

Send data

1. API Introduction

Robot object initialization

```
mc = MyCobot("/dev/ttyUSB0", 1000000)
```

Among them, the first parameter ["/dev/ttyUSB0"] is the control serial port device number, which can be modified according to actual conditions. The second parameter [1000000] is the serial port communication baud rate, which cannot be changed.

The API related to controlling the robotic arm is:

send_angle(id, degree, speed)

Function: Control the angle of a joint of the robot arm.

Parameter explanation:

- **【id】** : The input range of id is 1~6, corresponding to the joints of the robotic arm from bottom to top.
- **【degree】** : Servo angle value.

Each servo angle range is different, please check the list below.

Servo ID	Min value	Max value
1	-168°	168°
2	-135°	135°
3	-150°	150°
4	-145°	145°
5	-165°	165°
6	-180°	180°

- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

send_angles(degrees, speed)

Function: Control the angles of the six joints of the robotic arm.

Parameter explanation:

- **【degrees】** : A list of six joint angles, for example degrees = [0.43, -0.52, 0.17, -1.14, -0.35, -45.08].

- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

send_coord(id, coord, speed)

Function: Control the robotic arm to move toward a certain axis.

Parameter explanation:

- **【id】** : The input range of id is 1~3, corresponding to the robotic arm coordinate axis, 1: X axis, 2: Y axis, 3: Z axis.
- **【coord】** : Robotic arm coordinate parameters, please refer to the table below for input range.

Coordinate	Min value	Max value
x	-281 mm	281 mm
y	-281 mm	281 mm
z	-70 mm	412 mm

- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.

send_coords(coords, speed)

Function: Control the robotic arm to move toward a certain coordinate.

Parameter explanation:

- **【coords】** : Robotic arm coordinate parameters [x, y, z, rx, ry, rz].

The three parameters x, y, z represent the coordinates of the end of the robotic arm, and rx, ry, rz represent the orientation of the gripper.

Coordinate	Min value	Max value
x	-281 mm	281 mm
y	-281 mm	281 mm
z	-70 mm	412 mm
rx	-180°	180°
ry	-180°	180°
rz	-180°	180°

- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

set_encoder(id, encoder, speed)

Function: Control the potential value of a joint of the robotic arm.

Parameter explanation:

- **【id】** : The input range of id is 1~6, corresponding to the joints of the robot from bottom to top.
- **【encoder】** : Servo potential value, the value range is: 0~4096.
- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

set_encoders(encoders, speed)

Function: Control the potential values of the six joints of the robot.

Parameter explanation:

- **【encoders】** : List of robot arm joint potential values, e.g. encoders = [2048, 2048, 2048, 2048, 2048, 2048].
- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

send_radians(radians, speed)

Function: Control the arc value of the six joints of the robotic arm.

Parameter explanation:

- **【radians】** : List of robot arm joint arc values, e.g. radians = [0.008, -0.009, 0.003, -0.02, -0.006, -0.787].
- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

set_gripper_value(gripper, speed)

Function: Control the release and clamping of the robotic arm gripper.

Parameter explanation:

- **【gripper】** : The value of the robotic arm joint gripper is in the range of 0~100. 100 means the gripper is opened to the maximum angle.
- **【speed】** : Control the running speed, the range is [1-100]. The larger the value, the faster the movement.
- Return value: None.

2.About code

Code path: ~/jetcobot_ws/src/jetcobot_ctrl/scripts/send_data.ipynb

```
#!/usr/bin/env python3
#coding=utf-8
import os
import time
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle
from pymycobot.genre import Coord
```

Initialize the robot object

```
mc = MyCobot(str(os.getenv('MY_SERIAL')), 1000000)
speed = 50
```

Control joint 1° to 90°

```
id = Angle.J1.value
degree = 90
mc.send_angle(id, degree, speed)
```

Control the angles of the six joints of the robotic arm

```
degrees = [0.43, -0.52, 0.17, -1.14, -0.35, -45.08]
mc.send_angles(degrees, speed)
```

Control the Z-axis coordinate of the robotic arm.

Note: If the current robot arm posture coordinate is unreachable, the setting fails and the robot arm will not move.

```
id = Coord.Z.value
coord = 350
mc.send_coord(id, coord, speed)
```

Control the robot arm to move to a specific coordinate point.

Note: If the given coordinate is unreachable, the setting fails and the robotic arm will not move.

```
coords = [49.6, -63.3, 419, -92.11, -45.07, -88.41]
mc.send_coords(coords, speed)
```

Controls the potential value of the robotic arm.

It is rarely used in general.

```
id = Angle.J1.value
encoder = 1500
mc.set_encoder(id, encoder, speed)
```

```
encoders = [2048, 2048, 2048, 2048, 2048, 2048]  
mc.set_encoders(encoders, speed)
```

Controls the arc value of the robotic arm.

```
radians = [0.008, -0.009, 0.003, -0.02, -0.006, -0.787]  
mc.send_radians(radians, speed)
```

Controls the gripper angle of the robotic arm.

```
gripper = 100  
mc.set_gripper_value(gripper, speed)
```

3. Run program

Click the run button on jupyterlab to run the relevant programs.

