#include <SoftwareSerial.h>

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

SoftwareSerial RFID(D6, D5); // D6 - TX (RFID), D5 - RX (RFID)

const char \*ssid = "Hí lô";

const char \*password = "12345678d";

const char\* device\_token  = "8ceb36c810343326";

String URL = "http://192.168.43.202:80//rfidattendance/getdata.php";

String getData, Link;

String OldCardID = "";

String CardID ="";

unsigned long previousMillis = 0;

Dòng này khai báo các biến lưu trữ dữ liệu và thời gian

int flag=0,RX\_Flag=0;

char Code[14];

long Num=0;

int BUZZ = D1;

int LED1 = D3;

int LED2 = D7;

int redLed = D0;

int blueLed = D4;

int i;

long temp=0, last=0;

void setup() {

  //delay(1000);

  Serial.begin(115200);

  delay(100);

  RFID.begin(9600);

  pinMode(BUZZ, OUTPUT);

  pinMode(LED1, OUTPUT);

  pinMode(LED2, OUTPUT);

  pinMode(blueLed, OUTPUT);

  pinMode(redLed, OUTPUT);

  connectToWiFi();

}

* Hàm setup() được gọi khi Arduino được khởi động.
* Trong hàm này, tốc độ truyền dữ liệu của cổng nối tiếp được thiết lập là 115200 bps.
* Đồng thời, các chân kết nối với buzz và các đèn LED được khai báo là đầu ra.
* Hàm connectToWiFi() được gọi để kết nối đến mạng WiFi.

void loop() {

  //check if there's a connection to Wi-Fi or not

  if(!WiFi.isConnected()){

    connectToWiFi();

Kiểm tra xem ESP8266 đã kết nối WiFi hay chưa. Nếu không, hàm connectToWiFi() được gọi để kết nối lại.

  }

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

  if (millis() - previousMillis >= 15000) {

    previousMillis = millis();

    OldCardID="";

  }

Kiểm tra thời gian kể từ lần cuối cùng gửi dữ liệu thẻ. Nếu đã trôi qua 15 giây, biến OldCardID được đặt thành chuỗi rỗng để chuẩn bị cho việc gửi lại dữ liệu thẻ.

  delay(50);

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

  //look for new card

  ChuongTrinhDocThe();

  if(Num) {

    CardID = String(Num);

Chuyển đổi giá trị số của ID thẻ thành chuỗi và lưu trữ vào biến CardID.

    Num=0;

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

    if( CardID == OldCardID ){

 Kiểm tra xem ID thẻ mới đọc có trùng với ID thẻ trước đó hay không. Nếu trùng, không thực hiện các thao tác xử lý và quay trở lại vòng lặp chính.

      return;

    }

    else{

      OldCardID = CardID;

Nếu ID thẻ mới không trùng với ID thẻ trước đó, gán ID thẻ mới cho biến OldCardID để lưu trữ ID thẻ trước đó.

    }

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

    //  Serial.println(CardID);

    SendCardID(CardID);

    delay(1000);

  }

}

void ChuongTrinhDocThe()

{

    Read\_ID();

    if(RX\_Flag==1)

Kiểm tra xem đã nhận được dữ liệu từ module đọc thẻ RFID hay chưa.

    {

      digitalWrite(BUZZ, HIGH);

      delay(500);

      digitalWrite(BUZZ, LOW);

      for(i=5;i<11;i++)

Chuyển đổi mã hexa thành giá trị số của ID thẻ.

      {

        Num<<=4;

        if(Code[i]>64)  Num+=((Code[i])-55);

        else Num+=((Code[i])-48);

        //Serial.println(Num);

      }

      Serial.println(Num);

      //SendCardID( String(Num));

      while(RFID.read()>0);//

 Đọc hết dữ liệu còn lại trong bộ đệm của module đọc thẻ RFID.

      RX\_Flag=0;//

      //Num=0;

    }

}

void Read\_ID(void)

{

    int i=0;

    char temp;

    for(i=0;(RFID.available()>0);i++)//

    //while(RFID.available()>0)

    {

      temp=RFID.read();

      //Serial.print(temp);

      delay(2);

      //Serial.println(i);

Đọc dữ liệu từ module đọc thẻ RFID và lưu vào biến temp.

      if(temp==0X02)  //recieve the ssrt bit

        {

         flag=1;i=0;RX\_Flag=0;

đọc được byte bắt đầu của dữ liệu, thì thiết lập cờ flag bằng 1, đặt lại biến i bằng 0 và đặt cờ RX\_Flag bằng 0

        }

      if(flag==1)//detect the start bit and recieve data

      {

        if(temp==0X03)//detect the end code,

        {

         flag=0;  //zero clearing

         if(i==13) RX\_Flag=1;//

         else RX\_Flag=0;

         break;

Nếu temp bằng 0X03 tức là đọc được byte kết thúc của dữ liệu, thì đặt cờ flag bằng 0. Nếu biến i bằng 13, tức là đã đọc đủ 13 byte dữ liệu, thì đặt cờ RX\_Flag bằng 1; ngược lại, đặt cờ RX\_Flag bằng 0. Sau đó, thoát khỏi vòng lặp

        }

        Code[i]=temp;

      }

    }

     flag=0;//

}

//\*\*\*\*\*\*\*\*\*\*\*\*send the Card UID to the website\*\*\*\*\*\*\*\*\*\*\*\*\*

void SendCardID(String Card\_uid) {

  Serial.println("Sending the Card ID");

  if (WiFi.isConnected()) {

    WiFiClient client;

    HTTPClient http;    // Declare object of class HTTPClient

    // GET Data

    getData = "?card\_uid=" + String(Card\_uid) + "&device\_token=" + String(device\_token);

Xây dựng chuỗi getData bằng cách ghép các giá trị Card\_uid và device\_token với các chuỗi văn bản

    Link = URL + getData;

    http.begin(client, Link);

    int httpCode = http.GET();

* Gửi yêu cầu HTTP GET đến máy chủ thông qua phương thức http.GET() và lưu kết quả vào biến httpCode.

// Send the request

    if (httpCode > 0) {

      String payload = http.getString();    // Get the response payload

      Serial.println(httpCode);    // Print HTTP return code

      Serial.println(Card\_uid);     // Print Card ID

      Serial.println(payload);    // Print request response payload

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

        if (payload.substring(0, 5) == "login") {

          String user\_name = payload.substring(5);

          // Serial.println(user\_name);

        } else if (payload.substring(0, 6) == "logout") {

          String user\_name = payload.substring(6);

          // Serial.println(user\_name);

        } else if (payload == "succesful") {

          // Handle successful response

        } else if (payload == "available") {

          // Handle available response

        }

* Kiểm tra giá trị của payload để xử lý các trường hợp khác nhau. Nếu payload bắt đầu bằng "login", tức là ID của thẻ được chấp nhận và trả về tên người dùng, lưu tên người dùng vào biến user\_name. Nếu payload bắt đầu bằng "logout", tức là ID của thẻ bị từ chối và trả về tên người dùng, lưu tên người dùng vào biến user\_name. Nếu payload bằng "succesful", tức là gửi ID của thẻ thành công. Nếu payload bằng "available", tức là ID của thẻ không tồn tại trong hệ thống.

      } else {

        String error = http.errorToString(httpCode);

        Serial.println("HTTP error: " + error);

      }

    } else {

      Serial.println("Connection failed");

    }

    http.end();  // Close connection

  }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*connect to the WiFi\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void connectToWiFi(){

    WiFi.mode(WIFI\_OFF);        //Ngăn chặn vấn đề kết nối lại (mất quá nhiều thời gian để kết nối)

    delay(5000);

    WiFi.mode(WIFI\_STA);

    Serial.print("Đang kết nối với ");

    Serial.println(ssid);

    WiFi.begin(ssid, password);

    unsigned long connectionTimeout = millis();

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

      delay(500);

      Serial.print(".");

      // Timeout sau 15 giây

      if (millis() - connectionTimeout > 15000) {

        Serial.println("Hết thời gian kết nối");

        break;

      }

    }

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

      Serial.println("");

      Serial.println("Đã kết nối");

      Serial.print("Địa chỉ IP: ");

      Serial.println(WiFi.localIP());  //Địa chỉ IP được gán cho ESP của bạn

    } else {

      Serial.println("");

      Serial.println("Kết nối thất bại");

    }

    delay(1000);

}

//=======================================================================