ECE WORKSHOP

PROJECT REPORT

Title: Arduino Laser Security Alarm Project Report

1. Introduction:

The Arduino Laser Security Alarm project aims to create a simple yet effective security system using an Arduino board, a laser module, and a light-dependent resistor (LDR). The system is designed to detect any interruption in the laser beam and trigger an alarm to alert the user.

2. Project Overview:

The project involves the following key components and their functions:

Arduino Board:

The Arduino board acts as the brain of the system, controlling the overall operation. It receives input from the LDR and laser module, processes the data, and triggers the alarm when necessary.

Laser Module:

The laser module emits a continuous laser beam that serves as the security line. When the laser beam is interrupted by any object, the system detects the change in light intensity using the LDR.

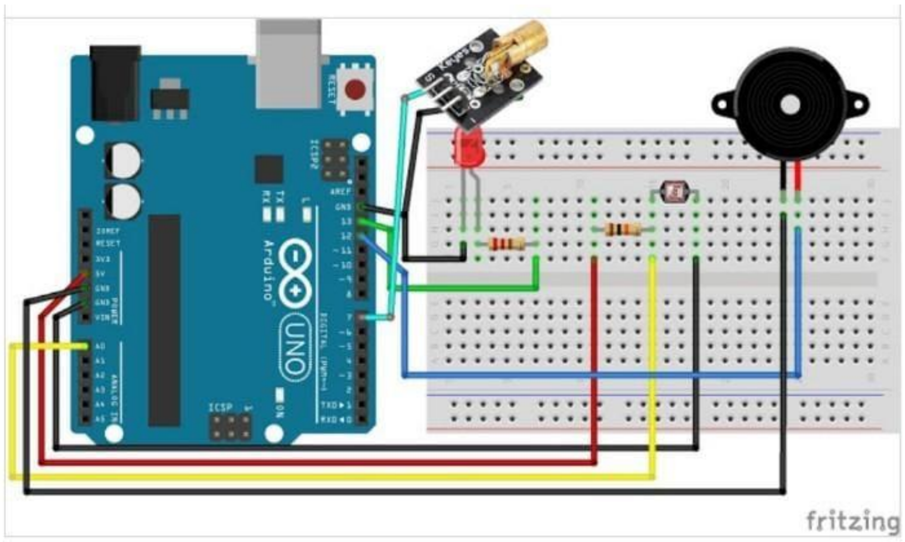
Light-Dependent Resistor (LDR):

The LDR acts as a light sensor and measures the amount of light falling on it. It is placed opposite the laser module, and any change in light intensity caused by an object blocking the laser beam is detected by the LDR.

Buzzer or Alarm:

The buzzer or alarm is triggered by the Arduino when the laser beam is interrupted, indicating a potential security breach.

3. Circuit Diagram:



A typical circuit diagram for the Arduino Laser Security Alarm project would include the following connections:

- Connect the VCC and GND pins of the laser module to the +5V and GND pins of the Arduino, respectively.
- Connect the output pin of the laser module to any digital pin (e.g., Pin 2) of the Arduino.
- Connect one terminal of the LDR to the +5V pin of the Arduino and the other terminal to analog pin (e.g., A0) of the Arduino.
- Connect a buzzer or alarm to any digital pin (e.g., Pin 3) of the Arduino.4.

4. Code:

```
const int triggeredLED = 7;
const int triggeredLED2 = 8;
const int RedLED = 4;
const int GreenLED = 5;
const int inputPin = A0;
const int speakerPin = 12;
const int armButton = 6;


boolean isArmed = true;
boolean isTriggered = false;

int buttonVal = 0;

int prev_buttonVal = 0;

int reading = 0;

int threshold = 0;



const int lowrange = 2000;

const int highrange = 4000;


void setup(){


  pinMode(triggeredLED, OUTPUT);

  pinMode(triggeredLED2, OUTPUT);

  pinMode(RedLED, OUTPUT);

  pinMode(GreenLED, OUTPUT);

  pinMode(armButton, INPUT);

  digitalWrite(triggeredLED, HIGH);

  delay(500);

  digitalWrite(triggeredLED, LOW);


   calibrate();

   setArmedState();

}


void loop(){
```

```
reading = analogRead(inputPin);



int buttonVal = digitalRead(armButton);
if ((buttonVal == HIGH) && (prev_buttonVal == LOW)){
  setArmedState();
  delay(500);
}

if ((isArmed) && (reading < threshold)){
  isTriggered = true;}

if (isTriggered){


   for (int i = lowrange; i <= highrange; i++)
  {
    tone (speakerPin, i, 250);
  }

  for (int i = highrange; i >= lowrange; i--)
  {
    tone (speakerPin, i, 250);
  }



  digitalWrite(triggeredLED, HIGH);
  delay(50);
  digitalWrite(triggeredLED, LOW);
```

```
    delay (50);
    digitalWrite(triggeredLED2, HIGH);
    delay (50);
    digitalWrite(triggeredLED2, LOW);
    delay (50);
    }

  delay(20);
}

void setArmedState(){

  if (isArmed){
    digitalWrite(GreenLED, HIGH);
    digitalWrite(RedLED, LOW);
    isTriggered = false;
    isArmed = false;
  } else {
    digitalWrite(GreenLED, LOW);
    digitalWrite(RedLED, HIGH);
    tone(speakerPin, 220, 125);
    delay(200);
    tone(speakerPin, 196, 250);
    isArmed = true;
  }
}
```

```
void calibrate(){

  int sample = 0;
  int baseline = 0;
  const int min_diff = 200;
  const int sensitivity = 50;
  int success_count = 0;

  digitalWrite(RedLED, LOW);
  digitalWrite(GreenLED, LOW);

  for (int i=0; i<10; i++){
    sample += analogRead(inputPin);
    digitalWrite(GreenLED, HIGH);
    delay (50);
    digitalWrite(GreenLED, LOW);
    delay (50);
  }

  do
  {
    sample = analogRead(inputPin);

    if (sample > baseline + min_diff){
      success_count++;
      threshold += sample;
      digitalWrite(GreenLED, HIGH);
      delay (100);
      digitalWrite(GreenLED, LOW);
      delay (100);
    } else {
      success_count = 0;
      threshold = 0;
    }

  } while (success_count < 3);

  threshold = (threshold/3) - sensitivity;

  tone(speakerPin, 196, 250);
  delay(200);
  tone(speakerPin, 220, 125);
}
```

5. Project Workflow:

The Arduino Laser Security Alarm project follows the following workflow:

Initialization:

- Set the digital pin connected to the laser module as an output pin.
- Set the digital pin connected to the buzzer as an output pin.
- Initialize the serial communication for debugging (optional).

Main Loop:

- Read the analog value from the LDR using the analogRead() function.

- Compare the LDR value with a predefined threshold value to determine if the laser beam is interrupted.

- If the beam is interrupted, trigger the alarm by setting the buzzer pin HIGH and display a message indicating the breach (optional).

- If the beam is not interrupted, keep the buzzer pin LOW and display a message indicating normal operation (optional).

- Delay for a certain period to avoid continuous triggering of the alarm due to minor interruptions.

6. Implementation and Testing:

- Assemble the circuit as per the circuit diagram.

- Upload the Arduino code to the Arduino board.

- Place the laser module and LDR in a suitable position, ensuring that the laser beam is uninterrupted.

- Power on the system and verify that the laser beam is properly aligned.

- Test the system by obstructing the laser beam and observing the alarm activation.

- Fine-tune the threshold value and placement of components if necessary.

7. Conclusion:

The Arduino Laser Security Alarm project demonstrates a practical application of Arduino in creating a simple yet effective security system. By utilizing a laser module and an LDR, the system can detect any interruption in the laser beam and trigger an alarm, providing an alert for potential security breaches. The project can be further expanded and customized to meet specific requirements and integrated with other security systems.

8. Limitations and Future Enhancements:

- The system's effectiveness may be limited if there are strong ambient light sources that interfere.