# L&T Project

**Problem Statement:**

Develop a prototype for a dam water level sensing and alert system. The system should be able to accurately measure water levels in a dam and provide timely alerts in case of abnormal increases in water levels, indicating a potential risk of flooding

**Scope of the Solution:**

Design and implement a water level sensing mechanism.

Develop a microcontroller-based system to process sensor data and trigger alerts.

Create a user interface for monitoring water levels and managing alerts

Ensure the system is robust, reliable, and cost-effective

**Required Components**

**Hardware:**

Arduino Uno or similar microcontroller board

Ultrasonic sensor (e.g, HC-SR04) for water level measurement

LCD display for real-time data visualization

Buzzer or LED for alert notification

Breadboard and jumper wires for circuit connections

**Software:**

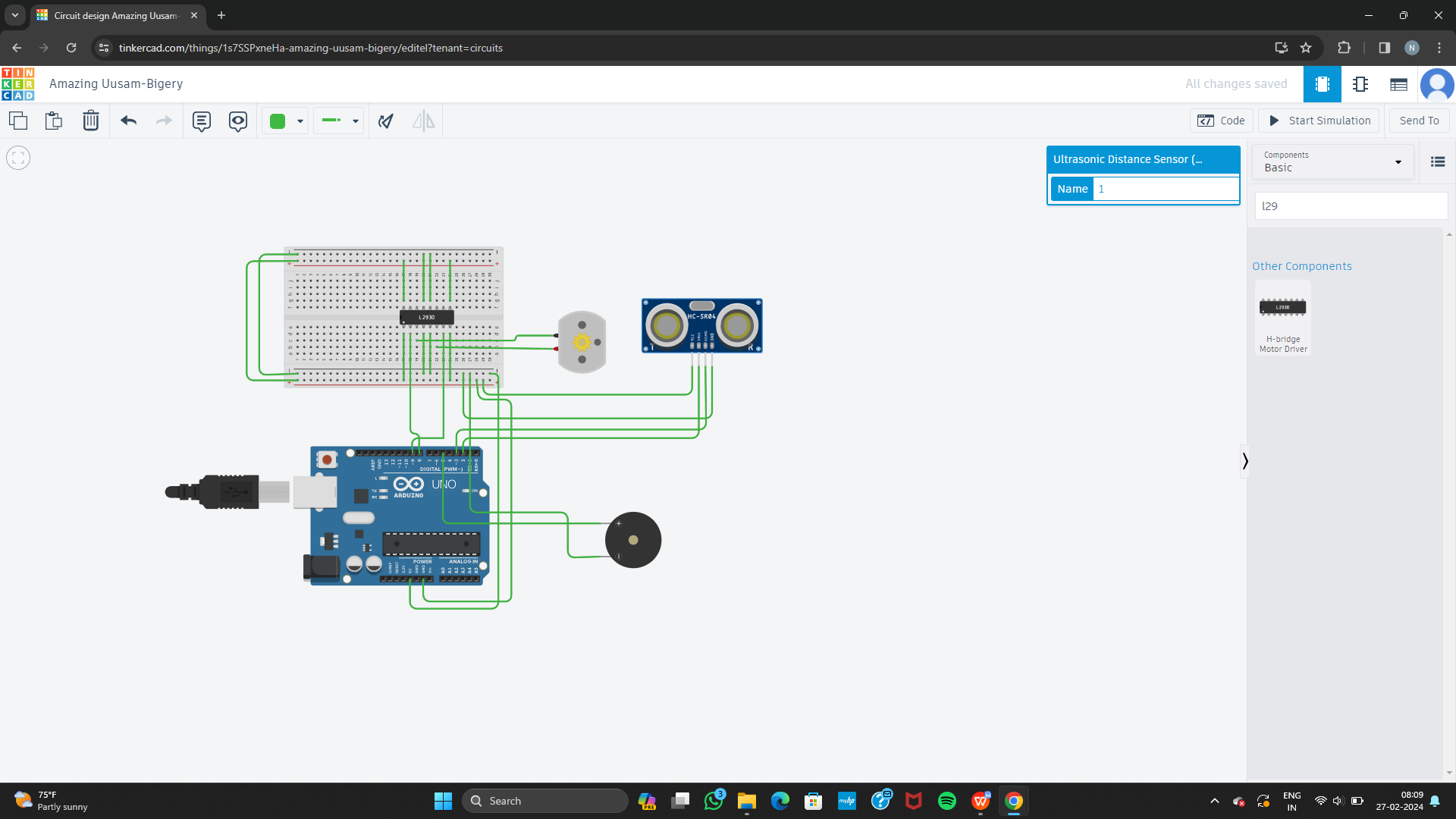
Arduino IDE for programming the microcontroller

Tinkercad for simulating the circuit

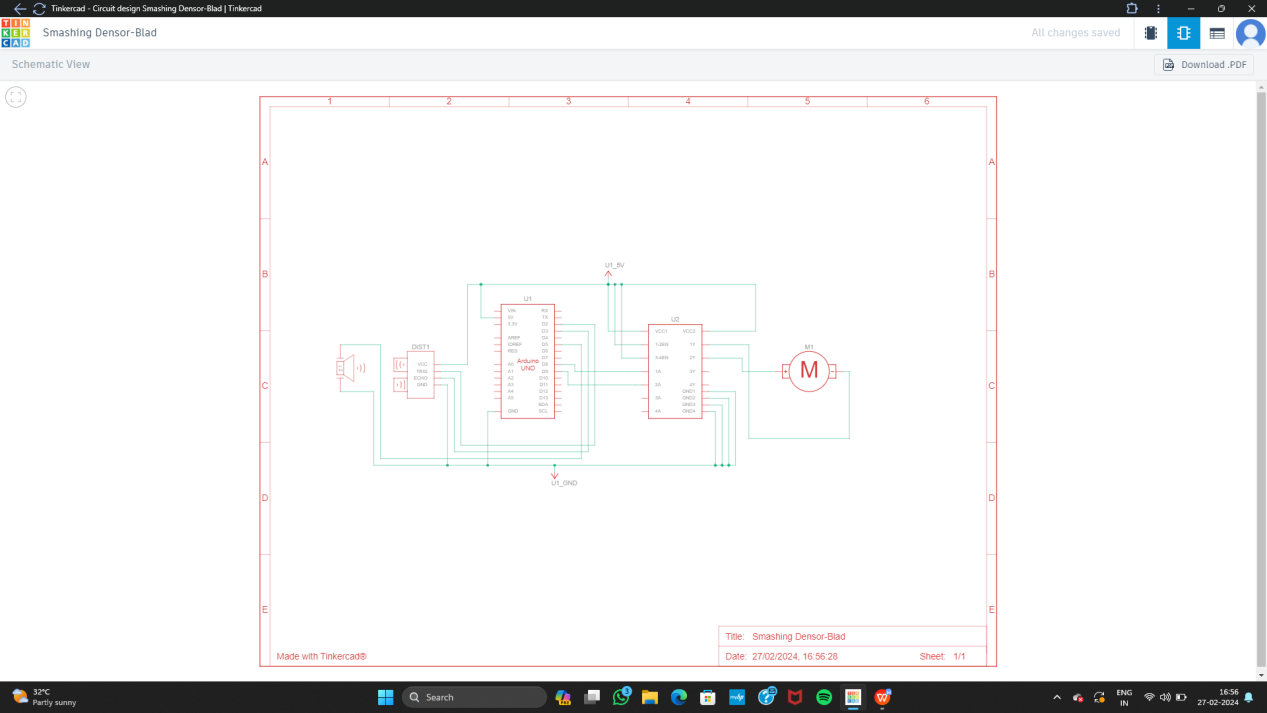
Gerber files for PCB fabrication (if desired)

Programming languages: C/C++ for microcontroller programming

**Circuit diagram**



**Gerber file**



**Code for the execution**

#include <LiquidCrystal.h>

#define trigPin 2

#define echoPin 3

#define buzzerPin 4

#define ledPin 5

LiquidCrystal lcd(12, 11, 7, 6, 5, 4);

void setup() {

lcd.begin(16, 2); // Initialize the LCD with 16 columns and 2 rows

lcd.print("Water Level:"); // Print initial message

pinMode(trigPin, OUTPUT); // Set trigPin as an output

pinMode(echoPin, INPUT); // Set echoPin as an input

pinMode(buzzerPin, OUTPUT); // Set buzzerPin as an output

pinMode(ledPin, OUTPUT); // Set ledPin as an output

}

void loop() {

long duration, distance;

// Send a pulse to the ultrasonic sensor to trigger a measurement

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Measure the duration of the pulse received by the echo pin

duration = pulseIn(echoPin, HIGH);

// Calculate the distance based on the speed of sound (approx. 0.034 cm per microsecond)

distance = (duration \* 0.034) / 2;

// Display the distance on the LCD

lcd.setCursor(0, 1); // Set cursor to the second row

lcd.print(distance); // Print the distance value

lcd.print(" cm"); // Print units

// Check if the water level is above a certain threshold (e.g., 30 cm)

if (distance > 30) {

// If water level is normal, turn off the buzzer and LED

digitalWrite(buzzerPin, LOW);

digitalWrite(ledPin, LOW);

} else {

// If water level is high, activate the buzzer and LED for alert

digitalWrite(buzzerPin, HIGH);

digitalWrite(ledPin, HIGH);

delay(1000); // Delay for 1 second

digitalWrite(buzzerPin, LOW);

digitalWrite(ledPin, LOW);

delay(1000); // Delay for 1 second

}

delay(1000); // Wait for 1 second before taking the next measurement

}

**Github link**