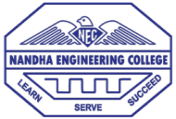


**RFID ATTENDANCE SYSTEM WITH**

**GOOGLE SHEET**

**A PROJECT BASED LEARNING REPORT**

# *Submitted by*

**HARIESH A (22EC023)**

**HARIHARAN T (22EC025)**

## **KUMAR** **T** (22EC046)

## 

*In partial fulfilment for the award of the degree*

*of*

# BACHELOR OF ENGINEERING

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**NANDHA ENGINEERING COLLEGE, ERODE**

**(AUTONOMOUS)**

**(Affiliated to Anna University, Chennai)**

DECEMBER 2024

**BONAFIDE CERTIFICATE**

Certified that this project report “**RFID ATTENDANCE SYSTEM WITH GOOGLE SHEET”** is the bonafide work of “**HARIESH.A (22EC023), HARIHARAN.T (22EC025), KUMAR.T (22EC046)**”, who carried out the Project Based Learning work under my supervision.

###### **SIGNATURE**

Mr.D.Anandakumar ME.,

Assistant Professor

PBL CO-ORDINATOR

Department of ECE

Nandha Engineering College

Erode-52

**ACKNOWLEDGEMENT**

The success of a work depends on team effort and co-operation of various people involved either directly or indirectly.

We express our sincere thanks to our beloved Chairman of Sri Nandha Educational Trust **Thiru.V.Shanmugan**, and our beloved Secretary, **Thiru.S.Thirumoorthi**, of Nandha Engineering College for providing us all the basic amenities to complete the course successfully.

We wish to express our deep sense of gratitude and thanks to our beloved Principal **Dr.U.S.Ragupathy**, for the excellent facilities provided during the course study and project work.

We wish to express our hearty thanks to **Dr.S.Kavitha**, Professor & Dean, **Dr.C.N.Marimuthu**, Professor& Dean (R&D), Department of ECE for providing

Continuous support during the project work.

We are highly indebted to our project guide **Mr.D.Anandakumar**, Assistant Professor, and Mr.M.Srinevasan, Assistant Professor, Project Lab In-charge, Department of ECE, for their guidance and motivation in completing the project.

We thank all the Teaching and Non-Teaching staff members of Electronics and

Communication Engineering Department for their support to our project work. We also express our thanks to our family members and friends for their encouragement, timely help and suggestions.

**TITLE OF CONTENT**

**CHAPTER NO TITLE PAGE NO**

**ABSTRACT** 5

1. **INTRODUCTION** 6
2. **COMPONENTS REQUIRED** 7
   1. HARDWARE REQUIRED 7
      1. ESP32 microcontroller 7
      2. LCD 16x2 7
      3. RFID Reader 9
      4. RFID Tag and card 10
   2. SOFTWARE REQUEIRED 11
      1. Arduino IDE 11
      2. Program Coding 11
3. **BLOCK DIAGRAM** 13
   1. Circuit diagram 13
   2. Working 14
4. **CONCLUSION** 15

**FUTURE SCOPE** 15

**REFERENCE** 16

**APPENDIX** 16

## ABSTRACT

The traditional manual attendance system is very time-consuming, It is insecure and this system can lead to human errors. This system is ineffective as our valuable time and work get wasted in organizing attendance on pen and paper. Hence to overcome this problem we have used a relational database system to store the real-time data of the students. For this project, we used RFID tags and readers to record the attendance of the students. To manipulate and represent the data based on the unique RFID tags, which get fast and easily scanned on the RFID reader. RFID technology is an automatic wireless identification system. This system is used to help the authority manage the attendance of students in a more organized, efficient, and time-saving manner. This particular system has been implemented in a prototype system that uses RFID tags and a reader to calculate attendance which proves its effectiveness over the normal attendance approach. The design of the system is simple, not expensive, and portable to use which makes it good for candidates and also for commercial and academic purposes.

**CHAPTER 1**

## INTRODUCTION

This paper focuses on the implementation of a smart attendance management system using Radio Frequency Identification (RFID ). In recent times technologies have minimized the time consuming process which improves the fast response with high reliability. Hence the idea of RFID emerged to compensate for the requirement in terms of automation in real-time applications. The conventional methods are old enough and are still used in taking student attendance in many institutions and schools. This particular type of attendance system has many loopholes like a student can give fake attendance to friends. If the list is missed they have to take reattendance, which is a very inefficient method. Right now most of the researchers basically concentrating on RFID systems. The main advantage and utility feasibility of RFID is access control. This could be working in miscellaneous environments. All the engineers who are working towards developing their innovative products for real-time applications focus their work on RFID. The conventional methods can also create interruptions as a result of passing a sheet of papers around during academic lectures, conferences presentations, or workshops. It disturbs both student’s and the teacher’s attentiveness and concentration. This in turn reduces the effectiveness of such sessions.Top of Form

Bottom of Form

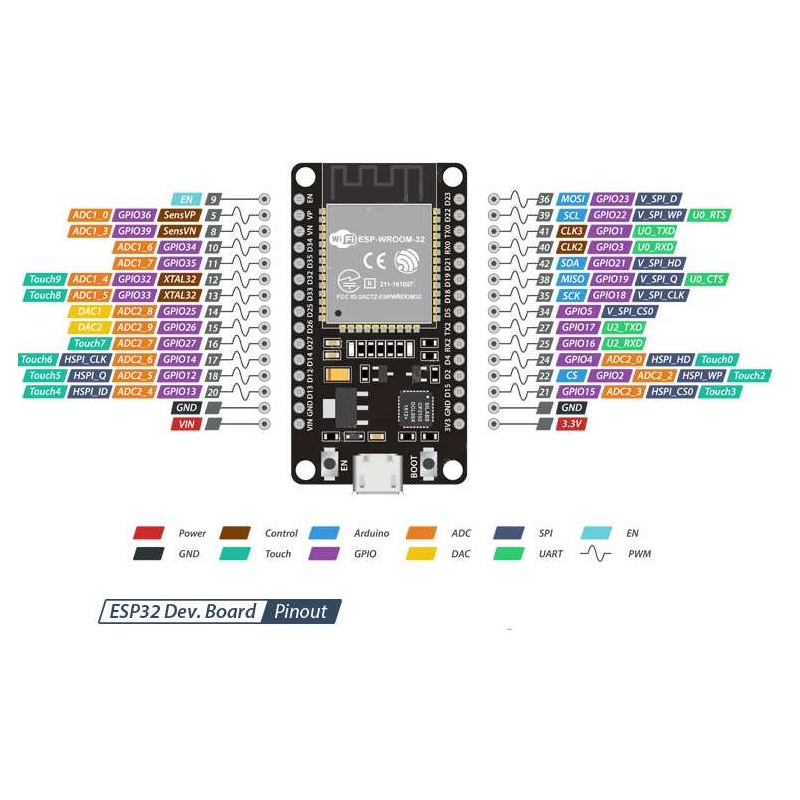
**CHAPTER 2**

**COMPONENTS REQUIRED**

## 2.1. HARDWARE REQUIRED

**2.1.1 ESP32 MICROCONTROLLER:**

The ESP32 microcontroller is a powerful and versatile chip developed by Espressif Systems. It is part of the ESP (Espressif Systems Platform) family, and it has gained significant popularity in the maker, hobbyist, and IoT (Internet of Things) communities due to its impressive features and capabilities. Here are some key aspects of the ESP32 microcontroller. The ESP32 microcontroller is a powerful and versatile chip developed by Espressif Systems. It is part of the ESP (Espressif Systems Platform) family, and it has gained significant popularity in the maker, hobbyist, and IoT (Internet of Things) communities due to its impressive features and capabilities. Here are some key aspects of the ESP32 microcontroller

****

**Fig 2.1.1: ESP32 Pin Configuration**

**2.1.2 LCD 16x2**

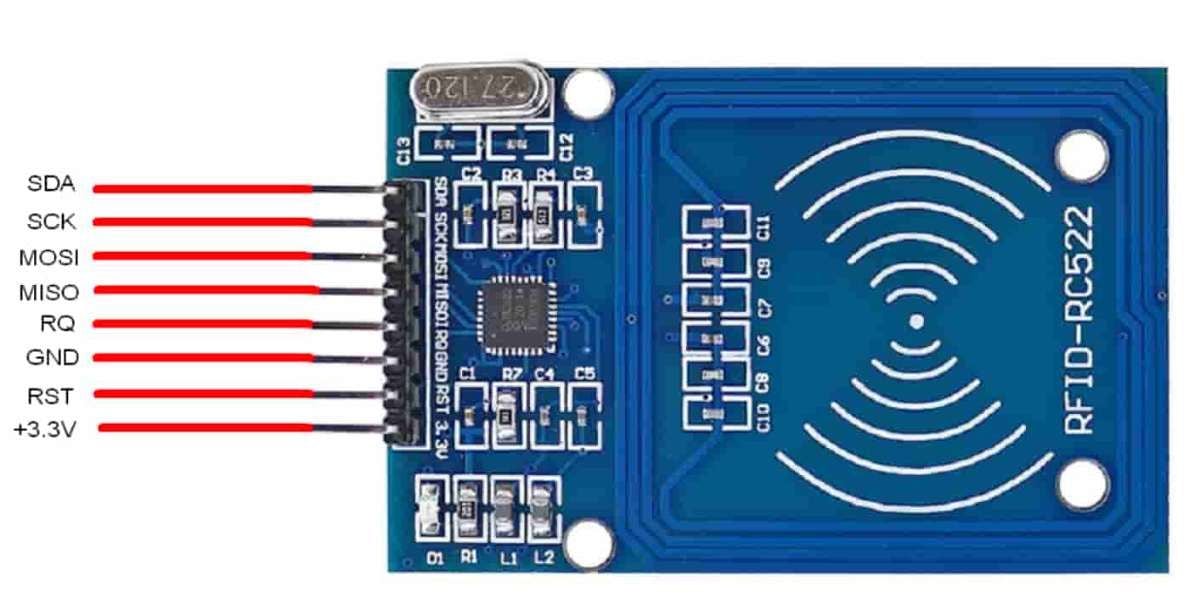
A 16×2 LCD display is a liquid crystal display that can show 16 characters in each of its two rows, providing a total of 32 characters of information.



**Fig 2.1.2: LCD 16x2**

## 2.1.3 RFID Reader

An RFID reader consists of an antenna and a Radio frequency module which basically generates a high-frequency electromagnetic field. As we all know an RFID tag is a passive device which means that it doesn’t have a power supply or a battery. An RFID reader has a microchip that is used to store and process the information. And also it has an antenna that is used to receive to transmit a signal. To read the information on the RFID tag it needs to be placed in very close range of the reader. An RFID reader basically generates an electromagnetic field that causes electrons to move through the tag in the antenna and it powers up the chip.



**fig 2.1.3 RFID RC522**

## 2.1.4 RFID TAGS AND CARDS

The RFID tags are differentiated as passive and active tags. If the device doesn’t have its power supply it is called a passive RFID tag. Thus, The passive tags have to be in very close range of an RFID reader and make use of the radio waves which are broadcasted by the reader to power the response alternatively if the device has its battery power to perform entire operations which are called active RFID tags.



**fig 2.1.4 RFID tags and cards**

## 2.2 SOFTWARE REQUEIRED

### 2.2.1 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a software application that provides a user-friendly platform for programming and uploading code to Arduino microcontrollers. It is a crucial tool for hobbyists, students, and professionals working with Arduino boards to create a wide range of electronic projects. The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

### 2.2.2 PROGRAM CODING

#include <Arduino.h>

#include <WiFi.h>

#include <SPI.h>

#include <MFRC522.h>

#include <HTTPClient.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

//------------------------------------------------------------------------------------------------

// Enter Google Script Deployment ID:

const char \*GScriptId = "AKfycby9RpfJnJBHVu3VGgf437m5p0ke5L80z5uViNamGVWXEegEFpnV95XzX8x8mGYLBd8adw";

//------------------------------------------------------------------------------------------------

// Enter network credentials:

const char\* ssid = "\*\*\*\*\*\*\*\*";

const char\* password = "\*\*\*\*\*\*\*\*\*";

//------------------------------------------------------------------------------------------------

// Enter command (insert\_row or append\_row) and your Google Sheets sheet name (default is Sheet1):

String payload\_base = "{\"command\": \"insert\_row\", \"sheet\_name\": \"Sheet1\", \"values\": ";

String payload = "";

//------------------------------------------------------------------------------------------------

// Google Sheets setup (do not edit)

const char\* host = "script.google.com";

String url = String("/macros/s/") + GScriptId + "/exec";

//------------------------------------------------------------

// Declare variables that will be published to Google Sheets

String student\_Rollno;

//------------------------------------------------------------

int blocks[] = {4,5,6,8,9};

#define total\_blocks (sizeof(blocks) / sizeof(blocks[0]))

//------------------------------------------------------------

#define RST\_PIN 2 // Customize these pins as per your ESP32 setup

#define SS\_PIN 4

#define BUZZER 14

//------------------------------------------------------------

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

MFRC522::MIFARE\_Key key;

MFRC522::StatusCode status;

//------------------------------------------------------------

/\* Be aware of Sector Trailer Blocks \*/

int blockNum = 2;

/\* Create another array to read data from Block \*/

/\* Length of buffer should be 2 Bytes more than the size of Block (16 Bytes) \*/

byte bufferLen = 18;

byte readBlockData[18];

//------------------------------------------------------------

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* setup Function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void setup() {

//----------------------------------------------------------

pinMode(BUZZER, OUTPUT);

Serial.begin(115200);

delay(10);

Serial.println('\n');

//----------------------------------------------------------

SPI.begin();

//----------------------------------------------------------

// Initialize LCD screen

lcd.init();

// Turn on the backlight

lcd.backlight();

lcd.clear();

lcd.setCursor(0,0); //col=0 row=0

lcd.print("Connecting to");

lcd.setCursor(0,1); //col=0 row=0

lcd.print("WiFi...");

//----------------------------------------------------------

// Connect to WiFi

WiFi.begin(ssid, password);

Serial.print("Connecting to ");

Serial.print(ssid); Serial.println(" ...");

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

Serial.println('\n');

Serial.println("Connection established!");

Serial.print("IP address:\t");

Serial.println(WiFi.localIP());

//----------------------------------------------------------

lcd.clear();

lcd.setCursor(0,0); //col=0 row=0

lcd.print("Connecting to");

lcd.setCursor(0,1); //col=0 row=0

lcd.print("NEC website...");

delay(5000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* loop Function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void loop() {

//----------------------------------------------------------------

lcd.clear();

lcd.setCursor(0,0); //col=0 row=0

lcd.print("Scan your ID");

lcd.setCursor(0,1);

lcd.print("Card or Tag...");

/\* Initialize MFRC522 Module \*/

mfrc522.PCD\_Init();

/\* Look for new cards \*/

/\* Reset the loop if no new card is present on RC522 Reader \*/

if ( ! mfrc522.PICC\_IsNewCardPresent()) {return;}

/\* Select one of the cards \*/

if ( ! mfrc522.PICC\_ReadCardSerial()) {return;}

/\* Read data from the same block \*/

Serial.println();

Serial.println(F("Reading last data from RFID..."));

//----------------------------------------------------------------

String values = "", data;

// Creating payload

for (byte i = 0; i < total\_blocks; i++) {

ReadDataFromBlock(blocks[i], readBlockData);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if(i == 0){

data = String((char\*)readBlockData);

data.trim();

student\_Rollno = data;

values = "\"" + data + ",";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

else if(i == total\_blocks-1){

data = String((char\*)readBlockData);

data.trim();

values += data + "\"}";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

else{

data = String((char\*)readBlockData);

data.trim();

values += data + ",";

}

}

//----------------------------------------------------------------

// Create json object string to send to Google Sheets

payload = payload\_base + values;

//----------------------------------------------------------------

lcd.clear();

lcd.setCursor(0,0); //col=0 row=0

lcd.print("Storing Data");

lcd.setCursor(0,1); //col=0 row=0

lcd.print("Please Wait...");

//----------------------------------------------------------------

// Publish data to Google Sheets

Serial.println("Publishing data...");

Serial.println(payload);

if(WiFi.status() == WL\_CONNECTED) {

HTTPClient http;

http.begin("https://" + String(host) + url);

http.addHeader("Content-Type", "application/json");

int httpCode = http.POST(payload);

if(httpCode > 0) {

if(httpCode == HTTP\_CODE\_OK) {

lcd.clear();

lcd.setCursor(0,0); //col=0 row=0

lcd.print("Roll no: "+student\_Rollno);

lcd.setCursor(0,1); //col=0 row=0

lcd.print("Present");

digitalWrite(BUZZER, HIGH);

delay(2000);

digitalWrite(BUZZER, LOW);

delay(2000);

}

} else {

Serial.println("Error on HTTP request");

lcd.clear();

lcd.setCursor(0,0); //col=0 row=0

lcd.print("Failed.");

lcd.setCursor(0,1); //col=0 row=0

lcd.print("Try Again");

}

http.end(); // Free resources

}

//----------------------------------------------------------------

// A delay of several seconds is required before publishing again

delay(5000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ReadDataFromBlock() function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ReadDataFromBlock(int blockNum, byte readBlockData[])

{

//----------------------------------------------------------------------------

/\* Prepare the key for authentication \*/

/\* All keys are set to FFFFFFFFFFFFh at chip delivery from the factory \*/

for (byte i = 0; i < 6; i++) {

key.keyByte[i] = 0xFF;

}

//----------------------------------------------------------------------------

/\* Authenticating the desired data block for Read access using Key A \*/

status = mfrc522.PCD\_Authenticate(MFRC522::PICC\_CMD\_MF\_AUTH\_KEY\_A, blockNum, &key, &(mfrc522.uid));

if (status != MFRC522::STATUS\_OK){

Serial.print("Authentication failed for Read: ");

Serial.println(mfrc522.GetStatusCodeName(status));

return;

} else {

Serial.println("Authentication success");

}

//----------------------------------------------------------------------------

/\* Reading data from the Block \*/

status = mfrc522.MIFARE\_Read(blockNum, readBlockData, &bufferLen);

if (status != MFRC522::STATUS\_OK) {

Serial.print("Reading failed: ");

Serial.println(mfrc522.GetStatusCodeName(status));

return;

} else {

readBlockData[16] = ' ';

readBlockData[17] = ' ';

Serial.println("Block was read successfully");

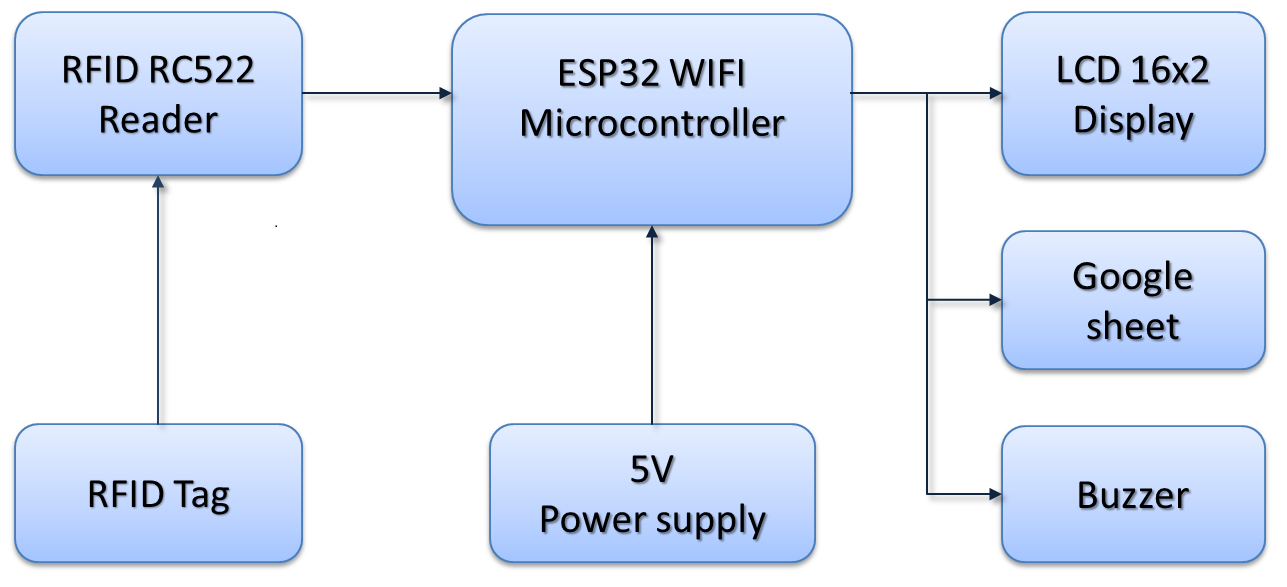
}

//----------------------------------------------------------------------------

}

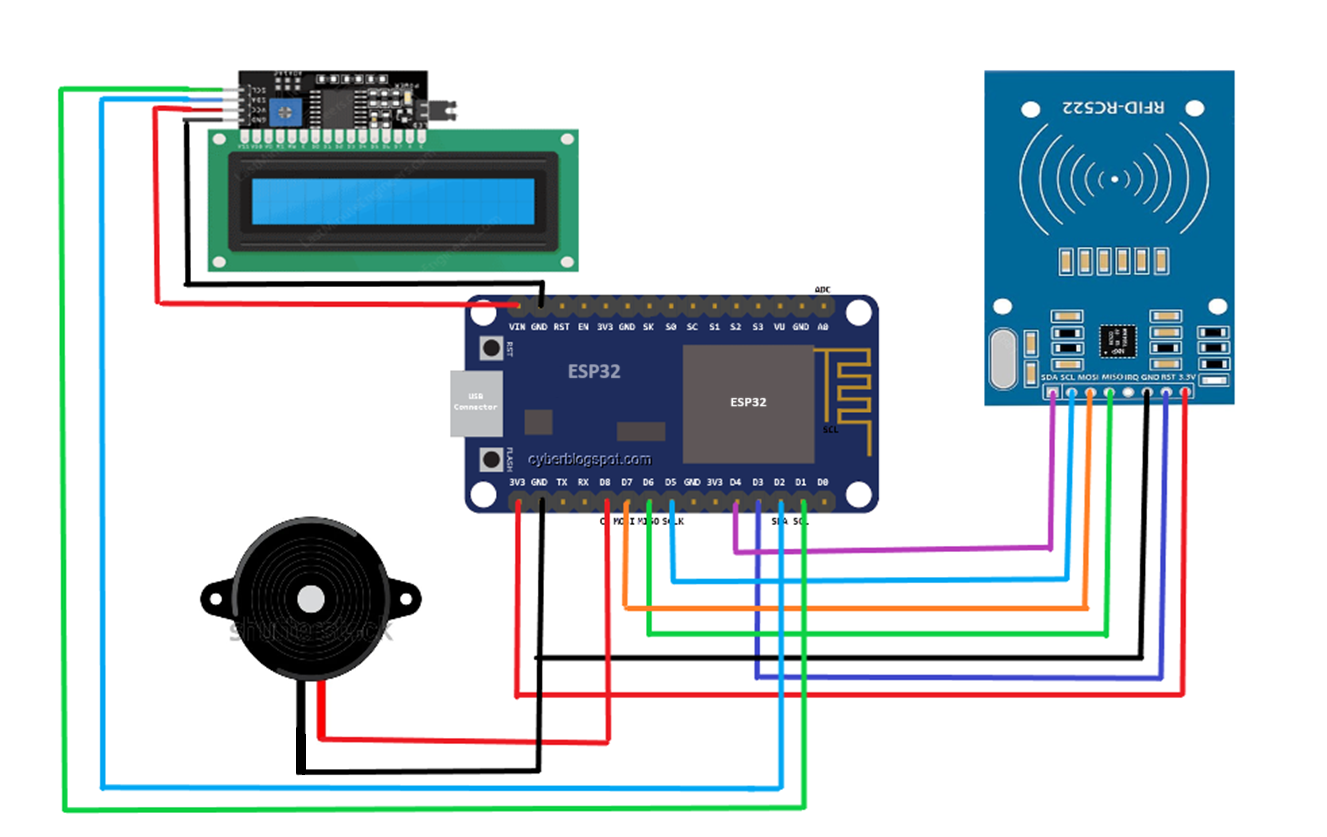
**CHAPTER 3**

## 3.1 BLOCK DIAGRAM OF PROPOSED SYSTEM



**fig 3.1 Block diagram of proposed system**

* The following fig shows the block diagram for our project Smart attendance system using RFID. Which mainly consists of ESP32, RFID Cards RFID readers, A breadboard, 16 x 2 LCD display.
* Here ESP32 acts as a Central processing unit (CPU) for controlling all the input/output components. For this project, we have used a 5v power supply to power up the ESP32 and other components.
* RFID reader module is interfaced with ESP32 to read the data from RFID cards/tags.16 x 2 LCD display is used to display the real- time attendance of the students /employee and the permanent attendance is stored on a google sheet using IoT.
* In this system, a student or employee has to place /put his card on an RFID reader. When the RFID reader reads the data it directly transfers the data to ESP32 and the real-time attendance will be displayed on a 16 x 2 LCD display and the permanent attendance is stored on a google sheet.

 **3.2 CIRCUIT DIAGRAM OF PROPOSED SYSTEM**

**fig 3.2 Circuit diagram of proposed system**

This system consists of 3 codes. 1) First one is the spreadsheet code which is written on the back end of the spreadsheet 2)Second one is the code in which we will register the card by loading his/her information. That means that the card will be permanently assigned to the student/employee. 3) The third code is the code through which we will scan our id card and then the attendance will be marked on the spreadsheet via ESP32.

In the second code, we will write the information which we want to save in the new fresh card. When the information is written in the code we simply have to upload the code. When the uploading is done we have to take our new fresh card close to the RFIDRC522 module once the information is loaded in the card the serial monitor will show that ”The Block is successfully Read” and ”Data is stored in Block No 2”. With this we have registered one card, same process is repeated to create the remaining cards, just change the information in the code, upload the code take a new fresh card, and scan.

After that in the third code, we have pasted a link to the google spreadsheet which is deployed/generated when we run the google spreadsheet code. So after registering the card we have to just simply upload the third code when the code is successfully uploaded we have to take our “Registered Card” close to the RFID-RC522 module once the card has scanned the buzzer will immediately beep for a few seconds and the serial monitor will give us the link and the HTTPS code. If the HTTPS code is 302 that means our data is successfully read on the spreadsheet.

**CHAPTER 4**

## CONCLUSION

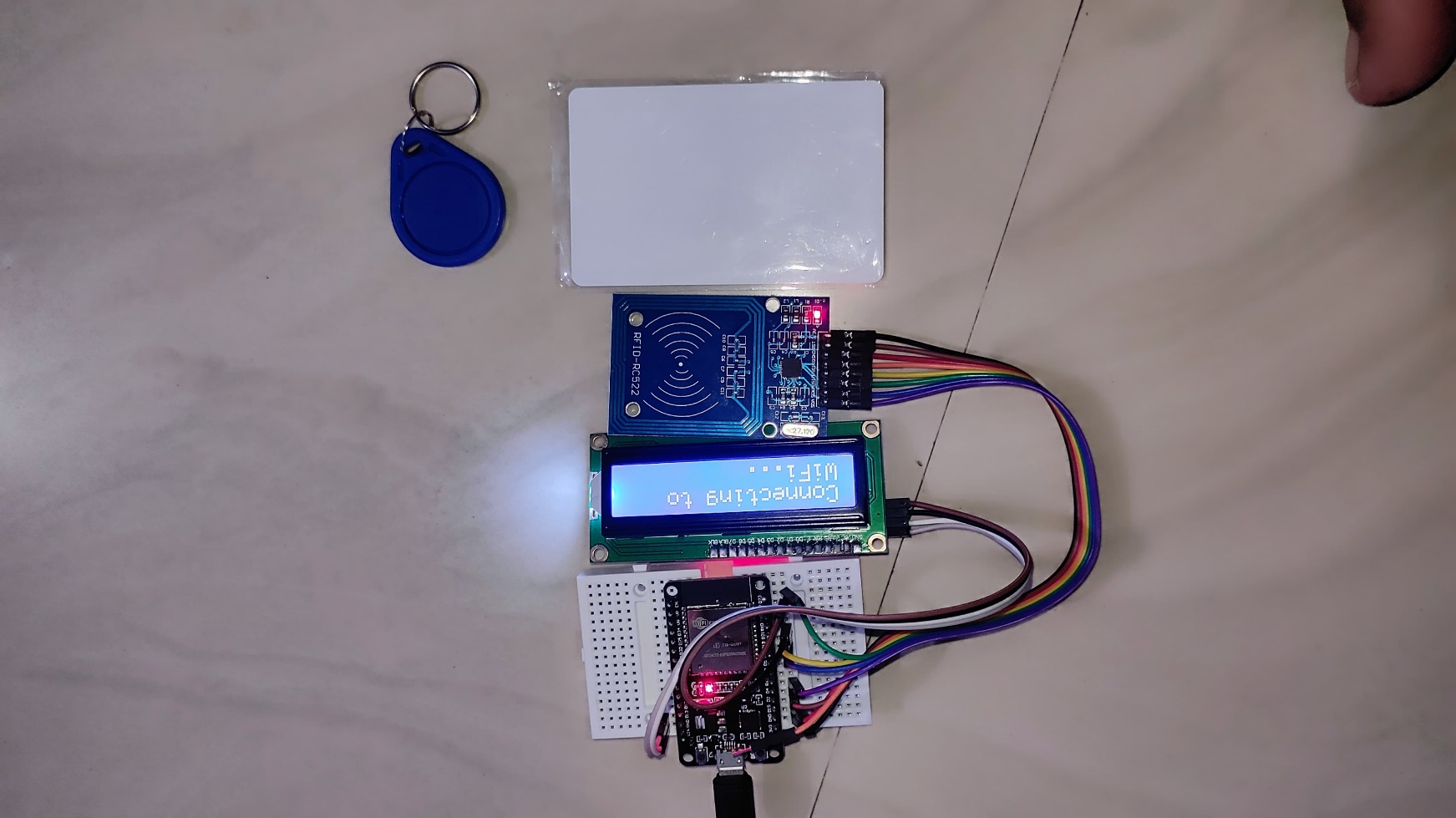
In conclusion, the integration of a robotic arm with the ardiuno UNO microcontroller represents a powerful and versatile solution for achieving precise and controlled motion in various applications. The combination of the robotic arm's mechanical components and the ardiuno UNO's computational capabilities opens up a wide range of possibilities for automation, remote control, and customization.

## FUTURE SCOPE

* Nothing is perfect in this world. We all are also no exception. Although, we have tried to present the system in modern technology in small scale and smart way.
* In further future, there can be more enhancement by developing a mobile app. To send SMS for alert notifications, we can also use GSM modules according to the hardware need. Things peak can be used for data analysis.
* To reduce the misuse of RFID tags, biometric technology like iris sensors or fingerprint sensors or image processing can be used for unique identity to improve authorization.

**APPENDIX**

**PROTOTYPE MODEL FOR RFID ATTENDANCE SYSTEM WITH GOOGLE SHEET**



**fig 4.4 Top view**