Agriculture Monitoring System Using ThingSpeak

Introduction

Agriculture Monitoring System using ThingSpeak is designed to provide farmers with real-time data on various environmental parameters affecting crop growth.everaging the capabilities of ThingSpeak, an Internet of Things (IoT) platform, this system collects, processes, and visualizes data from various sensors deployed in the field.

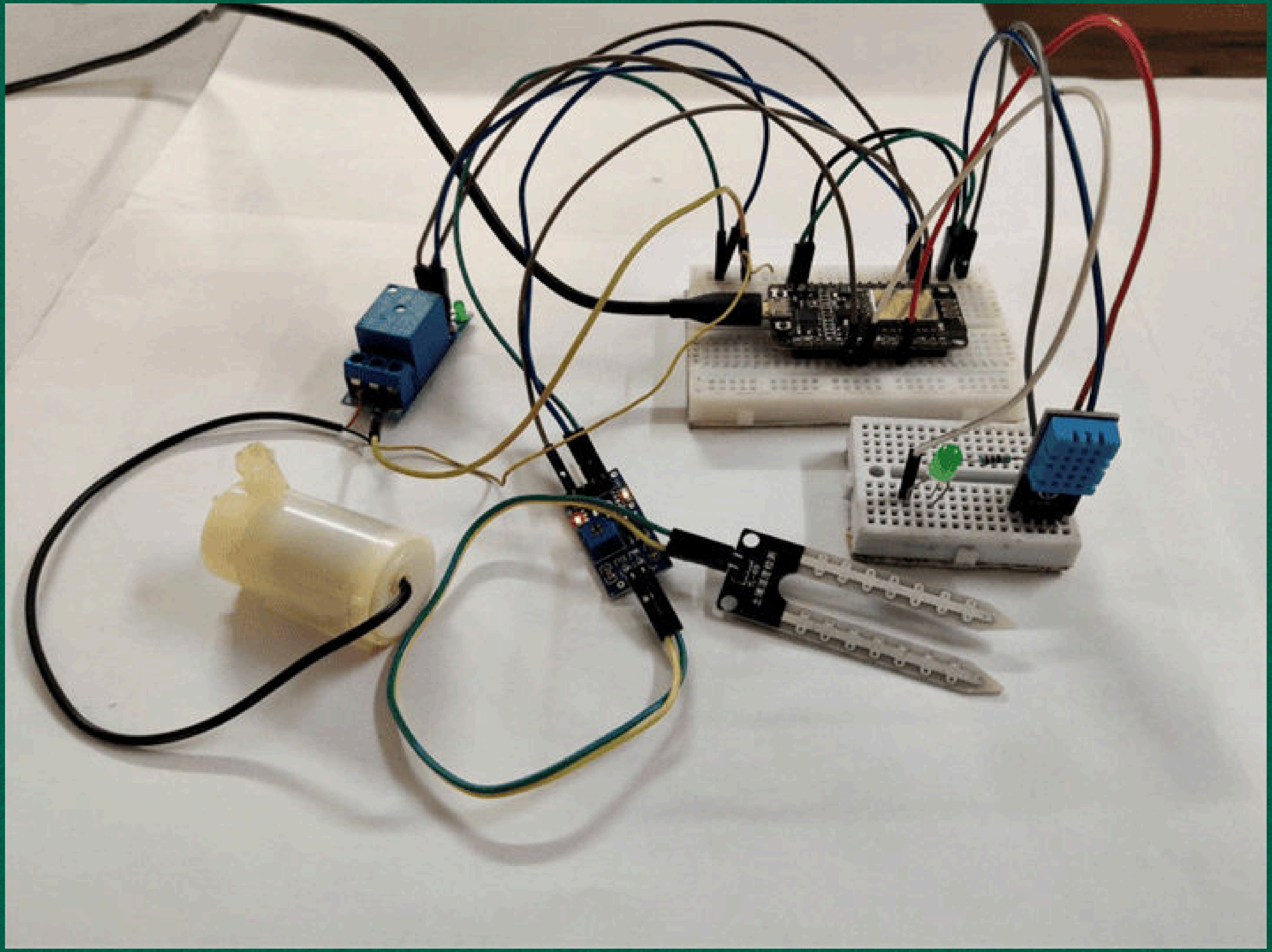
Aim

The primary objective of this project is to enhance agricultural practices by monitoring critical environmental factors such as soil moisture, temperature, and humidity. The integration with ThingSpeak enables farmers to access and analyze data remotely, facilitating informed decision-making for crop management.

Components Required

- Soil Moisture sensors
- Temperature and humidity sensors
- Microcontroller ESP32
- Internet connectivity module
- ThingSpeak IoT platform account
- Jumper wires
- Power supply

Circuit Diagram



Applications

1. Deploy an Agriculture Monitoring System with ThingSpeak for real-time data collection, analysis, and visualization, optimizing crop health and resource utilization
2. Leverage ThingSpeak to create a robust Agriculture Monitoring System, providing farmers with actionable insights based on sensor data for efficient crop management and yield optimization.

Conclusion

The Agriculture Monitoring System using ThingSpeak demonstrates the potential of IoT in agriculture. By providing farmers with real-time data on crucial environmental parameters, this system empowers them to make informed decisions, ultimately optimizing crop yield and resource utilization.

Thanks!

Department of ECE

COURSE:ELECRTONIC SYSTEM AUTOMATION

COURSE CODE:22SDEC02

PROJECT TITLE :BLUETOOTH BASED HOME AUTOMATION

2200040330 – SHASHANK
2200040315 – Mounika
2200040319 - Nagaraju

AIM OF THE PROJECT

Understanding how to interface Android app for sending commands or controlling functions on the ESP32 .

Creating a Bluetooth-based home automation project using the ESP32 microcontroller can be a rewarding and educational experience.

PROJECT OBJECTIVES

1. **Bluetooth Connectivity:**
Establish a reliable Bluetooth connection between the ESP32 and a mobile device (smartphone, tablet, etc.).

2. **Device Detection and Pairing:**
Implement a device discovery mechanism to detect and pair with nearby Bluetooth-enabled devices.

3. **Mobile App Interface:**
Create a user-friendly mobile application for controlling home automation devices.
Design an intuitive interface to monitor and control various devices.

4. **Home Appliance Control:**
Enable control of different home appliances such as lights, fans, and electronic devices through Bluetooth commands.

PROJECT OUTCOMES

1. **Smart Lighting Control:**
   Turn lights on/off.
   Adjust brightness levels.

2. **Temperature and Climate Control:**
   Monitor and control the temperature using sensors.
   Turn on/off heating or cooling devices based on predefined thresholds.

3. **Power Management:**
   Control and monitor power outlets.
   Schedule devices to turn on/off.

4. **Voice Control:**
   Integrate with voice assistants (like Amazon Alexa or Google Assistant) for hands-free control.

5. **Energy Efficiency:**
   Implement features to optimize energy consumption.

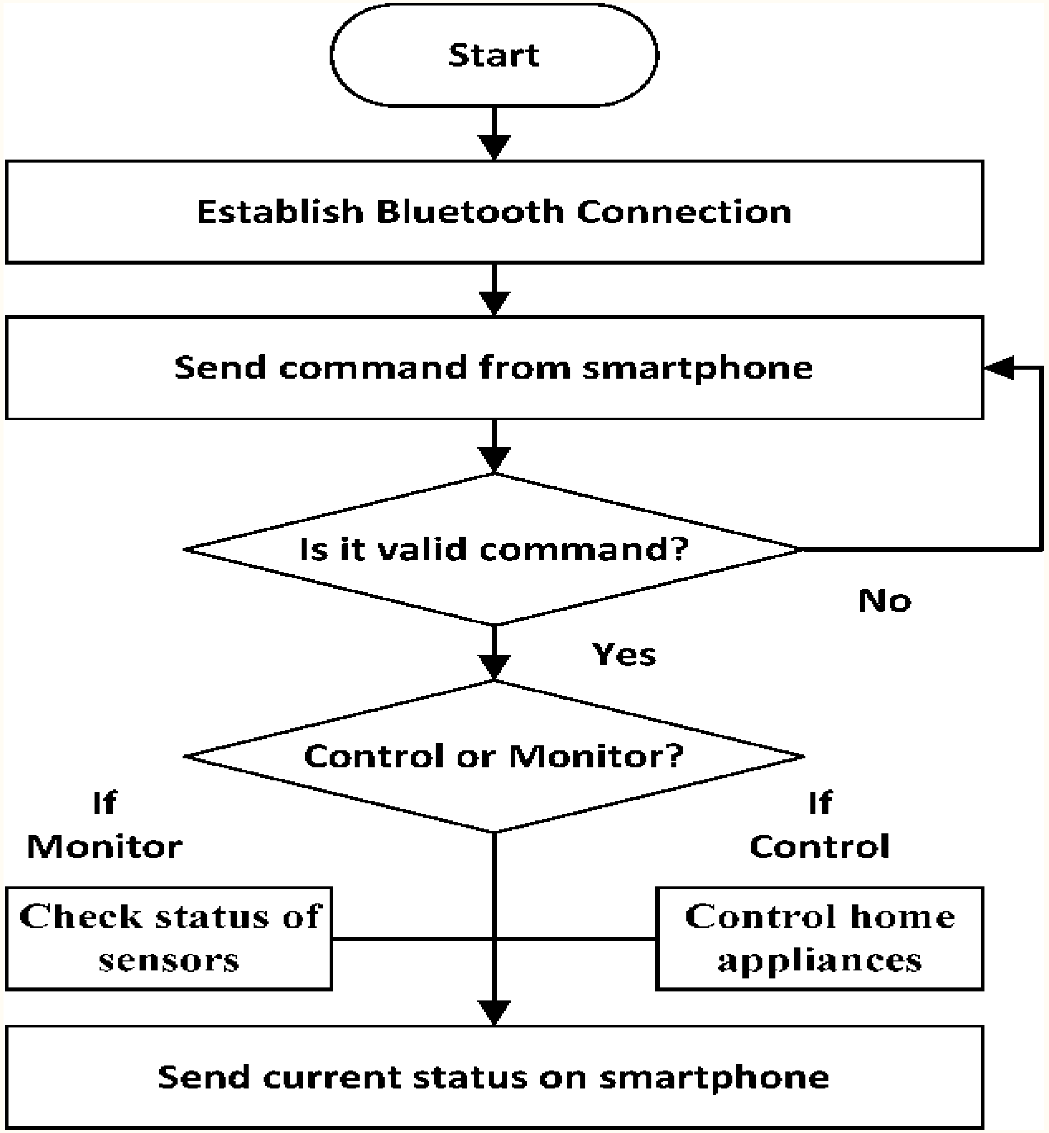
PROJECT INTRODUCTION

In the era of smart homes and interconnected devices, home automation has become a ubiquitous trend, offering convenience, energy efficiency, and enhanced security. Bluetooth technology has emerged as a versatile and reliable solution for short-range wireless communication, making it an ideal choice for creating seamless and responsive home automation systems.

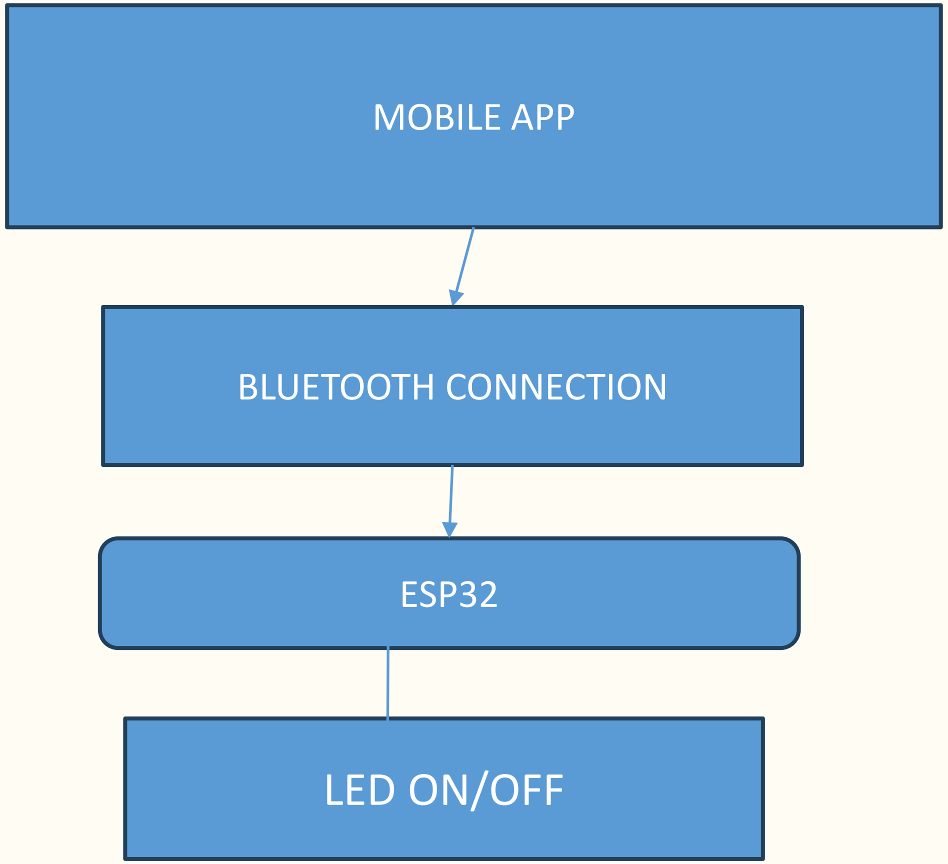
PROJECT DESCRIPTION

Experience the future of smart living with our Bluetooth-based Home Automation using ESP32 project. This cutting-edge technology empowers you to control and monitor various household appliances and devices seamlessly through your smartphone or tablet. Leveraging the ESP32's capabilities along with Bluetooth technology, this project empowers users to seamlessly manage their home environment with the convenience of their smartphones or tablets

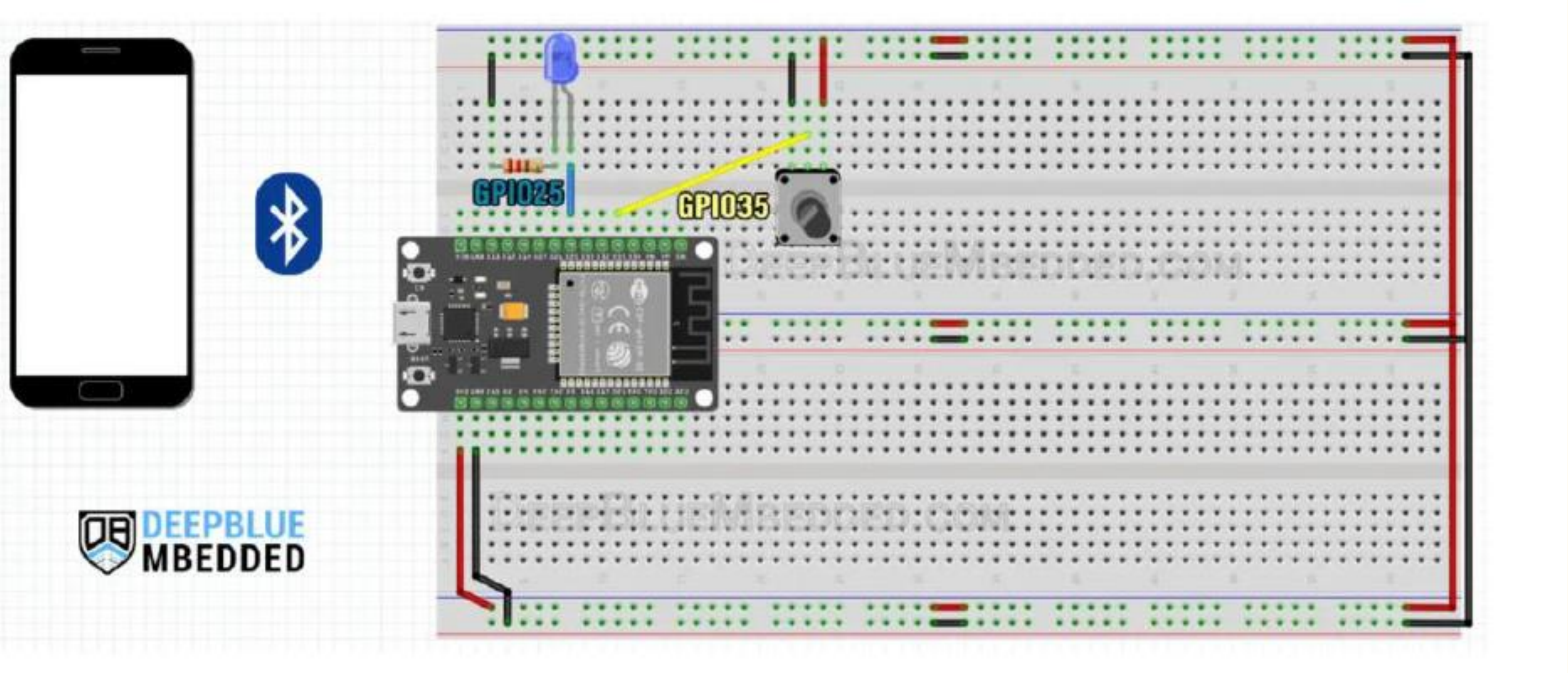
FLOW CHART


Start
Establish Bluetooth Connection
Send command from smartphone
Is it valid command?
No
Yes
Control or Monitor?
If Monitor
If Control
Check status of sensors
Control home appliances
Send current status on smartphone

BLOCK DIAGRAM



CIRCUIT DIAGRAM



HARDWARE & SOFTWARE REQUIREMENTS

Hardware Requirements:

1.ESP32 Development Board

2.CONNECTING WIRES

3.1MOBILE PHONE

4.Breadboard and Jumper Wires

5.LED

6.RESISTORS

Software Requirements:

1.Arduino IDE

2.ESP32 Board Support Package for Arduino IDE

3.USB Drivers

4.Serial Monitor Software

FUTURE SCOPE

A Bluetooth-based home automation project using ESP32 has significant potential for future development and expansion. Here are some future scope ideas that you can consider for enhancing your project:

1.Integration with Voice Assistants:

1. Implement voice control using popular voice assistants like Amazon Alexa or Google Assistant. This will allow users to control their home automation system through voice commands.

2.Mobile Application Development:

1. Develop a dedicated mobile application for your home automation system. This app can provide a user-friendly interface for controlling devices, monitoring the home, and receiving notifications.

3.Security Features:

1. Enhance the security aspects of your home automation system. Implement features such as door/window sensors, motion detectors, and surveillance cameras. Integrate the system with security protocols to ensure a secure environment.

APPLICATIONS

Bluetooth-based home automation using ESP32 microcontrollers opens up a wide range of possibilities for controlling various home devices and systems. Here are some application ideas for your project:

1. **Smart Lighting Control**: Create an application to control the brightness, color, and on/off status of LED lights or smart bulbs using Bluetooth. Users can adjust the lighting ambiance according to their preferences using a smartphone or tablet.
2. **Temperature and Humidity Monitoring**: Develop a system that measures temperature and humidity levels in different rooms of the house using sensors connected to ESP32 modules. Users can monitor and receive alerts about any abnormal conditions through a Bluetooth-enabled mobile app.
3. **Smart Door Locks**: Design a Bluetooth-controlled smart door lock system using ESP32 modules and servo motors. Users can lock/unlock doors remotely using their smartphones, grant access to guests or family members, and receive activity logs of door access.
4. **Automated Irrigation System**: Implement an automated irrigation system for indoor or outdoor plants using ESP32 modules connected to water pumps and moisture sensors. Users can schedule watering times, monitor soil moisture levels, and adjust irrigation settings using a Bluetooth-enabled mobile app.

Department of ECE

COURSE:ELECRTONIC SYSTEM AUTOMATION
COURSE CODE:22SDEC02R

PROJECT TITLE :Interrupt request based audio input from I2S microphone

2200040330 – SHASHANK
2200040315 – MOUNIKA
2200040319 - NAGARAJU

AIM OF THE PROJECT

The project entails capturing audio input from an I2S microphone using interrupt requests. This involves configuring both hardware and software components to efficiently handle incoming audio data triggered by interrupt requests from the microphone. The system's software must be capable of initializing the I2S interface, managing interrupt handling routines, processing the audio data, and integrating it into the broader system as needed. This approach minimizes latency and optimizes the handling of audio data for real-time applications or other processing requirements.

PROJECT OBJECTIVES

The project objectives are to:

1. Capture audio input from an I2S microphone.
2. Utilize interrupt requests for efficient handling of incoming audio data.
3. Develop software routines to process and integrate the audio data into the system.
4. Minimize latency and optimize audio data handling for real-time applications or processing needs.

PROJECT OUTCOMES

The project outcomes include:

1. Successful integration of an I2S microphone for audio input.
2. Implementation of interrupt-driven handling for efficient data processing.
3. Functional software routines for capturing, processing, and integrating audio data.
4. Improved latency management, enabling real-time applications or processing tasks.

PROJECT INTRODUCTION

The project introduces an efficient audio input system using an I2S microphone with interrupt-based communication. It aims to capture audio data promptly, minimize latency, and optimize system resources for real-time audio processing tasks. The project involves integrating the I2S microphone, configuring hardware interfaces, implementing interrupt handling mechanisms, and designing software algorithms for audio data processing.

ABSTRACT

Abstract:

This project explores the implementation of an interrupt-driven audio input system utilizing an I2S (Inter-IC Sound) microphone. The system aims to capture audio data efficiently, minimize latency, and optimize resources for real-time processing applications. By leveraging interrupt requests, the system can respond promptly to the microphone's output, ensuring timely data acquisition. The project involves configuring hardware interfaces, developing software routines for interrupt handling and audio processing, and integrating the components into a cohesive system. Through this endeavor, the project showcases the effectiveness of interrupt-based communication in enhancing audio input performance for various embedded systems and digital signal processing tasks.

HARDWARE & SOFTWARE REQUIREMENTS

Hardware Requirements:

1.ESP32 Development Board

2.INMP441 Microphone Module

3.CONNECTING WIRES

4.Breadboard and Jumper Wires

Software Requirements:

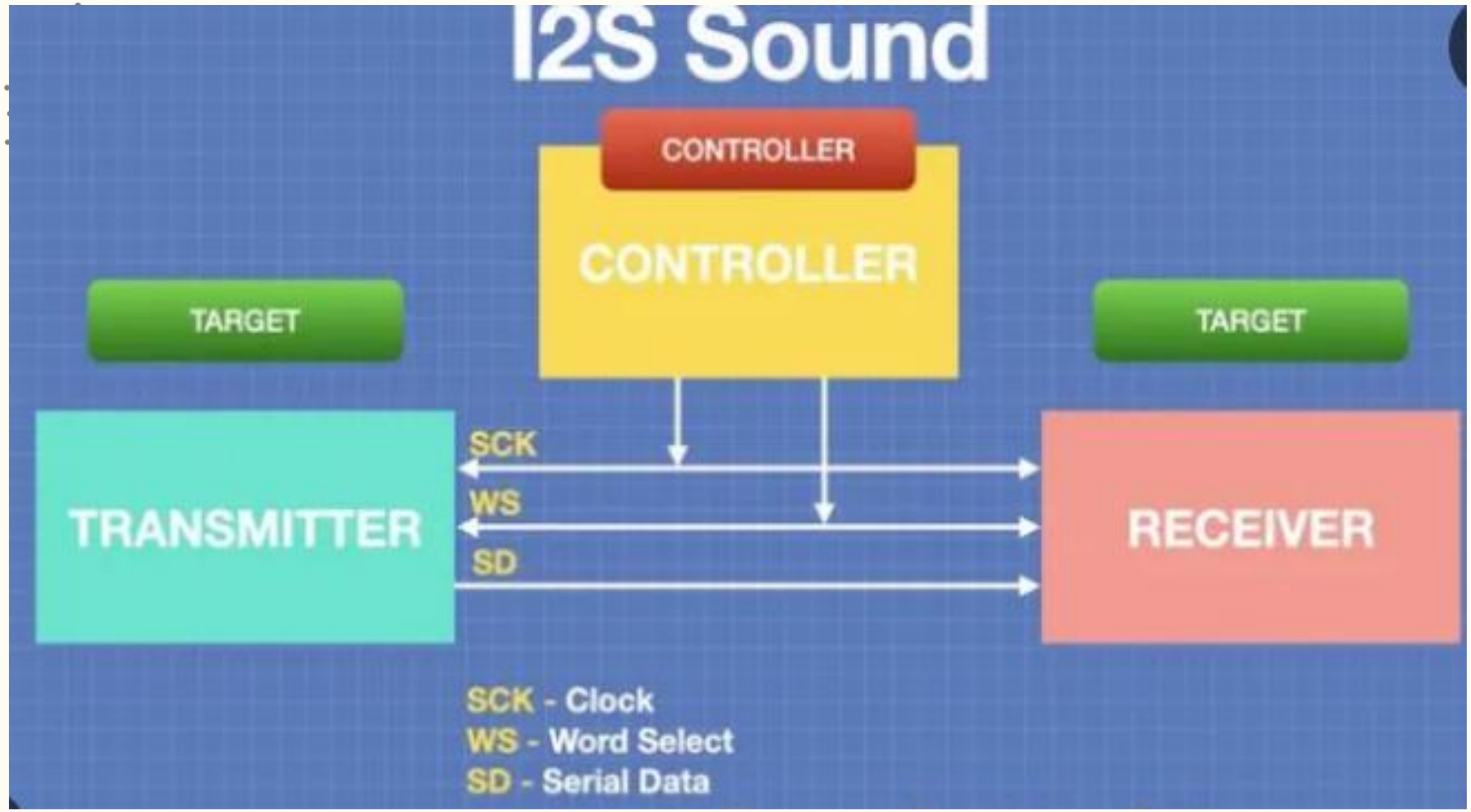
1.Arduino IDE

2.ESP32 Board Support Package for Arduino IDE

3.USB Drivers

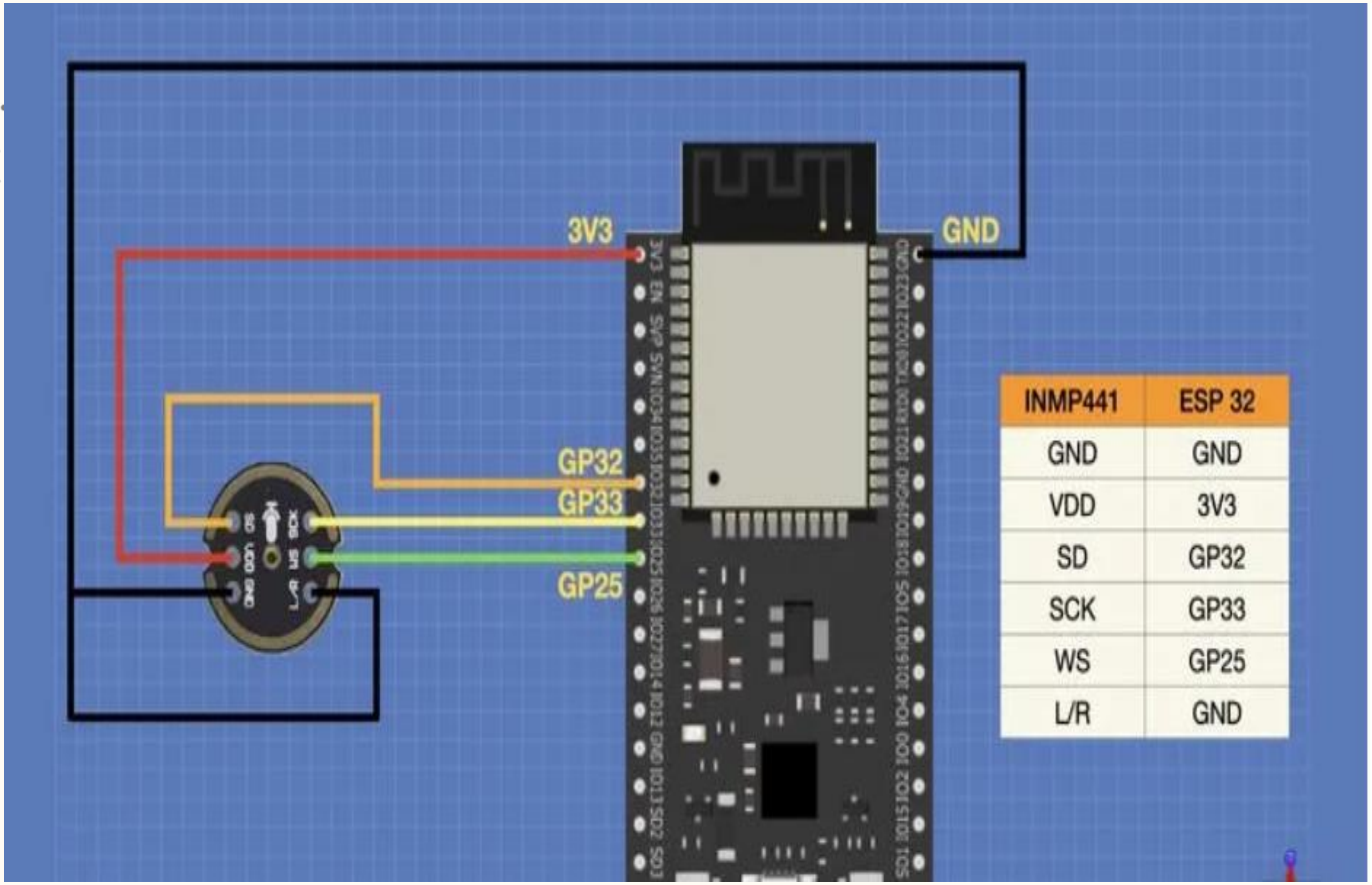
4.Serial Monitor Software

BLOCK DIAGRAM



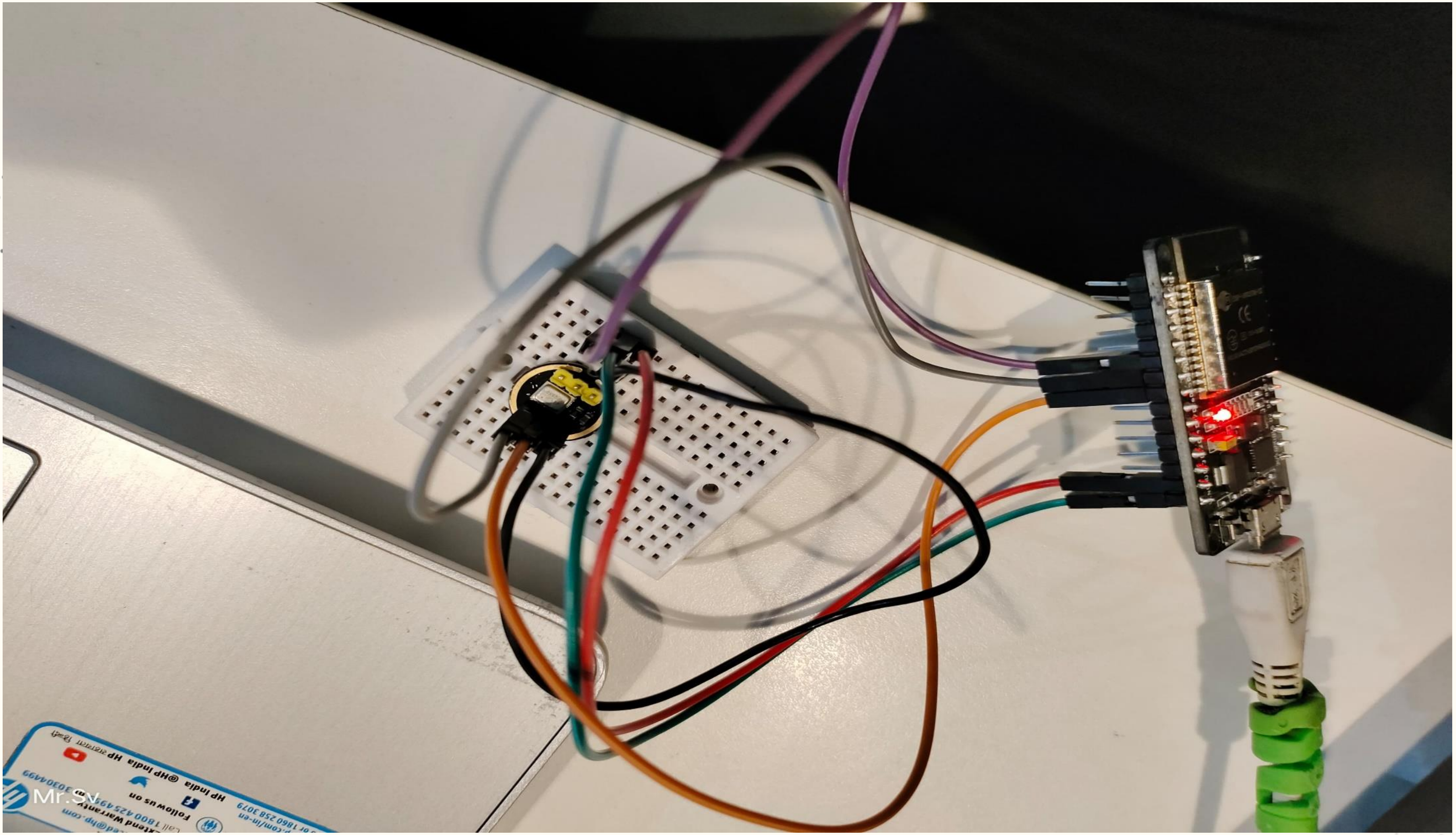
SCK - Clock
WS - Word Select
SD - Serial Data

CIRCUIT DIAGRAM

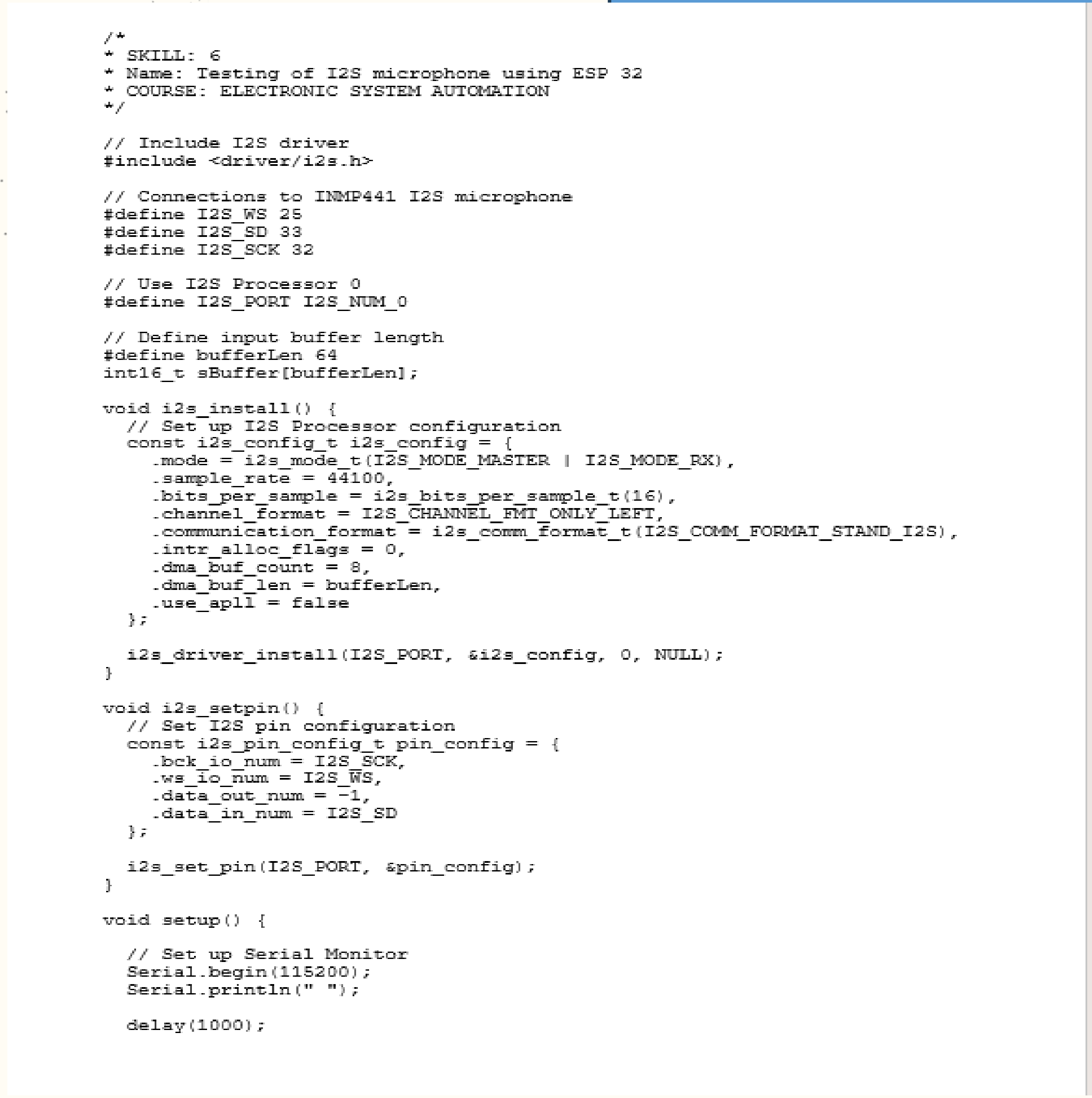


INMP441	ESP 32
GND	GND
VDD	3V3
SD	GP32
SCK	GP33
WS	GP25
L/R	GND

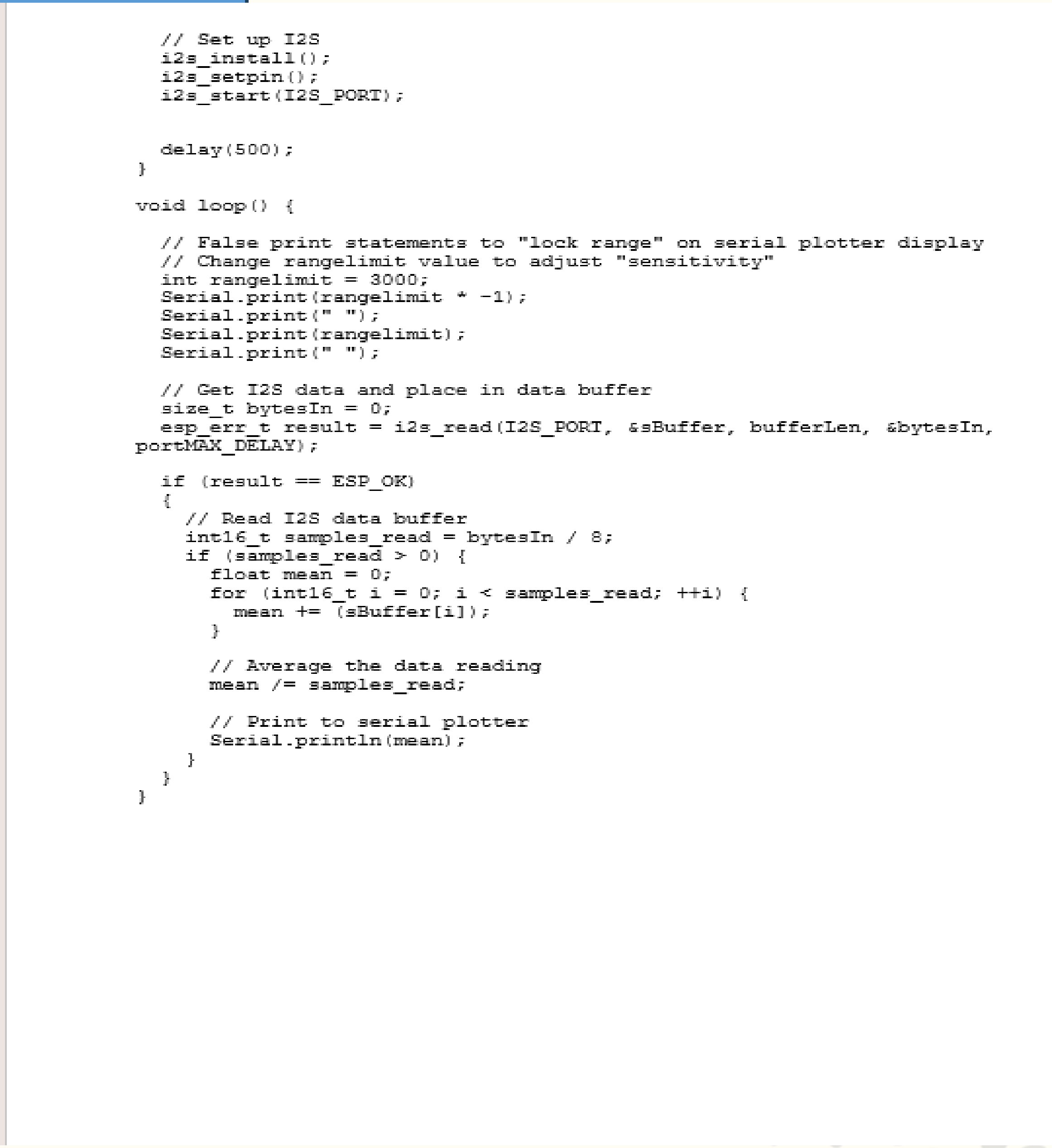
HARDWARE IMPLEMENTAION



CODE

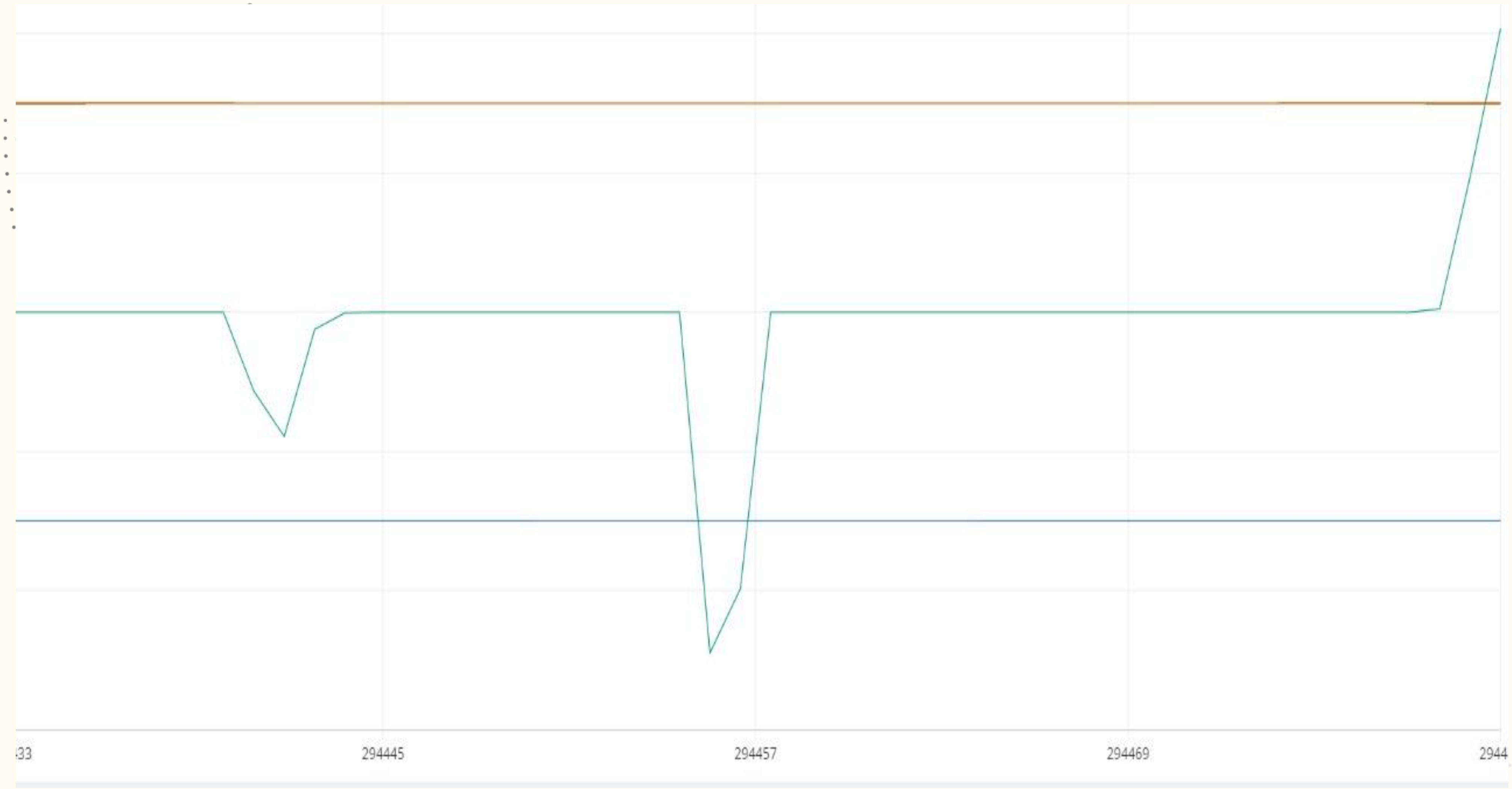


```
/*
* SKILL: 6
* Name: Testing of I2S microphone using ESP 32
* COURSE: ELECTRONIC SYSTEM AUTOMATION
*/

// Include I2S driver
#include <driver/i2s.h>

// Connections to INMP441 I2S microphone
#define I2S_WS 25
#define I2S_SD 33
#define I2S_SCK 32

// Use I2S Processor 0
#define I2S_PORT I2S_NUM_0

// Define input buffer length
#define bufferLen 64
int16_t sBuffer[bufferLen];

void i2s_install() {
  // Set up I2S Processor configuration
  const i2s_config_t i2s_config = {
    .mode = i2s_mode_t(I2S_MODE_MASTER | I2S_MODE_RX),
    .sample_rate = 44100,
    .bits_per_sample = i2s_bits_per_sample_t(16),
    .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
    .communication_format = i2s_comm_format_t(I2S_COMM_FORMAT_STAND_I2S),
    .intr_alloc_flags = 0,
    .dma_buf_count = 8,
    .dma_buf_len = bufferLen,
    .use_apll = false
  };

  i2s_driver_install(I2S_PORT, &i2s_config, 0, NULL);
}

void i2s_setpin() {
  // Set I2S pin configuration
  const i2s_pin_config_t pin_config = {
    .bck_io_num = I2S_SCK,
    .ws_io_num = I2S_WS,
    .data_out_num = -1,
    .data_in_num = I2S_SD
  };

  i2s_set_pin(I2S_PORT, &pin_config);
}

void setup() {

  // Set up Serial Monitor
  Serial.begin(115200);
  Serial.println(" ");

  delay(1000);
```



```
  // Set up I2S
  i2s_install();
  i2s_setpin();
  i2s_start(I2S_PORT);


  delay(500);
}

void loop() {

  // False print statements to "lock range" on serial plotter display
  // Change rangelimit value to adjust "sensitivity"
  int rangelimit = 3000;
  Serial.print(rangelimit * -1);
  Serial.print(" ");
  Serial.print(rangelimit);
  Serial.print(" ");

  // Get I2S data and place in data buffer
  size_t bytesIn = 0;
  esp_err_t result = i2s_read(I2S_PORT, &sBuffer, bufferLen, &bytesIn,
portMAX_DELAY);

  if (result == ESP_OK)
  {
    // Read I2S data buffer
    int16_t samples_read = bytesIn / 8;
    if (samples_read > 0) {
      float mean = 0;
      for (int16_t i = 0; i < samples_read; ++i) {
        mean += (sBuffer[i]);
      }

      // Average the data reading
      mean /= samples_read;

      // Print to serial plotter
      Serial.println(mean);
    }
  }
}
```

RESULTS



FUTURE SCOPE

The future scope of interrupt request-based audio input from I2S microphones includes advancements in real-time processing, integration with AI and machine learning, energy-efficient designs, IoT and embedded systems applications, multi-microphone array systems, and adaptive audio processing techniques. These developments promise enhanced performance, increased functionality, and broader application domains for interrupt-driven audio input technology.

APPLICATIONS

Interrupt request-based audio input from I2S microphones has several applications across various domains:

1.Voice Recognition Systems: I2S microphones with interrupt-driven audio input are crucial components in voice recognition systems used in smart assistants, automotive voice control, and home automation devices.

2.Speech Processing and Analysis: These systems enable real-time speech processing and analysis for applications such as speech-to-text conversion, emotion detection, and speaker recognition.

3.Telecommunications: Interrupt-driven audio input is used in telecommunication systems for tasks like echo cancellation, noise reduction, and audio conferencing, ensuring clear and high-quality voice communication.

4.Audio Recording and Playback Devices: I2S microphones with interrupt-based communication are integrated into audio recording devices, smartphones, and music players for capturing and playing back audio content.

5.Surveillance and Security Systems: Interrupt-driven audio input is employed in surveillance cameras and security systems to detect and analyze sounds for intruder detection, gunshot recognition, and abnormal event detection.

Department of ECE

COURSE:ELECRTONIC SYSTEM AUTOMATION

COURSE CODE:22SDEC02A

PROJECT TITLE : Interrupt request based posting of RFID data to Google spreadsheet

SHASHANK – 2200040330
MOUNIKA – 2200040315
NAGARAJU – 2200040319

AIM OF THE PROJECT

The project aims to enable the automatic posting of RFID data to a Google Spreadsheet using interrupt requests. This involves setting up an RFID reader to capture data, configuring a microcontroller or computer system to handle interrupts triggered by new RFID data, integrating with the Google Sheets API to send data to the spreadsheet, and testing the system for accuracy and reliability.

PROJECT OBJECTIVES

1. **Automated Data Capture**: Implement a system capable of automatically capturing RFID data from RFID tags.

2. **Interrupt-driven Processing**: Develop mechanisms for the system to process RFID data based on interrupt requests, ensuring efficient handling of incoming data.

3. **Integration with Google Spreadsheet**: Integrate the system with the Google Sheets API to facilitate the seamless transfer of RFID data to a designated spreadsheet.

4. **Real-time Data Posting**: Enable real-time posting of RFID data to the Google Spreadsheet, ensuring timely and accurate updates.

5. **Reliability and Accuracy**: Ensure that the system operates reliably and accurately, handling data transmission and processing without errors or interruptions.

PROJECT OUTCOMES

1. **Automated Data Management**: Achieve streamlined management of RFID data by automating the process of capturing and posting information to the Google Spreadsheet.

2. **Improved Efficiency**: Increase operational efficiency through the use of interrupt-driven processing, reducing delays and enhancing responsiveness to RFID data events.

3. **Real-time Data Updates**: Enable real-time updates of RFID data on the Google Spreadsheet, ensuring that users have access to the latest information without manual intervention.

4. **Enhanced Accessibility**: Facilitate easier access to RFID data for analysis, reporting, and decision-making by centralizing it within a Google Spreadsheet accessible to authorized users.

5. **Reliability and Scalability**: Establish a reliable and scalable system architecture capable of handling varying volumes of RFID data while maintaining consistent performance and data integrity.

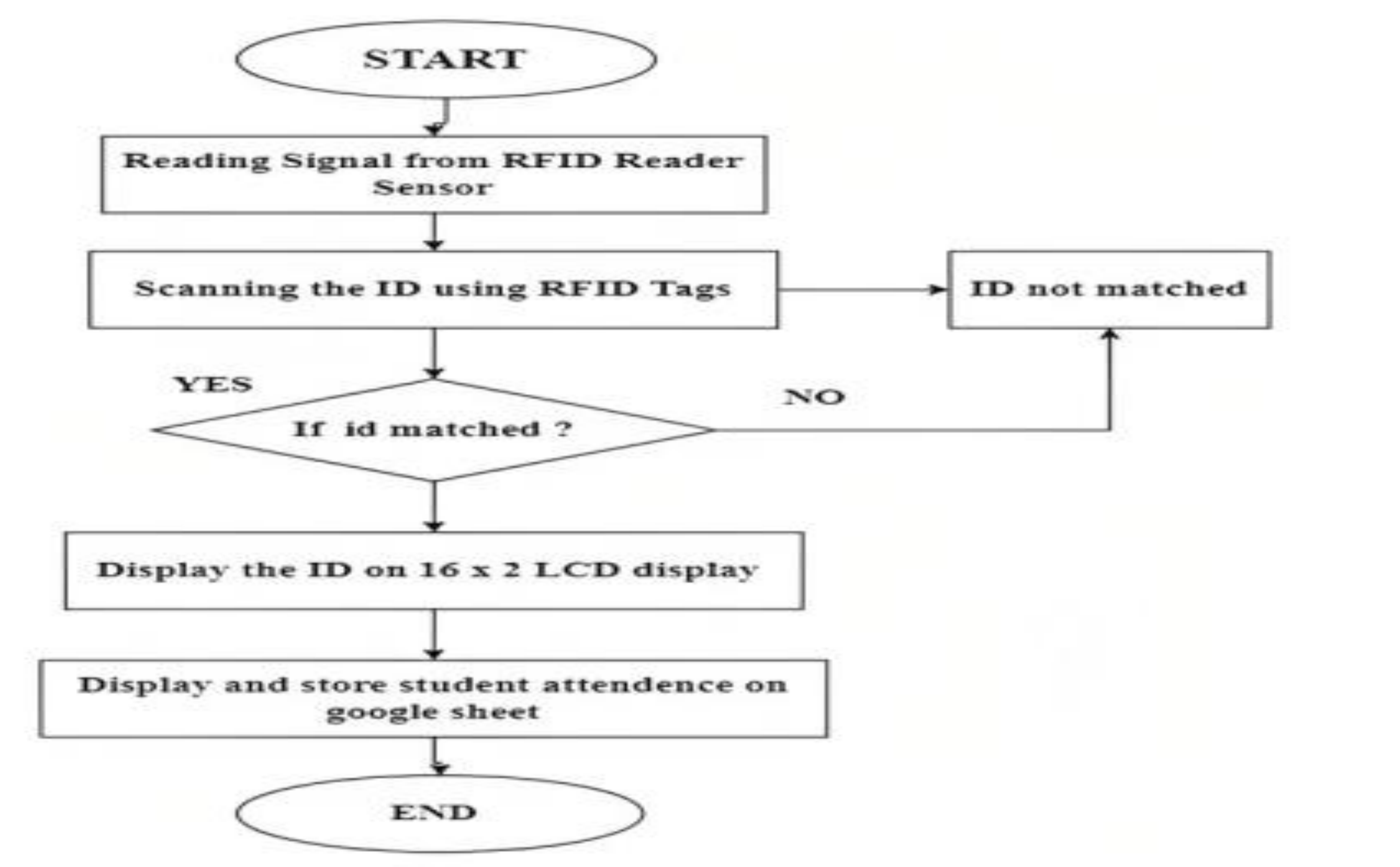
PROJECT INTRODUCTION

This project aims to seamlessly integrate Radio-Frequency Identification (RFID) technology with Google Spreadsheet through interrupt request mechanisms. RFID technology enables automatic identification and tracking, while interrupt requests facilitate real-time posting of RFID data to the cloud-based spreadsheet. By leveraging these technologies, the project seeks to enhance data management efficiency and accessibility, offering a reliable solution for uninterrupted RFID data integration.

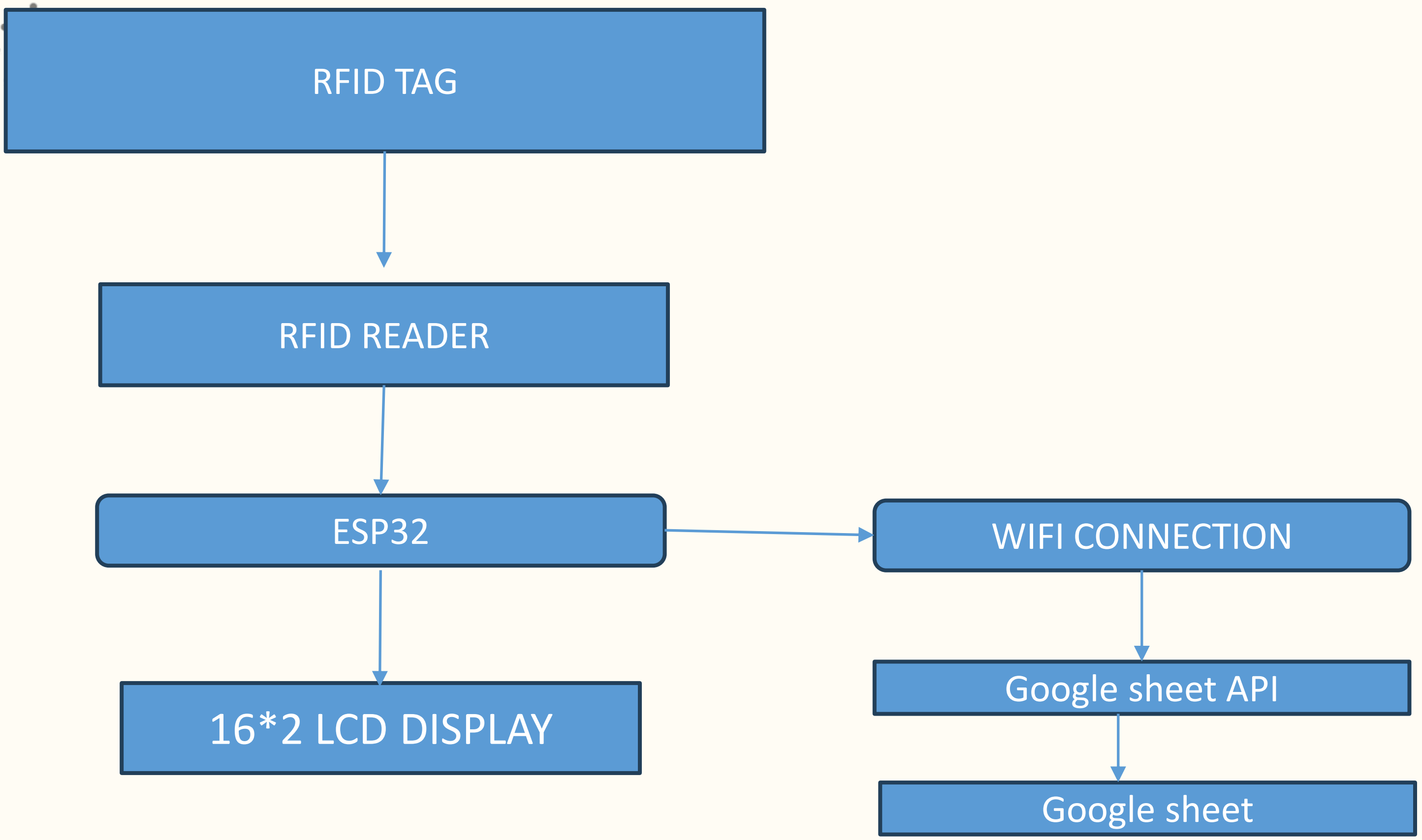
PROJECT DESCRIPTION

This project aims to integrate RFID data with Google Spreadsheet using interrupt requests. RFID technology automatically identifies and tracks objects, while interrupts facilitate real-time posting of data to the spreadsheet. The system enhances data management efficiency and accessibility, providing a reliable solution for RFID data integration.

FLOW CHART


START
Reading Signal from RFID Reader Sensor
Scanning the ID using RFID Tags
ID not matched
YES
If id matched ?
NO
Display the ID on 16 x 2 LCD display
Display and store student attendence on google sheet
END

BLOCK DIAGRAM



RFID TAG

RFID READER

ESP32

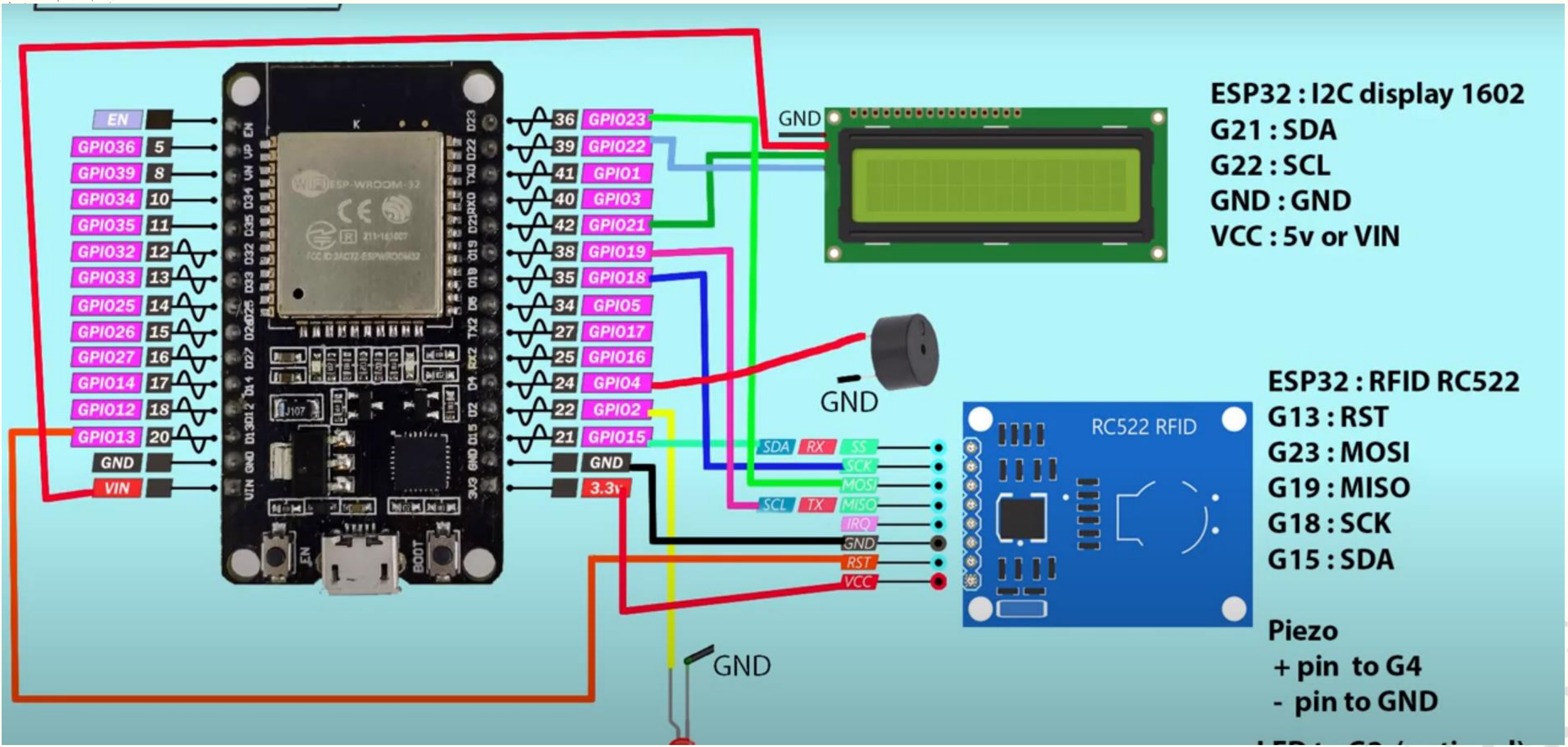
16*2 LCD DISPLAY

WIFI CONNECTION

Google sheet API

Google sheet

CIRCUIT DIAGRAM



ESP32 : I2C display 1602
G21 : SDA
G22 : SCL
GND : GND
VCC : 5v or VIN

ESP32 : RFID RC522
G13 : RST
G23 : MOSI
G19 : MISO
G18 : SCK
G15 : SDA

Piezo
+ pin to G4
- pin to GND

HARDWARE & SOFTWARE REQUIREMENTS

Hardware Requirements:

1.ESP32 Development Board

2.CONNECTING WIRES

3.16*2 LCD DISPLAY

4.Breadboard and Jumper Wires

5.RC522 RFID MODULE

6.RFID TAGS

Software Requirements:

1.Arduino IDE

2.ESP32 Board Support Package for Arduino IDE

3.USB Drivers

4.Serial Monitor Software

FUTURE SCOPE

Implementing an interrupt request-based posting of RFID data to a Google spreadsheet can significantly enhance the efficiency and reliability of attendance tracking systems. Here's a brief overview of the future scope of such a project:

1. **Enhanced Real-Time Tracking**: By utilizing interrupt requests, the system can capture RFID data instantly as events occur. This enables real-time attendance tracking, providing accurate insights into attendance patterns and trends.
2. **Scalability**: The project can be scaled to accommodate a larger number of RFID tags and readers, making it suitable for various applications ranging from small classrooms to large-scale events and organizations.
3. **Integration with Cloud Services**: Integrating with Google Sheets allows for seamless data storage and access from anywhere with an internet connection. In the future, the project could expand to integrate with other cloud-based platforms for enhanced analytics and reporting capabilities.
4. **Customization and Flexibility**: Future iterations of the project can focus on enhancing the customization options, allowing users to configure the system according to their specific requirements. This could include features such as customizable data fields, user access controls, and advanced reporting functionalities.

APPLICATIONS

The applications of interrupt request-based posting of RFID data to a Google spreadsheet are diverse and can be found in various industries and scenarios, including:

1. **Education**: Tracking student attendance in schools, colleges, and universities.
2. **Corporate Settings**: Managing employee attendance in offices, meetings, and conferences.
3. **Events Management**: Monitoring attendance at conferences, seminars, workshops, and trade shows.
4. **Healthcare Facilities**: Tracking patient visits and staff presence in hospitals, clinics, and medical facilities.
5. **Public Transport**: Monitoring passenger traffic on buses, trains, and subways for scheduling and planning.