

## Rapport de la séance 13

19 mars 2024

### Objectif de séance :

- Actionner les moteurs avec les codes de base fournis
- Parvenir à actionner les servomoteurs
- Découvrir l'utilisation de la caméra

### Actionner les moteurs avec les codes de base fournis :

Cette section est essentiellement présente pour évoquer la première interaction avec la JetSon et le système Jupyter fournis par Nvidia pour débiter avec la JetSon.

Jupyter est une interface permettant d'aborder facilement les notions de bases implémentées par Nvidia telle que le contrôle des moteurs. Celle-ci est remplie d'informations complémentaires afin de comprendre plus en détails le fonctionnement d'une simple ligne de code puisque le langage n'est plus Arduino, mais du Python.

Après avoir essayé et constaté le bon fonctionnement des moteurs, je préfère ne pas m'attarder sur cette notion déjà étudiée, car celle-ci est très semblable à Arduino grâce à Jupyter, ainsi que la librairie Python fourni par Nvidia.

### Tentative d'activation des servomoteurs :

Cette rubrique est indispensable à détailler contrairement à la précédente à propos des moteurs pour 2 raisons.

La première raison : l'activation de servomoteurs n'est pas traitée par l'interface Jupyter, il va donc être nécessaire d'acquérir des renseignements ailleurs que par les ressources fournies par Nvidia.

La seconde raison : les servomoteurs sont facilement implémentés avec Arduino, car il s'agit d'un composant de base pour comprendre les rudiments de cette carte, ce qui n'est pas le cas pour la JetSon.

Il est alors possible d'actionner les servomoteurs via un bus I2C, par le module PCA9685 :



Cependant, lors de la procédure d'installation pour aboutir à une rotation des servomoteurs, une erreur que je ne parviens pas à contourner s'est montré.

Il existe un forum qui répertorie mon problème sur le site de Nvidia, seulement la solution apportée n'a pas résolu mon problème. Je préfère le laisser sur le côté pour l'instant afin d'essayer la téléopération ainsi que la caméra.

Informations supplémentaires sur l'erreur à résoudre avant la prochaine séance :

Pour l'activation des servomoteurs, j'ai tout d'abord suivi un premier tutoriel d'un répertoire sur GitHub, qui possède également une vidéo explicative.

Lors d'une étape, il faut procéder à l'activation d'un fichier qui me rend l'erreur ci-contre.

```
Jetbot@jetson-4-3: ~/ServoKit/ServoKit
jetbot@jetson-4-3:~/ServoKit$ cd ServoKit/
jetbot@jetson-4-3:~/ServoKit/ServoKit$ ls
images          installServoKit.sh  README.md  servoPlay.py
installGamePad.sh LICENSE          scripts
jetbot@jetson-4-3:~/ServoKit/ServoKit$ ./installServoKit.sh
cp: cannot stat '/opt/nvidia/jetson-gpio/etc/99-gpio.rules': No such file or directory
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3-ubuntu18.04.8).
0 upgraded, 0 newly installed, 0 to remove and 633 not upgraded.
10 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Setting up libgomp1:arm64 (8.4.0-1ubuntu1~18.04) ...
Setting up libatomic1:arm64 (8.4.0-1ubuntu1~18.04) ...
Setting up libcc1-0:arm64 (8.4.0-1ubuntu1~18.04) ...
Setting up libc6-dbg:arm64 (2.27-3ubuntu1.6) ...
Setting up libtsan0:arm64 (8.4.0-1ubuntu1~18.04) ...
Setting up linux-libc-dev:arm64 (4.15.0-213.224) ...
Setting up liblsan0:arm64 (8.4.0-1ubuntu1~18.04) ...
Setting up libc-dev-bin (2.27-3ubuntu1.6) ...
Setting up libc6-dev:arm64 (2.27-3ubuntu1.6) ...
```

```
Jetbot@jetson-4-3: ~/ServoKit/ServoKit
-py3-none-any.whl
ERROR: Could not find a version that satisfies the requirement Adafruit-Blinka==7.0.0 (from adafruit-circuitpython-servokit) (from versions: 0.1.3, 0.1.4, 0.1.5, 0.1.6, 0.1.7, 0.1.8, 0.1.9, 0.1.10, 0.2, 0.2.1, 0.2.2, 0.2.3, 0.2.4, 0.2.5, 0.2.6, 0.2.7, 0.3.0, 0.3.1, 0.3.2, 0.4.0, 1.0.2, 1.1.0, 1.1.1, 1.2.0, 1.2.1, 1.2.2, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.3.0, 1.3.1, 1.3.2, 1.3.3, 1.3.4, 2.0.0, 2.0.1, 2.0.2, 2.1.0, 2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 2.2.0, 2.2.1, 2.3.0, 2.3.1, 2.3.2, 2.3.3, 2.4.0, 2.4.1, 2.5.0, 2.5.1, 2.5.2, 2.5.3, 2.6.0, 2.6.1, 3.0.0, 3.0.1, 3.0.2, 3.0.3, 3.0.4, 3.0.5, 3.0.6, 3.0.7, 3.0.8, 3.1.0, 3.1.1, 3.1.2, 3.2.0, 3.3.0, 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5, 3.3.6, 3.3.7, 3.3.8, 3.3.9, 3.3.10, 3.4.0, 3.4.1, 3.5.0, 3.5.1, 3.5.2, 3.6.0, 3.6.1, 3.7.0, 3.7.1, 3.8.0, 3.9.0, 3.10.0, 4.0.0, 4.1.0, 4.2.0, 4.3.0, 4.4.0, 4.5.0, 4.6.0, 4.7.0, 4.8.0, 4.9.0, 4.9.1, 4.9.2, 4.10.0, 4.10.1, 5.0.0, 5.0.1, 5.1.0, 5.2.0, 5.2.1, 5.2.2, 5.2.3, 5.2.4, 5.3.0, 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.4.0, 5.4.1, 5.5.0, 5.5.1, 5.5.2, 5.5.3, 5.5.4, 5.6.0, 5.7.0, 5.8.0, 5.8.1, 5.8.2, 5.9.0, 5.9.1, 5.9.2, 5.10.0, 5.11.0, 5.12.0, 5.13.0, 5.13.1, 6.0.0, 6.0.1, 6.0.2, 6.1.0, 6.2.0, 6.2.1, 6.2.2, 6.3.0, 6.3.1, 6.3.2, 6.4.0, 6.4.1, 6.4.2, 6.5.0, 6.6.0, 6.6.1, 6.6.2, 6.7.0, 6.8.0, 6.8.1, 6.8.2, 6.9.0, 6.9.1, 6.9.2, 6.10.0, 6.10.1, 6.10.2, 6.10.3, 6.11.0, 6.11.1, 6.12.0, 6.13.0, 6.13.1, 6.14.0, 6.14.1, 6.15.0)
ERROR: No matching distribution found for Adafruit-Blinka==7.0.0 (from adafruit-circuitpython-servokit)
WARNING: You are using pip version 19.3.1; however, version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
jetbot@jetson-4-3:~/ServoKit/ServoKit$
```

La solution de Nvidia ne résolvant pas le problème, j'ai opté pour un second tutoriel qui était uniquement une vidéo explicative, qui m'a conduit à la même erreur. Il sera donc nécessaire d'approfondir les recherches avant la prochaine séance pour éviter de rester bloqué à cette étape très longtemps.

### Téléopération du robot en prévision de suivi de ligne par caméra :

A première vue, cette section n'a pas de sens avec notre projet, car le robot doit être autonome. Cependant, l'interface Jupyter possède des ressources de nouveaux intéressants : un suivi de ligne par caméra à l'aide d'une IA.

Seulement, le bon fonctionnement, ainsi que la bonne compréhension des explications quant à cette rubrique de suivi de ligne, nécessite tout d'abord de comprendre la téléopération du robot.

Quel est le lien entre le suivi de ligne par caméra et la téléopération ?

Le suivi de ligne par caméra nécessite d'avoir une banque d'image indiquant le mouvement à effectuer selon l'environnement, pour cela un curseur est placé sur l'overlay de la caméra et l'utilisation d'une manette de contrôle (téléopération) permet de positionner rapidement le curseur sur l'image. Cette fonction est donc très utile pour minimiser le temps entre chaque image pour compléter rapidement, et ainsi en plus grande quantité, la banque de données.

Ainsi, après avoir suivi les informations de Jupyter, le robot est désormais pilotable à l'aide de la manette de contrôle (un GIF est disponible dans le README.md pour illustrer).

Objectif de séance prochaine :

- Résoudre le problème de code des servomoteurs
- Récolter les images pour la banque de données
- Essayer le suivi de ligne par caméra