FACULTY OF ENGINEERING
UNIVERSITY OF JAFFNA

AUTOMATIC ROOM LIGHT CONTROLLER WITH BIDIRECTIONAL VISITOR COUNTER

A Project Proposal by

Wanigasinghe U. A. – 2020/E/200

Uzair U.M. – 2020/E/201

Arachchi T. H. L. – 2020/E/202

Supervised by

Miss Mary Regin Kamalina A.

August 2024

CONTENTS

1. Introduction

In many public spaces like stadiums, malls, offices, and classrooms, relying on people to manually control lights can be inefficient. This project proposes an automatic room light controller with a bidirectional visitor counter built using an ESP32 board as a solution for this issue. This system aims to optimize energy use, improve occupancy awareness, and enhance security through real-time data transmission. Infrared (IR) sensors strategically placed at entry and exit points detect people entering and leaving the room. The ESP32, a powerful Wi-Fi and Bluetooth-enabled microcontroller, acts as the brain of the system. It processes the sensor data and controls the room lights accordingly. When someone enters, the lights turn on, promoting energy conservation by eliminating unnecessary illumination in empty spaces. Additionally, an OLED128*32 display module with I2C Serial interface shows the visitor count in real-time, providing valuable occupancy tracking information. This compact display offers clear visibility and low power consumption, making it ideal for this application. The integration of Wi-Fi capabilities through the ESP32 board allows for remote monitoring and data transmission. This feature enables the system to send occupancy information to a management app in real-time, enhancing security measures and facilitating better space management decisions. This project offers several advantages. Firstly, it promotes energy efficiency by automatically controlling lights based on occupancy. Secondly, the real-time visitor count displayed on the OLED provides valuable data for managing space utilization and resource allocation. Thirdly, the wireless connectivity allows for remote monitoring and data analysis, which can be crucial for security purposes and optimization of space usage. The project's implementation leverages readily available and cost-effective components. The ESP32 board provides a powerful and flexible platform for programming the system logic and enabling wireless communication. IR sensors are well-suited for detecting movement, and the OLED display offers a clear visual representation of the visitor count. By combining these elements with Wi-Fi connectivity, this automatic room light controller with a bidirectional visitor counter offers a practical and innovative solution for managing lighting and monitoring occupancy in various public spaces. It contributes to environmental benefits by reducing energy consumption, operational benefits by providing valuable data for improved space management, and security enhancements through real-time data transmission to management systems. This IoT-enabled approach allows for more comprehensive control and analysis of space utilization, making it a versatile tool for modern facility management.

2. Project Design and Implementation

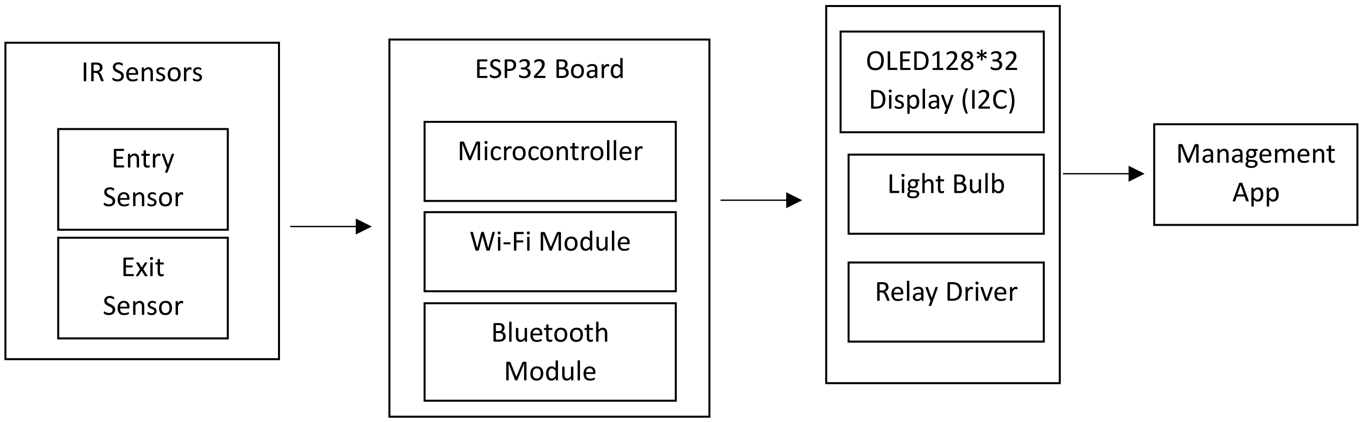
2.1 System Overview

The system consists of the following main components:

- Sensor section for detecting people
- Control section with ESP32 board
- Display section with OLED128*32 I2C Serial module
- Relay driver section for light control
- Wi-Fi and Bluetooth modules (integrated in ESP32) for data transmission
- Management app for receiving and analyzing occupancy data

2.2 Hardware Design

Block Diagram



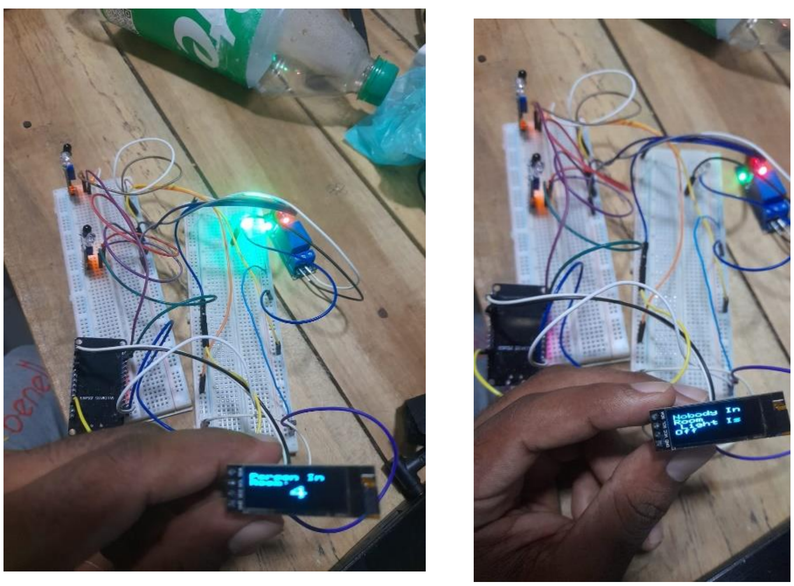


Fig 1 : Physical Implementation

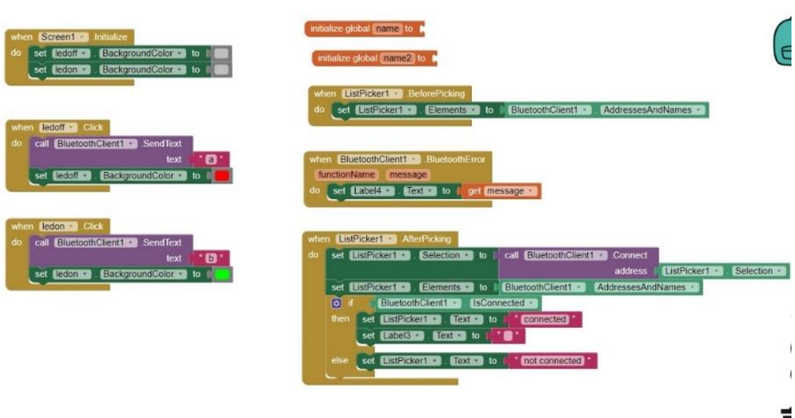


Fig 2 : App

Fig 3 : Bluetooth code

Circuit Diagram

Key Connections:

- Entry IR Sensor connected to ESP32 digital pin 17 (VP)
- Exit IR Sensor connected to ESP32 digital pin 18 (VN)
- Relay driver transistor connected to ESP32 digital pin 4(D2)
- OLED display connected in 4-bit mode
    - SCL pin of OLED connected to ESP32 digital pin 14(D22)
    - SDA pin of OLED connected to ESP32 digital pin 12(D21)
    - VCC pin of OLED connected to ESP32 Vin
    - GND pin of OLED connected to ESP32 GROUND

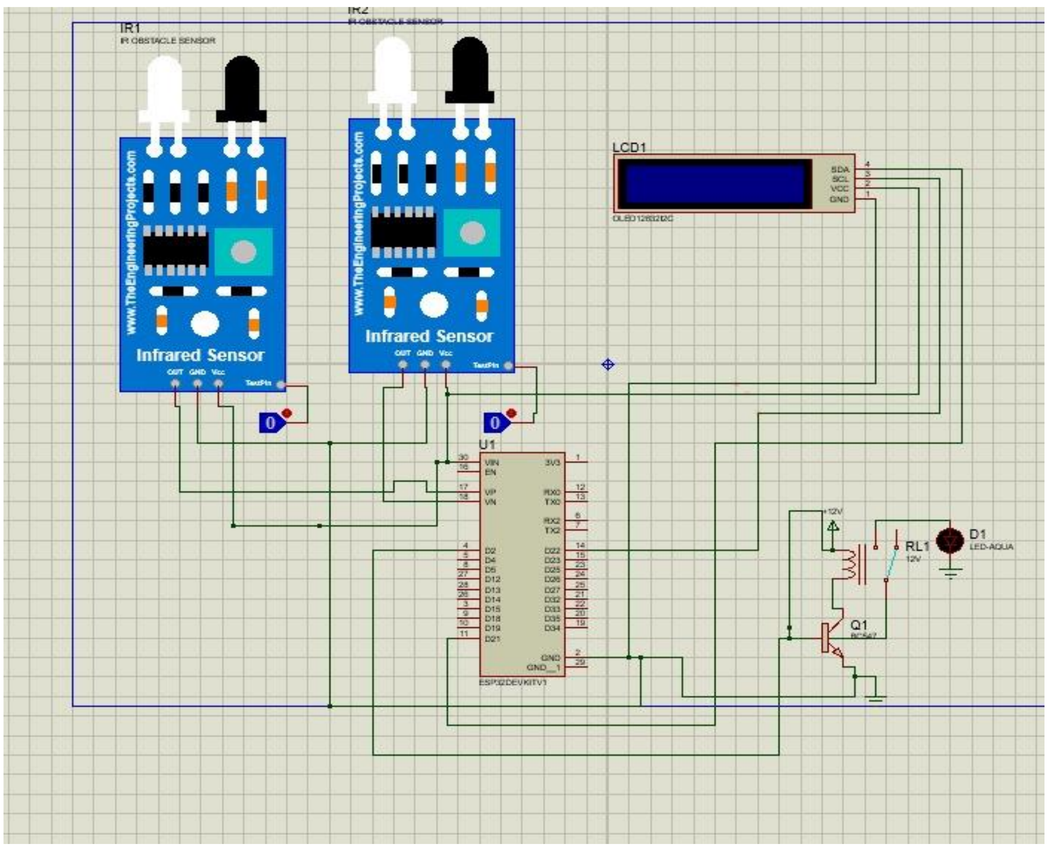


Fig 1 : Circuit Diagram

2.3 Interfacing

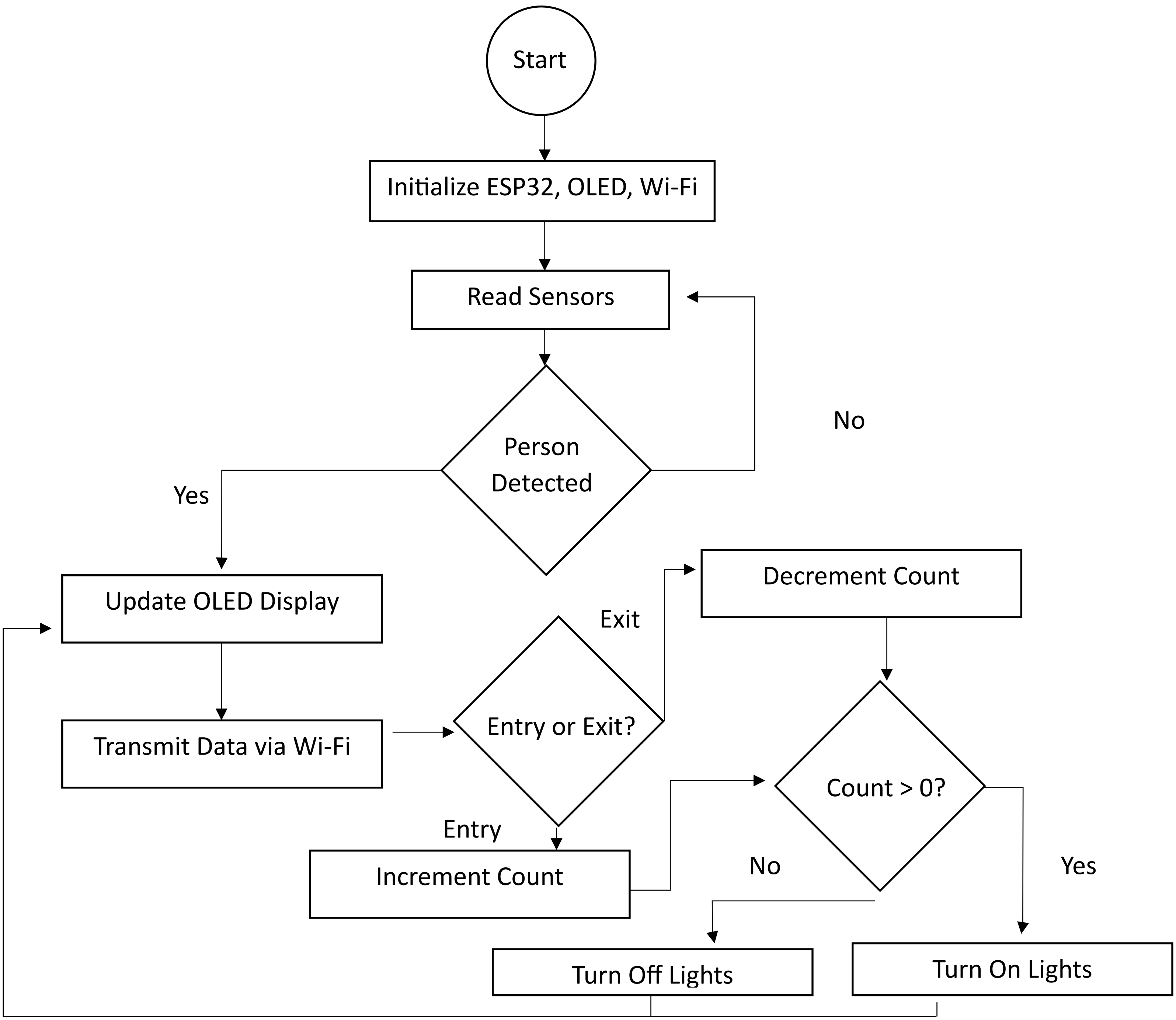
IR sensors interface with ESP32 for people detection
OLED128*32 I2C Serial interface for displaying visitor count
Relay interface with ESP32 for controlling lights
ESP32 board for system control and processing
Wi-Fi module (integrated in ESP32) for data transmission
Management app for receiving and analyzing occupancy data

2.4 Software Design

Key Functions:

- Sensor reading and interrupt handling
- Visitor count management (increment/decrement)
- OLED display updates
- Light control based on occupancy
- Wi-Fi communication for data transmission
- Display in the app

Flowchart



2.5 Implementation

Setting up the ESP32 development environment

- Installing the ESP32 board support package in Arduino IDE
- Configuring the IDE for ESP32 programming

Assembling the hardware components

- Connecting IR sensors to ESP32 GPIO pins
- Wiring the OLED128*32 display using I2C interface
- Integrating relay module for light control
- Setting up power supply for the ESP32 and peripherals

Writing and testing the ESP32 code

- Implementing IR sensor interrupt handling
- Developing visitor count management logic
- Coding OLED display update functions
- Implementing Wi-Fi connectivity and data transmission
- Creating light control algorithms based on occupancy

Integrating all components into a functional prototype

- Assembling the complete system in an enclosure
- Testing and calibrating IR sensors for accurate detection
- Verifying OLED display functionality
- Confirming proper light control operation

Developing and testing the management app

- Creating a user interface for real-time occupancy data
- Implementing data reception and storage functionality
- Testing Wi-Fi communication between ESP32 and the app

Final system testing and optimization

- Conducting comprehensive system tests in various scenarios
- Optimizing power consumption and sleep modes
- Fine-tuning Wi-Fi connectivity and data transmission
- Documenting the final implementation and user guidelines

3. Challenges and Solutions

1. Challenge: Ensuring accurate detection of people entering and exiting.

- Solution: Fine-tuning sensor placement and sensitivity.

2. Challenge: Coordinating multiple hardware elements.

- Solution: Systematic testing and debugging of each component.

4. Timeline

	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
Finalize project details, components, and develop a detailed system design. Procure necessary components									
Proposal submission.									
Develop the hardware prototype.									
Making the Project presentation Slides									
Mid review of the Project and Project Presentation Slide Submission. And also address feedback from the mid-review.									
Write code for sensor data acquisition, microcontroller control for light switching, and visitor count management.									
Integrate the LCD display and develop code for displaying the visitor count.									
Conduct system testing, refine code, and finalize documentation, and prepare final review presentation									
Final project demonstration and report, log books, individual reflection sheets submission.									

5. Components and Cost

Required Components	Cost Estimation (Rs)
IR Sensor	240
OLED Display Module	630
ESP32 Module	1500
Relay Module	500
LED	2000
Jumper wires	300
Total	**2800**

6. Reflection

This project allowed for the practical application of several key concepts and skills in the realm of embedded systems and IoT technologies:

- Embedded systems design principles: The project provided hands-on experience with utilizing the ESP32 board, integrating multiple sensors, and controlling outputs effectively. This reinforced a deep understanding of how to design and implement embedded systems tailored to specific applications.
- Sensor integration and calibration: Implementing IR sensors for accurate detection of room occupancy involved understanding sensor characteristics, calibration processes, and ensuring reliable data acquisition. This aspect of the project emphasized the importance of precision and accuracy in sensor-based systems.
- Microcontroller programming: Writing and debugging code for the ESP32 to process sensor data and control lighting was a critical component of the project. This experience enhanced my skills in programming microcontrollers, particularly in managing inputs, processing data, and generating appropriate outputs.
- Circuit design and implementation: Designing and assembling the hardware components required creating a reliable and efficient circuit. This involved selecting appropriate components, designing the layout, and ensuring the circuit's robustness and functionality.
- Energy efficiency concepts: Developing a system that reduces energy consumption by controlling lights based on occupancy highlighted the importance of energy efficiency in modern technology solutions. This project demonstrated how intelligent design and automation can contribute to significant energy savings.

Overall, the project offered a comprehensive learning experience, integrating theoretical knowledge with practical application. The successful implementation of an automatic room light controller with a bidirectional visitor counter illustrated the potential of IoT-enabled systems to enhance energy efficiency, improve space utilization, and bolster security measures in public spaces. This reflection underscores the value of interdisciplinary skills and innovative thinking in addressing real-world challenges.

7. Conclusion

This project successfully demonstrated the practical application of embedded systems and IoT technologies through the development of an automatic room light controller with a bidirectional visitor counter. By integrating the ESP32 microcontroller, IR sensors, and an OLED display, we created a system that not only optimizes energy use but also enhances occupancy awareness and security.

The project underscored the importance of precision in sensor integration and calibration, effective microcontroller programming, and robust circuit design. It also highlighted how intelligent automation can significantly contribute to energy efficiency, making it a valuable solution for modern facility management.

Through this project, we have shown that readily available and cost-effective components can be leveraged to create innovative and practical solutions for managing lighting and monitoring occupancy in various public spaces. The integration of wireless connectivity for remote monitoring and data transmission further enhanced the system's capabilities, making it a versatile tool for improving space utilization and security.

In conclusion, this project not only provided valuable hands-on experience with key concepts in embedded systems and IoT but also demonstrated the potential for these technologies to address real-world challenges effectively. The success of this project encourages further exploration and development of similar IoT-enabled solutions for other applications, contributing to smarter and more efficient management of resources in public spaces.

8. Work Breakdown Among Members

Wanigasinghe U. A. - 2020/E/200: Project Lead, Hardware development, Sensor Integration, and Code development for light control.

Uzair U.M. – 2020/E/201: Code development for visitor counting and LCD display integration.

Arachchi T.H.L. – 2020/E/202: System testing, documentation, and presentation preparation

9. References

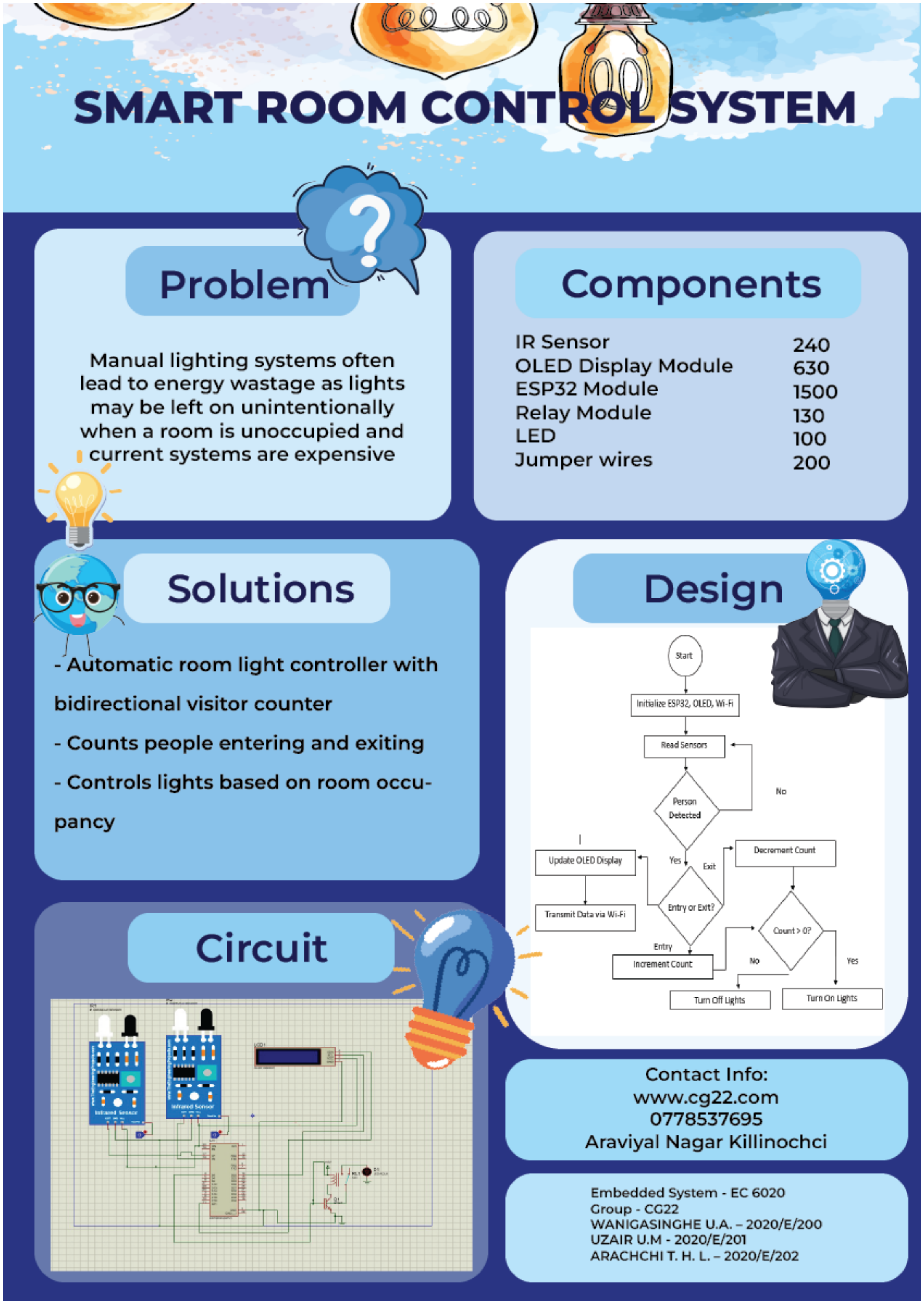
https://circuitdigest.com/microcontroller-projects/automatic-room-light-controller-with-bidirectional-visitor-counter-using-arduino

DIY Smart Lights: Build an Auto Room Lighting System (Easy Guide) (electronicshub.org)

(46) YouTube

10. Appendix

Poster



Arduino Code

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <BluetoothSerial.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1  // Reset pin not used with I2C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define IN_PIN 36
#define OUT_PIN 39
#define RELAY_PIN 2
#define LED_PIN 13  // Define LED pin

int count = 0;
bool lastStateIn = LOW;
bool lastStateOut = LOW;
unsigned long debounceDelay = 50;  // Adjust debounce delay as needed

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(9600);
  SerialBT.begin("ESP32_VC");

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    for (;;) ;  // Loop forever if the display allocation fails
  }

  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Visitor Counter");
  display.display();
  delay(1000);

  pinMode(IN_PIN, INPUT);
  pinMode(OUT_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);  // Initialize LED pin as output

  updateDisplay();
}
```

```
void loop() {
  checkPinState(IN_PIN, lastStateIn, true);
  checkPinState(OUT_PIN, lastStateOut, false);

  // Update LED and relay status based on count
  if (count > 0) {
    digitalWrite(RELAY_PIN, LOW);  // Turn on relay (assuming active low)
    digitalWrite(LED_PIN, HIGH);   // Turn on LED
  } else {
    digitalWrite(RELAY_PIN, HIGH); // Turn off relay (assuming active low)
    digitalWrite(LED_PIN, LOW);    // Turn off LED
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.println("Nobody In Room");
    display.setCursor(10, 32);
    display.println("Light Is Off");
    display.display();
  }

  if (SerialBT.available()) {
    char btChar = SerialBT.read();
    if (btChar == 'i') {
      incrementCount();
    } else if (btChar == 'd') {
      decrementCount();
    }
    sendBluetoothFeedback();  // Send the current count via Bluetooth
  }
}

void checkPinState(int pin, bool &lastState, bool isInPin) {
  bool currentState = digitalRead(pin);
  if (currentState != lastState) {
    delay(debounceDelay);
    currentState = digitalRead(pin);
    if (currentState != lastState) {
      lastState = currentState;
      if (currentState == HIGH) {
        if (isInPin) {
          incrementCount();
        } else {
          decrementCount();
        }
      }
    }
  }
```

```
}

void incrementCount() {
  count++;
  updateDisplay();
  sendBluetoothFeedback();  // Update count and send feedback
}

void decrementCount() {
  if (count > 0) {
    count--;
    updateDisplay();
    sendBluetoothFeedback();  // Update count and send feedback
  }
}

void updateDisplay() {
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Person In Room:");
  display.setCursor(60, 32);
  display.setTextSize(4);
  display.println(count);
  display.display();
}

void sendBluetoothFeedback() {
  SerialBT.print("Count: ");
  SerialBT.println(count);  // Send the current count via Bluetooth
}
```