SO-ARM100 - ROS2 et IA avec LeRobot

LeRobot sur Ubuntu

Installation

- Installer Miniconda pour Linux : l'environnement de développement Python

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
# Vérifier que la clé SHA256 de Miniconda3-latest-Linux-x86_64.sh ici :
https://repo.anaconda.com/miniconda/ correspond à :
sha256sum ~/Miniconda3-latest-Linux-x86_64.sh
bash ~/Miniconda3-latest-Linux-x86_64.sh
source ~/.bashrc
```

- Créer et activer l'environnement Conda

```
conda create -y -n lerobot python=3.10
conda activate lerobot
git clone https://github.com/huggingface/lerobot.git ~/lerobot
conda install ffmpeg -c conda-forge
cd ~/lerobot && pip install -e ".[feetech]"
```

Ne pas activer conda au démarrage : `conda config --set auto_activate_base false`

Ne pas configurer le shell pour initialiser conda au démarrage : `conda init --reverse $SHELL`

Configurer les servomoteurs

La carte `FE-URT-1` fournie par Feetech n'est pas détectée à cause d'un conflit avec un paquet de brail. On le désinstalle :

```
sudo apt-get autoremove brltty
```

https://askubuntu.com/questions/1321442/how-to-look-for-ch340-usb-drivers/1472246#1472246

https://github.com/huggingface/lerobot/blob/main/examples/10_use_so100.md#c-configure-the-motors

- Brancher la carte
- Trouver l'interface USB sur laquelle est branchée la carte, par ex. `/dev/ttyACM0`

```
python lerobot/scripts/find_motors_bus_port.py
```

- Changer les droits sur le interfaces USB

```
sudo chmod 666 /dev/ttyACM0
sudo chmod 666 /dev/ttyACM1
```

- Ouvrir Codium > File > Open Folder > `admin_ros/lerobot`
- Modifier le fichier de config

`gedit ~/lerobot/lerobot/common/robot_devices/robots/configs.py`

- Chercher la config du So100 en ligne 436 `class So100RobotConfig(ManipulatorRobotConfig):`
- Remplacer `port="/dev/tty.usbmodem58760431091",` pour le `leader_arms` (L446) et le `follower_arms` (L463) par le port découvert
- Brancher les servos un à un à la carte puis lancer le script d'initialisation, en incrémentant l'ID à chaque fois :

```
python lerobot/scripts/configure_motor.py \
  --port /dev/tty.usbmodem58760432961 \
  --brand feetech \
  --model sts3215 \
  --baudrate 1000000 \
  --ID 1
```

- Au fur et à mesure les brancher en série et/ou noter l'ID sur le moteur
- Les servos sont bougés à leur position centrale et l'offset mis à 0

Construction et assemblage mécanique

Une version 101 est sortie en 05/2025. Le Leader est plus simple à assembler, et plus besoin de démonter les servos pour enlever un engrenage et les rendre passifs. Il suffit d'acheter le kit de 6 servos avec 3 rapports de transmission différents :

- https://github.com/TheRobotStudio/SO-ARM100?tab=readme-ov-file#sourcing-parts

- https://www.alibaba.com/product-detail/6PCS-7-4V-STS3215-Servos-for_1601428584027.html?spm=a2747.product_manager.0.0.757c2c3clU7uH3
- Suivre le guide d'assemblage pour le SO101 : https://github.com/huggingface/lerobot/blob/main/examples/12_use_so101.md#step-by-step-assembly-instructions
- Pour le SO100 : https://github.com/huggingface/lerobot/blob/main/examples/10_use_so100.md#d-step-by-step-assembly-instructions

Astuces pour l'assemblage

- Mettre une vis sur l'arbre moteur et l'axe passif (à l'opposée de l'arbre moteur) quand il y a la place d'en mettre une (vérifier qu'il y aura la place après assemblage des éléments autour du moteur)
- Ne plus bouger les servos après leur initialisation qui les met à l'angle 0. Assembler les éléments de manière à ce que le robot soit en configuration initiale avec tous les moteurs à 0
- Si vous n'avez pas respecté le dernier point, il est possible d'ajouter un offset dans la configuration du servo

ROS2 et MoveIt2

Installer les paquets ROS2 du SO-ARM100 :

- Cloner le paquet dans un workspace ROS2 https://github.com/JafarAbdi/ros2_so_arm100
- Cloner le submodule https://github.com/TheRobotStudio/SO-ARM100 dans `so_arm100_description/SO-ARM100` (https://www.freecodecamp.org/news/how-to-use-git-submodules/)
- Ou simplement :

```
mkdir -p ~/ws_so_arm100/src
cd ~/ws_so_arm100/src
git clone --recurse-submodules https://github.com/JafarAbdi/ros2_so_arm100
cd ~/ws_so_arm100
sudo rosdep init
rosdep update && rosdep install --ignore-src --from-paths src -y
colcon build --symlink-install # dans une VM ajouter --parallel-workers 1
source install/setup.bash
ros2 launch so_arm100_moveit_config demo.launch.py hardware_type:=mock_components #
hardware_type:=real for running with hardware
```

Tester la démo en simulation :

- Lancer un des scripts : https://github.com/JafarAbdi/ros2_so_arm100?tab=readme-ov-file#usage

Pilotage ST3215 depuis un ESP32

https://github.com/sepastian/ESP32_ST3215

Revision #12
Created 8 April 2025 20:29:23 by admin_idf
Updated 21 May 2025 07:21:25 by admin_idf