Automated Railway Gate Control Using Embedded Systems

#include <NewPing.h>

// Pin definitions

#define TRIGGER\_PIN 5 // Ultrasonic sensor trigger pin

#define ECHO\_PIN 6 // Ultrasonic sensor echo pin

#define TRAIN\_SENSOR\_PIN 7 // Infrared sensor for train detection

#define GATE\_RELAY\_PIN 8 // Relay to control gate

#define BUZZER\_PIN 9 // Buzzer for alarm

#define WARNING\_LIGHT\_PIN 10 // LED for warning light

// Constants

#define MAX\_DISTANCE 200 // Maximum distance for vehicle detection

#define THRESHOLD\_DISTANCE 20 // Threshold distance in cm for vehicle detection

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE);

enum GateState {

OPEN,

CLOSING,

CLOSED,

OPENING

};

GateState gateState = OPEN;

void setup() {

pinMode(TRAIN\_SENSOR\_PIN, INPUT);

pinMode(GATE\_RELAY\_PIN, OUTPUT);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(WARNING\_LIGHT\_PIN, OUTPUT);

digitalWrite(GATE\_RELAY\_PIN, LOW); // Start with gate open

Serial.begin(9600);

}

void loop() {

int trainDetected = digitalRead(TRAIN\_SENSOR\_PIN);

unsigned int distance = sonar.ping\_cm();

if (trainDetected == HIGH) { // Train detected

Serial.println("Train detected!");

closeGate();

if (distance < THRESHOLD\_DISTANCE) {

triggerAlarm();

}

} else {

if (gateState == CLOSED && distance > THRESHOLD\_DISTANCE) {

openGate();

}

}

delay(100);

}

void closeGate() {

Serial.println("Closing gate...");

digitalWrite(GATE\_RELAY\_PIN, HIGH);

gateState = CLOSING;

digitalWrite(BUZZER\_PIN, HIGH);

digitalWrite(WARNING\_LIGHT\_PIN, HIGH);

delay(5000); // Set how long it takes to fully close

gateState = CLOSED;

}

void openGate() {

Serial.println("Opening gate...");

digitalWrite(GATE\_RELAY\_PIN, LOW);

gateState = OPENING;

digitalWrite(BUZZER\_PIN, LOW);

digitalWrite(WARNING\_LIGHT\_PIN, LOW);

delay(5000); // Set how long it takes to fully open

gateState = OPEN;

}

void triggerAlarm() {

Serial.println("Alert! Vehicle too close to gate!");

for (int i = 0; i < 5; i++) { // Beep for 5 times

digitalWrite(BUZZER\_PIN, HIGH);

delay(500);

digitalWrite(BUZZER\_PIN, LOW);

delay(500);

}

}