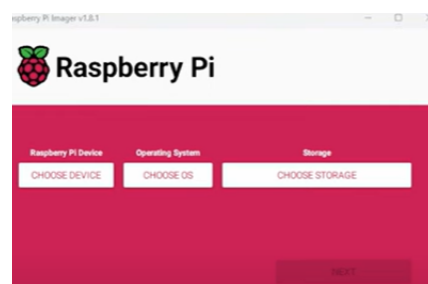


# Puzzle 1- Lector de targetes RC522

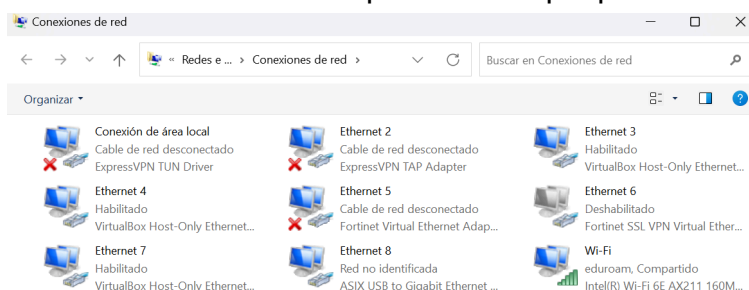
## Configuracions realitzades i connexions a internet

Primer de tot, per fer aquest projecte el primer va ser fer una cerca d'informació sobre les plaques raspberry pi i el que oferia cadascuna. Vaig optar pel model 4B amb 4G de RAM.

Tot seguit em vaig descarregar el software per windows 11 i el vaig configurar amb el tipus de dispositiu, tamany i on ho guardaria. Després havia d'especificar el nom del meu host, país, usuari i contrasenya, especificacions de la connexió per wifi i finalment activar el SSH. Tot això per guardar el sistema operatiu configurat en la tarjeta micro SD i posar-la al seu compartiment a la placa.



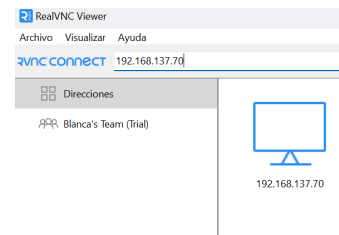
Per connectar la raspberry a l'ordinador ho vaig fer mitjançant un cable ethernet (i el seu adaptador pertinent tipus-C per l'ordinador). Primer de tot mirant les connexions de xarxa a l'ordinador i habilitant l'ús compartit de wifi pel port ethernet 8.



Viag descarregar l'aplicació putty per trobar la ip de la raspberry, posant el nom del host: raspberrypi.ns.net i, al obrir-se el terminal, fent un ifconfig.

```
blanca@raspberrypi: ~  
login as: blanca  
blanca@raspberrypi.local's password:  
Linux raspberrypi 6.6.31+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.31-1+rpt1 (2024-05-29) aarch64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Oct 1 19:47:58 2024  
blanca@raspberrypi:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.137.70 netmask 255.255.255.0 broadcast 192.168.137.255  
    inet6 fe80::3cfa:256a:ff75:d354 prefixlen 64 scopeid 0x20<link>  
    ether 2c:cf:67:5a:f0:a7 txqueuelen 1000 (Ethernet)  
    RX packets 435 bytes 62077 (60.6 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 211 bytes 29284 (28.5 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0
```

Després em vaig descarregar l'aplicació VNC Software i VNC Viewer per poder utilitzar un escriptori per la raspberry. Obrint el viewer i afegint la ip de la raspberry trobada vaig crear l'escriptori per aquest dispositiu.

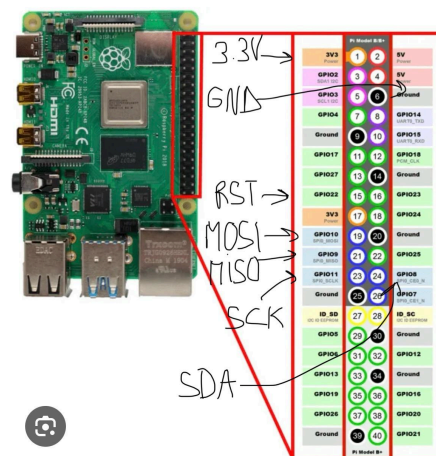


Una vegada vaig tenir l'escriptori vaig haver de modificar el fitxer config.txt per afegir al final del tot dues línies per activar el port spi:

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

També vaig modificar la configuració de la raspberry per assegurar-me de que la interface opció SPI estava habilitada.

Després vaig connectar la raspberry al perifèric RC522 connectant 7 dels 8 pins del perifèric als pins corresponents de la raspberry de la següent manera:



Em vaig assegurar de que la raspberry es comunicués amb el perifèric amb la comanda "dmesg | grep spi" i ja vaig poder descarregar-me les biblioteques del github del meu perifèric per escriure el codi pel primer puzzle amb l'ajuda del Visual Studio Code.

## **Biblioteques i paquets instal·lats**

Abans de trobar les biblioteques correctes em vaig equivocar descarregant-me una MFRC522 que estava en c2 i no en spi, i unes altres que tenien molts directoris i no trobava cap codi d' exemple útil així que no em van ajudar.

Per a que funcionés el perifèric vaig haver de descarregar dos paquets del github: SPI-Py , MFRC522-python.

El SPI-Py es un paquet amb llibreries i codis en python3 que permeten utilitzar qualsevol perifèric que es comuniqui pel port SPI.

El MFRC522-python es un paquet de llibreries i codis d'exemple del mòdul RC522 en concret.

## **Problemes trobats**

Abans de provar a connectar la placa amb el cable ethernet vaig provar a fer-ho amb wifi. Primer especificant al software que vaig posar a la targeta SIM el wifi al que connectaria l'ordinador i la placa. Després vaig utilitzar l'aplicació putty per connectar-me a la placa posant el nom del host: raspberrypi.local. Després al VNC viewer vaig tornar a posar el host per a que se'm connectés i creés un escriptori virtual per la placa. El problema va ser que al tornar a connectar-me un altre dia no em trobava el host així que vaig haver de canviar a ethernet.

## **Codi ben format**

```
#importacio de les llibreries necessaries per fer funcionar el perifric
```

```
import MFRC522
```

```
import signal
```

```
import time
```

```
#variable per indicar al perifric si ha de continuar llegint, inicialitzem a cert  
continue_reading = True
```

```
#Funcio per llegir i converir el que li arriba en un string en format hexadecimal
```

```
def uidToString(uid):
```

```
    mystring = ""
```

```
    for i in uid:
```

```
        mystring = format(i, '02X') + mystring
```

```
    return mystring
```

```
#Funcio que posa a fals la variable global que diu si el periferic pot continuar llegint o no si li arriba un Sigint (ctrl C)
```

```
def end_read(signal, frame):
```

```
    global continue_reading
```

```
    print("Ctrl+C captured, ending read.")
```

```
    continue_reading = False
```

```
# Si captura un SIGINT fa el end_read
```

```
signal.signal(signal.SIGINT, end_read)
```

```
# Crea un objecte de la classe MFRC522
```

```
MIFAREReader = MFRC522.MFRC522()
```

```

# Printa un missatge per començar a llegir
print("lector preparat, clica Ctrl-C per aturar l'execucio")

#Bucle per llegir tarjetes mentre la variable per llegir sigui certa
while continue_reading:

# Busca targetes a prop
    (status, TagType)=
MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)
    #Si l'estat es igual al mi_ok es que ha trobat una targeta
    if status == MIFAREReader.MI_OK:
        #Aconsegueix el UID de la tarjeta i l'estat
        (status, uid) = MIFAREReader.MFRC522_SelectTagSN()
        #Si encara esta la targeta, mostra per pantalla el UID de la tarjeta (guardat a la
        variable uid)
        if status == MIFAREReader.MI_OK:
            print("Card read UID: %s" % uidToString(uid))

    time.sleep(0.5) #atura mig segon per donar temps a treure tarjeta

```

Exemple execució alternant clauer i tarjeta:

```

blanca@raspberrypi:~/MFRC522-python $ python3 Puzzle1.py
lector preparat, clica Ctrl-C per aturar l'execucio
Card read UID: 24BAC4B3
Card read UID: FA88A303
Card read UID: 24BAC4B3
Card read UID: FA88A303
Card read UID: 24BAC4B3
^C Ctrl+C captured, ending read.
blanca@raspberrypi:~/MFRC522-python $

```