

# Test du capteur DHT22

## Le programme :

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import adafruit_dht

# Initial the dht device, with data pin connected to:
dhtDevice = adafruit_dht.DHT22(board.D18)

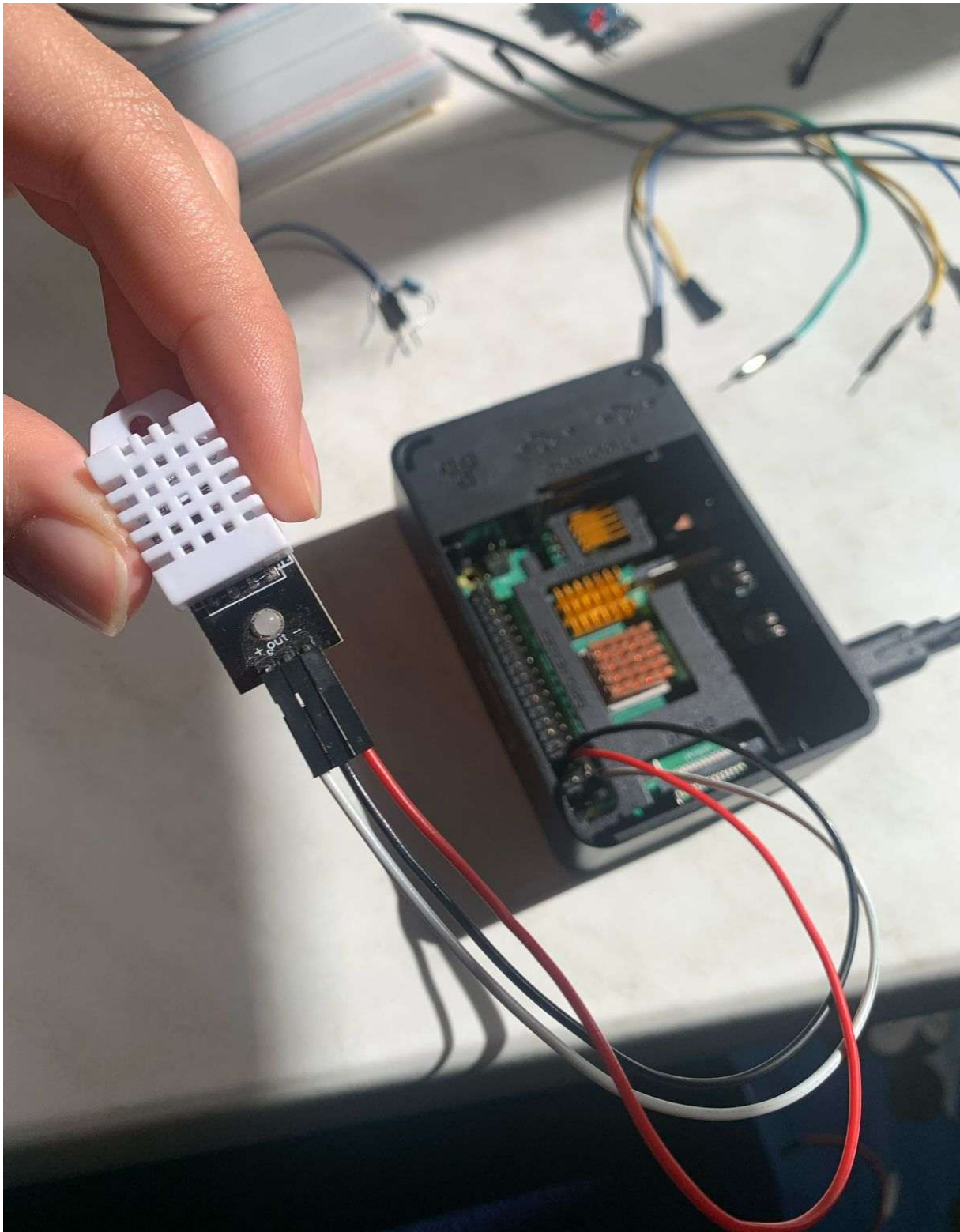
# you can pass DHT22 use_pulseio=False if you wouldn't like to use pulseio.
# This may be necessary on a Linux single board computer like the Raspberry Pi,
# but it will not work in CircuitPython.
# dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)

while True:
    try:
        # Print the values to the serial port
        temperature_c = dhtDevice.temperature
        temperature_f = temperature_c * (9 / 5) + 32
        humidity = dhtDevice.humidity
        print(
            "Temp: {:.1f} F / {:.1f} C   Humidity: {}% ".format(
                temperature_f, temperature_c, humidity
            )
        )

    except RuntimeError as error:
        # Errors happen fairly often, DHT's are hard to read, just keep going
        print(error.args[0])
        time.sleep(2.0)
        continue
    except Exception as error:
        dhtDevice.exit()
        raise error

    time.sleep(2.0)
```

### Le câblage :



- **VCC** du capteur → Broche 3.3V ou 5V du Raspberry Pi.
- **GND** du capteur → Broche GND du Raspberry Pi.
- **Data** du capteur → Broche GPIO4 .

## Les méthodes réaliser par le terminal :

### 1. Installation de la bibliothèque Adafruit pour le DHT22

- `sudo pip3 install adafruit-circuitpython-dht` #me donne erreur

**Objectif** : Installer la bibliothèque Adafruit pour interagir avec le capteur DHT22.

**Problème** : Rencontré une erreur ! Cela peut être dû à des conflits de versions ou à des restrictions de pip sur les systèmes modernes.

### 2. Installation alternative de la bibliothèque via apt

- `sudo apt install python3-adafruit-circuitpython-dht`

**Objectif** : Installer la bibliothèque Adafruit via le gestionnaire de paquets apt au lieu de pip.

**Avantage** : Cette méthode est souvent plus stable car elle utilise des paquets précompilés pour Raspberry Pi OS.

### 3. Exécution du script Python

- `python3 dht22_rpi4.py`

**Objectif** : Exécuter un script Python pour lire les données du capteur DHT22.

**Remarque** : Si le script n'existe pas encore, cette commande échouera. Vous devez d'abord créer le script.

### 4. Création ou modification du script Python

- `sudo nano dht_test.py`

**Objectif** : Ouvrir l'éditeur de texte nano pour créer ou modifier un script Python (dht\_test.py).

**Enregistrement** : Utilisez Ctrl + X, puis Y et Entrée pour sauvegarder et quitter.

### 5. Modification du script

- `sudo nano dht_test.py -l`

**Objectif** : Revenir au script pour le modifier.

**Modification** :

Décommenté la ligne `# dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)`.

Remplacé `board.D18` par `board.D4` pour spécifier que le capteur est connecté à la broche GPIO 4 du Raspberry Pi.

**Enregistrement** : Utilisez Ctrl + X, puis Y et Entrée pour sauvegarder.

### 6. Exécution du script modifié

- `sudo python3 dht_test.py`

**Objectif** : Exécuter le script modifié pour lire les données du capteur DHT22 connecté à la broche GPIO 4.

**Remarque** : si le câblage est correct et que le script est bien configuré, cette commande affichera la température et l'humidité

## 7. Mise à jour du système

- `sudo apt update`  
`pip install adafruit-blinka`

**Objectif** : Mettre à jour la liste des paquets disponibles sur votre système.

**Utilité** : Cela garantit que vous avez accès aux dernières versions des logiciels et des bibliothèques.

## 8. Installation forcée de la bibliothèque Adafruit Blinka

- `pip install adafruit-blinka adafruit-circuitpython-dht --break-system-packages`

**Objectif** : Installer la bibliothèque `adafruit-blinka` (nécessaire pour utiliser les bibliothèques Adafruit sur Raspberry Pi) et `adafruit-circuitpython-dht`.

**Problème** : Vous avez rencontré une erreur avec `pip install adafruit-blinka`, probablement due à des restrictions de sécurité sur les systèmes modernes.

**Solution** : L'option `--break-system-packages` force l'installation malgré ces restrictions.

## 9. Test final du script

- `python3 dht-test.py`

**Objectif** : Exécuter le script final pour tester le capteur DHT22.

**Arrêt** : Utilisez `Ctrl + C` pour arrêter l'exécution du script.

```
sifaoui@raspberrypi: ~  
File Edit Tabs Help  
sifaoui@raspberrypi:~$ sudo apt install python3-adafruit-circuitpython-dht  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
E: Unable to locate package python3-adafruit-circuitpython-dht  
sifaoui@raspberrypi:~$ python3 dht22_rpi4.py  
python3: can't open file '/home/sifaoui/dht22_rpi4.py': [Errno 2] No such file or directory  
sifaoui@raspberrypi:~$ sudo nano dht_test.py  
sifaoui@raspberrypi:~$ sudo nano dht_test.py - l  
Reading data from keyboard; type ^D or ^D^D to finish.  
sifaoui@raspberrypi:~$ sudo python3 dht-test.py  
Traceback (most recent call last):  
  File "/home/sifaoui/dht-test.py", line 2, in <module>  
    import board  
ModuleNotFoundError: No module named 'board'  
sifaoui@raspberrypi:~$ sudo apt update  
pip install adafruit-blinka  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Get:4 http://archive.raspberrypi.com/debian bookworm InRelease [39.3 kB]  
Get:5 http://archive.raspberrypi.com/debian bookworm/main armhf Packages [545 kB]  
Get:6 http://archive.raspberrypi.com/debian bookworm/main arm64 Packages [517 kB]  
Fetched 1,101 kB in 2s (518 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
5 packages can be upgraded. Run 'apt list --upgradable' to see them.  
error: externally-managed-environment  
  
* This environment is externally managed  
  ↳ To install Python packages system-wide, try apt install  
    python3-xyz, where xyz is the package you are trying to  
    install.  
  
    If you wish to install a non-Debian-packaged Python package,  
    create a virtual environment using python3 -m venv path/to/venv.  
    Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make  
    sure you have python3-full installed.  
  
    For more information visit http://rptl.io/venv  
  
note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You  
can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages  
  
hint: See PEP 668 for the detailed specification.  
sifaoui@raspberrypi:~$ pip install adafruit-blinka adafruit-circuitpython-dht --break-system-packages  
  
sifaoui@raspberrypi:~$ pip install adafruit-blinka adafruit-circuitpython-dht --break-system-packages  
Defaulting to user installation because normal site-packages is not writeable  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: adafruit-blinka in ./local/lib/python3.11/site-packages (8.55.0)  
Requirement already satisfied: adafruit-circuitpython-dht in ./local/lib/python3.11/site-packages (4.0.7)  
Requirement already satisfied: Adafruit-PlatformDetect>=3.70.1 in ./local/lib/python3.11/site-packages (from adafruit-blinka) (3.77.0)  
Requirement already satisfied: Adafruit-PureIO>=1.1.7 in ./local/lib/python3.11/site-packages (from adafruit-blinka) (1.1.11)  
Requirement already satisfied: binho-host-adapter>=0.1.6 in ./local/lib/python3.11/site-packages (from adafruit-blinka) (0.1.6)  
Requirement already satisfied: pyftdi>=0.40.0 in ./local/lib/python3.11/site-packages (from adafruit-blinka) (0.56.0)  
Requirement already satisfied: adafruit-circuitpython-typing in ./local/lib/python3.11/site-packages (from adafruit-blinka) (1.11.2)  
Requirement already satisfied: RPi.GPIO in ./local/lib/python3.11/site-packages (from adafruit-blinka) (0.7.1)  
Requirement already satisfied: rpi-ws281x>=4.0.0 in ./local/lib/python3.11/site-packages (from adafruit-blinka) (5.0.0)  
Requirement already satisfied: sysv-ipc>=1.1.0 in ./local/lib/python3.11/site-packages (from adafruit-blinka) (1.1.0)  
Requirement already satisfied: pyserial in /usr/lib/python3/dist-packages (from binho-host-adapter>=0.1.6->adafruit-blinka) (3.5)  
Requirement already satisfied: pyusb!=1.2.0,>=1.0.0 in ./local/lib/python3.11/site-packages (from pyftdi>=0.40.0->adafruit-blinka) (1.3.1)  
Requirement already satisfied: adafruit-circuitpython-busdevice in ./local/lib/python3.11/site-packages (from adafruit-circuitpython-typing->adafruit-blinka) (5.2.11)  
Requirement already satisfied: adafruit-circuitpython-requests in ./local/lib/python3.11/site-packages (from adafruit-circuitpython-typing->adafruit-blinka) (4.1.10)  
Requirement already satisfied: typing_extensions~=4.0 in /usr/lib/python3/dist-packages (from adafruit-circuitpython-typing->adafruit-blinka) (4.4.0)  
Requirement already satisfied: Adafruit-Circuitpython-ConnectionManager in ./local/lib/python3.11/site-packages (from adafruit-circuitpython-requests->adafruit-circuitpython-typing->adafruit-blinka) (3.1.3)  
sifaoui@raspberrypi:~$ python3 dht-test.py  
Temp: 76.1 F / 24.5 C Humidity: 64.5%  
Temp: 76.1 F / 24.5 C Humidity: 64.3%  
Temp: 76.1 F / 24.5 C Humidity: 64.2%  
Temp: 76.1 F / 24.5 C Humidity: 64.3%  
Temp: 75.9 F / 24.4 C Humidity: 64.4%  
Temp: 75.9 F / 24.4 C Humidity: 64.4%  
Temp: 75.9 F / 24.4 C Humidity: 64.5%  
Temp: 75.9 F / 24.4 C Humidity: 64.3%  
Checksum did not validate. Try again.  
Temp: 75.9 F / 24.4 C Humidity: 64.5%
```

### Dégradation de la température :

[illegible]



## Affichage de données sur un écran LCD

### Le programme:

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import adafruit_dht
import drivers

# Initial the dht device, with data pin connected to:
dhtDevice = adafruit_dht.DHT22(board.D4)
display = drivers.Lcd()

# you can pass DHT22 use_pulseio=False if you wouldn't like to use pulseio.
# This may be necessary on a Linux single board computer like the Raspberry Pi,
# but it will not work in CircuitPython.
# dhtDevice = adafruit_dht.DHT22(board.D4, use_pulseio=False)

try:
    while True:
        try:
            # Print the values to the serial port
            temperature_c = dhtDevice.temperature
            temperature_f = temperature_c * (9 / 5) + 32
            humidity = dhtDevice.humidity
            print(
                "Temp: {:.1f} F / {:.1f} C  Humidity: {}% ".format(
                    temperature_f, temperature_c, humidity
                )
            )
        except RuntimeError:
            pass
```

```

    # Display the values on the LCD

    display lcd_display_string("Temp: {:.1f}C".format(temperature_c), 1) # Write line of
text to first line of display

    display lcd_display_string("Humidity: {}%".format(humidity), 2) # Write line of text
to second line of display

    time.sleep(2) # Give time for the message to be read


except RuntimeError as error:
    # Errors happen fairly often, DHT's are hard to read, just keep going
    print(error.args[0])
    time.sleep(2.0)
    continue

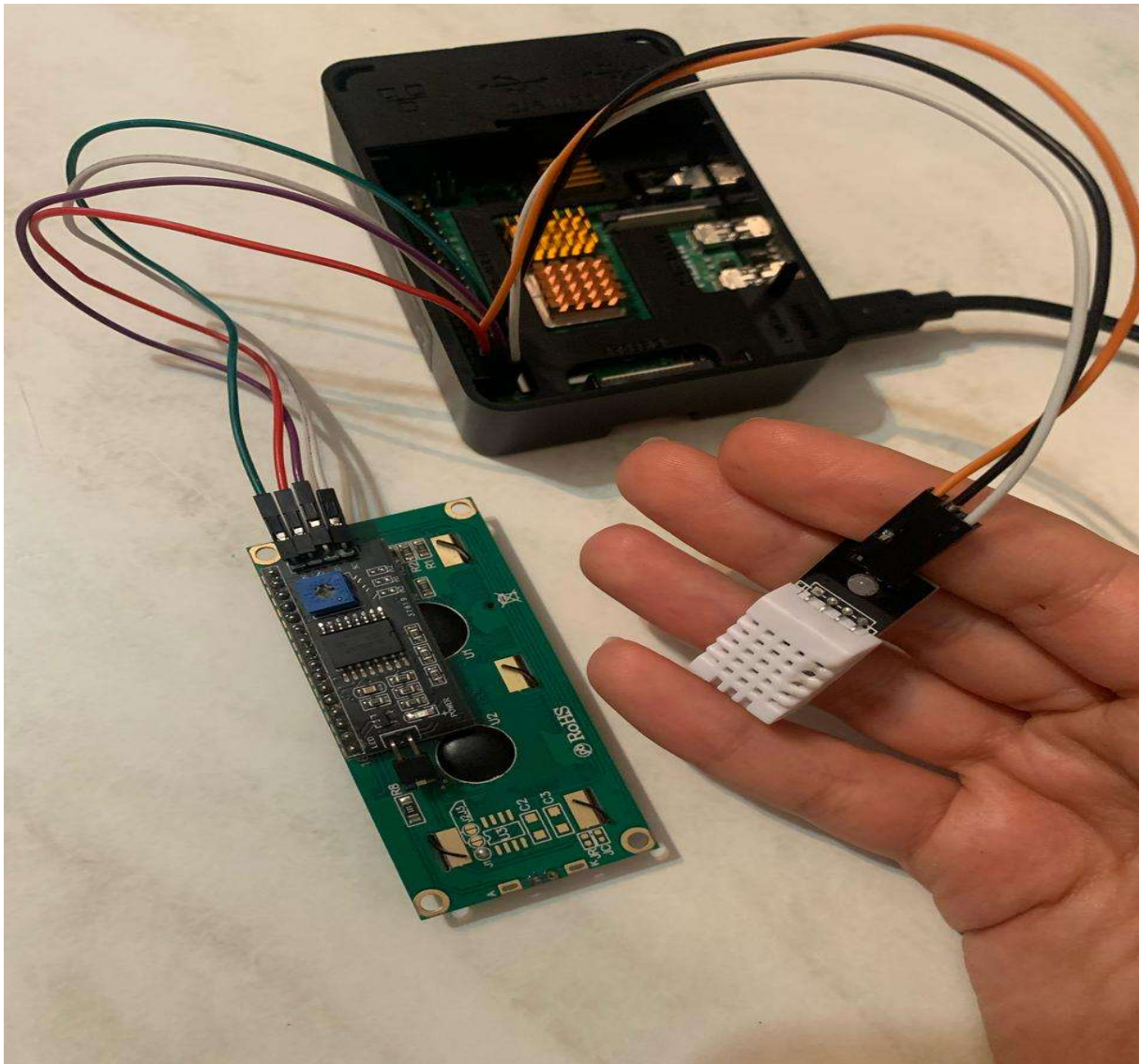
except Exception as error:
    dhtDevice.exit()
    raise error


except KeyboardInterrupt:
    # If there is a KeyboardInterrupt (when you press ctrl+c), exit the program and cleanup

```



## Le câblage:



VCC → 5V (Pin 2)

GND → GND (Pin 6)

SDA → GPIO (SDA, Pin 3)

SCL → GPIO3 (SCL, Pin 5)

## Les méthodes réaliser :

### 1. Configuration de l'I2C

#### Étapes :

1. Ouvrir Raspberry Pi Configuration :
2. Entrer sur Interface :
3. Cocher I2C :

**Objectif :** Activer le protocole I2C sur le Raspberry Pi pour permettre la communication avec l'écran LCD via I2C.

**Problème résolu :** Sans activation de l'I2C, le Raspberry Pi ne peut pas communiquer avec l'écran LCD.

**Résultat attendu :** L'I2C est activé, et le Raspberry Pi est prêt à communiquer avec des périphériques I2C comme l'écran LCD.

### 2. Installation des bibliothèques

#### a. Installation de RPi-GPIO-I2C-LCD

- `sudo pip3 install RPi-GPIO-I2C-LCD`

Si cela échoue, forcez l'installation avec :

- `sudo pip3 install --break-system-packages RPi-GPIO-I2C-LCD`

**Objectif :** Installer la bibliothèque RPi-GPIO-I2C-LCD pour contrôler l'écran LCD via I2C.

**Problème résolu :** Sans cette bibliothèque, vous ne pouvez pas envoyer de commandes à l'écran LCD depuis Python.

**Résultat attendu :** La bibliothèque est installée, et vous pouvez l'importer dans les scripts Python pour contrôler l'écran LCD.

#### b. Clonage du dépôt lcd

- `git clone https://github.com/the-raspberry-pi-guy/lcd`

**Objectif :** Télécharger des scripts et des exemples pour contrôler un écran LCD.

**Problème résolu :** Sans ces scripts, vous devriez écrire tout le code de zéro, ce qui est plus long et plus complexe.

**Résultat attendu :** Un dossier nommé lcd est créé, contenant des scripts prêts à l'emploi pour contrôler l'écran LCD.

### 3. Accéder au dossier lcd

- `cd lcd`

**Objectif :** Se placer dans le dossier lcd où se trouvent les fichiers clonés.

**Problème résolu :** Sans être dans le bon répertoire, vous ne pouvez pas accéder aux fichiers ou exécuter les scripts.

**Résultat attendu :** Vous êtes maintenant dans le dossier lcd, prêt à créer ou exécuter des scripts.

### 4. Créer un script

- `nano demo_lcd.py`

**Objectif :** Créer un fichier Python nommé `demo_lcd.py` pour écrire le programme qui contrôle l'écran LCD.

**Problème résolu :** Sans ce fichier, vous ne pouvez pas écrire ou exécuter de code pour l'écran LCD.

**Résultat attendu :** Un fichier `demo_lcd.py` est créé, prêt à être édité avec le code Python.

### 5. Écrire le programme

**Objectif :** Écrire le code Python dans `demo_lcd.py` pour contrôler l'écran LCD.

**Problème résolu :** Sans ce code, l'écran LCD ne peut pas afficher de texte ou de données.

**Résultat attendu :** Le fichier `demo_lcd.py` contient le code nécessaire pour afficher des informations sur l'écran LCD.

### 6. Exécuter le script d'installation

- `sudo sh install.sh`

**Objectif :** Exécuter le script `install.sh` pour installer les dépendances nécessaires.

**Problème résolu :** Sans ces dépendances, le programme ne peut pas fonctionner correctement.

**Résultat attendu :** Toutes les dépendances sont installées, et l'environnement est configuré pour utiliser l'écran LCD.

### 7. Exécuter le programme

- `python demo_lcd.py`

**Objectif :** Exécuter le programme Python pour afficher des informations sur l'écran LCD.

**Problème résolu :** Sans exécuter le programme, l'écran LCD reste vide.

**Résultat attendu :** L'écran LCD affiche les informations définies dans le programme

```
sifaoui@raspberrypi: ~/lcd
File Edit Tabs Help
[1/4] demo_lcd.
import time
import board
import adafruit_dht
import drivers

# Initial the dht device, with data pin connected to:
dhtDevice = adafruit_dht.DHT22(board.D4)
display = drivers.Lcd()

# you can pass DHT22 use_pulseio=False if you wouldn't like to use pulseio.
# This may be necessary on a Linux single board computer like the Raspberry Pi,
# but it will not work in CircuitPython.
# dhtDevice = adafruit_dht.DHT22(board.D4, use_pulseio=False)

try:
    while True:
        try:
            # Print the values to the serial port
            temperature_c = dhtDevice.temperature
            temperature_f = temperature_c * (9 / 5) + 32
            humidity = dhtDevice.humidity
            print(
                "Temp: {:.1f} F / {:.1f} C Humidity: {}% ".format(
                    temperature_f, temperature_c, humidity
                )
            )

            # Display the values on the LCD
            display lcd_display_string("Temp: {:.1f}C".format(temperature_c), 1) # Write line of text to first line of display
            display lcd_display_string("Humidity: {}%".format(humidity), 2) # Write line of text to second line of display
            time.sleep(2) # Give time for the message to be read

        except RuntimeError as error:
            # Errors happen fairly often, DHT's are hard to read, just keep going
            print(error.args[0])
            time.sleep(2.0)
            continue
        except Exception as error:
            dhtDevice.exit()
            raise error

except KeyboardInterrupt:
    # If there is a KeyboardInterrupt (when you press ctrl+c), exit the program and cleanup
    print("Cleaning up!")
    display lcd_clear()
```

AG Help AR Write Out AW Where Is AK Cut AT Execute AC Location M-U Undo M-A Set Mark  
AX Close AR Read File AN Replace AU Paste AJ Justify AJ Go To Line M-E Redo M-G Copy

```
Thonny - /home/sifaoui/lcd/demo.UTF-8 @ 48 : 24
File Edit View Run Tools Help
demo.UTF-8
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 import time
5 import board
6 import adafruit_dht
7 import drivers
8
9 # Initial the dht device, with data pin connected to:
10 dhtDevice = adafruit_dht.DHT22(board.D4)
11 display = drivers.Lcd()
12
13 # you can pass DHT22 use_pulseio=False if you wouldn't like to use pulseio.
14 # This may be necessary on a Linux single board computer like the Raspberry Pi,
15 # but it will not work in CircuitPython.
16 # dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)
17

Shell
Temp: 73.9 F / 23.3 C Humidity: 67.8%
Temp: 74.1 F / 23.4 C Humidity: 67.9%
Temp: 73.9 F / 23.3 C Humidity: 67.8%
Temp: 74.1 F / 23.4 C Humidity: 67.8%
Temp: 73.9 F / 23.3 C Humidity: 67.9%
Temp: 74.1 F / 23.4 C Humidity: 67.9%
Temp: 73.9 F / 23.3 C Humidity: 68.0%
Temp: 73.9 F / 23.3 C Humidity: 67.9%

Assistant
IndentationError:
expected an indented
block after 'except'
statement on line 45
demo.UTF-8, line 46
No specific suggestions
for this error (yet).
Let Thonny developers
know
Search the web
Warnings
May help you find the cause of
the error.
demo.UTF-8
Line 46: expected an
indented block after
'except' statement on
line 45 [syntax]
Was it helpful or
confusing?
General advice on
dealing with errors.
Local Python 3 • /usr/bin/python3
```

