List of figures

List of Tables

Contents

Chapter 1

INTRODUCTION

Information Technology (IT) has played a significant role in developing several aspects in academic sectors and domains such as student monitoring and management systems.

Attendance or daily register of understudies has turned into a vital assessment perspective in the current instructive framework in both universities and schools. The conventional attendance monitoring framework has a few impediments with the trend and the technology gap. For instance, passing the everyday attendance sheet to a huge number of students in a class is extremely risky and it hampers the consideration of the students in the class. It is waste of time as well as a student can deliberately enlist counterfeit attendance record in the day-by-day attendance sheet. On the off chance, if the teacher loses these documents, all the significant attendance records are lost without doubt.

Therefore, it is a critical subject to tracking and manages student’s attendance in school, college, and university environment. Since it can be helped to urge students to attend on time, amend the efficiency of the learning, increase learning grade, and finally boosting and improving the education level. So, there is a need to manage the student attendance records automatically by using information technology management system in a faculty to assist the maintaining attendance. Biometrics techniques are used to verify identification through their characteristics like face recognition, signatures, fingerprint, voice recognition, irises, barcode, Bluetooth, Near-Field Communication (NFC), RFID and so on.

RFID innovation has a tremendous task to carry out in the completion of the vision of associating objects around us to the internet. These items extend from huge structures, modern plants, planes, vehicles, machineries, any sort of merchandise, and explicit pieces of a bigger framework to people, animals and plants and even explicit body portions of them. The idea driving this is called Internet of Things (IoT).

1.1 RFID Technology: RFID is standard for Radio Frequency Identification which is the very latest concept of Internet of Things (IoT) and it is very similar technology of barcode system but with some higher advanced concept. It works by using transferring and receiving signal using Antenna and Integrated Circuit. It has two parts namely, RFID Tag and RFID Reader.

1.2 RFID Tag: A RFID Tag is an electronic tag that exchanges information with a RFID reader through radio waves. Almost every RFID Tags have two parts namely, Antenna and Integrated Circuit (IC). Antenna used for receives radio frequency waves and IC used for processing and store data.

1.3 RFID Reader: RFID reader is a device which used to gather information from RFID tag which used to track individual. RFID uses radio waves to transfer the data from tag to reader.

1.4 IOT (Internet of Things): The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable.

1.5 Arduino: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Chapter 2

LITERATURE REVIEW

RFID-based attendance systems are widely recognized for their efficiency and accuracy in tracking attendance. These systems use RFID tags and readers to automatically capture attendance data, which is then processed by a microcontroller. Among the microcontrollers available, the ESP8266 stands out due to its low cost, built-in Wi-Fi capabilities, and ease of integration with various IoT applications.

The ESP8266's ability to connect to the internet allows real-time data transfer, making it an ideal choice for IoT-based attendance systems. By interfacing the ESP8266 with RFID technology, attendance data can be automatically recorded and transmitted to cloud storage platforms, such as Google Sheets. This integration enables real-time monitoring and easy accessibility of attendance records from anywhere with internet access.

Google Sheets, a widely used web-based spreadsheet application, provides a user-friendly platform for storing and analyzing data. When combined with the ESP8266 and RFID technology, Google Sheets offers a convenient solution for automating attendance management. This approach not only reduces the need for manual data entry but also enhances data accuracy and accessibility.

Existing literature supports the use of RFID for contactless and accurate attendance tracking and highlights the ESP8266's versatility in IoT projects. Studies have demonstrated successful integrations of similar systems, underscoring the practicality and effectiveness of this approach. However, challenges such as connectivity issues and data security need to be carefully managed to ensure reliable system performance.

In summary, the integration of ESP8266 with RFID technology and Google Sheets represents an innovative approach to modern attendance systems, combining the benefits of automation, cloud storage, and real-time data access.

2.1 PROBLEM STATEMENT

Passing the everyday attendance sheet to a huge number of students in a class is risky as students can miss their attendance call and it can hamper students' attendance percentage. It is waste of time as well as a student can give a false attendance and cheat on the record in the day-by-day attendance sheet. If the teacher loses these documents, all the significant attendance records are lost without doubt.

2.2 STUDY OBJECTIVE

- To provide very higher accuracy and speed than a traditional paper-based system
- To provide security and easy tracking of data
- Implement fully automatic process

Chapter 3

SYSTEM OVERVIEW

COMPONENTS AND SUPPLIES

The proposed system consists of different components, a brief introduction about components is given below.

- NODEMCU ESP8266 board
- RC522 RFID Reader
- 16x2 I2c LCD Display
- RFID Tags
- Buzzer
- Bread Board
- Jumper Wires

3.1 NODEMCU ESP8266 board

- In this project, we have used NodeMcu which is an opensource platform
- NodeMcu is based on ESP8266 which can be used to connect objects and transfer data using Wifi
- In this project, the NodeMCU is used to send the attendance on the spreadsheet via the wifi module. Whatever the google spreadsheet code has generated the deployment link, we have to copy that link and paste it into the code and with the help, nodeMCU will send the data on the spreadsheet.

Working Principle of NODEMCU ESP8266 board:

The NodeMCU ESP8266 board is a low-cost, open-source IoT platform that combines a microcontroller with integrated Wi-Fi capabilities, making it ideal for IoT applications. The NodeMCU ESP8266 board works by leveraging its integrated Wi-Fi, GPIO capabilities, and microcontroller functions to execute user-defined tasks, connect to the internet, and interact with other devices, making it a versatile platform for IoT applications.

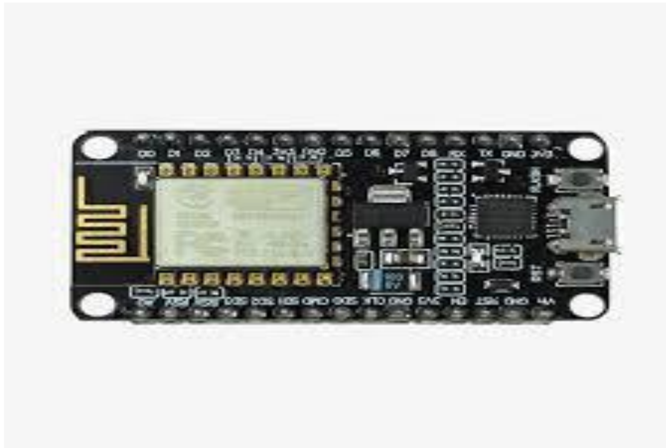


Figure 3.1: NODEMCU ESP8266 Board

3.2 RC522 RFID Reader

- Designed for low-power applications, the RC522 consumes relatively little power, making it suitable for battery-operated devices and IoT projects.
- The RC522 RFID Reader operates at a frequency of 13.56 MHz, which is commonly used for short-range communication with RFID tags.
- It uses the Serial Peripheral Interface (SPI) for communication with microcontrollers. It can also support I2C and UART communication, providing flexibility in interfacing with different devices.

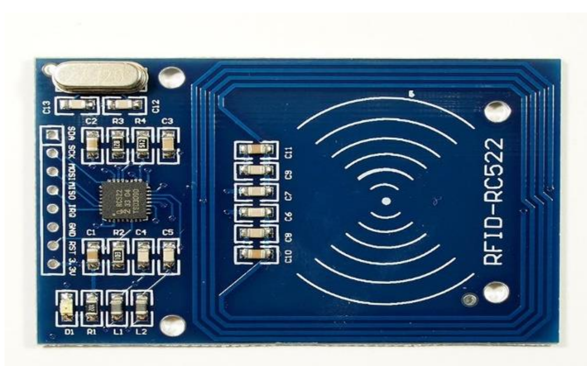


Figure 3.2: RC522 RFID Reader

3.3 16x2 I2c LCD Display

- Liquid crystal display is also used to display the Name, Time in and Time out of the authorized students and to display error message for unauthorized access
- The 16x2 I2C LCD typically operates at 5V, making it compatible with most microcontrollers like Arduino and ESP8266. Some versions can also operate at 3.3V
- The 16x2 LCD is often used to create simple user interfaces, allowing users to see information like sensor readings, menu options, or system status
- The 16x2 I2C LCD Display is a versatile and user-friendly module that provides a simple and efficient way to add text output to microcontroller projects

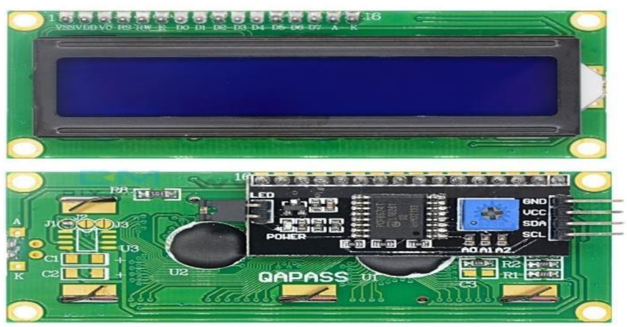


Figure 3.3: 16x2 I2c LCD Display

3.4 RFID Tags

- 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.

Figure 3.4: RFID Tags

3.5 Buzzer

- This buzzer also has an important role in this module
- Whenever the user will scan his/her card then this buzzer beeps for a few seconds so that the one who scanned the card will know that his card is scanned properly
- Without a buzzer, one can only assume the card has been detected but he/she will not be sure so the buzzer is important here

3.6 Bread Board

- A breadboard is a crucial tool in electronics prototyping, providing a platform for quickly and easily building and testing electronic circuits without soldering
- It allows components to be inserted into interconnected holes to create circuits
- Electronic components such as resistors, capacitors, integrated circuits (ICs) and wires are inserted into breadboard's holes
- Components can be connected by placing them in the same row or column, or by using jumper wires to bridge connections between different rows and columns

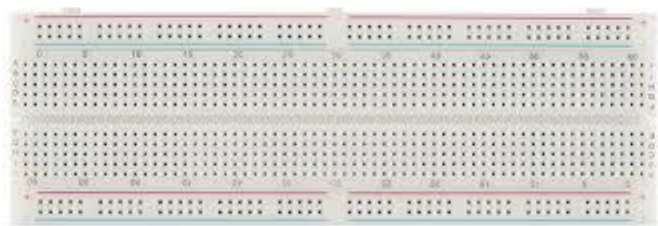


Figure 3.6: Bread Board

3.7 Jumper Wires

- Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female
- The difference between each is in the end point of the wire
- Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering
- Jumper wires are typically used with breadboard and other prototyping tools in order to make it easy to change a circuit as needed

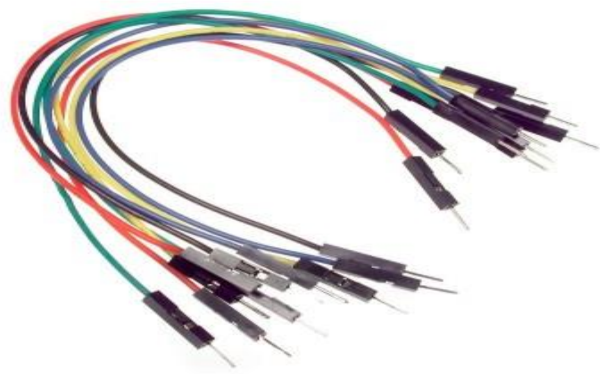


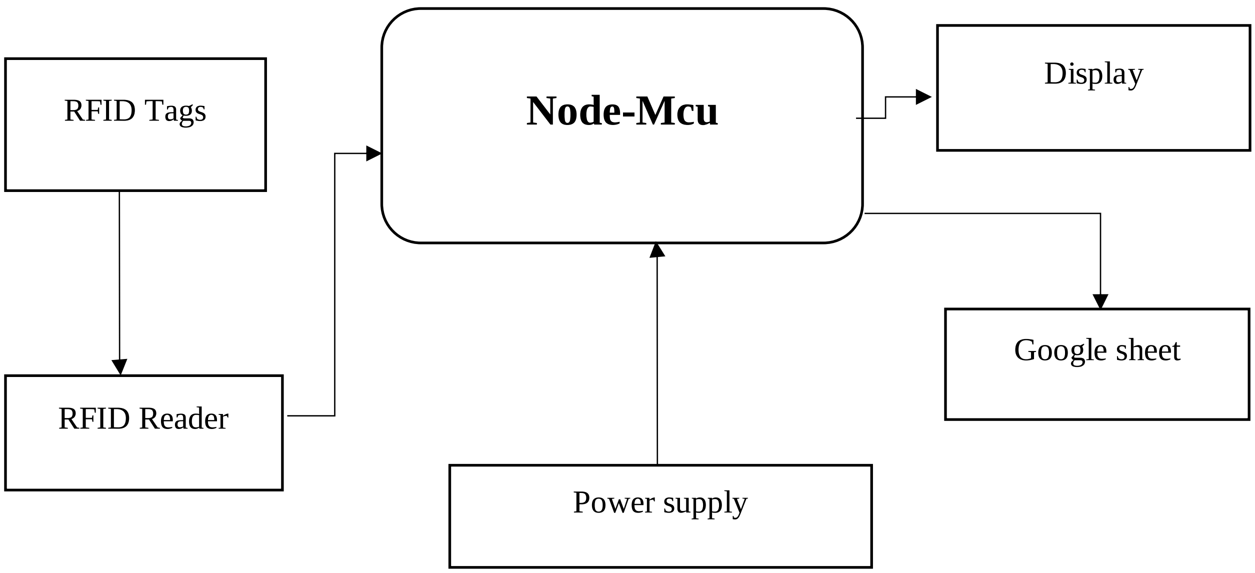
Figure 3.7: Jumper wires

Chapter 4

METHODOLOGY

4.1 BLOCK DIAGRAM

A block diagram is used to represent a control system in diagram form. In other words, the practical representation of a control system is its block diagram. Each element of the control system is represented with a block and the block is the symbolic representation of the transfer function of that element.



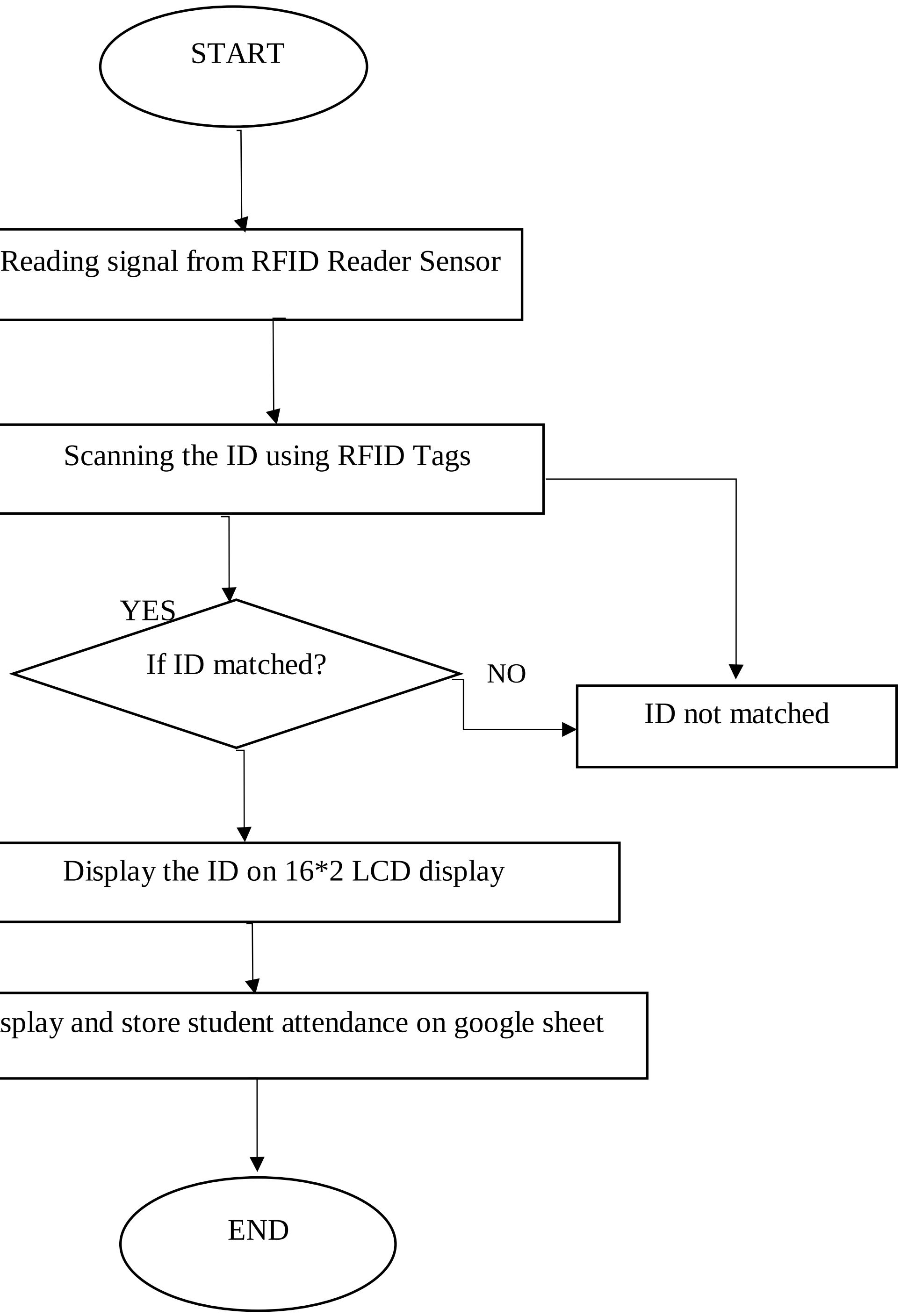
This is the block diagram of the project RFID-based Attendance System using Node-MCU, RFID. Here Node-MCU acts as a central processor for controlling all other components as input/output units.

The function of each block in the block diagram above is as follows:

- **RFID Reader:** the input block consists of an RFID reader, and the tag data card that the reader detects will be sent to the microcontroller
- Block microcontroller, data processor and central controller of the system
- Once it is verified by the microcontroller, data is stored in the memory unit

4.2 FLOW CHART

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.



4.3 PIN DIAGRAM

A pin out is a reference to the pins or contacts that connect an electrical device or connector. It describes the functions of transmitted signals and the circuit input/output (I/O) requirements. Each individual pin in a chip, connector or singular wire is defined in text, a table or a diagram.

Pin Out

A pin out is a reference to the pins or contacts that connect an electrical device or connector. It describes the functions of transmitted signals and the circuit input/output (I/O) requirements. Each individual pin in a chip, connector or singular wire is defined in text, a table or a diagram.

PIN Out of different modules:

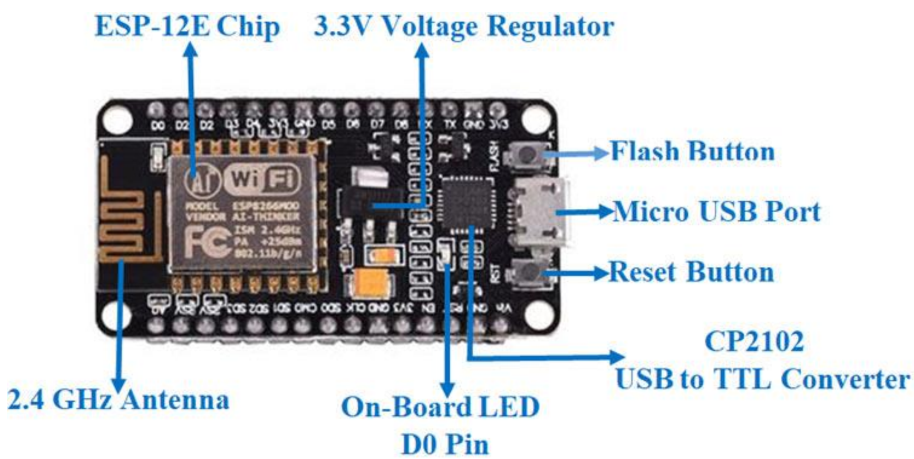


Figure 4.3: Pin Out of NODEMCU ESP8266 board

The NodeMCU ESP8266 board has several pins that can be used for various functions, such as digital input/output, analog input, power supply, and communication interfaces. Here's a breakdown of the pinout:

1. Power Pins:

- **Vin:** This pin can be used to supply an external power source (5V) to the NodeMCU
- **3V3:** Provides a regulated 3.3V output from the onboard voltage regulator. It can also power external components

2. Digital I/O Pins:

- **D0 to D8:** These are digital input/output pins. They can be configured as either input or output and can be used to read sensors, control LEDs, relays, etc
- **D3 (GPIO 0), D4 (GPIO 2):** These pins have additional functions, such as the boot mode. Care should be taken when using them
- **D9 (SD2) and D10 (SD3):** These pins are typically used for the onboard flash memory and should be used cautiously

3. Analog Input:

- **A0:** The analog input pin on the NodeMCU. It can read analog signals with a voltage range of 0 to 1V, though it can be extended with external voltage dividers

4. Communication Pins:

- **TX (D10 / GPIO1):** Transmit pin for UART communication
- **RX (D9 / GPIO3):** Receive pin for UART communication
- **SPI:**
    - **SD1 (GPIO 10):** Typically used as Master-In Slave-Out (MISO)
    - **CMD (GPIO 11):** Typically used as Master-Out Slave-In (MOSI)
    - **SD0 (GPIO 12):** Typically used as Serial Clock (SCLK)
    - **SS (GPIO 15):** Typically used as Slave Select (SS)

5. Other Important Pins:

- **RST:** Reset pin. Bringing this pin low will reset the microcontroller
- **EN:** Enable pin. This is used to enable or disable the 3.3V regulator. It must be held high for the board to operate

6. Special Function Pins:

- **GPIO 16 (D0**): Can be used to wake up the ESP8266 from deep sleep mode. It is also connected internally to the onboard LED (usually marked as D0)

RFID-RC522:

The RC522 is a 13.56MHz RFID module that is based on the MFRC522 controller from NXP semiconductors. The module can support I2C, SPI and UART and normally is shipped with a RFID card and key fob. It is commonly used in attendance systems and other person/object identification applications.

Figure 4.4: Pin Out of RFID-RC522

RC522 Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	Used to Power the module, typically 3.3V is used
2	RST	Reset pin – used to reset or power down the module
3	Ground	Connected to Ground of system
4	IRQ	Interrupt pin – used to wake up the module when a device comes into range
5	MISO/SCL/TX	MISO pin when used for SPI communication, acts as SCL for I2c and TX for UART.
6	MOSI	Master out slave in pin for SPI communication
7	SCK	Serial Clock pin – used to provide clock source
8	SS/SDA/Rx	Acts as Serial input (SS) for SPI communication, SDA for IIC and Rx during UART

16x2 I2c LCD Display Pin Configuration:

The 16x2 I2C LCD Display simplifies the connection process by using the I2C communication protocol, which reduces the number of pins required. 16x2 LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability, programmer friendly and available educational resources.

Figure 4.5: Pin Out of 16x2 I2c LCD Display

Here's the pin configuration for a typical 16x2 I2C LCD Display module:

Pin No:	Pin Name:	Description
1	Vss (Ground)	Ground pin connected to system ground
2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V)
3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.
4	Register Select	Connected to Microcontroller to shift between command/data register
5	Read/Write	Used to read or write data. Normally grounded to write data to LCD
6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement
7	Data Pin 0	Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data. These LCD's can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free.

4.4 SOFTWARE IMPLEMETATION

Arduino IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and displays errors. The console displays text output by the Arduino environment including complete error messages and other information.

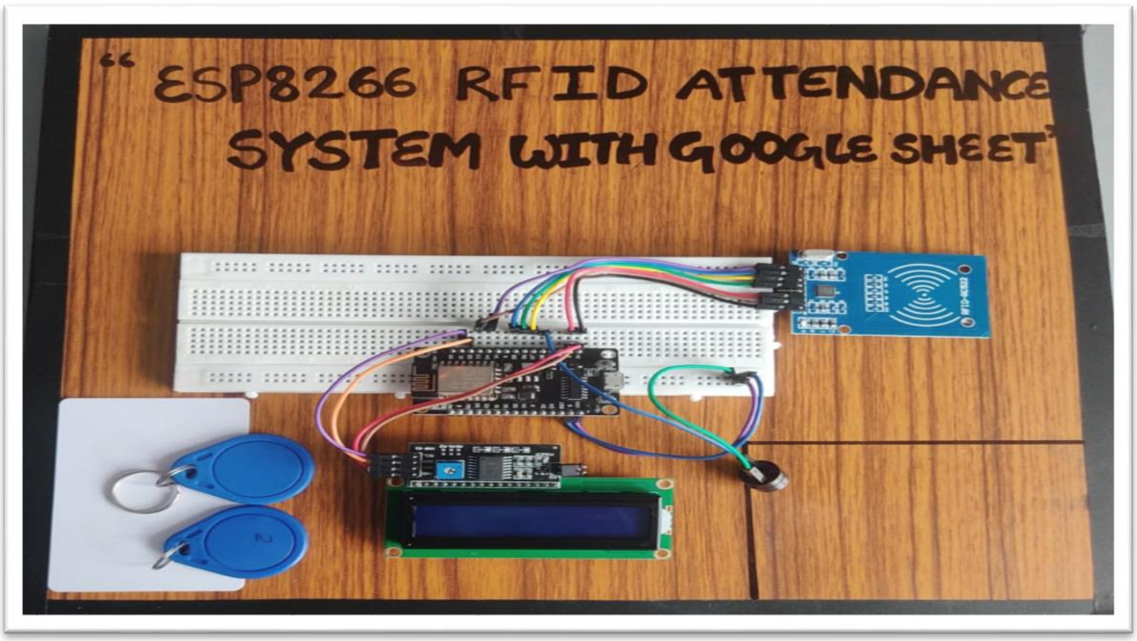
4.5 Working of the system

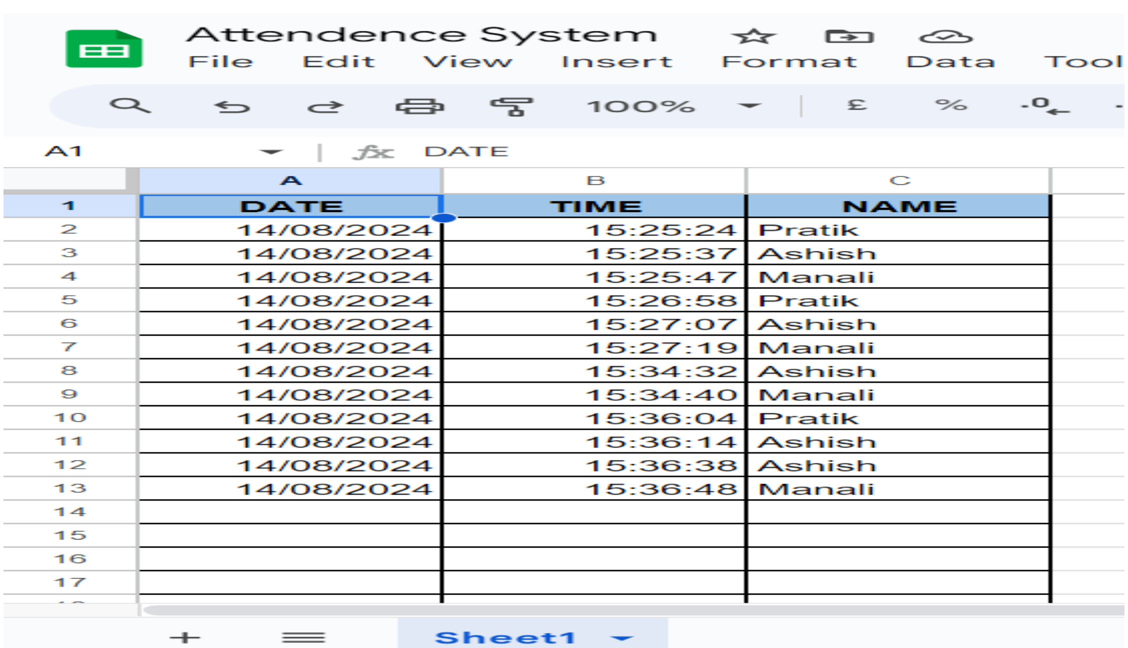
Student Attendance system is used to mark the attendance of students by recording the in time and out time of the students. It is included in colleges, school for students to get their attendance. RFID cards come in the size of credit card which is in white color. A student will be given RFID card and RFID reader will be placed on the door or the entry gate of school or college. Whenever students want to enter in the school/college, he/she must show the RFID card to the reader, student has to take the RFID card near to the RFID reader. RFID Reader will mark the attendance by fetching the RFID card number and swiped in time.

Chapter 5

OUTCOME/RESULT

The attendance of the students is stored in the memory unit. Attendance is stored in numerical format with different unique ID. Date & Time along with RFID unique ID is stored.





	DATE	TIME	NAME
2	14/08/2024	15:25:24	Pratik
3	14/08/2024	15:25:37	Ashish
4	14/08/2024	15:25:47	Manali
5	14/08/2024	15:26:58	Pratik
6	14/08/2024	15:27:07	Ashish
7	14/08/2024	15:27:19	Manali
8	14/08/2024	15:34:32	Ashish
9	14/08/2024	15:34:40	Manali
10	14/08/2024	15:36:04	Pratik
11	14/08/2024	15:36:14	Ashish
12	14/08/2024	15:36:38	Ashish
13	14/08/2024	15:36:48	Manali

SOURCE CODE:

1.RFID_ ATTENDANCE_GOOGLESHEETS CODE:

```
//Smart Attendance System with Google Sheets and LCD Display

#include <SPI.h>

#include <MFRC522.h>

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <WiFiClient.h>

#include <WiFiClientSecureBearSSL.h>

#include <LiquidCrystal_I2C.h>

#define RST_PIN  D3

#define SS_PIN   D4

#define BUZZER   D8

MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

MFRC522::StatusCode status;

int blockNum = 2;

byte bufferLen = 18;

byte readBlockData[18];

String card_holder_name;

const String sheet_url = "'1W9-yrbi3MPzjr-WY5EJkTCEMQeAl4n3xnvM-Lr-cUwo";  //Enter Google
Script URL
```

```
#define WIFI_SSID "Aman’s Vivo"  //Enter WiFi Name

#define WIFI_PASSWORD "Aman@2124"  //Enter WiFi Password

//Initialize the LCD display

LiquidCrystal_I2C lcd(0x3F, 16, 2);  //Change LCD Address to 0x27 if 0x3F doesnt work

void setup()

{

  Serial.begin(9600);

  //Serial.setDebugOutput(true);

  lcd.begin();

  lcd.backlight();

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("  Initializing  ");

  for (int a = 5; a <= 10; a++) {

    lcd.setCursor(a, 1);

    lcd.print(".");

    delay(500);

  }

  Serial.println();

  Serial.print("Connecting to AP");
```

```
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(200);
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  pinMode(BUZZER, OUTPUT);
  SPI.begin();
}
 void loop()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Scan your Card ");
  mfrc522.PCD_Init();
  if ( ! mfrc522.PICC_IsNewCardPresent()) {return;}
  if ( ! mfrc522.PICC_ReadCardSerial()) {return;}
  Serial.println();
```

```
Serial.println(F("Reading last data from RFID..."));
ReadDataFromBlock(blockNum, readBlockData);
//mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
Serial.println();
Serial.print(F("Last data in RFID:"));
Serial.print(blockNum);
Serial.print(F(" --> "));
for (int j=0 ; j<16 ; j++)
{
  Serial.write(readBlockData[j]);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Hey " + String((char*)readBlockData) + "!");
  //lcd.print(String((char*)readBlockData));
  //lcd.print("!")
}
Serial.println();
digitalWrite(BUZZER, HIGH);
delay(200);
digitalWrite(BUZZER, LOW);
delay(200);
digitalWrite(BUZZER, HIGH);
```

```
delay(200);

digitalWrite(BUZZER, LOW);

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

  std::unique_ptr<BearSSL::WiFiClientSecure>client(new BearSSL::WiFiClientSecure);

  //client->setFingerprint(fingerprint);

  client->setInsecure();

  card_holder_name = sheet_url + String((char*)readBlockData);

  card_holder_name.trim();

  Serial.println(card_holder_name);

  HTTPClient https;

  Serial.print(F("[HTTPS] begin...\n"));

  if (https.begin(*client, (String)card_holder_name)){

    Serial.print(F("[HTTPS] GET...\n"));

    int httpCode = https.GET();

    // httpCode will be negative on error

    if (httpCode > 0) {

      Serial.printf("[HTTPS] GET... code: %d\n", httpCode);

  lcd.setCursor(0, 1);

  lcd.print(" Data Recorded ");

  delay(2000);

    }

    else
```

```
    {Serial.printf("[HTTPS] GET... failed, error: %s\n", https.errorToString(httpCode).c_str());}
    https.end();
    delay(1000);
   }
   else {
    Serial.printf("[HTTPS} Unable to connect\n");
   }
  }
}
void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  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");
```

```
  }
  status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Reading failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  else {
    Serial.println("Block was read successfully");
  }
}
```

2.RFID tag name code:

```
#include <SPI.h>
#include <MFRC522.h>
constexpr uint8_t RST_PIN = D3;
constexpr uint8_t SS_PIN = D4;
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
/* Set the block to write data */
int blockNum = 2;
/* This is the actual data which is
```

```
going to be written into the card */

byte blockData [16] = {"Manali"};  // Change the name you want to store in RFID Tag

/* Create 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];

MFRC522::StatusCode status;

void setup()

{

  Serial.begin(9600);

  SPI.begin();

  mfrc522.PCD_Init();

  Serial.println("Scan a RFID Tag to write data...");

}

* Writ() function

void loop()

{

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

    key.keyByte[i] = 0xFF;

  }

  if ( ! mfrc522.PICC_IsNewCardPresent()){return;}
```

```
if ( ! mfrc522.PICC_ReadCardSerial()) {return;}

Serial.print("\n");

Serial.println("*Card Detected*");

Serial.print(F("Card UID:"));

for (byte i = 0; i < mfrc522.uid.size; i++){

  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

  Serial.print(mfrc522.uid.uidByte[i], HEX);

}

Serial.print("\n");

Serial.print(F("PICC type: "));

MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);

Serial.println(mfrc522.PICC_GetTypeName(piccType));

 Serial.print("\n");

 Serial.println("Writing to Data Block...");

 WriteDataToBlock(blockNum, blockData);

 Serial.print("\n");

 Serial.println("Reading from Data Block...");

 ReadDataFromBlock(blockNum, readBlockData);

 //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

 Serial.print("\n");

 Serial.print("Data in Block:");

 Serial.print(blockNum);
```

```
  Serial.print(" --> ");
  for (int j=0 ; j<16 ; j++){
    Serial.write(readBlockData[j]);
  }
  Serial.print("\n");
}
```

* Writ() function

```
void WriteDataToBlock(int blockNum, byte blockData[])
{
  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 Write: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  else {
    Serial.println("Authentication success");
  }
  status = mfrc522.MIFARE_Write(blockNum, blockData, 16);
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Writing to Block failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
```

```
    return;
  }
  else
  {Serial.println("Data was written into Block successfully");}
}
* ReadDataFromBlock() function
void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
  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");
  }
  status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
  if (status != MFRC522::STATUS_OK){
    Serial.print("Reading failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
```

```
  }
  else {
    Serial.println("Block was read successfully");
  }
}
```

3.APP SCRIPT Code:

```
// Smart Attendance System with Google Sheets and LCD Display
var ss = SpreadsheetApp.openById('1W9-yrbi3MPzjr-WY5EJkTCEMQeAl4n3xnvM-Lr-cUwo'); //Enter your googlesheets URL Id here
var sheet = ss.getSheetByName('Sheet1');
var timezone = "Asia/Kolkata"; //Set your timezone
function doGet(e){
  Logger.log( JSON.stringify(e) );
  //get gps data from ESP32
  if (e.parameter == 'undefined') {
    return ContentService.createTextOutput("Received data is undefined");
  }
  var Curr_Date = new Date();
  var Curr_Time = Utilities.formatDate(Curr_Date, timezone, 'HH:mm:ss');
  var name = stripQuotes(e.parameters.name);
  //Logger.log('name=' + name);
  var nextRow = sheet.getLastRow() + 1;
```

```
  sheet.getRange("A" + nextRow).setValue(Curr_Date);

  sheet.getRange("B" + nextRow).setValue(Curr_Time);

  sheet.getRange("C" + nextRow).setValue(name);

  //returns response back to ESP8266

  return ContentService.createTextOutput("Card holder name is stored in column C");

}

function stripQuotes( value ) {

  return value.toString().replace(/^["']|['"]$/g, "");

}

function doPost(e) {

  var val = e.parameter.value;

  if (e.parameter.value !== undefined){

    var range = sheet.getRange('A2');

    range.setValue(val);

  }

}
```

Chapter 6

CONCLUSION AND OVERVIEW

In this project attempt has been made to mark the attendance of the students using RFID technology. It has been up to mark, the use of RFID is a success, it is storing up data much faster than traditional method and with much higher accuracy. Just swiping the RFID tags attendance is being stored and time saving for the student and the teacher. It is obvious that the use of biometrics could improve some aspects of using this kind of system. High security level can be increased, much wider range of RFID can be set to capture the data over from distance. This system can be further improved by storing the attendance in the cloud or in much bigger database, even data can be sent to the parents about the presence and absence of the student, cards can be misused by the student's, proxy attendance can be given, where fingerprint can be a essential as well as implementing NFC Near field communication and improve and be much more secure to use.

REFERENCES:

[1] Ula, Mutammimul, et al. "A New Model of The Student Attendance Monitoring System Using RFID Technology." Journal of Physics: Conference Series. Vol. 1807. No. 1. IOP Publishing, 2021.

[2] Nivetha, R., et al. "Student Attendance System Using RFID." International Journal of Research in Engineering, Science and Management 3.10 (2020)

[4] Kariapper, R. K. A. R., and Suhail Razeeth. "RFID Based (IoT) Automatic Attendance System: A Survey Analysis." Available at SSRN 3372734 (2019)

[5] Rjeib, Hasanein D., et al. "Attendance and information system using RFID and web-based application for academic sector." International Journal of Advanced Computer Science and Applications 9.1 (2018)