

Compte rendu SAE 102

S'initier aux réseaux informatique

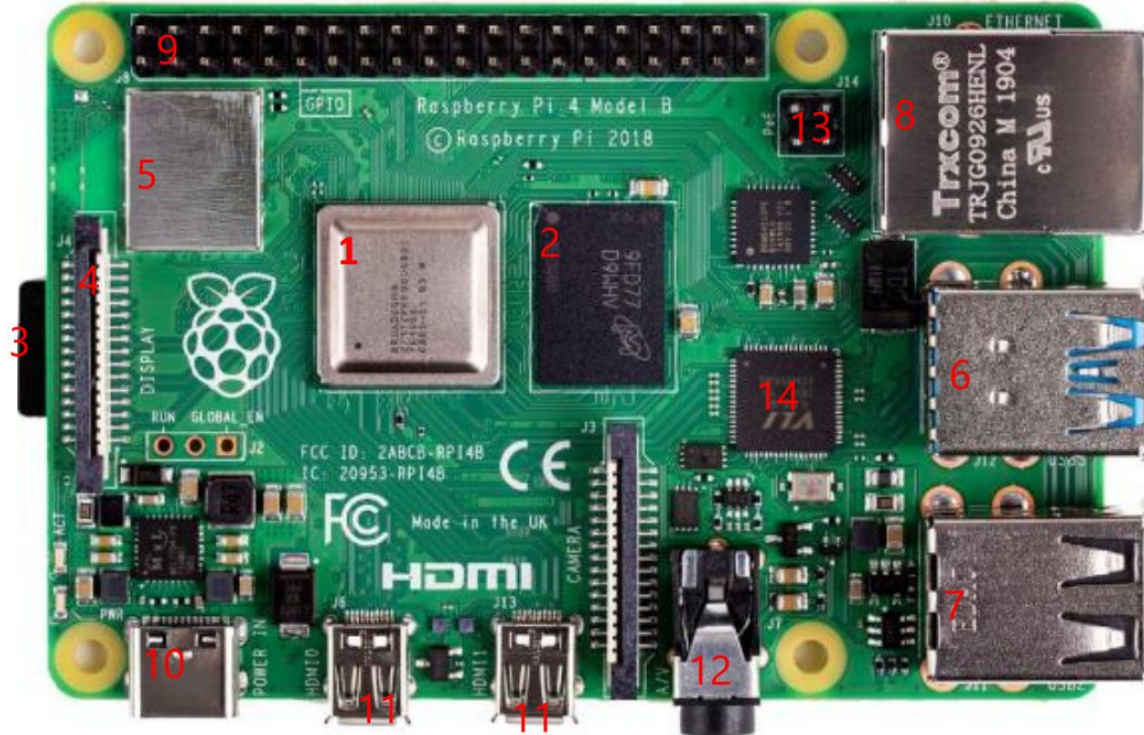
Tâche réaliser par :

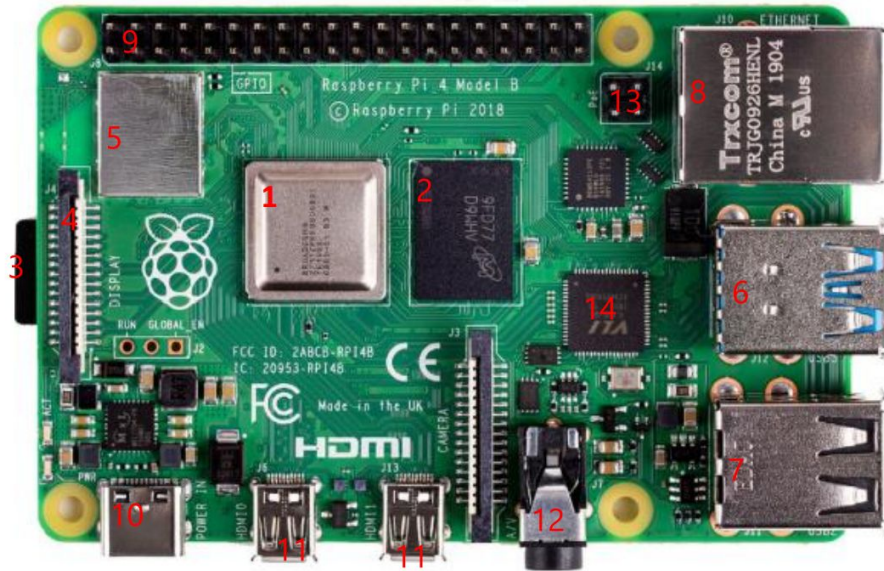
Emmanuel Grondin	a fait la vidéo et la diapo ,activer le ssh, a fait le montage du capteur dht22 et des leds(montage, etc)
Benoit Dijoux	
Yann Boisvillier	Installer l'Os, fait les document réponses

Présentation de la carte Raspberry Pi :

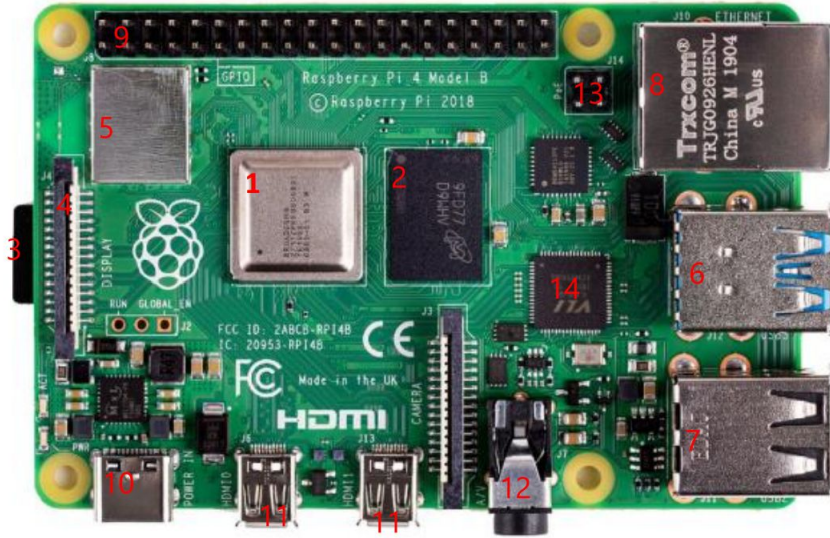
Repérer et donner la désignation des principaux constituants de la carte Raspberry Pi

DOCUMENT RÉPONSE :





Numéro	Composant
1	Microprocesseur (Central Processor Unit)
2	Mémoire vive (RAM)
3	Mémoire de stockage (MicroSD Card slot)
4	display port
5	2.4/5GHz Bluetooth 5.0
6	USB 3.0
7	USB 2.0
8	Port Ethernet



Numéro	Composant
9	broche GPIO
10	alimentation USB-C 5V
11	micro port HDMI
12	Ports audio (Audio jack)
13	Connecteur pour le HAT PoE (alimentation via ethernet)
14	contrôleur USB

Les spécifications et les caractéristiques des principaux composants de la Raspberry PI :

Catégorie	Valeur/Caractéristique	Commande Linux (si nécessaire)
Raspberry Pi 4		
Numéro de série	a0808ebe	vcgencmd otp_dump grep 28:
Type	Single Board Computer (SBC)	
Modèle	Raspberry Pi 4	

Processeur		
Modèle	Broadcom BCM2711	
Architecture	ARM Cortex-A72	
Nombre de CPU	4	
Nombre de Thread / CPU	4	
Fréquence (Hz)	1.5 GHz	cat /proc/cpuinfo

Mémoire		
Type mémoire vive	LPDDR4	
Capacité mémoire vive	4 Go (options disponibles : 2 Go, 8 Go)	<code>free -h</code>
Type mémoire de stockage	MicroSD	
Capacité mémoire stockage	Dépend la capacité de stockage de votre carte MicroSD	<code>df -h</code>

Wifi	@Mac Wifi (si interface existe)	
Norme	IEEE 802.11	<code>iwconfig</code>
Débit	[Débit de la connexion WiFi]	<code>iwconfig</code>
Bluetooth	5.66	<code>bluetoothctl</code> puis <code>version</code>

Vidéo		
Modèle GPU	VideoCore VI	
Taille Mémoire	Partagée avec la RAM (configurable dans le BIOS)	
Connectivité		
Réseau Filaire	Ethernet	
Norme	Gigabit	
Débit	[Débit de votre connexion]	ethtool eth0

HDMI	2
Version	2.0
Type de connecteur sortie	HDMI
USB	2 × USB 3.0, 2 × USB 2.0
Type	USB [Type]
Contraintes	
Température de fonctionnement	0°C à 50°C
Tension limite sur les E/S (GPIO)	3.3V

**Système
d'exploitation**

Nom
distribution

Raspbian

`lsb_release -a`

Version

12

`lsb_release -a`

**Environnement
de bureau**

bookworm

`echo
$DESKTOP_SESSION`

**Ports et
interfaces**

GPIO

40

[Nombre de
GPIO
disponibles]

➤ Donner le rôle de toutes les broches GPIO de la carte Raspberry PI à votre disposition (sous forme de tableau par exemple) :

La Raspberry Pi 4 dispose de 40 broches GPIO (General Purpose Input/Output), numérotées de 1 à 40.

Les rôle de chaque broche GPIO sur la Raspberry Pi 4 :

PIN	Fonction	Commentaire
1	3.3V	Alimentation 3.3V
2	5V	Alimentation 5V
3	GPIO2 SDA0 (I2C)	Données SDA pour le bus I2C
4	5V	Alimentation 5V
5	GPIO 3 SCL 0 (I2C)	Horloge SCL pour le bus I2C
6	GND	masse (Ground)
7	GPIO4	Broche GPIO 4
8	GPIO 14 TXD	Transmission de données pour l'UART (récepteur /émetteur asynchrone universel définit un protocole)

PIN	Fonction	Commentaire
9	GND	Terre (Ground)
10	GPIO 15 RXD	
11	GPIO17	Broche GPIO 17
12	GPIO18	Broche GPIO 18(PCM_CLK)
13	GPIO 27	Broche GPIO 27
14	GND	MASSE (Ground)
15	GPIO22	Broche GPIO 22
16	GPIO23	Broche GPIO 23

PIN	Fonction	Commentaire
19	GPIO 10 MOSI (SPI)	MOSI (Master Out Slave In) pour le bus SPI
20	GND	MASSE (Ground)
21	GPIO 9 MISO (SPI)	MISO (Master In Slave Out) pour le bus SPI
22	GPIO 25	Broche GPIO 25
23	GPIO 11 SCLK (SPI)	Horloge SCLK pour le bus SPI
24	GPIO 8 CE0 (SPI)	Chip Enable 0 pour le bus SPI

PIN	Fonction	Commentaire
25	GND	MASSE(Ground)
26	GPIO 7 CE1 (SPI)	Chip Enable 1 pour le bus SPI
27	GPIO 0 ID_SD	EEPROM ID pour la détection automatique HAT
28	GPIO 1 ID_SC	EEPROM ID pour la détection automatique HAT
29	GPIO5	Broche GPIO 5
30	GND	MASSE (Ground)
31	GPIO6	Broche GPIO 6
32	GPIO12	Broche GPIO 12

PIN	Fonction	Commentaire
33	GPIO13	Broche GPIO 13
34	GND	MASSE (Ground)
35	GPIO 19 (PCM_FMS)	Broche GPIO 19
36	GPIO16	Broche GPIO 16
37	GPIO 26	Broche GPIO 26
38	GPIO 20 (PCM_DIN)	Broche GPIO 20
39	GND	MASSE (Ground)
40	GPIO 21 (PCM_DOUT)	Broche GPIO 21

La valeur de la tension maximale à ne pas dépasser sur les broches du GPIO de la carte Raspberry Pi à votre disposition (hors broches d'alimentation)

La tension maximale tolérée sur les **broches GPIO** de **la Raspberry Pi** est de **3.3 volts**. **Cela s'applique à toutes les broches GPIO**, sauf (**5V et 3.3V**). Si vous appliquez une tension supérieure à 3.3V sur une broche GPIO, cela peut entraîner des dommages permanents à la carte.



Installer L'OS Raspbian :

Install Raspberry Pi OS using Raspberry Pi Imager

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.

[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`
in a Terminal window.



On l'installe au préalable
L'OS raspbian

Personnalisation de l'OS :

Personnalisation de l'OS

GÉNÉRAL SERVICES OPTIONS

☒ Nom d'hôte raspberrypi .local

☒ Définir nom d'utilisateur et mot de passe

Nom d'utilisateur : Manu

Mot de passe : ●●●●●●

☒ Configurer le Wi-Fi

SSID : RT-WIFI

Mot de passe : ●●●●●●●●●●●●●●●●

☐ Afficher le mot de passe ☐ SSID caché

Pays Wi-Fi : FR

☒ Définir les réglages locaux

Fuseau horaire : Indian/Reunion

Type de clavier : fr

ENREGISTRER

Configurer la Raspberry :

[illegible]

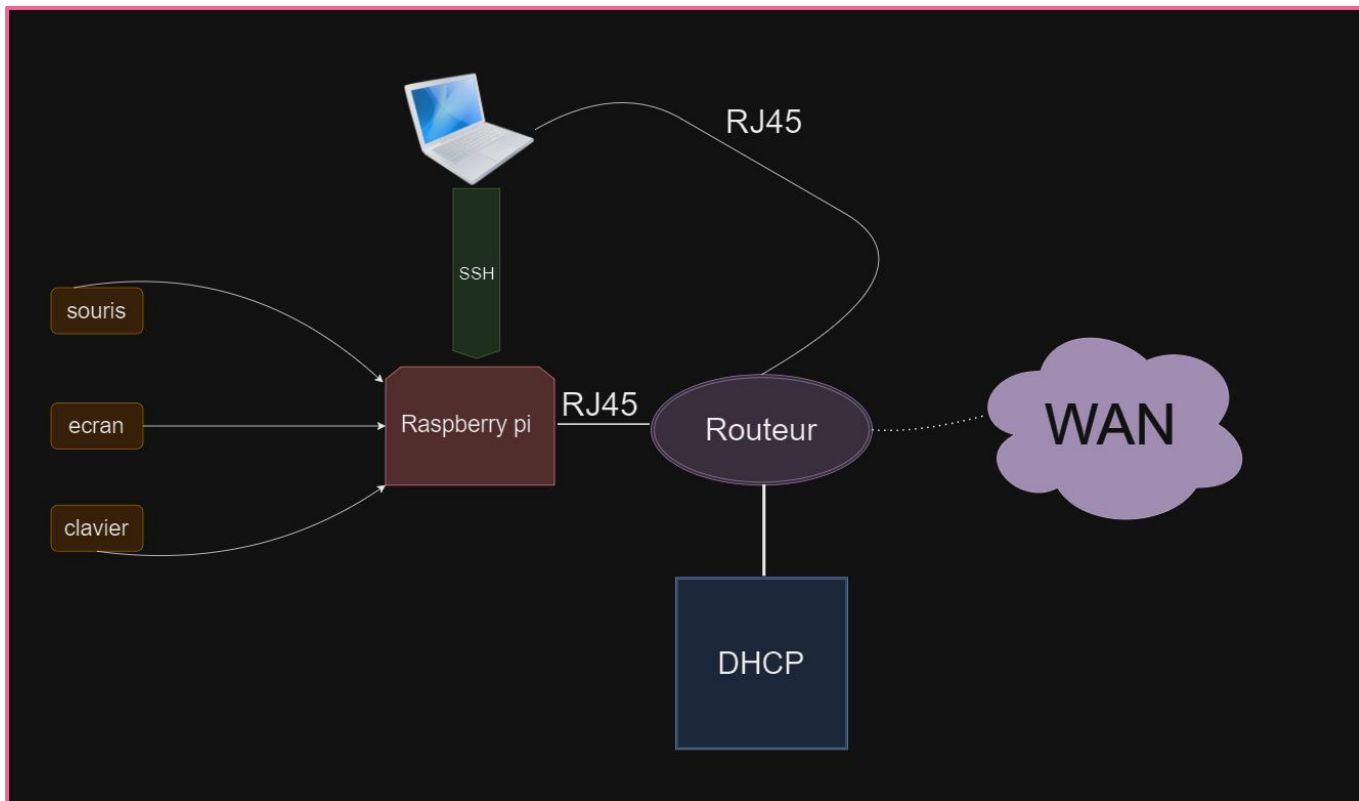
En tapant **#sudo raspi-config** on accède à la configuration de celle-ci

Activer SSH sur la raspberry :

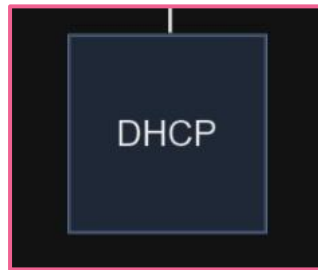
[illegible]

En activant **SSH** dans la configuration on pourra désormais se connecter à distance sur la raspberry (avec le login et le Mot de passe).

Le schéma de câblage de la carte Raspberry PI au modem-routeur :



Le rôle du service DHCP :



Le DHCP (Dynamic Host Configuration Protocol) est comme un **distributeur automatique d'adresses** sur un réseau. Il attribue automatiquement des adresses IP aux appareils qui se connectent, les aidant ainsi à communiquer efficacement sur le réseau sans que les utilisateurs aient à configurer manuellement chaque appareil avec une adresse IP.

En plus des adresses IP, **le DHCP** peut également **distribuer d'autres informations importantes aux appareils**, telles que les adresses des serveurs DNS (Domain Name System) et les passerelles par défaut, qui sont essentielles pour permettre aux appareils de naviguer sur Internet et de communiquer avec d'autres réseaux.

Récupérer l'adresse IP avec le routeur(deux solution):

The screenshot shows the web interface of a GR140DGM router. The left sidebar contains navigation menus for 'FGW Fiber Gateway', 'LAN Réseau local', and 'WAN Réseau étendu'. The main content area is titled 'LAN Réseau local' and has tabs for 'Caractéristiques', 'DNS local', 'Equipements', 'Statistiques', and 'Baux statiques'. The 'Equipements' tab is active, displaying a table of connected devices. The table has columns for 'Hôte', 'Port', 'Etat', 'MAC', 'IPv4', 'Type d'adresse', 'Durée du bail', and 'IP'. The device 'raspberrypi' is listed on 'LAN 2' with a status of 'on' and an IPv4 address of '192.168.1.121', which is highlighted with a red box. The 'Pas de nom' device is listed on 'LAN 1' with a static IP of '192.168.1.254'.

Hôte	Port	Etat	MAC	IPv4	Type d'adresse	Durée du bail	IP
Pas de nom	LAN 1	on	1C:61:B4:00:3F:14	192.168.1.254	statique	0 secondes	---
raspberrypi	LAN 2	on	E4:5F:01:27:AD:7C	192.168.1.121	dynamique	23 heures, 32 minutes, 18 secondes	---

En ayant connecter la raspberry au routeur(ici avec un RJ45), on peut récupérer l'adresse IP en se connectant au routeur (192.168.1.1)

Nmap dans le réseau privé pour trouver l'adresse IP:

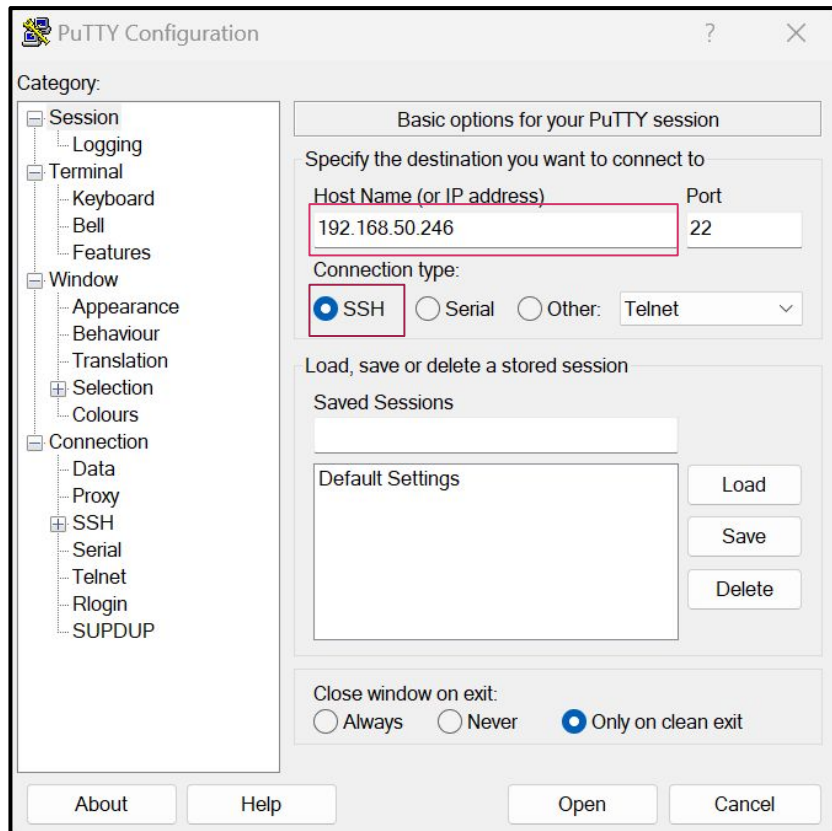
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ nmap -sn 192.168.18.1-254  
Starting Nmap 7.80 ( https://nmap.org ) at 2022-09-12 12:15 BST  
Nmap scan report for 192.168.18.1  
Host is up (0.029s latency).  
Nmap scan report for 192.168.18.12  
Host is up (0.017s latency).  
Nmap scan report for 192.168.18.16  
Host is up (0.094s latency).  
Nmap scan report for 192.168.18.20  
Host is up (0.0078s latency).  
Nmap scan report for 192.168.18.43  
Host is up (0.0088s latency).  
Nmap scan report for 192.168.18.47  
Host is up (0.13s latency).  
Nmap scan report for 192.168.18.56  
Host is up (0.12s latency).  
Nmap scan report for 192.168.18.58  
Host is up (0.00029s latency).  
Nmap scan report for 192.168.18.61  
Host is up (0.10s latency).  
Nmap scan report for 192.168.18.98  
Host is up (0.071s latency).  
Nmap done: 254 IP addresses (10 hosts up) scanned in 10.94 seconds  
pi@raspberrypi:~$
```

Installer nmap dans le terminal puis taper :

`"nmap -sn
<address_réseau>"`

pour scanner les adresses ip et puis trouver celle de la raspberry.

Se connecter en SSH sur la RASPBERRY :



Ici grâce à PuTTY (sur windows) on peut se connecter à distance via SSH en tapant l'adresse qu'on a pu récupérer au préalable. (il faudra bien sur connaître le login et le MDP pour se connecter en ssh)

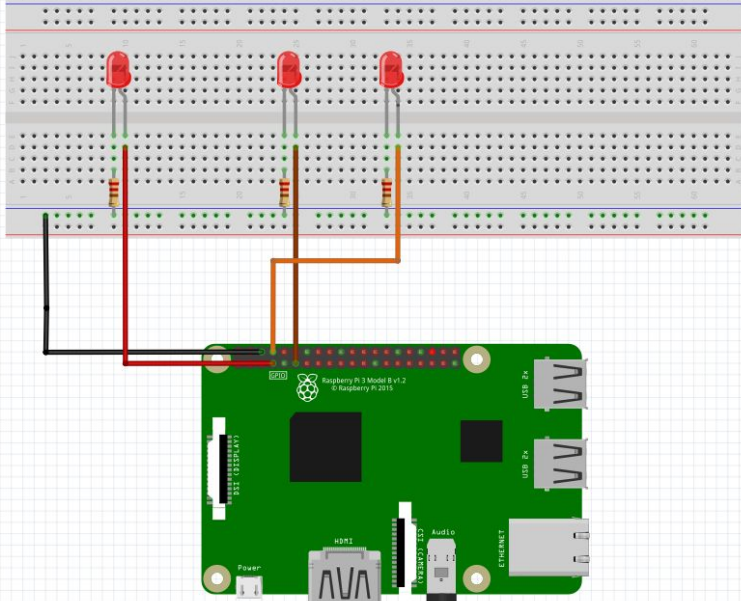
Rôle du ssh :

Le SSH (**Secure Shell**) est un protocole de communication sécurisé qui permet l'accès à distance à des ordinateurs et le transfert de fichiers via des connexions cryptées.

Il assure une authentification sécurisée des utilisateurs et des transferts de données protégés, ce qui en fait **un outil essentiel pour la gestion à distance et la sécurité des réseaux informatiques.**



Faire clignoter 3 leds grâce à un programme python :



```
import RPi.GPIO as GPIO
import time
```

```
# Définir le mode de numérotation des broches
GPIO.setmode(GPIO.BCM)
```

```
# Définir le numéro de la broche à utiliser
led_pin1 = 4
led_pin2 = 17
led_pin3 = 14
```

```
# Configurer la broche comme une sortie
GPIO.setup(led_pin1, GPIO.OUT)
GPIO.setup(led_pin2, GPIO.OUT)
GPIO.setup(led_pin3, GPIO.OUT)
```

```
try:
```

```
    while True:
```

```
        # Allumer la LED
        GPIO.output(led_pin1, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1)
        GPIO.output(led_pin2, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1)
        GPIO.output(led_pin3, GPIO.HIGH)
        print("LED allumée")
```

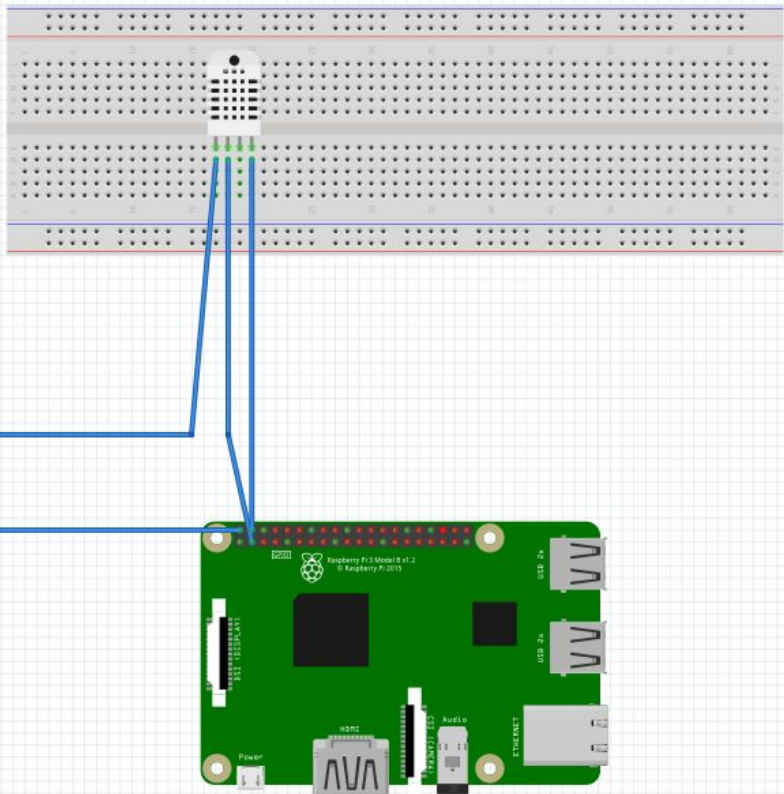
```
        time.sleep(1) #Attendre 1 seconde
        GPIO.output(led_pin1, GPIO.HIGH)
        GPIO.output(led_pin2, GPIO.HIGH)
        GPIO.output(led_pin3, GPIO.HIGH)
        time.sleep(1)
        # Éteindre la LED
        GPIO.output(led_pin1, GPIO.LOW)
        GPIO.output(led_pin2, GPIO.LOW)
        GPIO.output(led_pin3, GPIO.LOW)
        print("LED éteinte")
        time.sleep(1) # Attendre 1 seconde
```

```
except KeyboardInterrupt:
```

```
    # Arrêter le script proprement lorsque
    # Ctrl+C
    # est pressé
    GPIO.cleanup()
```

Ce programme (python) permet de faire clignoter une led 1 par 1

Capteur DHT22



#Bibliothèque

```
import RPi as GPIO
```

```
import Adafruit_DHT as dht
```

```
from time import sleep
```

```
DHT = 4
```

```
while True:
```

```
    #lire la température et l'humidité
```

```
    h,t = dht.read_retry(dht.DHT22, DHT)
```

```
    #Ecrire la température et l'humidité dans le terminal
```

```
    print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(t,h))
```

```
    sleep(5) #attendre 5s et re lire à nouveau
```

Résultat du programme :

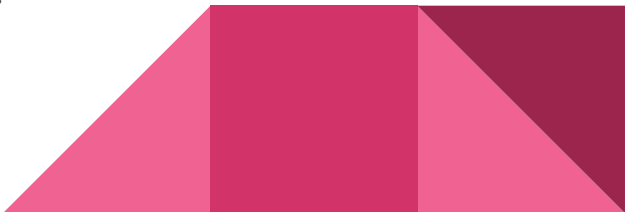
```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ nano capteur.py  
pi@raspberrypi:~ $ python capteur.py  
Temp=29.2*C Humidity=77.4%  
Temp=29.2*C Humidity=77.5%  
^CTraceback (most recent call last):  
  File "/home/pi/capteur.py", line 10, in <module>  
    sleep(5)  
KeyboardInterrupt  
  
pi@raspberrypi:~ $ python capteur.py  
Temp=29.2*C Humidity=77.6%  
Temp=29.3*C Humidity=77.7%  
Temp=29.3*C Humidity=77.7%  
Temp=29.2*C Humidity=77.7%  
Temp=29.3*C Humidity=77.7%  
Temp=29.3*C Humidity=77.8%  
Temp=29.2*C Humidity=77.8%  
Temp=29.3*C Humidity=77.8%  
^CTraceback (most recent call last):  
  File "/home/pi/capteur.py", line 10, in <module>  
    sleep(5)  
KeyboardInterrupt  
  
pi@raspberrypi:~ $ █
```

On peut donc voir que la température était de 29.3°C et l'humidité de 77.6% pendant la prise de capture de l'écran.

Le protocole MQTT :

MQTT(Message Queuing Telemetry Transport) est un protocole de messagerie simple et léger pour la communication entre appareils sur Internet. Il permet aux appareils de s'envoyer des messages de manière efficace, ce qui le rend idéal pour l'Internet des objets (IoT) et d'autres applications où la bande passante et la consommation d'énergie sont importantes.

MQTT fonctionne selon un modèle de publication/abonnement où les appareils peuvent publier des messages sur des sujets spécifiques auxquels d'autres appareils peuvent s'abonner pour recevoir ces messages.



CONCLUSION

Pour conclure **La Raspberry Pi** a révolutionné l'informatique avec sa **polyvalence, sa taille compacte et son faible coût**, stimulant l'innovation et l'apprentissage dans de nombreux domaines.

Son impact continue de façonner l'avenir de la technologie et de l'innovation.

