MFRC522 RFID Module

Introduction

MFRC522 is a kind of integrated read and write card chip. It is commonly used in the radio at 13.56MHz. Launched by the NXP Company, it is a low-voltage, low-cost, and small-sized non-contact card chip, a best choice of intelligent instrument and portable handheld device.

The MF RC522 uses advanced modulation and demodulation concept which fully presented in all types of 13.56MHz passive contactless communication methods and protocols. In addition, it supports rapid CRYPTO1 encryption algorithm to verify MIFARE products. MFRC522 also supports MIFARE series of high-speed non-contact communication, with a two-way data transmission rate up to 424kbit/s. As a new member of the 13.56MHz highly integrated reader card series, MF RC522 is much similar to the existing MF RC500 and MF RC530 but there also exists great differences. It communicates with the host machine via the serial manner which needs less wiring. You can choose between SPI, I2C and serial UART mode (similar to RS232), which helps reduce the connection, save PCB board space (smaller size), and reduce cost.

Features

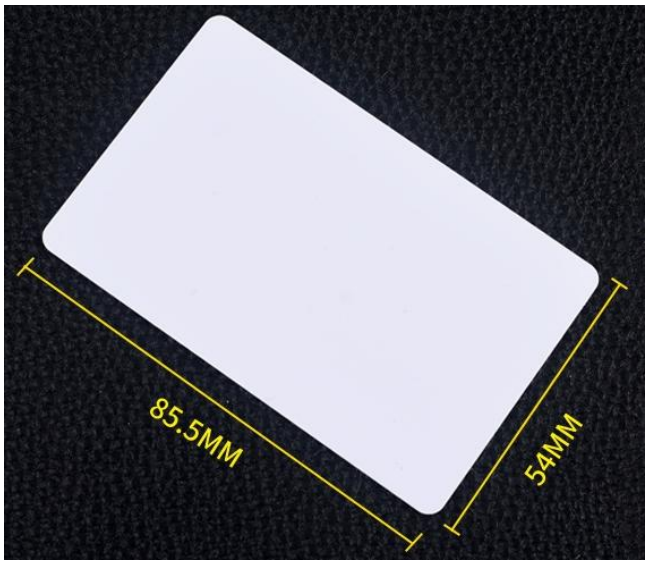
MF RC522 Module

- Operating Current: 13—26mA / DC 3.3V
- Idle Current: 10-13mA / DC 3.3V
- Sleep Current: <80uA
- Peak Current: <30mA
- Operating Frequency: 13.56MHz
- Supported Card Types: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire
- Dimensions: 59.9mm x 39.7mm
- Operating Ambient Temperature: -20 to 80 degrees Celsius
- Storage Temperature: -40 to 85 degrees Celsius
- Relative Humidity: 5%—95% non-condensing

Blue Buckle IC Key

- Frequency: 13.56MHz
- Size: 28x35.3x4(mm)
- Encapsulated with an ABS shell and filled with epoxy resin.

IC White Card



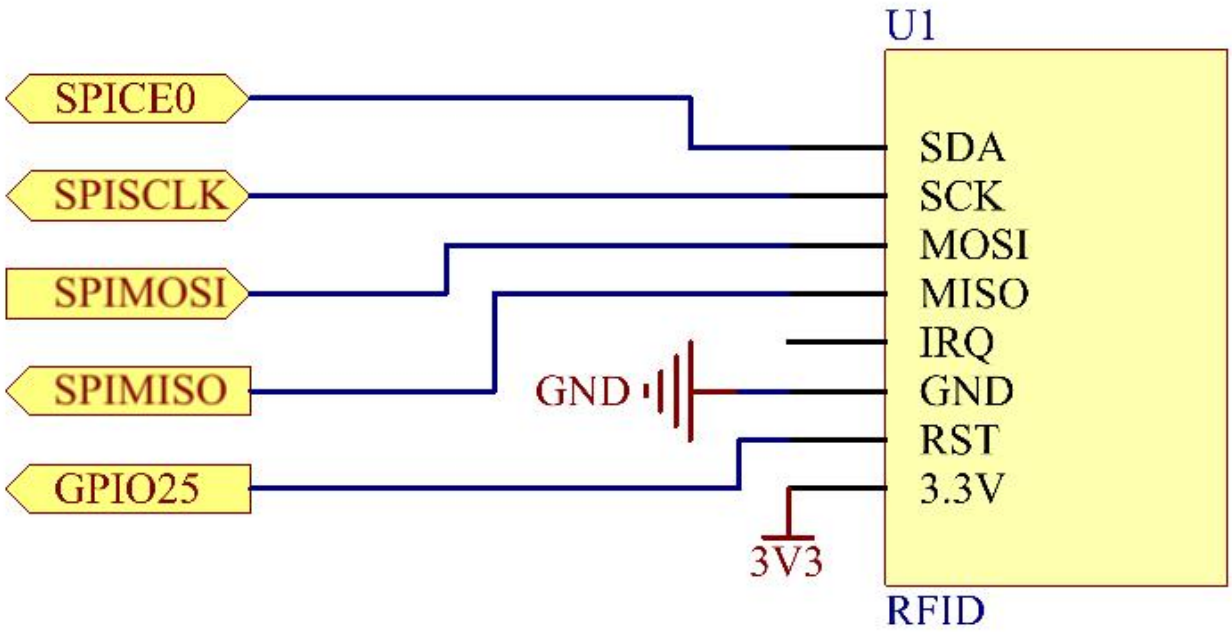
- Chip Type: IC
- Storage Capacity: 8Kbit, 16 partitions.
- Operating Frequency: 13.56MHz
- Communication Speed: 106KBoud
- Read/Write Distance: 2.5~10cm
- Read/Write Time: 1~2ms
- Operating Temperature: -20°C to +35°C
- Erase Life: >100,000 cycles
- Data Retention: >10 years
- Dimensions: ISO standard card, 85.5x54mm

- Packaging Material: PVC, PET, 0.13mm copper wire

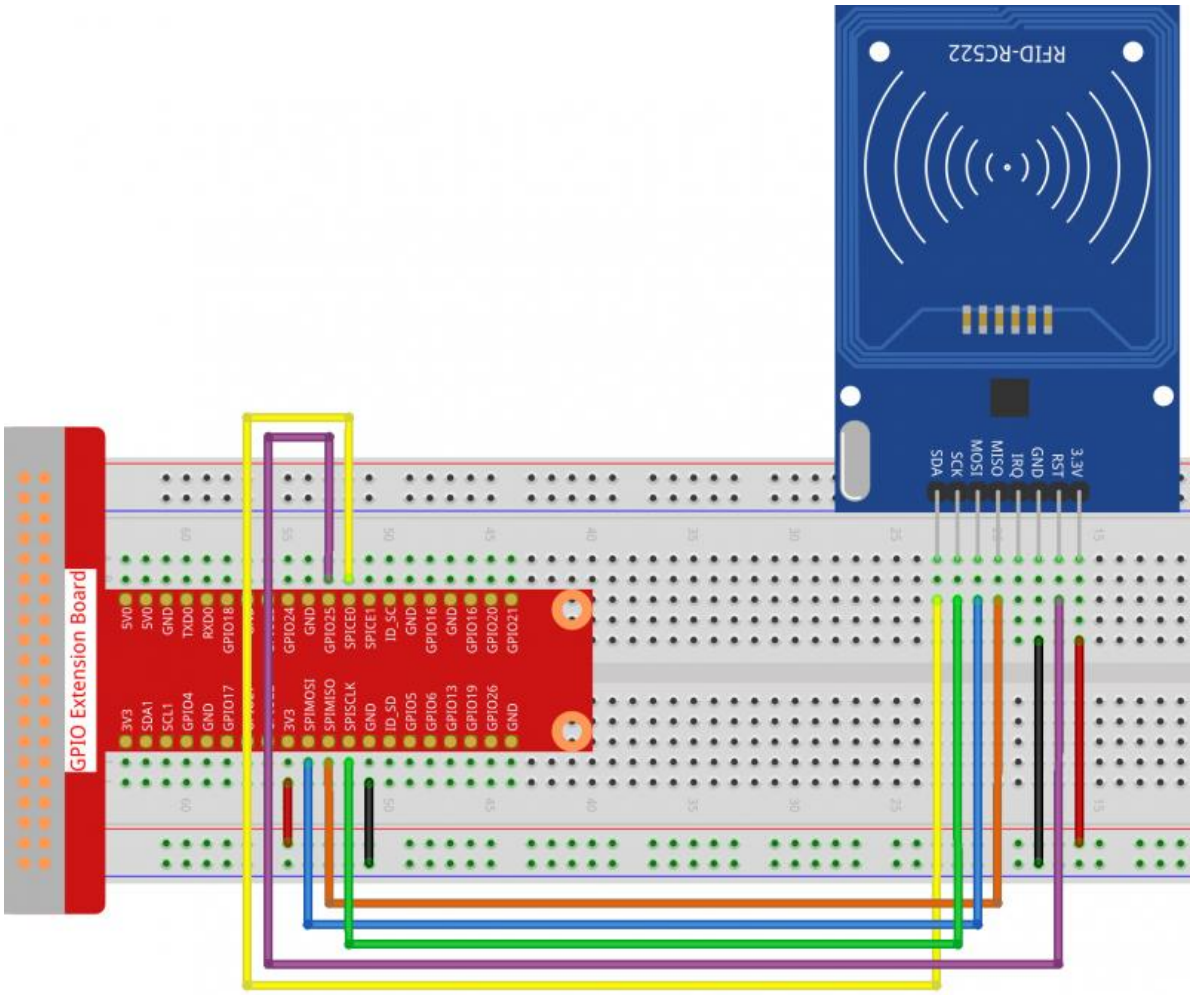
How to Use in Raspberry Pi?

Schematic Diagram

T-Board Name	physical	wiringPi	BCM
SPICE0	Pin 24	10	8
SPISCLK	Pin 23	14	11
SPIMOSI	Pin 19	12	10
SPIMISO	Pin 21	13	9
GPIO25	Pin 22	6	25



Build the Circuit



Setup SPI

To begin, enable the SPI port on your Raspberry Pi. If it's already enabled, you can skip this step. If you're uncertain, continue with the following:

```
sudo raspi-config
```

Navigate using the arrow keys to **3 Interfacing options** -> **SPI** -> **Yes** -> **OK** to complete the SPI setup.

Install Spidev and MFRC522

The spidev library facilitates interactions with the SPI, making it crucial for this tutorial. We'll need it to allow the Raspberry Pi to communicate with the RFID RC522. Update spidev on your Raspberry Pi using the following command:

```
sudo pip3 install --upgrade spidev
```

Next, install the MFRC522 library:

```
sudo pip3 install mfrc522
```

The MFRC522 library consists of two primary files: MFRC522.py and SimpleMFRC522.py. The former handles the core interaction with the RFID RC522 through the Pi ' s SPI interface. In contrast, SimpleMFRC522.py streamlines this by providing a more straightforward set of functions.

Run the Code

Start by downloading the code:

```
cd ~/
wget https://raw.githubusercontent.com/sunfounder/raphael-kit/master/python/2.2.10_write.py
cd ~/
wget https://raw.githubusercontent.com/sunfounder/raphael-kit/master/python/2.2.10_read.py
```

Execute the following to write a message to the card:

```
sudo python3 2.2.10_write.py
```

After launching 2.2.10_write.py, input your desired message and press Enter. Place the card on the MFRC522 module and wait for the "Data writing is complete" notification. You can then remove the card, or rewrite the message on a different card. Exit using Ctrl+C.

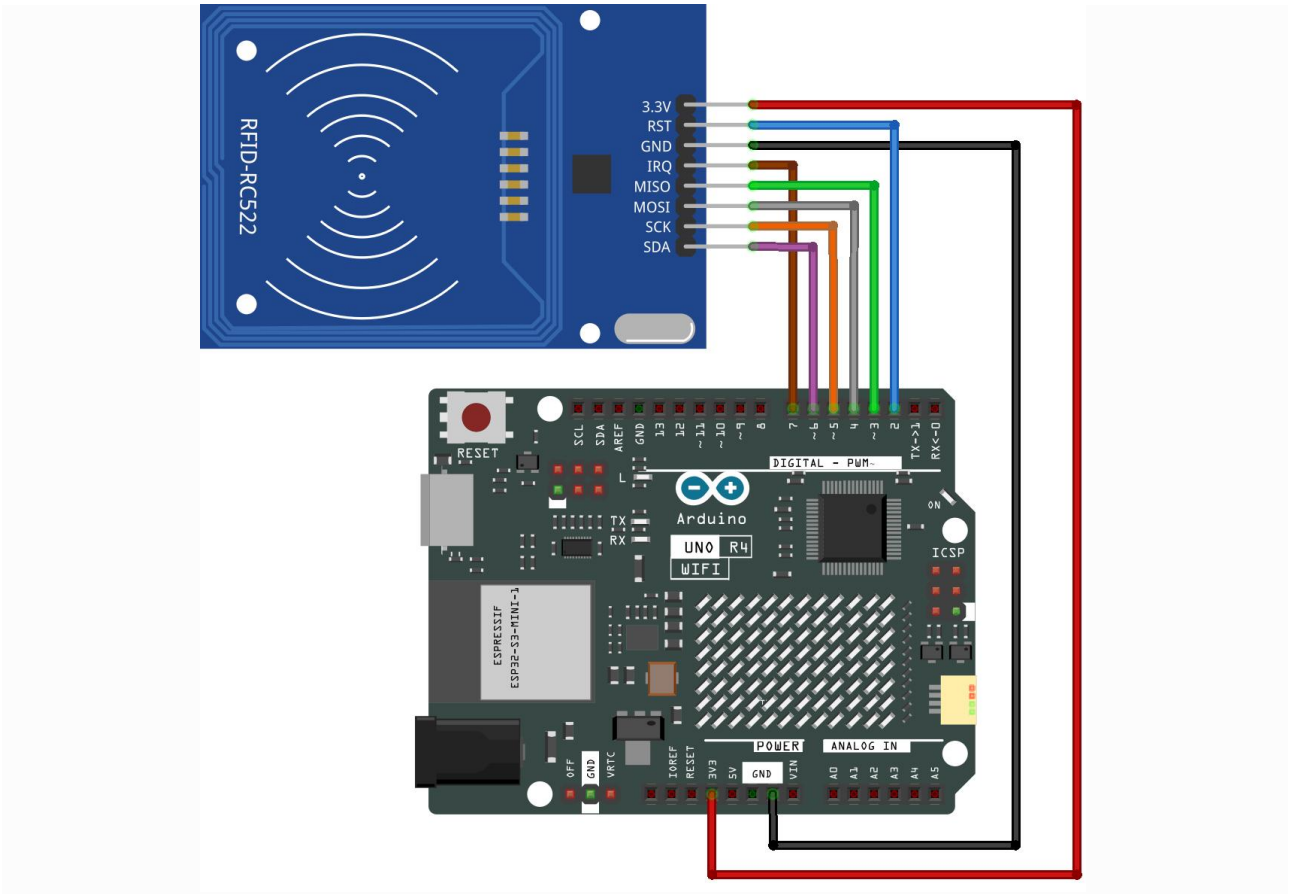
To read the data from the card or tag you've just written, run:

```
sudo python3 2.2.10_read.py
```

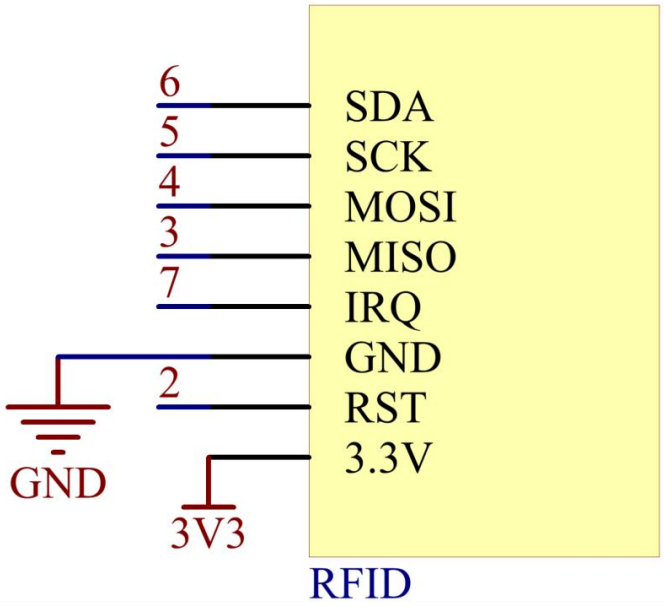
How to Use in Arduino?

Build the Circuit

In this example, we insert the RFID into the breadboard. Get the 3.3V of RFID connected to 3.3V, GND to GND, RST to pin 2, SDA to pin 6, SCK to pin 5, MOSI to pin 4, MISO to pin 3 and IRQ to pin 7.

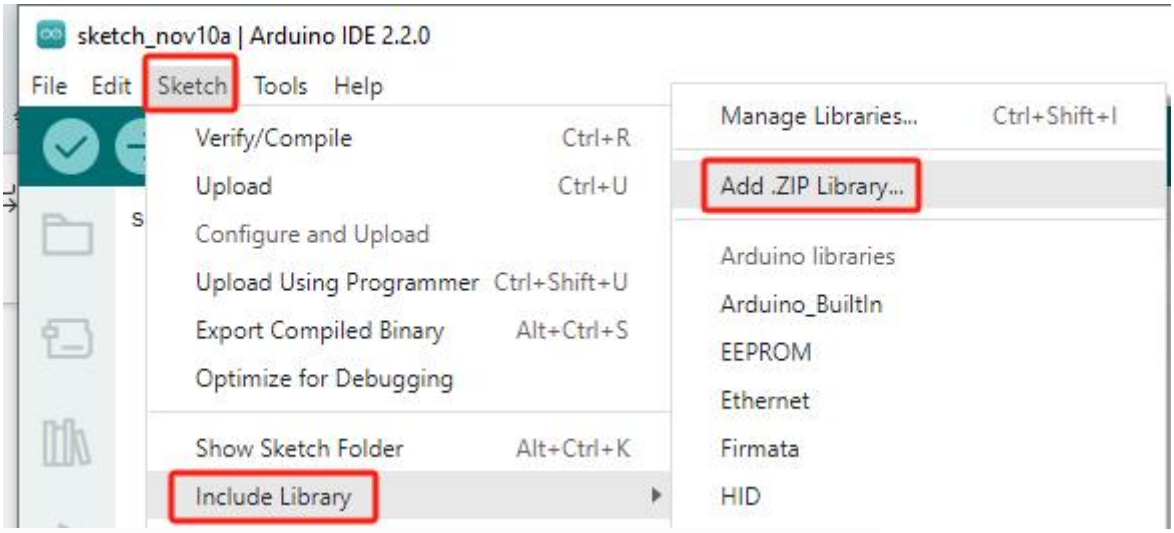


Schematic Diagram

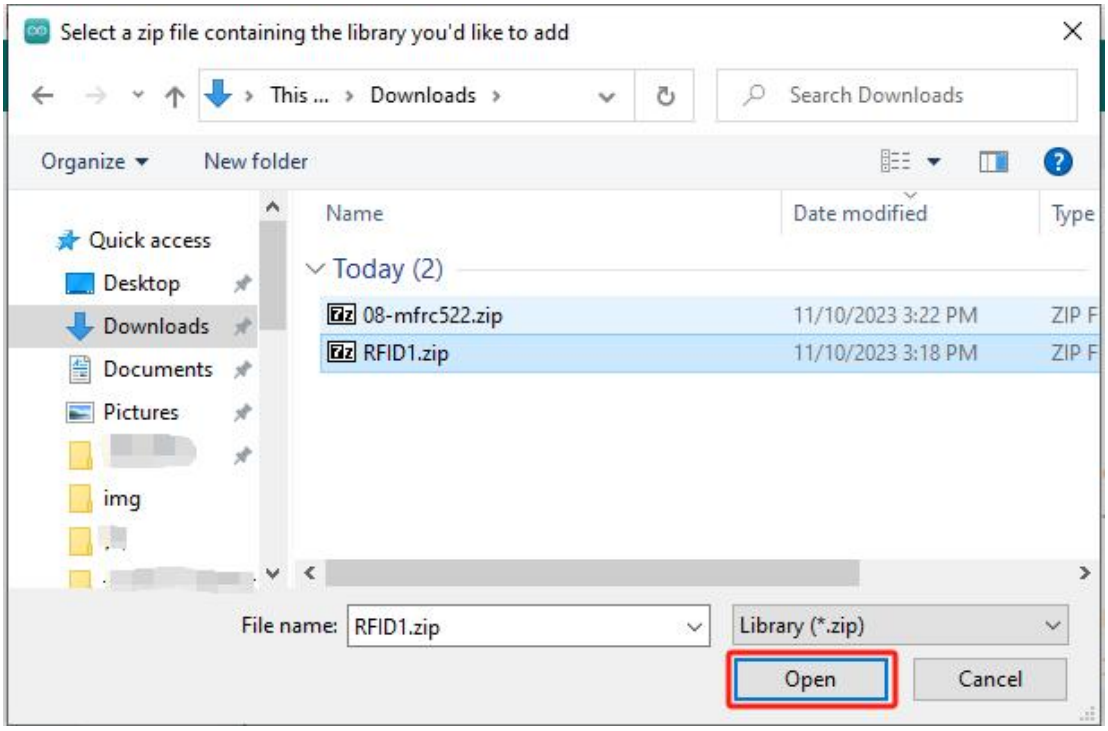


Install the Library

1. The RFID1 library is used here. You can click here: RFID1.zip to download it.
2. Open the Arduino IDE, click **Sketch** -> **Include Library** -> **Add .ZIP** Library.



3. Select the **RFID1.zip** you just downloaded and click **Open**.



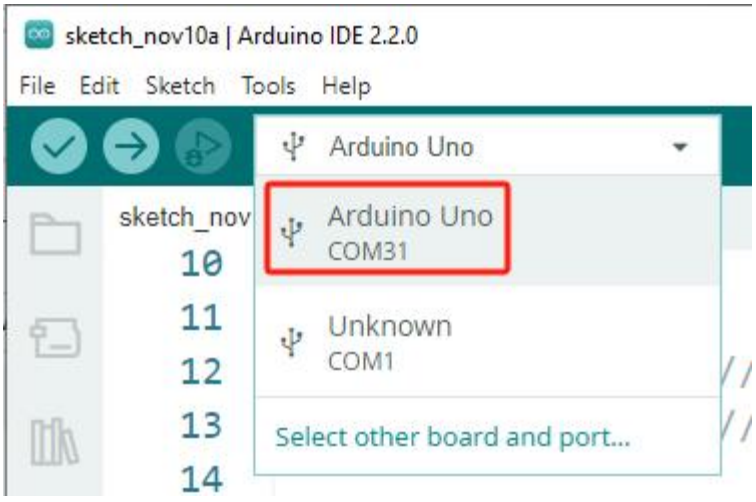
After a while it will indicate that it has been installed.

Run the Code

1. Access the code link below:

- 08-mfrc522

2. Copy the code into the Arduino IDE. Choose the board and port you use.



3. Click the **Upload** button.

4. Now, get your RFID card (secret key) close to the RFID Reader. The module will read the card information and then print it on the serial monitor.

Code Analysis

The functions of the module are included in the library rfid1.h.

```
#include <rfid1.h>
```

Library Functions:

```
RFID1 rfid;
```

Create a new instance of the rfid1 class that represents a particular RFID module attached to your Arduino.

```
void begin(IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN,SDA_PIN,RST_PIN)
```

Pin configuration.

- IRQ_PIN,SCK_PIN,MOSI_PIN,MISO_PIN: the pins used for the SPI communication.
- SDA_PIN: Synchronous data adapter.
- RST_PIN: The pins used for reset.

```
void init()
```

Initialize the RFID.

```
uchar request(uchar reqMode, uchar *TagType);
```

Search card and read card type, and the function will return the current read status of RFID and return MI_OK if successed.

- reqMode: Search methods. PICC_REQIDL is defined that 0x26 command bits (Search the cards that does not in the sleep mode in the antenna area).
- *TagType: It is used to store card type, and its value can be 4byte (e.g. 0x0400).

```
char * readCardType(uchar *TagType)
```

This function decodes the four-digit hexadecimal number of *tagType into the specific card type and returns a string. If passed 0x0400, “MFOne-S50” will be returned.

```
uchar anticoll(uchar *serNum);
```

Prevent conflict, and read the card serial number. The function will return the current reading status of RFID. It returns MI_OK if successed.

- *serNum: It is used to store the card serial number, and return the 4 bytes card serial number. The 5th byte is recheck byte(e.g. e.g. my magnetic card ID is 5AE4C955).