***Vấn đề/ người dùng mà dự án hướng đến: chân dung vấn đề ((dưới dạng mindmap), các ý tưởng về giải pháp***

Điểm danh luôn là một phần không thể thiếu trong các lớp học mà đặc biệt là ở trường, khi sĩ số một lớp thường ở mức từ 30 đến 50 học sinh. Mình nhận thấy việc điểm danh thủ công (bằng cách đếm từng người trong lớp) là một công việc gây mất thời gian và thậm chí có thể dẫn đến sai sót (đếm sai). Qua đó, mình đề xuất một phương án tiết kiệm thời gian hơn, chính xác hơn và trực quan hơn, đó là xây dựng một hệ thống Arduino kết hợp với RFID và Excel.

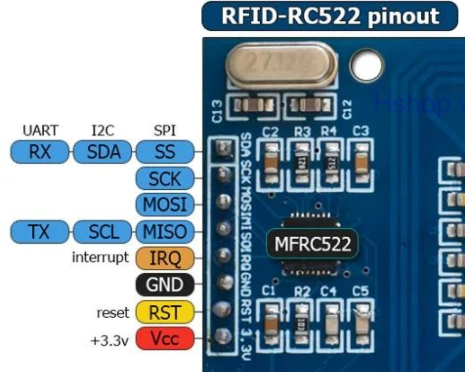
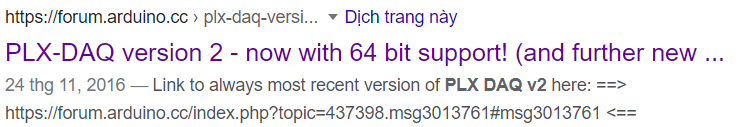
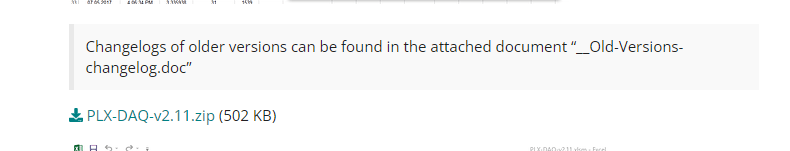
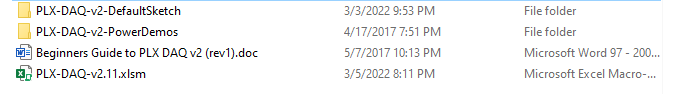
Mô tả về giải pháp: cách tiếp cận vấn đề, định nghĩa vấn đề, tóm tắt giải pháp (sơ đồ thông tin, lưu đồ)

* Giải pháp
  + Sử dụng mạch RFID để ghi thông tin vào móc khóa từ (để đơn giản hóa vấn đề, ghi vào dưới dạng số thứ tự bắt đầu từ 1 → 5)
  + Lúc điểm danh thì mạch RFID sẽ đọc thông tin của móc khóa rồi gửi thông tin lên Excel để thống kê và theo dõi (ai tới khi nào, bao nhiêu người đã tới, ai vắng)
  + Lưu ý: mỗi lần mạch RFID đọc/viết thành công thì buzzer sẽ phát tiếng

Chuẩn bị vật dụng

* Mạch Arduino Uno/Vietduino Uno Uno, mạch RFID NFC 13.56MHz RC522, 7 dây cắm đực cái, 2 dây cắm đực đực và 1 buzzer.

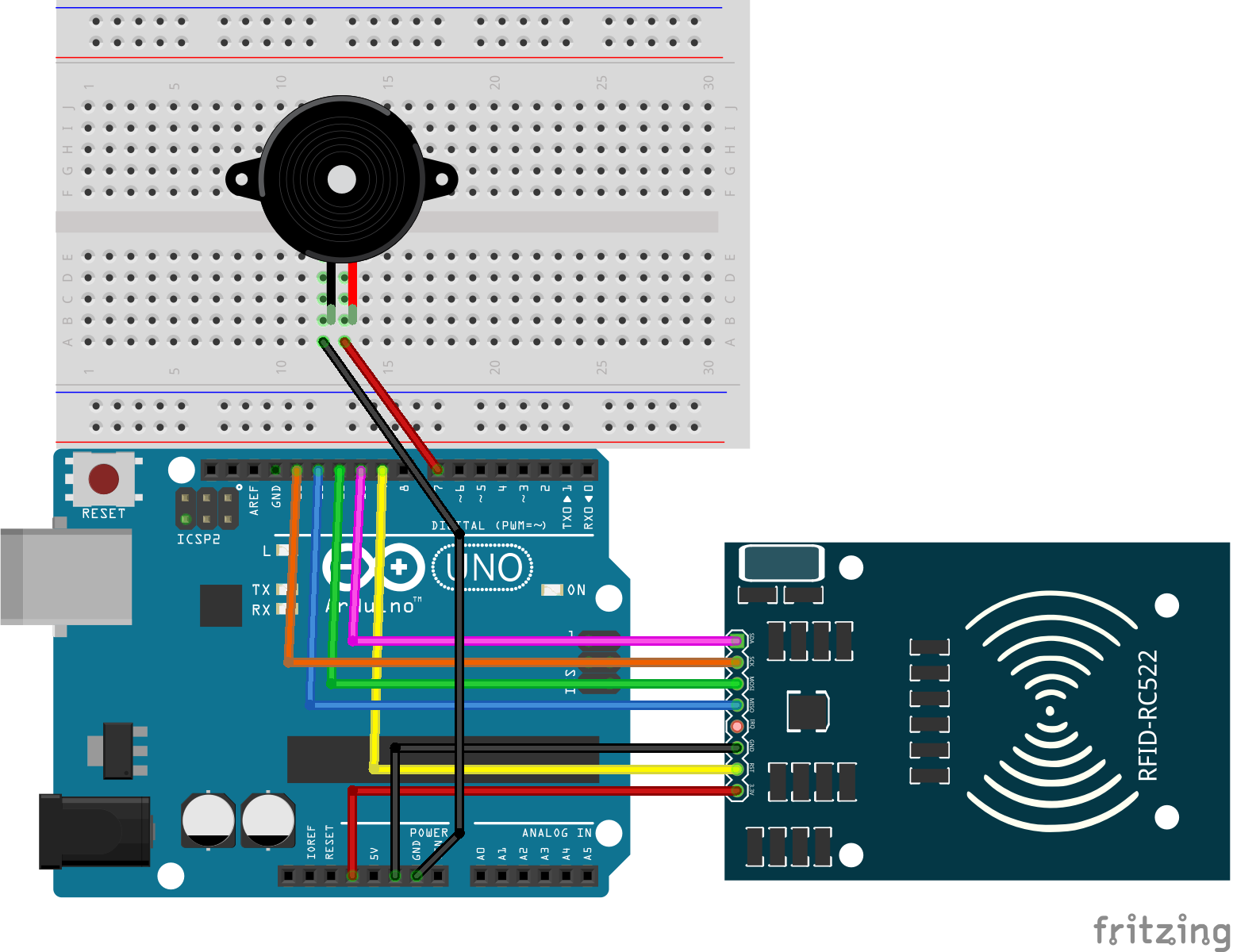
***Một số kiến thức cần biết***

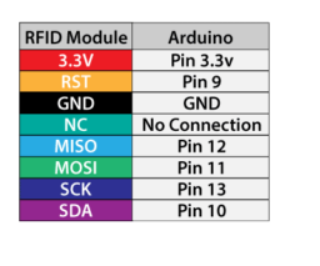
* Về phần cứng:
  + Mạch RFID
    - RFID là gì ???
      * RFID là chữ viết tắt của Radio Frequency IDentification (hay Nhận dạng qua tần số vô tuyến) và đây là một công nghệ dùng kết nối sóng vô tuyến để tự động xác định và theo dõi các thẻ nhận dạng gắn vào vật thể
    - Hệ thống RFID
      * Một hệ thống RFID thường bao gồm 2 thành phần chính là thẻ (tag) (chip RFID chứa thông tin) và đầu đọc (reader) đọc các thông tin trên chip.
    - Nguyên lý hoạt động của RFID
      * **Đầu đọc** phát ra sóng điện từ ở một tần số nhất định. Nếu **thẻ** nằm trong vùng hoạt động thì sẽ cảm nhận được sóng điện từ này, thu nhận năng lượng và từ đó phát lại cho **đầu đọc** biết mã số của mình. Từ đó **đầu đọc** nhận biết được tag nào đang trong vùng hoạt động.
    - RFID và Arduino
      * Đầu đọc: RFID RC522
        + Dây nối với mạch Arduino Uno/Vietduino Uno:
        + 
      * Tag: RFID Tag 13.56 MHz S50
        + Bộ nhớ 1024 Byte, được chia thành **16 vùng (Sectors)** (đánh số từ 0 → 15)
        + Mỗi vùng sẽ quản lý **4 khối (Blocks)** và mỗi khối có bộ nhớ **16 Byte**  
          (các khối đánh số từ 0, vùng 1 quản lý khối 0 → khối 3, vùng 2 quản lý khối 4 → khối 7, …)
        + **Khối thứ 4** của mỗi vùng là một Sector Trailer, có chứa các Access Bits (tạm dịch: bit truy cập) dùng để cấp quyền đọc và viết lên 3 khối còn lại trong vùng
        + **Khối thứ nhất của vùng 0** chứa thông tin của nhà phát hành và mã số UID (Unique IDentifier) (không được chỉnh sửa khối này).
  + Chuẩn giao tiếp SPI
    - SPI (Serial Peripheral Bus) là một chuẩn truyền thông nối tiếp tốc độ cao do hãng Motorola đề xuất.
    - Đây là kiểu truyền thông Master-Slave, trong đó có 1 chip **Master** điều phối quá trình truyền thông và các chip **Slaves** được điều khiển bởi Master vì thế truyền thông chỉ xảy ra **giữa Master và Slave**.
    - SPI là một cách truyền song công (full duplex) nghĩa là tại **cùng một thời điểm** quá trình truyền và nhận có thể xảy ra **đồng thời**.
    - SPI đôi khi được gọi là chuẩn truyền thông “4 dây” vì có 4 đường giao tiếp trong chuẩn này đó là SCK (Serial Clock), MISO (Master Input Slave Output), MOSI (Master Output Slave Input) và SS (Slave Select).
    - Cụ thể hơn:
    - ***SCK (Serial Clock)***: Thiết bị Master tạo xung tín hiệu SCK và cung cấp cho Slave. Xung này có chức năng giữ nhịp cho giao tiếp SPI. Mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi → Quá trình ít bị lỗi và tốc độ truyền cao.
    - ***MISO (Master Input Slave Output)***: Tín hiệu tạo bởi thiết bị Slave và nhận bởi thiết bị Master. Đường MISO phải được kết nối giữa thiết bị Master và Slave.
    - ***MOSI (Master Output Slave Input):*** Tín hiệu tạo bởi thiết bị Master và nhận bởi thiết bị Slave. Đường MOSI phải được kết nối giữa thiết bị Master và Slave.
    - ***SS (Slave Select)***: Chọn thiết bị Slave cụ thể để giao tiếp. Để chọn Slave giao tiếp thiết bị Master chủ động kéo đường SS tương ứng xuống mức 0 (Low).
    - Như vậy, ta có thể tóm tắt quá trình truyền dữ liệu qua giao tiếp SPI như sau:
      * Đầu tiên, tín hiệu đồng hồ SCK sẽ tạo đồng bộ trong hệ thống theo các xung nhịp của nó.
      * Sau đó, Master chọn Slave sẽ giao tiếp bằng cách cho điện áp chân SS về mức thấp.
      * Cuối cùng, việc truyền dữ liệu sẽ diễn ra trên chân MOSI của Master với MISO của Slave.
      * \*\*\*Bạn đọc có thể tham khảo nguyên lý hoạt động của SPI chi tiết hơn tại đây: [SPI cho dân analog – Phần 2/4: Nguyên lý hoạt động | Chuyên Mục Công Nghệ (chuyenmuccongnghe.com)](https://chuyenmuccongnghe.com/2014/08/06/spi-cho-dan-analog-phan-24-nguyen-ly-hoat-dong/)
  + Chuẩn giao tiếp SPI và Arduino
    - Mạch RFID RC522 có hỗ trợ giao tiếp SPI, I2C và UART. Tuy vậy, nếu không có điều chỉnh gì ở trên mạch RFID RC522 thì ta sẽ mặc định rằng mạch này kết nối với Arduino bằng giao tiếp SPI.
    - 
    - Dựa vào hình ở trên, ta có thể thấy chân được đánh tên là SDA trên mạch RFID RC522 thực chất có vai trò là đường SS trong chuẩn giao tiếp SPI, đường SDA trong chuẩn giao tiếp I2C và đường RX trong chuẩn giao tiếp UART.
    - Như vậy, ta không cần phải lo về việc “tại sao chuẩn giao tiếp SPI có đường SS mà trên mạch RFID RC522 lại chỉ có đường SDA?”.
    - \*\*\*Nếu bạn đọc có hứng thú với việc sử dụng chuẩn giao tiếp I2C để kết nối với Arduino, các bạn có thể tham khảo tại đây: [Arduino RFID RC522 Tutorial | Microcontroller Tutorials (teachmemicro.com)](https://www.teachmemicro.com/arduino-rfid-rc522-tutorial/)
* Về phần mềm:
  + Thư viện MFRC522
    - Tải thư viện qua Github của tài khoản **miguelbalboa** (bằng cách tìm kiếm “mfrc522 library” trên Google)
    - <https://github.com/miguelbalboa/rfid>
    - Mở Arduino IDE, di chuột vào **Sketch** ở góc trái trên, di chuột vào **Include Library**
    - Chọn **Add .ZIP Library** và chọn thư viện bạn đã tải về ở bước 1
  + Hàm toInt()
    - dùng để chuyển một chuỗi các chữ số thành một số (Ví dụ: “123” → 123)
  + PLX-DAQ (thông tin và cài đặt)
    - Phần mềm giúp Arduino gửi thông tin lên Microsoft Excel dưới dạng cột
    - Google PLX-DAQ v2, chọn theo link giống như hình:
    - 
    - Chọn link đầu tiên:
    - 
    - Bạn sẽ được dẫn tới một trang web khác, kéo xuống một xíu thì bạn sẽ thấy một link tải:
    - 
    - Cuối cùng, nhấp vào đường link để tải file zip về và giải nén, ta sẽ được một số file.
    - 
    - File Excel có tên **PLX-DAQ-v2.11.xlsm** là phần mềm mà ta sẽ sử dụng để quản lý dữ liệu được gửi lên từ Arduino.

# THỰC HIỆN DỰ ÁN

# 1. Lắp mạch điện (Fritzing)

Mạch chính:

Lưu ý: các cổng kết nối của mạch RFID RC522 trong mạch được kết nối với mạch Arduino Uno/Vietduino Uno như sau:



# 2. Hướng dẫn các câu lệnh:

## Phần 1. Sử dụng mạch RFID để ghi thông tin vào móc khóa từ bằng chương trình write.ino

### Phần 1.1. Khai báo:

#include <SPI.h>

#include <MFRC522.h>

// khai báo các thư viện cần dùng

#define RST\_PIN 9

#define SS\_PIN 10   
// khai báo PIN cho cổng RST và SS (các PIN này có thể thay đổi được)  
const int buzzer = 7;  
  
//khai báo PIN của buzzer  
  
MFRC522 mfrc522(SS\_PIN, RST\_PIN);   
  
//khởi tạo một đối tượng (instance) mfrc522 với các PIN tương ứng  
  
// chương trình chính sẽ hoạt động theo một vòng lặp cho tới khi nào tất cả các thẻ được ghi thông tin vào  
  
String IDs[5] = {"Thang","Ngoc","Minh","Thu","Nghia"};  
// IDs[] là một mảng gồm 5 phần tử chứa 5 ID của 5 người.   
// có thể thay đổi các phần tử của mảng IDs[] thành tên hoặc các thông tin khác   
// (nhưng vẫn phải đảm bảo độ dài nhỏ hơn hoặc bằng 16 ký tự, do một khối chỉ lưu trữ được 16 Byte)  
  
int index = 0;   
//index là chỉ số của mảng IDs[] trong vòng lặp   
// ví dụ: index = 0 thì IDs[index] = 1  
  
int tags = 5;  
// tags là số tags chưa đọc, nếu tags trở về bằng 0 thì quá trình viết dữ liệu lên thẻ không diễn ra nữa  
  
void setup() {

pinMode(buzzer, OUTPUT);

**// khai báo pinMode của buzzer**

**Serial**.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

**//khởi động cổng SPI và chip MFRC522**

**Serial**.println(F("Write personal data on a MIFARE PICC "));

}

### Phần 1.2. Các hàm con:

**// hàm** card\_info() **hiển thị các thông tin cơ bản của thẻ từ**

void card\_info(){

**Serial**.println();

**Serial**.print(F("Card UID:")); **// mã 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);

**// mfrc522.uid.uidByte[i] là một số thập phân (Decimal), còn mã UID được lưu thành một thập lục phân**

**// do đó, ta phải đổi mfrc522.uid.uidByte[i] sang hệ thập lục phân (Hexadecimal) khi in ra trên Serial monitor**

}

}

**// hàm get\_authenticate() lấy quyền truy cập vào một khối (block) nào đó**

**// return 0; nghĩa là không lấy được quyền truy cập**

**// return 1; nghĩa là đã lấy được quyền truy cập**

bool get\_authenticate(MFRC522::StatusCode status, MFRC522::MIFARE\_Key key, byte block){

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

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

**Serial**.print(F("PCD\_Authenticate() failed: "));

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

return 0;

}

else

{

**Serial**.println(F("PCD\_Authenticate() success "));

return 1;

}

}

**// hàm** preprocess() **tạo mảng buffer[] để viết thông tin vào thẻ từ**

void preprocess(byte buffer[16], byte len){

String temp = IDs[index];

**// biến thông tin cần ghi vào thẻ thành dạng chuỗi (String)**

for(int i = 0; i < len; ++i)

buffer[i] = 0;

for(int i = 0; i < temp.length(); ++i)

buffer[i] = temp[i];

**// gán từng phần tử của mảng buffer[] bằng một ký tự tương ứng của chuỗi** temp

buffer[temp.length()] = '\*';

**// gán vị trí cuối cùng của buffer là một kí tự đặc biệt để nhận biết điểm kết thúc (sẽ được nhắc lại trong Phần 2.2.2 hàm get\_ID())**

}

**// hàm** write\_info() **viết thông tin từ mảng** buffer[] **đã tạo từ hàm** preprocess()

void write\_info(MFRC522::StatusCode status, byte block, byte buffer[16], byte len){

status = mfrc522.MIFARE\_Write(block, buffer, len);

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

**Serial**.print(F("MIFARE\_Write() failed: "));

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

return;

}

else{

**Serial**.println(F("MIFARE\_Write() success "));

tone(buzzer, 3136, 250);

**//buzzer kêu báo hiệu đã viết thông tin lên móc khoá thành công**

}

}

### Phần 1.3. Hàm chính:

void loop() {

if(tags > 0){

**// nếu đầu đọc phát hiện có thẻ từ ở gần**

if ( ! mfrc522.PICC\_IsNewCardPresent()) {

return;

}

**// chọn một trong những thẻ phát hiện được**

if ( ! mfrc522.PICC\_ReadCardSerial()) {

return;

}

**// in thông tin thẻ**

card\_info();

**// lấy quyền truy cập vào khối 4**

MFRC522::StatusCode status;

MFRC522::MIFARE\_Key key;

for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

byte block = 4;

if (get\_authenticate(status, key, block) == 0) return;

**// tạo mảng** buffer[]

byte len = 16;

byte buffer[len];

preprocess(buffer, len);

**Serial**.println();

**// viết thông tin vào thẻ dựa vào thông tin trong** buffer[]

write\_info(status, block, buffer, len);

**// in ra thông tin đã ghi vào thẻ**

**Serial**.print("Card number: ");

**Serial**.print(IDs[index]);

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

mfrc522.PICC\_HaltA();

mfrc522.PCD\_StopCrypto1();

index++;

--tags;

}

}

## Phần 2. Sử dụng mạch RFID để đọc thông tin vào móc khóa từ và thống kê lên Excel

### Phần 2.1. Các câu lệnh thao tác với PLX-DAQ

* Trong hàm void setup():
  + **Serial**.println("CLEARDATA");
    - xóa các thông tin đã truyền lên Excel trước đó
  + **Serial**.println("LABEL,X,Y");
    - LABEL nằm ở đầu để thông báo đây là lệnh gán nhãn X và Y cho 2 cột đầu tiên trên sheet đầu tiên của file Excel
* Trong hàm void loop():
  + **Serial**.println("DATA,C,D");
    - DATA nằm ở đầu để thông báo đây là lệnh gán giá trị của C cho cột X và giá trị của D cho cột Y

### Phần 2.2. Đọc thông tin từ móc khoá từ bằng chương trình read.ino

#### Phần 2.2.1. Khai báo (gần giống với phần 1.1)

#include <SPI.h>

#include <MFRC522.h>

#define RST\_PIN 9

#define SS\_PIN 10

const int buzzer = 7;

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

void setup() {

pinMode(buzzer, OUTPUT);

**Serial**.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

**Serial**.println("CLEARDATA");

**Serial**.println("LABEL,CLOCK,ID");

}

#### Phần 2.2.2. Các hàm con:

**// hàm get\_authenticate() giống với phần 1.2:**

bool get\_authenticate(MFRC522::StatusCode status, MFRC522::MIFARE\_Key key, byte block){

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

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

**Serial**.print(F("PCD\_Authenticate() failed: "));

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

return 0;

}

else

{

**Serial**.println(F("PCD\_Authenticate() success "));

return 1;

}

}

**// hàm read\_ID() đọc thông tin từ móc khoá từ và lưu vào mảng buffer1[] với mỗi phần tử của mảng là một mã ASCII ứng với một kí tự của ID**

**// Ví dụ: Móc khoá có ID là Thang sẽ cho ra mảng**

**buffer[1] = [84,104,97,110,103]**

bool read\_ID(MFRC522::StatusCode status, byte block, byte buffer1[18], byte len){

status = mfrc522.MIFARE\_Read(block, buffer1, &len);

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

**Serial**.print(F("Reading failed: "));

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

return 0;

}

return 1;

}

**// hàm get\_ID() chuyển thông tin lưới dưới dạng mảng buffer1[] thành một chuỗi:**

String get\_ID(byte buffer1[18]){

String ID = "";

for (int i = 0; i < 16; i++)

**// sử dụng dấu ‘\*’ để kết thúc đọc thông tin (đã nhắc trong Phần 1.2 hàm preprocess())**

if((char)buffer1[i] == '\*') break;

else ID = ID + (char)buffer1[i];

**//** (char)buffer1[i] **biến buffer1[i] từ mã ASCII sang ký tự tương ứng**

return ID;

}

### Phần 2.3. Hàm chính:

void loop(){

if ( ! mfrc522.PICC\_IsNewCardPresent()) {

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial()) {

return;

}

byte block = 4;

byte len = 18;

byte buffer1[18];

**// lấy quyền truy cập**

MFRC522::StatusCode status;

MFRC522::MIFARE\_Key key;

for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

if (get\_authenticate(status, key, block) == 0) return;

**// tiến hành đọc, nếu hàm read\_ID trả về 1 thì thông tin đọc được từ móc khoá từ sẽ lưu vào mảng buffer1[], còn trả về 0 thì return**

if (read\_ID(status, block, buffer1, len) == 0) return;

**// viết lên Serial theo cú pháp của PLX-DAQ**

**Serial**.print("DATA,TIME,");

String ID = get\_ID(buffer1);

**Serial**.print(ID);

**Serial**.println();

**//buzzer kêu báo hiệu đã đọc thông tin từ móc khoá thành công**

tone(buzzer, 3136, 250);

**//có thể giảm thời gian delay xuống nếu muốn đọc thẻ nhanh hơn**

delay(1000);

mfrc522.PICC\_HaltA();

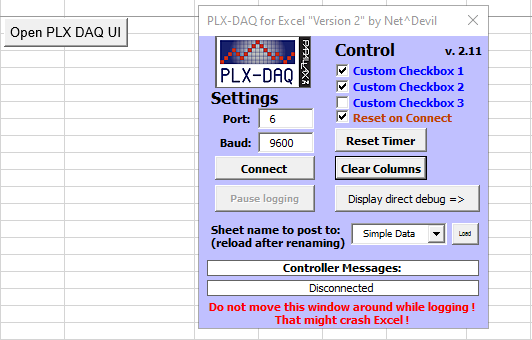
mfrc522.PCD\_StopCrypto1();

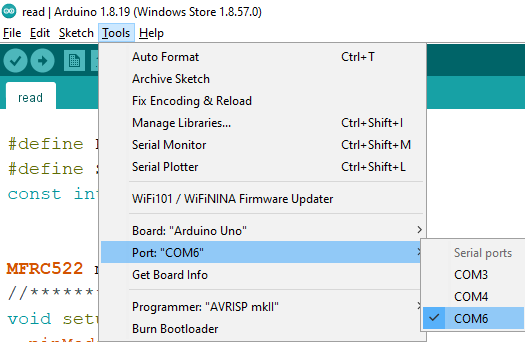
}

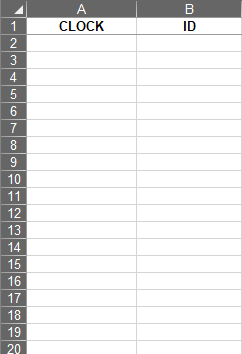
## **Phần 3. Thao tác với PLX-DAQ và Excel:**

### **Phần 3.1. Ghi thông tin lên file Excel:**

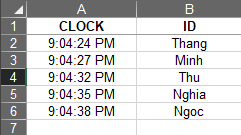
* Bước 1: Mở file Excel có tên **PLX-DAQ-v2.11** ở nơi bạn giải nén file **PLX-DAQ-v2.11.zip**, sau đó bấm vào nút “PLX DAQ UI” ở giữa sheet “Simple Data”. Một bảng các chức năng (tạm gọi là bảng PLX-DAQ) sẽ được hiện ra như hình ở dưới:



* Bước 2:
  + Chỉnh Port thành số tương ứng với Serial port trên Arduino IDE (ví dụ: trên Arduino IDE hiển thị bạn đang sử dụng cổng COM6 → Port 6).
  + 
  + Chỉnh Baud thành số bằng với Baud rate mà bạn đã khai báo thông qua đoạn code **Serial**.begin(Baud rate);.
* Bước 3: Upload (Ctrl + U) sketch **read.ino** trên Arduino IDE .
* Bước 4:
  + Quay về file Excel, bấm vào nút Connect trên bảng PLX-DAQ.
  + Khi đó, 2 cột đầu tiên của file Excel sẽ có nhãn lần lượt là **CLOCK và ID.**

****

* Bước 5: lần lượt cho mạch RFID đọc từng móc khoá từ, các thông tin về thời gian, ID sẽ được hiện lên file Excel (với mỗi lần đọc thành công thì buzzer sẽ phát ra tiếng)



Khi đọc xong, ta có thể ấn vào **Disconnect** trên bảng PLX-DAQ và nhấn vào **Clear Columns** để xoá những thông tin đã hiển thị lên file Excel.

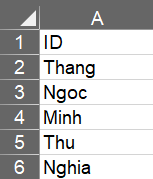
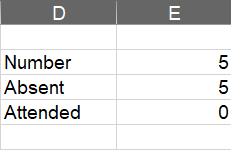
### **Phần 3.2. Điểm danh từ thông tin trên file Excel:**

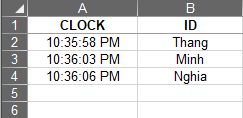
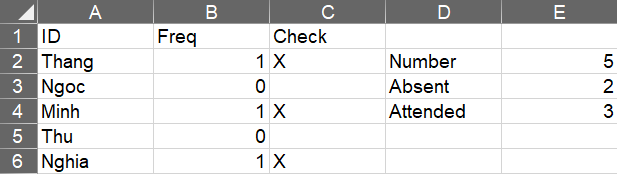
Ý tưởng: ta sẽ sử dụng một số hàm trong Excel:

* COUNTIF(range, criteria)
  + đếm số ô trong một phạm vi (range) thỏa một điều kiện (criteria)
* COUNTBLANK(range)
  + đếm các ô trống trong một phạm vi
* IF(logical\_test, value\_if\_true, value\_if\_false)
  + nếu điều kiện trong **logical test** là đúng thì sẽ thực hiện **value\_if\_true**, ngược lại thì thực hiện **value\_if\_false**
  + ta có thể **lồng** nhiều lệnh **IF** với nhau bằng cách đặt lệnh **IF** vào **value\_if\_true** hoặc **value\_if\_false.**

Vậy ta sẽ dùng hàm COUNTIF để đếm số lần mà một ID đã được điểm danh và xét nếu tần số của họ lớn hơn 0 thì đánh dấu có mặt. Bên cạnh đó, ta cũng sẽ sử dụng hàm COUNTBLANK để đếm số người đã điểm danh.

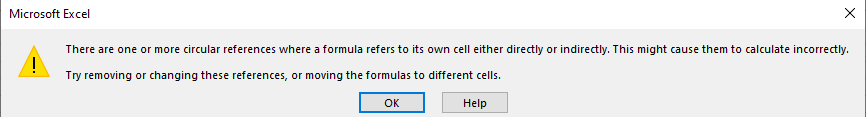
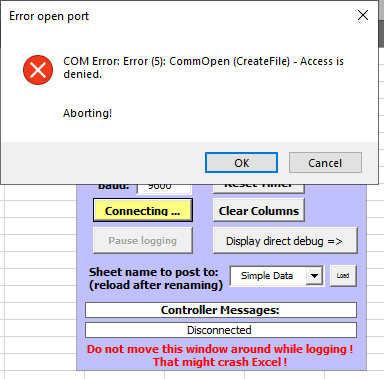
**Hướng dẫn chi tiết:**

* Bước 1: Trên file Excel, chọn **Further Sheet** ở gần góc trái dưới của file.
* Bước 2: Lần lượt nhập 5 ID mà ta đã lưu trên móc khoá vào từng dòng của cột A:
* 
* Bước 3:
  + Ở ô B2, nhập vào **=COUNTIF('Simple Data'!B:B, A2)**, nghĩa là đếm số giá trị giống với ID Thang trên **sheet Simple Data**.
  + Ở ô C2, nhập vào **=IF(B2>0, "X", "")**, nghĩa là nếu tần số xuất hiện của ID ở ô A2 lớn hơn 0 thì điểm danh ID có mặt bằng cách gán cho ô một **ký tự X**, ngược lại thì gán một **chuỗi rỗng**.
  + Ta di chuột vào mép phải dưới của ô B2 cho đến khi hiện dấu cộng, nhấp giữ và kéo xuống đến ô B6 để áp dụng lệnh đối với cả 5 ID. Thực hiện tương tự đối với các ô từ C2 → C6.
* Bước 4:
  + Ở ô E2, ta sẽ nhập vào sĩ số của lớp, cụ thể ở đây là 5.
  + Ở ô E3, nhập vào **=COUNTBLANK(C2:C6),** nghĩa là lấy giá đếm số ô trống trong phạm vi từ ô C2 đến ô C6. Giá trị của ô này bằng với số học sinh vắng của lớp.
  + Ở ô E4, nhập vào **=E2-E3**, cũng chính là sĩ số của lớp.
  + Sau khi nhập xong, chúng ta có như hình dưới đây:
  + 

Vậy khi kết hợp phần 3.1 và 3.2, nếu ta có các dữ liệu ở sheet Simple Data:  
   
 Thì ở bên sheet Further Sheet:  
 

Một số lưu ý:

Trong quá trình thực hiện và chạy thử dự án, mình có một số lưu ý như sau:

* Nếu bạn mua nhiều móc khoá hoặc thẻ từ để thực hiện dự án này, hãy đánh dấu chúng bằng cách ghi ID lên chúng bằng bút lông để tránh bị lẫn lộn.
* Nếu việc đọc móc khoá/thẻ từ gặp vấn đề (không đọc được, đọc bị lỗi,...) thì bạn có thể thử cắm lại dây USB kết nối với mạch Arduino.
* Các đoạn code thuộc thư viện MFRC522 trong **read.ino** và **write.ino** đều được tham khảo từ các Sketch mẫu của thư viện (ở phần File → Example → MFRC522), và ta không nên thay đổi chúng
* Câu lệnh IF ở Phần 3.2 ta cần lưu ý không được đặt điều kiện nằm ở ngoài cùng mà có **tham chiếu về ô chính nó** vì khi ấy chính ô đó vẫn chưa mang bất kỳ giá trị nào để so sánh (do đó giá trị trả về của lệnh IF đó sẽ **luôn bằng 0**)
  + Ví dụ: tại ô B1:   
    **=IF(B1=0,....)** thì không được,   
    còn **=IF(D2=D3, 1, IF(B1=0, 0, 1) )** thì hợp lệ.
* 
  + Đây là thông báo của Excel khi hàm ở một ô tham chiếu chính nó (circular reference).
* 
  + Đây là lỗi hiện lên khi bạn đang bật Serial Monitor ở Arduino IDE và bấm Connect trên bảng PLX-DAQ sau đó. Cách khắc phục là tắt Serial Monitor đi.

Kết:

Tuy đã dành kha khá thời gian để thực hiện hệ thống điểm danh tự động, mình cảm thấy tự hào vì nó hướng tới một mục đích thực tiễn và rất dễ nhận thấy. Hệ thống điểm danh tự động đã giúp lớp mình không cần phải trả qua việc được yêu cầu điểm danh bất chợt từ giáo viên và phải đếm thủ công lại từng người. Mặt khác, lớp mình luôn có thể trả lời chính xác sĩ số và những thành viên vắng mặt.   
 Qua quá trình thực hiện dự án, mình cũng đã học được rất nhiều kiến thức mới, từ cách hoạt động của hệ thống RFID cho đến việc hiển thị và quản lý dữ liệu trên Excel. Bên cạnh đó, mình đúc kết được nhiều bài học quý giá, một trong số đó là việc lên kế hoạch rõ ràng và khoa học trước khi thực hiện một dự án sẽ giúp ta tiết kiệm rất nhiều thời gian và công sức. Nhờ phân tích các đoạn code dài mà ban đầu mình viết nên thành những hàm con súc tích, mình mới có thể ước lượng thời gian hợp lý cho việc phân tích các đoạn code mà mình.

Tham khảo:

* [Cách hoạt động của RFID và cách tạo khóa cửa dựa trên Arduino | Việt Machine (vietmachine.com.vn)](https://vietmachine.com.vn/cach-hoat-dong-cua-rfid-va-cach-tao-khoa-cua-dua-tren-arduino.html)
* [Công Nghệ RFID Là Gì? Nguyên Lý Hoạt Động Và Ứng Dụng Của RFID | Tinh tế (tinhte.vn)](https://tinhte.vn/thread/cong-nghe-rfid-la-gi-nguyen-ly-hoat-dong-va-ung-dung-cua-rfid.3365075/)
* [In-Depth: What is RFID? How It Works? Interface RC522 with Arduino](https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/)
* [(lastminuteengineers.com)](https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/)
* [Arduino và giao tiếp SPI | Cộng đồng Arduino Việt Nam](http://arduino.vn/bai-viet/1081-arduino-va-giao-tiep-spi)
* [Giao tiếp SPI (deviot.vn)](https://deviot.vn/blog/giao-tiep-spi.74706311)
* [Chuẩn giao tiếp SPI | #3\_Giao tiếp điện tử cơ bản (kysungheo.com)](https://kysungheo.com/chuan-giao-tiep-spi/)