**Automated Railway Gate Control using Embedded Systems**

#include <Servo.h>

// Define pins

const int trigPin = 9;

const int echoPin = 10;

const int gateServoPin = 6;

const int buzzerPin = 7;

// Create a Servo object to control the gate

Servo gateServo;

void setup() {

Serial.begin(9600);

// Set up pins

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(buzzerPin, OUTPUT);

gateServo.attach(gateServoPin);

gateServo.write(0); // Gate initially closed

delay(1000); // Wait for system to initialize

}

void loop() {

long duration, distance;

// Send pulse to trigger pin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read the echo pin

duration = pulseIn(echoPin, HIGH);

// Calculate the distance

distance = (duration / 2) \* 0.0344;

// Print the distance to the Serial Monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// If a train is detected (e.g., within 30 cm), close the gate

if (distance < 30) {

closeGate();

} else {

openGate();

}

delay(500); // Wait before the next measurement

}

void openGate() {

// Open gate (move servo to 90 degrees)

gateServo.write(90);

digitalWrite(buzzerPin, LOW); // Turn off buzzer when gate is open

Serial.println("Gate Opened");

}

void closeGate() {

// Close gate (move servo to 0 degrees)

gateServo.write(0);

digitalWrite(buzzerPin, HIGH); // Activate buzzer when gate is closing

Serial.println("Gate Closed - Train Detected");

}