



IOT201 - LẬP TRÌNH IOT CƠ BẢN

BÀI 2.1: GIAO TIẾP VỚI NFC

- ⊙ Kết thúc bài học này, sinh viên có khả năng
 - ⊙ Lập trình giao tiếp NFC
 - ⊙ Lập trình giao tiếp I2C





Giới thiệu NFC



Lập trình giao tiếp với NFC



Giới thiệu I2C



Lập trình giao tiếp với I2C

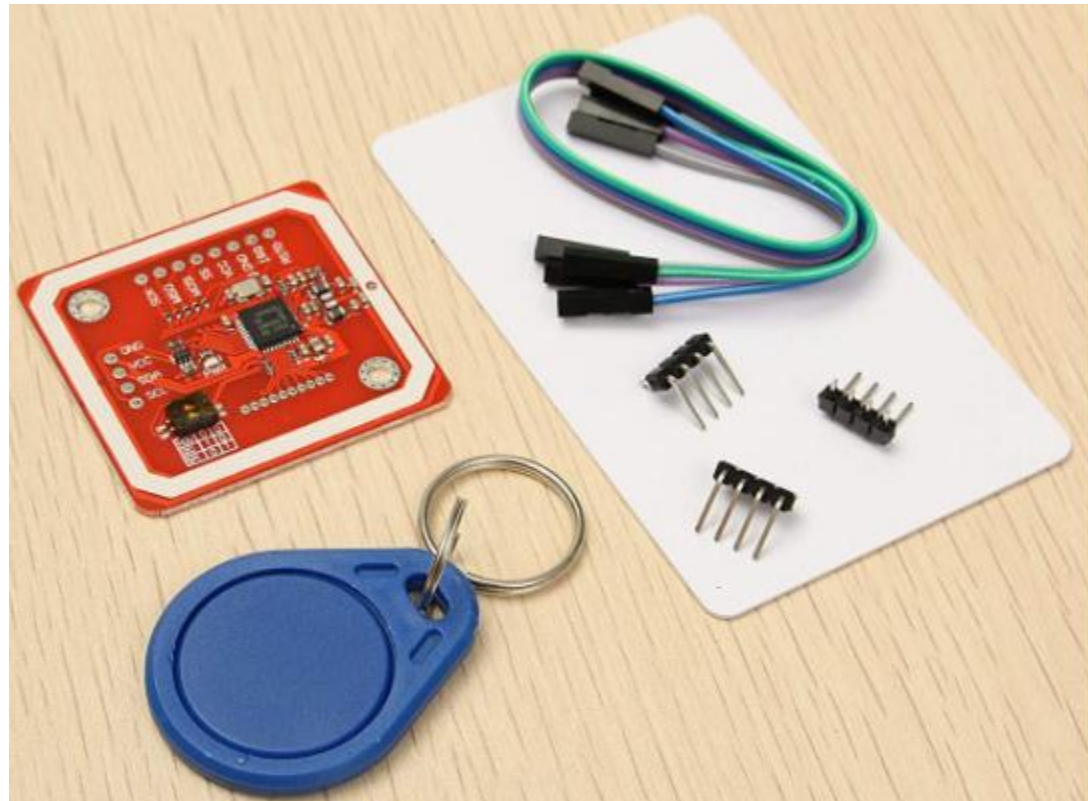


- ❑ Near field communication (NFC) là một bộ tiêu chuẩn dành cho điện thoại thông minh (hay các thiết bị tương tự) để thiết lập sự giao tiếp vô tuyến với nhau khi chúng được đặt gần nhau, thường trong khoảng cách vài cm.



- ❑ Công nghệ NFC sử dụng sóng vô tuyến để nhận dạng một cách tự động những đối tượng vật lý từ những vật thể sống đến những vật thể thụ động.
- ❑ Có thể nói, NFC là một công nghệ nhận dạng tự động(Auto-ID).
- ❑ Một số ví dụ của công nghệ có thể đã từng thấy như nhận dạng mã vạch, nhận dạng giọng nói...

- ❑ NFC là một công nghệ phổ biến những năm gần đây, ta thường được nghe đến điều này từ các hãng sản xuất điện thoại di động nổi tiếng và hầu hết các điện thoại cao cấp đều có hỗ trợ NFC.



❑ Chuẩn bị thiết bị

- ❖ 1 x [Arduino UNO](#)
- ❖ 1 x [Module RFID](#)
- ❖ 1 x [Servo](#) 90g (để điều khiển cửa đóng mở)



Nối dây

Arduino UNO	Module RFID	Servo
11	MOSI	
12	MISO	
13	SCK	
10	SDA	
9	RST	
3.3V	3.3V	
GND	GND	GND
5V		5V

Coding

```
#include <Servo.h>
#include <SPI.h>
#include <RFID.h>
#define SS_PIN 10
#define RST_PIN 9
Servo myservo;
RFID rfid(SS_PIN, RST_PIN);
```

❑ Coding

```
1.#include <Servo.h>
2.#include <SPI.h>
3.#include <RFID.h>
4.#define SS_PIN 10
5.#define RST_PIN 9
6.Servo myservo;
7.RFID rfid(SS_PIN, RST_PIN);
8.unsigned char reading_card[5]; // Mảng đọc mã card
9.unsigned char master[5] = { 134, 138, 22, 126, 100 }; // Mã Card phù hợp để mở cửa
10.unsigned char slave[5] = { 246, 253, 148, 53, 170 }; // Mã Card phù hợp để đóng cửa
11.unsigned char i, j;
12.void setup() {
```

❑ Coding

```
13.my servo.attach(8);  
14.Serial.begin(9600);  
15.SPI.begin();  
16.rfid.init();  
17.my servo.write(30);}  
18.void loop(){  
19.if (rfid.isCard()) {  
20.if (rfid.readCardSerial()) // Nếu có thẻ  
21.{  
22.for (i = 0; i < 5; i++) {  
23.reading_card[i] = rfid.serNum[i]; //Lưu  
    mã thẻ đọc được vào mảng reading_card  
24.}
```

❑ Coding

```
25. Serial.println();
26. //verification
27. for (i = 0; i < 5; i++) {
28. //So sánh từng phần tử của mảng reading_card với mảng
    master
29. if (reading_card[i] != master[i]){
30. break;}} Tương tự với thẻ Slave
31. for (j = 0; j < 5; j++) {
32. if (reading_card[i] != slave[i]) {
33. break;}}
34. if (i == 5) // Nếu các phần tử của mảng reading_card
    phù hợp hết với mảng master thì lúc này i chạy đến 5
35. {
36. myservo.write(180);} // đóng cửa
37. if (j == 5) {
38. myservo.write(30); // mở cửa
39. }}
40. rfid.halt();}}
```



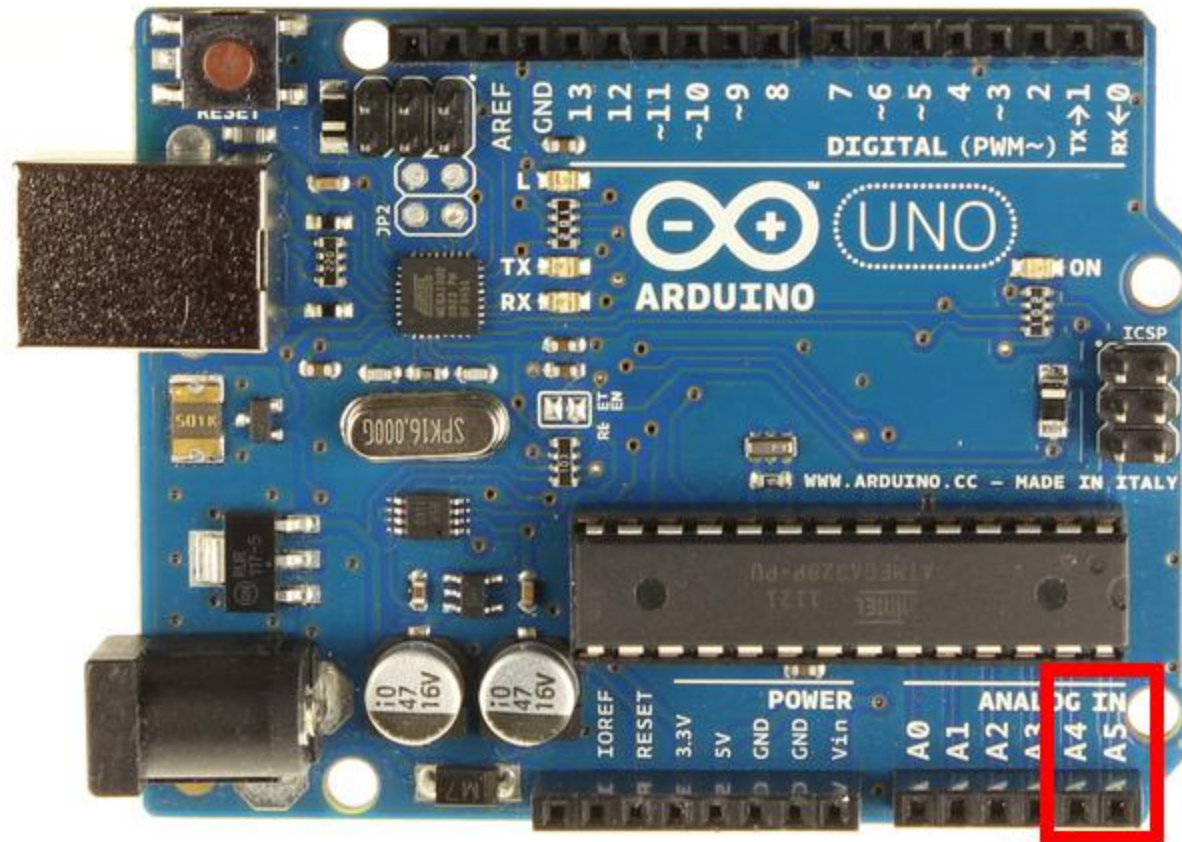
LẬP TRÌNH IOT CƠ BẢN

BÀI 2.2: GIAO TIẾP VỚI I2C

- ❑ I2C là viết tắt của "Inter-Integrated Circuit", một chuẩn giao tiếp được phát minh bởi Philips'.
- ❑ Nhằm đơn giản hóa việc trao đổi dữ liệu giữa các ICs. Đôi khi nó cũng được gọi là Two Wire Interface (TWI) vì chỉ sử dụng 2 kết nối để truyền tải dữ liệu, 2 kết nối của giao tiếp I2C gồm: SDA (Serial Data Line) và SCL (Serial Clock Line)

- ❑ Có hàng ngàn thiết bị sử dụng giao tiếp I2C, chẳng hạn như real-time clocks, digital potentiometers, temperature sensors, digital compasses, memory chips, FM radio circuits, I/O expanders, LCD controllers, amplifiers,...
- ❑ Trên board Arduino UNO, SDA là chân analog 4, SCL là chân analog 5.

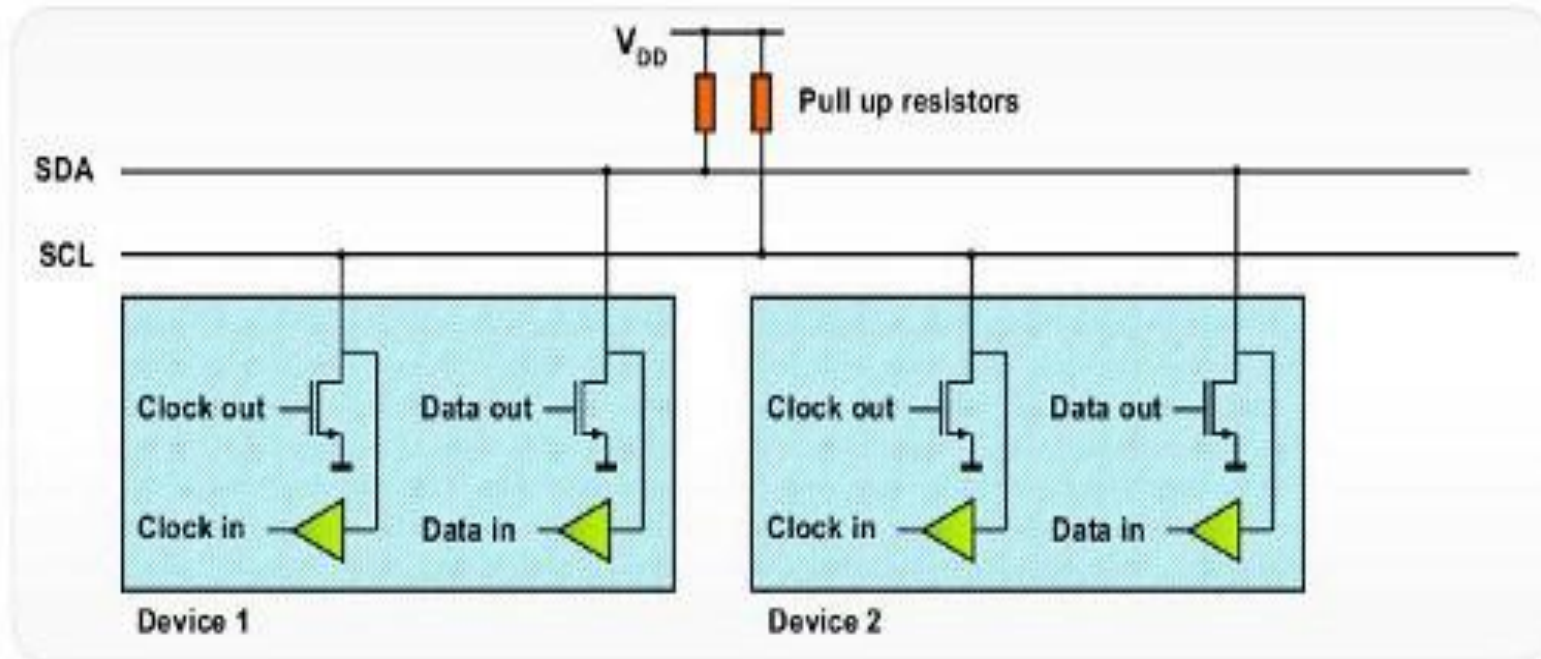
GIỚI THIỆU GIAO THỨC I2C



I2C Pins

- ❑ Trên I2C bus, sẽ có một thiết bị được coi là "Master", và trong hầu hết các trường hợp, Arduino là một "Master", mỗi IC được gắn trên I2C bus là một "Slave".
- ❑ Mỗi "slave" có một địa chỉ riêng ở dạng HEX (thập lục phân) để Arduino ("Master") có thể giao tiếp với nó.

GIỚI THIỆU GIAO THỨC I2C



- 2 wire bus:
 - SDA: Serial Data Line
 - SCL: Serial Clock Line

- ❑ Để I2C bus có thể hoạt động, chúng ta cần 2 điện trở pull-up, 4.7k.
- ❑ Tuy nhiên, nếu chỉ có một thiết bị I2C, chúng ta có thể bỏ qua điện trở pull-up vì trong MCU ATmega328 của Arduino UNO đã có sẵn 20k build-in resistors.

❑ Để sử dụng I2C bus trên Arduino, chúng ta sẽ cần sử dụng thư viện Wire.h (đây là built-in library của Arduino).

1. `Wire.begin(address (optional)) ;`

- ✓ Khởi tạo thư viện Wire.h và tham gia vào I2C bus.
- ✓ address: 7-bit địa chỉ của thiết bị "Slave" (optional); nếu không có địa chỉ thì coi như "Master".

2. `Wire.beginTransmission(address) ;`

- ✓ Bắt đầu truyền dữ liệu đến thiết bị "Slave" với address đã có.

1. `Wire.endTransmission()` ;

- ✓ Kết thúc truyền dữ liệu đến thiết bị "Slave" đã được bắt đầu bởi `Wire.beginTransmission(address)`.

2. `Wire.write(value)` ;

- ✓ Ghi dữ liệu lên thiết bị "Slave", được gọi giữa `beginTransmission()` và `endTransmission()`.

3. `Wire.read()` ;

- ✓ Đọc dữ liệu được truyền từ thiết bị "Slave" đến Arduino, được gọi sau `requestFrom()`.

4. `Wire.requestFrom(address, quantity)` ;

- ✓ Được sử dụng bởi thiết bị "Master" để yêu cầu dữ liệu từ thiết bị "Slave".
- ✓ `address`: là địa chỉ của thiết bị "Slave".
- ✓ `quantity`: số lượng bytes yêu cầu.

- ☑ Lập trình giao tiếp với NFC
- ☑ Lập trình giao tiếp với I2C





Cảm ơn