# Applied Industrial of Things

# Project Statement 14

# Flood Monitoring System

# Batch No:-47

**Department:** Electronic & Communication Engineering.

**College:** Sri Venkateswara College of Engineering and Technology

## Team:-

S.Lakshmi NarsimhaReddy-21781A04G8

Manish Kumar ray 21781A04M2

Pariyanasu Kumar Sharma-21781A04M4

Ranvir Kumar 21781A04M8

# Flood Monitoring System

**Introduction:**

Flood monitoring is a crucial aspect of managing and mitigating the impacts of flooding, a natural disaster that poses significant threats to communities and ecosystems. As climate change and urbanization continue to influence weather patterns and landscape dynamics, the frequency and severity of floods are on the rise, necessitating advanced monitoring systems to enhance preparedness and response efforts.

Flood monitoring involves the systematic observation and analysis of various environmental factors, such as rainfall, river water levels, soil moisture, and weather conditions, to predict and detect potential flood events. The goal is to provide timely and accurate information to authorities, emergency responders, and the public, enabling them to make informed decisions and take appropriate actions to minimize the impact of floods on lives and property.

This introduction will explore the key components of flood monitoring, including the types of technologies and methods employed, the benefits of monitoring systems, and the ongoing challenges in implementing effective flood monitoring programs. Additionally, it will highlight the importance of real-time data, remote sensing technologies, and community engagement in creating a comprehensive and responsive flood monitoring framework. As societies strive to adapt to the changing climate and its associated risks, flood monitoring remains a critical tool in building resilience and safeguarding communities against the devastating consequences of flooding.

**Problem Statement:**

Floods are a recurring natural disaster causing significant damage to lives and property. To mitigate the impact of floods, there's a need for an IoT Flood Monitoring System that provides real-time water level monitoring and timely alerts in flood-prone areas. The system should prioritize accuracy, low-cost implementation, and ease of deployment. Data source:

The source of data IoT Flood Monitoring System (openai.com)

**Scope of the solution:**

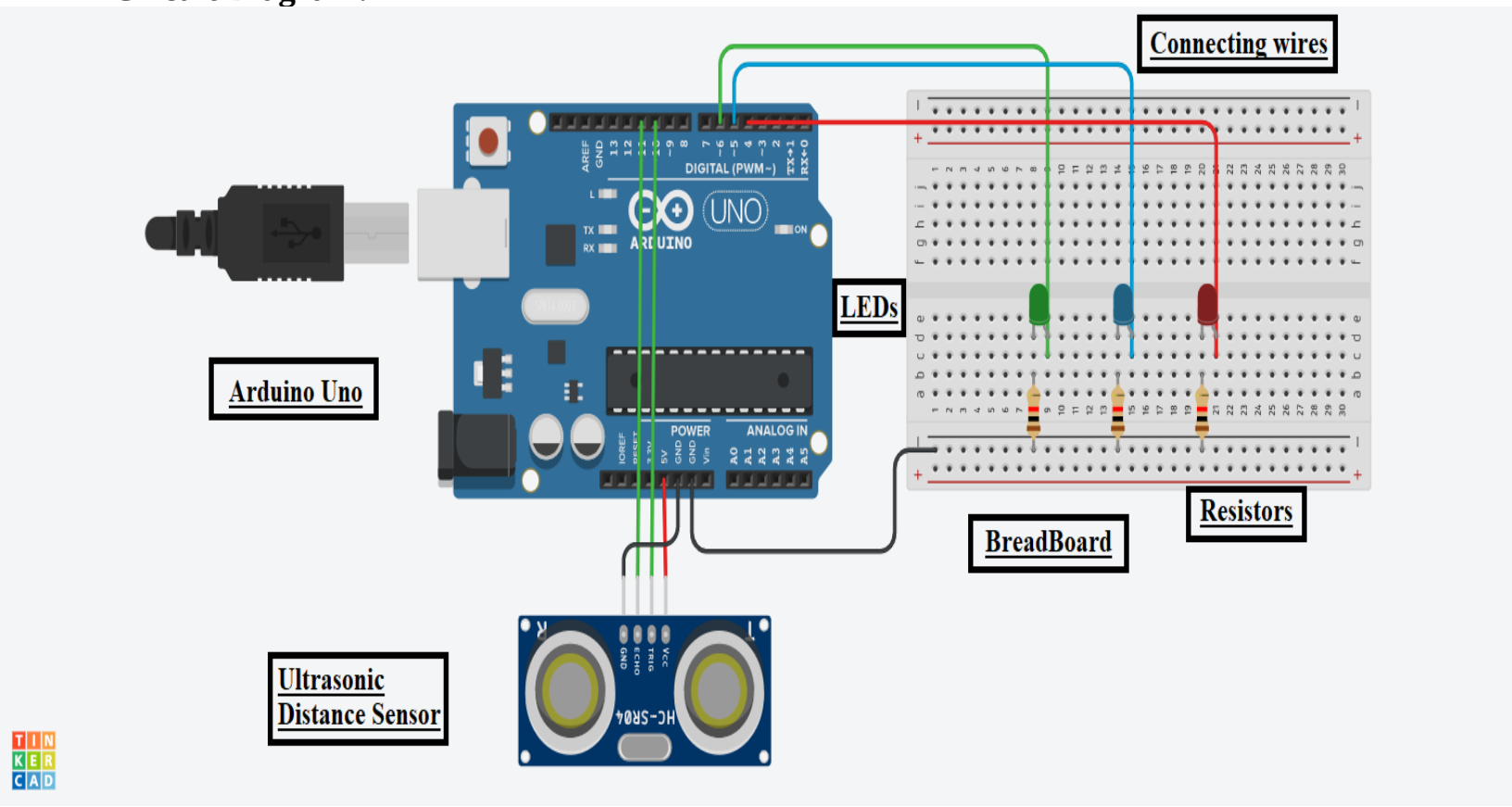
The IoT Flood Monitoring System will involve deploying water level sensors in flood-prone areas. These sensors will be connected to a central Raspberry Pi Pico, which will continuously monitor water levels. The system will send alerts when water levels exceed a predefined threshold, ensuring timely evacuation and response.

## ➢ Project: Flood Monitoring System

## Required Components to develop Solutions

1. Arduino Uno R3
2. 3 LEDs (of different colours)
3. 3 Resistors
4. 1 Small Breadboard
5. Ultrasonic Distance Sensor
6. Connecting Wires

**Circuit Diagram:**




The source of the data :- Tinkercad

.

## ➢ Steps To Build **Flood Monitoring System** With LEDs

**Step 1:** Arrange all the components on the Digital Board or your workspace Table.

***Ultrasonic Distance Sensor:***

**Step 2:** Connect the Power terminal of the Ultrasonic Distance Sensor to the 5V pin of the Arduino

**Step 3:** Connect the Trigger Terminal of the Ultrasonic Distance Sensor to the 10-number pin of the Arduino.

**Step 4:** Connect the Echo Terminal of the Ultrasonic Distance Sensor to the 11-number pin of the Arduino

**Step 5:** Connect the Ground Terminal of the Ultrasonic Distance Sensor to the GND pin of the

***LEDs***

**Step 6:** Connect the Cathode terminal of the LED to the GND pin of the Arduino through a resistor.

**Step 7:** Connect the Anode terminal of the LED to the 4, 5, & 6 number pins of the Arduino.

**Source Code:-**

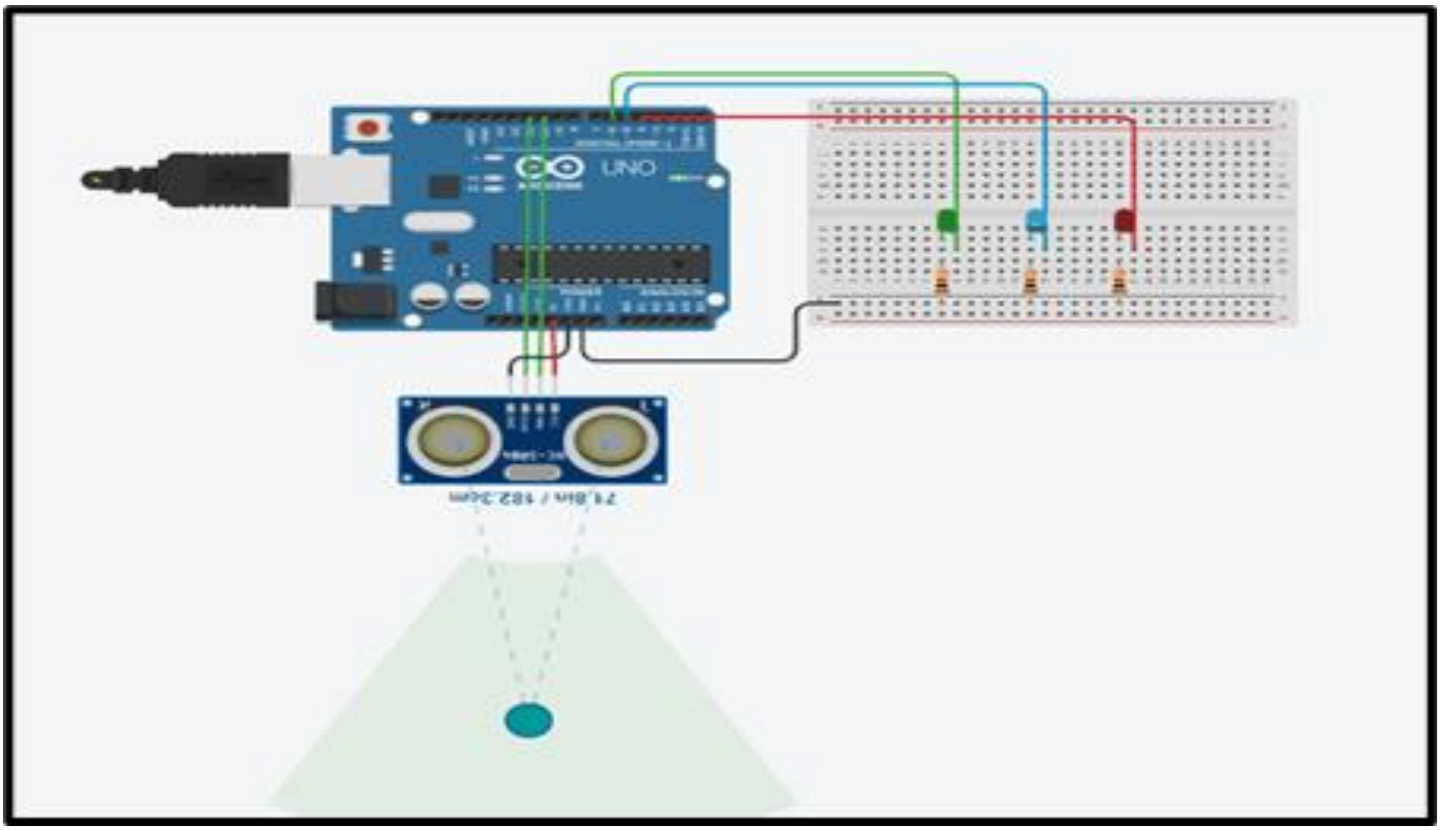
```
int distance = 0;
int red = 4;
int blue = 5;
int green = 6;
long readUltrasonicDistance(int triggerPin, int echoPin)
{
pinMode(triggerPin, OUTPUT);
digitalWrite(triggerPin, LOW);
 delayMicroseconds(2);


digitalWrite(triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(triggerPin, LOW);
```

```
 pinMode(echoPin, INPUT);



 return pulseIn(echoPin, HIGH);
 }
 void setup()
{
 pinMode(4, OUTPUT);
 pinMode(5, OUTPUT);
 pinMode(6, OUTPUT);
  }
 void loop()
{
  distance = 0.01723 * readUltrasonicDistance(10, 11);
 if(distance >= 200)
  {
 digitalWrite(red, HIGH);
 digitalWrite(blue, LOW);
 digitalWrite(green, LOW);
 delay(2000);
     }
  if(distance > 100 && distance < 200)
  {
     digitalWrite(red, LOW);
   digitalWrite(blue, HIGH);
  digitalWrite(green, LOW);
   delay(2000);
     }
      if(distance <= 100)
```

```
  {
   digitalWrite(red, LOW);
   digitalWrite(blue, LOW);
digitalWrite(green, HIGH);
  delay(2000);
     }
  }
```

**Source of code [Arduino Water Level Indicator With LEDs - My Project Ideas]**

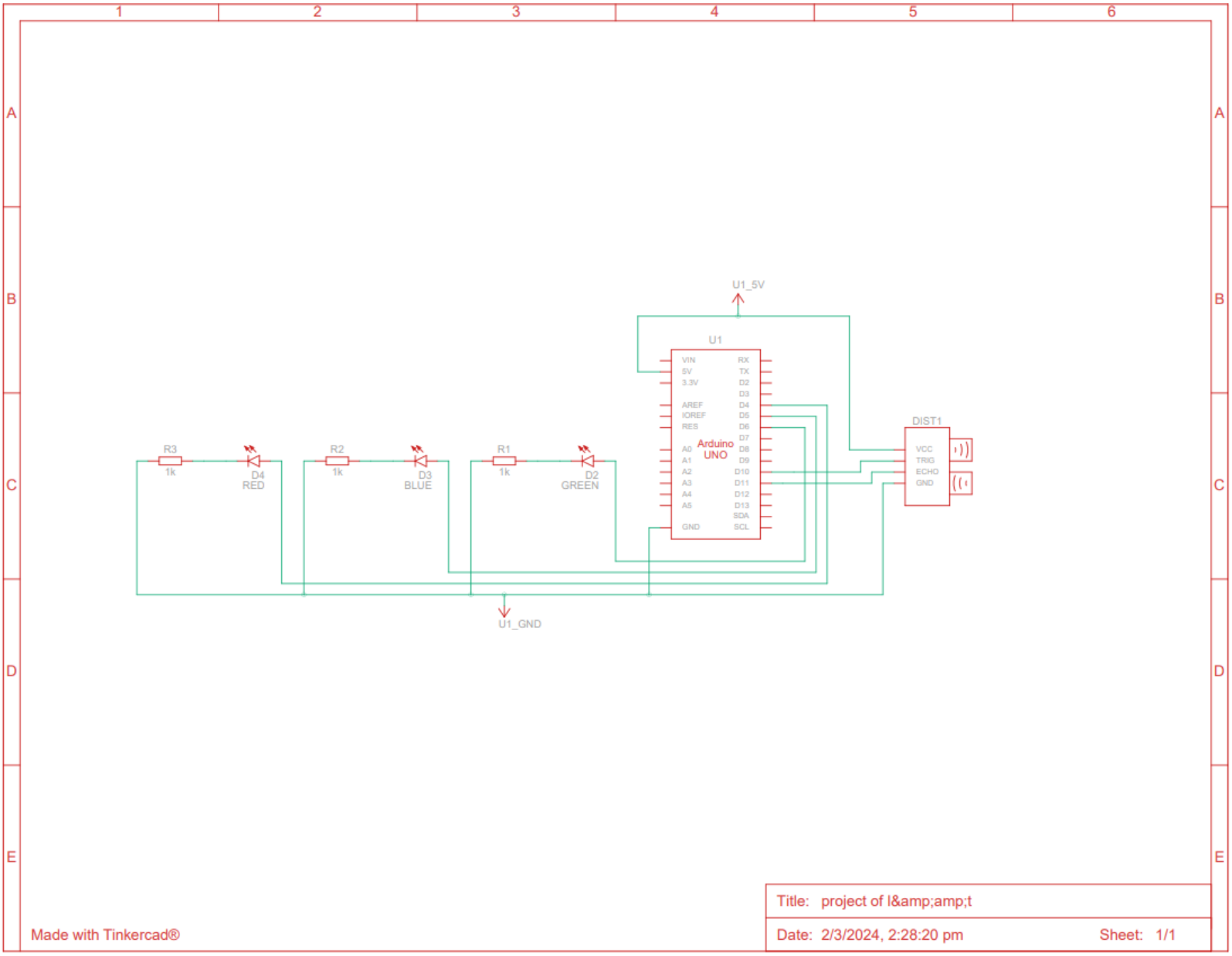
**Output:-**



**Simulated Circuit (TinkerCad):-**

**Link :- https://www.tinkercad.com/things/3rwet1ZW9G4-project-of-lampt**

## Gerber file:-




Github Link:- [Manish7461 (github.com)]

# Thank you