

Dynamixel Tutorial

Yi Zhang (yz892)

Quick start

1. Download [the repository](#).
2. Set up hardware as described in next page.
3. Run `dynamixel_sdk_sample_read_write.py` to try position control.
Or, run `my_current_control.py` to try current/torque control.

You might need to change `DEVICENAME` in
`my_global_variables_XL330M288T.py` if not using Mac.

Refer to 7th page on finding out which port is used by Dynamixel.

Dynamixel hardware setup

U2D2 PHB set

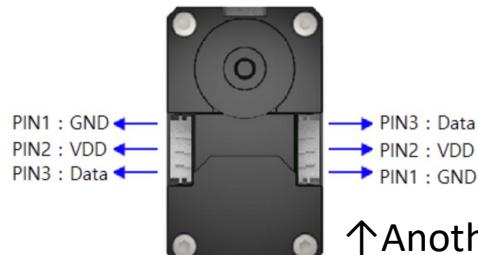
- Operating voltage: 3.5 V ~ 24.0 V
- Maximum current: 10.0 A
- TTL/RS485-based DYNAMIXEL



XL330-M288

(Dynamixel Protocol 2.0)

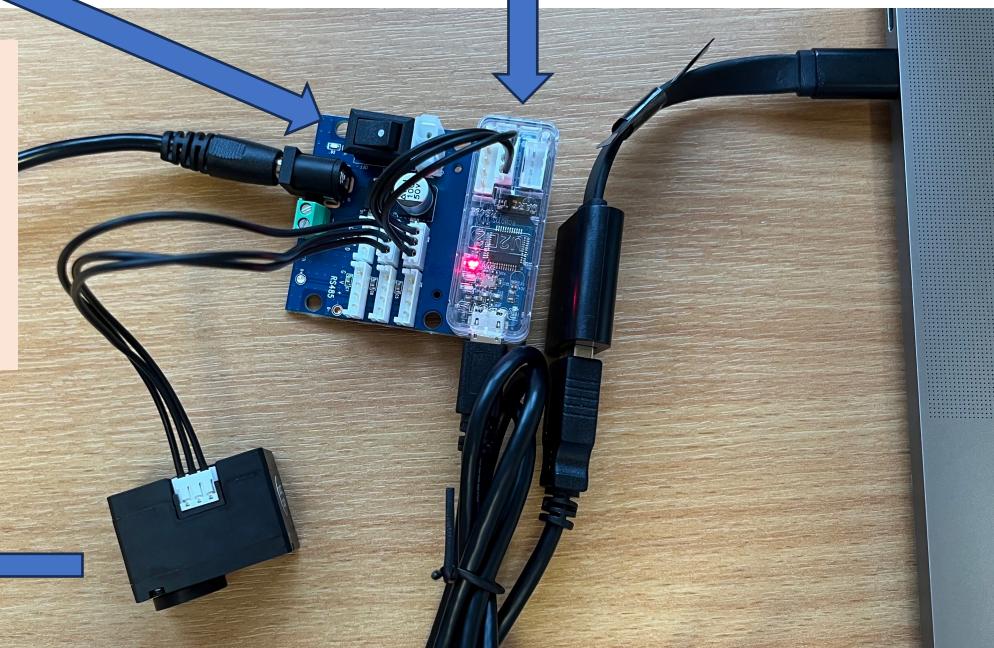
Input: 5.0 V recommended



↑Another option is directly supply power here
(Without using the PHB Set)

U2D2

- USB-DYNAMIXEL signal converter
- Driver available for Windows/Linux



SDK (library)

DYNAMIXEL SDK 3.7.31 is **already included** in the repository (DynamixelSDK-3.7.31 folder).

[Download here](#)

- DYNAMIXEL SDK 3.7.31.zip for ROS1, Python, C, C++, C#, MATLAB, LabView, Java
- DYNAMIXEL SDK 3.7.30.zip for ROS2
- ROS library (C++ library and Python module)

DYNAMIXEL SDK 3.7.31: Because all languages are included, import the language specific dynamixel_sdk (import python/src/dynamixel_sdk directory if using python)

Convinient software: Wizard 2.0

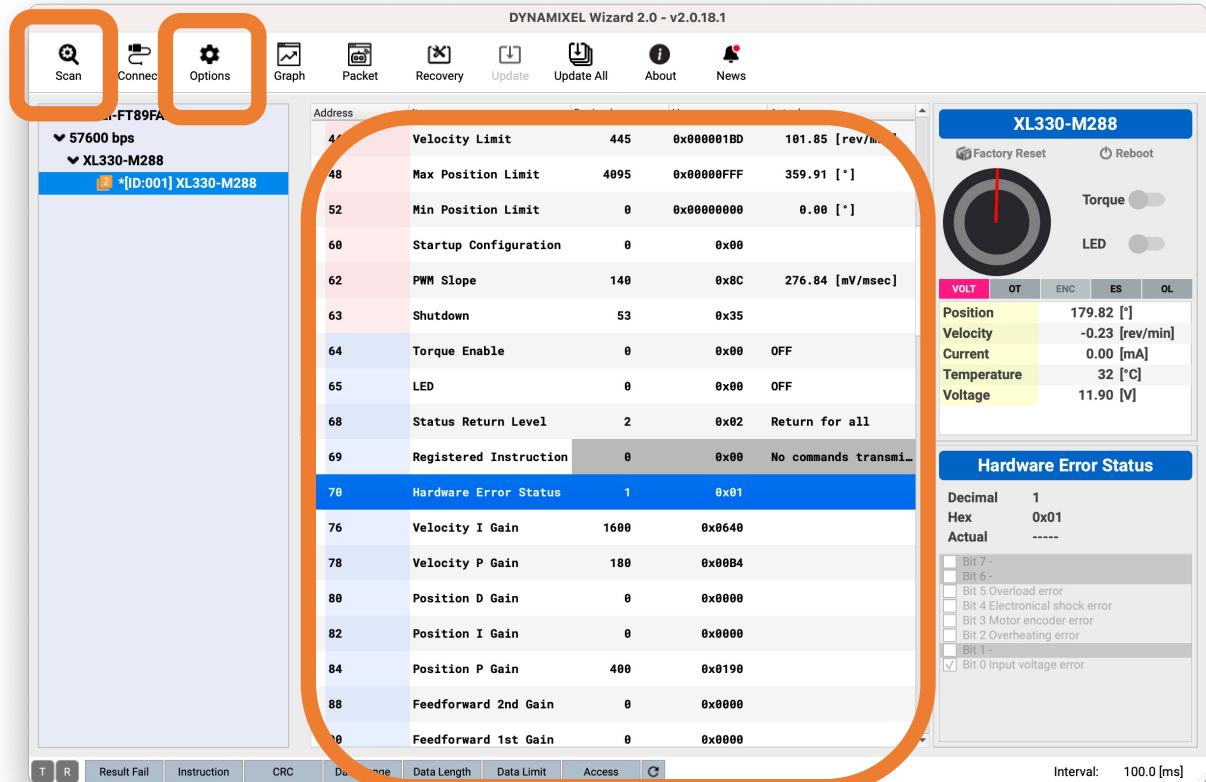
- Install [here](#)

How to use:

1. Set up the hardware.
2. In options, select scan range.
3. Click Scan to find connected device.
4. Read information from “control table”. Check details on dynamixel model page ([XL330-M288-T](#))

The application cannot be used while running code to communicate with DYNAMIXEL.

Scan and options



Control table

To use sample code from the Dynamixel library

- Sample codes are available in DynamixelSDK-3.7.31/python/tests/protocol*
 - Check your dynamixel model's protocol on [ROBOTIS e-Manual website](#).

Following are things you might need to add or change.

1. Import dynamixel_sdk from the appropriate source

```
os.sys.path.append('./DynamixelSDK-3.7.31/python/src')          # Path setting
from dynamixel_sdk import *                                     # Uses Dynamixel SDK library
```

2. Check all information matches your Dynamixel model.

Check which port is used by Dynamixel and change the DEVICENAME (refer to next page).

```
# Control table address
ADDR_PRO_TORQUE_ENABLE      = 64          # Control table address is different in Dynamixel model
ADDR_PRO_GOAL_POSITION       = 116
ADDR_PRO_PRESENT_POSITION    = 132

# Protocol version
PROTOCOL_VERSION             = 2.0         # See which protocol version is used in the Dynamixel

# Default setting
DXL_ID                       = 1           # Dynamixel ID : 1
BAUDRATE                      = 57600       # Dynamixel default baudrate : 57600
DEVICENAME                    = '/dev/tty.usbserial-FT89FANK' # Check which port is being used on your controller
                                         # ex) Windows: "COM1"   Linux: "/dev/ttyUSB0" Mac: "/dev/tty.usbserial-a"

TORQUE_ENABLE                 = 1           # Value for enabling the torque
TORQUE_DISABLE                = 0           # Value for disabling the torque
DXL_MINIMUM_POSITION_VALUE    = 10          # Dynamixel will rotate between this value
DXL_MAXIMUM_POSITION_VALUE    = 4095        # and this value (note that the Dynamixel would not move when the position value is
                                             # out of this range. Check product specification for the detailed info)
DXL_MOVING_STATUS_THRESHOLD   = 10          # Dynamixel moving status threshold

index = 0
dxl_goal_position = [DXL_MINIMUM_POSITION_VALUE, DXL_MAXIMUM_POSITION_VALUE]          # Goal position
```

3. Run the code! [Sample code explaination here](#)

Check which port is used by Dynamixel

- On Mac, type `ls /dev/tty*` in terminal to check connections. Easiest: plug in and out to see which one disappeared. It is the Dynamixel connection.
 - On Windows, Control Panel → Hardware and Sound → Device Manager. Find the corresponding COM port.

```
# Default setting
DXL_ID          = 1                      # Dynamixel ID : 1
BAUDRATE        = 57600                   # Dynamixel default baudrate : 57600
DEVICENAME      = '/dev/tty.usbserial-FT89FANK'    # Check which port is being used on your controller
                                                # ex) Windows: "COM1"    Linux: "/dev/ttyUSB0" Mac: "/dev/tty.usbserial-*"
```

Sample code explaination

`dynamixel_sdk_sample_read_write.py` is a sample code given by dynamixel (modified) that you can run for position control.

`my_current_control.py` is a sample code for current control, which uses self-defined functions in `my_utils.py` and parameter definitions in `my_global_variables_XL330M288T.py`.

Steps in code to do position or current control is:

1. Set control mode to position/current control
2. Enable torque
3. Set goal position/current
4. Read present position/current to check
5. Disable torque after the task is done

Reading and writing information

The code to control dynamixel is basically reading and writing information at specific addresses which are defined in control table on each model's page (ex. [XL330-M288-T](#)).

Size (Bytes) of the data is also specified on the control table.

For example, use `information, result, error = read4ByteTxRx(portHandler, Dynamixel_ID, Address_number)` to read a 4 byte information from `Address_number`.

Use `result, error = packetHandler.write2ByteTxRx(portHandler, Dynamixel_ID, Address_number, information)` to write information to `Address_number` which has a 2 byte size.

Enable torque

You need to 'enable torque' (writing TORQUE_ENABLE to ADDR_TORQUE_ENABLE) to access dynamixel internal information.

However, after enabling torque, a lot of settings are locked (you cannot write information to change them). To see what information is locked: enable torque on DYNAMIXEL Wizard application and check the control table on the app.

Thus,

- Set operating mode and various limits etc. (all the parameters that will be lock) before enabling torque.
- You might need to disable torque at the beginning of the code, if it was not disabled properly after last operation.