Project Based Learning Report

on

Counter using Proximity Sensor in ESP 8266

Submitted in the partial fulfillment of the requirements
For the Project based learning in Embedded System Design
In branch
Electronics & Communication

By

PRN-2114110433 ANSHUMAN

PRN-2114110436 SAYYAM BAKSHI

PRN-2114110443 ANAGHA DIXIT

Under the guidance of Course In charge
Dr. Arundhati Shinde

Department of Electronic & Communication Engineering

Bharati Vidyapeeth,
(Deemed to be University)
College of Engineering, Pune-411043

Academic Year: - 2023-24

Bharati Vidyapeeth

(Deemed to be University)

College of Engineering,

Pune – 411043

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Project Based Learning report entitled, "Counter using proximity Sensor in ESP 8266" is a bonafide work done by

PRN-2114110433	**ANSHUMAN**
PRN-2114110436	**SAYYAM BAKSHI**
PRN-2114110443	**ANAGHA DIXIT**

in partial fulfillment of the requirement for the award of credits for Project Based Learning (PBL) in **"Embedded System Design",** of Bachelor of Technology semester V, in Electronics and Communication Engineering.

Date:

Dr. Arundhati Shinde	Dr. Arundhati Shinde
Course In-charge	Head of Department – ECE

INDEX

1.PROBLEM STATEMENT

Developing a reliable and accurate counter system using IR (Infrared) sensors integrated with the ESP8266 microcontroller to track and count objects passing through a defined area.

2. INTRODUCTION

How Does counter using IR sensor Work?

This project counts the number of obstacles that pass in front of the IR sensor in one direction only. The value of the total counts or the count number is displayed on a 16×2 LCD display module. The module has an emitter which is an IR LED and a detector which is an IR photodiode. The IR sensor that we are using in this project is an active IR sensor. Whenever it detects an object inside its range the output generated by it is high otherwise the output is low. You can also adjust the range of the sensor by rotating the inbuilt trim pot. The count is zero initially and then incremented by one whenever something passes in front of it. You can also make iot base obstacle detector.

3. Components Required

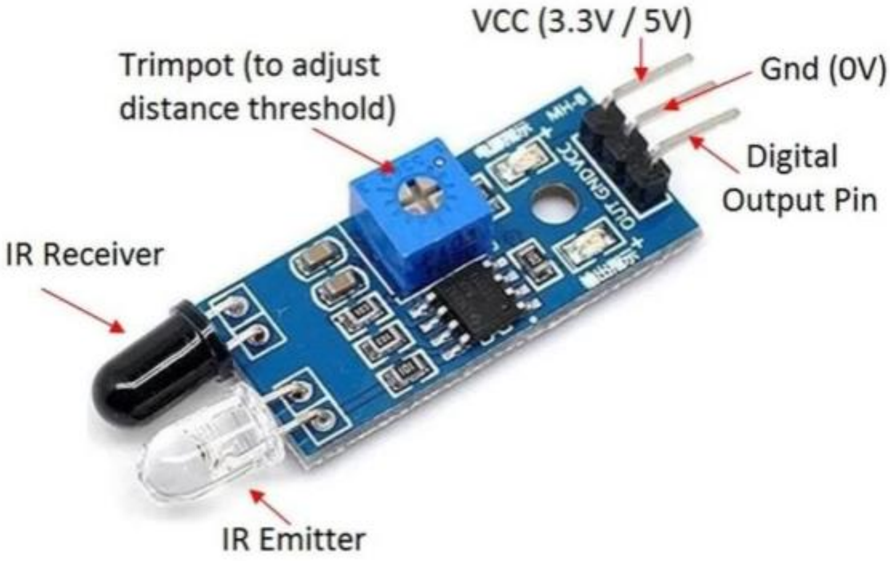
- Arduino UNO
- Led
- IR sensor
- Jumper wires
- Breadboard
- A Potentiometer
- USB cable for uploading the code

IR Sensor as Visitor Detector

The main element of this IoT project is the **IR Sensor** which works as a **Obstacle Detector**. Whenever the IR sensor detects an interrupt it counts the person and adds it to the previous value.

IR Sensor module has the great adaptive capability of the ambient light. It has an infrared transmitter and a receiver. The infrared emitting tube emits a certain frequency which when encounters an obstacle reflect back to the signal. The reflected signal is then received by the receiver tube. The other components in the circuit are Opamp, Variable Resistor & output LED.

The Sensor consists of the following electronics components.



1.IR LED Transmitter
IR LED emits light, in the range of Infrared frequency with a wavelength of 700nm – 1mm. IR LEDs have a light-emitting angle of approx. 20-60 degrees and has a range up to 5-10cm.

2. Photodiode Receiver
Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode looks like a LED, with a black color coating on its outer side.

3. LM358 Opamp
LM358 is an Operational Amplifier (Op-Amp) used as a voltage comparator in the IR sensor. The comparator circuit compares the threshold voltage set using the preset and the photodiode's series resistor voltage. When the Photodiode's series resistor voltage drop is greater than the threshold voltage, the Op-Amp output is high or vice-versa.
When the Op-Amp output is high the LED at the output terminal turns ON. This indicates an object detection.

4. Variable Resistor
The variable resistor here is preset. It is used to calibrate the distance range at which the object should be detected.

3. THEORY

ESP 8266
The ESP8266 is a popular and versatile Wi-Fi module and microcontroller platform used for various Internet of Things (IoT) applications. It was developed by Espressif Systems, a Chinese company, and has gained widespread popularity due to its low cost, ease of use, and a strong community of developers and enthusiasts. Here's some information about the ESP8266:
1. Microcontroller and Processor: The ESP8266 module is powered by the Tensilica L106 32-

bit microcontroller and features a low-power Xtensa LX106 core.

2. Wireless Connectivity: The primary feature of the ESP8266 is its built-in Wi-Fi connectivity, making it capable of connecting to Wi-Fi networks and communicating with other devices or services over the internet.

3. Memory: It comes with onboard flash memory for program storage, ranging from 512 KB to 16 MB depending on the variant. It also has RAM available for program execution and data storage.

4. GPIO Pins: The ESP8266 modules typically offer a number of general-purpose input/output (GPIO) pins, which can be used for digital input/output, PWM (Pulse Width Modulation), and more.

5. Development Environment: The ESP8266 can be programmed using a variety of programming languages and development environments, including Arduino IDE, PlatformIO, and the Espressif IoT Development Framework (ESP-IDF).

6. Programming Languages: You can program the ESP8266 using languages such as C, C++, Lua, and MicroPython. The Arduino IDE and PlatformIO offer a simplified way to program it using C/C++.

7. Community and Support: The ESP8266 has a large and active community of developers and users who provide tutorials, libraries, and support through forums and online communities.

8.Applications: The ESP8266 is widely used in IoT projects, home automation, sensor monitoring, smart appliances, and various other applications where Wi-Fi connectivity is required.

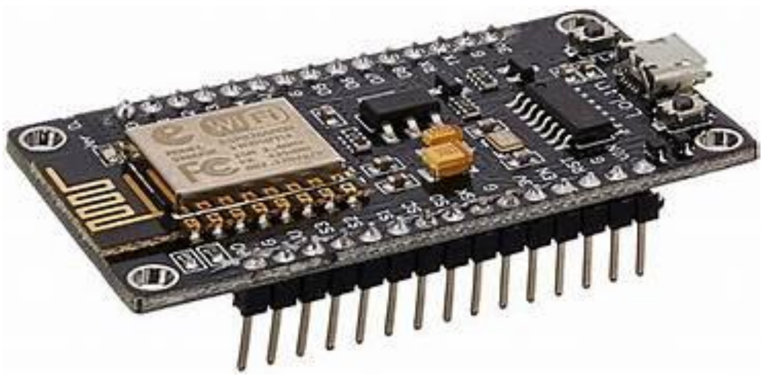
9.Variants: There are several variants of the ESP8266, with different pin configurations and features. The ESP-01, ESP-12E, and NodeMCU are some of the commonly used modules.

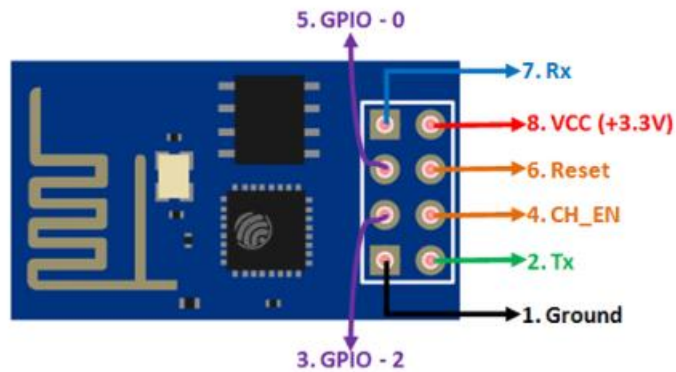
10.Integration with Other Devices: The ESP8266 can easily communicate with other microcontrollers, sensors, and devices, making it suitable for building complex IoT systems.

11. Low Cost: One of the key advantages of the ESP8266 is its affordability, making it accessible to hobbyists, students, and professionals working on IoT projects.

12. OTA Updates: It supports Over-The-Air (OTA) firmware updates, allowing you to remotely update the firmware on the device without the need for physical access.

When working with the ESP8266, it's important to understand the specific module variant you are using, its pinout, and the development environment you prefer to work with. The ESP8266 has been a game-changer in the world of IoT due to its capabilities, affordability, and ease of use, and it continues to be a popular choice for IoT enthusiasts and developers.





ESP8266 Pinout

ESP8266 Pin Configuration

Pin Number	Pin Name	Alternate Name	Normally used for	Alternate purpose
1	**Ground**	-	Connected to the ground of the circuit	-
2	**TX**	GPIO – 1	Connected to Rx pin of programmer/uC to upload program	Can act as a General purpose Input/output pin when not used as TX

3	**GPIO-2**	-	General purpose Input/output pin	-
4	**CH_EN**	-	Chip Enable – Active high	-
5	**GPIO - 0**	Flash	General purpose Input/output pin	Takes module into serial programming when held low during start up
6	**Reset**	-	Resets the module	-
7	**RX**	GPIO – 3	General purpose Input/output pin	Can act as a General purpose Input/output pin when not used as RX
8	**Vcc**	-	Connect to +3.3V only	

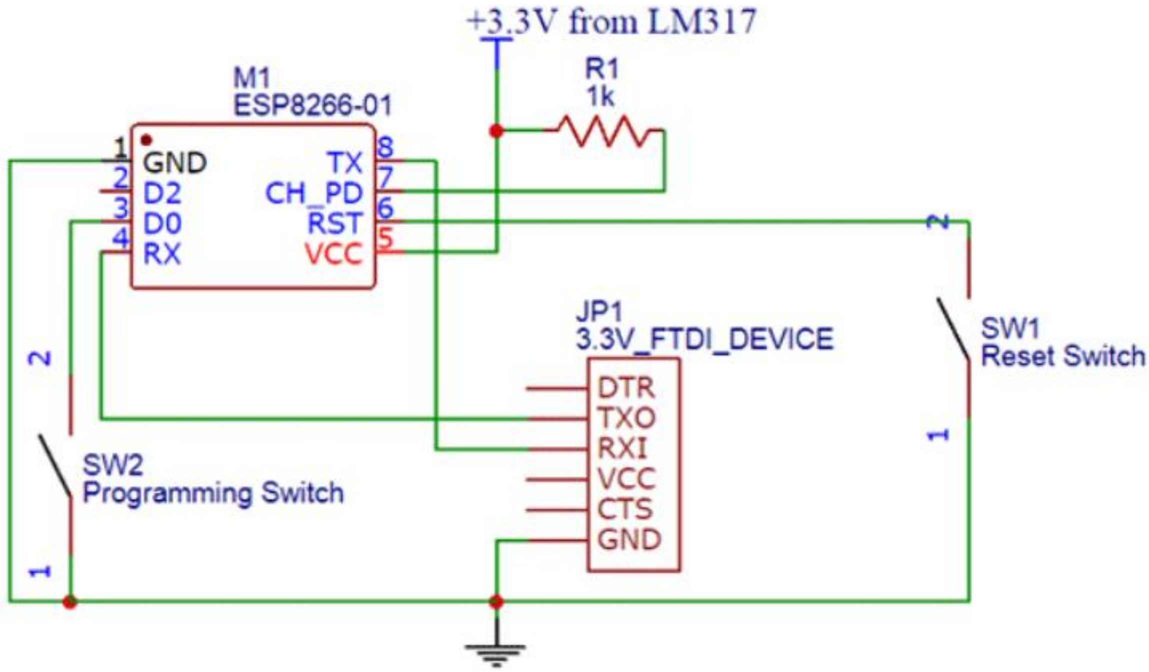
ESP8266 Features

- Low cost, compact and powerful Wi-Fi Module
- Power Supply: +3.3V only
- Current Consumption: 100mA
- I/O Voltage: 3.6V (max)
- I/O source current: 12mA (max)
- Built-in low power 32-bit MCU @ 80MHz
- 512kB Flash Memory
- Can be used as Station or Access Point or both combined
- Supports Deep sleep (<10uA)
- Supports serial communication hence compatible with many development platform like Arduino
- Can be programmed using Arduino IDE or AT-commands or Lua Script

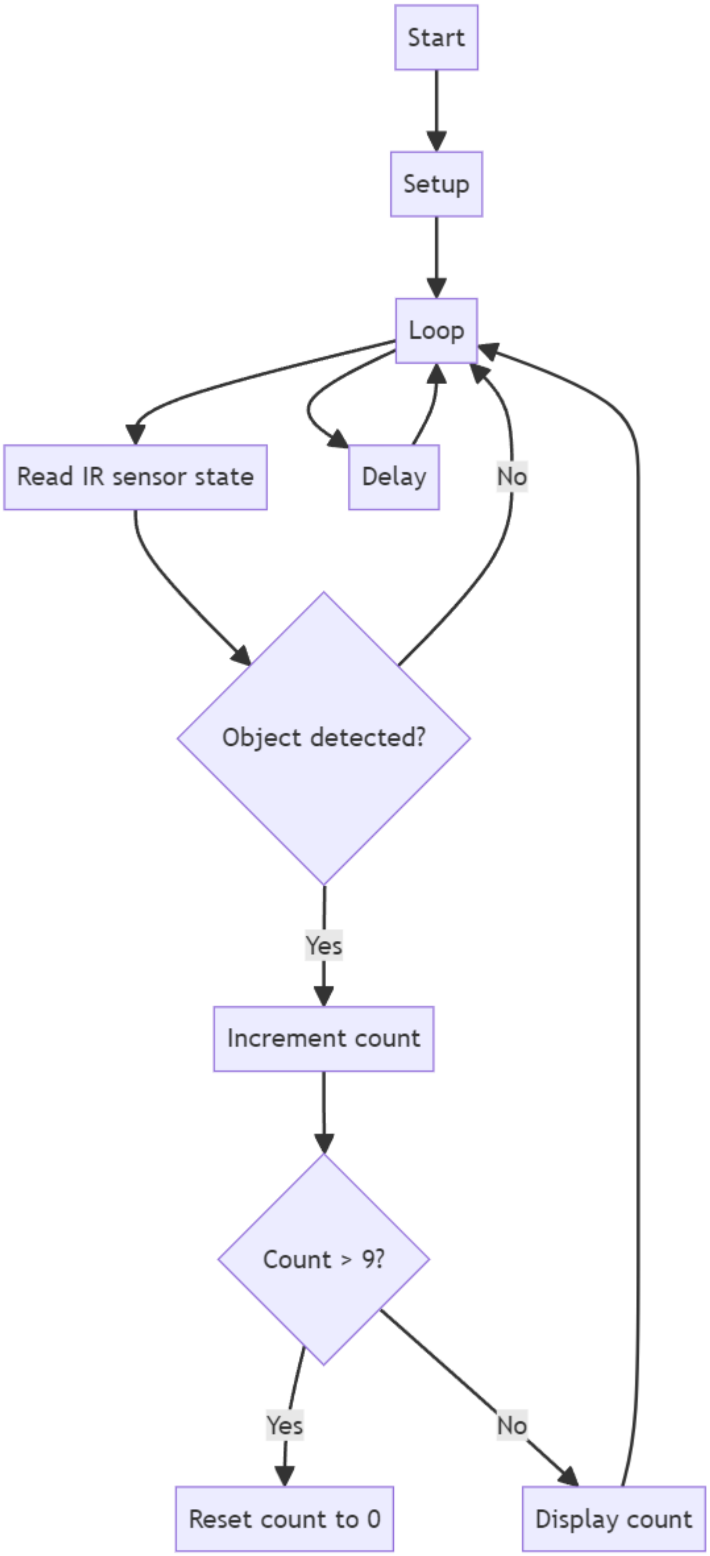
ESP8266 Boot Option

GPIO – 0	GPIO – 2	Mode	Used For
High	High	Flash Mode	Run the program that is already uploaded to the module
Low	High	UART Mode	Programming mode- to program using Arduino or any serial communication

Simplified circuit diagram for using the ESP8266 module



4. FLOWCHART



Start
Setup
Loop
Read IR sensor state
Delay
No
Object detected?
Yes
Increment count
Count > 9?
Yes
No
Reset count to 0
Display count

5. ALGORITHM

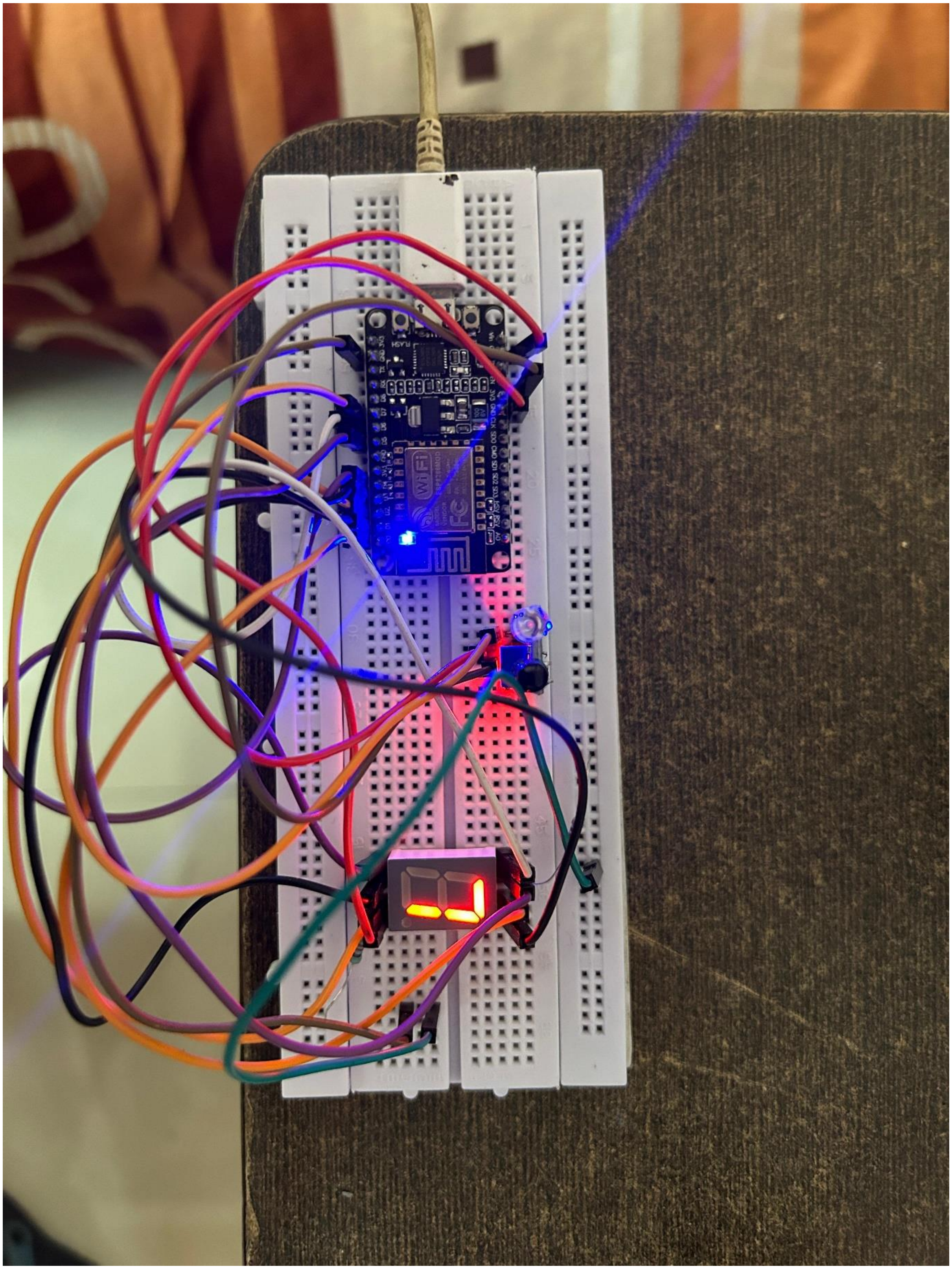
1. The given code is for an Arduino program that uses an IR sensor and a 7-segment display to count the number of objects detected.
2. The code starts by including the IRremote library, which provides functions for working with IR sensors. It then defines the pin numbers for the segments of the 7-segment display and the pin number for the IR sensor.
3. Next, the code defines a 2-dimensional array called `digitSegments` that represents the segments required to display each digit from 0 to 9 on the 7-segment display. Each digit is represented by an array of 7 elements, where a value of 1 indicates that the segment should be turned on and a value of 0 indicates that the segment should be turned off.
4. In the `setup()` function, the code sets the segment pins as outputs and initializes the IR sensor pin as an input. It also starts the serial communication at a baud rate of 115200.
5. The `loop()` function is where the main logic of the program resides. It first reads the state of the IR sensor using the `digitalRead()` function. If an object is detected (IR sensor state is LOW), the `count` variable is incremented and checked if it exceeds 9. If it does, the `count` is reset to 0. Then, the `displayCount()` function is called to update the 7-segment display with the new count.
6. After updating the display, there is a delay of 1 second using the `delay()` function to make the count visible for a short time.
7. The `displayDigit()` function takes a digit as input and updates the segment pins of the 7-segment display based on the `digitSegments` array.
8. The `displayCount()` function simply calls the `displayDigit()` function with the current count as the input.
9. Overall, the code continuously reads the state of the IR sensor and increments the count whenever an object is detected. It then updates the 7-segment display with the new count.

6. SOURCE CODE

```
#include <IRremote.h>

const int segmentPins[] = {D0, D1, D2, D3, D4, D5, D6}; // Pins D0 to D6 for segments 'a' to 'g'
const int IR_RECEIVER_PIN = D7; // IR sensor OUT pin connected to D7 on ESP8266

// Define the segments for each digit (common cathode)
const byte digitSegments[10][7] = {
  {1, 1, 1, 1, 1, 1, 0}, // 0
  {0, 1, 1, 0, 0, 0, 0}, // 1
  {1, 1, 0, 1, 1, 0, 1}, // 2
  {1, 1, 1, 1, 0, 0, 1}, // 3
  {0, 1, 1, 0, 0, 1, 1}, // 4
  {1, 0, 1, 1, 0, 1, 1}, // 5
  {1, 0, 1, 1, 1, 1, 1}, // 6
  {1, 1, 1, 0, 0, 0, 0}, // 7
  {1, 1, 1, 1, 1, 1, 1}, // 8
  {1, 1, 1, 1, 0, 1, 1}  // 9
};
```

```
int count = 0;
void setup() {
  for (int i = 0; i < 7; i++) {
    pinMode(segmentPins[i], OUTPUT);
  }

  // Initialize the IR sensor pin
  pinMode(IR_RECEIVER_PIN, INPUT);

  Serial.begin(115200);
}

void loop() {
  // Read the state of the IR sensor
  int irSensorState = digitalRead(IR_RECEIVER_PIN);

  if (irSensorState == LOW) {
    // Object detected
    count++;

    if (count > 9) {
      count = 0;
    }

    displayCount();
  }

  // Delay to make the count visible for a short time
  delay(1000);
}

void displayDigit(int digit) {
  if (digit >= 0 && digit <= 9) {
    for (int i = 0; i < 7; i++) {
      digitalWrite(segmentPins[i], digitSegments[digit][i]);
    }
  }
}

void displayCount() {
  // Display the count on the 7-segment display
  displayDigit(count);
}
```

7. CIRCUIT DIAGRAM



8. CONCLUSION

Using IR sensors with the ESP8266 microcontroller offers a versatile and cost-effective solution for object counting. It's highly adaptable, provides connectivity for data transmission, and is affordable. However, ensuring accuracy and addressing placement challenges are important considerations. Overall, it's a valuable tool for various IoT and automation applications.

9.COURSE OUTCOME

CO6 – Select & use the appropriate ESP module for real world application

Thus, Course outcome 6 is satisfied.

CO3- Identify the features and architecture of the ESP modules

Thus, Course outcome 3 is satisfied.

10. REFERENCES

1. **Online Documentation and Tutorials:**
   Platforms like Adafruit, SparkFun, and GitHub often provide detailed tutorials and code examples for integrating IR sensors with ESP8266. You can search for specific projects or guides related to IR sensor counters.

2. **Online Forums and Communities:**
   Websites like Arduino Forum, ESP8266 Community Forum, and Stack Overflow (in the Arduino and ESP8266 tags) are excellent resources. Users often share their projects and solutions, providing valuable insights and code snippets.

3. **YouTube Tutorials:**
   YouTube hosts numerous tutorials related to ESP8266 and IR sensors. You can find step-by-step guides and demonstrations by searching for keywords like "ESP8266 IR sensor counter tutorial."

4. **Books and Online Courses:**
   Books on IoT and microcontroller programming often cover projects involving sensors and ESP8266. Websites like Udemy, Coursera, and Amazon offer books and courses that might include relevant information.

5. **Datasheets and Manufacturer Documentation:**
   Refer to the datasheets and technical documentation provided by manufacturers of IR sensors and ESP8266 modules. These documents offer in-depth technical insights into the components' functionalities and usage.

6. **GitHub Repositories:**
   GitHub hosts open-source projects. Searching for repositories related to "ESP8266 IR sensor counter" or similar keywords might lead you to relevant codebases and examples.