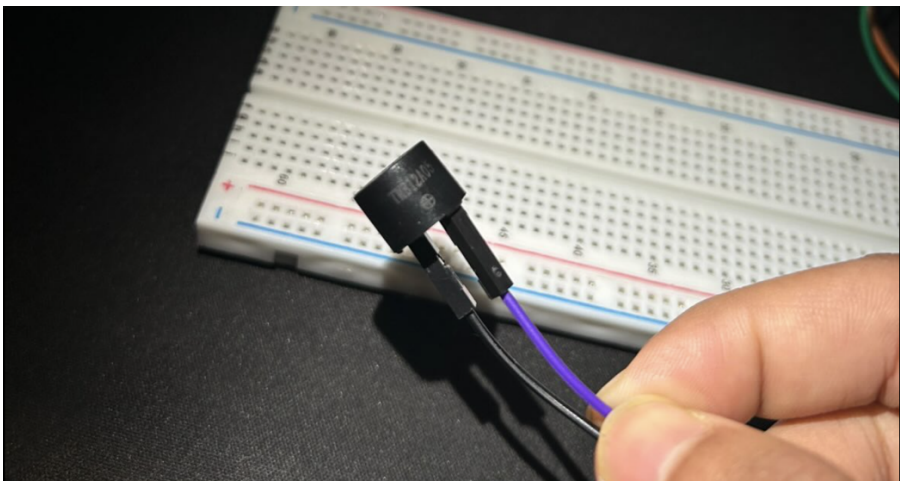


Figure 27 Wire ends on right holes (Solution)

4.4 Test 4: To check if the LCD displays specific message when water is not detected

Test	4
Objective	To check if the LCD displays specific message when water is not detected.
Activity	The code was compiled.
Expected Result	The LCD would display "Normal Day!".
Actual Result	The LCD displayed "Normal Day!".
Conclusion	The test was successful.

Table 5 Testing if the LCD displays the normal message

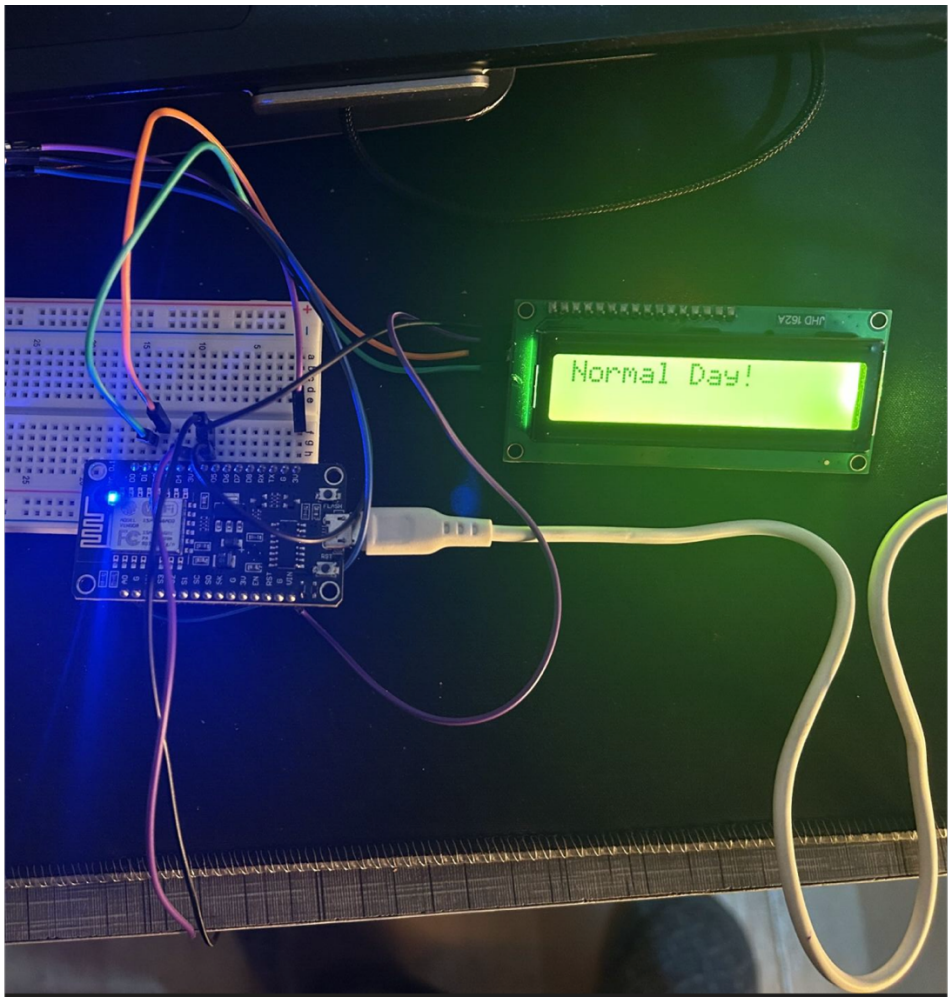


Figure 28 LCD displays "Normal Day!"

4.5 Test 5: To check if the LCD displays specific message when water is detected

Test	5
Objective	To check if the LCD displays specific message when water is detected.
Activity	The raindrop sensor was put on a water reserve.
Expected Result	The LCD would display "FLOOD DETECTED!".
Actual Result	The LCD displayed "FLOOD DETECTED!".
Conclusion	The test was successful.

Table 6 Testing if the LCD displays the alert message

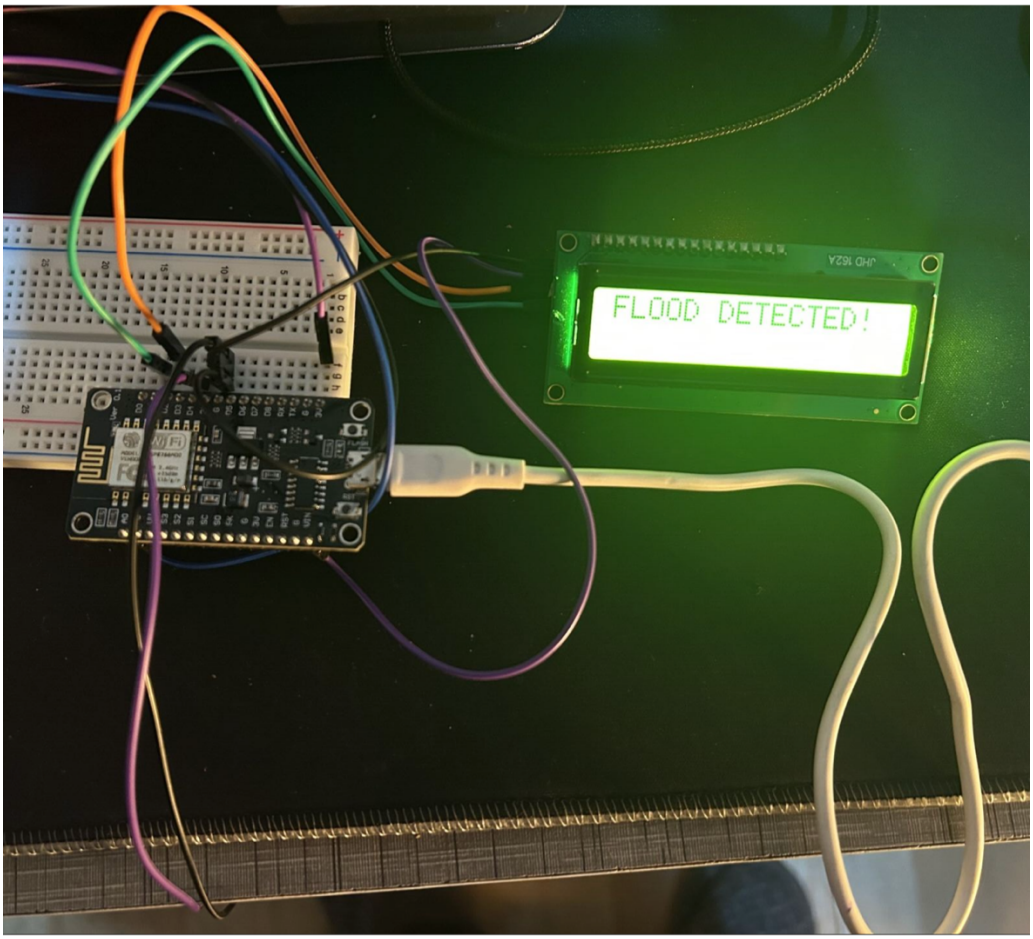


Figure 29 LCD display "FLOOD DETECTED!"

5. Future Works

In future development, we plan to conduct tests in riverbanks to determine the level of water when it is flooded. Designing a flood detector includes upgraded rain-drop sensors for more accurate water level detection which detects and aware with a message according to the percentage of the water. For the accurate indication and monitoring the sensor displays an alert, such as "10 % filled,”,"40% filled," and 80% filled,". This simple design could be an example of real-world scenarios and could be utilized for pattern recognition and predictive analysis.

This design aims for valuable collaboration with the local community which would provide a deep understanding of environmental challenges. The level of water is divided into three categories: safe, alert and emergency.

The design structure is very crucial as it will assist the ones who live near the water bodies. People become more cautious of the uncertain flood because of this design structure. Therefore, the design plays a very crucial role for many decision makers during the flood.

6. Conclusion:

The flood monitoring system effectively provides real-time water level monitoring and timely alerts, demonstrating its potential to minimize the impact of flooding.

The system's combination of advanced sensor technology, reliable data processing, and immediate alert mechanisms offers a comprehensive solution to a critical problem faced by many accurately and provides consistent performance across varied conditions. The System's modular design makes it adaptable for future improvement, such as incorporating predictive analytics using weather data or transitioning to sustainable, solar-powered configuration for deployment in remote areas. The effectiveness of the system will be enhanced to facilitate data reporting from various sensor monitoring units at the same time to a server or cloud storage through a central network gateway.

Consequently, we seek to facilitate the use of IOT technology across various network connections. Based on the water level risk, the system will activate early flood alerts for authorities and residents to implement preventative measures. Future enhancement will focus on refining this aspect to further strengthen the system's utility and resilience. The project demonstrates a significant step toward leveraging technology for disaster management. By minimizing the devasting impact of floods through early detection and timely alerts, the flood monitoring system contributes to safer, more resilient communities.

7. Bibliography

ORACLE. (2024). *What is iot?* Retrieved from oracle: https://www.oracle.com/internet-of-things/

Unicef. (2024, October 01). *Unicef*. Retrieved from Flooding in Nepal: https://www.unicef.org/nepal/press-releases/35-children-die-nepal-after-devastating-rainfall-and-landslides-unicef

robocraze. (n.d.). *What is a Raindrop Sensor?* Retrieved from robocraze: https://robocraze.com/blogs/post/what-is-a-raindrop-sensor?srsltid=AfmBOoom6UkC4uGHJPy9ObWCQrd7UoY176Xth0Apx1Qx9UXeCuBNXIs3

Team, n. m. (2018). Retrieved from nodemcu: https://www.nodemcu.com/index_en.html

Ardubots. (2024, May 5). *What is Arduino IDE? A Deep Dive into the Arduino IDE*. Retrieved from ardubots: https://ardubots.com/what-is-arduino-ide-a-deep-dive-into-the-arduino-ide/

Perera, A. (2021, February 20). *Fritzing, all you need to know*. Retrieved from automatismosmundo: https://automatismosmundo.com/en/fritzing-all-you-need-to-know/

k, v. (2012, Aug 14). *Buzzers and it's function* . Retrieved from India Study Channel: https://www.indiastudychannel.com/resources/154564-Buzzers-it-s-function.aspx

Hemmings, M. (2018, Jan 30). *Sparkfun Education*. Retrieved from What is a Jumper Wire?: https://blog.sparkfuneducation.com/what-is-jumper-wire

Wagner, C. (2024, Sep 27). *What is a Breadboard?* Retrieved from CircuitBread: https://www.circuitbread.com/ee-faq/what-is-a-breadboard

CD-Team. (2023, August 25). *LCD 16×2 Pinout, Commands, and Displaying Custom Characters* . Retrieved from electronicsforu: https://www.electronicsforu.com/technology-trends/learn-electronics/16x2-lcd-pinout-diagram

Teja, R. (2024, April 1). *Getting Started with NodeMCU | A Beginner's Guide*. Retrieved from ElectronicsHub: https://www.electronicshub.org/getting-started-with-nodemcu/

Hanna, K. T. (2022, June). *micro USB*. Retrieved from techtarget: https://www.techtarget.com/whatis/definition/micro-USB

8. Appendix:

8.1 Source Code:

```
// Starting of code
// Importing required libraries for I2C communication and LCD display
#include <Wire.h>              // Library for I2C communication (used by LCD)
#include <LiquidCrystal_I2C.h> // Library for controlling the LCD display

// Defining the LCD address and size (16 columns, 2 rows)
LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD address: 0x27, 16x2 display

// Defining pins for the sensor and buzzer
#define SENSOR_PIN A0   // Analog pin connected to the sensor
#define BUZZER_PIN D4   // Digital pin connected to the buzzer
#define SDA_PIN D2      // I2C data pin for LCD
#define SCL_PIN D1      // I2C clock pin for LCD

// Defining a threshold value to detect water presence
#define THRESHOLD 600   // Sensor value below this indicates water presence

// Variable to keep track of whether the buzzer is on or off
bool buzzerOn = false; // Initially, the buzzer is off

// Setup function runs once when the system starts
void setup() {
  // Initialize I2C communication for the LCD using the defined pins
  Wire.begin(SDA_PIN, SCL_PIN);

  // Initialize the LCD display
  lcd.init();       // Start the LCD
```

```
  lcd.backlight();   // Turn on the LCD backlight

  // Set the buzzer pin as an output so it can be turned on or off
  pinMode(BUZZER_PIN, OUTPUT);

  // Start serial communication to print values for debugging
  Serial.begin(115200); // Set baud rate to 115200 (speed of communication)

  // Display "Normal Day!" at startup on the LCD
  displayMessage("Normal Day!"); // Shows "Normal Day!" initially
}

// Loop function runs continuously after setup
void loop() {
  // Step 1: Read the sensor value (measures water level)
  int sensorValue = analogRead(SENSOR_PIN); // Get the value from the sensor
  Serial.println(sensorValue);              // Print the sensor value to the serial monitor

  // Step 2: Check if the sensor value is below the threshold (indicates water is present)
  if (sensorValue < THRESHOLD) { // Water detected (low sensor value)
    if (!buzzerOn) {            // If the buzzer is off, turn it on
      digitalWrite(BUZZER_PIN, HIGH); // Turn on the buzzer
      buzzerOn = true;              // Update buzzer status to "on"
      displayMessage("FLOOD DETECTED!"); // Show warning message on LCD and serial
    }
  }
  // Step 3: Check if the sensor value is above the threshold (no water detected)
  else { // No water detected (high sensor value)
    if (buzzerOn) {             // If the buzzer is on, turn it off
      digitalWrite(BUZZER_PIN, LOW);  // Turn off the buzzer
```

```
      buzzerOn = false;              // Update buzzer status to "off"
      displayMessage("Normal Day!");  // Show normal message on LCD and serial
    }
  }

  // Step 4: Add a short delay for system stability
  delay(10); // Delay of 10 milliseconds to prevent too frequent updates
}

// Function to display a message on the LCD and print it to the serial monitor
void displayMessage(const char* message) {
  lcd.clear();         // Clear any previous message on the LCD
  lcd.setCursor(0, 0); // Set the cursor to the first line, first column
  lcd.print(message);  // Display the given message on the LCD
  Serial.println(message); // Also print the message to the serial monitor
}
// End of Code
```

8.2 Individual Contribution Plan

The individual contribution plan of each member of the group to complete the project are mentioned below:

Name	Task	Contribution
Sujal Parajuli	**Report:** Introduction, Development, Design diagram **Practical:** Setting up components with breadboard, Debugging, Code Implementation, Testing	**22%**
Shrine Ghimire	**Report:** Individual Contribution Plan, Background (Requirement Analysis, Design diagram), Results and Finding **Practical:** Code Implementation, Testing	**18%**
Sarthak Baniya	**Report:** Background (System Overview), Future Works **Practical:** Modeling Spaces, Jump Wires Connection	**15%**
Sakshyam Kafle	**Report:** Acknowledgement, Abstract, Design diagram **Practical:** Research, Development	**15%**
Niran Bhatta	**Report:** Aim and Objectives, Appendix **Practical:** Device Monitoring, Debugging	**15%**
Paras Kumar Yadav	**Report:** Background (System Overview), Conclusion **Practical:** Device Monitoring	**15%**

Table 7 Individual Contribution Plan Table

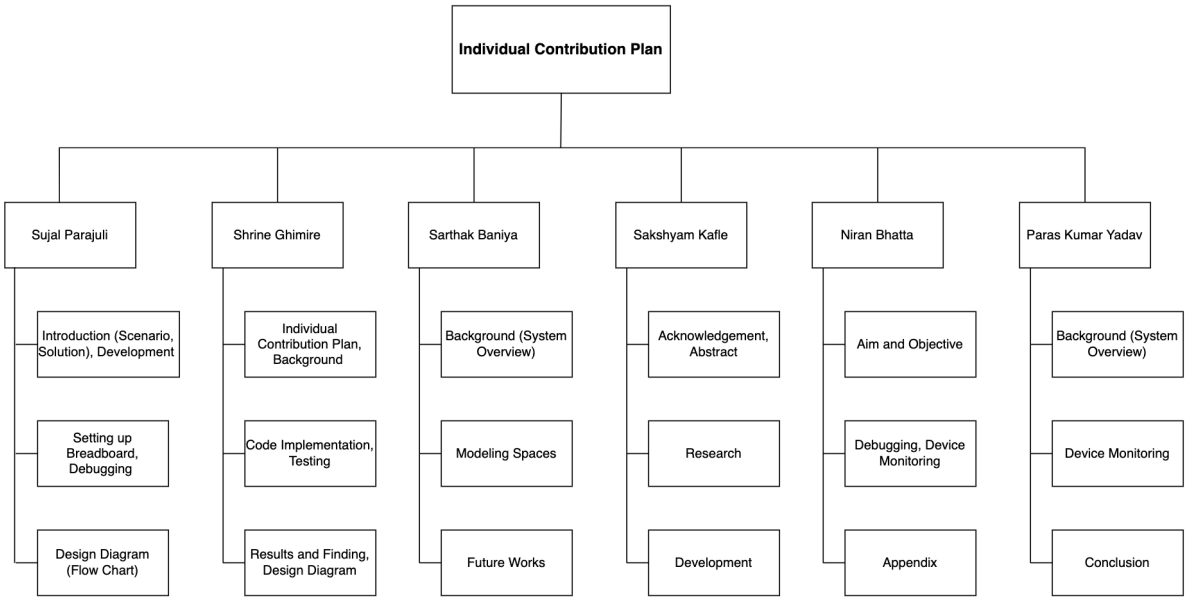


Figure 30 Individual Contribution Plan

8.3 Work Breakdown Structure Chart:

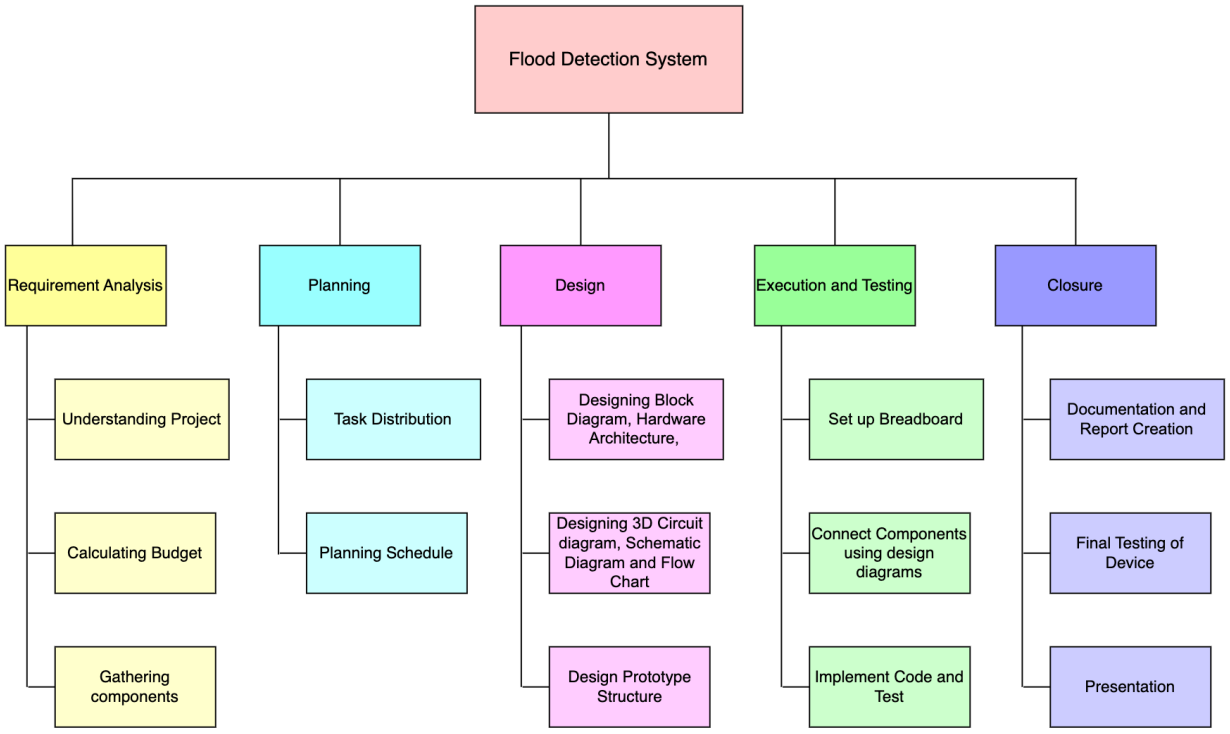


Figure 31 Work Breakdown Structure