Check Point RFID Authentication with R81

Heiko Ankenbrand

Version 1.0

03/02/2021

Install Raspberry

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

More read here:
https://www.raspberrypi.org/software/

Configure the network basics.

Install Apache

1. First, update the available packages by typing the following command into the Terminal:

$ sudo apt update

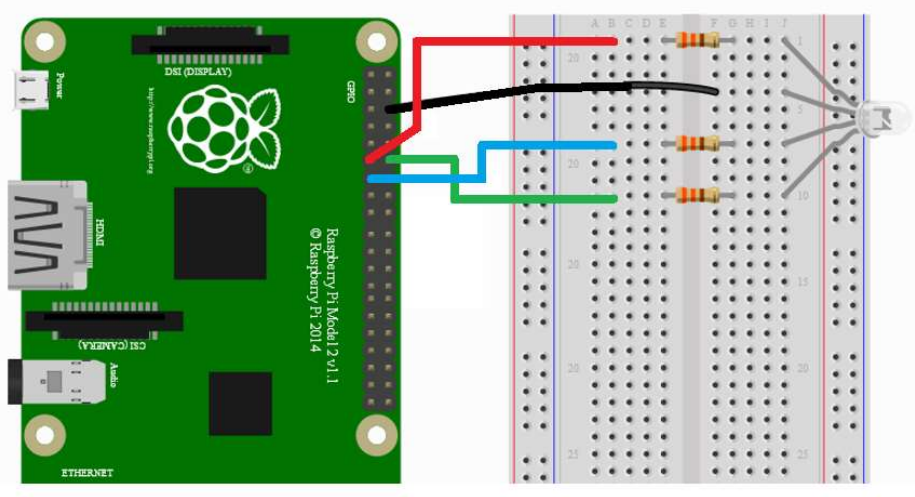
2. Then, install the apache2 package with this command:

$ sudo apt install apache2 -y

Assembling the RGB LED

1. Connect the common cathode (longest leg) to a ground pin; connect each of the other legs (representing the red, green, and blue anodes) to any GPIO pins. You should use three limiting resistors 330 ohm (one per anode).

- **R** connects to **Pin 17**
- **GND** connects to **Pin 6**
- **G** connects to **Pin 18**
- **B** connects to **Pin 27**



Assembling the RFID RC522

One thing you will notice when purchasing an RFID RC522 Reader is that 90% of them don’t come with the header pins already soldered in. The missing pins mean you will have to do it yourself, luckily soldering header pins is a rather simple task, even for beginners.

1. First off, if the header pins you received with your RC522 isn’t the correct size, then snap them down, so you only have a single row of eight pins.

2. Place the header pins up through the holes of your RC522. One handy trick is to put the long side of the header pins into a breadboard and then putting the circuit over the top of the header pins. The breadboard will hold the pins tightly making it easier to solder them to the RFID RC522 circuit.

3. Now using a hot soldering iron and some solder, slowly solder each of the pins. Remember it is best to heat the joint slightly before applying solder to it, this will ensure that the solder will adhere more to the joint and reduce the chances of creating a cold joint. We also recommend being careful with the amount of solder you apply.

4. With the header pins now soldered to your RFID circuit, it is now ready to use, and you can continue with the tutorial.

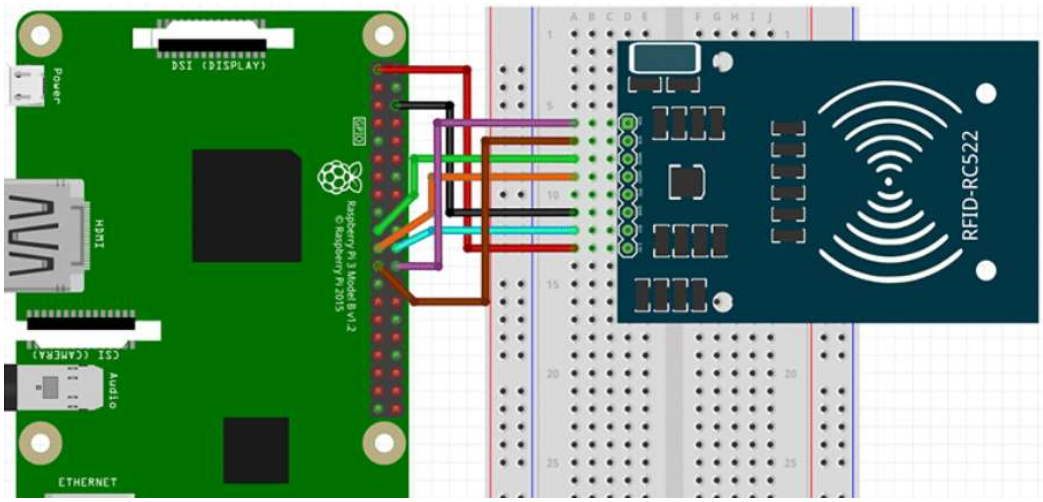
Wiring the RFID RC522

On your RFID RC522 you will notice that there are 8 possible connections on it, these being **SDA** (Serial Data Signal), **SCK** (Serial Clock), **MOSI** (Master Out Slave In), **MISO** (Master In Slave Out), **IRQ** (Interrupt Request), **GND** (Ground Power), **RST** (Reset-Circuit) and **3.3v** (3.3v Power In). We will need to wire all of these but the **IRQ** to our Raspberry Pi's GPIO pins.

You can either wire these directly to the GPIO Pins or like we did in this tutorial, plug the RFID RC522 into our Breadboard then wire from there to our Raspberry Pi's GPIO Pins.

Wiring your RFID RC522 to your Raspberry Pi is fairly simple, with it requiring you to connect just 7 of the GPIO Pins directly to the RFID reader. Follow the table below, and check out our GPIO guide to see the positions of the GPIO pins that you need to connect your RC522 to.

- **SDA** connects to **Pin 24**
- **SCK** connects to **Pin 23**
- **MOSI** connects to **Pin 19**
- **MISO** connects to **Pin 21**
- **GND** connects to **Pin 6**
- **RST** connects to **Pin 22**
- **3.3v** connects to **Pin 1**



Setting up Raspberry for the RFID RC522

Let's begin by first opening the **raspi-config** tool, and we can do this by opening the terminal and running the following command.

```
$ sudo raspi-config
```

This tool will load up a screen showing a variety of different options.

On here use the **arrow keys** to select "**5 Interfacing Options**". Once you have this option selected, press **Enter**.

3. Now on this next screen, you want to use your **arrow keys** to select "**P4 SPI**", again press Enter to select the option once it is highlighted.

4. You will now be asked if you want to enable the SPI Interface, select **Yes** with your **arrow keys** and press **Enter** to proceed. You will need to wait a little bit while the **raspi-**config tool does its thing in enabling SPI.

5. Once the SPI interface has been successfully enabled by the **raspi-config** tool you should see the following text appear on the screen, "**The SPI interface is enabled**".

Before the SPI Interface is fully enabled we will first have to restart the Raspberry Pi. To do this first get back to the terminal by pressing **Enter** and then **ESC**.

```
$ sudo reboot
```

6. Once your Raspberry Pi has finished rebooting, we can now check to make sure that it has in fact been enabled. The easiest way to do this is to run the following command to see if **spi_bcm2835** is listed.

```
$ lsmod | grep spi
```

If you see **spi_bcm2835**, then you can proceed on with this tutorial and skip on to the next section. If for some reason it had not appeared when you entered the previous command, try following the next three steps.

7. If for some reason the SPI module has not activated, we can edit the boot configuration file manually by running the following command on our Raspberry Pi.

Getting Python ready for the RFID RC522

1. Before we start programming, we first need to update our Raspberry Pi to ensure it's running the latest version of all the software. Run the following two commands on your Raspberry Pi to update it.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2. Now the final thing we need before we can proceed is to install **python3-dev**, **python-pip** and **git** packages. Simply run the following command on your Raspberry Pi to install all of the required packages for this guide on setting up your RFID reader.

```
$ sudo apt-get install python3-dev python3-pip
```

3. To begin, we must first install the Python Library spidev to our Raspberry Pi using the python "**pip**" tool that we downloaded in the previous step.

The spidev library helps handle interactions with the SPI and is a key component to this tutorial as we need it for the Raspberry Pi to interact with the RFID RC522.

```
$ sudo pip3 install spidev
```

4. Now that we have installed the spidev library to our Raspberry Pi we can now now proceed to installing the MFRC522 library using pip as well.

```
$ sudo pip3 install mfrc522
```

Use the RFID RC522

1. Now lets start off by making a folder where we will be storing our couple of scripts.

We will be calling this folder “rfid”, create it by running the following command.

$ mkdir ~/rfid

2. Begin by changing directory into our newly cloned folder.

$ cd ~/rfid

3. Begin writing our **CP.py** Python script with vi. Write the following lines of code from the box between „CUT>>> … <<<CUT“.

$ vi CP.py

CUT>>>

```
import sys
import RPi.GPIO as GPIO
from time import sleep

strings = [[1037879776439,"10.10.52.181",0,"Test User A"],[554357207136,"192.168.204.91",0,"Test
User B"]]

GPIO.setmode(GPIO.BCM)
led=18
led1=17
GPIO.setup(led, GPIO.OUT)
GPIO.setup(led1, GPIO.OUT)
GPIO.output(led, GPIO.LOW)
GPIO.output(led1, GPIO.LOW)
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()

try:
    while True:
        il=999999
        print("-------------------------")
        id, text = reader.read()
        print("ID: %s" % (id))
        for i in range(len(strings)):
            if id == strings[i][0]:
                print("IP: %s " % (strings[i][1]))
                print("US: %s " % (strings[i][3]))
                il = i
                if strings[i][2] == 0:
                    strings[i][2] = 1
                    GPIO.output(led, GPIO.HIGH)
                    print ("ST: Allow IP")
                else:
                    strings[i][2] = 0
```

```
            GPIO.output(led1, GPIO.HIGH)
            print ("ST: Deny IP")
        if il == 999999:
          GPIO.output(led1, GPIO.HIGH) # rot
          print ("ST: no allowed user")
        else:
          f = open("/var/www/html/checkpoint.txt", "w")
          f.write('{\n')
          f.write('      "version": "1.0",\n')
          f.write('      "description": "Generic Data Center file RFID_IP",\n')
          f.write('      "objects": [\n')
          f.write('                        {\n')
          f.write('                            "name": "RFID_IP",\n')
          f.write('                            "id": "e7f18b60-f22d-4f42-8dc2-050490ecff00",\n')
          f.write('                            "description": "Example for IPv4 addresses",\n')
          f.write('                            "ranges": [\n')
          for i in range(len(strings)):
              if 1 == strings[i][2]:
                  f.write(' "')
                  f.write(strings[i][1])
                  f.write('",\n')
          f.write('                            ]\n')
          f.write('                        }\n')
          f.write('     ]\n')
          f.write('}\n')
          f.close()
        sleep(2)
        GPIO.output(led, GPIO.LOW)
        GPIO.output(led1, GPIO.LOW)

except KeyboardInterrupt:
    GPIO.cleanup()
    raise
```

<<<CUT

4. Now that we have written our script, we will want to test it out. Before testing out the script make sure that you have changed the IP addresses and the RFID tag (red marked) in the script. Once ready, type the following command into your Raspberry Pi's terminal.

$ sudo python3 CP.py

5. With that done, simply place your RFID Tag on top of your RFID RC522 circuit.

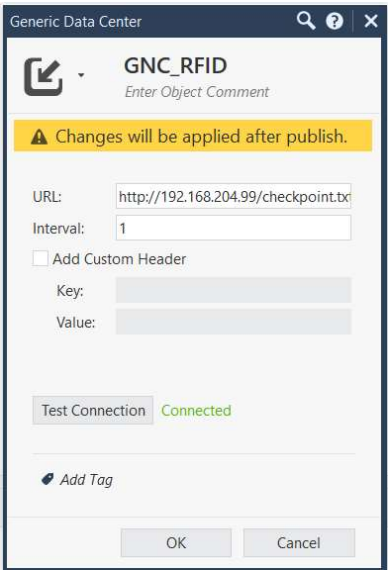
Create a Generic Data Center Object

The Generic Data Center feature provides the ability to enforce access to/from IP addresses defined in JSON files located on Raspberry Pi web servers on the Security Management machine. The „Generic Data Center Objects“ are updated automatically on the Security Gateway each time the JSON file change. There is no need to install policy for the updates to take effect.

1. Create a Generic Data Center Object

Use the IP address from the Raspberry Pi web servers:

URL: http://< Raspberry Pi IP>/checkpoint.txt
Interval: 1 sec



2. Create a rule with the Generic Data Center Object „RFID_IP“

Source	Destination	VPN	Services & Applications	Action	Track
RFID_IP	Any	Any	https http dns	Accept	Log

3. Install the policy.

Now it is done😊!