**INSTALLATION OF RFID CARD SYSTEM IN OUR COLLEGE**

**Introduction:**

* This project aims to use single card for many purposes related to academic, mess, library with some advanced technology to speed up daily tasks related to ID card.
* Most of the time it’s not possible to carry all the card with us so we face many problems in our daily lives.
* This problem can be solved by using RFID card which have a unique id through which we can fetch data from the server and all the necessary details can be obtained digitally.
* By combining all the cards into one by using RFID technology we can not only rid ourselves of maintaining many cards but also we can digitize our daily works.

**Motivation:**

* We have observed that our wallets get filled with many cards such as college id card, mess card, hall card, library card,club members card and many more personal card like atm and others.
* During exam time to enter into the library we have to go through a very large queue to enter the log details which is time consuming and irritating during the exams.
* Not only library each and every place where we have to maintain the log details we have to go through a very lengthy process and also the maintenance of these data is quite hectic.

**Project details :**

The above problems can be solved by having a single RFID which can be programmed to contain the unique number which corresponds to the data stored in the database of that student or faculty.

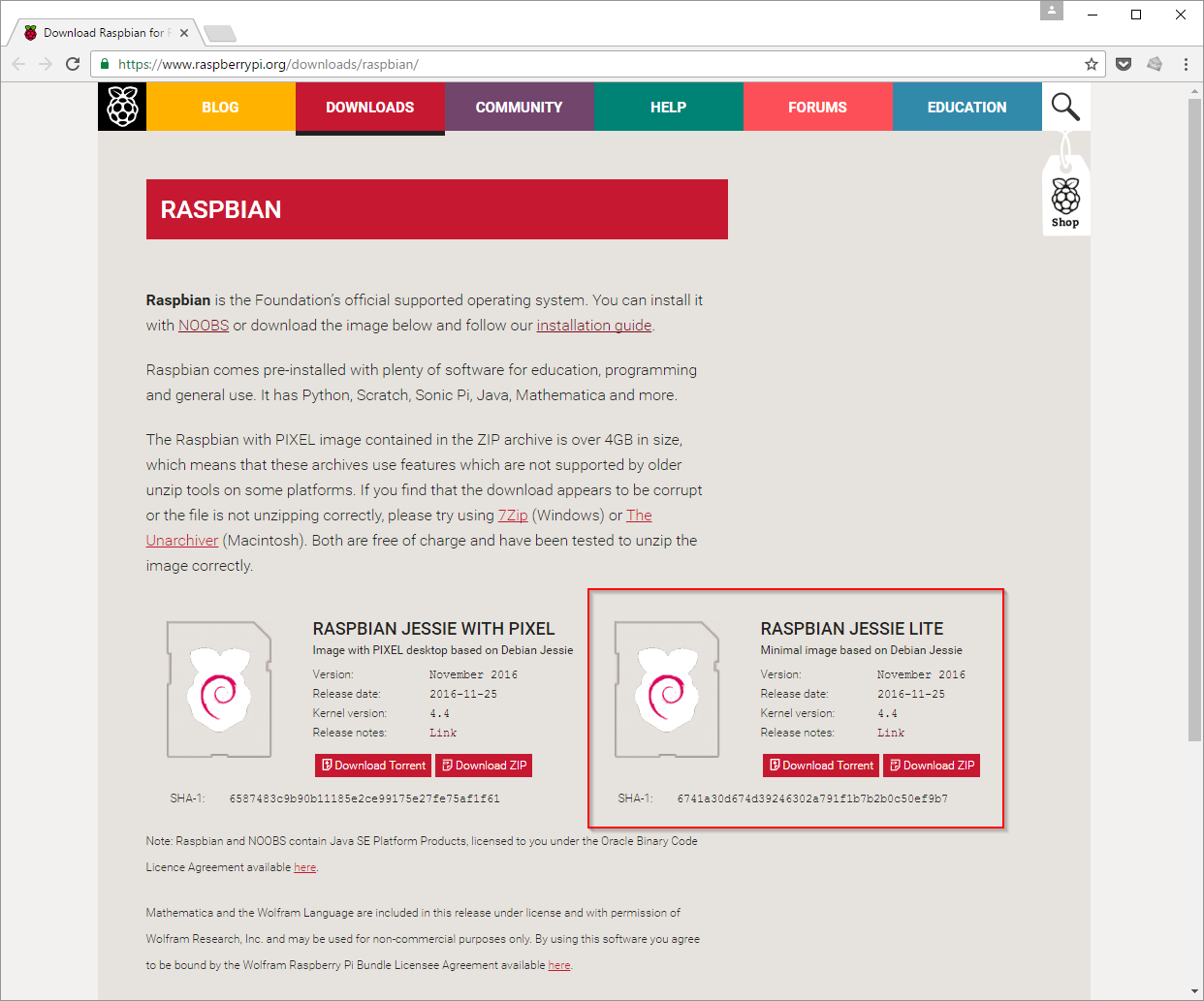
The RFID reader will fetch the data stored in the card and request the server and server will send back the relevant data back.

**STEPS TO INSTALL THE RASPBIAN OS:**

**Headless Raspberry Pi Setup**

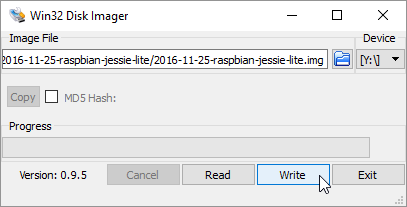
**Step 1. Downloading Raspbian Image**

We head over [here](https://www.raspberrypi.org/downloads/raspbian/) to download a copy of the Raspbian image. The “Lite” version will do.



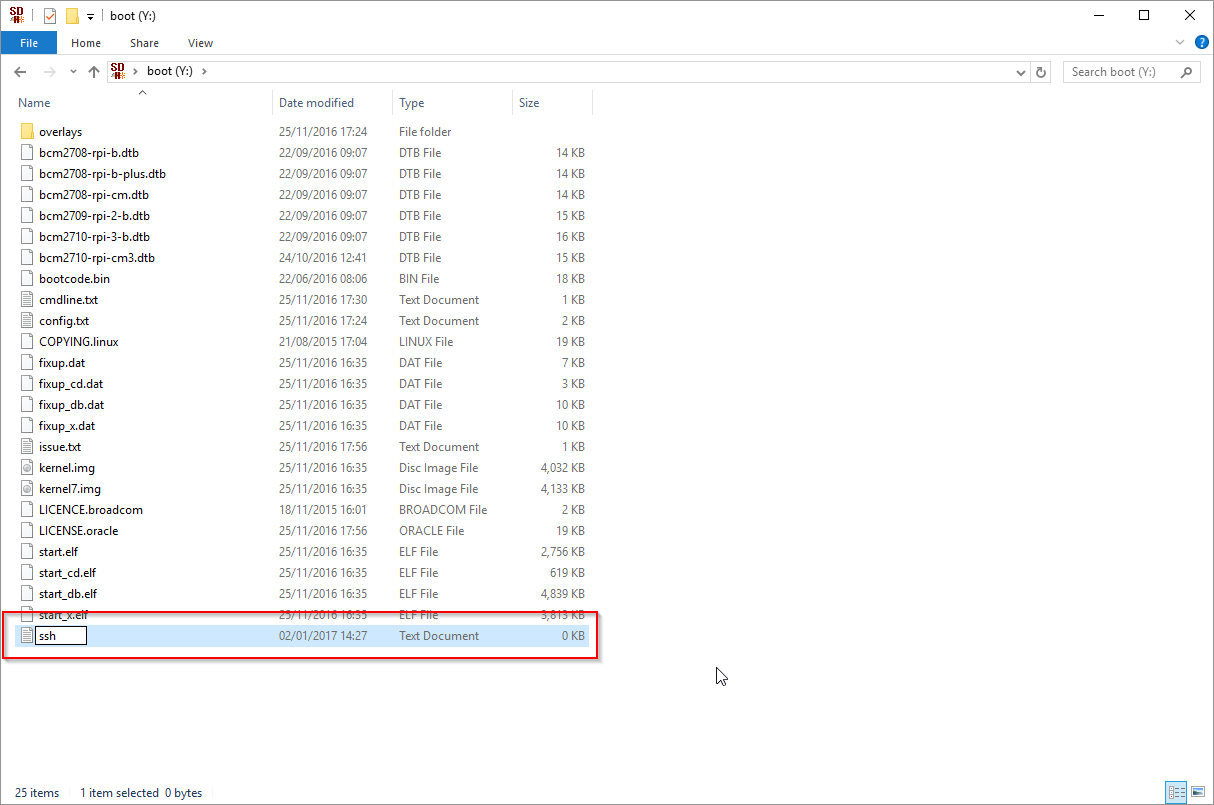
**Step 2. Writing Image to SD Card**

Then we write the image to SD card. The detailed instructions can be found [here](https://www.raspberrypi.org/documentation/installation/installing-images/README.md).



**Step 3. Adding “SSH” File to the SD Card Root**

Then we enable [SSH](https://en.wikipedia.org/wiki/Secure_Shell) by placing a file named “ssh” (without any extension) onto the boot partition of the SD card:

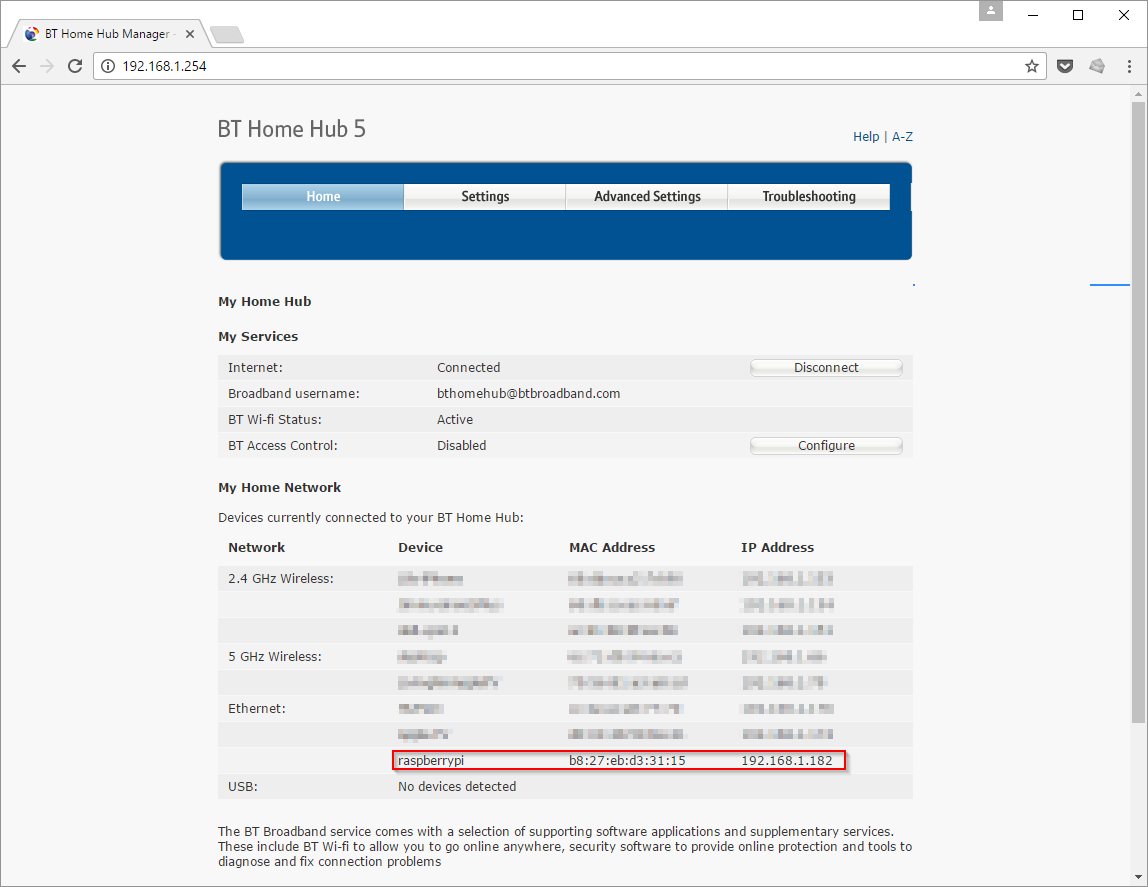


**Step 4. Booting our Pi**

We insert our prepared SD card, power and a network cable into the Pi.

**Step 5. Finding our Pi’s IP Address**

To configure our Pi, we need the IP address. We can find this in your Router’s DHCP lease allocation table:

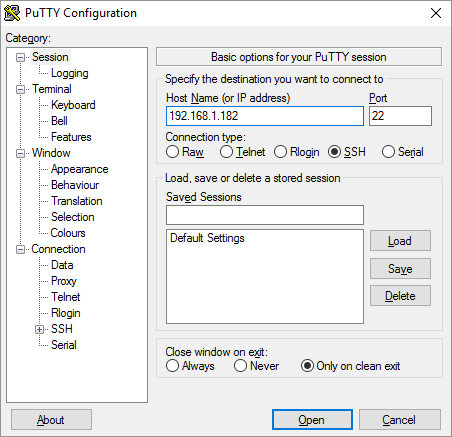


**Step 6. SSH into our Pi**

We use [PuTTY](http://www.putty.org/) (SSH client) to access the Pi. The default credentials are:

username: pi

password: raspberry



**Step 7. Configuring our Pi**

We can now configure our Pi via sudo raspi-config

**STEPS TO PROGRAM THE RFID READER:**

Wiring the RFID RC522

On our RFID RC522 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.

The following table shows our connections:

**SDA** connects to **Pin 24**.

**SCK** connects to **Pin 23**.

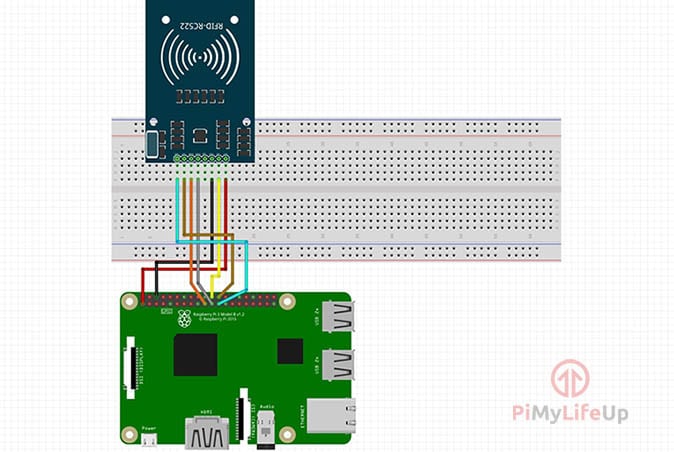
**MOSI** connects to **Pin 19**.

**MISO** connects to **Pin 21**.

**GND** connects to **Pin 6**.

**RST** connects to **Pin 22**.

**3.3v** connects to **Pin 1**.





Setting up Raspbian for the RFID RC522

Before we begin the process of utilizing the RFID RC522 on our Raspberry Pi we will first have to make changes to its configuration. By default, the Raspberry Pi has the SPI (Serial Peripheral Interface) disabled, which is a bit of a problem as that is what our RFID reader circuit runs through. Following the steps below we can configure our Raspberry Pi and Raspbian to utilize the SPI interface.

**1.** First we have to open the **raspi-config** tool, we can do this by opening terminal and running the following command:

sudo raspi-config

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

On here we select “**5 Interfacing Options**“.

**3.** Now on this next screen we select “**P4 SPI**“ and press Enter to select the option once it is highlighted.

**4.** Select Yes to enable the SPI Interface.

**5.** Once the SPI interface has been successfully enabled by the **raspi-config** tool we 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 we go back to the terminal by pressing **Enter** and then **ESC**.

Typing the following linux command into the terminal on your Raspberry Pi to restart your Raspberry Pi.

sudo reboot

**6.** Once our 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 we see **spi\_bcm2835** then we can proceed.

Getting Python ready for the RFID RC522

Now that we have wired up our RFID RC522 circuit to the Raspberry Pi we can now power it on and begin the process of programming simple scripts in Python to interact with the chip.

.

**1.** We first need to update our Pi

sudo apt-get update  
sudo apt-get upgrade

**2.** We need to install python on our Pi.

sudo apt-get install python2.7-dev

**3.** To begin we must first clone the Python Library SPI Py and install it to our Raspberry Pi. This library helps handle interactions with the SPI and we need it for the Raspberry Pi to interact with the RFID RC522.

We run the following two commands on your Raspberry Pi to clone the source code.

cd ~  
git clone https://github.com/lthiery/SPI-Py.git

**4.** With the SPI Py Python Library now cloned to our Raspberry Pi we need to install it.This can be done by:

cd ~/SPI-Py  
sudo python setup.py install

**5.** Now that we have installed SPI-Py we can now clone our RFID RC522 Python code from the [PiMyLifeUp Github](https://github.com/pimylifeup/MFRC522-python). 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.

To clone this repository, we type the following two commands into your Raspberry Pi.

cd ~  
git clone https://github.com/pimylifeup/MFRC522-python.git

**6.** With the repository now saved to our Raspberry Pi we can begin programming for our RFID RC522.

Writing with the RFID RC522

To write data into the RFID card we type the following code into Write.py:

#!/usr/bin/env python  
  
import RPi.GPIO as GPIO  
import SimpleMFRC522  
  
reader = SimpleMFRC522.SimpleMFRC522()  
  
try:  
        text = raw\_input('New data:')  
        print("Now place your tag to write")  
        reader.write(text)  
        print("Written")  
finally:  
        GPIO.cleanup()

Now that we have written our script, we will want to finally test it out. Before testing out the script we make sure to have a RFID tag handy. Once ready,we type the following command into your Raspberry Pi’s terminal.

sudo python Write.py

Working steps:

* 1. Take input from the user
  2. Generate a random number
  3. Send url Request with the registration number and unique card number
  4. Write the number to the card

Reading with the RFID RC522

Now that we have written our script to write to RFID tags using our RC522 we can now write a script that will read this data back off the tag.

We start off by changing directory to make sure we are in the right place, and then we can run **nano** to begin writing our **Read.py** script.

cd ~/MFRC522-python  
sudo nano Read.py

#!/usr/bin/env python  
  
import RPi.GPIO as GPIO  
import SimpleMFRC522  
  
reader = SimpleMFRC522.SimpleMFRC522()  
  
try:  
        id, text = reader.read()  
        print(id)  
        print(text)  
finally:  
        GPIO.cleanup()

Now that we have finally finished our Read.py script we need to test it out. Before we test out the script we grab one of your RFID tags that we want to read. Once ready, we type the following command into our Raspberry Pi’s terminal.

sudo python Read.py

With the script now running, all we 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 us

An example of what a successful output would look like is displayed below.

pi@raspberrypi:~/MFRC522-python $ sudo python Read.py  
827843705425  
pimylifeup

Working steps:

1. Fetching the unique card number
2. Sending url REQUEST with the unique card number
3. Fetching the response data and using it as we want

**Server Set-up:**

For making the server we used server-side scripting language php. Basically we are creating a database which contains the informations regarding students and the card.

**Working step:**

While registration when students are filling their forms the information is stored in the database and during card issue to respective students all he needs to do is write registration number and tap the card.

For different platform like library, mess, academic section different software is there which fetch different data from the server.

**During Registration**

* 1. Fetching data from the url by REQUEST method
  2. Establish connection with server
  3. Selecting the database
  4. Inserting the data into students details table in the database

**While issuing the Card**

1. Fetch the registration number and the Unique Card Number from the url by REQUEST method
2. Establish connection with server
3. Selecting the database
4. Inserting the data into registration-unique number table

**While using the Card**

1. Fetch the unique card number from url by REQUEST method
2. Establish connection with server
3. Selecting the database
4. From the registration-unique number table we get the registration number and from using that we can get the details of student and send back the details as a JSON.

**Required Components :**

* RFID card
* RFID reader
* Raspberrypi

**SOFTWARE REQUIREMENTS:**

Lamp-server

Raspbian-OS

PROGRAMMING LANGUAGES USED:

Python

PHP

MySQL

HTML

CSS