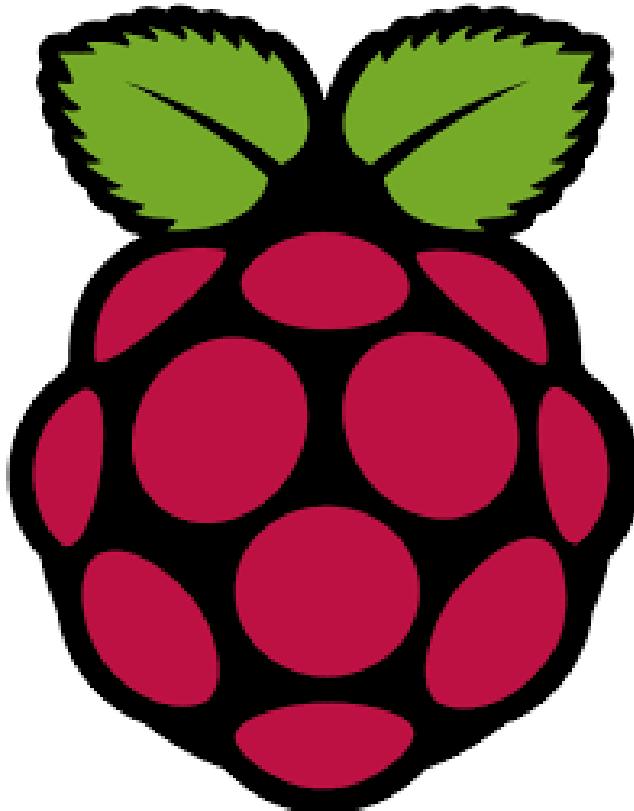


SAE 1.02

PROJET FICHE 1

DECOUVERTE DE LA CARTE RASPBERRY PI



Membre du groupe	
Houmadi	Soirihi
Hoarau	Thomas
Abdou	Imane

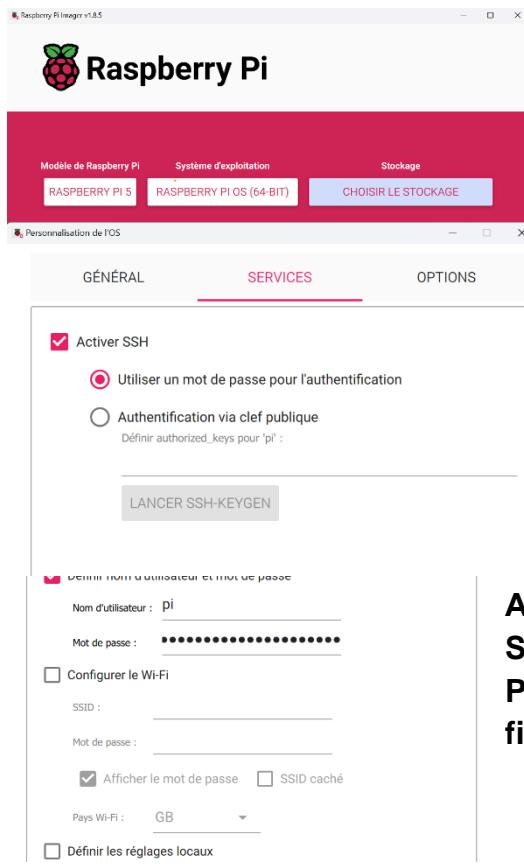
1/ Installation OS de la Raspberry Pi 5.....	3
2/ Scan du réseau	5
2.1/ Adresse IP par DHCP	5
2.3 / Logiciel Advanced IP Scanner	6
2.3/ Scanner de ports Nmap	6
3/ Connexion en SSH.....	6
3.1/ SSH avec Putty	7
3.2/ SSH en ligne de commande (Terminale)	8
4/ Schéma câblage des composants (led, dht22 et raspberry pi).....	9
Présentation du capteur DHT22	10
4.1/ Schéma de Câblage final :	12
5/ Connexion à la Raspberry Pi 5 depuis Visual Studio Code.....	13
5.1/ Mise en place de l'environnement Python.....	13
5.2/ Création d'un alias	14
5.2/ Allumer une led en Python	14
5.3/ Lire la température du capteur DHT22 Python.....	16
5.3.1/ Installation de adafruit_dht.....	16
6/ Mise en place d'un VPN	18
7/ Script Python pour température, humidité et led	19
7.1/ Recensement des pin sortie/entre	19
7.2/ Installation de la librairie PAHO	19
7.3/ Création du script Python.....	19
7.3.1/ Importation des librairies.....	19
7.3.2/ Déclaration des variables.....	20
7.3.3/ Configuration du client MQTT	20
7.3.4/ Fonction pour allumer la led.....	20
7.3.5/ Envoyer les données	21

1/ Installation OS de la Raspberry Pi 5

➤ Première phase :

Rendre une carte Raspberry PI opérationnelle afin de pouvoir la programmer (ou l'administrer) sans avoir à brancher un écran et un clavier. On se connectera à la carte par le biais du protocole SSH (Secure Shell), peu importe le réseau dans lequel elle est connectée.

Pour cela nous devons tout d'abord passer par la configuration et l'installations de l'os de la Rasberry pi 5



Ont choisis le modèle le modèle RASPBERRY PI 5 avec un système d'exploitation RASPBERRY PI OS (64-bit)

Activer le serveur SSH

Afin de pouvoir se connecter a distances à notre RASPBERRY on doit activer le protocole SSH

Mettre un nom d'utilisateur, on a choisi PI et comme mot de passe on a choisi Raspberry5

Après avoir flashé l'OS sur une carte micro SD nous pouvons l'insérer dans la Raspberry PI 5 pour intégrer nos modifications et donc finaliser la configuration.

➤ Deuxième phase :

Elle portera sur la commande à distance d'une LED connectée à la carte Raspberry Pi par le biais d'une résistance.

Pour ça nous allons d'abord utiliser le logiciel **Fritzing** pour simuler des circuits de pilotage de LED virtuellement, et une fois réussi nous pourront passer au circuit physique que nous connecteront à la Raspberry Pi.

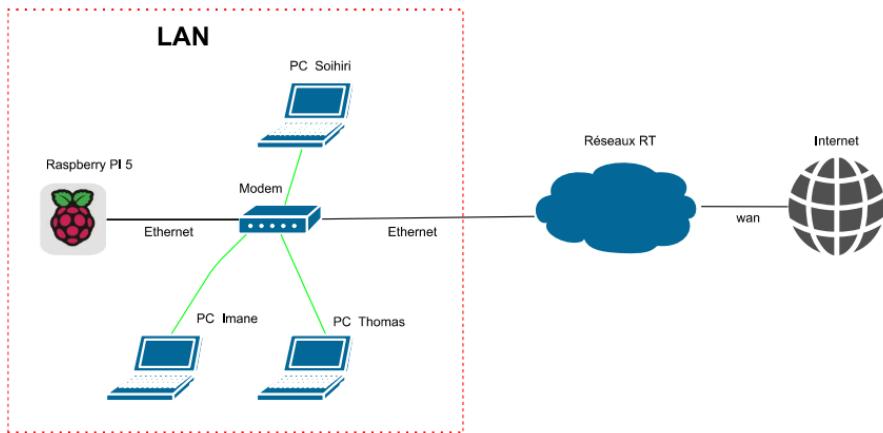
Ensuite, il nous suffira d'intégrer des programmes que nous pourrons exécuter à distance pour le pilotage des LEDs.

➤ Troisième phase :

Interfacer des capteurs (température, humidité, luminosité ...) afin que la carte Raspberry Pi puisse acquérir des mesures et les transmettre (dans le réseau local et/ou un réseau distant).

2/ Scan du réseau

2.1/ Adresse IP par DHCP



Le rôle du **service DHCP** est d'attribuer automatiquement une adresse IP et d'autres paramètres réseau (passerelle, DNS, etc.) aux appareils connectés au modem, afin de simplifier la gestion des adresses IP.

Baux DHCP dynamiques

Retrouvez tous les équipements dont l'adresse IP est attribuée par le serveur DHCP de votre Livebox.

Équipement	Adresse IP dynamique	Adresse MAC
	192.168.1.19	
raspberrypi	192.168.1.141	2C:CF:67:57:56:3C
	192.168.1.14	
	192.168.1.11	
Device-20	192.168.1.53	

2.3 / Logiciel Advanced IP Scanner

Liste des résultats		Favoris		
Statut	Nom	IP	Fabricant	Adresse MAC
	.home	192.168.1.11		
	.home	192.168.1.14		
		192.168.1.19		
	device-20.home	192.168.1.53		
	.home	192.168.1.92	Intel Corporate	
	raspberrypi.home	192.168.1.141		2C:CF:67:57:56:3C

6 actif, 0 inactif, 239 inconnu

	Information par défaut	Information modifiée
Router Username	d6nw5v1x2pc7st9m	admin
Router Password	d6nw5v1x2pc7st9m	o8058W!mFtS1X\$jK
Adresse MAC	48:F8:B3:22:F4:62	48:F8:B3:22:F4:62

Scan d'adresse IP, logiciel Advanced IP Scanner

On raspberrypi 192.168.1.141 2C:CF:67:57:56:3C

	Raspberry PI 5
Identifiant	pi
Mot de passe	Raspberry5
Adresse MAC	2C:CF:67:57:56:3C
Adresse IP	192.168.1.141

2.3/ Scanner de ports Nmap

L'autre méthode consiste à utiliser Nmap (Network Mapper) est un outil open-source. Avec la commande `nmap -sP 192.168.1.0/24` qui va scanner toutes les adresses IP du sous-réseau 192.168.1.0 à 192.168.1.255 et afficher les machines allumer/actives.

Résultat
<pre>Nmap scan report for lynksys E2500 (192.168.1.1) Host is up (0.0040s latency). MAC Address: 48:F8:B3:22:F4:62 Nmap scan report for raspberrypi.home (192.168.1.141) Host is up (0.0050s latency). MAC Address: 2C:CF:67:57:56:3C (Unknown) Nmap done: 256 IP addresses (7 hosts up) scanned in 7.14 seconds</pre>

3/ Connexion en SSH

Pour la première configuration, il va falloir passer en interface Graphique. Car par défaut il refuse la connexion SSH tant que le mot de passe par défaut “raspberry” n'est pas changé. Donc on exécute la commande :

```
$ sudo passwd pi
```

Puis on saisit un nouveau mot de passe. On redémarre. On vérifie si le SSH est activé

```
$ sudo service ssh
```

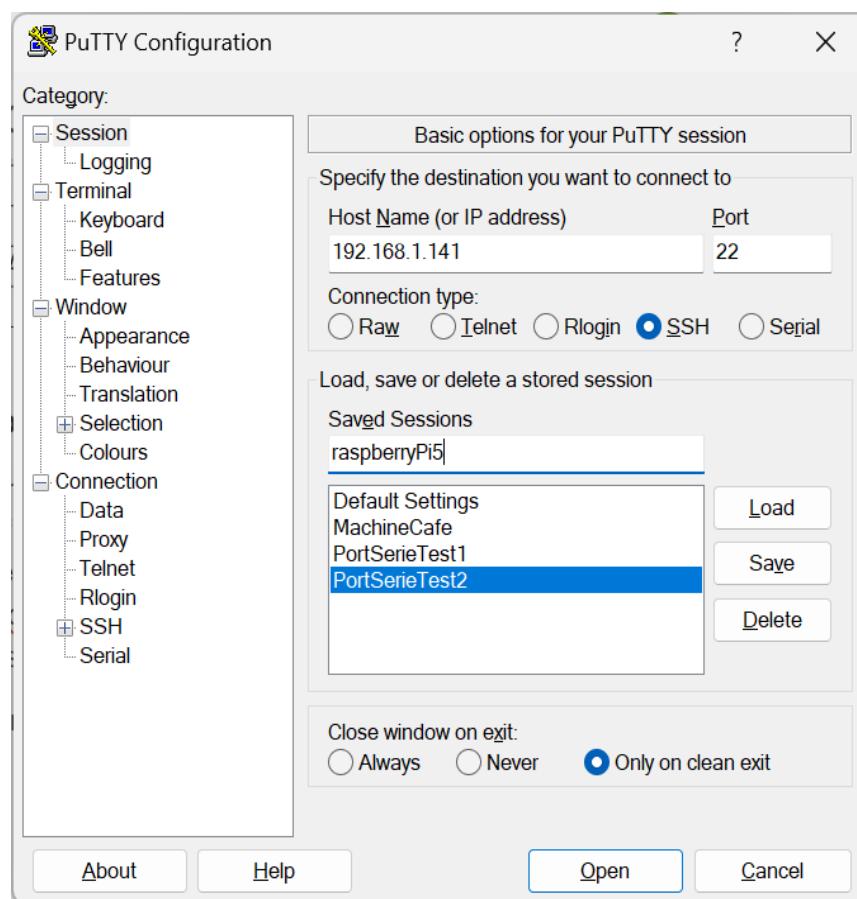
et on autorise le port de SSH par défaut 22; On installe ufw pour le pare-feu, qu'on trouve assez simple à utiliser.

```
$ sudo apt install ufw ensuite $ sudo ufw allow 22
```

Cela est nécessaire si la Raspberry pi refuse la connexion ssh.

3.1/ SSH avec Putty

Puis à partir du logiciel Putty on peut se connecter à distance on peut sauvegarder la session en précisant un nom dans “Saved Sessions” > Save pour une connexion rapide sans si d'adresse IP.

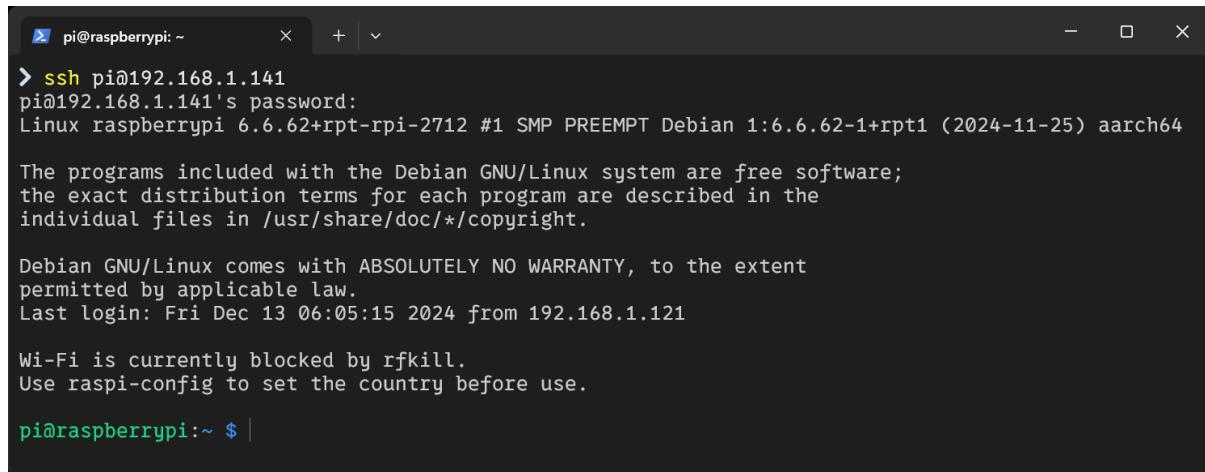


3.2/ SSH en ligne de commande (Terminale)

Phase 1

Sinon en ligne de commande

```
$ ssh pi@adresse_ip ou $ ssh pi@adresse_ip -p 22.
```



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows the following text:

```
> ssh pi@192.168.1.141
pi@192.168.1.141's password:
Linux raspberrypi 6.6.62+rpt-rpi-2712 #1 SMP PREEMPT Debian 1:6.6.62-1+rpt1 (2024-11-25) 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: Fri Dec 13 06:05:15 2024 from 192.168.1.121

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

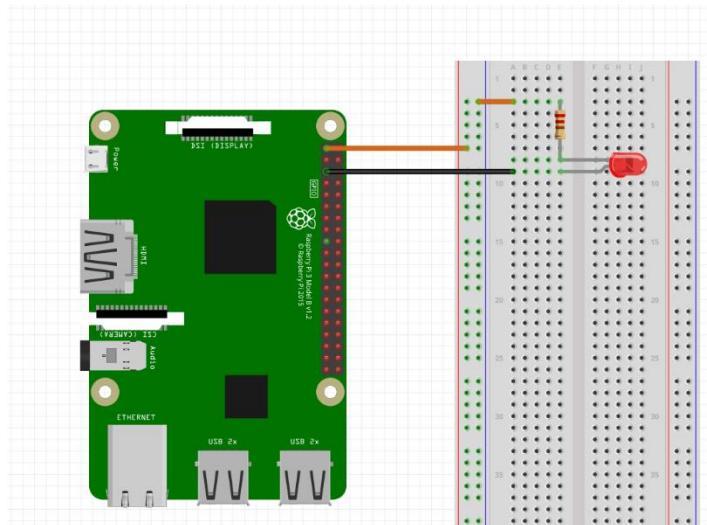
pi@raspberrypi:~ $ |
```

Après ces différentes étapes nous sommes donc connectés à la Raspberry à distance grâce au réseau sans y connecter de périphérique quelconque.

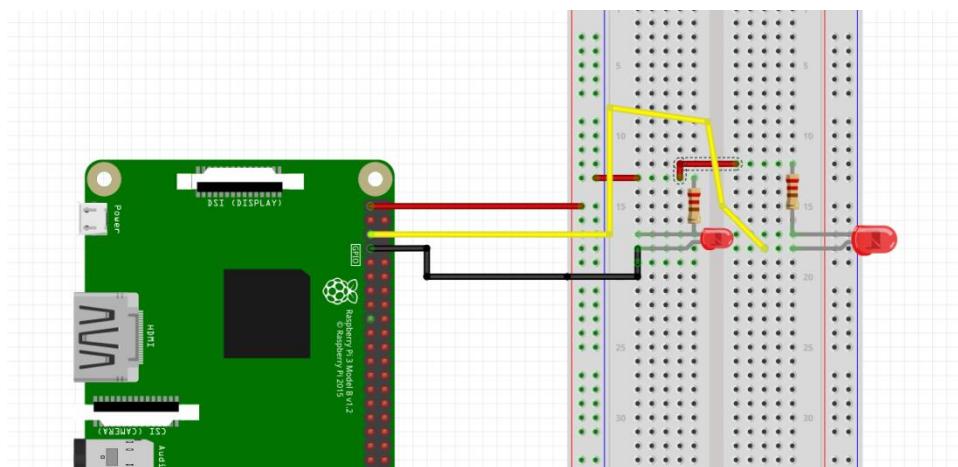
4/ Schéma câblage des composants (led, dht22 et raspberry pi)

Phase 2

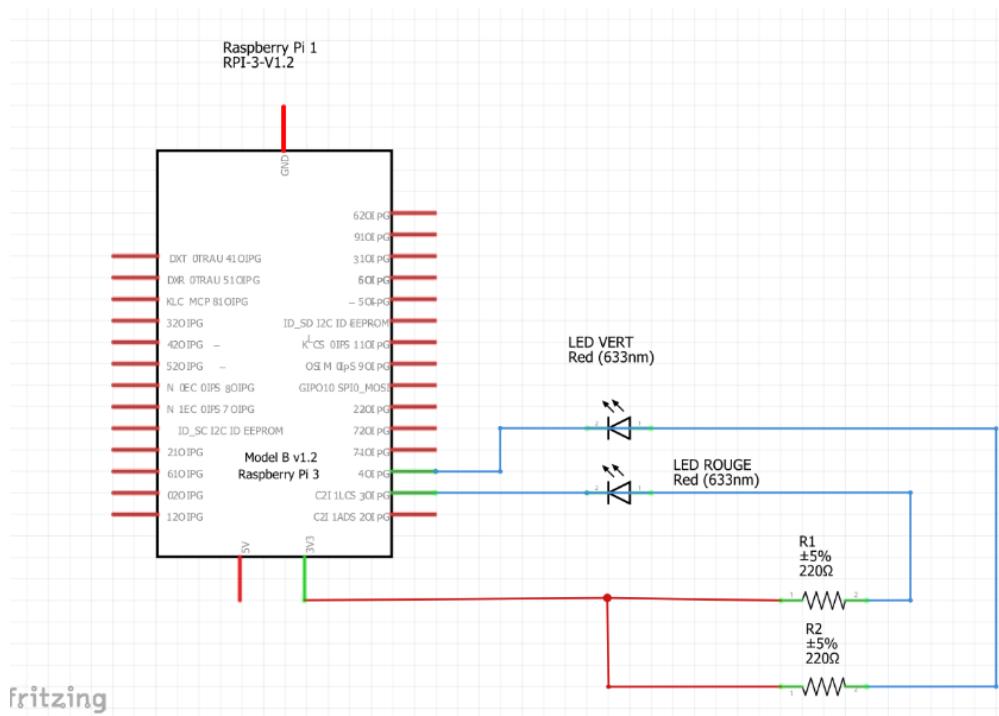
Grâce au logiciel **Fritzing** nous pouvons simuler des circuits de pilotage de LED



Circuit de pilotage d'une LED



Circuit de pilotage de 2 LEDs



Capteur DHT22

Présentation du capteur DHT22

Le capteur DHT22 est un appareil permettant la mesure de la température et l'humidité de l'air ambiant grâce à sa combinaison deux en un d'un capteur d'humidité capacitif et d'une thermistance, le tout avec une précision élevée et ne nécessite pas de composants supplémentaires pour pouvoir être utilisé.



Alimentation : 3.3 à 6 Vcc

Consommation maxi: 1.5 mA

Consommation au repos: 50 µA

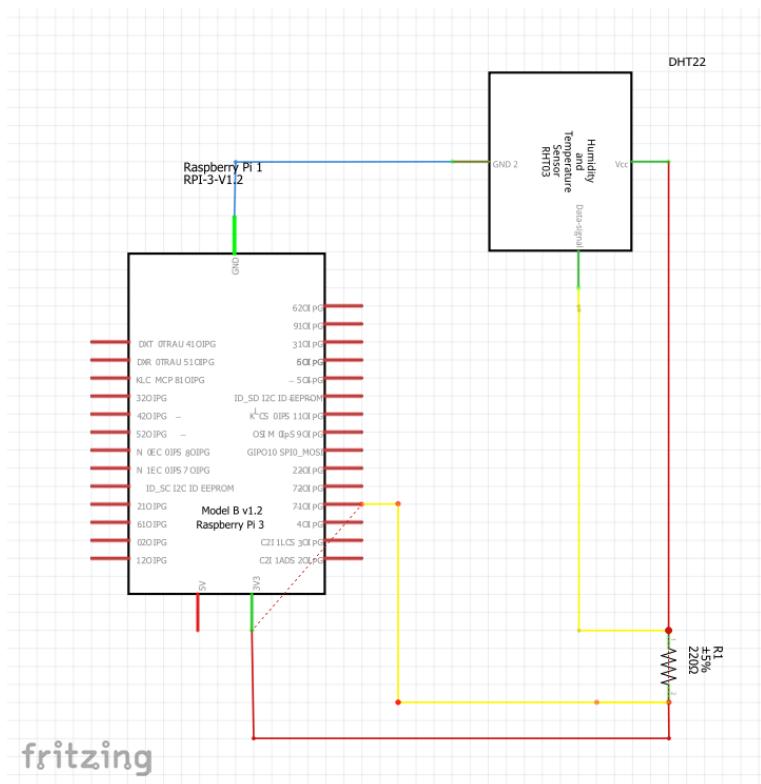
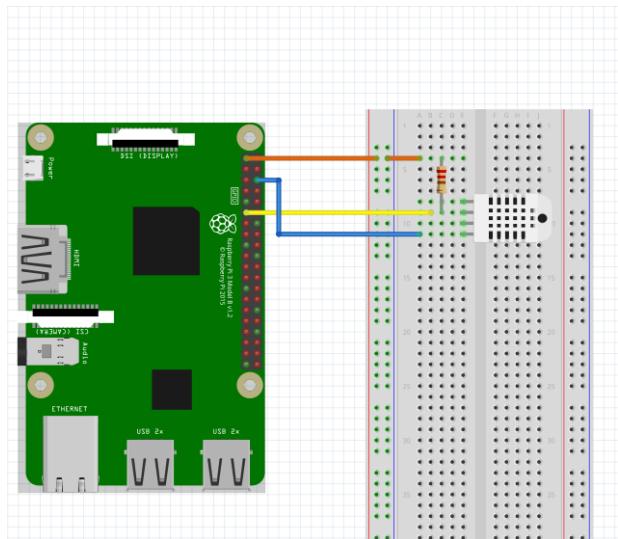
Plage de mesure :

- température: -40 à +80 °C
 - humidité: 0 à 100 % RH

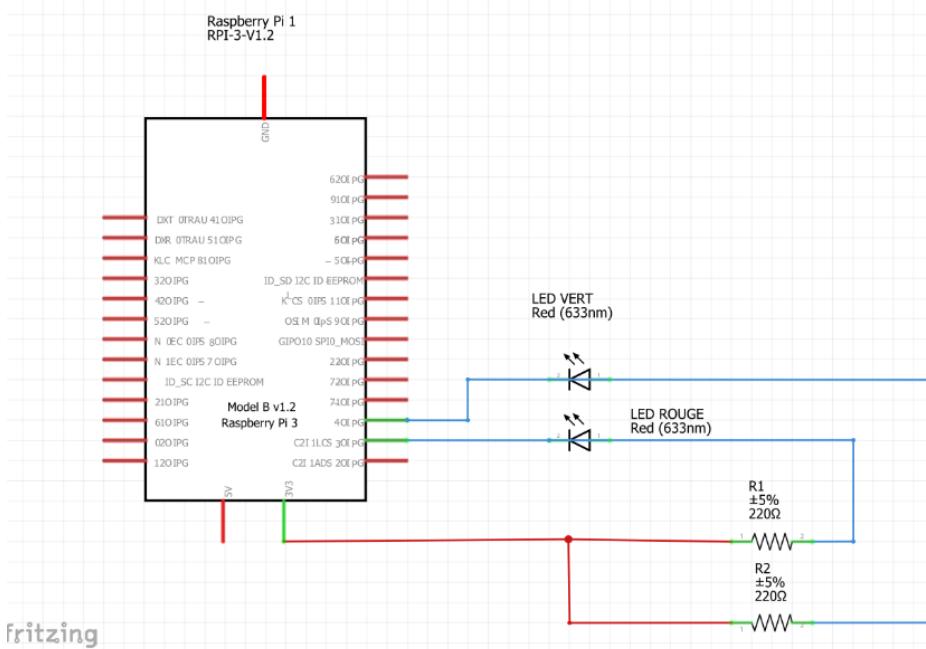
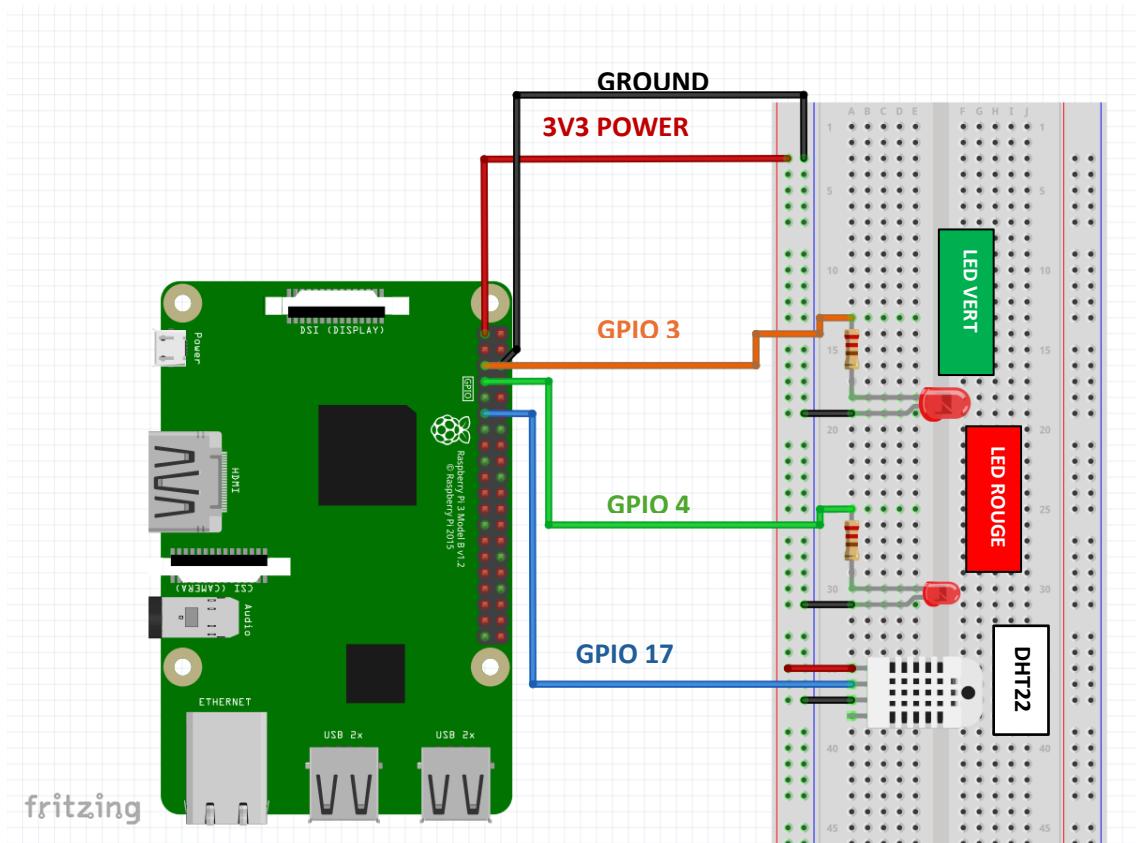
Précision:

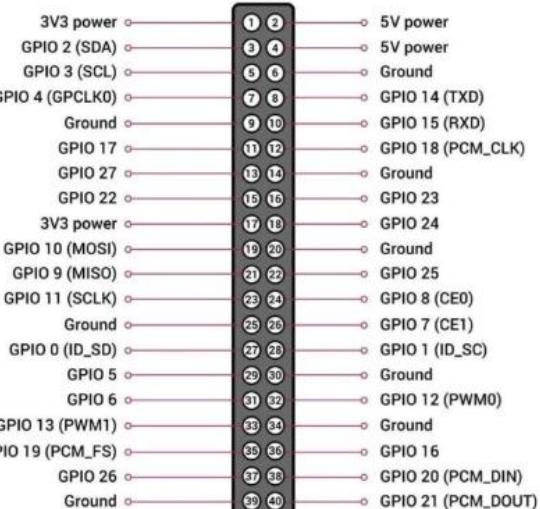
- température: $\pm 0,5$ °C
 - humidité: ± 2 % RH

Dimensions: 25 x 15 x 9 mm



4.1/ Schéma de Câblage final :





40 GPIO Pins Description of Raspberry Pi 5

5/ Connexion à la Raspberry Pi 5 depuis Visual Studio Code

Dans le fichier "**C:\Users\UTILISATEUR_ACTUEL\.ssh\config**" double clique et ouvrir avec Bloc-notes.

On rajoute les lignes suivent :

```
Host RaspberryPi5
  HostName 192.168.1.141
  User pi
```

Sauvegarder puis aller sur VS Code. Cliquer sur l'icône tout en bas à gauche qui par défaut Bleu ensuite faite « Se connecter à l'hôte et vous devriez voir "RaspberryPi5" de votre Raspberry cliquer ensuite fait « Fichier > Ouvrir un dossier » Sélectionner votre dossier de travail et ouvrez votre fichier .py.

Avant d'exécuter le programme python nous devons installer un environnement virtuel python étant donné que nous utilisons la Raspberry PI 5

5.1/ Mise en place de l'environnement Python

Pour cela nous allons utiliser la commande :

```
Python3 -m venv <nom de dossier>
```

Ensute pour activer l'environnement nous devons utiliser la commande :

```
~$ source <nom du dossier>/bin/activate
```

Après avoir créé et activer l'environnement virtuel nous pouvons exécuter les différents programmes.

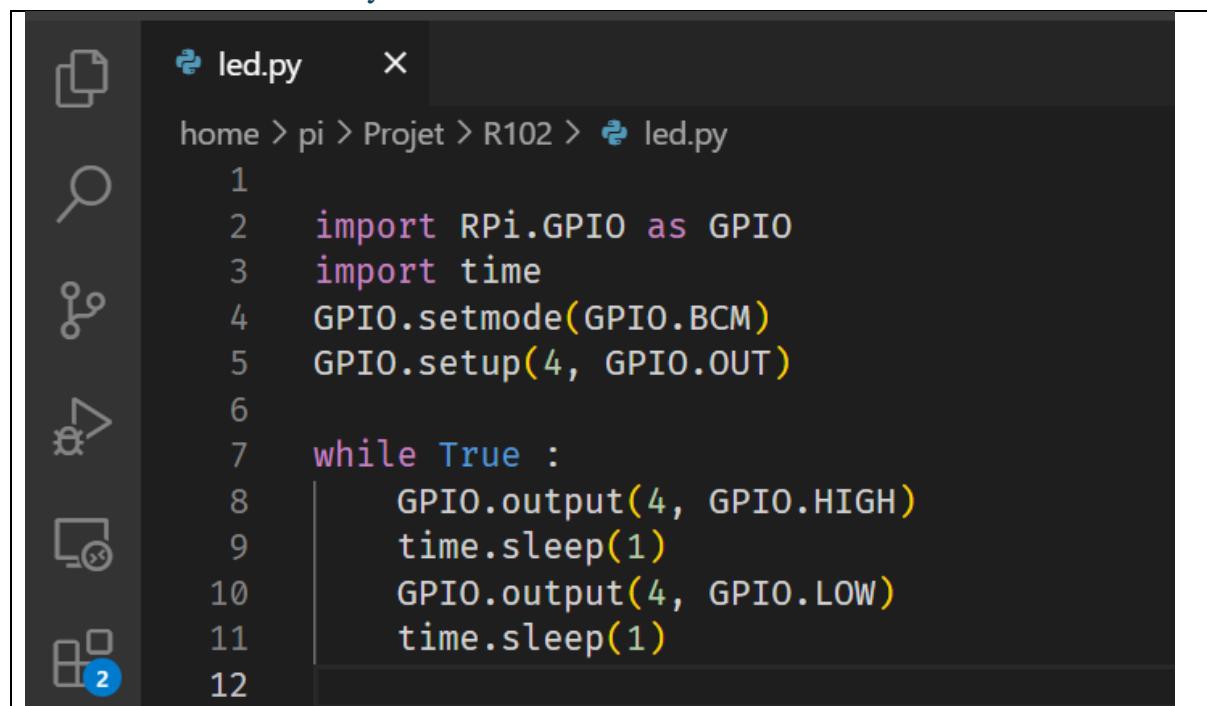
5.2/ Crédation d'un alias

Les alias permettent de créer des raccourcis pour des commandes longues ou complexes. Ils améliorent l'efficacité et la productivité en réduisant le temps de frappe des commandes répétitives. Notre alias va porter sur l'activation du venv.

```
~$ echo 'alias envPy="source  
/home/pi/Projet/R102/env/bin/activate"' >> ~/.bashrc  
  
~$ envPy  
(venv) ~$
```

Voilà il suffit maintenant de faire envPy pour activer l'environnement virtuel Python et de lancer code python.

5.2/ Allumer une led en Python



```
led.py      ×  
home > pi > Projet > R102 > led.py  
1  
2 import RPi.GPIO as GPIO  
3 import time  
4 GPIO.setmode(GPIO.BCM)  
5 GPIO.setup(4, GPIO.OUT)  
6  
7 while True :  
8     GPIO.output(4, GPIO.HIGH)  
9     time.sleep(1)  
10    GPIO.output(4, GPIO.LOW)  
11    time.sleep(1)  
12
```

Une fois le programme fait nous devons créer un répertoire :

```
$ sudo mkdir home/pi/Projet/R102
```

(Création des répertoires projet et R102)

Une fois le répertoire créé nous devons créer un fichier ou nous y intégrerons le programme

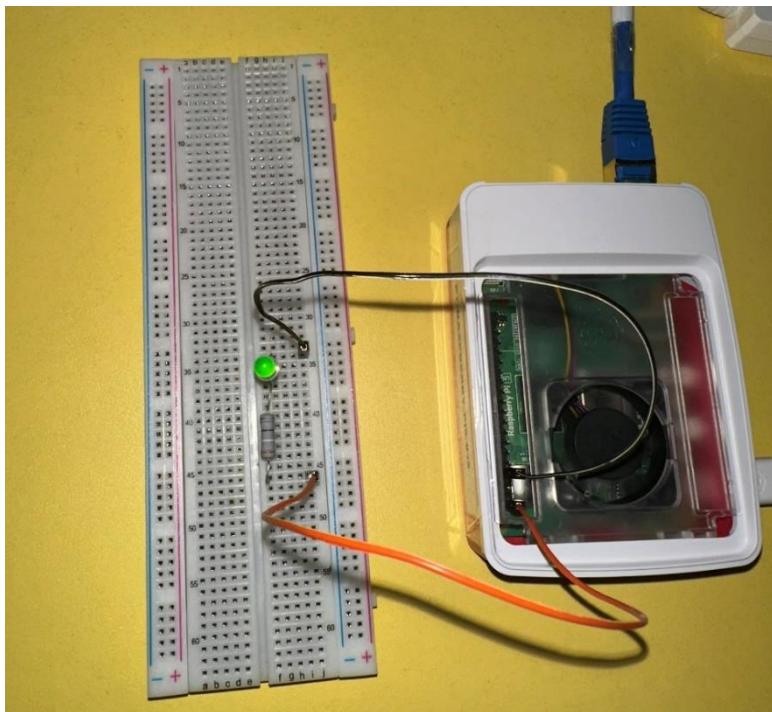
```
$ touch home/pi/Projet/R102/ led.py
```

Ensuite, grâce au logiciel Visual Code nous pouvons intégrer le programme au fichier led.py

Enfin, pour exécuter le programme il suffit de taper :

```
$ python3 home/pi/Projet/R102/led.py
```

Une fois ces étapes terminer et avec le programme ci-dessus nous sommes capables de faire clignoter une LED



Nous pouvons ensuite modifier le programme pour faire clignoter l'une après l'autre 2 LEDs (rouge et verte) ensuite le programme pour le DHT22.

```
led.py
1
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(4, GPIO.OUT)
7 GPIO.setup(3, GPIO.OUT)
8
9 while True :
10     GPIO.output(4, GPIO.HIGH)
11     GPIO.output(3, GPIO.LOW)
12
13     time.sleep(0.5)
14
15     GPIO.output(4, GPIO.LOW)
16     GPIO.output(3, GPIO.HIGH)
17
18     time.sleep(0.5)
```

5.3/ Lire la température du capteur DHT22 Python

Avant de débuter le code pour le DHT22 il va falloir installer les librairies nécessaires, c'est un ensemble d'outil qui communiquer avec un matériel.

5.3.1/ Installation de adafruit_dht

```
~$ pip3 install adafruit-circuitpython-dht
```

```
🐍 dht.py > ...
1
2     import adafruit_dht
3     import time
4     import board
5
6     dht_device = adafruit_dht.DHT22(board.D17)
7
8     while (True):
9         try:
10             temp_c = dht_device.temperature
11             temp_f = temp_c * (9/5) + 32
12             print(temp_c)
13             time.sleep(2)
14         except RuntimeError as err:
15             print(err.args[0])
```

6/ Mise en place d'un VPN

Pour que chacun puisse travailler de manière sécurisée et à distance, nous avons mis en place un VPN avec le service Tailscale, qui est très facile à configurer. Voici comment l'installer et l'utiliser.

Installation de Tailscale sur Raspberry Pi 5

Installez les paquets de transport HTTPS :

```
sudo apt-get install apt-transport-https
```

Ajoutez la clé de Tailscale :

```
curl -fsSL https://pkgs.tailscale.com/stable/raspbian/bullseye.noarmor.gpg |  
sudo tee /usr/share/keyrings/tailscale-archive-keyring.gpg > /dev/null
```

Ajoutez le référentiel Tailscale :

```
curl -fsSL https://pkgs.tailscale.com/stable/raspbian/bullseye.tailscale-  
keyring.list | sudo tee /etc/apt/sources.list.d/tailscale.list
```

Mettez à jour les paquets :

```
sudo apt-get update
```

Installez Tailscale :

```
sudo apt-get install tailscale
```

Installation de Tailscale sur Windows

Téléchargez Tailscale pour Windows depuis le site officiel. Installez Tailscale en suivant les instructions à l'écran.

Connexion à Tailscale et accès à la Raspberry Pi

Sur la Raspberry Pi, démarrez Tailscale et connectez-vous :

```
sudo tailscale up
```

Pour se déconnecter

```
sudo tailscale logout
```

Sur l'ordinateur Windows, télécharger et ouvrez l'application Tailscale et connectez-vous avec le même compte utilisé sur la Raspberry Pi. Surtout ne pas créer de compte avec le compte RT “rt-iut.re”

Une fois connecté, vous devriez voir la Raspberry Pi listée dans l'interface de Tailscale “admin console”. Maintenant on peut accéder à distance à la Raspberry Pi en utilisant son adresse IP fournie par Tailscale. Ensuite, on invitera les autres pour qu'il faire partie du même réseau VPN après avoir créé leur compte.

7/ Script Python pour température, humidité et led

On va créer un script python nommé script.py qui va centraliser la logique du projet pour afficher la température et humidité.

7.1/ Recensement des pin sortie/entre

Matériel	Numéro GPIO
Led Rouge	3
Led Vert	4
DHT22	17

7.2/ Installation de la librairie PAHO

```
~$ pip install paho-mqtt
```

7.3/ Création du script Python

7.3.1/ Importation des librairies

```
import adafruit_dht
import board
import ssl
import time
```

```
import paho.mqtt.client as mqtt
import lgpio
```

7.3.2/ Déclaration des variables

```
# Ouvrir la puce GPIO
gpio = lgpio.gpiochip_open(0)
lgpio gpio_claim_output(gpio, 4)
lgpio gpio_claim_output(gpio, 3)

# Sélection du DHT
dht_device = adafruit_dht.DHT22(board.D17)
```

7.3.3/ Configuration du client MQTT

```
# Configuration du client MQTT
topic = "/v1.6/devices/sae102/temperature"
TLS_CERT_PATH = "./FICHIER.pem" # Certificat MQTT

client = mqtt.Client()
client.username_pw_set("TON_TOKEN", "") # Token Ubidots

client.tls_set(
    ca_certs=TLS_CERT_PATH,
    certfile=None,
    keyfile=None,
    cert_reqs=ssl.CERT_REQUIRED,
    tls_version=ssl.PROTOCOL_TLSv1_2,
    ciphers=None
)
client.tls_insecure_set(False)
client.connect("industrial.api.ubidots.com", 8883)
print("Connecté à MQTT broker")
```

7.3.4/ Fonction pour allumer la led

```
def ALLUME_LED_(led, etat=0):
    if 'gpio' in globals():
```

```

if led == 'rouge':
    lgpio.gpio_write(gpio, 3, etat)
elif led == 'vert':
    lgpio.gpio_write(gpio, 4, etat)
else:
    print("La LED spécifiée n'est pas reconnue.")
else:
    print("Le module GPIO n'est pas initialisé.")
return led

```

7.3.5/ Envoyer les données

```

# Boucle principale pour envoyer les données
while True:
    try:
        temp = dht_device.temperature
        humidity = dht_device.humidity
        if temp is not None and humidity is not None:
            if temp >= 32.80:
                led_status = ALLUME_LED_('rouge', 1)
            else:
                led_status = ALLUME_LED_('vert', 1)

            msg = '{"value": %.2f, "context": {"temperature": %.2f, "humidity": %.2f, "led": "%s"}}' % (temp, temp, humidity, led_status)

            result = client.publish(topic, msg)
            client.loop()
    else:
        print("Échec de la récupération des données du capteur")
    except Exception as error:
        dht_device.exit()
        raise error
    time.sleep(10)

```