```
/*

Statement       :    RFID based Student Identity System

Date of Exp.    :    07/04/2024

Author          :    Mansi Mandhane(A-24) & Aabha Nimje(A-33)

*/

*// Code for writing Student details in RFID card and tag

#include <SPI.h>

#include <MFRC522.h>

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

//GPIO 0 --> D3

//GPIO 2 --> D4

const uint8_t RST_PIN = D3;

const uint8_t SS_PIN = D4;

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

MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

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

/* Be aware of Sector Trailer Blocks */

int blockNum = 4;

/* Create array to read data from Block */

/* Length of buffer should be 4 Bytes

more than the size of Block (16 Bytes) */

byte bufferLen = 18;
byte readBlockData[18];

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

MFRC522::StatusCode status;

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


void setup()

{

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

//Initialize serial communications with PC

Serial.begin(9600);

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

//Initialize SPI bus

SPI.begin();

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

//Initialize MFRC522 Module

mfrc522.PCD_Init();

Serial.println("Scan a MIFARE 1K Tag to write data...");

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

}

void loop()

{  //-----------------------------------------------------------
-------------------

/* 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;

}

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

/* 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;}

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

Serial.print("\n");

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

/* Print UID of the Card */

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");
```

```
/* Print type of card (for example, MIFARE 1K) */

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

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

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

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

byte buffer[18];

byte len;

//wait until 20 seconds for input from serial

Serial.setTimeout(20000L);

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

Serial.println(F("----------------------------------------"));

Serial.println(F("Enter Student ID, ending with #"));

len = Serial.readBytesUntil('#', (char *) buffer, 16);

//add empty spaces to the remaining bytes of buffer

for (byte i = len; i < 16; i++) buffer[i] = ' ';

blockNum = 4;

WriteDataToBlock(blockNum, buffer);

ReadDataFromBlock(blockNum, readBlockData);

dumpSerial(blockNum, readBlockData);

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

Serial.println(F("----------------------------------------"));
Serial.println(F("Enter First Name, ending with #"));

len = Serial.readBytesUntil('#', (char *) buffer, 16);

for (byte i = len; i < 16; i++) buffer[i] = ' ';

blockNum = 5;

WriteDataToBlock(blockNum, buffer);

ReadDataFromBlock(blockNum, readBlockData);

dumpSerial(blockNum, readBlockData);

//––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––
----------------

Serial.println(F("---------------------------------------"));

Serial.println(F("Enter Last Name, ending with #"));

len = Serial.readBytesUntil('#', (char *) buffer, 16);

for (byte i = len; i < 16; i++) buffer[i] = ' ';

blockNum = 6;

WriteDataToBlock(blockNum, buffer);

ReadDataFromBlock(blockNum, readBlockData);

dumpSerial(blockNum, readBlockData);


Serial.println(F("---------------------------------------"));

Serial.println(F("Enter Phone Number, ending with #"));

len = Serial.readBytesUntil('#', (char *) buffer, 16);

for (byte i = len; i < 16; i++) buffer[i] = ' ';

blockNum = 8;

WriteDataToBlock(blockNum, buffer);
```

```
ReadDataFromBlock(blockNum, readBlockData);

dumpSerial(blockNum, readBlockData);

Serial.println(F("---------------------------------------"));

Serial.println(F("Enter Address, ending with #"));

len = Serial.readBytesUntil('#', (char *) buffer, 16);

for (byte i = len; i < 16; i++) buffer[i] = ' ';

blockNum = 9;

WriteDataToBlock(blockNum, buffer);

ReadDataFromBlock(blockNum, readBlockData);

dumpSerial(blockNum, readBlockData);

}


void WriteDataToBlock(int blockNum, byte blockData[])

{

//Serial.print("Writing data on block ");

//Serial.println(blockNum);

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

/* Authenticating the desired data block for write 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 Write: ");
Serial.println(mfrc522.GetStatusCodeName(status));

return;

}

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

else {

//Serial.print("Authentication OK - ");

}

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

/* Write data to the block */

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("Write OK");

}

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

}

/*****************************************************************
*************************************
```

```
* ReadDataFromBlock() function

*****************************************************************
***********************************/

void ReadDataFromBlock(int blockNum, byte readBlockData[])

{

//Serial.print("Reading data from block ");

//Serial.println(blockNum);

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

/* 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.print("Authentication OK - ");

}

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

/* 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("Read OK");

}


* dumpSerial() function


void dumpSerial(int blockNum, byte blockData[])
```

```
{
Serial.print("\n");
Serial.print("Data saved on block");
Serial.print(blockNum);
Serial.print(": ");
for (int j=0 ; j<16 ; j++){
Serial.write(readBlockData[j]);
}
Serial.print("\n");
//Empty readBlockData array
for( int i = 0; i < sizeof(readBlockData);  ++i )
readBlockData[i] = (char)0; //empty space
}
```

```
// Code for displaying data in RFID card and tag on google sheet


#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <SPI.h>

#include <MFRC522.h>

#include <HTTPSRedirect.h>

#include <Wire.h>


// Enter Google Script Deployment ID:

const char *GScriptId =
"AKfycbx7MFxZYUhf1XLtHwzynaVSc7Jc5XjSDVWMDyT0kz1H5Qua_loUK05iwVZ
CT9AutM4";

// Enter network credentials:

const char* ssid     = "Rajendra";

const char* password = "dewalkar@2002";


// 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";
const int   httpsPort   = 443;

const char* fingerprint = "";

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

HTTPSRedirect* client = nullptr;


// Declare variables that will be published to Google Sheets

String student_id;

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

#define total_blocks  (sizeof(blocks) / sizeof(blocks[0]))

#define RST_PIN  0  //D3

#define SS_PIN   2  //D4

#define BUZZER   4  //D2


MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

MFRC522::StatusCode status;

byte bufferLen = 18;

byte readBlockData[18];


void setup() {

Serial.begin(9600);

delay(10);

SPI.begin();
```

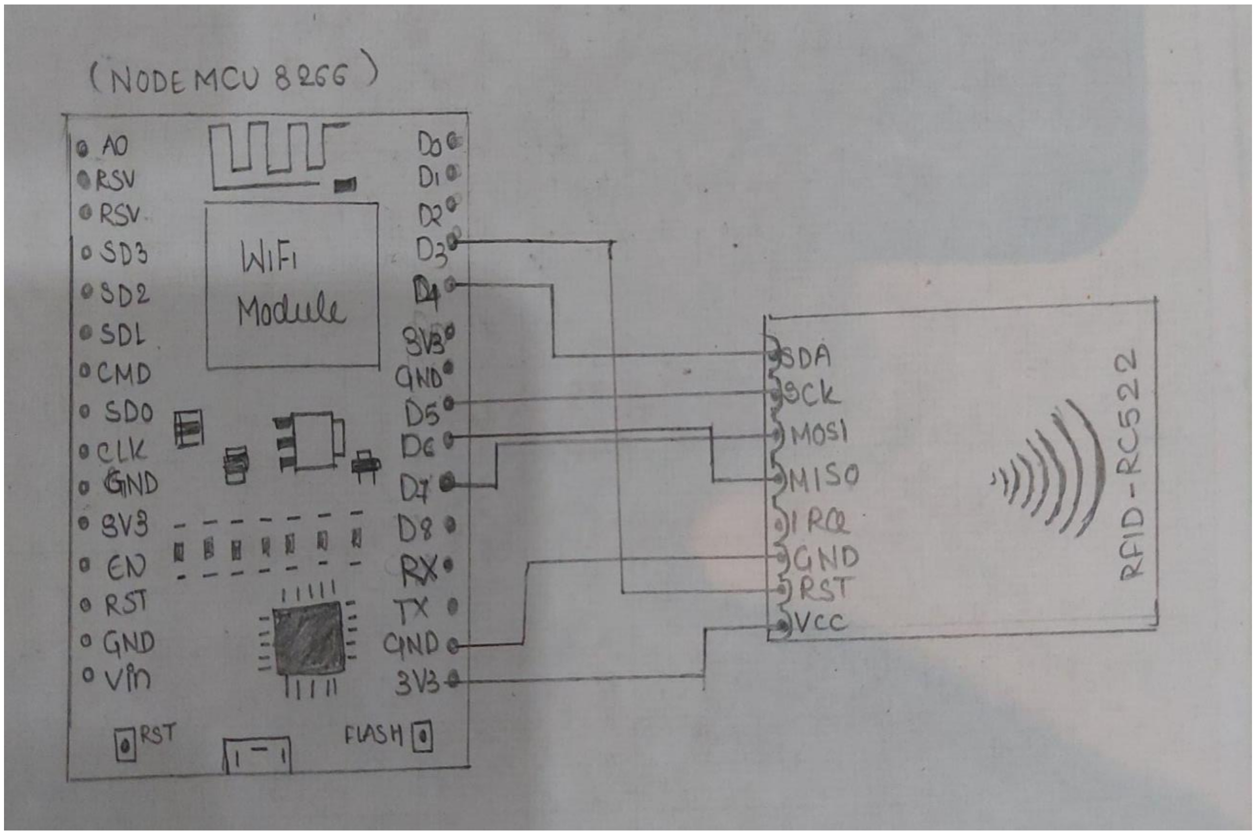
```
WiFi.begin(ssid, password);

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());

client = new HTTPSRedirect(httpsPort);

client->setInsecure();

client->setPrintResponseBody(true);

client->setContentTypeHeader("application/json");


Serial.print("Connecting to ");

Serial.println(host);


bool flag = false;
```

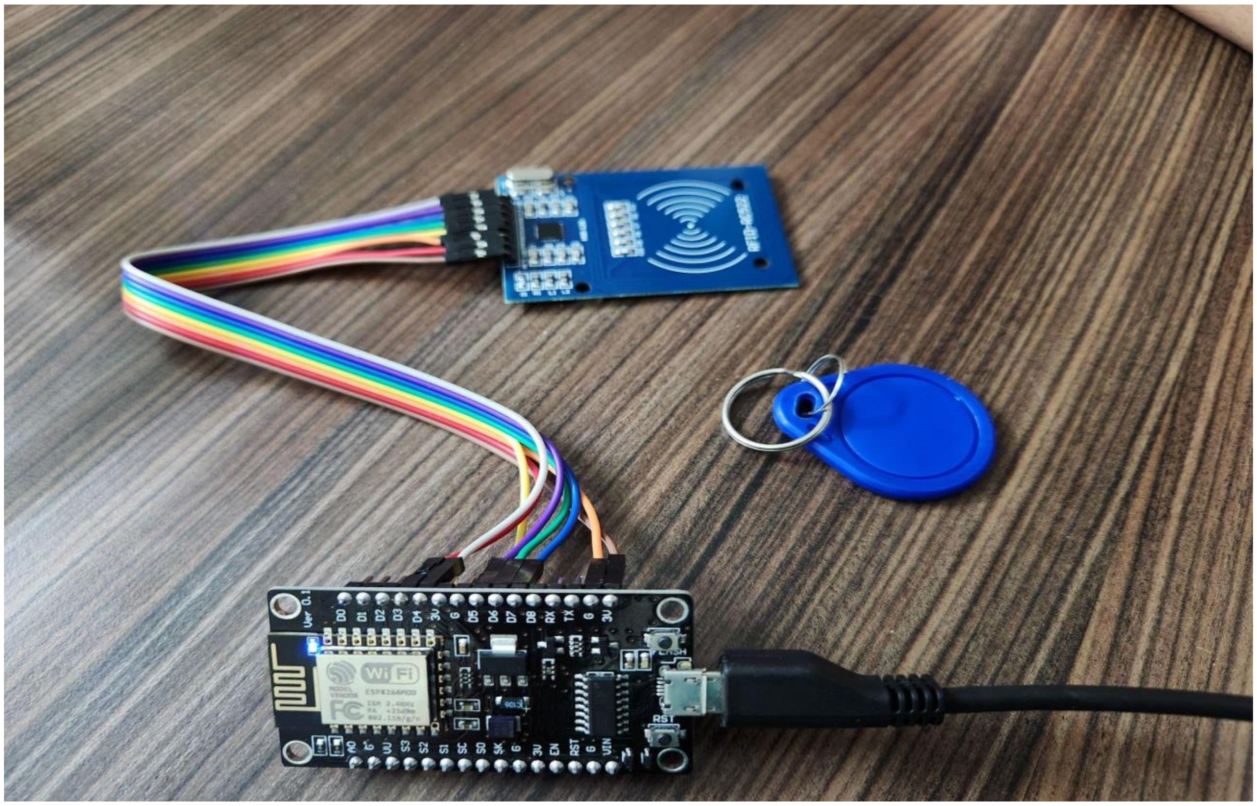
```
for(int i=0; i<5; i++){

int retval = client->connect(host, httpsPort);

if (retval == 1){

flag = true;

Serial.println("Connected. OK");

break;

}

else

Serial.println("Connection failed. Retrying...");

}


if (!flag){

Serial.print("Could not connect to server: ");

Serial.println(host);

delay(5000);

return;

}

delete client;

client = nullptr;

}

void loop() {

static bool flag = false;

if (!flag){
```

```
client = new HTTPSRedirect(httpsPort);

client->setInsecure();

flag = true;

client->setPrintResponseBody(true);

client->setContentTypeHeader("application/json");

}

if (client != nullptr){

if (!client->connected())

{client->connect(host, httpsPort);}

}

else{Serial.println("Error creating client object!");}

mfrc522.PCD_Init();

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

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

Serial.println();

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

String values = "", data;

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

ReadDataFromBlock(blocks[i], readBlockData);

if(i == 0){

data = String((char*)readBlockData);

data.trim();

student_id = 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 + ",";

}

}

payload = payload_base + values;

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

Serial.println(payload);

if(client->POST(url, host, payload)){

Serial.println("Published successfully");

}

else{

Serial.println("Error while connecting");

}
```

```
delay(5000);

}

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 {

readBlockData[16] = ' ';

readBlockData[17] = ' ';

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

}

}
```

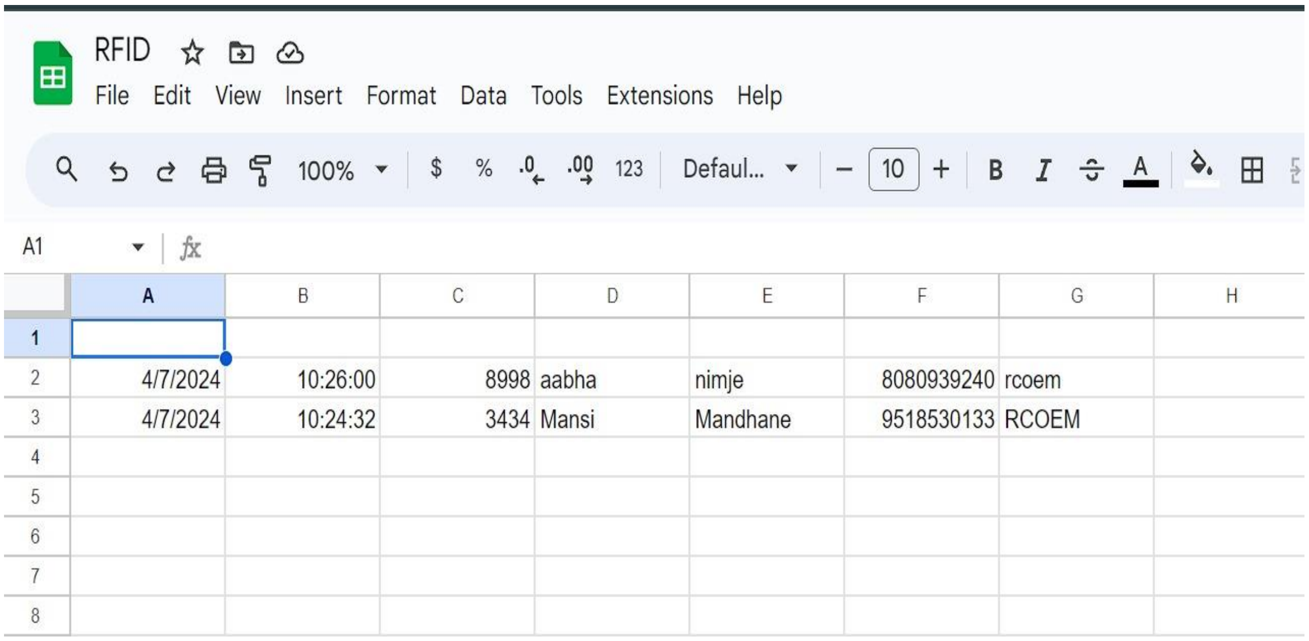
//Circuit Diagranm





Mansi Mandhane(A-24)
Aabha Nimje (A-33)

```
*// Output on Google sheet
```



	A	B	C	D	E	F	G	H
1								
2	4/7/2024	10:26:00	8998	aabha	nimje	8080939240	rcoem	
3	4/7/2024	10:24:32	3434	Mansi	Mandhane	9518530133	RCOEM	
4								
5								
6								
7								
8								