### 2.Modul RFID

****

**Mô tả:**

* Đây là module dựa trên module RF RC522 gần module giao tiếp trường
* Tần số làmviệc là 13.56MHz và bạn có thể đọc và ghi tag 13.56MHz với module này.

**Đặc tính:**

* Gia tiếp trường tiêu chuẩn, tương thích hoàn hảo với Arduino UNO / MEGA2560.
* Dễ sử dụng với thư viện của Arduino: RFIDMaster.
* Điện áp hoạt động: 4.5~5.2V.
* Mức logic giao tiếp: Onboard level converter chip, tương thích với 3.3V / 5V.
* Dòng hoạt động: 13-~26mA.
* Dòng Idle: 10~13mA.
* Dòng nghỉ: <80uA.
* Dòng đỉnh: <50mA.
* Tần số hoạt động: 13.56MHz.
* Hỗ trợ những dạng card: mifare1 S50, mifare1 S70 MIFARE Ultralight, mifare Pro, MIFARE DESFire.
* Tốc độ truyền dữ liệu: Tối đa 10Mbit/s.

**Cấu tạo:** Một thiết bị hay một hệ thống RFID được cấu tạo bởi hai thành phần chính là thiết bị đọc ( reader) và thiết bị phát mã RFID có gắn chip hay còn gọi là tag. Thiết bị đọc được gắn antenna để thu- phát sóng điện từ, thiết bị phát mã RFID tag được gắn với vật cần nhận dạng, mỗi thiết bi RFID tag chứa một mã số nhất định và không trùng lặp nhau.  
**Nguyên lý hoạt động:** Thiết bị RFID reader phát ra sóng điện từ ở một tần số nhất định, khi thiết bị RFID tag trong vùng hoạt động sẽ cảm nhận được sóng điện từ này và thu nhận năng lượng từ đó phát lại cho thiết bị RFID Reader biết mã số của mình. Từ đó thiết bị RFID reader nhận biết được tag nào đang trong vùng hoạt động.

**Cấu tạo của thể MIFARE**

Cấu trúc trong thẻ MEFIARE 13.56 Mhz

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sector | Block | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 | Serial Number | | | | CB | Manufactor Data | | | | | | | | | | | |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Key A | | | | | | FF | 07 | 08 | xx | Key B | | | | | |
| 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Key A | | | | | | FF | 07 | 08 | xx | Key B | | | | | |
| 2 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Key A | | | | | | FF | 07 | 08 | xx | Key B | | | | | |
|  | …. | … | … | … | … |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Key A | | | | | | FF | 07 | 08 | xx | Key B | | | | | |

CB: Serial number check key

Serial Numbers: xx xx xx xx hex, mỗi thẻ có giá trị khác nhau, do nhà sản xuất quy định, Block 0 của Sector 0, không thể ghi dữ liệu đè lên được.

Block: 1,2 dùng để chứa dữ liệu

Block3: Key A: mật khầu. Key B vửa là mật khẩu vừa là dữ liệu nếu cần thiết.

Mỗi Sector có một Key truy nhập riêng, Pass mặt định khi thẻ mới mua về là: FF FF FF FF FF FF

**Bước 1: Dò tìm kiểm tra có thể trên module nhận thẻ**

status = nfc.requestTag(MF1\_REQIDL, data); 🡺 status trả về giá trị là MI\_OK nếu có thể trên module nhận thẻ.

**Bước 2: Kiểm tra có nhiều thẻ đặt lên một module nhận thẻ**

status = nfc.antiCollision(data); 🡺 data sẽ chứa giá trị của Block 0 của Sector 0.

**Bước 3: Chỉ ra thẻ muốn truy xuất**

nfc.selectTag(serial); 🡺 trong dó serial chứa mã thẻ (5 byte đầu tiên của block 0, sector 0) của thẻ cần truy nhập.

**Bước 4: Truy nhập mật khẩu thẻ**.

status = nfc.authenticate(MF1\_AUTHENT1A, i, keyA, serial); 🡺Statuc trả vê MI\_OK nếu thẻ được đăng nhập, giá trị i là Block cần đăng nhập ( 0 – 63 ), key A là pass, serial chứa mã thẻ.

**Bước 5: Đọc hoặc ghi thẻ**

status = nfc.writeToTag(i, data\_write); 🡺 nếu status bằng MI\_OK thì đã ghi thành công, i là giá trị Block cần ghi, data\_write chứa giá trị cần ghi ( 16 byte).

status = nfc.readFromTag(i, data); 🡺 nếu status bằng MI\_OK thì đã đọc thành công, i là giá trị Block cần ghi, data chứa giá trị đọc về ( 16 byte).