SO-ARM100 - Robotique éducative

Les bases d'un bras robot

https://docs.phospho.ai/learn/overview

Pilotage bras robot avec Scratch

https://www.poppy-education.org/activites/initiation-ergo-jr-et-scratch/

Pilotage du SO-ARM100 avec Phospho

https://docs.phospho.ai/learn/overview

Simulation et pilotage du SO-ARM100 avec ROS2

> Attention avant d'utiliser le robot avec ROS2, il faut avoir calibré les servomoteurs, par ex. avec le script de calibration du projet LeRobot

https://github.com/JafarAbdi/ros2_so_arm100

https://discourse.openrobotics.org/t/interactive-so-101-ik-in-ros-2-with-viser-robokin/53850

Jumeau numérique

Pilotage de la simulation ou du vrai robot

https://github.com/tessel-la/robo-boy

Adapter le tuto suivant au SO-ARM100 : https://innovation.iha.unistra.fr/books/robotique-open-source/page/programmer-un-robot-avec-moveit2-jumeau-numerique

Contrôle des moteurs par un GUI de "jogging"

Joint Trajectory Controller

```
ros2 run rqt_joint_trajectory_controller rqt_joint_trajectory_controller
```

https://github.com/tessel-la/robo-boy

Contrôle de l'outil sans collisions via le plugin MoveIt de RViz

Cartesian Trajectory

```
ros2 launch so_arm100_moveit_config moveit_rviz.launch.py
```

Réalisation d'un programme en Python

https://moveit.picknik.ai/main/doc/examples/motion_planning_python_api/motion_planning_python_api_tutorial.html#single-pipeline-planning-pose-goal

```
# set plan start state to current state
panda_arm.set_start_state_to_current_state()

# set pose goal with PoseStamped message
pose_goal = PoseStamped()
pose_goal.header.frame_id = "panda_link0"
pose_goal.pose.orientation.w = 1.0
pose_goal.pose.position.x = 0.28
pose_goal.pose.position.y = -0.2
pose_goal.pose.position.z = 0.5
panda_arm.set_goal_state(pose_stamped_msg=pose_goal, pose_link="panda_link8")

# plan to goal
plan_and_execute(panda, panda_arm, logger)
```

En utilisant l'environnement de développement Jupyter Notebook
https://moveit.picknik.ai/main/doc/examples/jupyter_notebook_prototyping/jupyter_notebook_prototyping_tutorial.html

Pilotage du bras robot par LeRobot (IA, VR, etc.)

Environnement Python sous Windows ou Linux

Contrôle du bras par clavier ou manette

Avec LeRobot+Phospho https://docs.phospho.ai/basic-usage/teleop

ou avec ROS2+MoveIt2
https://moveit.picknik.ai/main/doc/examples/jupyter_notebook_prototyping/jupyter_notebook_prototyping_tutorial.html

Contrôle du bras par Oculus Quest

Compatible LeRobot (WIndows et Ubuntu) :

- alternative gratuite et open source à phospho
- https://github.com/vladfatu/telerobot

Depuis Windows :

- Appli Oculus Phospho https://docs.phospho.ai/examples/teleop
- 222€ https://www.meta.com/en-gb/experiences/phospho-teleoperation/8873978782723478/

Depuis Ubuntu avec ROS2 et moveit_servo :

- https://github.com/ZorAttC/franka_vr
- https://moveit.picknik.ai/main/doc/examples/realtime_servo/realtime_servo_tutorial.html
- https://github.com/rail-berkeley/oculus_reader
  - Enable Oculus Quest development mode
  - Always allow USB debugging from this computer
  - Connexion USB (ADB) ou wifi

Autre : https://github.com/lts0429/teleoperation

Contrôle du bras via un modèle d'IA

https://docs.phospho.ai/basic-usage/inference

- Créer un compte huggingface.ia
- Sign In dans phosphobot
- Dans les paramètres, ajouter la clé d'API huggingface
- Par défaut l'inférence du modèle d'IA qui permet de piloter le robot depuis l'image des caméras tournera sur un GPU sur les serveurs de huggingface ou phospho
- On peut faire tourner l'inférence du modèle d'IA sur le PC local s'il a une bonne carte graphique NVidia
- Suivre ces instructions : https://github.com/phospho-app/phosphobot/tree/main/inference#setup-a-server
- Démarrer le serveur d'inférence uv
- `uv run ACT/server.py --model_id=<CHEMIN_VERS_LE_MODELE_LOCAL>`
- Appeler le serveur d'inférence depuis un script python : https://docs.phospho.ai/basic-usage/inference#2-call-your-inference-server-from-a-python-script

Pilotage ST3215 depuis un ESP32 (embarqué, micro-ROS)

Sans utiliser la carte de contrôle Feetech/Waveshare

(5€) : https://github.com/sepastian/ESP32_ST3215

Pilotage bluetooth depuis un smartphone

Utiliser l'ESP32 pour faire l'interface bluetooth vers une télécommande smartphone ? Avec ou sans carte de contrôle officielle (cf. ci-dessus) ?

Micro-ROS avec bras robot

- approche pour grasping référencée capteur via un TOF-sensor

https://micro.ros.org/docs/tutorials/demos/openmanipulator_demo/

- Télécommander une pose relative de la pince via un capteur https://micro.ros.org/docs/tutorials/demos/moveit2_demo/

> 6-DoF Inertial Measurement Unit (LSM6DSL), composed of an accelerometer and a gyroscope, and a 3-DoF magnetometer (LIS3MDL). The fusion of the

measurements fetched by these sensors outputs the pose, or relative orientation of the board with respect to a fixed reference frame.

Revision #12
Created 21 May 2025 07:22:15 by admin_idf
Updated 6 May 2026 11:13:54 by admin_idf