



## Workshop Unit 4

Interfacing & Controlling your Robotic Hand

**Barnabas Gavin Cangan** 

14 October 2024





#### Overview



- 1. Required Materials
- 2. Structure of the Code Framework
- 3. Setup and running the example code





### **Required Material**



Dynamixel Motors	Dynamixel Equipment	Your own Computer
Each motor must have a unique ID. See the guide for how to change the Dynamixel IDs.	<ul> <li>U2D2 w/ USB cable</li> <li>U2D2 Power Hub Board</li> <li>12V Power Supply</li> </ul>	<ul> <li>Linux operating system</li> <li><i>Python</i> installed</li> <li><i>Dynamixel-SDK</i> installed</li> </ul>





#### Code Framework

# joint angles tendon lengths motor positions gripper definitions

motor information

#### **Gripper Defs:**

- Definitions of the Gripper
- Stores joint, tendon and motor definitions

#### **example.py** Define your joint-level motions and poses.



- Calculate motor positions
- Initialize & terminate motors
- Calibrate hand & read config

#### **Dynamixel Client:**

- Read & Write motor param.
- Communication w/ motors

#### **Finger Kinematics:**

- Calculate free tendon length from joint position





Setup and running the example code

- Setup the Dynamixel motors
  - Set unique IDs for each motor
  - Set baudrate to 3 Mbps
- Run the example code
  - Run *example.py* and check if everything works
- Adjust the framework to your application
   Implement your kinematics





## **TH** zürich





