






## IOT201 - LẬP TRÌNH IOT CƠ BẢN

### BÀI 3.1: LẬP TRÌNH ĐIỀU KHIỂN BẰNG BLUETOOTH

- ⊙ Kết thúc bài học này, sinh viên có khả năng
  - ⊙ Lập trình điều khiển bluetooth bằng điện thoại android.
  - ⊙ Lập trình giao tiếp SPI



-  Giới thiệu
-  Module bluetooth hc06
-  Giao tiếp SPI



- ❑ Kết nối Arduino với các thiết bị, module điện tử khác là 1 việc tương đối dễ dàng.
- ❑ Kết nối Arduino với module Bluetooth và điều khiển nó thông qua điện thoại Android.
- ❑ Có thể ứng dụng để điều khiển các thiết bị điện trong nhà như bật tắt đèn, quạt, bật tắt bình nóng lạnh, ... hoặc tự làm một chiếc ô tô điều khiển từ xa bằng Bluetooth.

## □ Chuẩn bị

- ❖ Arduino Uno
- ❖ Module Bluetooth HC06
- ❖ Điện thoại chạy hệ điều hành Android
- ❖ Breadboard
- ❖ Dây cắm
- ❖ Đèn led
- ❖ Điện trở 560 Ohm

# GIỚI THIỆU MODULE BLUETOOTH HC06



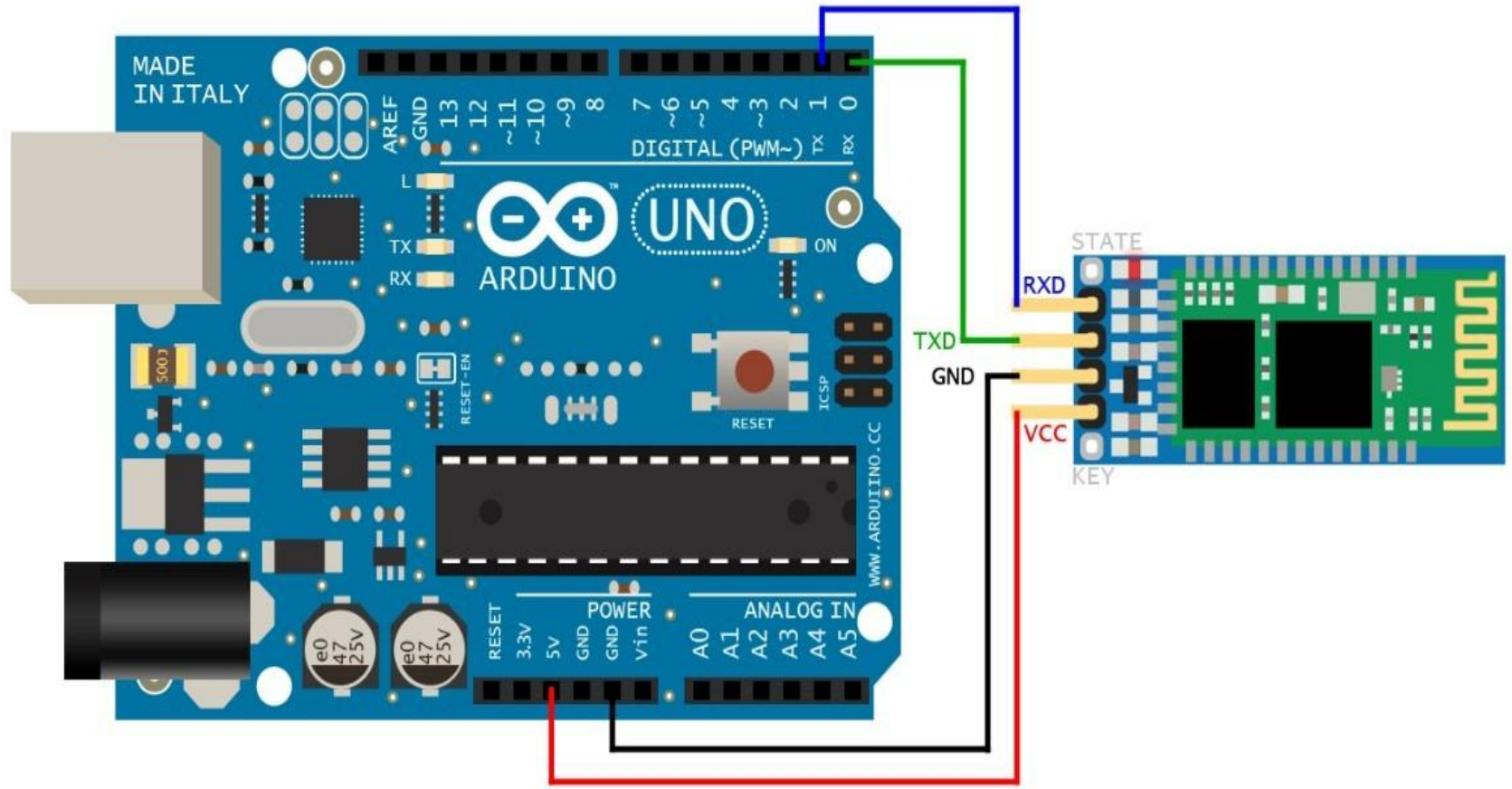
- ❑ Module này gồm 4 chân GND, VCC, TX, RX Khi kết nối chỉ cần nối chân TX với chân 0 và chân RX nối với chân 1 trên Arduino sau đó có thể lập trình gửi và nhận dữ liệu như 1 cổng Serial thông thường.
- ❑ Module này có 3 loại Master, Slave và loại chạy được cả 2 chế độ Master và Slave, ở ví dụ này sử dụng loại Slave, khi kết nối với điện thoại điền mật khẩu mặc định là: 1234

- ❑ Phần mềm trên điện thoại Android trong ví dụ này sử dụng phần mềm [DroidDuino](#), có thể tìm và tải trên Google Play.
- ❑ Giao diện và cách sử dụng phần mềm rất đơn giản, dễ sử dụng.





## □ Sơ đồ kết nối

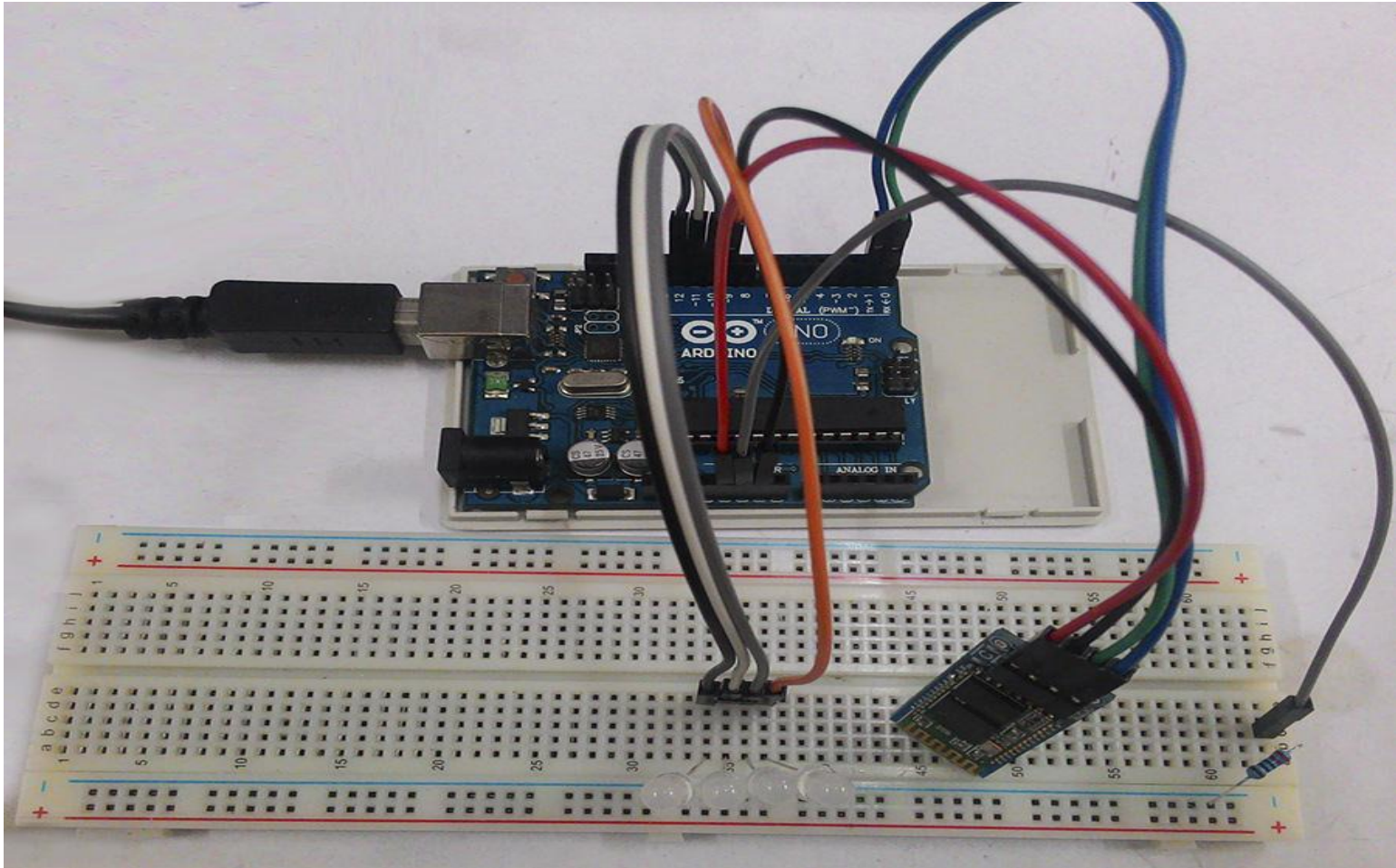


## □ Lập trình

```
1. // Khai báo biến
2. char state;
3. void setup() {
4. // Cài đặt các chân bạn muốn điều khiển thành thành Ouput
5. // Ở đây tôi sử dụng 4 chân 9, 10, 11, 12
6. pinMode(9, OUTPUT);
7. pinMode(10, OUTPUT);
8. pinMode(11, OUTPUT);
9. pinMode(12, OUTPUT);
10. Serial.begin(9600); // Kết nối bluetooth module ở tốc độ 9600
11.}
12. void loop() {
13.
14. if(Serial.available() > 0){
15. // Đọc giá trị nhận được từ bluetooth
16. state = Serial.read();
17. } else
18. state = 0;
19. Serial.println(state); //Thực hiện điều khiển các chân 9, 10, 11, 12
```

```
20.     switch (state) {
21.     case '1':
22.     digitalWrite(9, HIGH);
23.     break;
24.     case '2':
25.     digitalWrite(9, LOW);
26.     break;
27.     case '3':
28.     digitalWrite(10, HIGH);
29.     break;
30.     case '4':
31.     digitalWrite(10, LOW);
32.     break;
33.     case '5':
34.     digitalWrite(11, HIGH);
35.     break;
36.     case '6':
37.     digitalWrite(11, LOW);
38.     break;
39.     case '7':
40.     digitalWrite(12, HIGH);
41.     break;
42.     case '8':
43.     digitalWrite(12, LOW);
44.     break;
45.     default:
46.     break;
47.     }}
```

## □ Hình ảnh thực tế





# LẬP TRÌNH IOT CƠ BẢN

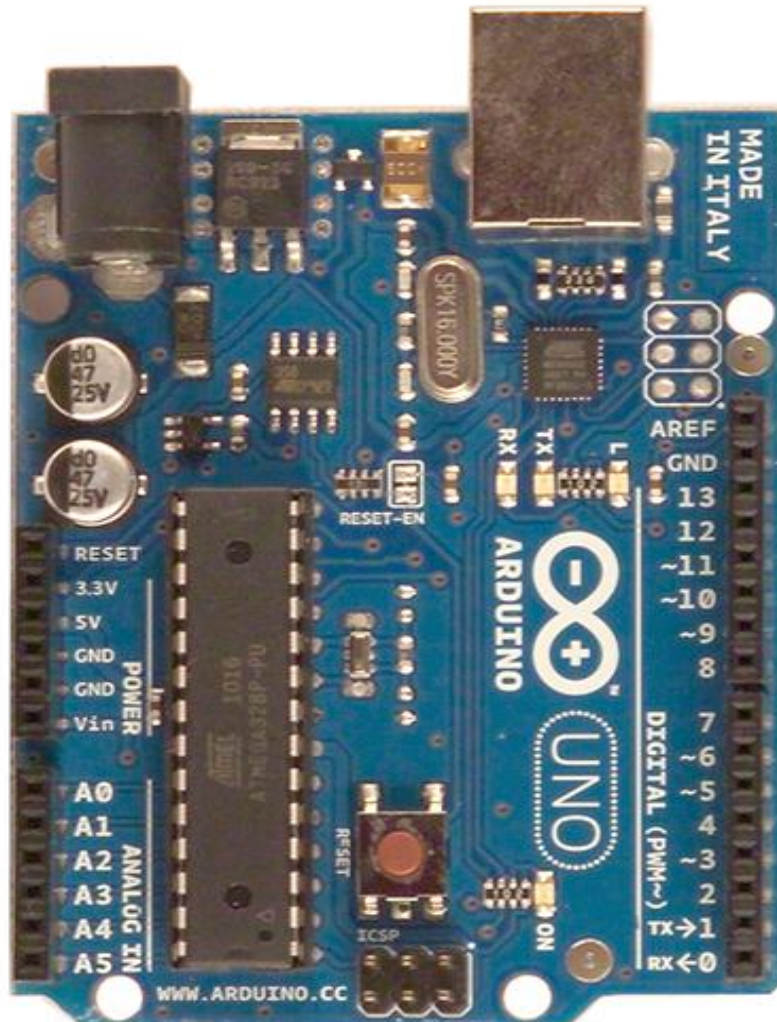
---

## BÀI 3.2: GIAO TIẾP SPI

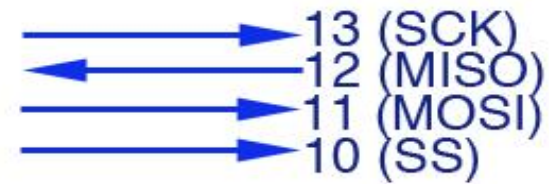
- ❑ SPI (Serial Peripheral Bus) là một chuẩn truyền thông nối tiếp tốc độ cao do hãng Motorola đề xuất.
- ❑ Đây là kiểu truyền thông Master-Slave, trong đó có 1 chip Master điều phối quá trình truyền thông và các chip Slaves được điều khiển bởi Master vì thế truyền thông chỉ xảy ra giữa Master và Slave.

- ❑ 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).





SPI pins

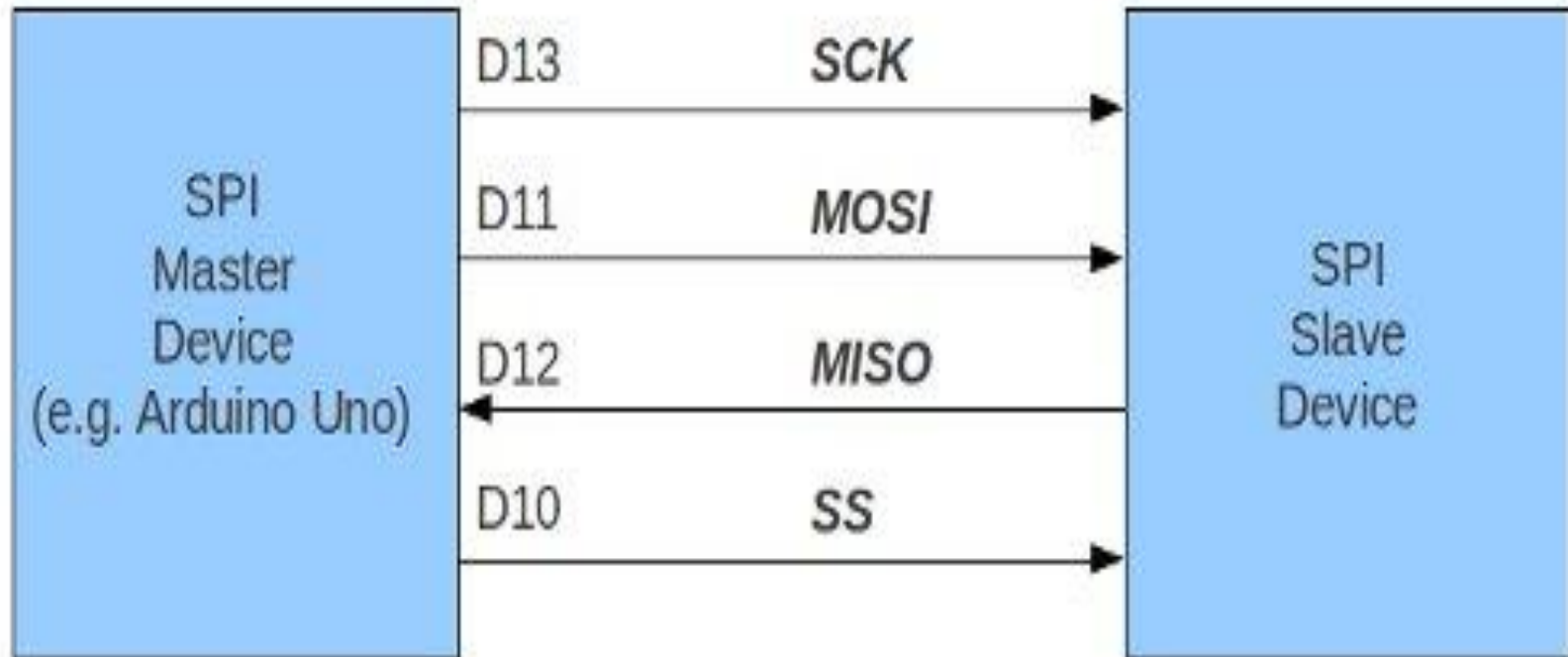


- ❑ **SCK:** Xung giữ nhịp cho giao tiếp SPI, vì SPI là chuẩn truyền đồng bộ nên cần 1 đường giữ nhịp, mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi.
- ❑ Đây là điểm khác biệt với truyền thông không đồng bộ mà chúng ta đã biết trong chuẩn UART. Sự tồn tại của chân SCK giúp quá trình truyền ít bị lỗi và vì thế tốc độ truyền của SPI có thể đạt rất cao. Xung nhịp chỉ được tạo ra bởi chip Master.

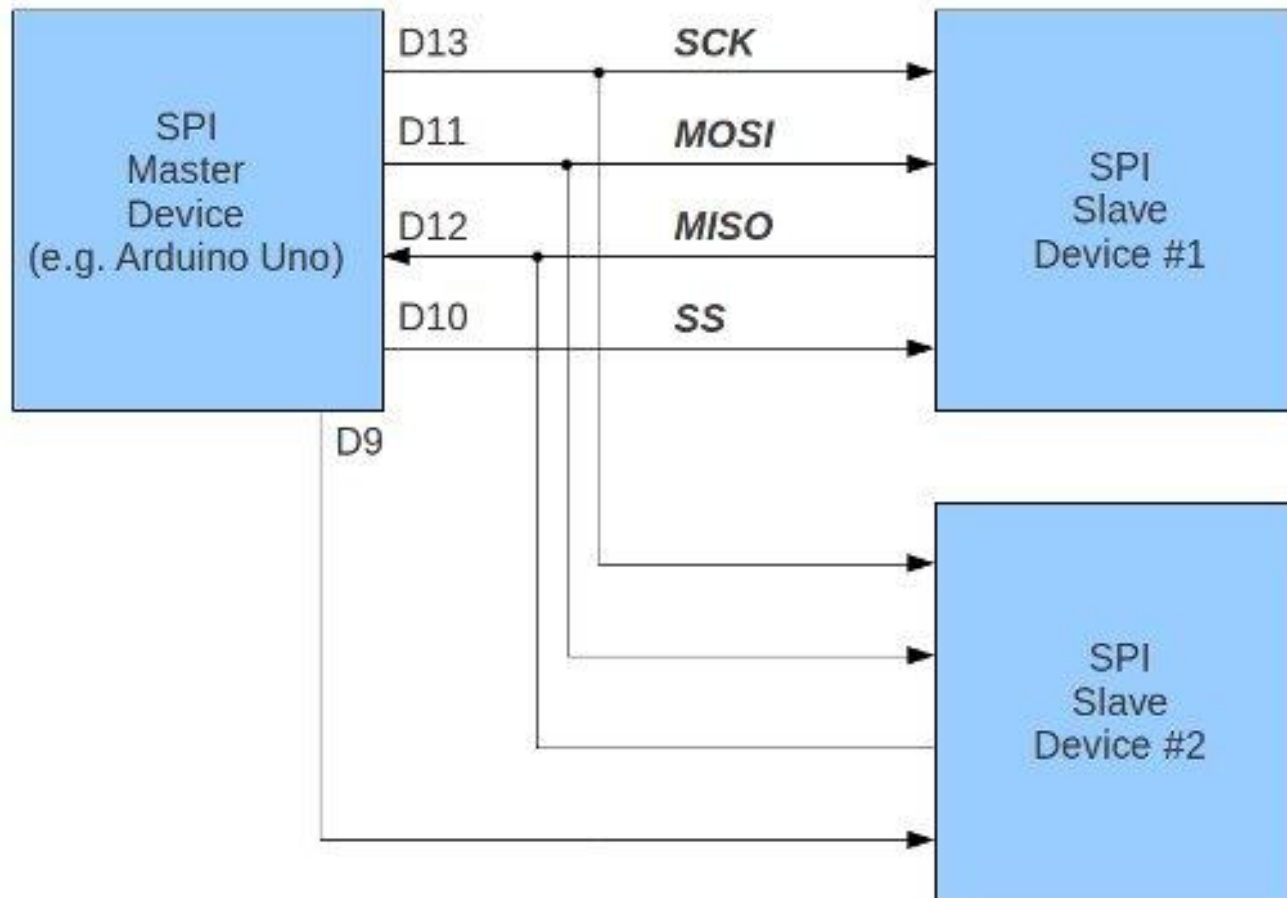
- ❑ **MISO– Master Input / Slave Output:** nếu là chip Master thì đây là đường Input còn nếu là chip Slave thì MISO lại là Output. MISO của Master và các Slaves được nối trực tiếp với nhau..
- ❑ **MOSI – Master Output / Slave Input:** nếu là chip Master thì đây là đường Output còn nếu là chip Slave thì MOSI là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.

- ❑ **SS – Slave Select:** SS là đường chọn Slave cần giao tiếp, trên các chip Slave đường SS sẽ ở mức cao khi không làm việc.
- ❑ Nếu chip Master kéo đường SS của một Slave nào đó xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó.
- ❑ Chỉ có 1 đường SS trên mỗi Slave nhưng có thể có nhiều đường điều khiển SS trên Master, tùy thuộc vào thiết kế của người dùng.

- ❑ **MISO** - Mang các dữ liệu từ các thiết bị SPI về arduino
- ❑ **MOSI** - Mang các dữ liệu từ Arduino đến các thiết bị SPI
- ❑ **SS** - Chọn thiết bị SPI cần làm việc
- ❑ **SCK** - dòng đồng bộ



- ❑ Dữ liệu được truyền qua lại giữa 2 đường MISO và MOSI. Điều này chỉ thực hiện được khi Dòng SS được thiết lập ở mức thấp LOW.
- ❑ Nói cách khác, để giao tiếp với một thiết bị SPI chúng ta cần thiết lập các dòng SS với thiết bị ở mức thấp LOW, sau đó giao tiếp với nó, sau đó thiết lập các dòng SS trở lại mức cao HIGH. Nếu chúng ta có hai hoặc nhiều thiết bị SPI trên cùng 1 bus, chúng sẽ được kết nối như sau





- ☑ Lập trình điều khiển bluetooth bằng điện thoại android.
- ☑ Lập trình giao tiếp SPI.





**Cảm ơn**