

Interfacing Raspberry Pi with RFID.

Installation Manual

RFID stands for **Radio Frequency Identification** uses radio frequency to read information stored in a RFID card or tag. Each card has a unique ID and this makes it a perfect choice for many authentication applications. The RFID authentication systems are easy to design and are cheap in cost. Interfacing RFID Reader with Raspberry Pi can be very useful as you can implement a wide range of applications like:

- Access Control
- Authentication
- e-Ticket ,e-Payment ,e-Toll
- Attendance System

Hardware Requirements

1. Raspberry Pi Model 3 B/B+
2. RFID Reader (RC 522)
3. RFID Tags or Cards
4. Jumper wires (Female to Male)
5. Breadboard

Here, I am using **RFID Reader RC 522**



This module came with two different styles of header pins, one of which needed to be soldered onto the PCB.

The RFID RC522 is a very low-cost RFID (Radio-frequency identification) reader and writer that is based on the MFRC522 microcontroller. This microcontroller provides its data through the SPI protocol and works by creating a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags.

- **Software Requirements**

1. Raspbian Stretch OS
2. SPI Supporting Libraries
3. RC522 Python Library

Connect your RFID reader with Raspberry Pi's GPIO Pins.

RFID Reader Board Pin	RPI Physical Pin	Raspberry Function
SDA	24	GPIO8 (SPI_CE0_N)
SCK	23	GPIO11 (SPI0_-CLK)
MOSI	19	GPIO10 (SPI0_MOSI)
MISO	21	GPIO9 (SPI0_MISO)
IRQ	UNUSED	
GND	6	GND
RST	22	GPIO25 (GPIO_GEN6)
3.3V	1	3.3V PWR

Step 1: Update Raspberry Pi

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get upgrade
  
```

Step 2: Enable SPI Interface using sudo raspi-config tool

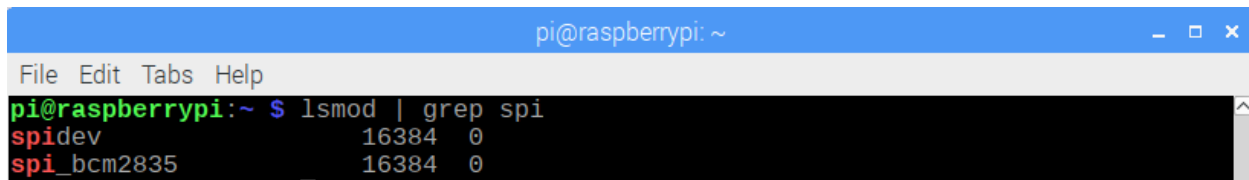
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo raspi-config
  
```

Select using Arrow Keys → Interfacing Options → SPI → Enable

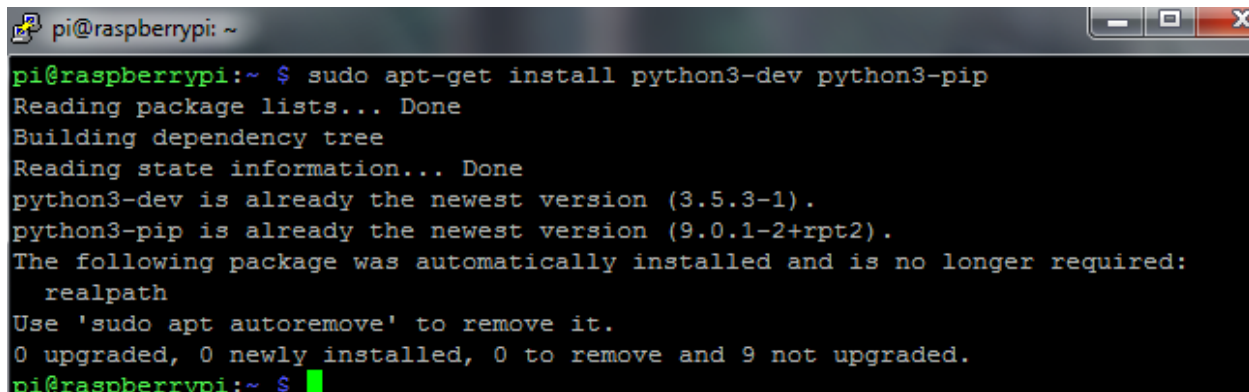
Step 3: Reboot Raspberry Pi

Step 4: Check to make sure that SPI has been enabled.

A terminal window titled 'pi@raspberrypi: ~' with a menu bar (File, Edit, Tabs, Help). The command 'lsmod | grep spi' is entered, resulting in two lines of output: 'spidev 16384 0' and 'spi_bcm2835 16384 0'.

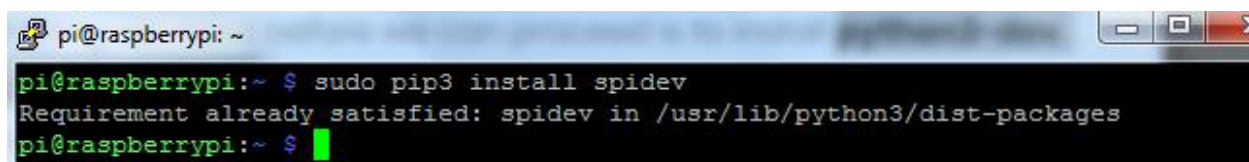
```
pi@raspberrypi:~ $ lsmod | grep spi
spidev 16384 0
spi_bcm2835 16384 0
```

Step 5: Install python3-dev, python3-pip packages to setting up RFID reader

A terminal window titled 'pi@raspberrypi: ~' showing the command 'sudo apt-get install python3-dev python3-pip'. The output indicates that both packages are already the newest versions (3.5.3-1 and 9.0.1-2+rpt2). It also notes that 'realpath' was automatically installed and is no longer required. The command '0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.' is shown at the end.

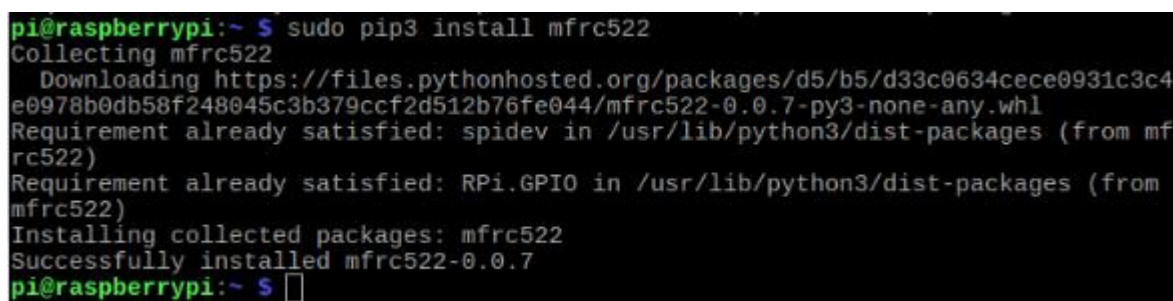
```
pi@raspberrypi:~ $ sudo apt-get install python3-dev python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-dev is already the newest version (3.5.3-1).
python3-pip is already the newest version (9.0.1-2+rpt2).
The following package was automatically installed and is no longer required:
  realpath
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
pi@raspberrypi:~ $
```

Step 6: Install spidev to Raspberry Pi using pip. The spidev library helps to handle interactions with the SPI

A terminal window titled 'pi@raspberrypi: ~' showing the command 'sudo pip3 install spidev'. The output states 'Requirement already satisfied: spidev in /usr/lib/python3/dist-packages'.

```
pi@raspberrypi:~ $ sudo pip3 install spidev
Requirement already satisfied: spidev in /usr/lib/python3/dist-packages
pi@raspberrypi:~ $
```

Step 7: Install the MFRC522 library using pip that helps talk to the RC522 module over the SPI interface

A terminal window titled 'pi@raspberrypi:~' showing the command 'sudo pip3 install mfrc522'. The output shows the collection of the package, downloading from a URL, and confirming that dependencies 'spidev' and 'RPi.GPIO' are already satisfied. The package is then successfully installed.

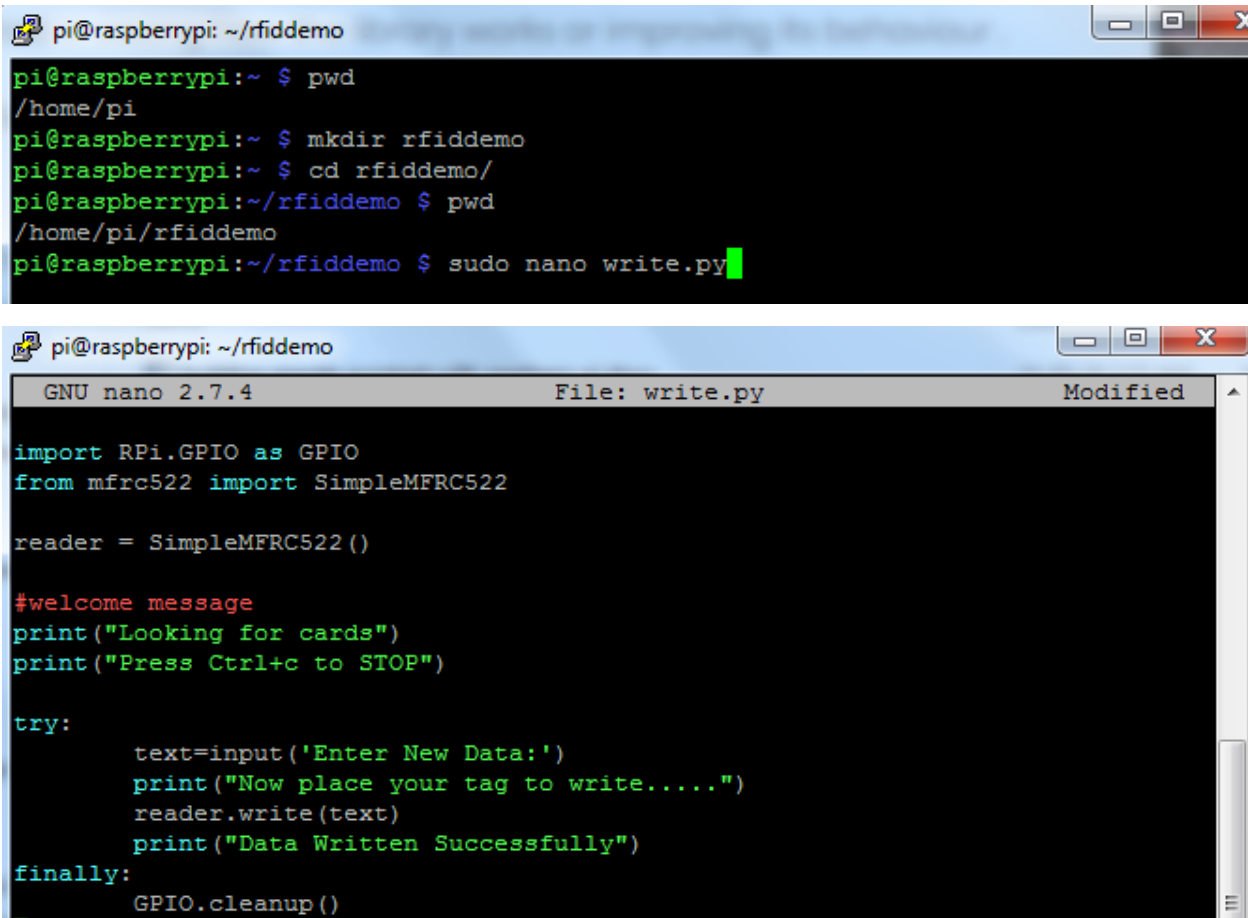
```
pi@raspberrypi:~ $ sudo pip3 install mfrc522
Collecting mfrc522
  Downloading https://files.pythonhosted.org/packages/d5/b5/d33c0634cece0931c3c4e0978b0db58f248045c3b379ccf2d512b76fe044/mfrc522-0.0.7-py3-none-any.whl
Requirement already satisfied: spidev in /usr/lib/python3/dist-packages (from mfrc522)
Requirement already satisfied: RPi.GPIO in /usr/lib/python3/dist-packages (from mfrc522)
Installing collected packages: mfrc522
Successfully installed mfrc522-0.0.7
pi@raspberrypi:~ $
```

There are two files included in this repository:

MFRC522.py which is an implementation of the RFID RC522 circuit.

SimpleMFRC522.py that takes the MFRC522.py file and greatly simplifies it.

Step 8: Write Python script which is used to write data from the RC522 to your RFID tags.



The first screenshot shows a terminal window on a Raspberry Pi with the command prompt `pi@raspberrypi: ~/rfiddemo`. The user enters the following commands: `pwd` (returns `/home/pi`), `mkdir rfiddemo`, `cd rfiddemo/`, `pwd` (returns `/home/pi/rfiddemo`), and `sudo nano write.py`. The second screenshot shows the `nano` text editor editing `write.py`. The code in the file is as follows:

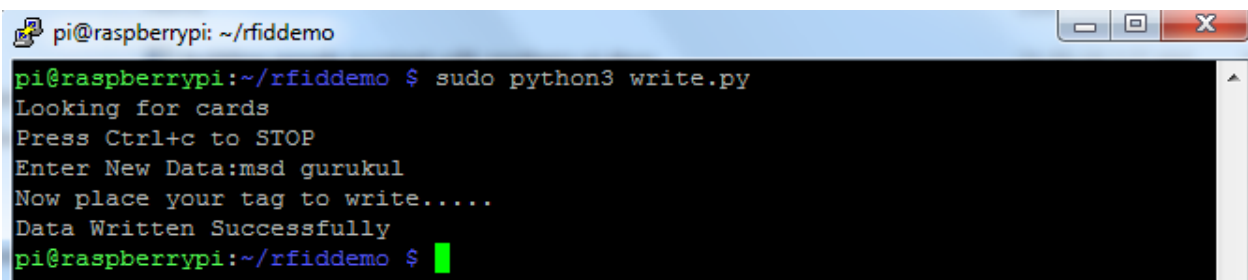
```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

#welcome message
print("Looking for cards")
print("Press Ctrl+c to STOP")

try:
    text=input('Enter New Data:')
    print("Now place your tag to write.....")
    reader.write(text)
    print("Data Written Successfully")
finally:
    GPIO.cleanup()
```

Run above script



The screenshot shows the terminal window with the command `sudo python3 write.py` executed. The output of the script is as follows:

```
Looking for cards
Press Ctrl+c to STOP
Enter New Data:msd gurukul
Now place your tag to write.....
Data Written Successfully
pi@raspberrypi:~/rfiddemo $
```

When you run script, it asked to write in the new data, in my case I am going to just type in msd gurukul. Press Enter when you are happy with what you have written.

With that done, simply place your RFID Tag on top of your RFID RC522 circuit. As soon as it detects it, it will immediately write the new data to the tag. You should see “Data Written successfully” appear in your command line if it was successful.

Step 8: Write Python script which is used to read this data back off the RFID tag.

```
pi@raspberrypi: ~/rfiddemo
pi@raspberrypi:~/rfiddemo $ pwd
/home/pi/rfiddemo
pi@raspberrypi:~/rfiddemo $ sudo nano read.py
```

```
GNU nano 2.7.4 File: read.py

import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

#welcome message
print("Looking for cards")
print("Press Ctrl+c to STOP")

try:
    id, text = reader.read()
    print(id)
    print(text)
finally:
    GPIO.cleanup()
```

Run above script

```
pi@raspberrypi: ~/rfiddemo
pi@raspberrypi:~/rfiddemo $ sudo python3 read.py
Looking for cards
Press Ctrl+c to STOP
32432970100
msd gurukul
pi@raspberrypi:~/rfiddemo $
```

With the script now running, all you need to do is place your RFID Tag on top of your RFID RC522 circuit. As soon as the Python script detects the RFID tag being placed on top, it will immediately read the data and print it back out to you.

If you successfully receive data back from your **read.py** script with the text that you pushed to the card using your **write.py** script then you have successfully set up your Raspberry Pi to connect with your RFID RC522 Circuit.

That's all !!!

Thank you....