INDUSTRIAL APPLICATIONS OF MICROCONTROLLERS - A PRACTICE BASED APPROACH

Mid Term Project

By:
Aranya Pal
22BEE0064
Vellore Insitute of Technology
Batch:03

Project Details:

Title: Automated Slug detection system for Water- tanks

Description: Develop a Microcontroller Based prototype which alert the user to clean the water tank as the sludge level is more

1_Problem_Statement:

Problem or challenge:

* Inefficient and unreliable monitoring of water tank sludge levels.
* Potential health risks due to contaminated water from sludge buildup.
* Time-consuming and labor-intensive manual cleaning processes.

Context and significance:

* Water is a vital resource, and ensuring its quality is crucial for public health and well-being.
* Sludge accumulation in water tanks is a common issue that affects households, communities, and industries.
* Effective water tank management is essential for preventing water-borne diseases and optimizing water usage.

Existing solutions or approaches and limitations:

* Manual inspection: This method is time-consuming, labor-intensive, and prone to human error.
* Water quality testing: While essential for determining water purity, it does not directly measure sludge levels.
* Professional tank cleaning services: These can be expensive and require scheduling, leading to potential delays.

These existing approaches often fail to provide real-time monitoring, early warning systems, or automated solutions to the problem of sludge buildup.

2_Scope_of_solution:

Focus: Develop a microcontroller-based prototype to detect elevated sludge levels in a water tank and alert the user.

System Inputs: Ultrasonic sensor readings to measure water level.

System Outputs: Visual and audible alert (LED, buzzer).

Limitations: Prototype will be designed for a single tank and assumes a static tank environment.

Solution Aims

Accurately measure water level and calculate sludge level based on tank height.

Reliably detect when sludge level exceeds a predefined threshold.
Provide clear and timely alerts to the user.

Demonstrate proof of concept for a more comprehensive water tank management system.

Constraints and Limitations

Hardware Limitations: The chosen microcontroller and sensors may have limitations in terms of accuracy, processing power, and memory.

Environmental Factors: The system's performance may be affected by water temperature, turbidity, and sensor fouling.

Prototype Scope: This project focuses on the core functionality of sludge level detection and alerting. Additional features (e.g., data logging, remote monitoring) are out of scope.

Power Consumption: The system should be designed to minimize power consumption for potential battery operation.

3_Required_Components:

Hardware Components:

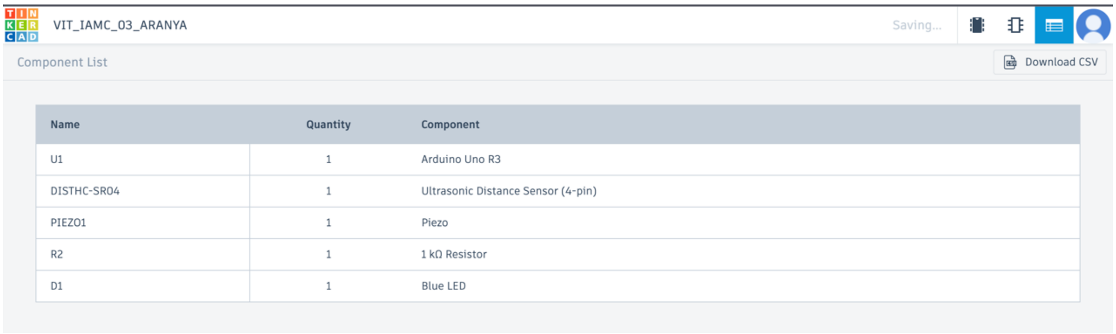
1. Microcontroller: Arduino Uno
2. Ultrasonic Sensor: HC-SR04
3. Active Buzzer: For audible alert
4. LED: For visual indication
5. Resistor: To limit current to the LED
6. Jumper Wires: For connections
7. Breadboard: For prototyping
8. Power Supply: Computer USB Port
9. Water Tank: For testing purposes

Software Components:

IDE: Arduino IDE (for Arduino-based projects)
Programming Language: C++ (Arduino language)

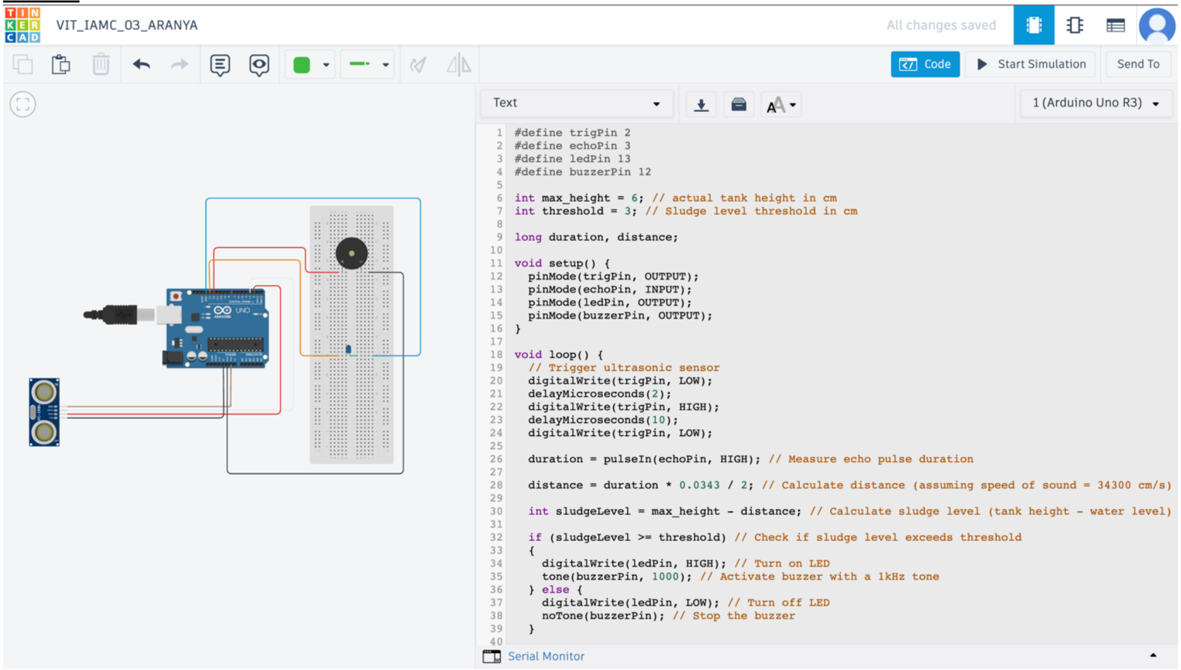
4_Simulated_Circuit_(TINKERCAD):

Components:



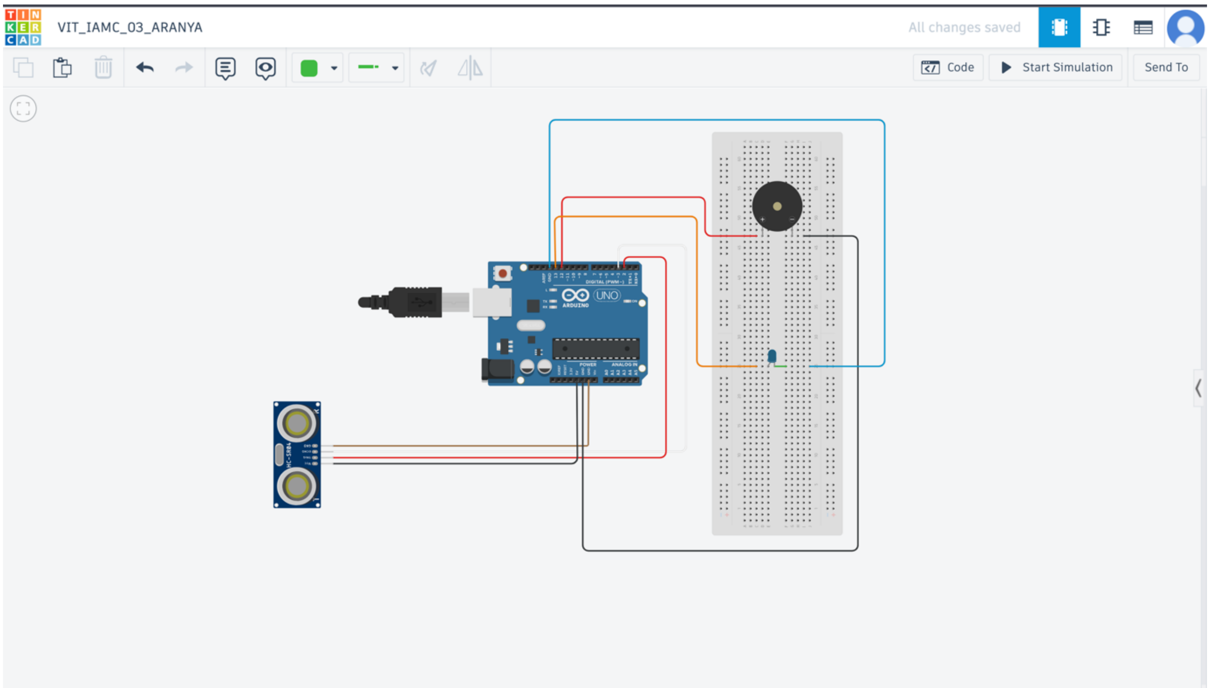
Name	Quantity	Component
U1	1	Arduino Uno R3
DISTHC-SR04	1	Ultrasonic Distance Sensor (4-pin)
PIEZO1	1	Piezo
R2	1	1 kΩ Resistor
D1	1	Blue LED

Code:

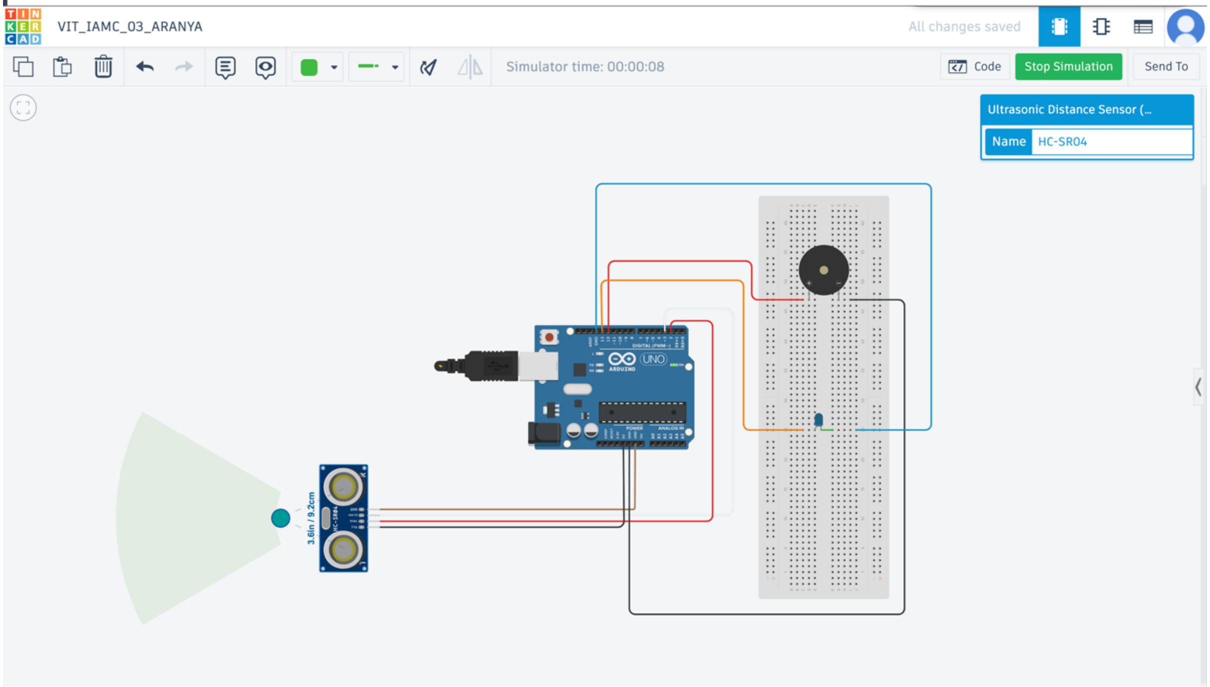


```
#define trigPin 2
#define echoPin 3
#define ledPin 13
#define buzzerPin 12

int max_height = 6; // actual tank height in cm
int threshold = 3; // Sludge level threshold in cm

long duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  // Trigger ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH); // Measure echo pulse duration

  distance = duration * 0.0343 / 2; // Calculate distance (assuming speed of sound = 34300 cm/s)

  int sludgeLevel = max_height - distance; // Calculate sludge level (tank height - water level)

  if (sludgeLevel >= threshold) // Check if sludge level exceeds threshold
  {
    digitalWrite(ledPin, HIGH); // Turn on LED
    tone(buzzerPin, 1000); // Activate buzzer with a 1kHz tone
  } else {
    digitalWrite(ledPin, LOW); // Turn off LED
    noTone(buzzerPin); // Stop the buzzer
  }

```

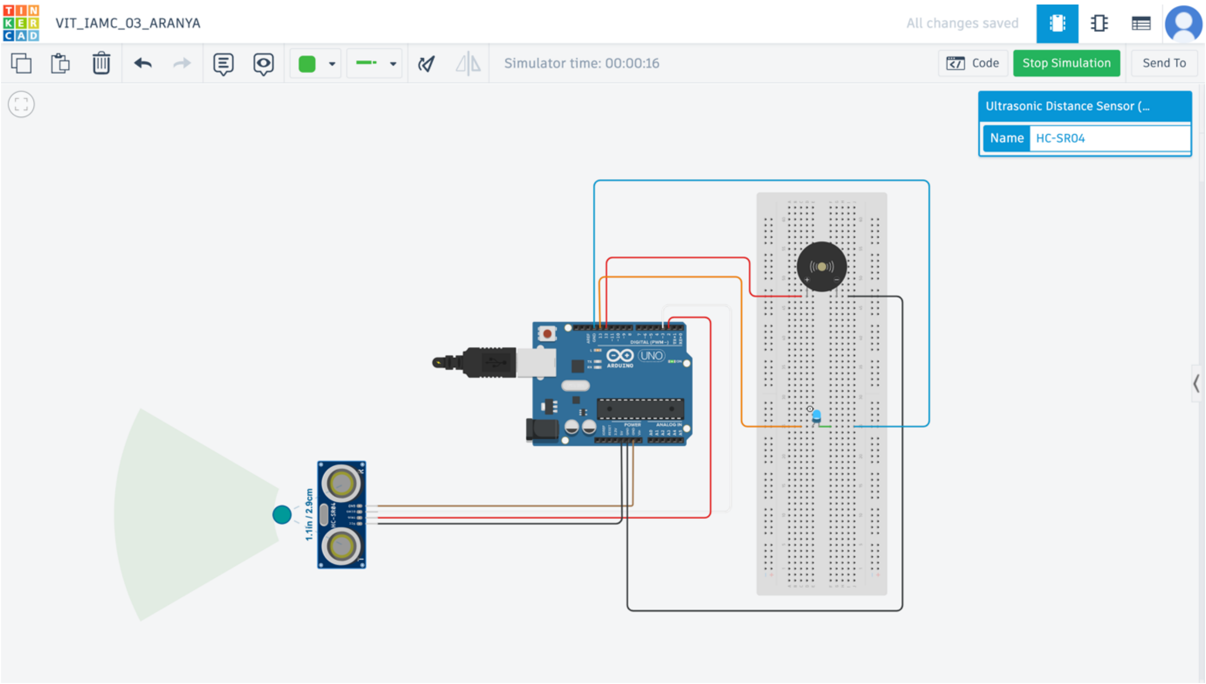
Circuit:



Buzzer- OFF



Buzzer- ON



VIT_IAMC_03_ARANYA

Simulator time: 00:00:16

Ultrasonic Distance Sensor (...

Name HC-SR04

5_Video_Of_The_Demo:

Here is the drive link for the video of the Demo:

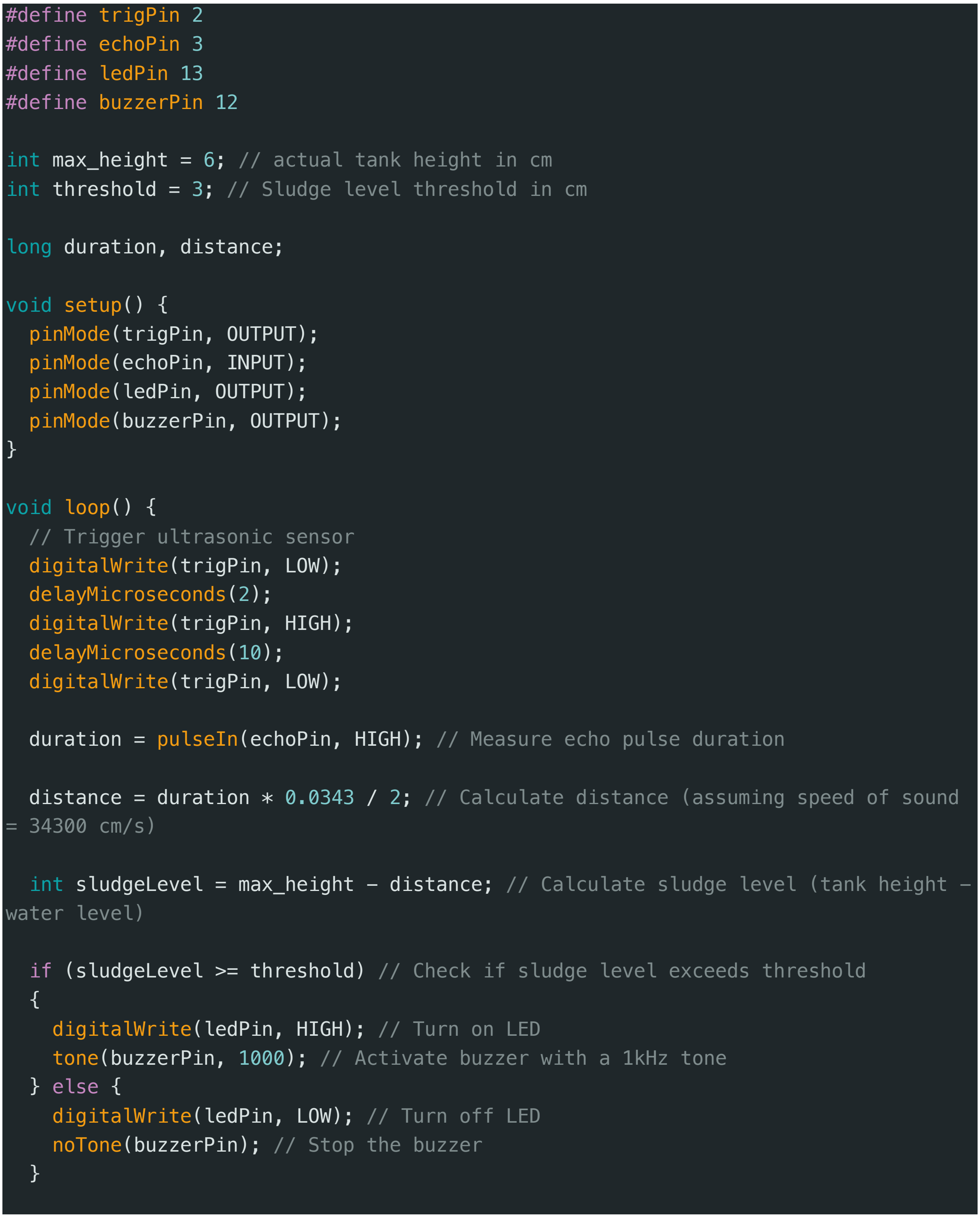
https://drive.google.com/file/d/1O1hHZ29ze3XZYyoVlLxDRecvPN-S35s_/view?usp=sharing

6_Gerber_File:

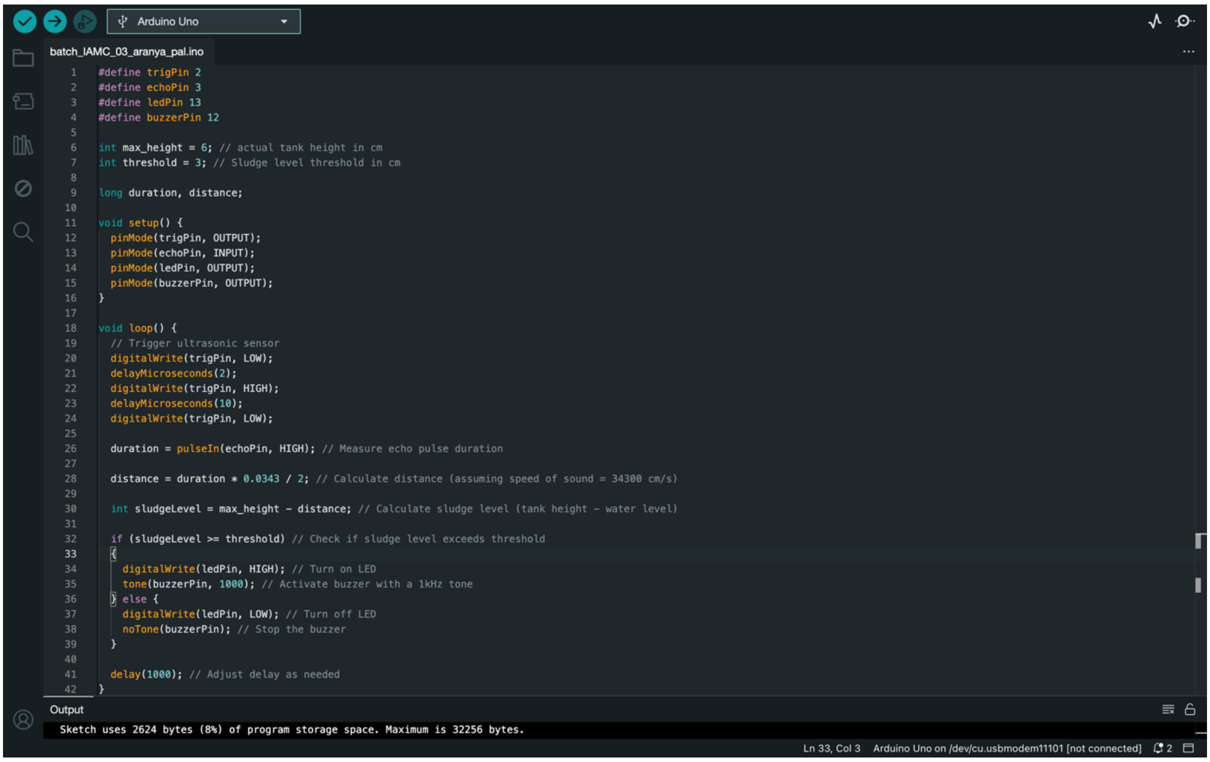
Gerber files are used for PCB (Printed Circuit Board) manufacturing. Since this prototype is designed for breadboard use, a Gerber file was not necessary.

7:Code_For_The_Solution:

Language used: C++



```cpp
#define trigPin 2
#define echoPin 3
#define ledPin 13
#define buzzerPin 12

int max_height = 6; // actual tank height in cm
int threshold = 3; // Sludge level threshold in cm

long duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  // Trigger ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH); // Measure echo pulse duration

  distance = duration * 0.0343 / 2; // Calculate distance (assuming speed of sound
= 34300 cm/s)

  int sludgeLevel = max_height - distance; // Calculate sludge level (tank height -
water level)

  if (sludgeLevel >= threshold) // Check if sludge level exceeds threshold
  {
    digitalWrite(ledPin, HIGH); // Turn on LED
    tone(buzzerPin, 1000); // Activate buzzer with a 1kHz tone
  } else {
    digitalWrite(ledPin, LOW); // Turn off LED
    noTone(buzzerPin); // Stop the buzzer
  }
  delay(1000); // Adjust delay as needed
}
```



```
#define trigPin 2
#define echoPin 3
#define ledPin 13
#define buzzerPin 12

int max_height = 6; // actual tank height in cm
int threshold = 3; // Sludge level threshold in cm

long duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  // Trigger ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH); // Measure echo pulse duration

  distance = duration * 0.0343 / 2; // Calculate distance (assuming speed of sound = 34300 cm/s)

  int sludgeLevel = max_height - distance; // Calculate sludge level (tank height - water level)

  if (sludgeLevel >= threshold) // Check if sludge level exceeds threshold
  {
    digitalWrite(ledPin, HIGH); // Turn on LED
    tone(buzzerPin, 1000); // Activate buzzer with a 1kHz tone
  } else {
    digitalWrite(ledPin, LOW); // Turn off LED
    noTone(buzzerPin); // Stop the buzzer
  }

  delay(1000); // Adjust delay as needed
}
```

```
Sketch uses 2624 bytes (8%) of program storage space. Maximum is 32256 bytes.
```

GITHUB Link

https://github.com/palaranya/VIT-IAMC-03