Dam Water level Alert System

AIM –

The aim of this project is to develop a real-time, automated dam water level sensing and alert system using microcontroller-based technology and water level sensors. The system will utilize ultrasonic sensor interfaced with an Arduino Uno R3 microcontroller to continuously monitor the water level in a dam. Upon reaching predefined threshold levels, the system will automatically trigger alerts through buzzer, LEDs and LCD, enabling timely response and improved flood risk management.

PROBLEM STATEMENT –

In the face of increasing climate change and unpredictable rainfall patterns, the risk of dam overflows and subsequent flooding has become a serious concern. These events can lead to loss of life, damage to property and agriculture. Traditional water level monitoring methods are often manual, lack real-time capabilities and are prone to delays in response.

To address this issue, there is a growing need for a reliable, automated system that can continuously monitor dam water levels and generate timely alerts when critical thresholds are reached. This project contributes to Sustainable Development Goal 6 (Clean Water and Sanitation) by promoting efficient water resource management and to SDG 13 (Climate Action) by enhancing disaster risk reduction through early warning systems.

SCOPE OF THE SOLUTION –

The proposed solution focuses on the development of a low-cost, Arduino-based system capable of monitoring dam water levels in real time and issuing alerts when critical thresholds are reached. The scope includes:

1. Sensor Integration: Using ultrasonic sensors to detect and measure the dam's water level accurately.
2. Microcontroller-Based Processing: Employing an Arduino board to process sensor data and make real-time decisions.
3. Visual Data Display: Displaying real-time water level readings and alert messages on an LCD screen for on-site visibility.
4. Alert System: Generating local alerts through buzzers and LEDs.
5. Power Efficiency: Supporting battery operation for remote deployment.
6. Scalability: Designed for easy adaptation to various dam sizes and conditions with minimal changes.

This system is suitable for dam authorities, water resource departments and disaster management teams to ensure early warning, improve safety, and support sustainable water resource management.

<u>REQUIRED COMPONENTS</u> –

<u>IDE Name</u>: Arduino IDE

<u>Software</u>: Tinkercad, EasyEDA

<u>Hardware</u>:

Sr. No.	Components	Qty.
1	Arduino Uno R3	1
2	Ultrasonic Sensor (HC-SR04)	1
3	16 x 2 LCD	1
4	LED	4 (red, orange, yellow, green)
5	Resistor	2 (1 kΩ, 220 Ω)
6	Buzzer	1
7	Servo motor	1
8	Breadboard	1
9	Jumper wires	-

<u>WORKING PRINCIPLE</u> –

The system operates by continuously sensing the water level in a dam using a suitable sensor, such as an ultrasonic sensor, placed at a fixed position above or within the water surface.

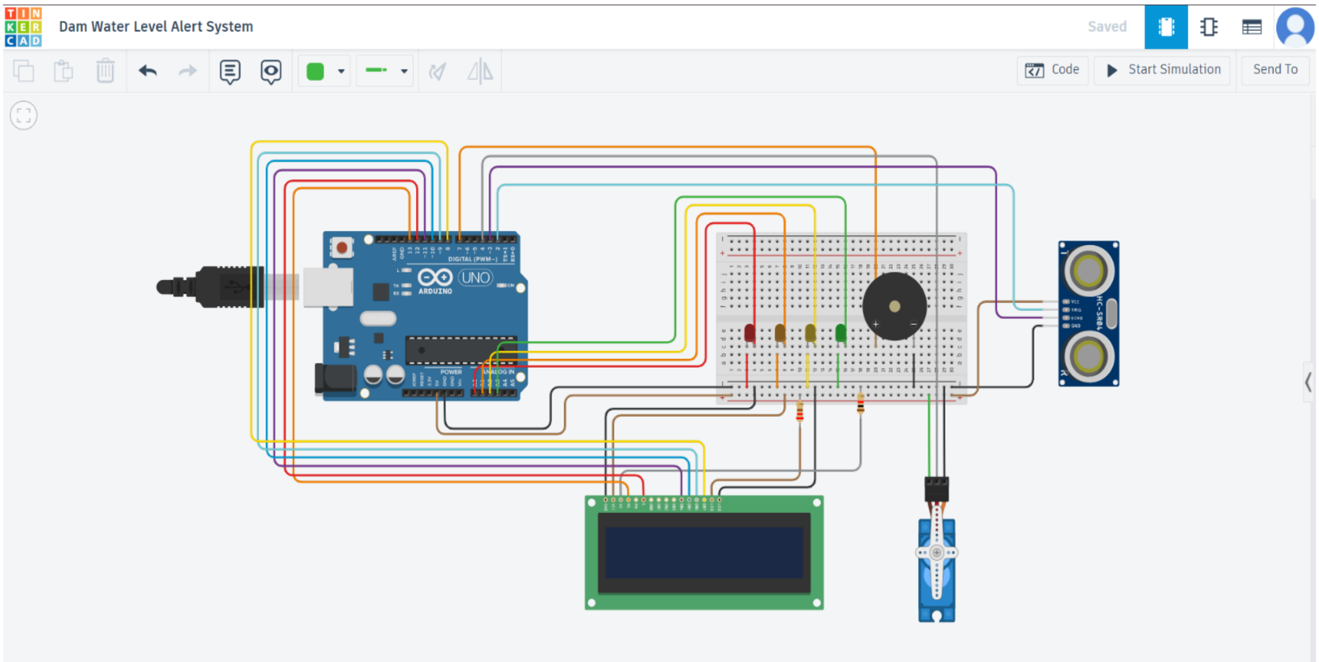
1. <u>Sensing</u>: The sensor detects the current water level and sends an analog or digital signal to the Arduino microcontroller.
2. <u>Processing</u>: The Arduino processes this data and compares the measured level against predefined threshold values (e.g., safe, low, medium and danger levels).
3. <u>Display</u>: The current water level status is shown in real-time on a 16x2 LCD, allowing local personnel to view the readings directly.
4. <u>Alert Generation</u>: If the water level exceeds the given threshold, the system activates a buzzer and LED to indicate a rising risk.
5. <u>Power Supply</u>: The entire system can be powered through the mains or battery depending on deployment conditions.

This real-time monitoring and alert mechanism enables proactive decision-making to prevent flooding and ensure dam safety.

SIMULATED CIRCUIT –

https://www.tinkercad.com/things/bl6FyQ5ndAB-dam-water-level-alert-system?sharecode=A_GhaiiVqzUw119I0Y7A5W6wuiDzckglPG_lqzoxoyA

The circuit is successfully simulated on Tinkercad.



CODE FOR THE SOLUTION –

sketch_dam.ino

```
#include <Servo.h>     //library servo

#include <LiquidCrystal.h>    //library LCD

const int trig = 2, echo = 3;

int z;

int red = A0, orange = A1, yellow = A2, green = A3;

// Red, Orange, Yellow and Green LEDs connected to pins A0, A1, A2 and A3, respectively

int buzzer = 7;

int duration, Distanceincm;

Servo myservo;          // variable to store the data position

LiquidCrystal lcd (13, 12, 11, 10, 9, 8);

void setup () {

  lcd.begin (16, 2);       //16 = Column, 2 = Line
```

```
  pinMode (trig, OUTPUT); //set the mode of the trig as OUTPUT
  pinMode (echo, INPUT);      //set the mode of the echo as INPUT
  pinMode (red, OUTPUT);     //set the mode from red as OUTPUT
  pinMode (yellow, OUTPUT);          //set the mode from yellow as OUTPUT
  pinMode (green, OUTPUT);//set the mode from green as OUTPUT
  pinMode (orange, OUTPUT);          //set the mode from orange as OUTPUT
  digitalWrite (green, HIGH); //makes the green variable value HIGH
  myservo.attach (4);   //servo motor cable data signal is connected to pin 4 of the Arduino
}
void loop () {
  digitalWrite (trig, HIGH);     //make the value of the trig variable HIGH
  delay (15);     //time delay of 15 ms
  digitalWrite (trig, LOW);      //make the trig variable value LOW
  duration = pulseIn (echo, HIGH);
//pulseIn totals the waiting time of echo pin - when it is high, the waiting time will be stored
in memory
  Distanceincm = (duration / 58.2);     //calculations to be made as the distance from the
ultrasonic to the surface of the water
  z = 335-Distanceincm; //calculation of water level
  lcd.setCursor (0, 0);  //set at column 0 and row 0
  lcd.print ("Height : ");          //write down "Height : "
  lcd.print(z); //write down the calculation results of the variable z
  delay (1000); //time delay of 1 second
  if (Distanceincm > 240) {
    lcd.clear ();  //to clear the LCD screen
    lcd.setCursor (0, 1);//set on column 0 and row 1
    lcd.print ("Condn : Safe"); //writes "Condn : Safe"
    digitalWrite (red, LOW);
    digitalWrite (orange, LOW);
    digitalWrite (yellow, LOW);
```

```
  digitalWrite (green, HIGH);
  noTone(buzzer);      //the buzzer does not sound
  myservo.write (0); //move the servo by 0 degrees
  delay (20);   //time delay of 20 ms
}
if (Distanceincm > 160 && Distanceincm <= 240) {
  lcd.clear (); //to clear the LCD screen
  lcd.setCursor (0, 1);//set on column 0 and row 1
  lcd.print ("Condn : Low"); //writes "Condn : Low"
  digitalWrite (red, LOW);
  digitalWrite (orange, LOW);
  digitalWrite (yellow, HIGH);
  digitalWrite (green, LOW);
  noTone(buzzer);      //the buzzer does not sound
  myservo.write (30);//move the servo by 30 degrees
  delay (20);   //time delay of 20 ms
}
if (Distanceincm > 80 && Distanceincm <= 160) {
  lcd.clear ();  //to clear the LCD screen
  lcd.setCursor (0, 1);            //set on column 0 and row 1
  lcd.print ("Condn : Medium");       //writes "Condn : Medium"
  digitalWrite (red, LOW);
  digitalWrite (orange, HIGH);
  digitalWrite (yellow, LOW);
  digitalWrite (green, LOW);
  noTone(buzzer);      //the buzzer does not sound
  myservo.write (60);//move the servo by 60 degrees
  delay (20);   //time delay of 20 ms
}
```

```
  if (Distanceincm > 0 && Distanceincm <= 80) {
    lcd.clear (); //to clear the LCD screen
    lcd.setCursor(0, 1); //set on column 0 and row 1
    lcd.print ("Condn : DANGER!!"); //writes "Condn : DANGER!!"
    digitalWrite (red, HIGH);
    digitalWrite (orange, LOW);
    digitalWrite (yellow, LOW);
    digitalWrite (green, LOW);
    buzz (); //go to label 'buzz' function to sound the buzzer
    myservo.write (90);//move the servo by 90 degrees
    delay (20);   //time delay of 20 ms
  }
}
void buzz () {
  tone(buzzer,1000,500); //buzzer sound setting with a freq. 1000 Hz and a long time 500 ms
  delay (300);   //time delay of 300 ms
}
```

VIDEO OF THE DEMO –

Dam Water Level Alert System.mp4

RESULTS –

The Arduino-based dam water level sensing and alert system was successfully implemented and tested under controlled conditions. The system performed as expected, meeting the core objectives of real-time monitoring and threshold-based alerting. The key results observed are:

1. Accurate Water Level Measurement: The ultrasonic sensor reliably measured water levels within a ±1 cm error margin.
2. LCD Display Output: The 16x2 LCD successfully displayed real-time water level data in centimeters and corresponding status messages (e.g., "Safe", "Low", "Medium", "Danger").
3. Alert Mechanisms: Different colours of LED alerts were triggered when water reached the given thresholds and a buzzer alert was triggered when water reached the danger threshold.
4. System Response Time: The system responded to changes in water level within 1 second, ensuring timely alert generation.

5. Power Performance: The system operated stably on both USB power and 9V battery supply during tests.

These results demonstrate that the system is suitable for real-time dam water level monitoring, early flood warning and can be scaled for field deployment with minimal modification.

CHALLENGES AND LIMITATIONS –

Despite the successful implementation and testing of the system, certain challenges and limitations may affect performance or deployment in real-world scenarios:

1. Environmental Interference: Dust, debris, fog, or water splashes can affect sensor readings. Extreme weather conditions (heavy rain, lightning) may cause irregular measurements or damage to components.
2. Sensor Range Limitations: Most low-cost ultrasonic sensors (e.g., HC-SR04) have limited range (usually 2–400 cm), which may not be suitable for deep dams without modification or use of industrial-grade sensors.
3. Remote Alerting: System does not generate SMS alerts, which limits its usability in remote monitoring.
4. Data Storage and Analysis: Without IoT/cloud integration or SD card logging, long-term data collection and trend analysis are not feasible.

FUTURE IMPROVEMENTS –

To enhance the reliability, scalability and usability of the system for real-world deployment, the following improvements are recommended:

1. GSM Module Integration: We can add a GSM Module (SIM800L/SIM900) to send SMS alerts to authorities when danger level is reached.
2. IoT Integration: We can incorporate Wi-Fi (ESP8266/ESP32) or LoRa modules to upload real-time data to cloud platforms (e.g., Thingspeak) for remote access and monitoring.
3. Data Logging and Analytics: We can add an SD card module or cloud database to store water level data and use this data for trend analysis, prediction of flood risks and better dam management decisions.
4. Enhanced Power Management: We can use solar panels with rechargeable batteries for sustainable and autonomous operation in remote areas.
5. Weatherproof Enclosure: We can design and implement a robust, waterproof and dustproof enclosure (IP65 or higher) to protect electronics during harsh weather conditions.
6. Extended Range and Industrial Sensors: We can use industrial-grade sensors (e.g., radar or capacitive level sensors) for deeper or larger dam reservoirs beyond the range of HC-SR04.
7. Local Control and Automation: We can integrate with actuators or valves to automate dam gates or sluice control when critical levels are detected.