

# **Real-Time Occupancy Detection System**

**PROJECT REPORT FOR THE COURSE: INDUSTRIAL  
APPLICATIONS OF MICROCONTROLLERS BY L&T  
EDUTECH**

**NAME: RAGHAV KISHORE VENKATESH**

**EMAIL: [raghav.kishore2022@vitstudent.ac.in](mailto:raghav.kishore2022@vitstudent.ac.in)**

**CAMPUS: VIT CHENNAI**

**Github Repository Link:**

**[https://github.com/raghavkishore2005/L-T Project](https://github.com/raghavkishore2005/L-T_Project)**

**Aim:** To develop a **microcontroller-based real-time room occupancy detection system** using motion sensors and automation logic to reduce power consumption by turning off electrical appliances when no human presence is detected for a certain time.

**Scope:**

The system is designed for:

- Small to medium-sized **indoor spaces** like rooms, cabins, or labs
- **Energy efficiency** by cutting off fans and lights when the room is unoccupied
- Use in **smart buildings** or **IoT-based home automation**
- Scalable by integrating with cloud platforms, WiFi/Bluetooth modules, or smart meters

**Components:**

Component	Quantity	Purpose
Arduino UNO / Nano 33 IoT	1	Main microcontroller
PIR Motion Sensors (HC-SR501 / EKMC1609112)	2	Detect motion at entry and exit
Red LED	2	Visual indicator for Fan & Light

Component	Quantity	Purpose
Resistors (1kΩ)	2	Current limiting for LEDs
Breadboard / PCB	1	Prototype circuit
Jumper Wires	10+	Wiring and connections
Power Supply / USB	1	Power for Arduino

Tool	Use
TinkerCad	Circuit simulation & testing
EasyEDA	Schematic + Gerber file generation
Arduino IDE	Code upload & Serial Monitor
GitHub	Final submission repository

## Implementation:

### a. Logic Overview

- **Two PIR sensors** are placed at the room entrance: one facing **outside**, and one facing **inside**.

- When motion is detected **from outside to inside**, the system assumes **entry**.
- When motion is detected **from inside to outside**, the system assumes **exit**.
- A **cooldown timer (30 seconds)** starts after exit detection. If no re-entry occurs, appliances are turned OFF.

## b. Arduino Code

The Arduino code:

- Monitors PIR pins using `digitalRead()`
- Controls LEDs using `digitalWrite()`
- Uses `millis()` to track cooldown without blocking
- Logs all activity via `Serial.println()`

Code:

```
// Pin configuration
const int PIR_OUTDOOR = 2;
const int PIR_INDOOR = 3;
const int LIGHT_PIN = 8;
const int FAN_PIN = 9;

// State
bool roomOccupied = false;
unsigned long lastExitTime = 0;
const unsigned long cooldownTime = 30000;

// Flags for detecting sequence
bool outdoorTriggered = false;
```

```

bool indoorTriggered = false;
unsigned long triggerTime = 0;
const unsigned long triggerWindow = 3000;

void setup() {
    pinMode(PIR_OUTDOOR, INPUT);
    pinMode(PIR_INDOOR, INPUT);
    pinMode(LIGHT_PIN, OUTPUT);
    pinMode(FAN_PIN, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    int outdoorMotion = digitalRead(PIR_OUTDOOR);
    int indoorMotion = digitalRead(PIR_INDOOR);
    unsigned long currentTime = millis();

    // ENTRY detection: Outdoor PIR triggers first, then Indoor
    if (outdoorMotion == HIGH && !roomOccupied) {
        outdoorTriggered = true;
        triggerTime = currentTime;
    }
    if (outdoorTriggered && indoorMotion == HIGH && (currentTime - triggerTime
<= triggerWindow)) {
        roomOccupied = true;
        digitalWrite(LIGHT_PIN, HIGH);
        digitalWrite(FAN_PIN, HIGH);
        Serial.println("Person entered, Light & Fan ON");
        outdoorTriggered = false;
        delay(1000);
    }

    // EXIT detection: Indoor PIR triggers first, then Outdoor
    if (indoorMotion == HIGH && roomOccupied) {
        indoorTriggered = true;
        triggerTime = currentTime;
    }
    if (indoorTriggered && outdoorMotion == HIGH && (currentTime - triggerTime
<= triggerWindow)) {
        lastExitTime = currentTime;
        Serial.println("Person exited, Starting cooldown");
        indoorTriggered = false;
        delay(1000);
    }

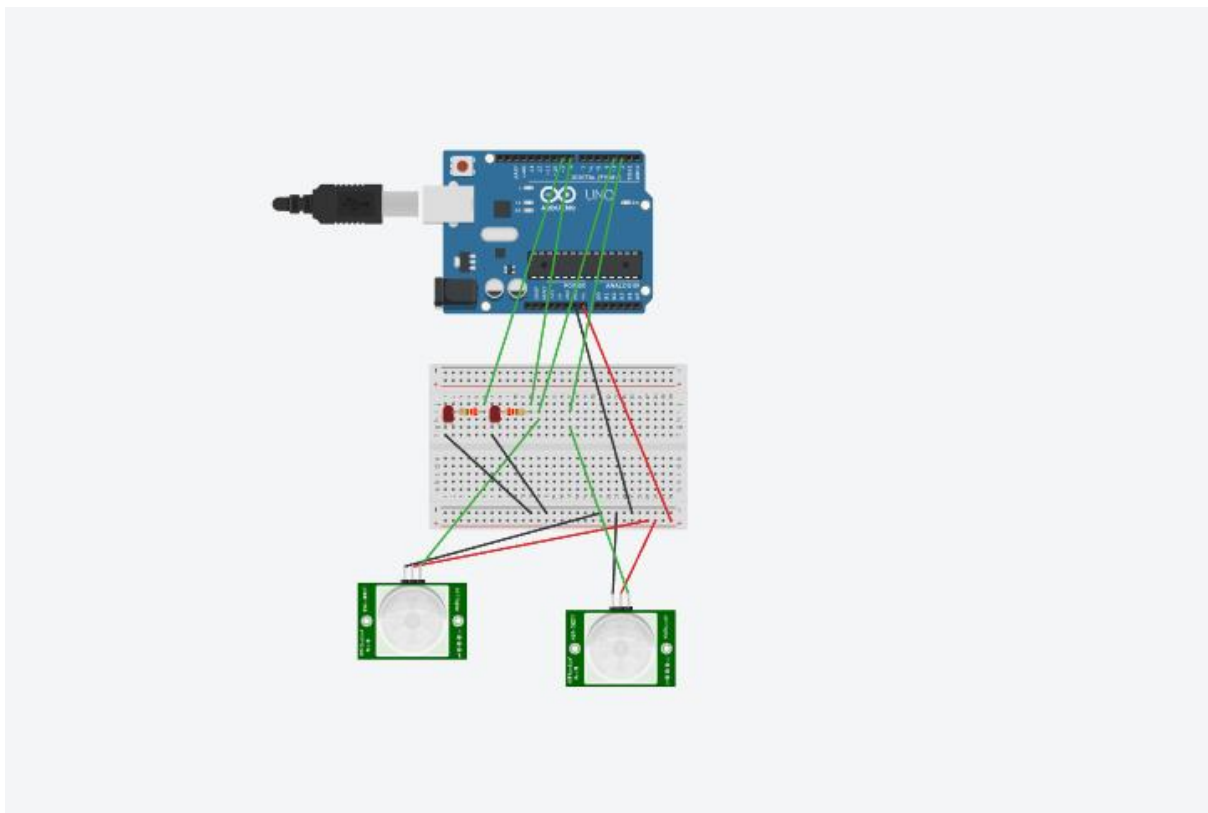
    // Cooldown logic: Turn off after 30 sec if no one returns
    if (roomOccupied && (currentTime - lastExitTime >= cooldownTime) &&
lastExitTime != 0) {

```

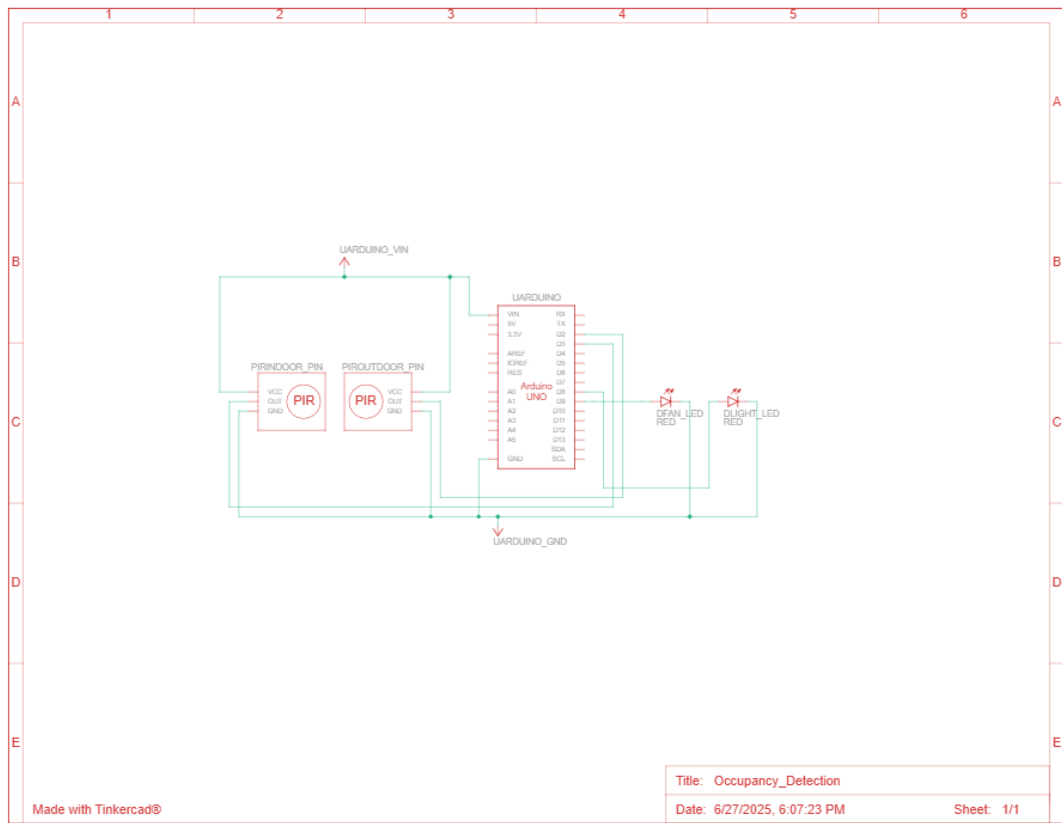
```
roomOccupied = false;
digitalWrite(LIGHT_PIN, LOW);
digitalWrite(FAN_PIN, LOW);
Serial.println("Room now empty, Light & Fan OFF");
lastExitTime = 0;
}
}
```

### c. Simulation & Hardware

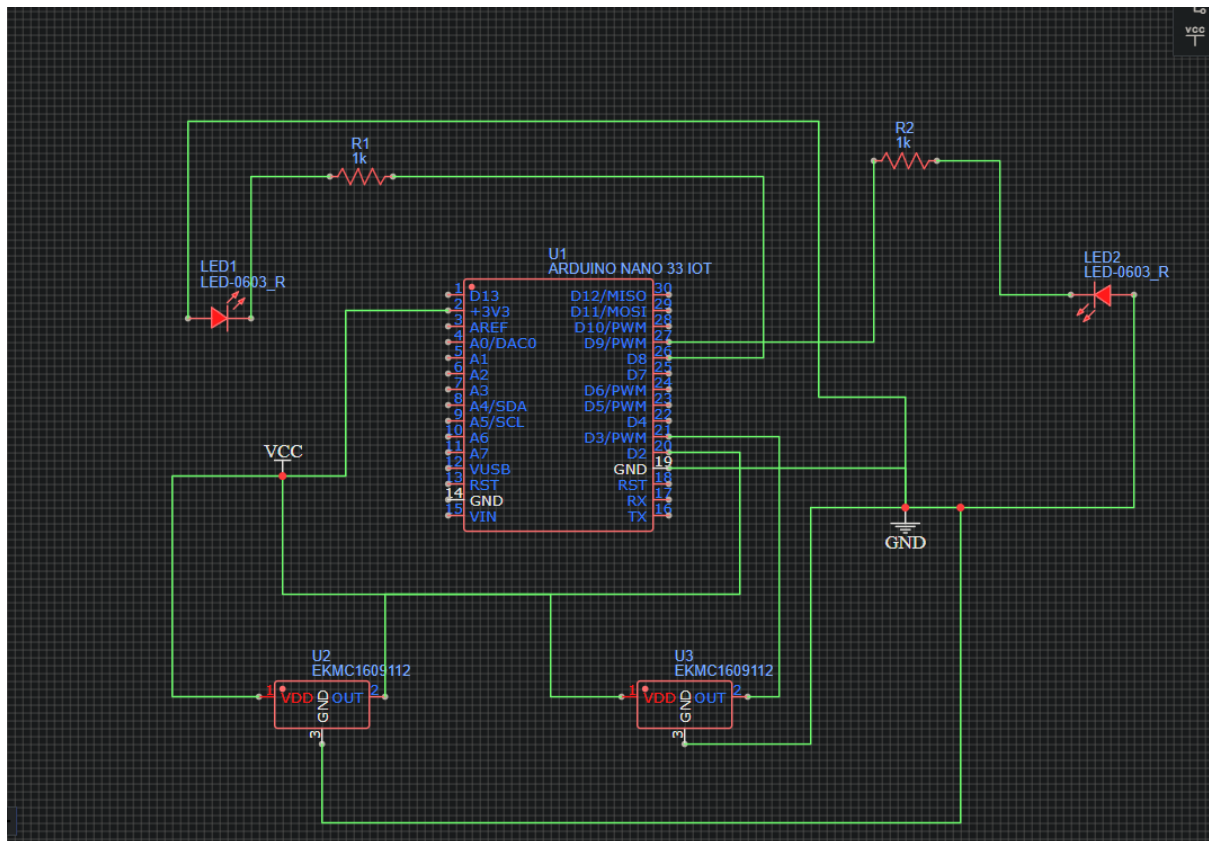
- Simulated on TinkerCad for proof of logic
- Schematic recreated in EasyEDA
- **Gerber file** generated for PCB fabrication
- LEDs simulate appliance status (ON/OFF)



Circuit diagram in TinkerCad



Schematic in TinkerCad



PCB file in EasyEDA

## Results:

- System **accurately tracks** when a person enters or exits
- Lights and fans **turn ON** instantly on entry
- If no motion is detected for **30 seconds post-exit**, the system turns **OFF appliances**
- System is **non-intrusive** (PIR-based, no camera needed)
- Works **consistently** during simulation and hardware tests



**Conclusion:** This real-time occupancy detection system successfully detects human presence using two PIR sensors, automates appliance control, and demonstrates the potential of microcontroller-based energy-efficient systems in smart environments. The system can be scaled further for cloud integration and IoT-based analytics.