

Level Crossing Code

```

#include <Servo.h>

Servo myservo;
int up = 18;                // The up position of the servo
int down = 54;              // The down position of the servo
int currentPosition = up;   // The current position of the servo
unsigned long currentTime;  // Create the main timer
unsigned long triggerTime;   // Holds the time that the train entered the section
unsigned long servoDelay = 30; // The delay time to slow the barrier movement.
Higher number is a slower barrier.
unsigned long servoTime;     // The servo timer
bool trainDetected = false;  // Records if a train has been detected

void setup() {
    pinMode(3, OUTPUT);      // Alarm pin set as output
    pinMode(4, INPUT_PULLUP); // Train detection sensor on pin 4
    myservo.write(up);        // Set the servo to the up position
    myservo.attach(5);        // Servo on pin 5

    // Split LEDs to avoid drawing too much current through a single pin
    pinMode(6, OUTPUT);      // 2 x Orange LED
    pinMode(7, OUTPUT);      // 2 x Orange LED
    pinMode(8, OUTPUT);      // 2 x Red LED Left
    pinMode(9, OUTPUT);      // 2 x Red LED Left
    pinMode(10, OUTPUT);     // 2 x Red LED Right
    pinMode(11, OUTPUT);     // 2 x Red LED Right
}

void loop() {

    currentTime = millis();

    if (digitalRead(4) == HIGH){        // Check if a train is detected

        // ===== TRAIN DETECTED =====

        // Check if this is the first time that the train has been detected and set the
        trigger time
        if (trainDetected == false){
            trainDetected = true;
            triggerTime = currentTime;
            servoTime = currentTime;
        }

        // ##### LIGHTS #####
        // If the time is less than 3 seconds then turn amber lights on
        if (currentTime - triggerTime < 3000){

```

```

    digitalWrite(6, HIGH);
    digitalWrite(7, HIGH);
}

// If the time is between 3 and 3.5 seconds then amber lights off and all red on
else if(currentTime - triggerTime < 3500){
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(10, HIGH);
    digitalWrite(11, HIGH);
}

//If the time is greater than 3.5 seconds then flash the red lights, swapping
every 330 milliseconds
else{
    if(currentTime % 330 > 175){
        digitalWrite(8, HIGH);
        digitalWrite(9, HIGH);
        digitalWrite(10, LOW);
        digitalWrite(11, LOW);
    }
    else{
        digitalWrite(8, LOW);
        digitalWrite(9, LOW);
        digitalWrite(10, HIGH);
        digitalWrite(11, HIGH);
    }
}

// ##### BARRIERS #####
// If the time is over 4.5 seconds then lower the barriers
if (currentTime - triggerTime > 4500){
    if (currentPosition < down){
        if(currentTime - servoTime > servoDelay){
            currentPosition = currentPosition + 1;
            myservo.write(currentPosition);
            servoTime = currentTime;
        }
    }
}

// ##### ALARM #####
// Alarm comes on as soon as a train is detected and goes off when the barriers
are down.
// Tone changes every 400ms.

```

```

if(currentPosition < down){
    if(currentTime % 400 > 200){ // Check to see if 400ms had passed.
        tone(3, 784);           // Play the first tone on pin 3 at 784Hz
    }
    else{
        tone(3, 1046);          // Play the second tone on pin 3 at 1046Hz
    }
}
else{
    noTone(3);
}
}

// ===== NO TRAIN DETECTED =====

else{

    trainDetected = false;    // Set trigger to false

    // ##### ALARM #####
    noTone(3);                // Cancel tone on pin 3

    // ##### BARRIERS #####
    // Raise barriers to up position
    if (currentPosition > up){
        if (currentTime - servoTime > servoDelay){
            currentPosition = currentPosition - 1;
            myservo.write(currentPosition);
            servoTime = currentTime;
        }
    }

    // ##### LIGHTS #####
    // Turn off amber LEDs immediately
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);

    // Turn off red lights when barriers are up
    if (currentPosition > up){
        if(currentTime % 330 > 175){
            digitalWrite(8, HIGH);
            digitalWrite(9, HIGH);
            digitalWrite(10, LOW);
            digitalWrite(11, LOW);
        }
        else{
            digitalWrite(8, LOW);

```

```
        digitalWrite(9, LOW);
        digitalWrite(10, HIGH);
        digitalWrite(11, HIGH);
    }
}
else{
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
}
}

delay(10); // Small delay to give the code a break

}
```