

# Videos and Codes for Display

Videos given below are for reference.

**Notice:** The baud rates are different depending on the type of device. Before using them, refer to the information related thereto. The serial port number can be viewed through [calculator device manager](#) or a serial helper.

## 1 Controlling RGB Light Panel

### myCobot

```
from pymycobot.mycobot import MyCobot

from pymycobot import PI_PORT, PI_BAUD      # When using the Raspberry Pi
version of mycobot, you can refer to these two variables to initialize MyCobot,
if not, you can omit this line of code
import time
#The above needs to be written at the beginning of the code, which means
importing the project package

# 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 windows version
mc = MyCobot("COM3", 115200)

i = 7
#loop 7 times
while i > 0:
    mc.set_color(0,0,255) #blue light on
    time.sleep(2)        #wait for 2 seconds
    mc.set_color(255,0,0) #red light on
    time.sleep(2)        #wait for 2 seconds
    mc.set_color(0,255,0) #green light on
    time.sleep(2)        #wait for 2 seconds
```

```
i -= 1
```

## myArm

```
from pymycobot.myarm import MyArm
import time
#The above needs to be written at the beginning of the code, which means
importing the project package

# Initialize a MyArm object
mc = MyArm("/dev/ttyAMA0", 115200)

i = 7
#loop 7 times
while i > 0:
    mc.set_color(0,0,255) #blue light on
    time.sleep(2)        #wait for 2 seconds
    mc.set_color(255,0,0) #red light on
    time.sleep(2)        #wait for 2 seconds
    mc.set_color(0,255,0) #green light on
    time.sleep(2)        #wait for 2 seconds
    i -= 1
```

## 2 Controlling Arms to Move Them to Starting Point

```
from pymycobot.mycobot import MyCobot
from pymycobot import PI_PORT, PI_BAUD    # When using the Raspberry Pi
version of mycobot, you can refer to these two variables to initialize MyCobot

# 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
mc = MyCobot(PI_PORT, PI_BAUD)

# Check whether the program can be burned into the robot arm
```

```

if mc.is_controller_connected() != 1:
    print("Please connect the robot arm correctly for program writing")
    exit(0)

# Fine-tune the robotic arm to ensure that all the bayonets are aligned in the
adjusted position
# Subject to the alignment of the mechanical arm bayonet, this is only a case
mc.send_angles([0, 0, 0, 0, 0, 0], 30)

# To calibrate the position at this time, the calibrated angular position
represents [0,0,0,0,0,0], and the potential value represents
[2048,2048,2048,2048,2048,2048]
# The for loop is equivalent to the method set_gripper_ini()
#for i in range(1, 7):
    #mc.set_servo_calibration(i)

```

### 3 Single-Joint Motion

```

from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle
import time

# 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 for Raspberry Pi
# mc = MyCobot(PI_PORT, PI_BAUD)
# Create object code for M5
mc=MyCobot('COM3',115200)

# Robotic arm recovery
mc.send_angles([0, 0, 0, 0, 0, 0], 40)
time.sleep(3)

# Control joint 3 to move 70°
mc.send_angle(Angle.J3.value,70,40)
time.sleep(3)

```

```

# Control joint 4 movement -70°
mc.send_angle(Angle.J4.value,-70,40)
time.sleep(3)

# Control joint 1 to move 90°
mc.send_angle(Angle.J1.value,90,40)
time.sleep(3)

# Control joint 5 movement -90°
mc.send_angle(Angle.J5.value,-90,40)
time.sleep(3)

# Control joint 5 to move 90°
mc.send_angle(Angle.J5.value,90,40)
time.sleep(3)

```

## 4 Multi-Joint Motion

```

import time
from pymycobot import MyCobot
# 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 for Raspberry Pi
# mc = MyCobot(PI_PORT, PI_BAUD)
# Create object code for M5
mc=MyCobot('COM3',115200)

# Robotic arm recovery
mc.send_angles([0, 0, 0, 0, 0, 0], 50)
time.sleep(2.5)

# Control different angles of rotation of multiple joints
mc.send_angles([90,45,-90,90,-90,90],50)
time.sleep(2.5)

```

```
# Return the robotic arm to zero
mc.send_angles([0,0,0,0,0,0],50)
time.sleep(2.5)

# Control different angles of rotation of multiple joints
mc.send_angles([-90,-45,90,-90,90,-90],50)
time.sleep(2.5)
```

## 5 Swaying Arms Left and Right

```
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle
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

# Initialize a MyCobot object
mc = MyCobot("COM3", 115200)
# Get the coordinates of the current location
angle_datas = mc.get_angles()
print(angle_datas)

#By passing the angle parameter, let each joint of the robotic arm move to the
position
mc.send_angles([0, 0, 0, 0, 0, 0], 50)
print(mc.is_paused())
# Set the waiting time to ensure that the robotic arm has reached the specified
position
# while not mc.is_paused():
time.sleep(2.5)

# Move joint 1 to the 90 position
mc.send_angle(Angle.J1.value, 90, 50)

# Set the waiting time to ensure that the robotic arm has reached the specified
position
time.sleep(2)

# set loop times
num = 5

# The following code can make the robotic arm swing left and right
while num > 0:
    # Move joint 2 to the 50 position
    mc.send_angle(Angle.J2.value, 50, 50)

    # Set the waiting time to ensure that the robotic arm has reached the
specified position
    time.sleep(1.5)

    # Move joint 2 to the -50 position
    mc.send_angle(Angle.J2.value, -50, 50)
```

```

    # Set the waiting time to ensure that the robotic arm has reached the
    specified position
    time.sleep(1.5)

    num -= 1

# Make the robotic arm retract. You can manually swing the robotic arm, and then
use the get_angles() function to get the coordinate sequence, use this function
to let the robotic arm reach the position you want.
mc.send_angles([88.68, -138.51, 155.65, -128.05, -9.93, -15.29], 50)

# Set the waiting time to ensure that the robotic arm has reached the specified
position
time.sleep(2.5)

# Let the robotic arm relax, you can manually swing the robotic arm
mc.release_all_servos()

```

## 6 Let Robot Dance

```

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

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
    #
    # such as:
    #     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 for Raspberry Pi version below
    mc = MyCobot(PI_PORT, PI_BAUD)

    # set start time
    start = time.time()

    # Let the robotic arm reach the specified position
    mc.send_angles([-1.49, 115, -153.45, 30, -33.42, 137.9], 80)

```

```

# Determine if it reaches the specified position
while not mc.is_in_position([-1.49, 115, -153.45, 30, -33.42, 137.9], 0):
    # Return the robotic arm to motion
    mc.resume()
    # Let the robotic arm move for 0.5s
    time.sleep(0.5)
    # Pause arm movement
    mc.pause()
    # Determine if the move timed out
    if time.time() - start > 3:
        break

# set start time
start = time.time()
# Let the exercise last for 30 seconds
while time.time() - start < 30:
    # Let the robotic arm quickly reach this position
    mc.send_angles([-1.49, 115, -153.45, 30, -33.42, 137.9], 80)
    # Set the color of the light to [0,0,50]
    mc.set_color(0, 0, 50)
    time.sleep(0.7)
    # Let the robotic arm quickly reach this position
    mc.send_angles([-1.49, 55, -153.45, 80, 33.42, 137.9], 80)
    # Set the color of the light to [0,50,0]
    mc.set_color(0, 50, 0)
    time.sleep(0.7)

```

## 7 Controlling Gripper

```

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
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, 2048 will report
an error, so change it to 255
mc.set_gripper_value(255, 70)
time.sleep(3)
# Let the gripper reach the state of 1500 at a speed of 70, 1500 will report
an error, so change it to 255
mc.set_gripper_value(255, 70)
time.sleep(3)

num=5
while num>0:
    # 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 to quickly close the gripper at a speed
of 70
    mc.set_gripper_state(1, 70)
    time.sleep(3)
    num-=1

# 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
    #
    #   such as:
    #       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 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)
```