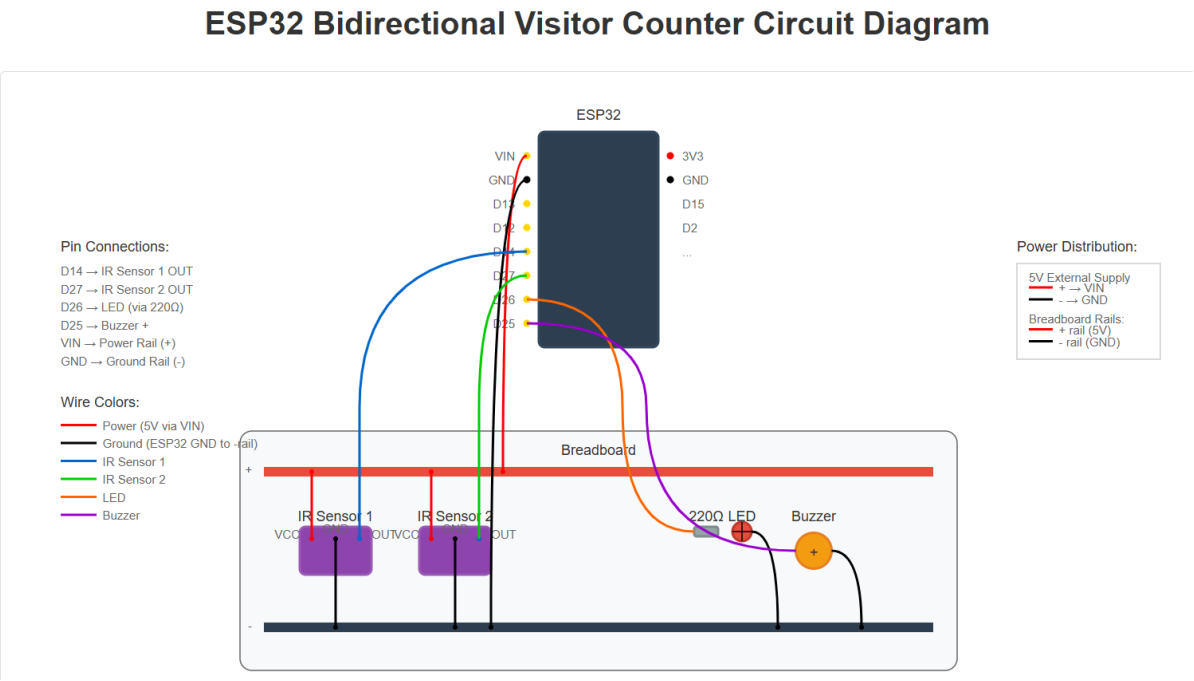
# Bidirectional Visitor Counter using ESP32, IR Sensors, Buzzer, LED and Blynk App

Project Report and Documentation

## 1 Circuit Diagram




### 1.1 Component List

- ESP32 Development Board
- 2× IR Obstacle Avoidance Sensors
- 1× LED (any color)
- 1× 220 Resistor
- 1× Active Buzzer (3-5V)
- 1× Breadboard
- Jumper wires (Male-to-Male and Male-to-Female)

### 1.2 Pin Connection Summary

| Component | Pin | ESP32 Pin |
|---|---|---|
| IR Sensor 1 | OUT | D14 |
| IR Sensor 2 | OUT | D27 |
| LED | Anode (+) | D26 (via 220 resistor) |
| Buzzer | Positive (+) | D25 |
| All Components | VCC/+ | 3.3V |
| All Components | GND/- | GND |

Table 1: Pin Connection Table

## 2 Arduino Code

```
#define BLYNK_PRINT Serial

#define BLYNK_TEMPLATE_ID "TMPL6fTt5tuUg"
#define BLYNK_TEMPLATE_NAME "Bidirectional Visitor Counter"
#define BLYNK_AUTH_TOKEN "VXuRa-O8gETyth0bwScmeDtL0NASoAwZ"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Blynk Auth Token
char auth[] = "VXuRa-O8gETyth0bwScmeDtL0NASoAwZ";
// WiFi credentials - Use your home's WiFi name and password here
char ssid[] = "kingFisher";
char pass[] = "BortY8728@";

// Pin Configuration
#define IR_SENSOR_1 14
#define IR_SENSOR_2 27
#define LED_PIN     26
#define BUZZER_PIN  25

// Variables
int count = 0;
const int threshold = 10; //for sound buzzer
bool lastIR1 = HIGH, lastIR2 = HIGH;
unsigned long lastTriggerTime = 0;
const unsigned long debounceDelay = 300; // ms

void setup() {
  Serial.begin(115200);
  pinMode(IR_SENSOR_1, INPUT_PULLUP);
  pinMode(IR_SENSOR_2, INPUT_PULLUP);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
```

```
36   digitalWrite(BUZZER_PIN, LOW);
37
38   Blynk.begin(auth, ssid, pass);
39 }
40
41 void loop() {
42   Blynk.run();
43
44   bool ir1 = digitalRead(IR_SENSOR_1) == HIGH;
45   bool ir2 = digitalRead(IR_SENSOR_2) == HIGH;
46
47   enum State { IDLE, IR1_TRIGGERED, IR2_TRIGGERED } ;
48   static State state = IDLE;
49   static unsigned long firstTriggerTime = 0;
50   const unsigned long sequenceTimeout = 10000;
51
52   unsigned long now = millis();
53
54   switch (state) {
55     case IDLE:
56       // Start entry sequence if IR1 triggers
57       if (ir1 && !ir2) {
58         state = IR1_TRIGGERED;
59         firstTriggerTime = now;
60       }
61       // Start exit sequence if IR2 triggers
62       else if (ir2 && !ir1) {
63         state = IR2_TRIGGERED;
64         firstTriggerTime = now;
65       }
66       break;
67
68     case IR1_TRIGGERED:
69       // Wait for IR2 to be triggered within the timeout for
          entry
70       if (ir2 && now - firstTriggerTime < sequenceTimeout) {
71         count++;
72         Serial.println("Entry detected");
73         updateDevices();
74         state = IDLE;
75       }
76       // If sensors released or timeout passed, reset
77       else if ((!ir1 && !ir2) || (now - firstTriggerTime >=
          sequenceTimeout)) {
78         state = IDLE;
79       }
80       break;
81
82     case IR2_TRIGGERED:
83       // Wait for IR1 to be triggered within the timeout for exit
84       if (ir1 && now - firstTriggerTime < sequenceTimeout) {
```

```
85          if (count > 0) count--;
86          Serial.println("Exit detected");
87          updateDevices();
88          state = IDLE;
89        }
90        // If sensors released or timeout passed, reset
91        else if ((!ir1 && !ir2) || (now - firstTriggerTime >=
              sequenceTimeout)) {
92          state = IDLE;
93        }
94        break;
95    }
96  }
97
98
99  void updateDevices() {
100   // Light control
101   digitalWrite(LED_PIN, count > 0 ? HIGH : LOW);
102   // Buzzer control
103   if (count >= threshold) {
104     digitalWrite(BUZZER_PIN, HIGH);
105     delay(500);
106     digitalWrite(BUZZER_PIN, LOW);
107   }
108   // Blynk updates
109   Blynk.virtualWrite(V0, count); // Value Display widget
110   Blynk.virtualWrite(V1, count > 0 ? 255 : 0); // LED widget
111 }
112
113 BLYNK_WRITE(V2) {
114   if (param.asInt() == 1) {
115     count = 0;
116     updateDevices();
117     Serial.println("Count reset via Blynk image button");
118   }
119 }
```

# 3 Code Explanation

- **Blynk and WiFi Setup:** Auth token, WiFi SSID, and password are initialized.
- **Pin Definitions:** IR sensors on D14 and D27, LED on D26, and buzzer on D25.
- **Visitor Logic:** Uses a finite state machine (FSM) to detect entry and exit sequences based on IR sensor trigger order.
- **updateDevices():** Updates LED, buzzer, and sends virtual pin values to Blynk app.
- **BLYNK_WRITE():** Resets count if virtual pin V2 is pressed from Blynk image button.

# 4 Hardware Connections (ESP32 + Breadboard)

## ESP32 Pin Layout

- **EN side (connected to breadboard):** VIN, GND, D13, D12, D14, D27, D26, D25
- **BOOT side (open):** 3V3, GND, D15, D2, etc.

## Wiring Instructions

1. **Power**
   - 3.3V (BOOT side) → Breadboard positive rail (male-to-female wire)
   - GND (BOOT side) → Breadboard negative rail (male-to-female wire)
2. **IR Sensor 1 (connected to D14)**
   - VCC → Breadboard + rail
   - GND → Breadboard - rail
   - OUT → D14 (male-to-male)
3. **IR Sensor 2 (connected to D27)**
   - VCC → Breadboard + rail
   - GND → Breadboard - rail
   - OUT → D27 (male-to-male)
4. **LED (connected to D26)**
   - Anode (+, longer leg) → Resistor → D26 (male-to-male)
   - Cathode (-, shorter leg) → GND rail
5. **Buzzer (connected to D25)**
   - + → D25 (male-to-male)
   - - → GND rail

## Jumper Wire Types

- Male-to-Male: Breadboard to ESP32 inserted pins
- Male-to-Female: ESP32 BOOT side pins to breadboard power rails

# 5 Project Features

## 5.1 Bidirectional Detection

The system uses two IR sensors positioned in sequence to detect the direction of movement:

- **Entry:** IR Sensor 1 triggered first, then IR Sensor 2
- **Exit:** IR Sensor 2 triggered first, then IR Sensor 1

## 5.2 Visual and Audio Feedback

- **LED Indicator:** Turns ON when count ¿ 0, OFF when count = 0
- **Buzzer Alert:** Activates when visitor count reaches threshold (10 people)

## 5.3 IoT Integration

- **Blynk App:** Real-time monitoring via smartphone
- **Virtual Pins:** V0 (count display), V1 (LED status), V2 (reset button)
- **Remote Reset:** Count can be reset remotely via Blynk app

# 6 Blynk App Configuration

## 6.1 Virtual Pin Setup

| Virtual Pin | Widget | Function |
|---|---|---|
| V0 | Value Display | Shows current visitor count |
| V1 | LED Widget | Visual indicator (ON/OFF) |
| V2 | Button Widget | Reset counter to zero |

Table 2: Blynk Virtual Pin Configuration

## 6.2 Required Blynk Credentials

- Template ID: TMPL6fTt5tuUg
- Template Name: Bidirectional Visitor Counter
- Auth Token: VXuRa-O8gETyth0bwScmeDtL0NASoAwZ

# 7 How It Works

## 7.1 Finite State Machine Logic

The system operates using a three-state finite state machine:

1. **IDLE State:** Waiting for sensor trigger
2. **IR1_TRIGGERED:** First sensor activated, waiting for second
3. **IR2_TRIGGERED:** Second sensor activated, waiting for first

## 7.2 Detection Sequence

- When a person approaches from outside:
  1. IR Sensor 1 detects person → State changes to IR1_TRIGGERED
  2. IR Sensor 2 detects person → Count increments → State returns to IDLE
- When a person exits from inside:
  1. IR Sensor 2 detects person → State changes to IR2_TRIGGERED
  2. IR Sensor 1 detects person → Count decrements → State returns to IDLE

## 7.3 Timeout Protection

A 10-second timeout prevents false counts from incomplete sequences or sensor noise.

# 8 Troubleshooting

## 8.1 Common Issues

- **Sensors not responding:** Check 3.3V power supply and ground connections
- **False counts:** Adjust sensor sensitivity potentiometers or increase timeout
- **Blynk connection issues:** Verify WiFi credentials and auth token
- **LED/Buzzer not working:** Check pin connections and component polarity

## 8.2 Calibration Tips

- Position sensors 50-80cm apart for optimal detection
- Adjust sensor height to waist level (80-100cm from ground)
- Test detection range by adjusting onboard potentiometers
- Ensure sensors face each other across the doorway

# 9 Future Enhancements

- Data logging with timestamps
- Email/SMS notifications when threshold is reached
- Web dashboard for historical data analysis
- Battery backup for power outages
- Multiple room monitoring with additional sensor pairs

The system provides an affordable and scalable solution for visitor monitoring in various applications including retail stores, offices, exhibitions, and smart home automation.