

Étapes

Étapes à suivre pour refaire le projet (conception, construction, réalisation, manipulation...)

- Étape 1 : Se connecter à la Raspberry Pi via SSH
- Étape 2 : Hardware x Software
- Étape 3 : Website, Service, Entretien
- Étape 4 : Assemblage

Étape 1 : Se connecter à la Raspberry Pi via SSH

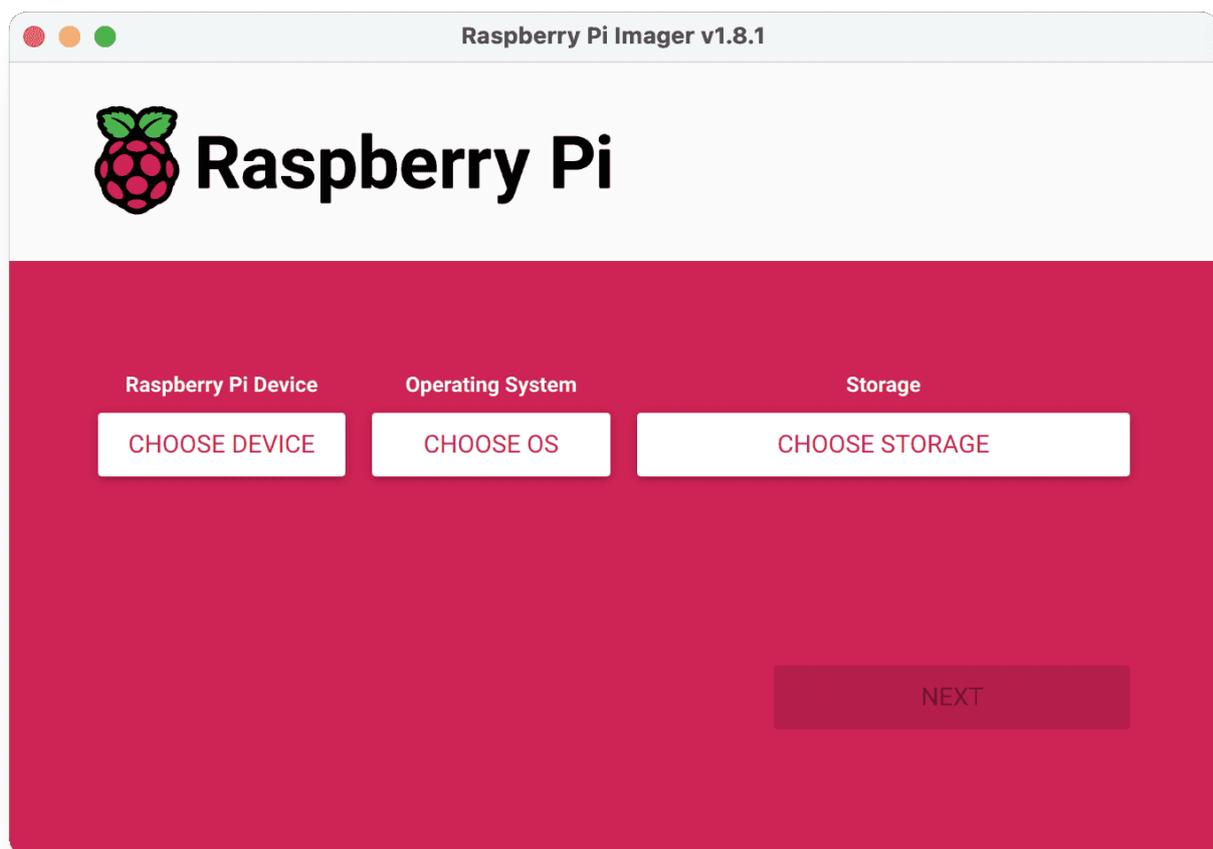
Nous voulions une configuration minimale sans dépendre de périphériques extérieurs pour développer le système. C'est pourquoi nous avons choisi un mode sans écran (*Headless Mode*). Nous voulions que cette approche aide les e-makers à répliquer le système sans dépendre de ces composants (périphériques), qui peuvent poser problème s'ils ne sont pas disponibles (comme dans notre cas, un câble mini HDMI vers HDMI), afin de leur faciliter le passage à un environnement non-Desktop (sans Bureau) s'ils le souhaitent. Le hub USB peut même être supprimé et la caméra peut être connectée au port micro USB du Raspberry Pi avec un connecteur USB vers micro USB.

Pour ce faire, nous nous sommes connectés à la Raspberry Pi, un *nano-ordinateur*, via SSH. SSH est un protocole qui permet d'envoyer des commandes à un ordinateur sur un réseau non sécurisé (comme le réseau public de la fac) de façon sécurisée. Cependant, cela ne suffisait pas pour la phase de développement. Nous avons rapidement constaté qu'un environnement de Bureau était très utile pour diverses tâches (telles que le scripting, le transfert ou encore la gestion des fichiers). Nous avons pu utiliser pleinement la version de bureau en nous connectant au Raspberry Pi (Pi) via VNC, est un système de partage de bureau graphique qui permet de contrôler un ordinateur à distance, projetant ainsi la sortie HDMI sur notre écran sans fil. Bien que cela ne soit pas nécessaire pour la réplication ou la maintenance du système, cela s'est avéré très utile pendant le développement.

1.1 : Préparation de la carte SD

- Télécharger le Raspberry Pi Imager :
 - Visitez le site web de Raspberry Pi et téléchargez le **Raspberry Pi Imager** pour Windows.
 - Installez et lancez le logiciel.
- Insérer la carte SD:
 - Insérez votre carte micro SD dans le lecteur de cartes SD et connectez-le à votre PC .
- Écrire le système d'exploitation :
 - Dans le Raspberry Pi Imager, cliquez sur "*CHOOSE OS*" et sélectionnez "*Raspberry Pi OS (64-bit)*" ou une autre version selon vos besoins.
 - Cliquez sur "*CHOOSE SD CARD*" et sélectionnez votre carte micro SD.
 - Cliquez sur "*WRITE*" pour écrire le système d'exploitation sur la carte SD. Cela prendra quelques minutes.

Attention, cela va effacer toutes les données de la carte SD. Veuillez vérifier qu'elle ne contient rien d'important avant de continuer.



1.2 : Configurer le Pi pour un fonctionnement sans écran

- Activer SSH :
 - Après la fin du processus d'écriture, vous devez activer SSH pour permettre les connexions à distance. Éjectez et réinsérez la carte SD dans votre PC.
 - Ouvrez la carte SD dans l'Explorateur de fichiers et créez un fichier vide nommé 'ssh' (sans extension) à la racine de la carte SD. Ce fichier active SSH au démarrage.
- Configurer le Wi-Fi:
 - Créez un fichier nommé *wpa_supplicant.conf* à la racine de la carte SD.
 - Ouvrez ce fichier dans un éditeur de texte de votre choix et ajoutez le contenu suivant, en remplaçant "YOUR_SSID" et "YOUR_PASSWORD" par le SSID et le mot de passe de votre réseau Wi-Fi :

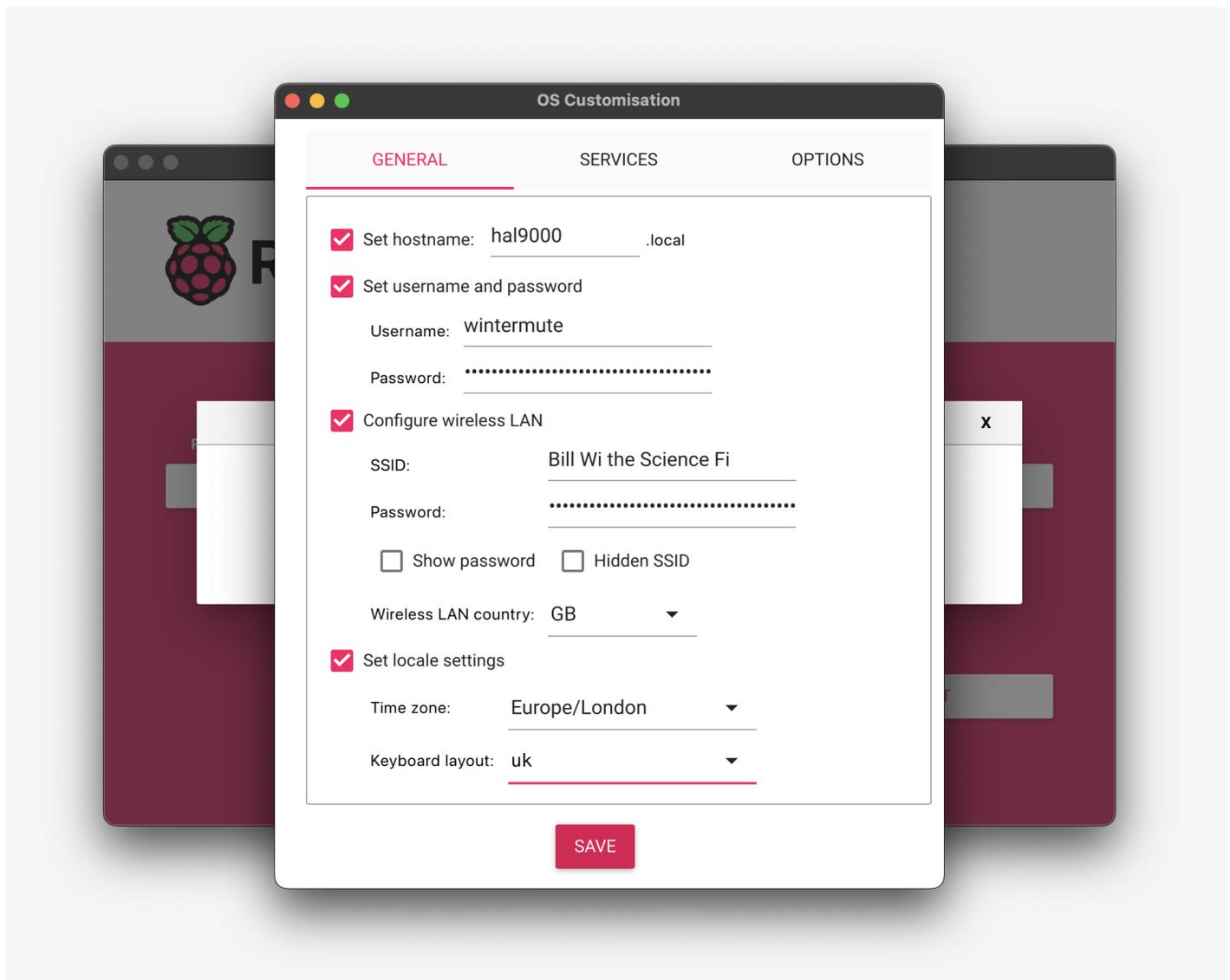
```
country=FR
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="YOUR_SSID"
```

```
psk="YOUR_PASSWORD"
key_mgmt=WPA-PSK
}
```

- Enregistrez et fermez le fichier.

Attention, la Raspberry Pi ne peut se connecter qu'au Wi-Fi de 2.4GHz.

Astuce : Vous pouvez passer ces étapes et faire tout cela à travers Raspberry Pi Imager. Dans le menu de personnalisation générale de l'OS, avant de flasher l'OS, une option apparaît vous demandant si vous voulez personnaliser l'OS. Sélectionnez "Edit Settings" et vous y êtes.



Il vous est conseillé de ne pas changer le *hostname* qui par défaut est "raspberrypi". Changez votre *username* et *password* à quelque chose de simple, par exemple "pi" et "pi". Remplissez le reste des informations restantes sur la page. Ensuite, naviguez vers "SERVICES" et activez SSH. Cochez "Enable SSH" et sélectionnez "Use password authentication" (normalement sélectionné automatiquement lorsque vous cochez SSH).

1.3 : Démarrer le Raspberry Pi

- Insérer la carte SD dans le Raspberry Pi :
 - Éjectez en toute sécurité la carte SD de votre PC et insérez-la dans le Raspberry Pi Zero 2W.
- Mettre sous tension le Raspberry Pi :
 - Connectez le Raspberry Pi au hub USB ;
 - Connectez le Raspberry Pi à une source d'alimentation à l'aide d'un câble micro USB.

1.4 : Se connecter au Raspberry Pi depuis votre ordinateur

- Trouver l'adresse IP du Raspberry Pi :
 - Utilisez l'interface web de votre routeur pour trouver l'adresse IP du Raspberry Pi, ou utilisez un outil de balayage réseau comme Advanced IP Scanner.

Dans notre cas, nous avons partagé la connexion Internet depuis notre téléphone portable (iOS) pour connecter le Raspberry Pi à Internet. Malheureusement, il n'y a pas moyen de voir les adresses IP des appareils connectés à votre connexion sur iOS. Cependant, après un peu de recherche, nous avons découvert que les adresses IP des appareils connectés se situent entre 172.20.10.0 et 172.20.10.15.

iPhone hotspot : <https://charsiurice.wordpress.com/2021/02/08/iphone-hotspot-subnet-info/>

- Se connecter via SSH :
 - Ouvrez un terminal sur votre PC ;
 - Utilisez la commande SSH pour vous connecter à votre Raspberry Pi. Remplacez "pi" par votre nom d'utilisateur (par défaut, c'est "pi"), et "x.x.x.x" par l'adresse IP de votre Raspberry Pi :

```
ping x.x.x.x #pour verifier si votre pi est connecter , utile pour scannez rapidement et trouver sur quelle adress ip votre pi ce situe
```

```
ssh pi@x.x.x.x
```

- Lorsque vous y êtes invité(e), entrez le mot de passe (*raspberrypi* si vous ne l'avez pas modifié).

1.5 : Configuration finale

- Mise à jour et mise à niveau :
 - Une fois connecté(e), mettez à jour le logiciel de votre Raspberry Pi en utilisant la commande suivante :

```
sudo apt update && sudo apt full-upgrade -y
```

Si vous utilisez une version d'OS avec environnement de bureau, vous pourriez rencontrer des problèmes, notamment sur le Pi Zero 2W qui dispose de 512 Mo de mémoire et peut ne pas avoir suffisamment d'espace swap pour effectuer la mise à jour.

- À partir d'une fenêtre Terminal, entrez la commande "swapon". La réponse vous donnera le nom du fichier, sa taille et l'utilisation du fichier swap.
- Par défaut, la taille du fichier swap est de 100 Mo, ce qui pourrait ne pas être suffisant avec 512 Mo de mémoire.
- Modifiez le fichier `/etc/dphys-swapfile` en changeant `CONF_SWAPSIZE=100` à `CONF_SWAPSIZE=1024`. Redémarrez, puis avant de faire quoi que ce soit, entrez la commande "swapon" dans une fenêtre Terminal pour vérifier l'utilisation de l'espace swap.

```
sudo nano /etc/dphys-swapfile
```

Plus d'informations sont disponibles aux liens suivants :

<https://forums.raspberrypi.com/viewtopic.php?t=359240>

<https://github.com/RPi-Distro/repo/issues/353>

Informations additionnelles et d'autres problèmes rencontrés avec leurs solutions

Vous pouvez trouver des informations additionnelles a propos du choix de la carte dans le rapport complet section 4.1 : The board.

En cas de complications, veuillez consulter les liens suivants :

Documentation Raspberry Pi : <https://www.raspberrypi.com/documentation/computers/getting-started.html>

Raspberry Pi Zero 2W First time minimal setup : <https://youtu.be/3eGP00uXcUU?feature=shared>

Pour ce connecter en VNC : <https://youtu.be/yn59qX-Td3E?feature=shared>

Différence entre upgarde et full upgrade :

<https://forums.raspberrypi.com/viewtopic.php?t=264816&sid=ce4fb96eab34f88644b77686bf9472c>

Vous pouvez facilement transférer des fichiers entre votre machine et la Pi à travers VNC mais cela nécessite un OS avec version bureautique.

Pour des raisons de sécurité, changez le mot de passe par défaut :

```
passwd
```

Pour vérifiez l'utilisation des ressources système :

```
top
```

Manipulation des périphériques USB via le terminal

Trouver le périphérique USB :

1. Utilisez la commande *lsblk* pour lister tous les périphériques de bloc.
2. Identifiez le périphérique USB dans la sortie, généralement listé sous *'/dev/sdX'* ou *'/sdX'* où X est une lettre correspondant au périphérique.

```
pi@raspberrypi:~ $ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda         8:0  1  7.5G  0 disk
└─sda1      8:1  1  7.5G  0 part /media/pi/8097-5224
mmcblk0    179:0  0  29G  0 disk
├─mmcblk0p1 179:1  0  512M  0 part /boot/firmware
└─mmcblk0p2 179:2  0  28.5G  0 part /
pi@raspberrypi:~ $
```

Monter le périphérique USB :

1. Créez un répertoire point de montage à l'aide de *'mkdir'*, par exemple :

```
mkdir ~/usb
```

2. Montez le périphérique USB sur le répertoire point de montage à l'aide de

```
sudo mount /dev/sdXN ~/usb
```

en remplaçant *'/dev/sdXN'* par le fichier de périphérique réel.

Accéder au périphérique USB :

1. Changez votre répertoire courant vers le répertoire point de montage à l'aide de 'cd', par exemple :

```
cd ~/usb
```

Copier des fichiers :

1. Utilisez la commande 'cp' pour copier des fichiers depuis votre système vers le périphérique USB, par exemple :

```
cp /chemin/vers/fichier/source ~/usb
```

Démonter le périphérique USB :

1. Assurez-vous de ne pas être dans le répertoire point de montage du périphérique USB.
2. Démontez le périphérique USB à l'aide de

```
sudo umount ~/usb
```

Nettoyage (optionnel) :

1. Supprimez le répertoire point de montage s'il n'est plus nécessaire à l'aide de 'rmdir', par exemple :

```
rmdir ~/usb
```

N'oubliez pas de remplacer les espaces réservés comme '/dev/sdXN' par le fichier de périphérique réel et d'ajuster les chemins et les commandes selon votre configuration système.

Une fois ces étapes suivies, vous devriez pouvoir gérer efficacement les clés USB via le terminal. En cas de problème ou si vous avez d'autres questions, n'hésitez pas à nous contacter !

Copier des fichiers vers le périphérique USB :

Utilisez la commande 'cp' suivie du fichier source ou du répertoire et du répertoire de destination (le périphérique USB monté).

Pour un seul fichier :

```
cp chemin/vers/le/fichier /chemin/vers/usb/monté
```

Pour plusieurs fichiers ou tous les fichiers dans un répertoire :

```
cp -r * /chemin/vers/usb/monté
```

Problème :

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
```

Lors de la tentative de connexion SSH à l'adresse IP 172.20.10.4 depuis un ordinateur Windows, un avertissement est affiché indiquant que l'identification de l'hôte distant a changé. Cela est dû au fait que la clé d'hôte de votre Raspberry Pi a été modifiée, ce qui peut se produire lorsqu'une nouvelle carte SD est utilisée ou lors d'une réinstallation du système.

Solution :

Pour résoudre ce problème, vous devez supprimer l'ancienne entrée de clé d'hôte de votre fichier 'known_hosts' sur votre PC Windows. Voici comment procéder :

1. Ouvrez **PowerShell** sur votre PC Windows.
2. Supprimez l'ancienne clé dans le fichier 'known_hosts' en utilisant la commande suivante dans PowerShell pour supprimer la clé d'hôte offensante :

```
ssh-keygen -R 172.20.10.4
```

Cette commande supprime l'ancienne clé pour l'adresse IP spécifiée (ici, '172.20.10.4') du fichier 'known_hosts'.

3. Vérifiez la suppression de l'entrée :
Vous pouvez également ouvrir manuellement le fichier 'C:\Users\admin\.ssh\known_hosts' dans Notepad et supprimer la ligne correspondant à '172.20.10.4'. Enregistrez et fermez le fichier après la suppression.

4. Connectez-vous à nouveau :
Après avoir supprimé l'ancienne entrée de clé d'hôte, essayez de vous connecter à nouveau à votre Raspberry Pi avec la commande SSH :

```
ssh pi@172.20.10.4
```

Vous devriez maintenant être invité(é) à accepter la nouvelle clé d'hôte. Voici un exemple de ce à quoi cela ressemble généralement :

```
...
```

```
The authenticity of host '172.20.10.4 (172.20.10.4)' can't be established.
```

```
ECDSA key fingerprint is SHA256:V+qB6UChyimRn+LquGUGpyYMDRyrfYmC4EAJ6Z3E3pl.
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

```
...
```

5. Acceptez la nouvelle clé d'hôte :

Cliquez sur 'yes' et appuyez sur Entrée pour ajouter la nouvelle clé d'hôte à votre fichier 'known_hosts'. Vous devriez maintenant pouvoir vous connecter sans problème à votre Raspberry Pi !

Cette procédure permet de résoudre le problème de vérification de la clé d'hôte modifiée lors de la connexion SSH à votre Raspberry Pi depuis votre PC Windows.

FIN

Étape 2 : Hardware x Software

La deuxième étape dans la fabrication du système consiste à mettre en place le système de capture. Dans cette section, nous allons créer un environnement virtuel, installer les bibliothèques nécessaires pour le travail, et d'autres étapes additionnelles pour la configuration des composants.

Pour se cultiver

pip est utilisé pour télécharger et installer des paquets directement depuis PyPI (Python Package Index), hébergé par la Python Software Foundation. Il s'agit d'un gestionnaire de paquets spécialisé exclusivement pour les paquets Python.

apt-get, en revanche, est utilisé pour télécharger et installer des paquets depuis les dépôts Ubuntu hébergés par Canonical.

Voici quelques différences clés entre l'installation des paquets Python avec apt-get et pip :

1. Disponibilité des Paquets: Canonical fournit des paquets pour un ensemble sélectionné de modules Python. En revanche, PyPI héberge une gamme beaucoup plus large de modules Python. Par conséquent, de nombreux modules Python ne peuvent pas être installés via apt-get mais sont disponibles sur PyPI.
2. Contrôle des Versions : Canonical héberge généralement une seule version de chaque paquet (généralement la plus récente ou celle sortie récemment). Avec apt-get, vous ne pouvez pas choisir quelle version d'un paquet Python installer. En revanche, pip vous permet d'installer n'importe quelle version d'un paquet précédemment téléchargé sur PyPI. Cette flexibilité est essentielle pour gérer les conflits de dépendances.
3. Portée de l'Installation : apt-get installe les modules Python de manière globale sur le système, ce qui signifie qu'ils sont disponibles globalement. Cela peut entraîner des conflits si différents projets nécessitent différentes versions du même paquet. pip résout ce problème en installant les paquets dans l'environnement virtuel du projet (virtualenv). Cette isolation garantit que les dépendances sont gérées indépendamment pour chaque projet.
4. Conventions de Nom de Paquet : Canonical nomme généralement les paquets Python 2 comme `python-<nom_du_paquet>` et les paquets Python 3 comme `python3-<nom_du_paquet>`. En revanche, pip simplifie l'installation des paquets en utilisant `<nom_du_paquet>` pour les paquets Python 2 et Python 3.

Choix entre apt-get et pip:

apt-get et pip sont tous deux des gestionnaires de paquets matures qui gèrent la résolution des dépendances lors de l'installation. Vous pouvez utiliser l'un ou l'autre en fonction de vos préférences. Cependant, envisagez d'utiliser **pip** dans les scénarios suivants :

- Lorsque vous avez besoin d'installer une version spécifique d'un paquet Python.
- Lorsque vous souhaitez installer des paquets dans l'environnement virtuel d'un projet.
- Lorsque le paquet dont vous avez besoin n'est disponible que sur PyPI.

Si vous êtes à l'aise avec l'installation de paquets globalement et qu'il n'y a pas de conflits de versions, apt-get ou pip conviendront pour l'installation générale de paquets Python.

C'est quoi un Environment Virtuelle : <https://www.geeksforgeeks.org/python-virtual-environment/>

Installation d'un Environment Virtuelle

On utilisera **pip** depuis un environnement virtuel pour installer les bibliothèques.

Une fois connecté à votre Raspberry Pi via SSH:

1 : Naviguer vers votre répertoire de projet

Accédez au répertoire où vous souhaitez créer votre environnement virtuel, ex :

```
/home/pi/Desktop/scripts
```

2 : Créer l'environnement virtuel

Utilisez la commande 'python3 -m venv venv' pour créer un environnement virtuel. Remplacez 'venv' par le nom que vous souhaitez donner à votre environnement virtuel.

```
python3 -m venv venv
```

3 : Activer l'environnement virtuel

```
source venv/bin/activate
```

4 : Désactiver l'environnement virtuel

Vous ne pourriez pas lancer le code si environnement virtuelle est désactiver

```
deactivate
```

5 : Installer les modules requis

Si `pip` n'est pas déjà installé sur votre système, vous pouvez l'installer en utilisant la commande adaptée à votre distribution Linux. Par exemple

```
sudo apt-get install python3-pip
```

puis

```
pip install adafruit-circuitpython-rgb-display
```

```
pip install adafruit-blinka busio digitalio
```

```
pip install opencv-python
```

```
pip install pillow
```

Vérifier les bibliothèques installées

```
(venv) pi@raspberrypi:~/Desktop/scripts $ pip list
```

Package	Version
-----	-----
Adafruit-Blinka	8.45.0
adafruit-circuitpython-busdevice	5.2.9
adafruit-circuitpython-connectionmanager	3.1.1
adafruit-circuitpython-requests	4.1.1
adafruit-circuitpython-rgb-display	3.12.4
adafruit-circuitpython-typing	1.10.3
Adafruit-PlatformDetect	3.71.0
Adafruit-PureIO	1.1.11
binho-host-adapter	0.1.6
numpy	2.0.0
opencv-python	4.10.0.84
pillow	10.3.0
pip	23.0.1

pyftdi	0.55.4
pyserial	3.5
pyusb	1.2.1
RPi.GPIO	0.7.1
rpi-ws281x	5.0.0
setuptools	66.1.1
sysv-ipc	1.1.0
typing_extensions	4.12.2

Le code réel

Vous pouvez trouver une documentation de chaque classe et les références à la fin de cette page.

Vous pouvez installer les fichiers depuis le Git `'/scripts/ras_pi_zero2W/os5'` ou bien depuis la section Fichier sources , sauvegarder tous les fichiers dans un répertoire ex : os5

puis **rendez les fichiers exécutable** avec par exemple :

pour tous le répertoire :

```
chmod -R +x os5
```

ou bien chaque fichier individuellement :

```
chmod +x button.py
```

Explications des fichiers :

```
(venv) pi@raspberrypi:~/Desktop/scripts $ ls  
camera_feed_test.py camera_properties.txt os1 os2 os3 os4 os5 venv
```

camera_feed_test.py est utilisé pour tester les différentes capacités de votre caméra. Nous allons apprendre ensuite comment les trouver et les stocker dans un fichier, par exemple : camera_properties.txt. Les fichiers os1...5 sont les différentes itérations du système. Nous allons nous concentrer uniquement sur os5. Pour plus d'informations à propos des autres systèmes d'exploitation, veuillez consulter le rapport complet.

```
(venv) pi@raspberrypi:~/Desktop/scripts/os5 $ ls  
button.py camera.py draw.py main.py menu.py __pycache__ rotary_encoder.py screen.py
```

button.py :

Ce fichier contient la classe responsable du contrôle du bouton. Dans cette version, on peut détecter des clics simples et des clics longs de 6 secondes, accédez au fichier pour modifier cette durée. Plus d'infos dans le rapport.

Veillez n'utiliser que le pin GPIO3 pour le bouton, car il servira à allumer et éteindre le système avec le même bouton. Assurez vous que votre interface I2C est éteinte (Elle est par défaut)

rotary_encoder.py :

Ce fichier contient la classe responsable du contrôle de l'encodeur rotatif. Plus d'infos dans le rapport. Pas de préférence sur les GPIO pins.

Il est toujours recommandé d'ajouter des résistances lorsque vous utilisez des capteurs pour limiter le courant en entrée dans les broches GPIO, que ce soit des résistances de pull-up ou de pull-down.

Plus d'informations à propos du bouton et de l'encodeur rotatif plus d'informations dans la section 4.3 du rapport

screen.py :

Ce fichier contient la classe responsable d'initialiser l'écran. On utilise l'interface SPI0 pour communiquer avec l'écran, c'est pourquoi nous avons besoin de l'activer au préalable. Pour changer les GPIO utilisés, vous devez accéder directement à la classe et les modifier. Cette classe ne prend pas les pins comme argument, contrairement aux classes `button` et `rotary_encoder`.

Comment activer l'interface SPI

Depuis un terminal tapez cette commande

```
sudo raspi-config
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password      Change password for the current u
2 Network Options          Configure network settings
3 Boot Options             Configure options for start-up
4 Localisation Options     Set up language and regional sett
5 Interfacing Options      Configure connections to peripher
6 Overclock               Configure overclocking for your P
7 Advanced Options        Configure advanced settings
8 Update                  Update this tool to the latest ve
9 About raspi-config       Information about this configurat

<Select>                  <Finish>
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera                 Enable/Disable connection to the
P2 SSH                    Enable/Disable remote command lin
P3 VNC                    Enable/Disable graphical remote a
P4 SPI                    Enable/Disable automatic loading
P5 I2C                    Enable/Disable automatic loading
P6 Serial                 Enable/Disable shell and kernel m
P7 1-Wire                 Enable/Disable one-wire interface
P8 Remote GPIO            Enable/Disable remote access to G

<Select>                  <Back>
```

camera.py :

Ce fichier contient la classe Camera, responsable de la capture vidéo et de son enregistrement.

comments trouvez les caractéristiques de votre caméra

v4l2 documentation : <https://www.mankier.com/1/v4l2-ctl>

Listez tous les périphériques vidéo :

'v4l2-ctl --list-devices'

```
(venv) pi@raspberrypi:~/Desktop/scripts/os5 $ v4l2-ctl --list-devices
```

```
bcm2835-codec-decode (platform:bcm2835-codec):
```

```
  /dev/video10
```

```
  /dev/video11
```

```
  /dev/video12
```

```
  /dev/video18
```

```
  /dev/video31
```

```
  /dev/media2
```

```
bcm2835-isp (platform:bcm2835-isp):
```

```
  /dev/video13
```

```
  /dev/video14
```

```
  /dev/video15
```

```
  /dev/video16
```

```
  /dev/video20
```

```
  /dev/video21
```

```
  /dev/video22
```

```
  /dev/video23
```

```
  /dev/media0
```

```
  /dev/media1
```

```
HD Webcam C525 (usb-3f980000.usb-1.2):
```

```
  /dev/video0
```

```
  /dev/video1
```

```
  /dev/media3
```

Détectez votre appareil, puis :

Lister les formats vidéo supportés et les résolutions d'un périphérique vidéo spécifique :

'v4l2-ctl --list-formats-ext --device path/to/video_device'

```
(venv) pi@raspberrypi:~/Desktop/scripts/os5 $ v4l2-ctl --list-formats-ext --device /dev/video0
```

```
ioctl: VIDIOC_ENUM_FMT
```

```
  Type: Video Capture
```

```
  [0]: 'YUYV' (YUYV 4:2:2)
```

Size: Discrete 640x480

Interval: Discrete 0.033s (30.000 fps)

Interval: Discrete 0.042s (24.000 fps)

Interval: Discrete 0.050s (20.000 fps)

Interval: Discrete 0.067s (15.000 fps)

Interval: Discrete 0.100s (10.000 fps)

Maintenant copier toutes ces informations dans un fichier .txt et tester les avec **camera_feed_test.py**

```
def main():
    # Initialize the camera (0 is the default camera)
    cap = cv2.VideoCapture(0)
    #set the video dimensions(resolution)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 800)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 448)

    # Set the frame rate
    #cap.set(cv2.CAP_PROP_FPS, 24)

    if not cap.isOpened():
        print("Error: Could not open video capture.")
        return

    w=cap.get(3)
    h=cap.get(4)
    print('width=',w)
    print('height=',h)
    print('capturing fps = ',cap.get(5))
```

Quoi choisir et comment : Testez tous les formats, commencez par changer

`cv2.CAP_PROP_FRAME_WIDTH` et `cv2.CAP_PROP_FRAME_HEIGHT` dans `camera_feed_test.py`. Ensuite, vérifiez à combien de FPS vous enregistrez. Notez tout cela dans une feuille quelque part, puis choisissez parmi toutes les options testées les plus pertinentes pour vous. Par exemple, nous avons choisi 5 options :

- Résolution de 800 x 448, enregistrement à 30 FPS
- Résolution de 800 x 600, enregistrement à 24 FPS

- Résolution de 800 x 448, enregistrement à 30 FPS
- Résolution de 1920 x 1080, enregistrement à 5 FPS
- Résolution de 1280 x 1080, enregistrement à 10 FPS

Enfin, ouvrez votre fichier `camera.py` et modifiez le chemin de sauvegarde de vos vidéos dans la méthode `set_rec_settings`. Par exemple, nous avons choisi : `/home/pi/Desktop/videos`.

```
def set_rec_settings(self,wfps,dimensions):  
    self.video.set(cv.CAP_PROP_FRAME_WIDTH,dimensions[0])  
    self.video.set(cv.CAP_PROP_FRAME_HEIGHT, dimensions[1])  
    self.result = cv.VideoWriter(  
        os.path.join('/home/pi/Desktop/videos', f'{self.file}__{dimensions[0]}x{dimensions[1]}.avi'),  
        cv.VideoWriter_fourcc(*'XVID'),wfps, dimensions)
```

Pour plus d'informations à propos de la caméra, des choix de captures, ou de l'enregistrement de fichiers, veuillez consulter le rapport complet.

Plus d'informations à propos de la camera dans la section 4.2 du rapport

draw.py

Ce fichier contient la classe Draw, responsable de créer les graphiques pour le menu et ses options. Chaque instance de cette classe ne peut créer qu'une seule image et la stocker. Ainsi, pour créer et stocker plusieurs images (plutôt que de créer une image à chaque fois), nous devons créer autant d'instances de cette classe que d'images souhaitées.

menu.py

La classe Menu est responsable d'intégrer toutes les parties du système ensemble. Vous pouvez trouver plus d'informations à ce sujet dans le rapport complet.

Plus d'informations à propos du menu et de l'écran dans la section 4.4 du rapport

main.py

Ce fichier contient la boucle principale de ce système. Ce fichier sera exécuté pour faire fonctionner le système.

```
def main():
    # Initialize the screen, draw, molette, and button
    screen = Screen()
    draw0=Draw(user_id="user 123",height=screen.height,width=screen.width)
    draw1=Draw(user_id="user 123",height=screen.height,width=screen.width)
    draw2=Draw(user_id="user 123",height=screen.height,width=screen.width)
    draw3=Draw(user_id="user 123",height=screen.height,width=screen.width)

    camera=Camera()
    button=Button(pin=3)
    molette=RotaryEncoder(clk_pin=board.D27, dt_pin=board.D22)

    # Initialize the menu
    menu = Menu(screen, draw0,draw1,draw2,draw3, molette, button,camera)

    # Add menu items
    menu.add_item("1 min",wfps=10,dimensions=(800,448),char='@ W I D E V G A ',mode='S L O W M O T I O
N')
    menu.add_item("5 min",wfps=22,dimensions=(800,600),char='@ S U P E R V G A ',mode='N O R M A L')
    menu.add_item("30 min",wfps=28,dimensions=(800,448),char='@ W I D E V G A ',mode='E C O - N O R M A
L')

    menu.add_item("1 h",wfps=100,dimensions=(1920,1080),char='@ F U L L H D' ,mode=' T I M E - L A P S E ')

    menu.add_item("3 h",wfps=600,dimensions=(1280,720),char='@ H D',mode='E C O - T I M E - L A P S E ')

    #menu.add_item("C A M V I E W ")

    menu.menu_0()
    #menu.menu_1()
    menu.menu_3()
```

Modifications à apporter :

1. Changez les options de votre menu comme indiqué. Les dimensions dépendent des options de votre caméra que vous avez choisies précédemment.

2. Attention à la fréquence d'écriture (w-fps) : W-fps n'est pas la fréquence de capture de votre caméra mais la fréquence d'écriture des images dans votre fichier.

En résumé :

- Pour les modes Time-Lapse, vous voulez une fréquence d'écriture supérieure à la fréquence de capture. Par exemple, pour nous : 720x1280, fréquence de capture = 10 FPS, fréquence d'écriture = 600.
- Pour les modes normaux, recherchez une fréquence d'écriture approximativement égale à la fréquence de capture.
- Pour les modes slow-motion, cherchez une fréquence d'écriture inférieure à la fréquence de capture. Par exemple, chez nous : 30 FPS de capture et 10 FPS d'écriture.

3. Enfin, changez les broches de votre encodeur rotatif selon vos préférences.

Pour plus d'informations sur les fréquences de capture, consultez la section finale du rapport.

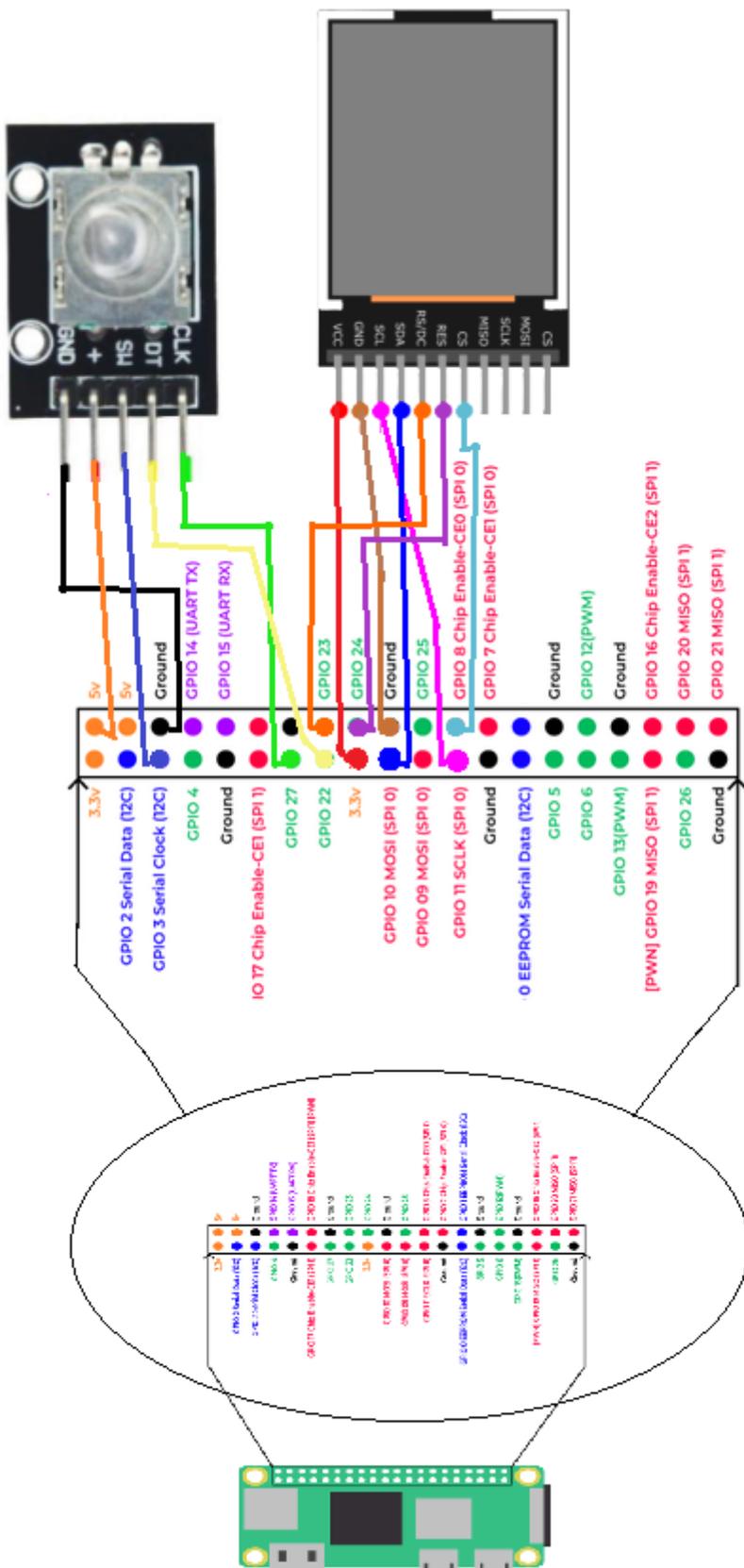
Assembler le Système

- **Rotary Encoder + Button:**

- VCC = 5V
- Ground = Ground
- SW = GPIO3
- CLK = GPIO27(votre choix)
- DT = GPIO22(votre choix)

- **Screen:**

- VCC = 3.3V
- Ground = Ground
- SCL = GPIO11 (SCLK) : Ligne d'horloge série pour la communication SPI. Ce pin fournit le signal d'horloge pour synchroniser la transmission des données.
- SDA = GPIO10 (MOSI) : Master Out Slave In pour la communication SPI. Ce pin envoie des données du Pi vers l'écran.
- DC = GPIO23(votre choix) : Pin d'entrée/sortie à usage général, utilisé ici pour la ligne de sélection de registre/commande de données (RS/DC) pour différencier les commandes et les données.
- RES = GPIO24(votre choix) Pin d'entrée/sortie à usage général, utilisé pour la ligne de réinitialisation (RES) pour réinitialiser l'affichage.
- CS = GPIO8 (CE0) : Activation de la puce 0, utilisée pour sélectionner le périphérique pour la communication sur le bus SPI.



Testez

Maintenant que tout est prêt, testez votre système. Les fichiers `rotary_encoder.py` et `button.py` contiennent tous des codes pour vous aider à vérifier que vos composants fonctionnent correctement. Vous pouvez les exécuter et les tester. Pour tester le système complet, lancez `main.py`.

(Assurez-vous que votre environnement virtuel est activé.)

```
python3 rotary_encoder.py
```

```
python3 bouton.py
```

```
python3 main.py
```

Documentation des Classes - Programmation

Classe : Bouton

Constructeur :

__init__(self, pin) : Initialise l'objet Bouton avec la broche GPIO spécifiée.

pin : Le numéro de la broche GPIO connectée au bouton.

Méthodes :

- **update(self)** : Met à jour l'état du bouton en lisant la broche GPIO et en vérifiant les pressions simples et longues.
- **get_State(self)** : Renvoie l'état actuel du bouton, indiquant s'il a été pressé ou longuement pressé.
- **cleanup(self)** : Nettoie les paramètres GPIO lorsque le programme se termine.

Référence:

RPi-GPIO documentation : <https://pypi.org/project/RPi.GPIO/>

Classe : RotaryEncoder

Constructeur :

__init__(self, clk_pin, dt_pin) : Initialise l'objet RotaryEncoder avec les broches d'horloge et de données spécifiées.

clk_pin : La broche GPIO utilisée pour le signal d'horloge.

dt_pin : La broche GPIO utilisée pour le signal de données.

Méthodes :

- **update(self)** : Met à jour la position du codeur rotatif en fonction de l'état des broches d'horloge et de données.
- **set_pos(self, pos)** : Définit le compteur de position sur la valeur spécifiée.

où pos est a nouvelle valeur de position à définir.

Références:

Digitalio docuemntation : <https://docs.circuitpython.org/en/latest/shared-bindings/digitalio/index.html>

I-gpio documentation : <https://rpi-lgpio.readthedocs.io/en/latest/>

Getting started with rotary encoders : <https://core-electronics.com.au/guides/getting-started-with-rotary-encoders-examples-with-raspberry-pi-pico/>

Classe : Screen

Constructeur :

__init__(self) : Initialise l'objet Screen avec la communication SPI et la configuration d'affichage.

Méthodes :

disp_img(self, image) : Affiche l'image fournie sur l'écran.
image : L'objet image à afficher.

Références:

adafruit_rgb_display documentation :

https://docs.circuitpython.org/projects/rgb_display/en/latest/api.html

busio documentation : <https://docs.circuitpython.org/en/latest/shared-bindings/busio/>

Classe : Camera

Constructeur :

__init__(self) : Initialise l'objet Camera avec des attributs pour la capture vidéo, l'enregistrement vidéo et le nom de fichier.

Méthodes :

- **init_capture(self)** : Initialise la capture vidéo depuis la caméra par défaut.
- **rtrn_frames(self)** : Capture et renvoie une image de la vidéo en direct.
- **save_frames(self, frame)** : Enregistre une image avec un superposition de la date et l'heure actuelles dans le fichier vidéo.
frame : L'image vidéo à enregistrer.
- **create_file(self)** : Génère un nom de fichier basé sur la date et l'heure actuelles.
- **set_rec_settings(self, wfps, dimensions)** : Définit les paramètres d'enregistrement, y compris la résolution et le taux d'images par seconde, et initialise l'enregistreur vidéo.
wfps : Le nombre d'images par seconde souhaité pour la vidéo.
dimensions : Un tuple spécifiant la largeur et la hauteur de la vidéo.
- **turn_off(self)** : Libère les ressources de capture vidéo et d'enregistrement vidéo.
Arguments : Aucun

Références:

open cv VideoCapture documentation :

https://docs.opencv.org/3.4/d8/dfe/classcv_1_1VideoCapture.html#a57c0e81e83e60f36c83027dc2a188e80

open cv Video Flags :

https://docs.opencv.org/3.4/d4/d15/group_videoio_flags_base.html#gaeb8dd9c89c10a5c63c139bf7c4f5704d

open cv VideoWrite documentation :

https://docs.opencv.org/4.x/dd/d9e/classcv_1_1VideoWriter.html

Classe : Draw

Constructeur :

__init__(self, height, width) : Initialise l'objet Draw avec une hauteur et une largeur spécifiées, créant un canevas d'image pour dessiner.

height : Hauteur de la zone de dessin.

width : Largeur de la zone de dessin.

Méthodes :

- **draw_msg(self, message, position)** : Dessine un message à la position spécifiée sur l'image.
message : Le message texte à dessiner.
position : Un tuple (x, y) indiquant la position où dessiner le message.
- **clear_img(self)** : Efface l'image en la remplissant avec la couleur de fond.
- **rtrn_img(self)** : Renvoie l'image actuelle.
- **convert(self, image)** : Convertit une image OpenCV en une image PIL et la redimensionne.
image : L'image OpenCV à convertir.
- **bound_mssg(self, message, position, color)** : Dessine un message avec une boîte de délimitation à la position spécifiée avec la couleur spécifiée.
message : Le message texte à dessiner.
position : Un tuple (x, y) indiquant la position où dessiner le message.
color : La couleur de la boîte de délimitation.
- **draw_arc_with_text(self, xy, start, end, fill='white', width=2, text_list=[], font=None)** : Dessine un arc avec du texte placé le long de celui-ci.
xy : Un tuple (x0, y0, x1, y1) définissant la boîte englobante de l'arc.
start : L'angle de départ de l'arc.
end : L'angle de fin de l'arc.
fill : La couleur de l'arc et du texte.
width : La largeur de la ligne de l'arc.
text_list : Une liste d'éléments texte à placer le long de l'arc.
font : La police à utiliser pour le texte (optionnel).

Référence:

PIL documentation : <https://pillow.readthedocs.io/en/stable/>

Classe : Menu

Constructeur :

__init__(self, screen, draw0, draw1, draw2, molette, bouton, camera) : Initialise un objet Menu avec des composants nécessaires pour l'affichage, le dessin, la navigation, et la capture vidéo.

screen : Instance de la classe Screen représentant l'écran d'affichage.

draw0, draw1, draw2 : Instances de la classe Draw pour dessiner différents écrans de menu (draw0 pour le menu principal, draw1 pour le menu d'arrêt, draw2 pour le menu d'enregistrement).

molette : Instance de la classe RotaryEncoder pour contrôler la navigation dans le menu.

bouton : Instance de la classe Button pour gérer les pressions de bouton.

camera : Instance de la classe Camera pour la capture et l'enregistrement vidéo.

Méthodes :

- **add_item(self, name, wfps, dimensions, char, mode)** : Ajoute un élément au menu avec ses réglages associés.
 - name : Nom ou libellé de l'élément de menu.
 - wfps : Taux d'images par seconde pour l'enregistrement vidéo.
 - dimensions : Dimensions de résolution pour l'enregistrement vidéo.
 - char : Caractéristique supplémentaire ou information sur l'élément de menu.
 - mode : Mode opérationnel de l'élément de menu.
- **menu_0(self)** : Affiche le menu principal (draw0) avec des options sélectionnables.
 - Description : Efface le canevas draw0, dessine les éléments du menu en forme d'arc, et affiche le menu sur l'écran.
- **control_menu_0(self)** : Contrôle la navigation et l'interaction au sein de menu_0.
 - Description : Met à jour la sélection actuelle à l'aide de l'encodeur rotatif (molette). Met en surbrillance l'élément de menu sélectionné à l'écran. Gère les pressions de bouton pour lancer des actions.
- **menu_1(self)** : Affiche le menu de confirmation d'arrêt (draw1).
 - Description : Efface le canevas draw1 et affiche un message de confirmation d'arrêt à l'écran.
- **control_menu_1(self)** : Contrôle l'interaction au sein de menu_1.
 - Description : Lance l'arrêt du système lors d'une pression sur le bouton. Efface le message de confirmation d'arrêt après l'exécution de la commande d'arrêt.
- **menu_2(self)** : Méthode de placeholder pour une future option de menu.
 - Description : Actuellement ne fait rien mais prévue pour une fonctionnalité future.
- **control_menu_2(self)** : Méthode de placeholder pour contrôler l'interaction au sein de menu_2.
 - Description : Actuellement ne fait rien mais prévue pour une fonctionnalité future.
- **menu_3(self)** : Affiche le menu d'enregistrement (draw2).
 - Description : Efface le canevas draw2 et affiche un message d'indication d'enregistrement à l'écran.
- **control_menu_3(self)** : Contrôle l'interaction au sein de menu_3.
 - Description : Capture des images depuis la caméra et les enregistre en continu jusqu'à ce qu'une pression sur le bouton indique d'arrêter. Affiche le statut d'enregistrement à l'écran.
- **menu_control(self)** : Méthode principale pour contrôler le système de menu.
 - Description : Met à jour et affiche continuellement le menu actuel en fonction de

l'entrée de l'utilisateur (encodeur rotatif et pressions de bouton). Exécute les actions de menu correspondantes telles que la navigation dans les menus, l'arrêt du système, le démarrage de l'enregistrement vidéo et l'affichage de la vue de la caméra.

FIN

Étape 3 : Website, Service, Entretien

Dans cette section, nous allons apprendre comment héberger un site web sur la Raspberry Pi pour télécharger les vidéos, puis nous passerons à la création de services pour lancer le code du système 'main.py' au démarrage. Enfin, nous gérerons l'entretien du système en créant des tâches qui s'exécutent régulièrement pour effacer les vidéos sauvegardées.

Website

Pour se Cultiver

Nginx :

Nginx est un serveur web open-source connu pour sa haute performance, sa stabilité, sa faible consommation de ressources et sa capacité à gérer un grand nombre de connexions simultanées. Il est souvent utilisé comme serveur web ou serveur proxy inverse, ainsi que pour la mise en cache, l'équilibrage de charge, et plus encore.

PHP est un langage de script côté serveur conçu pour le développement web, mais il est également utilisé comme langage de programmation généraliste. Pour exécuter du code PHP sur un serveur, vous avez besoin de l'interpréteur PHP installé, tout comme vous avez besoin de l'interpréteur Python pour exécuter du code Python.

PHP-FPM (PHP FastCGI Process Manager) :

PHP-FPM est une implémentation alternative de PHP FastCGI avec des fonctionnalités supplémentaires utiles pour les sites web à forte charge. Il permet à PHP de fonctionner efficacement avec Nginx. Bien que Nginx ne puisse pas directement interpréter PHP, il transmet les requêtes PHP à PHP-FPM, qui traite le code et renvoie le résultat à Nginx.

1-Préparation

1 -Créer un répertoire pour stocker tous les fichiers en rapport avec la structure de votre site, par exemple :

```
mkdir /home/pi/Desktop/website
```

2 -Installer Nginx et PHP

```
sudo apt install nginx
```

```
sudo apt install php8.2-fpm
```

3 -Démarrer et activer Nginx

```
sudo systemctl start nginx
```

```
sudo systemctl enable php8.2-fpm
```

Normalement, après cette étape, vous pouvez visiter votre site en tapant l'adresse IP de votre Raspberry Pi dans votre navigateur web. Vous devriez voir cette page.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

2-Modifier la Configuration de Nginx

Ouvrez le fichier de configuration par défaut pour l'éditer :

```
sudo nano /etc/nginx/sites-available/default
```

Remplacez le contenu du fichier en modifiant `root` avec le chemin du répertoire de votre site web et `alias` avec le chemin du répertoire où vous stockez vos vidéos, par exemple :

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /home/pi/Desktop/website;  
    index index.php index.html index.htm;  
  
    server_name _;
```

```

location / {
    try_files $uri $uri/ =404;
}

location /videos/ {
    alias /home/pi/Desktop/videos/;
    autoindex on;
    allow all;
}

location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
}

location ~ /\.ht {
    deny all;
}
}

```

explication :

listen 80 default_server; et listen [::]:80 default_server;;

- Ces directives indiquent à Nginx d'écouter les connexions entrantes sur le port 80, le port par défaut pour HTTP.
- default_server signifie que ce bloc de serveur sera le serveur par défaut si aucun autre bloc de serveur ne correspond à la demande entrante.

index index.php index.html index.htm;;

- Spécifie les fichiers par défaut à servir lorsqu'un répertoire est demandé. Nginx cherchera index.php, index.html, et index.htm dans cet ordre.

server_name _;;

- C'est un nom de serveur générique. Cela signifie que ce bloc de serveur gèrera les requêtes qui ne correspondent à aucun autre bloc de serveur avec un nom de serveur spécifique.

location / { ... }:

- Ce bloc configure la manière de gérer les requêtes pour l'URL racine (c'est-à-dire la page d'accueil) et toutes les autres URL non couvertes par d'autres blocs location.

```
try_files $uri $uri/ =404;;
```

- Cette directive tente de servir le fichier spécifié par l'URI (\$uri).
- Si le fichier n'existe pas, il essaie le répertoire (\$uri/).
- Si aucun des deux n'existe, il renvoie une erreur 404.

```
location /videos/ { ... }:
```

- Ce bloc configure la manière de gérer les requêtes pour les URL commençant par /videos/.

```
alias /home/pi/Desktop/videos/;
```

- Définit le répertoire correspondant aux requêtes URL /videos/ à /home/pi/Desktop/videos/.

```
autoindex on;;
```

- Active la liste automatique des répertoires pour le répertoire /videos/ si aucun fichier index n'est trouvé.

```
allow all;;
```

- Autorise l'accès à tous les utilisateurs.

```
location ~ /\.php$ { ... }:
```

- Ce bloc gère les requêtes pour les fichiers PHP. La regex ~ /\.php\$ correspond à toute requête se terminant par .php.

```
include snippets/fastcgi-php.conf;;
```

- Inclut les paramètres de configuration pour FastCGI, nécessaire pour traiter les fichiers PHP.

```
fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;;
```

- Passe les requêtes PHP au socket Unix PHP-FPM (FastCGI Process Manager) pour traitement.

```
location ~ /\.ht { ... }:
```

- Ce bloc gère les requêtes pour les fichiers commençant par .ht, tels que les fichiers .htaccess.

```
deny all;;
```

- Interdit l'accès à ces fichiers pour des raisons de sécurité, car les fichiers .htaccess sont utilisés pour la configuration et ne doivent pas être accessibles publiquement.

3-Rendre le site plus beau

Créer un fichier pour le style.

```
sudo nano /home/pi/Desktop/website/style.css
```

Ajouter ca :

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  max-width: 600px;
  width: 100%;
}

h1 {
  text-align: center;
  color: #333;
}

ul {
  list-style: none;
  padding: 0;
}

li {
  margin: 10px 0;
}

a {
  text-decoration: none;
```

```
color: #007bff;
font-weight: bold;
}

a:hover {
color: #0056b3;
}
```

en plus vous pouvez ajouter une image à votre site, sauvegardez la dans le répertoire de votre site.

4-Ajoutez un fichier PHP dans le répertoire de votre site web

Créer le fichier index.php

```
sudo nano /home/pi/Desktop/website/index.php
```

ensuite ajouter ca :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Video Downloads</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <h1>Video Downloads</h1>
    
    <ul>
      <?php
        $dir = '/home/pi/Desktop/videos/';
        $webPath = '/videos/';

        if ($handle = opendir($dir)) {
          while (false !== ($file = readdir($handle))) {
            if ($file != "." && $file != ".." && !is_dir($dir . $file)) {
              $filePath = $webPath . $file;
              echo "<li><a href='" . htmlspecialchars($filePath) . "' download>" . htmlspecialchars($file) .
```

```
"</a></li>";
    }
}
closedir($handle);
} else {
    echo "<p>Failed to open directory: $dir</p>";
}
?>
</ul>
</div>
</body>
</html>
```

explication :

Le document commence par une déclaration `<!DOCTYPE html>`, indiquant qu'il s'agit d'un document HTML5.

La balise `<html>` définit la langue comme étant l'anglais avec `lang="en"`.

La section `<head>` contient des métadonnées et des liens, notamment :

- `<meta charset="UTF-8">` définit l'encodage des caractères à UTF-8.
- `<title>` définit le titre de la page à "Video Downloads".
- `<link rel="stylesheet" href="style.css">` lie un fichier CSS externe nommé style.css.

La section `<body>` contient le contenu principal de la page :

- Une div avec la classe `container` encapsule le contenu.
- Une balise `<h1>` avec le titre "Video Downloads".
- Une balise `` affichant une image (images.png) avec une largeur maximale de 100%.
- Une liste non ordonnée (``) qui sera peuplée dynamiquement avec des éléments de liste (``) par PHP.

Le script PHP est intégré dans le HTML en utilisant les balises `<?php ... ?>`.

`$dir` est défini comme le chemin du répertoire contenant les vidéos (`/home/pi/Desktop/videos/`).

`$webPath` est défini comme le chemin URL correspondant au répertoire des vidéos (`/videos/`).

`opendir($dir)` tente d'ouvrir le répertoire spécifié par `$dir`. Si cela réussit, il retourne un handle de répertoire (`$handle`).

`readdir($handle)` lit les entrées du handle de répertoire. Cette boucle continue jusqu'à ce que toutes les entrées aient été lues.

Cette condition assure que les entrées spéciales (. et ..) et les sous-répertoires sont exclus, ne laissant que les fichiers réguliers.

`$filePath` combine le chemin web avec le nom de fichier pour créer l'URL du fichier.

`htmlspecialchars($filePath)` garantit que le chemin du fichier est encodé en toute sécurité pour la sortie HTML.

`htmlspecialchars($file)` garantit que le nom du fichier est encodé en toute sécurité pour la sortie HTML.

Un élément `` est créé avec une balise `<a>`. L'attribut `download` incite le navigateur à télécharger le fichier plutôt qu'à l'ouvrir.

`closedir($handle)` ferme le handle de répertoire.

Si `opendir($dir)` échoue, un message d'erreur est affiché.

5-Modifier les Permissions

```
sudo chown -R www-data:www-data /home/pi/Desktop/website
sudo chown -R www-data:www-data /home/pi/Desktop/videos
sudo chmod -R 755 /home/pi/Desktop/website
sudo chmod -R 755 /home/pi/Desktop/videos
```

Changement de propriétaire : Les commandes `chown` s'assurent que l'utilisateur `www-data` (qui est typiquement l'utilisateur sous lequel le serveur web fonctionne) possède les répertoires et leur contenu. C'est important car cela permet au serveur web de lire et d'exécuter les fichiers selon les besoins.

Changement des permissions : Les commandes `chmod` définissent les permissions afin que le propriétaire (`www-data`) ait un contrôle total (lecture, écriture, exécution), tandis que le groupe et les autres peuvent lire et exécuter les fichiers, mais pas les modifier. Cela permet de sécuriser les répertoires tout en autorisant l'accès nécessaire pour servir les fichiers via le serveur web.

6-Lancer votre site

Redémarrer Nginx et PHP-FPM

```
sudo systemctl reload nginx
```

Maintenant vous pouvez visiter votre site en tapant l'adresse IP de votre Raspberry Pi dans votre navigateur web.

7-Guide de Dépannage

Je ne suis pas un expert en développement web, mais j'ai fait de mon mieux pour vous expliquer comment tout cela fonctionne.

Voici un petit guide de dépannage pour vous aider à résoudre vos problèmes.

Vérification des Permissions

Assurez-vous que l'utilisateur du serveur web (www-data) possède les droits appropriés sur les répertoires et fichiers concernés :

```
sudo ls -l /home/pi/Desktop/website
sudo ls -l /home/pi/Desktop/videos
sudo ls -l /path/to/other/directories
```

Vérification des Logs d'Erreurs Nginx

Vérifiez le journal des erreurs de Nginx pour toute erreur liée à l'accès aux fichiers ou aux permissions :

```
sudo tail -f /var/log/nginx/error.log
```

Comportement du Navigateur

Consultez la Console du Navigateur :

Utilisez les outils de développement du navigateur (F12) pour vérifier les erreurs dans la console liées à l'accès aux fichiers ou aux permissions.

Gestion des Services au Démarrage

Liste des Services Configurés pour Démarrer

Utilisez systemctl pour voir quels services sont configurés pour démarrer au boot :

```
systemctl list-unit-files --type=service | grep enabled
```

Vérification des Services en Cours d'Exécution

Vérifiez l'état d'un service spécifique :

```
systemctl status nginx
systemctl status php8.2-fpm
```

Vérification des Logs Détaillés

Consultez les journaux détaillés pour un service spécifique en cas de problème :

```
sudo journalctl -xeu nginx.service  
sudo journalctl -xeu php-fpm.service
```

Validation de la Configuration Nginx

Assurez-vous que la configuration Nginx est correcte en utilisant la commande de test :

```
nginx -t
```

Service

Dans cette partie, nous allons voir comment exécuter le code de notre système `main.py` (de la section précédente) au démarrage.

Pour ce Cultiver

Un fichier de service est un fichier de configuration utilisé par le gestionnaire de systèmes et de services systemd pour définir et gérer les services sous Linux. Ce fichier contient des informations sur la manière dont un service particulier doit être démarré, arrêté et géré par systemd. Il définit les commandes à exécuter pour démarrer le service, les dépendances du service, ainsi que d'autres paramètres qui contrôlent le comportement du service.

Comparaison entre systemd et crontab

Systemd et crontab sont deux outils utilisés pour programmer et gérer des tâches sur un système Linux, mais ils sont utilisés dans des contextes différents et ont des forces et des objectifs distincts. Systemd est conçu pour la gestion des services (processus de longue durée), garantissant leur démarrage au boot, leur redémarrage en cas d'échec, et leur arrêt en douceur. Il gère des dépendances complexes entre services et offre une journalisation centralisée avec journalctl. Il est également capable de redémarrer automatiquement les services en cas de défaillance et de surveiller leur statut. Systemd fournit une interface standardisée sur diverses distributions Linux, ce qui le rend idéal pour la gestion des services en arrière-plan tels que les serveurs web et les serveurs de bases de données, ainsi que pour les scripts critiques nécessitant un fonctionnement continu. Par exemple, exécuter un script Python en tant que service continu qui doit redémarrer en cas de panne est un cas d'utilisation typique pour systemd.

En revanche, crontab est idéal pour la planification des tâches à des moments spécifiques, des intervalles ou des dates. Il est simple et direct à configurer pour des tâches périodiques et peut exécuter des tâches aussi fréquemment que chaque minute. Chaque utilisateur peut avoir son propre crontab, permettant une planification des tâches au niveau de l'utilisateur. Crontab est donc approprié pour les tâches de maintenance périodiques comme les sauvegardes et les nettoyages, ainsi que pour les tâches uniques ou récurrentes comme l'envoi d'e-mails quotidiens ou la génération de rapports. Par exemple, exécuter un script chaque jour à minuit pour sauvegarder

une base de données est une tâche typique pour crontab.

1-Créer un fichier de service Systemd

Créez un fichier de service pour gérer votre script Python. Enregistrez-le dans `/etc/systemd/system/` avec une extension `.service`, par exemple, `camera_system.service`.

```
sudo nano /etc/systemd/system/camera_system.service
```

2-Modifier le fichier de service

Dans l'éditeur `nano`, ajoutez le contenu suivant au fichier de service :

```
[Unit]
Description=Camera System Service
After=local-fs.target

[Service]
User=root
Group=root
WorkingDirectory=/home/pi/Desktop/scripts/os5
Environment="PATH=/usr/bin:/usr/local/bin:/usr/sbin:/home/pi/Desktop/scripts/venv/bin"
ExecStart=/bin/bash -c '. /home/pi/Desktop/scripts/venv/bin/activate && python main.py'
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Explication :

- **[Unit]** :

- `Description` : Une brève description du service.
- `After` : S'assure que le service démarre après que le système de fichiers local soit disponible.

- **[Service]** :

- `User` : Exécute le service en tant qu'utilisateur `root`.
- `Group` : Exécute le service dans le groupe `root`.
- `WorkingDirectory` : Définit le répertoire de travail pour le script.
- `Environment` : Définit le chemin pour inclure les répertoires nécessaires, y compris l'environnement virtuel Python.

- `ExecStart` : Spécifie la commande pour exécuter votre script Python.
- `Restart` : Redémarre automatiquement le service s'il se termine avec un statut non nul.
- `RestartSec` : Attend 10 secondes avant de redémarrer le service.

- **[Install]** :

- `WantedBy` : S'assure que le service démarre en mode multi-utilisateur (démarrage normal).

3-Vérifiez les Permissions et la Propriété des Fichiers

Exécutez la commande suivante pour vérifier les permissions et la propriété de votre script Python :

```
ls -l /home/pi/Desktop/scripts/os5/main.py
```

Exemple de Sortie :

```
-rwxr-xr-x 1 pi pi 1234 Jun 27 12:34 /home/pi/Desktop/scripts/os5/main.py
```

Explication de la Sortie :

- `rwxr-xr-x` : Ce sont les permissions pour le fichier.
- `rwx` : Le propriétaire (`pi`) a les permissions de lecture, écriture et exécution.
- `r-x` : Le groupe (`pi`) a les permissions de lecture et exécution.
- `r-x` : Les autres ont les permissions de lecture et exécution.
- `1` : Le nombre de liens physiques vers le fichier.
- `pi` : Le propriétaire du fichier.
- `pi` : Le groupe du fichier.
- `1234` : La taille du fichier en octets.
- `Jun 27 12:34` : La date et l'heure de la dernière modification du fichier.
- `/home/pi/Desktop/scripts/os5/main.py` : Le chemin vers le fichier.

Veuillez vérifier les permissions et la propriété en utilisant la commande ci-dessus et confirmer que l'utilisateur `pi` a les permissions nécessaires.

4- Recharger la configuration de Systemd

Rechargez la configuration du gestionnaire Systemd pour reconnaître le nouveau fichier de service et toutes les modifications qui y ont été apportées.

```
sudo systemctl daemon-reload
```

5- Activer et démarrer le service

Activez le service systemd pour qu'il démarre automatiquement au démarrage.

```
sudo systemctl enable camera_system.service
```

Démarrez le service systemd pour lancer votre script Python.

```
sudo systemctl start camera_system.service
```

6-Guide de Dépannage

Arrêter le service

```
sudo systemctl stop camera_system.service
```

Désactiver le démarrage automatique

```
sudo systemctl disable camera_system.service
```

Vérifiez l'état du service.

```
sudo systemctl status camera_system.service
```

Vérifiez les journaux du service :

```
sudo journalctl -u camera.service
```

Entretien

On va utiliser crontab pour entretenir notre système et automatiser la tâche de suppression de vidéos.

1-Ouvrez Crontab pour l'édition

```
crontab -e
```

2-Ajoutez une tâche Cron pour supprimer les vidéos

ajoutez cette ligne :

```
0 2 * * * find /home/pi/Desktop/videos -name "*.avi" -type f -exec rm {} \; >>  
/home/pi/Desktop/videos/deletion.log 2>&1
```

puis sauvegardez et quittez

Explication de l'emploi de temps Cron (0 2 * * *) :

- 0 : Minutes (0-59)
- 2 : Heure (0-23)
- * : Jour du mois (1-31)
- * : Mois (1-12)
- * : Jour de la semaine (0-6, où 0 est dimanche)

Ce planning Cron (0 2 * * *) exécute la commande find tous les jours à 2h00 du matin (0 2) pour trouver et supprimer (-delete) les fichiers (-type f) correspondant à *.avi .

```
find /home/pi/Desktop/videos -name "*.avi" -type f -exec rm {} \;
```

: Recherche tous les fichiers *.avi dans le répertoire /home/pi/Desktop/videos et les supprime (-exec rm {} \;).

/home/pi/Desktop/videos/deletion.log : Redirige la sortie standard (stdout) de la commande find (qui inclut les messages sur les fichiers supprimés) et l'ajoute au fichier deletion.log.

2>&1 : Redirige les erreurs standard (stderr) vers stdout, de sorte que stdout et stderr sont capturés dans le fichier journal.

3-Créez le fichier deletion.log pour enregistrer les messages de suppression.

```
touch /home/pi/Desktop/videos/deletion.log
```

4- Ajuster les permissions du fichier

Définissez les permissions appropriées pour que l'utilisateur exécutant la tâche cron puisse écrire dans le fichier journal. Par exemple, si votre utilisateur est pi :

```
chmod 664 /home/pi/Desktop/videos/deletion.log  
chown pi:pi /home/pi/Desktop/videos/deletion.log
```

chmod 664 : Accorde les permissions de lecture et d'écriture à l'utilisateur et au groupe, et la permission de lecture aux autres.

chown pi : Change la propriété du fichier à l'utilisateur pi et au groupe pi.

5-Vérifier les permissions

Vérifiez les permissions et la propriété du fichier journal.

```
ls -l /home/pi/Desktop/videos/deletion.log
```

Assurez-vous que les permissions affichent (rw-rw-r--) et que la propriété est correcte (pi ou votre utilisateur et votre groupe).

6-Vérifier la tâche Cron

```
crontab -l
```

References :

Host a website on your raspberry Pi : <https://www.youtube.com/watch?v=Y1mNeWwj8D0>

Run a program on your raspberry Pi : <https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/>

FIN

Étape 4 : Assemblage

Maintenant que tout le système fonctionne bien et a été testé, nous allons passer à l'étape de construction de la boîte. Visitez le GitHub ou installez les fichiers nécessaires. Vous pouvez soit imprimer en 3D, soit découper la boîte avec une découpeuse laser, puis l'assembler et commencer à documenter. Pour plus d'informations sur la documentation, je vous invite à consulter la thèse de Madame Rigaud. Pour savoir comment utiliser une découpeuse laser, demandez à votre FabLab.

Plus d'info a propos de la boite dans la section 5 du rapport.