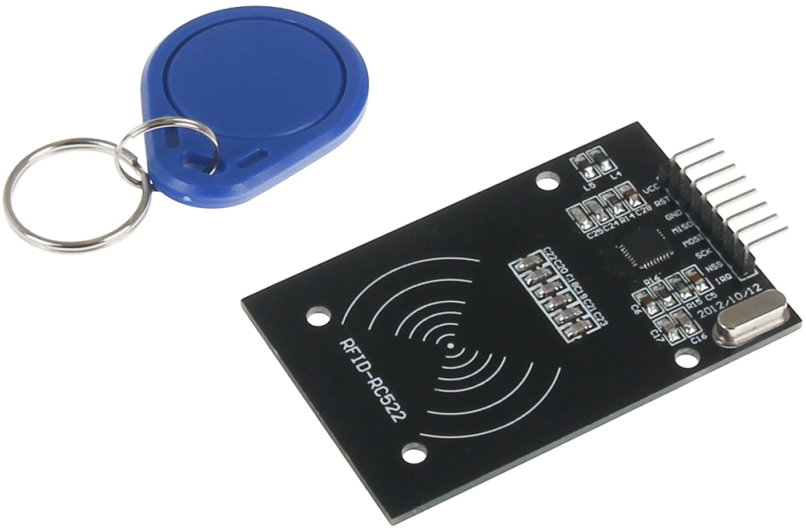
RFID RC522



Index

Dear customer,

Thank you for purchasing our product.
Please find our instructions below.

1. Using with an Arduino

1.1 Connecting the module

Connect the RFID module, as seen in image 1 and table 1, to your PINs of your Arduino.

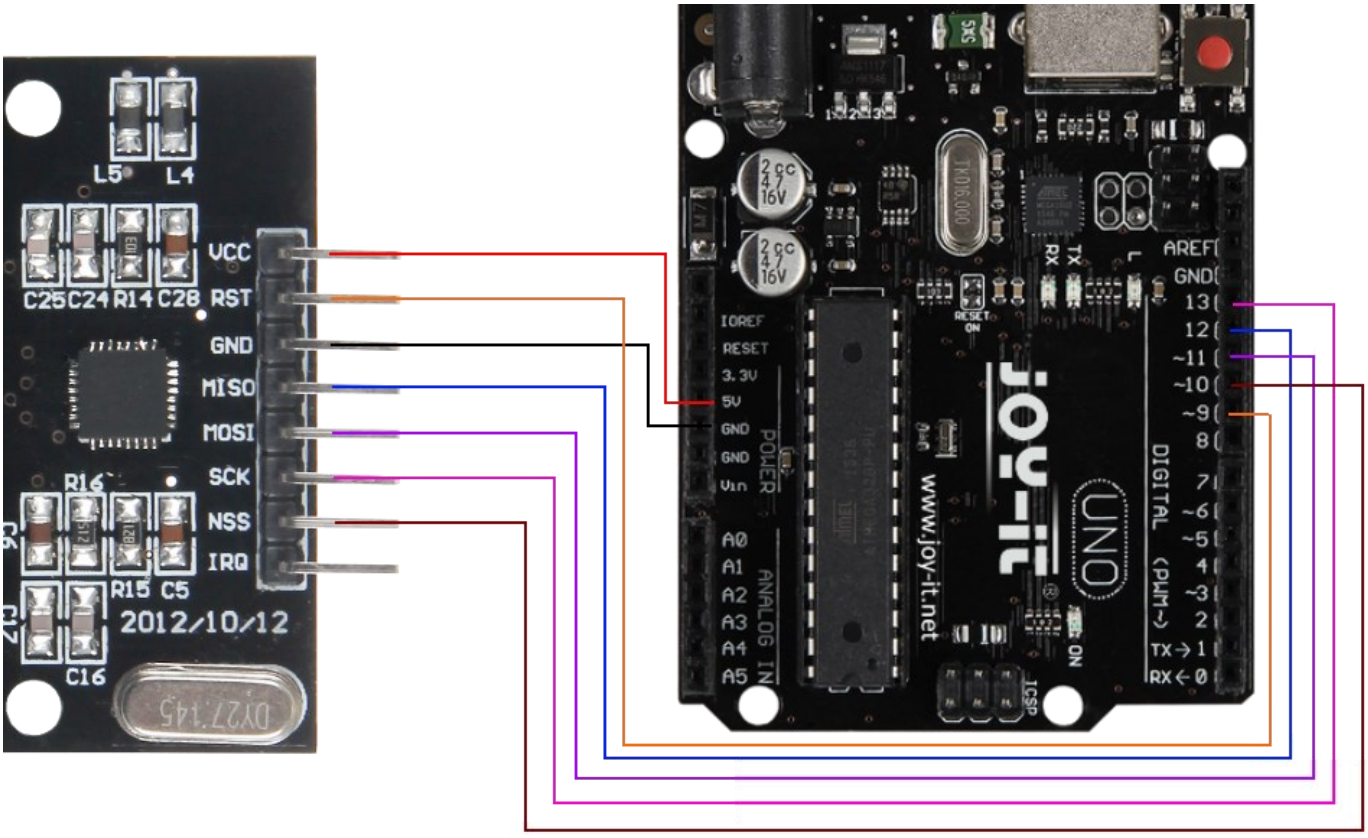


Image 1: Connection between RFID module and Arduino

RFID PIN	Arduino PIN
VCC	5V
RST	9
GND	GND
MISO	12
MOSI	11
SCK	13
NSS	10

Table 1: PIN-Connection between RFID-Module and Arduino

1.2 Installing the module

In the following, you can find a working code example.
In this example, a connection with the RFID device is established and the data is read and printed to the console.

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         9
#define SS_PIN          10

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  //Initialising the RFID-Module
      Serial.begin(9600);
      while (!Serial);
      SPI.begin();
      mfrc522.PCD_Init();
      mfrc522.PCD_DumpVersionToSerial();  //printing details of the reading-device
      Serial.println(F("Scan PICC to see UID, type, and data blocks..."));
}

void loop() {
      //searching for new devices
      if ( ! mfrc522.PICC_IsNewCardPresent()) {
            return;
      }

      if ( ! mfrc522.PICC_ReadCardSerial()) {
            return;
      }

      //data-collection of the RFID device
      mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

Code 1: Example-Code for Arduino

2. Using with a Raspberry Pi

2.1 Connecting the module

Connect the RFID-module, as seen in image 2 and table 2, to the PINs of your Raspberry Pi.

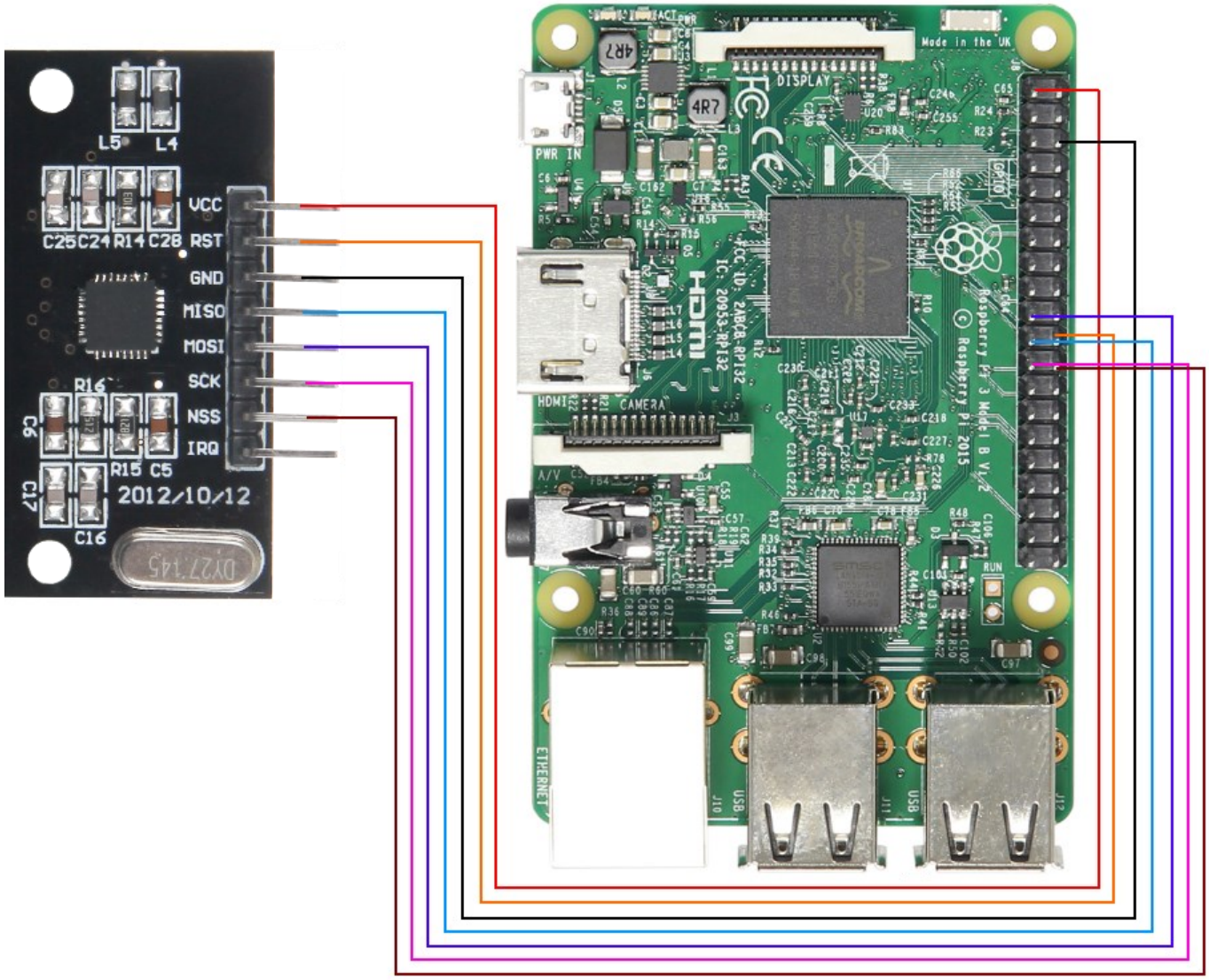


Image 2: Connection between RFID-module and Raspberry Pi

RFID PIN	Raspberry Pi PIN
VCC	PIN 1 (3V Power)
RST	PIN 22 (BCM 25)
GND	PIN 6 (Ground)
MISO	PIN 21 (BCM 9)
MOSI	PIN 19 (BCM 10)
SCK	PIN 23 (BCM 11)
NSS	PIN 24 (BCM 8)
IRC	/

Table 2: PIN-Connection between RFID-module and Raspberry Pi

2.2 Installation des Moduls

If you are already using the latest Raspbian System on your Raspberry Pi, then you can skip this step and continue with step 3.

Install the latest Raspbian image to your SD-Card with the help of the „Win32 Disk Imager“-Tool, which you can find at the following Link.

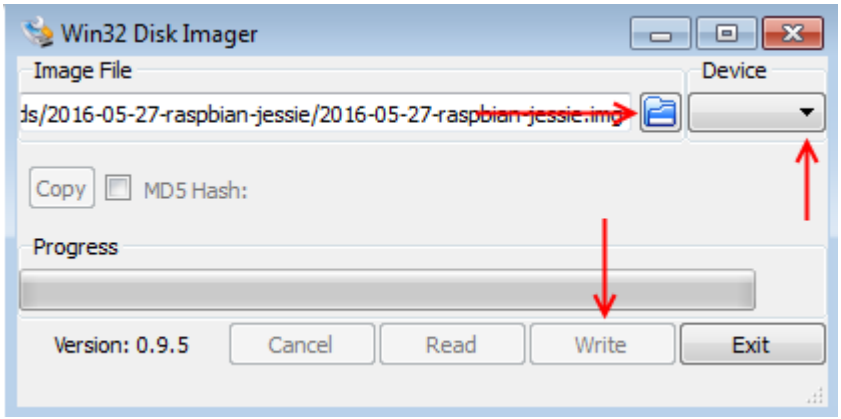


Image 3: Screenshot of the Win32 Disk Imager Tool

2.3 – Installation der Bibliotheken

As soon as your installation is finished, open up your terminal and enter the following commands:

```
sudo apt-get install python-pip python-dev build-essential
sudo pip install RPi.GPIO
```

Terminal 1: Installing the Python-Library

```
sudo nano /boot/config.txt
```

Terminal 2: Editing the Boot-Config

In the opened file, enter the following commands at the very end. You can save the file by pressing **CTRL+O** (confirm with enter) and leave the editor by pressing **CTRL+X**.

```
device_tree_param=spi=on
dtoverlay=spi-bcm2708
```

Code 2: Inserting to Boot-Config

Next, open up the Raspi-Config.

```
sudo raspi-config
```

Terminal 3: Opening the Raspi-Config

At **Advanced Options**, choose **SPI** to activate the SPI module.
You can now leave this menu.
A reboot is required.

```
sudo reboot
```

Terminal 4: Reboot Raspberry

2.4 Installing the Software

To allow you the easiest usage as possible, we are using an extension made by **Louis Thiery & Connor Wolf**.
This extension was published under the terms of the GNU General Public License V2.

At first, we are installing additional packages to access the SPI-Bus and to download additional libraries.

```
sudo apt-get install git python-def --yes
git clone https://github.com/lthiery/SPI-Py.git
cd SPI-Py
sudo python setup.py install
cd ..
git clone https://github.com/mxgxw/MFRC522-python.git && cd MFRC522-python
```

Terminal 4: Installing more packages

The basic SPI installation is now done.
At last, we need to install the MFRC522 library.
Use the following command:

```
git clone https://github.com/mxgxw/MFRC522-python.git && cd MFRC522-python
```

Terminal 5: Installing MFRC522 Library

You can now read RFID devices by using the command:

```
sudo python Read.py
```

Terminal 6: Reading from RFID device

You can also write to your device.
To do this, you need to edit the Write.py file first.

```
sudo nano Write.py
```

Terminal 6: Editing the Write.py file

You need to edit the data variable to what fits your needs.
16 numbers (between 0 and 255) need to be entered into the variable, just like in the following example:

```
# Variable for the data to write
data = [114, 97, 115, 112, 98, 101, 114, 114, 121, 45, 116, 117, 116, 111, 114, 0]
```

Code 3: Inserting to Write.py

If you are done, editing your file, you and execute the writing routine and write to your RFID device.

```
sudo python Write.py
```

Terminal 7: Writing to RFID device