

If you want to implement the **Laser Security System** using an **ESP32**, you can enhance the project with **Wi-Fi capabilities**, allowing remote monitoring and alerts via a web dashboard or mobile app.

✂ Components Required

1. **ESP32** – Main microcontroller with Wi-Fi support
2. **Laser Module** – Acts as a security beam
3. **LDR (Light Dependent Resistor)** – Detects laser interruption
4. **Resistor (10KΩ)** – Used for voltage division with LDR
5. **Buzzer** – Alerts when the beam is broken
6. **LED** – Indicates alarm status
7. **Jumper Wires & Breadboard** – For circuit connections
8. **Power Supply (5V)** – To power the ESP32

⚡ Circuit Connection

1. **LDR Setup:**
 - One end of the LDR connects to **3.3V**.
 - The other end connects to **GPIO34 (Analog Input)** of ESP32 and also to **GND** via a **10KΩ resistor**.
2. **Buzzer & LED:**
 - Buzzer connected to **GPIO26**.
 - LED connected to **GPIO27**.
3. **Laser Module:**
 - VCC to **3.3V**
 - GND to **GND**
 - If it's controllable, connect signal pin to **GPIO25**.

Features & Enhancements

- ✓ **Wi-Fi Monitoring** – ESP32 can send security alerts to a web dashboard or mobile app.
- ✓ **Blynk / Telegram Alerts** – Get real-time alerts on your phone.
- ✓ **Email Notifications** – Send emails when an intrusion is detected.

How It Works

1. **ESP32 connects to Wi-Fi** and starts a **web server**.
2. When someone **crosses the laser beam**, the **LDR value changes**.
3. **ESP32 triggers an alarm** (Buzzer & LED).
4. You can **monitor the system** by entering the ESP32's IP in a web browser.
5. If an intrusion is detected, the page will display 🚨 **Intruder Detected!** 🚨.

Further Enhancements

- ✓ **Add Blynk or Telegram alerts**
- ✓ **Control via Mobile App** (ESP32 sends notifications)
- ✓ **Capture images using ESP32-CAM** when intrusion is detected
- ✓ **Battery-powered version** for outdoor security

Code:

```
#include <WiFi.h>
```

```
#include <WebServer.h>
```

```
#define LDR_PIN 34 // LDR sensor on GPIO34 (Analog)
```

```
#define BUZZER 26 // Buzzer on GPIO26
```

```
#define LED 27    // LED on GPIO27
```

```
const char* ssid = "Your_WiFi_SSID";    // Your WiFi name
```

```
const char* password = "Your_WiFi_Password"; // Your WiFi password
```

```
WebServer server(80); // Web server on port 80
```

```
int threshold = 600; // Adjust based on ambient light conditions
```

```
void handleRoot() {
```

```
    int ldrValue = analogRead(LDR_PIN);
```

```
    String message = "Laser Status: " + String(ldrValue) + "<br>";
```

```
    if (ldrValue < threshold) {
```

```
        message += " 🚨 Intruder Detected! 🚨 ";
```

```
    } else {
```

```
        message += " ✅ Secure Area";
```

```
    }
```

```
    server.send(200, "text/html", message);
```

```
}
```

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    pinMode(BUZZER, OUTPUT);
```

```
    pinMode(LED, OUTPUT);
```

```
WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi...");

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("\nConnected!");
Serial.println(WiFi.localIP());

server.on("/", handleRoot);
server.begin();
}

void loop() {
    int ldrValue = analogRead(LDR_PIN);
    Serial.println(ldrValue);

    if (ldrValue < threshold) { // If beam is broken
        digitalWrite(BUZZER, HIGH);
        digitalWrite(LED, HIGH);
        Serial.println("🚨 Intruder detected!");
    } else {
        digitalWrite(BUZZER, LOW);
        digitalWrite(LED, LOW);
    }

    server.handleClient();
}
```

```
delay(100);
```

```
}
```

