

Security Access using MFRC522 RFID Reader with Arduino

This blog post shows a simple example on how to use the MFRC522 RFID reader. I'll do a quick overview of the specifications and demonstrate a project example using an Arduino.

Description

RFID means radio-frequency identification. RFID uses electromagnetic fields to transfer data over short distances. RFID is useful to identify people, to make transactions, etc...

You can use an RFID system to open a door. For example, only the person with the right information on his card is allowed to enter. An RFID system uses:

- **tags** attached to the object to be identified, in this example we have a keychain and an electromagnetic card. Each tag has his own identification (UID).



- two-way radio transmitter-receiver, the **reader**, that send a signal to the tag and read its response.



Specifications

- Input voltage: 3.3V
- Price: approximately 3\$ ([check best price on Maker Advisor](#))
- Frequency: 13.56MHz

Library download

Here's the library you need for this project:

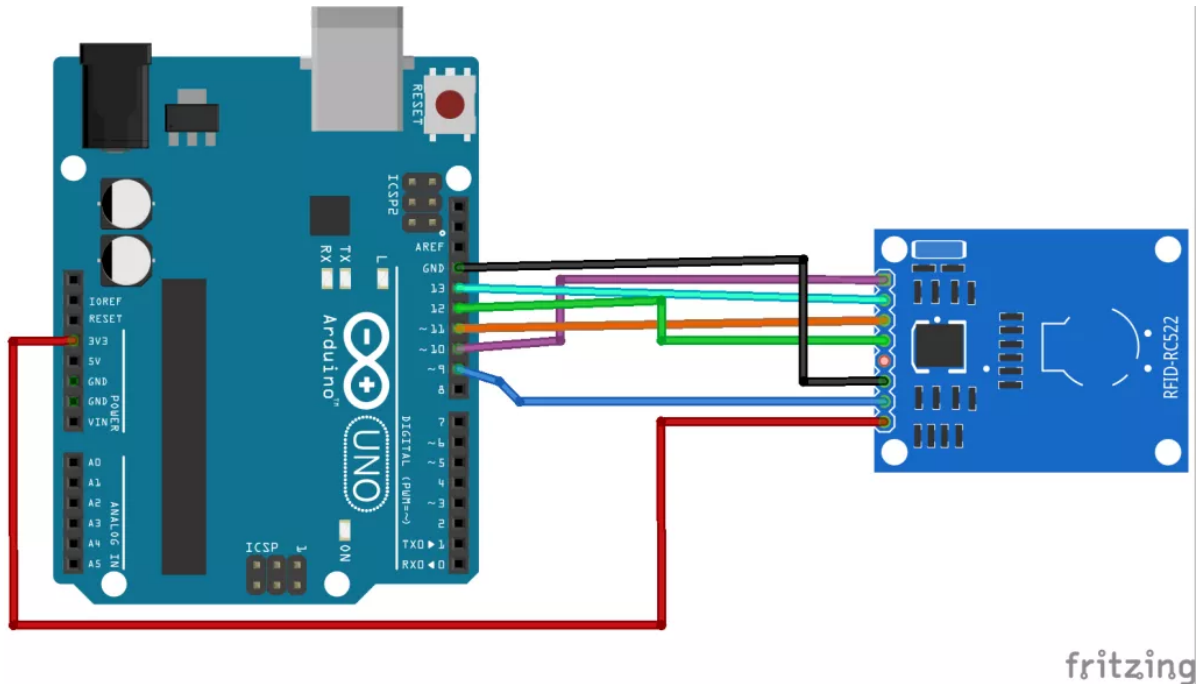
1. Download the [RFID library here](#) created by miguelbalboa
2. Unzip the RFID library
3. Install the RFID library in your Arduino IDE
4. Restart your Arduino IDE

Pin wiring

Pin	Wiring to Arduino Uno
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	unconnected
GND	GND
RST	Digital 9
3.3V	3.3V

Caution: You must power this device to 3.3V!

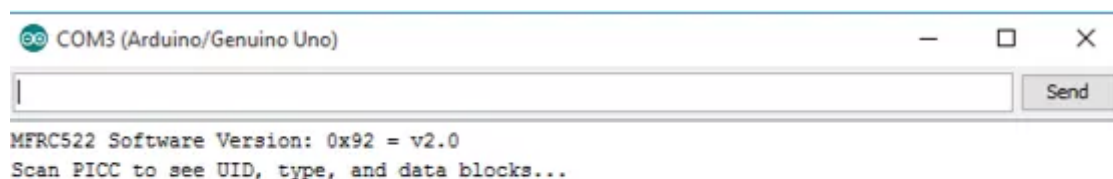
Circuit



Reading Data from a RFID tag

After having the circuit ready, go to File > Examples > MFRC522 > DumpInfo and upload the code. This code will be available in your Arduino IDE (after installing the RFID library).

Then, open the serial monitor. You should see something like the figure below:



Approximate the RFID card or the keychain to the reader. Let the reader and the tag closer until all the information is displayed.

COM3 (Arduino/Genuino Uno)

```

MFRC522 Software Version: 0x92 = v2.0
Scan PICC to see UID, type, and data blocks...
Card UID: BD 31 15 2B
PICC type: MIFARE 1KB
Sector Block  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 AccessBits
  15    63  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        62  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        61  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  14    59  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        58  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        57  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        56  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  13    55  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        54  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        53  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        52  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  12    51  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        49  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        48  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  11    47  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        46  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        45  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        44  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
  10    43  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        42  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        41  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
        40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
   9    39  00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
        38  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]

```

This is the information that you can read from the card, including the card UID that is highlighted in yellow. The information is stored in the memory that is divided into segments and blocks as you can see in the previous picture.

You have 1024 bytes of data storage divided into 16 sectors and each sector is protected by two different keys, A and B.

Write down your UID card because you'll need it later.

Upload the following code.

```

/*
*

```

```
*  
* Created by FILIPEFLOP  
*  
*/  
  
#include <SPI.h>  
#include <MFRC522.h>  
  
#define SS_PIN 10  
#define RST_PIN 9  
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.  
  
void setup()  
{  
  Serial.begin(9600); // Initiate a serial communication  
  SPI.begin();        // Initiate SPI bus  
  mfrc522.PCD_Init(); // Initiate MFRC522  
  Serial.println("Approximate your card to the reader...");  
  Serial.println();  
  
}  
void loop()
```

[View raw code](#)

In the piece of code above you need to change the *if (content.substring(1) == "REPLACE WITH YOUR UID")* and type the UID card you've written previously.

Demonstration

Now, upload the code to your Arduino and open the serial monitor.

Security Access using MFRC522 RFID Reader



Approximate the card you've chosen to give access and you'll see:

```
COM3 (Arduino/Genuino Uno)
|
Approximate your card to the reader...

UID tag : BD 31 15 2B
Message : Authorized access
```

If you approximate another tag with another UID, the denial message will show up:

```
COM3 (Arduino/Genuino Uno)
|
Approximate your card to the reader...

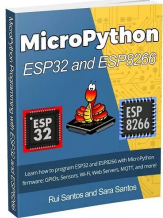
UID tag : 22 4A 9C 0B
Message : Access denied
```

I hope you found this tutorial useful.

You can contact me by leaving a comment. If you like this post probably you might like my next ones, so please support me by [subscribing my blog](#).

Thanks for reading,

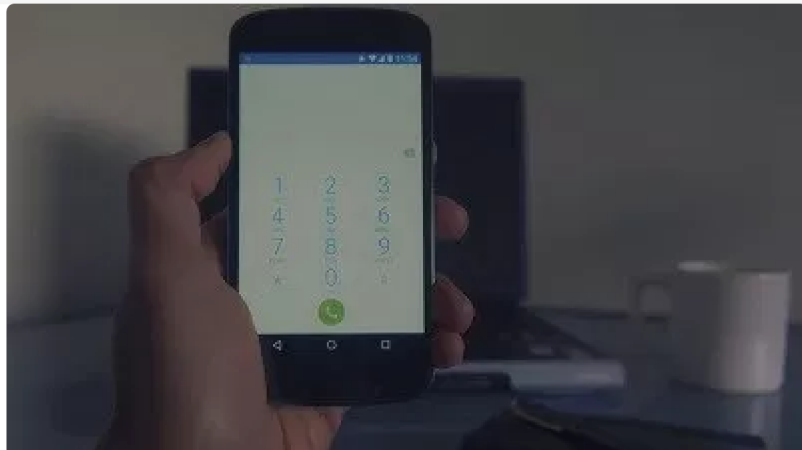
-Rui Santos



[eBook] MicroPython Programming with ESP32 and ESP8266

Learn how to program and build projects with the ESP32 and ESP8266 using MicroPython firmware [DOWNLOAD »](#)

Recommended Resources



[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.

