

# COMPTE RENDU CCF 2023

---

Dylan PERINETTI

---



Raspberry Pi  
Foundation

## GLOSSAIRE

BDD : Base De Données (DB - en Anglais)

DEL : Lampe Électroluminescente (LED - en Anglais)

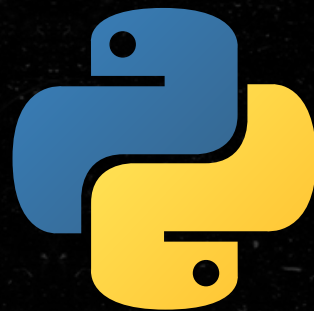
GPIO : General Purpose Input/Output

OS : Operating system (Système d'exploitation - en Français)

SD : Secure Digital

RFID : Radio Frequency IDentification

SPI : Serial Peripheral Interface



# COMPTE RENDU CCF 2023



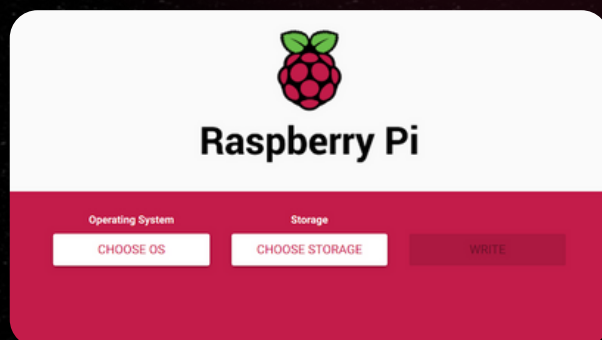
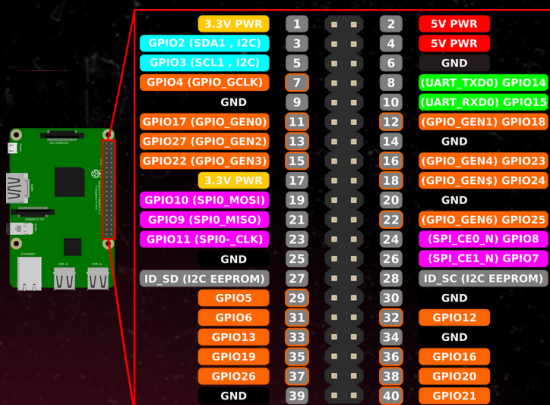
Raspberry Pi  
Foundation

Dylan PERINETTI

## ETAPE 1 : Raspbian OS

- Téléchargement de Raspberry Pi Imager
- Sélection et Installation de l'OS ( Bullseye 64-bits with no desktop environment ) sur la carte micro SD
- Ajout d'un fichier nommé "ssh" sur la carte micro SD
- Introduction de la carte micro SD dans la Raspberry Pi 3B+
- Connexion de la carte Raspberry Pi au réseau (RJ45)
- Recherche sur le réseau de la Raspberry ( nmap ou IPscanner )

Ex : `nmap -sS 172.16.20.0/23`



## ETAPE 2 : GPIO

- Connexion à la Raspberry par ssh  
Ex : `ssh 172.16.21.118 -lpi`
- Activation du GPIO et SPI dans le menu Raspi-config
- Connexion sur le GPIO des DELs et du MFRC522
- Test du fonctionnement des DELs (ANNEXE1-2)
- Test du fonctionnement du capteur RFID MFRC522
- Vérification de l'avancement des travaux des autres techniciens



# COMPTE RENDU CCF 2023

Dylan PERINETTI



Raspberry Pi  
Foundation

## ETAPE 3 : Phase de test

- Ping de l'adresse du serveur de base de données depuis la Raspberry Pi

Ex : `ping 172.16.21.72`

- Test de connexion à l'aide d'un script python3 (ANNEXE1-3)
- Modification et test du code de production (ANNEXE1-1)
- Test de différents badges RFID
- Vérification des DELs et des lignes ajoutés dans la BDD



## ETAPE FINALE

- Création d'un shell et ajout de ce dernier dans crontab (ANNEXE1-4)
- Mise en commun des travaux effectués par chaque technicien
- Vérification de la réalisation des objectifs fixés dans le cahier des charges
- Test de toutes les situations spécifiques
- Recette du projet

```
thelinuxproject@Perinetti:~$ ping 172.16.21.72
PING 172.16.21.72 (172.16.21.72): 56 data bytes
Request timeout for icmp_seq 0
^C
--- 172.16.21.72 ping statistics ---
2 packets transmitted, 0 packets received, 100.0% packet loss
- - -
```

# ANNEXE1-1

Dylan PERINETTI



Raspberry Pi  
Foundation

```
PythonProd.py UNREGISTERED

1  #!/usr/bin/env python
2  PORTENUMBER = 1
3  # Import des bibliothèques
4  import mysql.connector
5  import time
6  import RPi.GPIO as GPIO
7  from mfrc522 import SimpleMFRC522
8  from datetime import datetime
9
10 def allumerLed(_led):
11     GPIO.output(_led, 1) # allumer la led
12     time.sleep(5)        # temporisation en seconde
13     GPIO.output(_led, 0) # éteindre la led
14
15 #Établir la connexion à la base de données
16 connexion = mysql.connector.connect(
17     host='172.16.21.72',
18     user='Porte',
19     password='Porte',
20     database='luxeccf'
21 )
22
23 PIN_GREEN = 18           # utilisation de la pin 16
24 PIN_RED = 16             # utilisation de la pin 18
25
26
27 while 1==1:
28     curseur = connexion.cursor() # Créer un curseur pour exécuter les requêtes SQL
29     reader = SimpleMFRC522()     # associer la fonction a la variable reader
30
31     GPIO.setmode(GPIO.BOARD)    # numérotation BOARD
32     GPIO.setup(PIN_GREEN, GPIO.OUT) # activation de la pin 16 en sortie
33     GPIO.setup(PIN_RED, GPIO.OUT)  # activation de la pin 16 en sortie
34     try:
35         id, login = reader.read() # récupérer l'id de la carte et la chaîne de caractère qui y est contenue
36         curseur.execute("SELECT id FROM users WHERE CardNumber = %s", (id,)) # Exécute la requête SQL dans la base
37         resultat = curseur.fetchone() # Récupérer le contenu de la réponse dans la variable resultat
38
39         if resultat is not None:
40             print(resultat)
41             curseur.execute("INSERT INTO history(ID,UserID,DT,DoorNumber) VALUES (NULL, %s, NOW(), %s)", resultat+(PORTENUMBER,))
42             connexion.commit()
43             print("Bienvenue " + login)
44             print("Accès autorisé")
45             allumerLed(PIN_GREEN)
46         else:
47             # insert in BAD
48             print("Accès refusé")
49             allumerLed(PIN_RED)
50     finally:
51         GPIO.cleanup()
52         curseur.close()
53         connexion.close()
54
```

Line 29, Column 80 Spaces: 4 Python



# ANNEXE1-2

Dylan PERINETTI



Raspberry Pi  
Foundation

```
acces (2).py  UNREGISTERED
1 # Import des bibliotheques
2 import RPi.GPIO as GPIO
3 import time
4
5 pin_green = 15 # utilisation de la pin 15
6 pin_red = 16 # utilisation de la pin 16
7 GPIO.setmode(GPIO.BOARD) # numérotation BOARD
8 GPIO.setup(pin_red, GPIO.OUT) # activation de la pin 16 en sortie
9 GPIO.setup(pin_green, GPIO.OUT) # activation de la pin 16 en sortie
10
11 try:
12     GPIO.output(pin_red, 1) # allumer la led
13     time.sleep(5) # temporisation en seconde
14     GPIO.output(pin_red, 0) # eteindre la led
15     GPIO.output(pin_green, 1) # allumer la led
16     time.sleep(5) # temporisation en seconde
17     GPIO.output(pin_green, 0) # eteindre la led
18 except:
19     GPIO.cleanup() # extinction de la led à l'arrêt du programme
```

Line 1, Column 27 Spaces: 4 Python

# ANNEXE1-3



Raspberry Pi  
Foundation

Dylan PERINETTI

```
PytonProd.py UNREGISTERED
1 #!/usr/bin/env python
2
3 import mysql.connector# Import de la bibliothèque
4
5 #Établir la connexion à la base de données
6 db = mysql.connector.connect(
7     host='172.16.21.72',
8     user='mennan',
9     password='Gre@tB@lls0ffire',
10    database='luxé-ccf'
11 )
12 cursor = db.cursor()
13 try:
14     cursor.execute("SELECT VERSION()")
15     results = cursor.fetchone()
16     ver = results[0]
17     if (ver is None):
18         return False
19     else:
20         return True
21 except:
22     print "ERROR IN CONNECTION"
23     return False
```

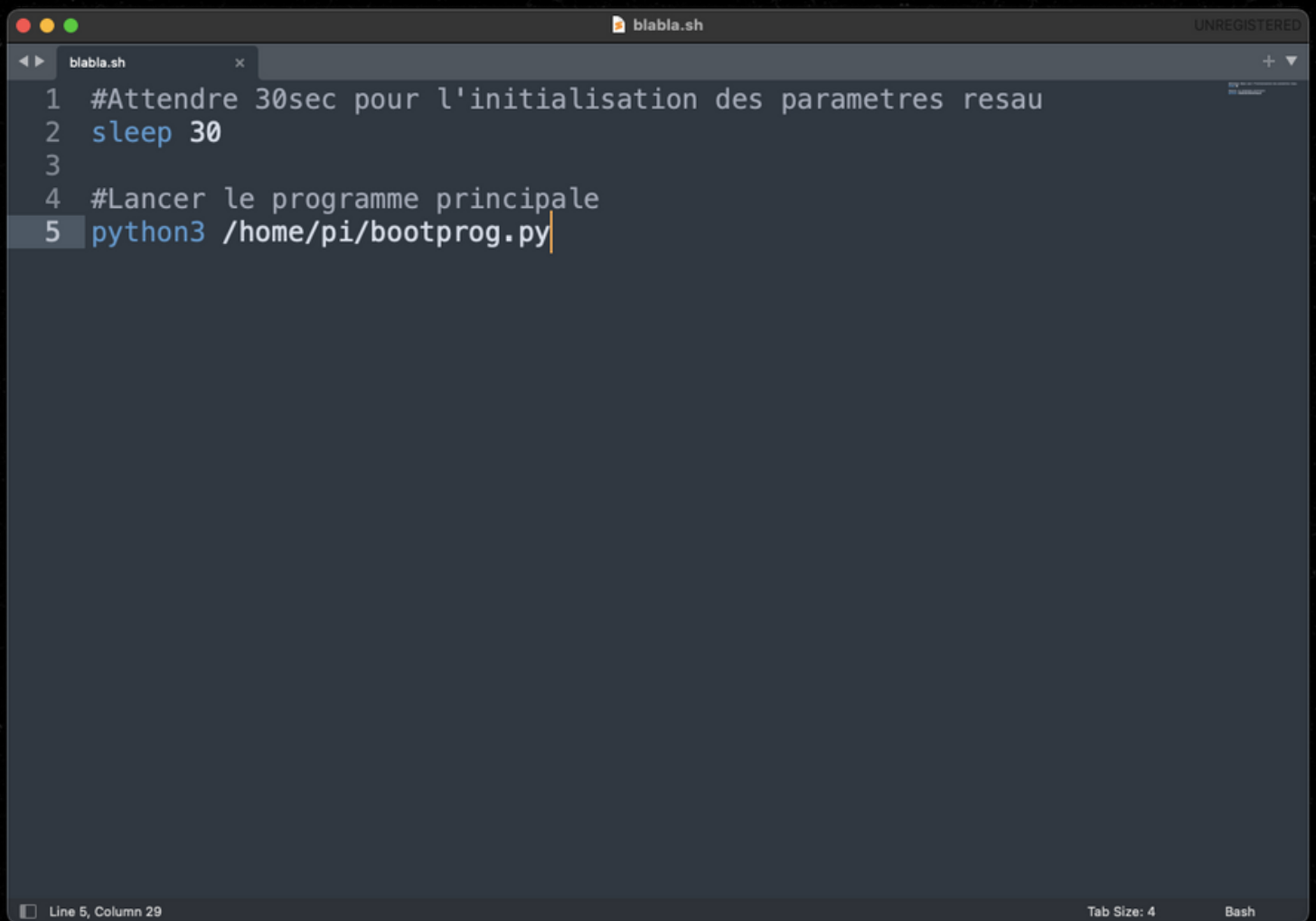
Line 2, Column 1 Spaces: 4 Python

# ANNEXE1-4

Dylan PERINETTI



Raspberry Pi  
Foundation

A screenshot of a terminal window titled 'blabla.sh'. The window contains a shell script with five lines of code. The first line is a comment in French: '#Attendre 30sec pour l\'initialisation des parametres resau'. The second line is 'sleep 30'. The third line is empty. The fourth line is another comment: '#Lancer le programme principale'. The fifth line is 'python3 /home/pi/bootprog.py'. The cursor is at the end of the fifth line. The terminal window has a dark background and a light-colored text. The status bar at the bottom shows 'Line 5, Column 29', 'Tab Size: 4', and 'Bash'.

```
1 #Attendre 30sec pour l'initialisation des parametres resau
2 sleep 30
3
4 #Lancer le programme principale
5 python3 /home/pi/bootprog.py
```