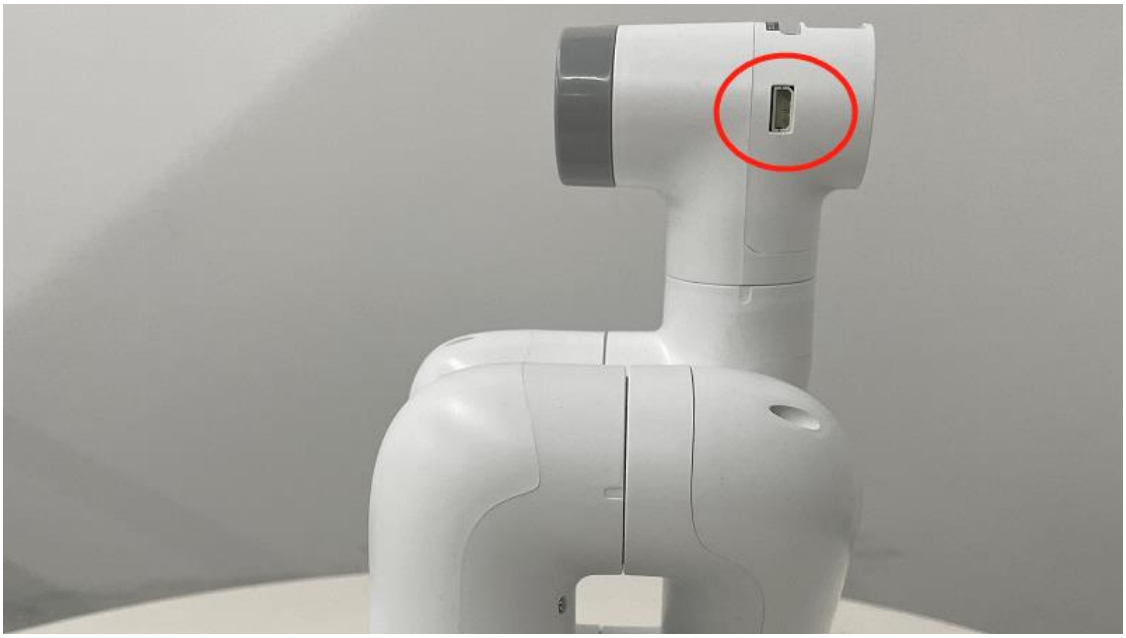
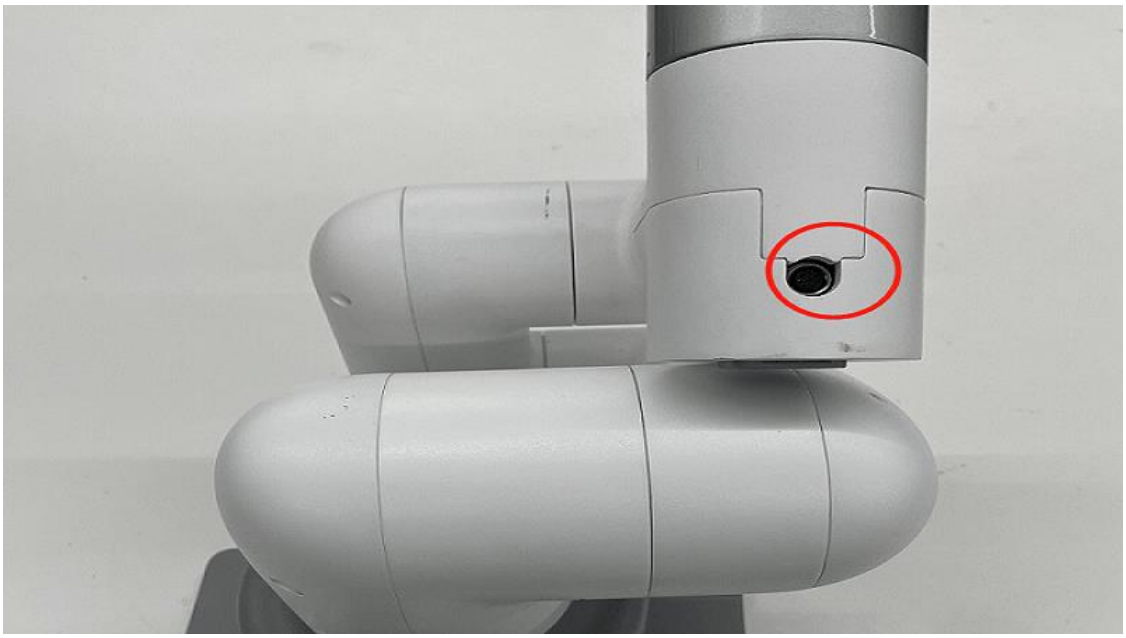
Gripper Control

First install and connect the gripper onto the robot arm. Different types of gripper is compatible with different types of robots. Refer to **2.8 Accessories** for more information.

> **Notice:**
>
> For MyCobot 280, the adaptive gripper is attached to Atom.
>
> 
>
> The electric gripper is attached to 495 port.
>
> 
>
> * MyCobot 280-m5 is not compatible with electric gripper, and MyCobot 320-m5 is only compatible with electric gripper.

myCobot

Controlling Gripper

is_gripper_moving()

- **Function:** determine whether the gripper is running
- Return Value:
  - `0`: means that the gripper is not running
  - `1`: means that the gripper is running
  - `-1`: means an error

set_gripper_value(value, speed, gripper_type=None)

- **Function:** to make the gripper rotate to a specified position at a specified speed
- Description of parameters:
  - `value`: means the position to be reached by the gripper, ranging from 0 to 256.
  - `speed`: means the speed of rotation, ranging from 0 to 100
  - `gripper_type`

    :
    - 1 - Adaptive gripper (default to adaptive gripper)
    - 3 - Parallel gripper
- **Return Value:** None

get_gripper_value(gripper_type=None)

- **Function:** to obtain the data information on the `encoder` of the gripper
- Description of parameters:
  - `gripper_type`

    :
    - 1 - Adaptive gripper (default to adaptive gripper)
    - 3 - Parallel gripper
- **Return Value:** to obtain the data information of the gripper

set_gripper_state(flag, speed, _type=None)

- **Function:** to make the gripper enter a specified state at a specified speed
- Description of parameters:
  - `flag`: 1 means that the gripper is closed, 0 means thatthe gripper is open.
  - `speed`: means the speed at which the gripper reaches a specified state, ranging from 0 to 100
  - `_type`

    :
    - 1 - Adaptive gripper (default to adaptive gripper)
    - 2 - 5 Fingered Spirit Skilled Hand
    - 3 - Parallel gripper

- **Return value:** None

set_gripper_int()

- **Function:** to set the initial position of the gripper. Set the current position to 2048
- **return value:** None

Simple Demo

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD  # When using the Raspberry Pi version of
mycobot, these two variables can be referenced to initialize MyCobot
import time
#Enter the above code to import the packages required by the project

def gripper_test(mc):
    print("Start check IO part of api\n")
    # Check if the gripper is moving
    flag = mc.is_gripper_moving()
    print("Is gripper moving: {}".format(flag))
    time.sleep(1)

    # Set the current position to (2048).
    # Use it when you are sure you need it.
    # Gripper has been initialized for a long time. Generally, there
    # is no need to change the method.
    # mc.set_gripper_ini()
    # Set joint point 1 to rotate to the position of 2048
    mc.set_encoder(1, 2048)
    time.sleep(2)
    # Set six joint positions and let the robotic arm rotate to this position at
a speed of 20
    mc.set_encoders([1024, 1024, 1024, 1024, 1024, 1024], 20)
    time.sleep(3)
    # Get the position information of joint point 1
    print(mc.get_encoder(1))
    # Set the gripper to rotate to the position of 2048
    mc.set_encoder(7, 2048)
    time.sleep(3)
    # Set the gripper to rotate to the position of 1300
    mc.set_encoder(7, 1300)
    time.sleep(3)

    # Let the gripper reach the state of 2048 at a speed of 70
    mc.set_gripper_value(2048, 70)
    time.sleep(3)
    # Let the gripper reach the state of 1500 at a speed of 70
    mc.set_gripper_value(1500, 70)
    time.sleep(3)

    # Set the state of the gripper to quickly open the gripper at a speed of 70
    mc.set_gripper_state(0, 70)
    time.sleep(3)
    # Set the state of the gripper so that it quickly closes the gripper at a
speed of 70
```

```
    mc.set_gripper_state(1, 70)
    time.sleep(3)

    # Get the value of the gripper
    print("")
    print(mc.get_gripper_value())


if __name__ == "__main__":
    # MyCobot class initialization requires two parameters:
    #   The first is the serial port string, such as:
    #       linux:  "/dev/ttyUSB0"
    #          or "/dev/ttyAMA0"
    #       windows: "COM3"
    #   The second is the baud rate::
    #       M5 version is:  115200
    #
    #    Example:
    #       mycobot-M5:
    #           linux:
    #              mc = MyCobot("/dev/ttyUSB0", 115200)
    #          or mc = MyCobot("/dev/ttyAMA0", 115200)
    #           windows:
    #              mc = MyCobot("COM3", 115200)
    #       mycobot-raspi:
    #           mc = MyCobot(PI_PORT, PI_BAUD)
    #
    # Initialize a MyCobot object
    # Create object code here for Raspberry Pi version below
    mc = MyCobot(PI_PORT, PI_BAUD)
    # make it move to zero position
    mc.set_encoders([2048, 2048, 2048, 2048, 2048, 2048], 20)
    time.sleep(3)
    gripper_test(mc)
```

myBuddy

Controlling Gripper

is_gripper_moving(id)

- **Function** Judge whether the gripper is moving or not
- **Parameters**

  id – 1/2 (L/R)
- **Returns**
  - 0 - not moving
  - 1 - is moving
  - -1 - error data

set_gripper_value(id, value, speed)

- **Function** Set gripper value
- **Parameters**

- **id** – 1/2 (L/R)
- **value** (*int*) – 0 ~ 100
- **speed** (*int*) – 0 ~ 100

get_gripper_value(id)

- **Function** Get the value of gripper.
- **Parameters**

  id – 1/2 (L/R)
- **Returns**

  gripper value (int)

set_gripper_calibration(id)

- **Function** Set the current position to zero, set current position value is 2048.
- **Parameters**

  id – 1/2 (L/R)

is_gripper_moving(id)

- **Function** Judge whether the gripper is moving or not
- **Parameters**

  id – 1/2 (L/R)
- **Returns**
  - 0 - not moving
  - 1 - is moving
  - -1 - error data

myPalletizer

Simple Demo

```
from pymycobot.mypalletizer import MyPalletizer
from pymycobot.genre import Angle
import time
#Enter the above code to import the packages required by the project

# initiate MyPalletizer
mc = MyPalletizer("COM3", 115200)

# let joint2 move to 30 degree at the speed of 50
mc.send_angle(2,30,50)
# waite for 2 seconds
time.sleep(2)

#set a variable num, and then set a loop
num = 5
while num > 0:
    #let gripper open at the speed of 70
    mc.set_gripper_state(0,70)
```

```
    # waite for 2 seconds
    time.sleep(2)
    # let gripper close at the speed of 70
    mc.set_gripper_state(1, 70)
    # waite for 2 seconds
    time.sleep(2)
    num -= 1
```

myArm

Simple Demo

```
from pymycobot.myarm import MyArm
import time
#Enter the above code to import the packages required by the project

def gripper_test(mc):
    print("Start check IO part of api\n")
    # Check if the gripper is moving
    flag = mc.is_gripper_moving()
    print("Is gripper moving: {}".format(flag))
    time.sleep(1)

    # Set the current position to (2048).
    # Use it when you are sure you need it.
    # Gripper has been initialized for a long time. Generally, there
    # is no need to change the method.
    # mc.set_gripper_ini()
    # Set joint point 1 to rotate to the position of 2048
    mc.set_encoder(1, 2048)
    time.sleep(2)
    # Set six joint positions and let the robotic arm rotate to this position at
a speed of 20
    mc.set_encoders([1024, 1024, 1024, 1024, 1024, 1024,1024], 20)
    time.sleep(3)
    # Get the position information of joint point 1
    print(mc.get_encoder(1))
    # Set the gripper to rotate to the position of 2048
    mc.set_encoder(7, 2048)
    time.sleep(3)
    # Set the gripper to rotate to the position of 1300
    mc.set_encoder(7, 1300)
    time.sleep(3)

    # Let the gripper reach the state of 2048 at a speed of 70
    mc.set_gripper_value(2048, 70)
    time.sleep(3)
    # Let the gripper reach the state of 1500 at a speed of 70
    mc.set_gripper_value(1500, 70)
    time.sleep(3)

    # Set the state of the gripper to quickly open the gripper at a speed of 70
    mc.set_gripper_state(0, 70)
    time.sleep(3)
```

```
    # Set the state of the gripper so that it quickly closes the gripper at a
speed of 70
    mc.set_gripper_state(1, 70)
    time.sleep(3)

    # Get the value of the gripper
    print("")
    print(mc.get_gripper_value())


if __name__ == "__main__":

    # Initialize a MyArm object
    mc = MyArm("/dev/ttyAMA0", 115200)
    # make it move to zero position
    mc.set_encoders([2048, 2048, 2048, 2048, 2048, 2048, 2048], 20)
    time.sleep(3)
    gripper_test(mc)
```