Koneru Lakshmaiah Education Foundation (Deemed to be University)

Course Code: 22SDEC02

Course Name: Electronic system Automation

(Skill Development Project)

A Project Report

On

Agricultural Monitoring and visualization in Thing speak

SUBMITTED BY:

ID NUMBER	NAME
2200040315	BARATAM MOUNIKA
2200040319	NAGARAJU
2200040330	KONDI SHASHANK

UNDER THE GUIDANCE OF

Dr. AV Prabhu & Dr. Saleem Akram

(DEEMED TO BE UNIVERSITY)

Green fields, Vaddeswaram – 522 502
Guntur Dist., AP, India.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

DECLARATION

The Project Report entitled **"**Agricultural Monitoring and visualization in Thing speak **"** is a record of bonafide work of **SHASHANK(2200040330), Mounika.B (2200040315), Nagaraju (2200040319** submitted in partial fulfillment for the award of B.Tech in Electronics and Communication Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

(DEEMED TO BE UNIVERSITY)

CERTIFICATE

This is to certify that the Project Report entitled "Agricultural Monitoring and visualization in Thing speak " is being submitted by **SHASHANK(2200040330), Mounika.B (2200040315), Nagaraju (2200040319)** submitted in partial fulfilment for the award of B.Tech in Electronics and Communication Engineering to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of Supervisor

Signature of the HOD

Signature of the External Examiner

ABSTRACT

Sensor Deployment: Various sensors such as temperature, humidity, soil moisture, and light sensors are strategically deployed across the agricultural field to monitor key environmental parameters affecting crop growth.

Data Collection: The sensors continuously collect data on environmental conditions, which is then transmitted to the ThingSpeak IoT platform via a wireless communication protocol such as Wi-Fi or LoRaWAN.

Data Processing: ThingSpeak processes the incoming data in real-time, performing calculations or transformations as needed to derive useful insights. For example, it may calculate the average temperature or humidity over a certain period.

Visualization: The processed data is then visualized using customizable graphs, charts, and gauges on the ThingSpeak dashboard. Farmers can easily interpret the data and track changes in environmental conditions over time.

Alerts and Notifications: The system can be configured to trigger alerts or notifications based on predefined thresholds. For example, farmers may receive alerts when the soil moisture levels drop below a certain threshold, indicating the need for irrigation.

Remote Monitoring: Farmers can access the ThingSpeak dashboard from any internet-enabled device, allowing for remote monitoring of their crops. This enables proactive decision-making and timely interventions to optimize crop yield and quality.

Historical Data Analysis: ThingSpeak stores historical data, allowing farmers to analyze trends and patterns in environmental conditions over extended periods. This data can inform long-term planning and decision-making processes.

Integration with Other Systems: ThingSpeak can be integrated with other agricultural management systems or third-party services, allowing for seamless data exchange and interoperability.

Literature Survey

Article
http://iyfbodn.com/?dn=examplelink.com&pid=9POT3387I&pbsubid=d97e8ab1-9bbb-4883-b4bc-1b1e238b37ba&noads=http%3A%2F%2Fiyfbodn.com%2F%3Fdn%3Dexamplelink.com%26skipskenzo%3Dtrue

Paper:
http://iyfbodn.com/?dn=examplelink.com&pid=9POT3387I&pbsubid=d97e8ab1-9bbb-4883-b4bc-1b1e238b37ba&noads=http%3A%2F%2Fiyfbodn.com%2F%3Fdn%3Dexamplelink.com%26skipskenzo%3Dtrue

Thesis: "Smart agriculture in Embedded Systems" by Nguyen

INDEX

AIM OF THE PROJECT

The aim of the project is to develop an effective agriculture monitoring system using an Internet of Things (IoT) platform, such as ThingSpeak. This system will enable farmers to gather real-time data on various environmental parameters crucial for crop growth and management. By leveraging IoT technology, the project seeks to provide farmers with valuable insights into their crops' health and environmental conditions, facilitating data-driven decision-making processes.

Specifically, the project aims to achieve the following objectives:

Data Collection: Deploy sensors across the agricultural field to collect data on key environmental parameters such as temperature, humidity, soil moisture, and light intensity.

Data Transmission: Utilize wireless communication protocols, such as Wi-Fi or LoRaWAN, to transmit the sensor data to the ThingSpeak IoT platform in real-time.

Data Processing: Process the incoming sensor data on the ThingSpeak platform to derive useful insights and trends. This may involve calculations, data aggregation, and statistical analysis.

Visualization: Present the processed data in a user-friendly manner using customizable graphs, charts, and gauges on the ThingSpeak dashboard. This allows farmers to easily interpret the data and track changes over time.

Alerting Mechanism: Implement an alerting mechanism to notify farmers of any abnormal or critical conditions detected by the sensors. This ensures timely intervention to mitigate potential risks to crop health.

Historical Data Analysis: Store historical sensor data on the ThingSpeak platform for future analysis and reference. This enables farmers to analyze trends and patterns in environmental conditions over extended periods, aiding in long-term planning and decision-making.

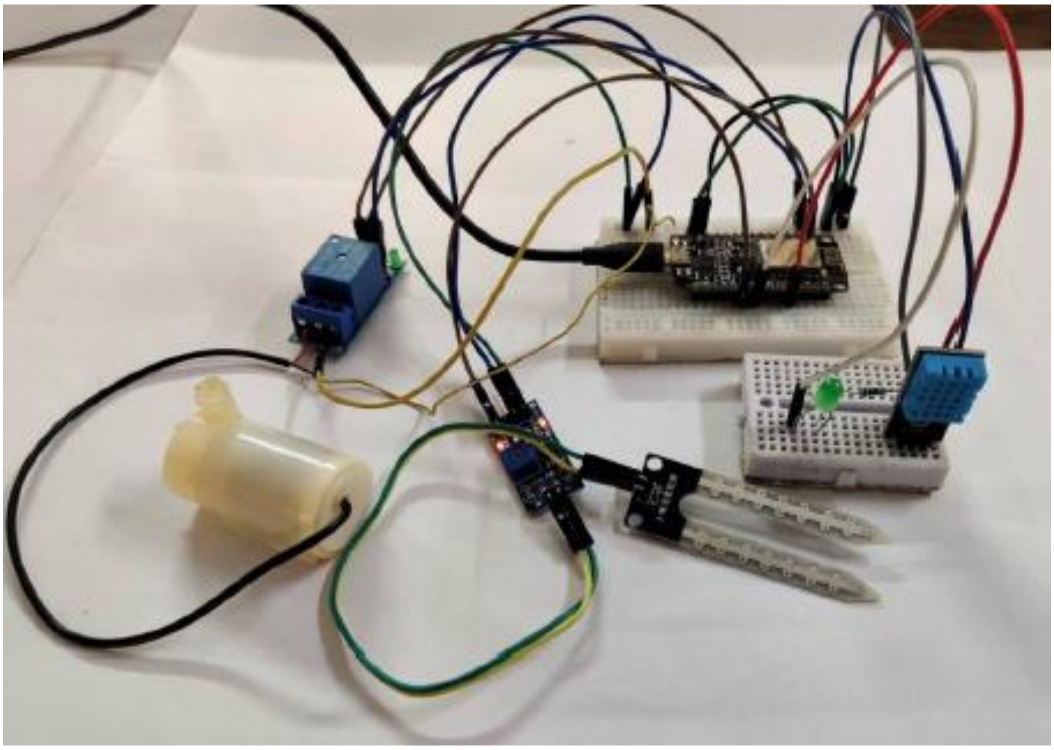
PROCEDURE

1. Planning: Identify parameters to monitor and select suitable sensors.
2. Deployment: Place sensors in the field.
3. Setup: Connect sensors to microcontrollers/IoT devices.
4. ThingSpeak: Create an account, set up channels, and obtain API keys.
5. Data Transmission: Program devices to send sensor data to ThingSpeak.
6. Visualization: Configure ThingSpeak to display data with graphs/charts.
7. Alerts: Set thresholds for alerts on critical conditions.
8. Testing: Ensure sensors transmit accurate data and alerts work.
9. User Interface: Develop a simple interface for users.
10. Deployment: Install the system in the field.

COMPONENTS REQUIRED

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

CIRCUIT DIAGRAM



CODE:

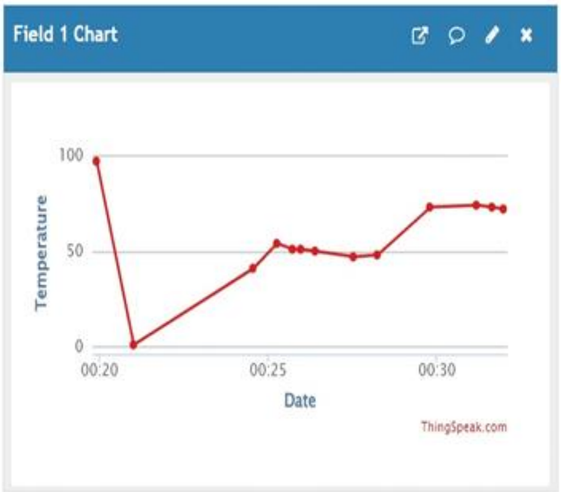
```
#include <DHT.h> // Including library for dht
#include "MQ135.h"
#include <ESP8266WiFi.h>
String apiKey = "MA15S3HU37NJGAH3"; // Enter your Write API key from ThingSpeak
const char *ssid = "OPPO A54"; // replace with your wifi ssid and wpa2 key
const char *pass = "jyoshi05";
const char* server = "api.thingspeak.com";
const int sensorPin= 0;
#define DHTPIN 4 //Connect to GPIO 2 in NodeMCU Board
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup()
{
Serial.begin(115200);
delay(10);
dht.begin();
Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
{
```

```
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
```

```
}
void loop()
{
int value = analogRead(A0);
value = map(value,400,1023,100,0);
float h = dht.readHumidity();
float t = dht.readTemperature();
if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com
{
String postStr = apiKey;
postStr +="&field1=";
postStr += String(value);
postStr +="&field2=";
postStr += String(t);
postStr +="&field3=";
postStr += String(h);
postStr += "\r\n\r\n";
client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);
Serial.print("Temperature: ");
```

```
Serial.print(t);Serial.print(" degrees Celcius, Humidity: "):
Serial.print(h);
Serial.print("%, Soil Moisture ");
Serial.print(value);
Serial.println("%. Send to Thingspeak.");
}
client.stop();
Serial.println("Waiting...");
// thingspeak needs minimum 15 sec delay between updates
delay(1000);
```

OUTPUT



Koneru Lakshmaiah Education Foundation (Deemed to be University)

Course Code: 22SDEC02

Course Name: Electronic system Automation

(Skill Development Project)

A Project Report

On

Bluetooth-Based Home Automation System using Sarito Hobby App

SUBMITTED BY:

ID NUMBER	NAME
2200040330	SHASHANK
2200040315	MOUNIKA
2200040319	NAGARAJU

UNDER THE GUIDANCE OF

Dr. AV Prabhu & Dr. Saleem Akram

(DEEMED TO BE UNIVERSITY)

Green fields, Vaddeswaram – 522 502
Guntur Dist., AP, India.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

DECLARATION

The Project Report entitled **"Bluetooth-Based Home Automation System using Sarito Hobby App"** is a record of bonafide work of **Shashank (2200040330), Mounika(2200040315), Nagaraju (2200040319)** submitted in partial fulfillment for the award of B.Tech in Electronics and Communication Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

(DEEMED TO BE UNIVERSITY)

CERTIFICATE

This is to certify that the Project Report entitled "Bluetooth-Based Home Automation System using Sarito Hobby App" is being submitted by **Shashank (2200040330), Mounika(2200040315), Nagaraju (2200040319)** submitted in partial fulfilment for the award of B.Tech in Electronics and Communication Engineering to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of Supervisor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGMENT

It is great pleasure for me to express my gratitude to our honourable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project report.

I express the sincere gratitude to our Vice Chancellor, **Dr G P S Varma** for his administration towards our academic growth.

I express the sincere gratitude to our Pro VC, **Dr N Venkatram** for his administration towards our academic growth.

I record it as my privilege to deeply thank our principal, **Dr.T.K.Ramakrishna Rao** for providing us the efficient faculty and facilities to make our ideas into reality.

I express sincere gratitude to our pioneer **Dr. Suman Maloji,** Vice-Principal & HOD-ECE for his leadership and constant motivation provided in successful completion of our academic semester.

I express my sincere thanks to our project mentors Dr. AV Prabhu & Dr. Saleem Akram for their novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

ABSTRACT

This project presents a Bluetooth-based home automation system integrated with the Sarito Hobby app for seamless control of household appliances and systems. Home automation systems offer convenience, energy efficiency, and enhanced security by allowing users to remotely monitor and control various devices within their homes. The Sarito Hobby app provides a user-friendly interface for configuring and managing connected devices via Bluetooth communication. The project involves designing and implementing the hardware and software components of the home automation system, including Bluetooth-enabled microcontrollers, actuators, and sensors. The Sarito Hobby app serves as the central control hub, allowing users to remotely turn on/off appliances, adjust settings, and receive status notifications. The project aims to demonstrate the feasibility and effectiveness of Bluetooth-based home automation solutions and provide a customizable platform for implementing smart home functionalities.

Literature Survey

Article: "Bluetooth Low Energy (BLE) for Internet of Things (IoT): A Survey" by Gupta et al

Paper: "Design and Implementation of a Bluetooth-based Home Automation System" by Kumar and Singh

Thesis: "Development of a Bluetooth-Based Home Automation System" by Patel

INDEX

S.NO	TITLE
1	Introduction
2	Aim of the Project
3	Proposed Methodology (Include Block Diagram/Flow Chart)
4	Components Explanation
5	Implementation Methodology (Circuit and Implementation)
6	Results and Discussions (With Screenshots)
7	Source Code
8	Conclusion and Future Scope

AIM OF THE PROJECT

The central aim of this project is to design and implement a Bluetooth-based home automation system integrated with the Sarito Hobby app, with the objective of providing users with convenient and intuitive control over household appliances and systems. Home automation systems offer numerous benefits, including enhanced energy efficiency, convenience, and security.

By leveraging Bluetooth connectivity, the project aims to enable users to remotely monitor and control various devices within their homes using their smartphones or tablets. The Sarito Hobby app will serve as the central control hub, allowing users to easily configure and manage connected devices. The project will involve designing and implementing the hardware and software components of the home automation system, including Bluetooth-enabled microcontrollers, actuators, and sensors.

Additionally, the project will explore the integration of advanced features such as voice control and scheduling functionalities to further enhance user experience and system functionality. Ultimately, the goal is to demonstrate the feasibility and effectiveness of Bluetooth-based home automation solutions, providing users with a customizable platform for implementing smart home functionalities.

PROCEDURE

1. Connect the ESP32 board to the Bluetooth module, relay module & buzzer

using jumper wires.

2. Connect the relay module to the bulb, ensuring proper electrical connections.

3. Install the Sarito Hobby app on your smartphone and pair it with the Bluetooth

module.

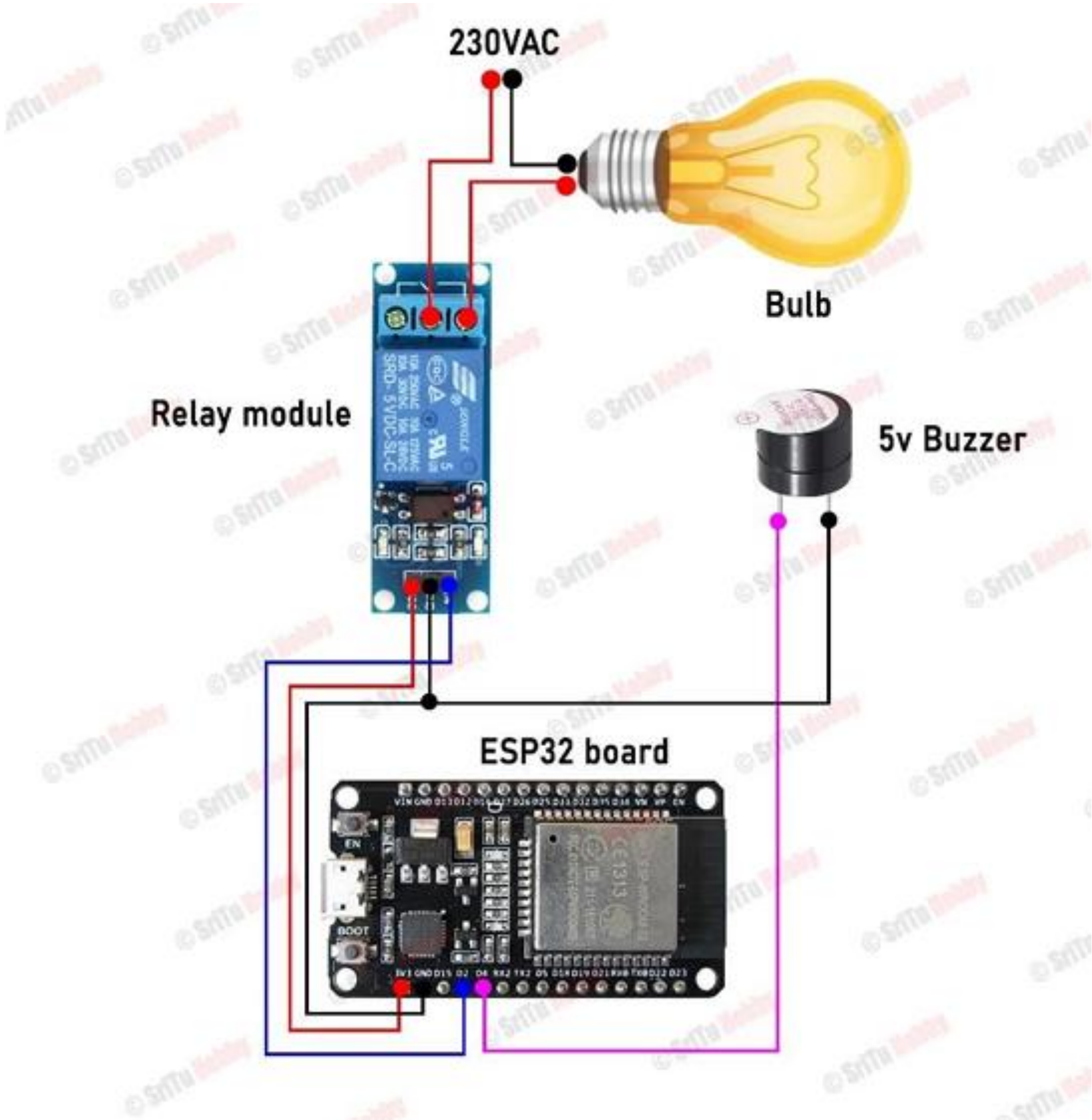
4. Write a program using Arduino IDE to: - Establish Bluetooth communication. - Receive voice commands from the Sarito Hobby app. - Control the relay to turn the bulb on or off based on recognized commands. - Activate the buzzer for unrecognized commands.

5. Upload the code to the ESP32 board.

COMPONENTS REQUIRED

- ESP32 Board
- Relay module
- Buzzer
- Bulb
- Jumper wires
- Sarito Hobby mobile app
- Smartphone with Bluetooth capability

CIRCUIT DIAGRAM

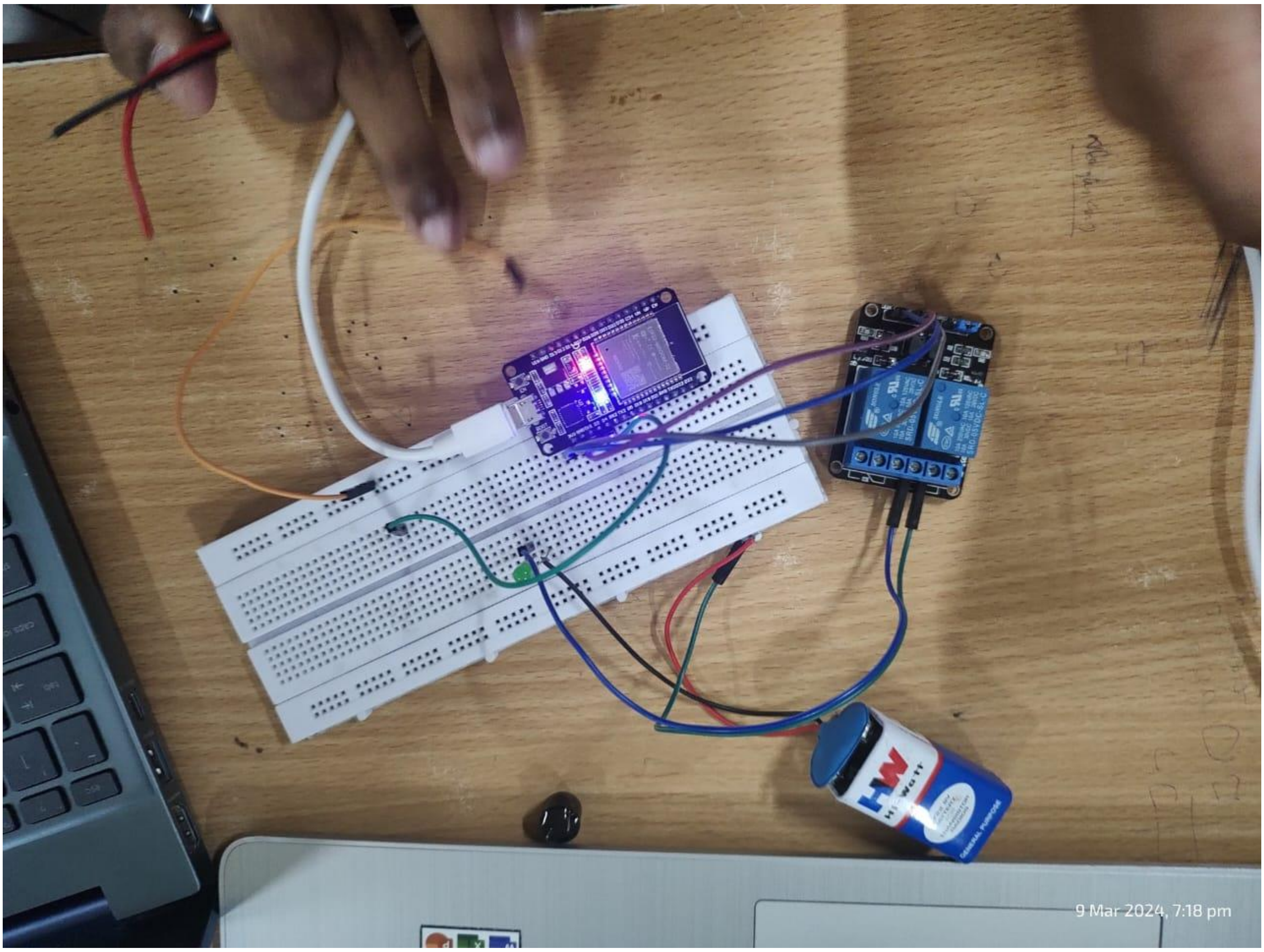


CODE:

```
//Include the library file
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
#define relay 2 // Relay pin
#define buzzer 4 // Buzzer pin
void setup() {
Serial.begin(115200);
SerialBT.begin("SriTu Hobby"); //Bluetooth device name
Serial.println("The device started, now you can pair it with bluetooth!");
pinMode(relay, OUTPUT);
pinMode(buzzer, OUTPUT);
digitalWrite(relay, HIGH);
}
void loop() {
if (SerialBT.available()) {
String value = SerialBT.readString();
Serial.println(value);
if (value == "turn on the bulb") {
digitalWrite(relay, LOW);
Serial.println("on");
} else if (value == "turn off the bulb") {
digitalWrite(relay, HIGH);
Serial.println("off");
} else {
digitalWrite(buzzer, HIGH);
delay(100);
digitalWrite(buzzer, LOW);
delay(100);
digitalWrite(buzzer, HIGH);
delay(100);
```

```
digitalWrite(buzzer, LOW);
delay(100);
digitalWrite(buzzer, HIGH);
delay(100);
digitalWrite(buzzer, LOW);
delay(100);
}
}
}
```

OUTPUT



Koneru Lakshmaiah Education Foundation (Deemed to be University)

Course Code: 22SDEC02

Course Name: Electronic system Automation

(Skill Development Project)

A Project Report

RFID Data posting on Google Sheets

SUBMITTED BY:

ID NUMBER	NAME
2200040315	BARATAM MOUNIKA
2200040319	NAGARAJU
2200040330	KONDI SHASHANK

UNDER THE GUIDANCE OF

Dr. AV Prabhu & Dr. Saleem Akram

(DEEMED TO BE UNIVERSITY)

Green fields, Vaddeswaram – 522 502
Guntur Dist., AP, India.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

DECLARATION

The Project Report entitled **"**RFID Data posting on Google Sheets**"** is a record of bonafide work of **SHASHANK(2200040330), Mounika.B (2200040315), Nagaraju (2200040319** submitted in partial fulfillment for the award of B.Tech in Electronics and Communication Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

CERTIFICATE

This is to certify that the Project Report entitled "RFID Data posting on Google Sheets" is being submitted by **SHASHANK(2200040330), Mounika.B (2200040315), Nagaraju (2200040319)** submitted in partial fulfilment for the award of B.Tech in Electronics and Communication Engineering to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of Supervisor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGMENT

It is great pleasure for me to express my gratitude to our honourable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project report.

I express the sincere gratitude to our Vice Chancellor, **Dr G P S Varma** for his administration towards our academic growth.

I express the sincere gratitude to our Pro VC, **Dr N Venkatram** for his administration towards our academic growth.

I record it as my privilege to deeply thank our principal, **Dr.T.K.Ramakrishna Rao** for providing us the efficient faculty and facilities to make our ideas into reality.

I express sincere gratitude to our pioneer **Dr. Suman Maloji,** Vice-Principal & HOD-ECE for his leadership and constant motivation provided in successful completion of our academic semester.

I express my sincere thanks to our project mentors Dr. AV Prabhu & Dr. Saleem Akram for their novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

ABSTRACT

Scripts for logging data to Google Sheets. Data logging is essential for monitoring and analyzing various parameters in applications such as environmental monitoring, industrial automation, and scientific research. The project involves interfacing ESP32 microcontrollers with sensors to collect data from the physical environment. Google Scripts are used to create custom functions that enable communication between ESP32 devices and Google Sheets. Data logged by ESP32 devices is transmitted to Google Sheets in real-time or at predefined intervals, where it is stored and organized for further analysis. The project aims to provide a cost-effective and scalable solution for data logging applications, leveraging the ubiquity and accessibility of Google Sheets for data storage and analysis. Additionally, the project demonstrates the versatility of ESP32 microcontrollers in IoT applications and highlights the integration of cloud-based services for data management and visualization.

Literature Survey

Paper: "ESP32 for IoT Applications: A Comprehensive Guide" by ESPRESSIF Systems

Post: "Integrating ESP32 with Google Sheets for Real-Time Data Logging" by IoTBits

Documentation: "Google Apps Script" by Google Developers.

INDEX

AIM OF THE PROJECT

The primary aim of this project is to develop a data logging system using ESP32 microcontrollers and Google Scripts for logging data to Google Sheets. Data logging is essential for monitoring and analyzing various parameters in applications such as environmental monitoring, industrial automation, and scientific research.

The project seeks to establish a seamless and scalable mechanism for capturing, storing, and analyzing data from various sources in real-time. By leveraging the capabilities of ESP32 microcontrollers and Google Sheets, the project aims to facilitate data-driven decision-making and enhance system efficiency. The project will involve interfacing ESP32 microcontrollers with sensors to collect data from the physical environment.

Google Scripts will be used to create custom functions that enable communication between ESP32 devices and Google Sheets, allowing data logged by ESP32 devices to be transmitted to Google Sheets for storage and analysis. Ultimately, the project aims to provide a cost-effective and versatile solution for data logging applications, leveraging the ubiquity and accessibility of Google Sheets for data storage and analysis.

PROCEDURE

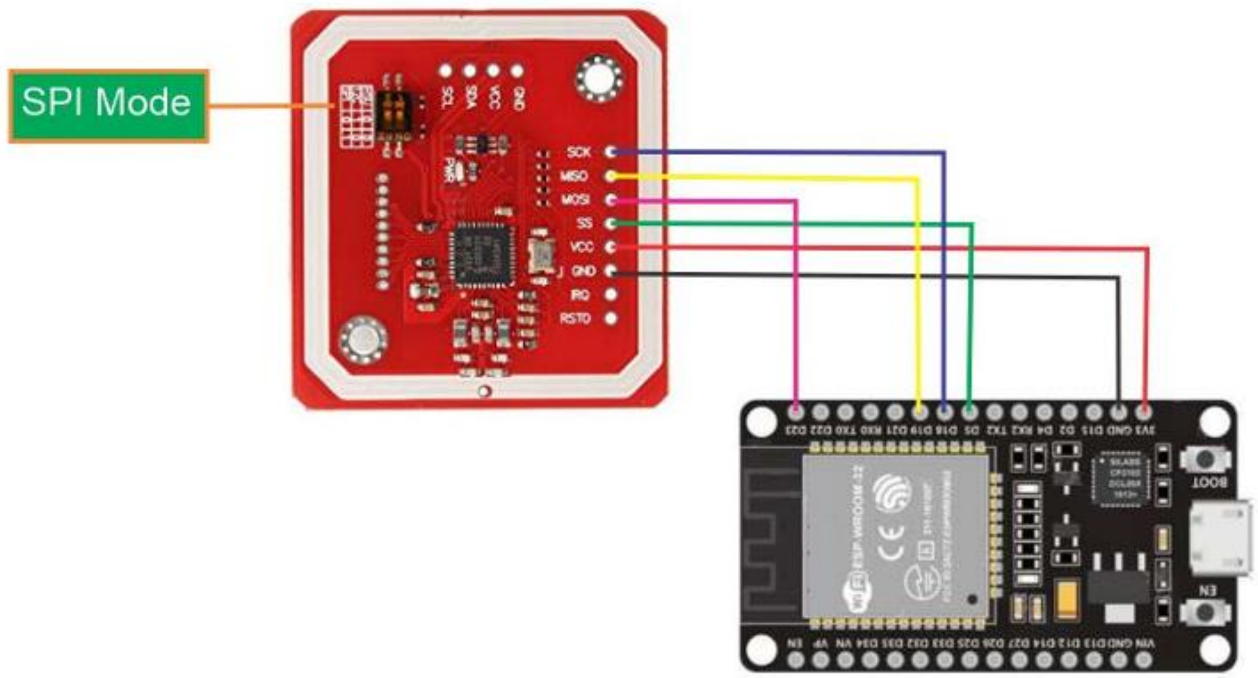
1. Connect the sensors to the ESP32 and configure the microcontroller to read sensor data.
2. Set up the ESP32 with Wi-Fi access to connect to the internet.
3. Create a Google Sheets spreadsheet and obtain its associated API key and Sheet ID.

4. Develop an Arduino sketch to send sensor data to the Google Sheets using the Google Sheets API.
5. Create a Google Script to handle incoming data and log it into the spreadsheet.
6. Schedule the Google Script to run periodically for continuous data logging.

COMPONENTS REQUIRED

- ESP32 microcontroller
- Sensors (e.g., temperature, humidity)
- USB cable for ESP32 programming
- Google account for Google Sheets access
- Google Sheets spreadsheet for data storage
- Arduino IDE with ESP32 support
- Internet connection (Wi-Fi)

CIRCUIT DIAGRAM



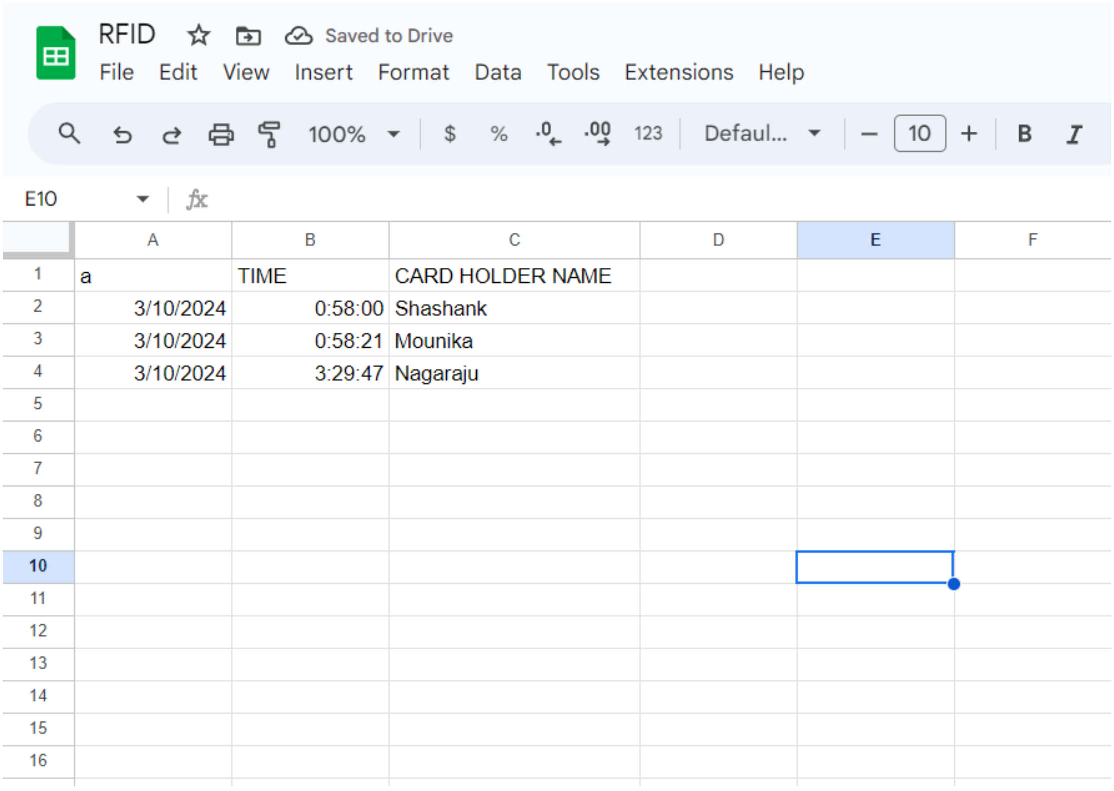
CODE:

```
//Include required libraries

#include "WiFi.h"

#include <HTTPClient.h>

#include "time.h"

const char* ntpServer = "pool.ntp.org";

const long  gmtOffset_sec = 19800;

const int   daylightOffset_sec = 0;

// WiFi credentials

const char* ssid = "SKYNET 4G";

 // change SSID

const char* password = "jobitjos";    // change password

// Google script ID and required credentials

String GOOGLE_SCRIPT_ID = "AKfycby-snBh-5j0jsiQBWfC

XB1FWy38lks4VHcxLBIGNadeCVcSzUoozHzvazIWv9EcA6a";    // change

Gscript ID

int count = 0;

void setup() {

delay(1000);

Serial.begin(115200);

delay(1000);

// connect to WiFi

Serial.println();

Serial.print("Connecting to wifi: ");

Serial.println(ssid);

Serial.flush();

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

delay(500);

Serial.print(".");

}

// Init and get the time

configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);

}

void loop() {

if (WiFi.status() == WL_CONNECTED) {

static bool flag = false;

struct tm timeinfo;

if (!getLocalTime(&timeinfo)) {

Serial.println("Failed to obtain time");

return;
```

```
}
char timeStringBuff[50]; //50 chars should be enough
strftime(timeStringBuff, sizeof(timeStringBuff), "%A, %B %d %Y
%H:%M:%S", &timeinfo);
String asString(timeStringBuff);
asString.replace(" ", "-");
Serial.print("Time:");
Serial.println(asString);
String urlFinal =
"https://script.google.com/macros/s/"+GOOGLE_SCRIPT_ID+"/exec?"+"date=
 " + asString + "&sensor=" + String(count);
Serial.print("POST data to spreadsheet:");
Serial.println(urlFinal);
HTTPClient http;
http.begin(urlFinal.c_str());
http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
int httpCode = http.GET();
Serial.print("HTTP Status Code: ");
Serial.println(httpCode);
//---------------------------------------------------------------------
//getting response from google sheet
String payload;
if (httpCode > 0) {
payload = http.getString();
Serial.println("Payload: "+payload);
}
//---------------------------------------------------------------------
http.end();
}
count++;
delay(1000);
}
......................................................................................................
//Include required libraries
#include "WiFi.h"
#include <HTTPClient.h>
// WiFi credentials
const char* ssid = "SKYNET 4G";
 // change SSID
const char* password = "jobitjos";    // change password
// Google script ID and required credentials
String GOOGLE_SCRIPT_ID = "AKfycby-snBh-5j0jsiQBWfC
```

```
XB1FWy38lks4VHcxLBIGNadeCVcSzUoozHzvazIWv9EcA6a";    // change
Gscript ID
void setup() {
delay(1000);
Serial.begin(115200);
delay(1000);
// connect to WiFi
Serial.println();
Serial.print("Connecting to wifi: ");
Serial.println(ssid);
Serial.flush();
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
}
void loop() {
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
String url = "https://script.google.com/macros/s/" + GOOGLE_SCRIPT_ID +
"/exec?read";
Serial.println("Making a request");
http.begin(url.c_str()); //Specify the URL and certificate
http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
int httpCode = http.GET();
String payload;
if (httpCode > 0) { //Check for the returning code
payload = http.getString();
Serial.println(httpCode);
Serial.println(payload);
}
else {
Serial.println("Error on HTTP request");
}
http.end();
}
delay(1000);
}
```

OUTPUT



a	TIME	CARD HOLDER NAME
3/10/2024	0:58:00	Shashank
3/10/2024	0:58:21	Mounika
3/10/2024	3:29:47	Nagaraju

Koneru Lakshmaiah Education Foundation (Deemed to be University)

Course Code: 22SDEC02

Course Name: Electronic system Automation

(Skill Development Project)

A Project Report

On

Interrupt Request-Based Audio Input from I2S Microphone

SUBMITTED BY:

ID NUMBER	NAME
2200040315	BARATAM MOUNIKA
2200040319	NAGARAJU
2200040330	KONDI SHASHANK

UNDER THE GUIDANCE OF

Dr. AV Prabhu & Dr. Saleem Akram

Green fields, Vaddeswaram – 522 502
Guntur Dist., AP, India.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

DECLARATION

The Project Report entitled **"**Interrupt Request-Based Audio Input from I2S Microphone**"** is a record of bonafide work of **SHASHANK(2200040330), Mounika.B (2200040315), Nagaraju (2200040319** submitted in partial fulfillment for the award of B.Tech in Electronics and Communication Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

K L E F

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(DST-FIST Sponsored Department)

CERTIFICATE

This is to certify that the Project Report entitled "Interrupt Request-Based Audio Input from I2S Microphone" is being submitted by **SHASHANK(2200040330), Mounika.B (2200040315), Nagaraju (2200040319)** submitted in partial fulfilment for the award of B.Tech in Electronics and Communication Engineering to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of Supervisor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGMENT

It is great pleasure for me to express my gratitude to our honourable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project report.

I express the sincere gratitude to our Vice Chancellor, **Dr G P S Varma** for his administration towards our academic growth.

I express the sincere gratitude to our Pro VC, **Dr N Venkatram** for his administration towards our academic growth.

I record it as my privilege to deeply thank our principal, **Dr.T.K.Ramakrishna Rao** for providing us the efficient faculty and facilities to make our ideas into reality.

I express sincere gratitude to our pioneer **Dr. Suman Maloji,** Vice-Principal & HOD-ECE for his leadership and constant motivation provided in successful completion of our academic semester.

I express my sincere thanks to our project mentors Dr. AV Prabhu & Dr. Saleem Akram for their novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

ABSTRACT

The project aims to enhance real-time audio processing efficiency by implementing an interrupt request-based approach for handling input from an I2S microphone. Traditional methods of audio processing often involve continuous polling of input devices, leading to inefficiencies in resource utilization and latency issues. By contrast, utilizing interrupt requests allows the system to asynchronously handle audio input events, only triggering processing routines when necessary. This approach minimizes processing overhead and ensures timely response to audio input events, ultimately improving the overall performance of the audio processing system. The project involves designing and implementing interrupt service routines (ISRs) to manage audio input interrupts effectively. Additionally, optimizations such as buffer management and event-driven processing will be explored to further enhance system efficiency. The project aims to demonstrate the benefits of interrupt request-based audio input handling and provide insights into its application in real-world audio processing systems.

Literature Survey

Article: "Interrupt-Driven I2S Audio Interface for Portable Embedded Systems" by Smith et al.

Paper: "Efficient Real-Time Audio Processing Techniques for Embedded Systems" by Johnson and Patel.

Thesis: "Optimizing Audio Processing Using Interrupt Requests in Embedded Systems" by Nguyen

INDEX

AIM OF THE PROJECT

The primary aim of this project is to develop a real-time audio processing system capable of efficiently handling audio input from an I2S microphone by leveraging interrupt requests. Traditional audio processing methods often involve continuous polling of input devices, which can lead to inefficiencies in resource utilization and increased latency.

The project aims to design and implement interrupt service routines (ISRs) that effectively manage audio input interrupts, triggering processing routines only when necessary.

By implementing interrupt request-based handling, the project seeks to minimize processing overhead and latency, thereby enhancing the overall responsiveness and efficiency of the system.

Additionally, the project will explore optimizations such as buffer management and event-driven processing to further improve system performance. Ultimately, the goal is to demonstrate the feasibility and benefits of interrupt request-based audio input handling in real-world applications, paving the way for more efficient and responsive audio processing systems.

PROCEDURE

1. Connect the I2S microphone to the microcontroller using the appropriate pins and level shifters.

2. Configure the microcontroller to handle I2S communication.

3. Implement an interrupt service routine (ISR) to handle incoming audio data.

4. Set up the system to trigger interrupts based on specific conditions

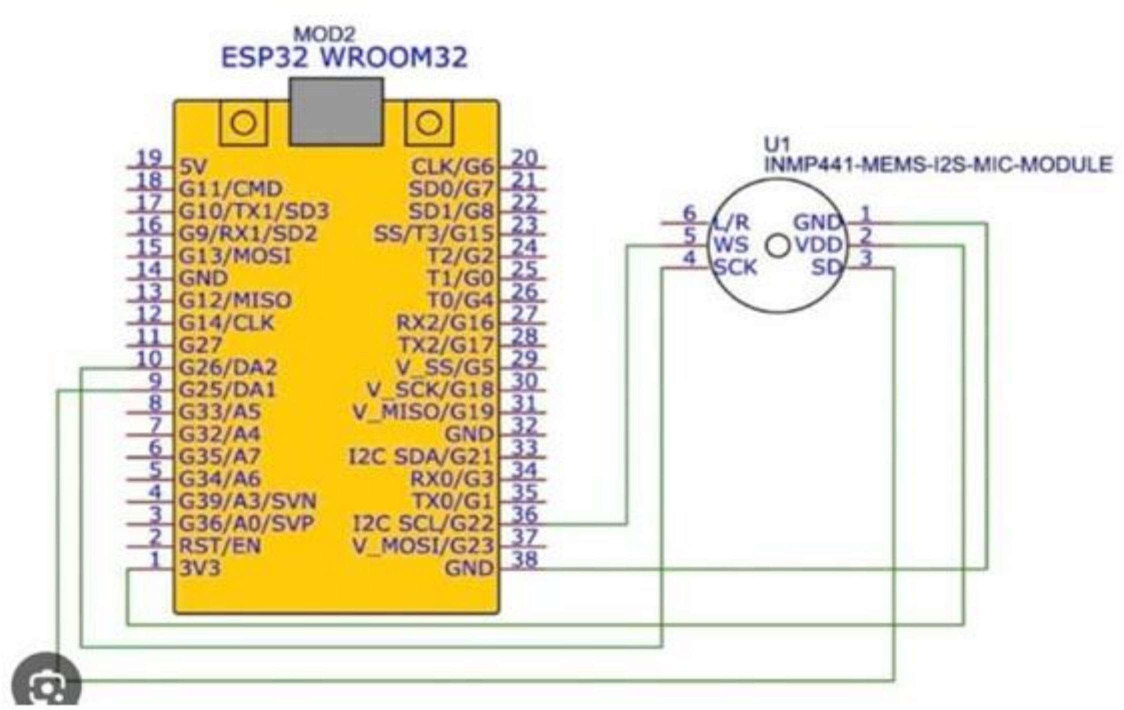
5. Develop algorithms or processes to process and analyze audio data within the ISR.

6. Utilize the processed data for specific applications (e.g., voice recognition, audio recording).

COMPONENTS REQUIRED

- I2S Microphone (INMP441)
- Microcontroller ESP32
- Jumper wires
- Power supply
- Speaker

CIRCUIT DIAGRAM



CODE:

```
#include <driver/i2s.h>
#include <SPI.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];
// Connections to MAX98357 amplifier
#define MAX98357_DOUT 26
#define MAX98357_BCLK 27
#define MAX98357_LRC 14
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_16BIT,
.channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
.communication_format = 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 = I2S_PIN_NO_CHANGE,
.data_in_num = I2S_SD
};
i2s_set_pin(I2S_PORT, &pin_config);
}
void max98357_setup() {
// Configure MAX98357 amplifier pins
pinMode(MAX98357_DOUT, OUTPUT);
pinMode(MAX98357_BCLK, OUTPUT);
pinMode(MAX98357_LRC, OUTPUT);
}
void max98357_write(uint8_t reg, uint16_t data) {
// Write data to MAX98357 amplifier via SPI protocol
digitalWrite(MAX98357_LRC, HIGH);
SPI.beginTransaction(SPISettings(2000000, MSBFIRST, SPI_MODE0));
SPI.transfer(0x70 | (reg >> 7)); // Set control bits to address mode
SPI.transfer(reg << 1);
SPI.transfer16(data);
SPI.endTransaction();
digitalWrite(MAX98357_LRC, LOW);
}
void max98357_init() {
// Initialize MAX98357 amplifier
max98357_write(0x01, 0x21); // Power up
max98357_write(0x08, 0x00); // Enable DAC
```

```
}
void setup() {
// Set up Serial Monitor
Serial.begin(115200);
Serial.println(" ");
delay(1000);
// Set up I2S
i2s_install();
i2s_setpin();
i2s_start(I2S_PORT);
// Set up MAX98357 amplifier
max98357_setup();
max98357_init();
delay(500);
}
void loop() {
// False print statements to "lock range" on serial plotter display
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 * 2, &bytesIn, portMAX_DELAY);
if (result == ESP_OK) {
int16_t samples_read = bytesIn / 2;
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;
// Write audio data to MAX98357 amplifier
max98357_write(0x0A, static_cast<uint16_t>(mean)); // Amplify the sound (adjust gain)
// Print to serial plotter
Serial.println(mean);
}
}
}
```

OUTPUT