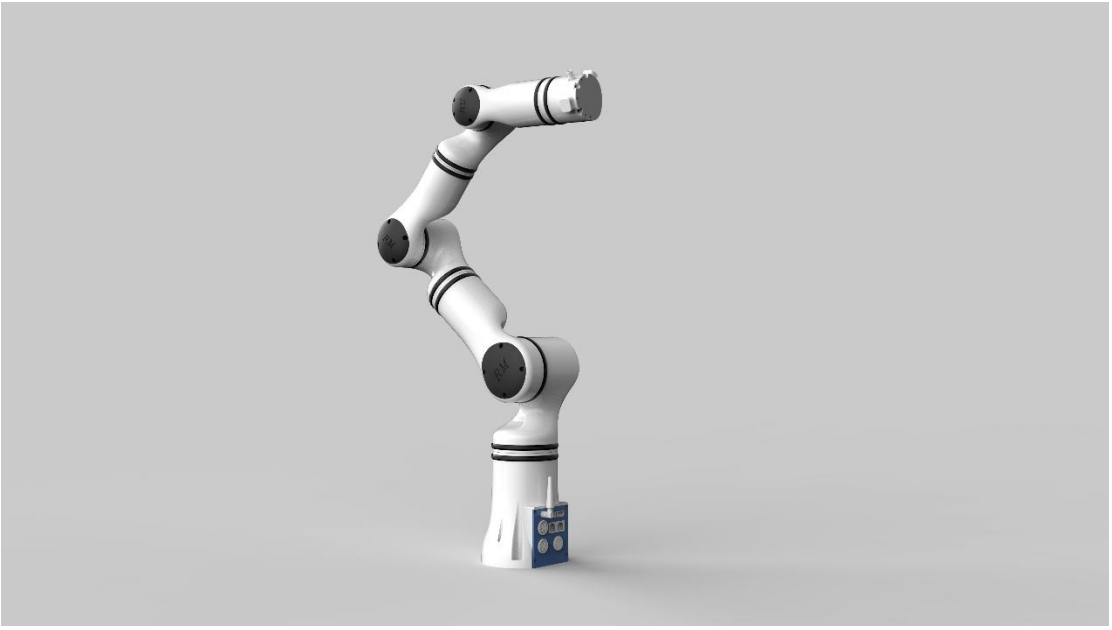
睿尔曼机械臂接口函数说明（Python）V1.7



睿尔曼智能科技（北京）有限公司

文件修订记录:

版本号	时间	备注
V1.0	2023-10-20	拟制
V1.1	2023-11-27	文档勘误 修复 Python API 读多圈、多个保持寄存器报错 修复 Python API 透传力位混合补偿 Force_Position_Move_Pose 报错 完善部分函数注释
V1.2	2023-12-27	新增电子围栏相关接口 增加 modbus-TCP 主站和 modbus-RTU 从站接口 新增查询示教参考坐标系接口 新增自碰撞安全检测相关接口 新增读取软件信息接口 新增关节驱动器转速、加速度以及最大最小限位设置查询接口 优化开始复合模式拖动示教、movej、movel、movec、movej_p、一键设置关节限位等接口 新增查询夹爪信息接口 新增设置、查询机械臂仿真/真实模式
V1.3	2024-3-30	适配 GEN72、ECO63 机械臂 运动接口优化，适配控制器 1.5.0 软件版本 修复灵巧手相关接口阻塞模式不生效问题 修复夹爪夹取松开速度低时执行成功返回 7 的问题，新增超时时间设置 修改 SensorType 枚举项名称 B、ZF、SF 为_B、_ZF、_SF 优化电子围栏与虚拟墙数据管理接口 新增工作坐标系和工具坐标系的更新接口 新增工具坐标系包络参数相关接口

		新增虚拟墙相关接口 新增在线编程文件信息修改接口 新增 IO 控制默认运行在线编程文件接口 新增全局点位相关接口
V1.4	2024-5-30	新增样条曲线运动 Moves 新增读多个输入寄存器接口 Read_Multiple_Input_Registers Send_TrajectoryFile 发送在线编程文件新增保存到控制器 id 参数
V1.5	2024-7-3	适配 GEN72 型号机械臂
V1.6	2024-8-5	修复使用控制器 1.6.0 时无法使用 UDP 的问题 修改运动接口交融半径
V1.7	2024-8-19	修复交融半径调用失败 新增计算平移、旋转运动位姿算法接口 调整 movej、movel、movc 等运动接口参数顺序 添加交融半径使用注意事项 适配控制器 1.6.0 主动上报接口自定义项配置、获取及上报信息获取

目录

Python 语法简介

Python 是一种流行的高级编程语言，它易于学习和阅读，拥有丰富的库和生态系统。以下是 Python 的一些基本语法要点：

1. **注释**：用`#`符号表示单行注释，注释用于提供代码的解释和文档说明。

```
# 这是一个单行注释
```

多行注释可以使用三个引号（单引号或双引号）括起来：

```
'''
这是一个
多行注释
'''
```

2. **缩进**：Python 使用缩进来表示代码块，而不是花括号或其他符号。通常使用 4 个空格作为缩进级别。

```
if True:
    print("这是缩进的代码块")
```

3. **变量**：变量用于存储数据，无需显式声明类型，Python 会自动识别。变量名区分大小写。

```
x = 10
name = "realman"
```

4. **数据类型**：Python 支持多种数据类型，包括整数（int）、浮点数（float）、字符串（str）、布尔值（bool）、列表（list）、元组（tuple）、集合（set）、字典（dict）

等。

```
num = 42

pi = 3.14

message = "Hello, World!"

is_true = True

my_list = [1, 2, 3]

my_tuple = (1, 2, 3)

my_set = {1, 2, 3}

my_dict = {"name": "Alice", "age": 30}
```

5. **运算符：**Python 支持各种运算符，包括算术运算符（+、-、*、/、%）、比较运算符（==、!=、<、>）、逻辑运算符（and、or、not）等。

```
x = 5

y = 3

result = x + y  # 加法

is_equal = x == y  # 相等比较

is_greater = x > y  # 大于比较

logical_result = (x > 0) and (y < 10)  # 逻辑与运算
```

6. **条件语句：**使用`if`、`elif`（可选）和`else`（可选）来执行条件分支。

```
if condition:

    # 如果条件为真，执行这里的代码

elif another_condition:

    # 如果上面条件不满足，且这个条件为真，执行这里的代码
```

```
else:
    # 如果上面的条件都不满足，执行这里的代码
```

7. **循环：**Python 支持`for`和`while`循环。

```
for i in range(5):  # 使用 range 函数生成一组数字
    print(i)


while condition:
    # 当条件为真时，执行这里的代码
```

8. **函数：**使用`def`关键字定义函数。

```
def greet(name):
    print("Hello, " + name)


greet("realman")
```

9. **列表推导式：**用于创建新的列表。

```
squares = [x**2 for x in range(5)]
```

10. **异常处理：**使用`try`和`except`捕获和处理异常。

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("除以零错误")
```

这只是 Python 的一些基本语法要点。要深入了解 Python 语言，可以查看 Python 的官方文档或参考更详细的教程和学习资源。

1.简介

本文档为 Python 二次开发接口说明文档。

支持平台：Linux、Windows

Python 版本：python 3.10 及以上

2.功能介绍

Python API 通过调用 C 版本的睿尔曼机械臂接口函数库实现，Windows 和 Linux 环境下都可使用，只需替换对应的库文件即可。

接口函数将用户指令封装成标准的 JSON 格式下发给机械臂，并解析机械臂回传的数据提供给用户。

接口函数基于 TCP/IP 协议编写，其中：

机械臂默认 IP 地址：192.168.1.18，端口号：8080

无论是 WIFI 模式还是以太网口模式，机械臂均以该 IP 和端口号对外进行 socket 通信，机械臂为 Server 模式，用户为 Client 模式。

3.使用说明

提供的 Python 包 robotic_arm_package 内包括三个文件：

1. Log_setting.py：日志的配置文件；
2. robotic_arm.py：调用 C 库的代码封装。
3. RM_Base.dll：C 的接口函数库，该库的版本需要与平台以及 Python 版本对应，**用户可根据需要自行更换**。例如在 Linux-x86 环境使用 Python，则应

在 C 的文件夹中选择 Linux_x86 的 release 版本的库：

以下是一些 Python 包的基础使用说明：

1. 导入包或模块： 使用 import 语句导入模块或包，以便在代码中使用它们。例如：

```
from robotic_arm_package.robotic_arm import *
```

这将导入 robotic_arm_package 中的 robotic_arm 模块，并使用 * 导入了该模块中的所有内容，使用户能够在代码中使用其中的函数和变量。

2. 使用包或模块中的函数或变量：导入包之后可以使用其中定义的函数、变量或类。例如，要使用 Arm 类中的 API_Version()查询 API 版本：

```
from robotic_arm_package.robotic_arm import *

robot= Arm(RM65, '192.168.1.18')

# API 版本信息

print(robot.API_Version())
```

4.数据类型说明

4.1. 控制器错误类型

序号	错误代码（16 进制）	错误内容

1	0x0000	系统正常
2	0x1001	关节通信异常
3	0x1002	目标角度超过限位
4	0x1003	该处不可达，为奇异点
5	0x1004	实时内核通信错误
6	0x1005	关节通信总线错误
7	0x1006	规划层内核错误
8	0x1007	关节超速
9	0x1008	末端接口板无法连接
10	0x1009	超速度限制
11	0x100A	超加速度限制
12	0x100B	关节抱闸未打开
13	0x100C	拖动示教时超速
14	0x100D	机械臂发生碰撞
15	0x100E	无该工作坐标系
16	0x100F	无该工具坐标系
17	0x1010	关节发生掉使能错误

4.2. 关节错误类型

序号	错误代码（16 进制）	错误内容
1	0x0000	关节正常
2	0x0001	FOC 错误

3	0x0002	过压
4	0x0004	欠压
5	0x0008	过温
6	0x0010	启动失败
7	0x0020	编码器错误
8	0x0040	过流
9	0x0080	软件错误
10	0x0100	温度传感器错误
11	0x0200	位置超限错误
12	0x0400	关节 ID 非法
13	0x0800	位置跟踪错误
14	0x1000	电流检测错误
15	0x2000	抱闸打开失败
16	0x4000	位置指令阶跃警告
17	0x8000	多圈关节丢圈数
18	0xF000	通信丢帧

4.3. API 错误类型

除实例化机械臂对象接口外，其他的接口返回值都是一个元组，Get 类的接口的返回值形式（RetVal，data），RetVal 为错误码，返回 0 为成功，返回其他值可查询下表；data 为获取到的数据。

如调用 Get_Current_Arm_State 接口返回（RetVal, joint, pose, arm_err,

sys_err）：

调用成功时返回 ：(0, [0, 0, 0, 0, 0, 0], [0, 0, 0.857, 0, 0, 3.145], 0, 0)。

由于网络连接问题，获取机械臂状态失败时返回：('7: SOCKET_TIME_OUT', [0.0, 0.0, 0.0, 0.0, 0.0, 0.0], [0.0, 0.0, 0.0, 0.0, 0.0, 0.0], 0, 0)。

序号	错误代码（16 进制）	错误内容
1	0x0000	系统运行正常
2	0x0001	消息请求返回 FALSE
3	0x0002	机械臂未初始化或输入型号非法
4	0x0003	非法超时时间
5	0x0004	Socket 初始化失败
6	0x0005	Socket 连接失败
7	0x0006	Socket 发送失败
8	0x0007	Socket 通讯超时
9	0x0008	未知错误
10	0x0009	数据不完整
11	0x000A	数组长度错误
12	0x000B	数据类型错误
13	0x000C	型号错误
14	0x000D	缺少回调函数
15	0x000E	机械臂异常停止
16	0x000F	轨迹文件名称过长

17	0x0010	轨迹文件校验失败
18	0x0011	轨迹文件读取失败
19	0x0012	控制器忙，请稍后再试
20	0x0013	非法输入
21	0x0014	数据队列已满
22	0x0015	计算失败
23	0x0016	文件打开失败
24	0x0017	力控标定手动停止
25	0x0018	没有可保存轨迹
26	0x0019	UDP 监听接口运行报错

4.4. 结构化数据的类定义

结构化数据的类定义列举了一些常用结构体，其他结构体和数据类型请查看 robotic_arm.py。

4.4.1. 位姿结构体 Pose

```
class Pose(ctypes.Structure):

    _fields_ = [

        ('position', Pos),   # 位置

        ('quaternion', Quat),    # 四元数
```

```
        ('euler', Euler)      # 欧拉角

    ]
```

结构体成员

position

位置坐标，float 类型，单位：m

quaternion

四元数

euler

姿态角，float 类型，单位：rad

4.4.2. 坐标系结构体 FRAME

```
class FRAME(ctypes.Structure):

    _fields_ = [('frame_name', FRAME_NAME),   # 坐标系名称

                ('pose', Pose),   # 坐标系位姿

                ('payload', ctypes.c_float),   # 坐标系末端负载重量

                ('x', ctypes.c_float),   # 坐标系末端负载位置
```

```
                ('y', ctypes.c_float),   # 坐标系末端负载位置

                ('z', ctypes.c_float)]   # 坐标系末端负载位置
```

结构体成员

frame_name

坐标系名称

Pose

坐标系位姿

payload

坐标系末端负载重量　单位 g

x,y,z

坐标系末端负载位置，单位：m

4.4.3. 关节状态结构体 JOINT_STATE

```
class JOINT_STATE(ctypes.Structure):

    _fields_ = [

        ("temperature", ctypes.c_float * ARM_DOF),

        ("voltage", ctypes.c_float * ARM_DOF),

        ("current", ctypes.c_float * ARM_DOF),
```

```
        ("en_state", ctypes.c_byte * ARM_DOF),

        ("err_flag", ctypes.c_uint16 * ARM_DOF),

        ("sys_err", ctypes.c_uint16),

    ]
```

结构体成员

temperature

关节温度，float 类型，单位：摄氏度

voltage

关节电压，float 类型，单位：V

current

关节电流，float 类型，单位：mA

en_state

使能状态

err_flag

关节错误代码，unsigned int 类型

sys_err

机械臂系统错误代码，unsigned int 类型

4.4.4. 无线网络信息结构体 WiFi_Info

```
class WiFi_Info(ctypes.Structure):

    _fields_ = [("channel", ctypes.c_int),

                ("ip", ctypes.c_char * 16),

                ("mac", ctypes.c_char * 18),

                ("mask", ctypes.c_char * 16),

                ("mode", ctypes.c_char * 5),

                ("password", ctypes.c_char * 16),

                ("ssid", ctypes.c_char * 32)]
```

结构体成员

mode

网络模式（ap 代表热点模式，sta 代表联网模式）

ip

网络地址

mask

子网掩码

mac

MAC 地址

channel

无线信道

ssid

无线名称

password

无线密码

4.4.5. 回调函数数据结构体

基础系列透传时可通过初始化时注册回调函数接收机械臂状态信息，以下为基础系列的回调函数数据结构体：

```
class CallbackData(ctypes.Structure):

    _fields_ = [

        ("sockhand", ctypes.c_int),   # 返回调用时句柄

        ("codeKey", ctypes.c_int),   # 调用透传接口类型

        ("errCode", ctypes.c_int),   # API 解析错误码

        ("pose", Pose),   # 当前位姿
```

```
            ("joint", ctypes.c_float * 7),    # 当前关节角度

            ("nforce", ctypes.c_int),    # 力控方向上所受的力

            ("sys_err", ctypes.c_uint16)    # 系统错误

        ]
CANFD_Callback = ctypes.CFUNCTYPE(None, CallbackData)
```

I 系列通过设置 UDP 接口主动推送并注册回调函数获取到机械臂信息，以下为 I 系列回调函数数据结构体：

```
class RobotStatus(ctypes.Structure):

    _fields_ = [

        ("errCode", ctypes.c_int),    # API 解析错误码

        ("arm_ip", ctypes.c_char_p),    # 推送消息的机械臂 IP

        ("arm_err", ctypes.c_uint16),    # 机械臂错误码

        ("joint_status", JointStatus),    # 当前关节状态
```

```
            ("force_sensor", ForceData),    # 力数据

            ("sys_err", ctypes.c_uint16),    # 系统错误码

            ("waypoint", Pose)    # 路点信息

        ]
RealtimePush_Callback = ctypes.CFUNCTYPE(None, RobotStatus)
```

5.接口库函数说明

5.1. 连接相关函数

该部分函数用来控制 socket 连接的打开与关闭。

5.1.1. 实例化机械臂

Arm(dev_mode, ip, pCallback)	
描述	实例化一个机械臂对象。与机械臂建立 Socket 连接以控制机械臂。
参数	（1）dev_mode 目标设备型号 RM65、RML63_II 、ECO65、RM75、GEN72。若传入型号非法，则默认机械臂为六轴。 （2）ip 机械臂 IP 地址 （3）pCallback

	（基础系列）用于接收透传接口回调函数， 不需要则不传入参数。
返回值	成功：返回一个机器人对象； 失败：销毁创建的对象。

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接 RM_65 机械臂
robot = Arm(RM65, "192.168.1.18")
```

5.1.2. 查询连接状态 Arm_Socket_State

Arm_Socket_State()	
描述	用于查询连接状态。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.1.3. API 反初始化 RM_API_UnInit

RM_API_UnInit()	
描述	API 反初始化

5.1.4. 查询 API 版本信息 API_Version

API_Version()	
描述	查询 API 版本信息
返回值	API 版本号

代码示例：

```
from robotic_arm_package.robotic_arm import *


robot= Arm(RM65, '192.168.1.18')
```

```
# API 版本信息
print(robot.API_Version())
```

5.1.5. 关闭连接 Arm_Socket_Close

Arm_Socket_Close()	
描述	该函数用于关闭与机械臂的 socket 连接。
返回值	（1）ArmSocket Socket 句柄。

5.2. 关节配置函数

睿尔曼机械臂在出厂前所有参数都已经配置到最佳状态，一般不建议用户修改关节的底层参数。若用户确需修改，首先应使机械臂处于非使能状态，然后再发送修改参数指令，参数设置成功后，发送关节恢复使能指令。需要注意的是，关节恢复使能时，用户需要保证关节处于静止状态，以免上使能过程中关节发生报错。关节正常上使能后，用户方可控制关节运动。

5.2.1. 设置关节最大速度 Set_Joint_Speed

Set_Joint_Speed(joint_num, speed, block)	
描述	该函数用于设置关节最大速度，单位：°/s。建议使用默认最大速度，如需更改，设置的关节最大加速度与最大速度的比值需要≥1.5，否则可能出现运动异常。
参数	（1）joint_num 关节序号，1~7 （2）speed

	关节转速，单位：°/s （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.2. 设置关节最大加速度 Set_Joint_Acc

Set_Joint_Acc(joint_num, acc, block)	
描述	该函数用于设置关节最大加速度，单位：°/s²。建议使用默认关节最大加速度，如需更改，设置的关节最大加速度与最大速度的比值需要≥1.5，否则可能出现运动异常。
参数	（1）joint_num 关节序号，1~7 （2）acc 关节转速，单位：°/s² （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.3. 设置关节最小限位 Set_Joint_Min_Pos

Set_Joint_Min_Pos(joint_num, joint, block)

描述	该函数用于设置关节所能到达的最小限位，单位：°。
参数	（1）joint_num 关节序号，1~7 （2）joint 关节最小位置，单位：° （3）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.4. 设置关节最大限位 Set_Joint_Max_Pos

Set_Joint_Max_Pos(joint_num, joint, block)	
描述	该函数用于设置关节所能达到的最大限位，单位：°
参数	（1）joint_num 关节序号，1~7 （2）joint 关节最大位置，单位：° （3）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
输出参数	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.5. 设置关节最大速度 Set_Joint_Drive_Speed

Set_Joint_Drive_Speed(joint_num, speed)	
描述	该函数用于设置关节最大速度，单位：°/s。建议使用默认最大速度，如需更改，设置的关节最大加速度与最大速度的比值需要≥1.5，否则可能出现运动异常。
参数	（1）joint_num 关节序号，1~7 （2）speed 关节转速，单位：°/s
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.6. 设置关节最大加速度 Set_Joint_Drive_Acc

Set_Joint_Drive_Acc(joint_num, acc)	
描述	该函数用于设置关节最大加速度，单位：°/s²。建议使用默认关节最大加速度，如需更改，设置的关节最大加速度与最大速度的比值需要≥1.5，否则可能出现运动异常。
参数	（1）joint_num 关节序号，1~7 （2）acc 关节转速，单位：°/s²
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.7. 设置关节最小限位 Set_Joint_Drive_Min_Pos

Set_Joint_Drive_Min_Pos(joint_num, joint)	
描述	该函数用于设置关节所能到达的最小限位，单位：°。
参数	（1）joint_num 关节序号，1~7 （2）joint 关节最小位置，单位：°
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.8. 设置关节最大限位 Set_Joint_Drive_Max_Pos

Set_Joint_Drive_Max_Pos(joint_num, joint)	
描述	该函数用于设置关节所能达到的最大限位，单位：°
参数	（1）joint_num 关节序号，1~7 （2）joint 关节最大位置，单位：°
输出参数	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.9. 设置关节使能 Set_Joint_EN_State

Set_Joint_EN_State(joint_num, state, block)	
描述	该函数用于设置关节使能状态。
参数	（1）joint_num

	关节序号，1~7 （2）state true-上使能，false-掉使能 （3）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.10. 设置关节零位 Set_Joint_Zero_Pos

Set_Joint_Zero_Pos(joint_num, block)	
描述	该函数用于将当前位置设置为关节零位。
参数	（1）joint_num 关节序号，1~7 （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.11. 清除关节错误代码 Set_Joint_Err_Clear

Set_Joint_Err_Clear(joint_num, block)	
描述	该函数用于清除关节错误代码。
参数	（1）joint_num

	关节序号，1~7 （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.2.12. 自动设置限位 Auto_Set_Joint_Limit

Auto_Set_Joint_Limit(limit_mode)	
描述	该函数用于自动设置限位。
参数	（1）limit_mode 设置类型，1-正式模式，各关节限位为规格参数中的软限位和硬限位
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *



robot= Arm(RM65, '192.168.1.18')

lim = [178, 130, 135, 178, 130, 360]
# 设置关节 1-6 最小限位

test = [150, 120, 100, 150, 120, 180]

for i in range(1, 6):

    robot.Set_Joint_EN_State(i, False)
```

```
        time.sleep(1)
        robot.Set_Joint_Min_Pos(i, -test[i])
        time.sleep(1)
        robot.Set_Joint_EN_State(i, True)
        time.sleep(1)


print(f'关节最小限位：{robot.Get_Joint_Min_Pos()}')


# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.3. 关节参数查询函数

5.3.1. 查询关节最大速度 Get_Joint_Speed

Get_Joint_Speed()	
描述	该函数用于查询关节最大速度。
参数	
返回值	成功返回：(0, speed) speed：关节 1~7 转速数组，单位：°/s； 失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *
```

```
robot= Arm(RM65, '192.168.1.18')

# 关节最大速度

print(robot.Get_Joint_Speed())

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.3.2. 查询关节最大加速度 Get_Joint_Acc

Get_Joint_Acc()	
描述	该函数用于查询关节最大加速度。
参数	
返回值	成功返回：(0, acc) acc：存放关节 1~7 加速度数组，单位：°/s²； 失败返回：错误码，查询 API 错误类型。

5.3.3. 获取关节最小限位 Get_Joint_Min_Pos

Get_Joint_Min_Pos()	
描述	该函数用于获取关节最小限位。
参数	
返回值	成功返回：(0, min_joint) min_joint：存放关节 1~7 最小限位数组，单位：°； 失败返回：错误码，查询 API 错误类型。

5.3.4. 获取关节最大限位 Get_Joint_Max_Pos

Get_Joint_Max_Pos()

描述	该函数用于获取关节最大限位。
参数	
返回值	成功返回：(0, max_joint) max_joint：存放关节 1~7 最大限位数组，单位：°； 失败返回：错误码，查询 API 错误类型。

5.3.5. 查询关节最大速度(驱动器)Get_Joint_Drive_Speed

Get_Joint_Drive_Speed()	
描述	该函数用于查询关节最大速度。
参数	
返回值	成功返回：(0, speed) speed：关节 1~7 转速数组，单位：°/s； 失败返回：错误码，查询 API 错误类型。

5.3.6. 查询关节最大加速度(驱动器)Get_Joint_Drive_Acc

Get_Joint_Drive_Acc()	
描述	该函数用于查询关节最大加速度。
参数	
返回值	成功返回：(0, acc) acc：存放关节 1~7 加速度数组，单位：°/s²； 失败返回：错误码，查询 API 错误类型。

5.3.7. 获取关节最小限位(驱动器)Get_Joint_Drive_Min_Pos

Get_Joint_Drive_Min_Pos()	
描述	该函数用于获取关节最小限位。

参数	
返回值	成功返回：(0, min_joint) min_joint：存放关节 1~7 最小限位数组，单位：°； 失败返回：错误码，查询 API 错误类型。

5.3.8. 获取关节最大限位(驱动器)Get_Joint_Drive_Max_Pos

Get_Joint_Drive_Max_Pos()	
描述	该函数用于获取关节最大限位。
参数	
返回值	成功返回：(0, max_joint) max_joint：存放关节 1~7 最大限位数组，单位：°； 失败返回：错误码，查询 API 错误类型。

5.3.9. 获取关节使能状态 Get_Joint_EN_State

Get_Joint_EN_State()	
描述	该函数用于获取关节使能状态。
参数	
返回值	成功返回：(0, state) state：关节 1~7 使能状态数组，1-使能状态，0-掉使能状态； 失败返回：错误码，查询 API 错误类型。

5.3.10. 获取关节错误代码 Get_Joint_Err_Flag

Get_Joint_Err_Flag()	
描述	该函数用于获取关节错误代码。
参数	

返回值	成功返回：（0, state，bstate） state：存放关节错误码（请参考 api 文档中的关节错误码）； bstate：关节抱闸状态(1 代表抱闸未打开，0 代表抱闸已打开)； 失败返回：错误码，查询 API 错误类型。

5.3.11. 查询关节软件版本号 Get_Joint_Software_Version

Get_Joint_Software_Version()	
描述	该函数用于查询关节软件版本号。
参数	
返回值	成功返回：(0, version) version：存放关节 1~7 关节软件版本号 失败返回：错误码，查询 API 错误类型。

5.4. 机械臂末端运动参数配置

5.4.1. 设置末端最大线速度 Set_Arm_Line_Speed

Set_Arm_Line_Speed(speed,block)	
描述	该函数用于设置机械臂末端最大线速度。建议使用默认最大线速度，如需更改，设置的机械臂末端最大线加速度与最大线速度的比值需要≥3，否则可能出现运动异常。

参数	（1）speed 末端最大线速度，单位 m/s （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.4.2. 设置末端最大线加速度 Set_Arm_Line_Acc

Set_Arm_Line_Acc(acc,block)	
描述	该函数用于设置机械臂末端最大线加速度。建议使用默认最大线加速度，如需更改，设置的机械臂末端最大线加速度与最大线速度的比值需要≥3，否则可能出现运动异常。
参数	（1）acc 末端最大线加速度，单位 m/s^2 （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.4.3. 设置末端最大角速度 Set_Arm_Angular_Speed

Set_Arm_Angular_Speed(speed,block)	
描述	该函数用于设置机械臂末端最大角速度。建议使用默认最大角速度，如需

	更改，设置的机械臂末端最大角加速度与最大角速度的比值需要≥3，否则可能出现运动异常。
参数	（1）speed 机械臂末端最大角速度，单位 rad/s （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.4.4. 设置末端最大角加速度 Set_Arm_Angular_Acc

Set_Arm_Angular_Acc(acc,block)	
描述	该函数用于设置机械臂末端最大角加速度。建议使用默认最大角加速度，如需更改，设置的机械臂末端最大角加速度与最大角速度的比值需要≥3，否则可能出现运动异常。
参数	（1）acc 末端最大角加速度，单位 rad/s^2 （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.4.5. 获取末端最大线速度 Get_Arm_Line_Speed

Get_Arm_Line_Speed()	
描述	该函数用于获取机械臂末端最大线速度。
参数	
返回值	成功返回：(0, speed) speed：末端最大线速度，单位 m/s； 失败返回：错误码，查询 API 错误类型。

5.4.6. 获取最大末端线加速度 Get_Arm_Line_Acc

Get_Arm_Line_Acc()	
描述	该函数用于获取机械臂末端最大线加速度。
参数	
返回值	成功返回：(0, acc) acc：末端最大线加速度，单位 m/s^2； 失败返回：错误码，查询 API 错误类型。

5.4.7. 获取末端最大角速度 Get_Arm_Angular_Speed

Get_Arm_Angular_Speed()	
描述	该函数用于获取机械臂末端最大角速度。
参数	
返回值	成功返回：(0, speed) speed：末端最大角速度，单位 rad/s； 失败返回：错误码，查询 API 错误类型。

5.4.8. 获取末端最大角加速度 Get_Arm_Angular_Acc

Get_Arm_Angular_Acc()	
描述	该函数用于获取机械臂末端最大角加速度。

参数	
返回值	成功返回：(0, acc) acc：末端最大角加速度，单位 rad/s^2； 失败返回：错误码，查询 API 错误类型。

5.4.9. 设置机械臂末端参数为初始值 Set_Arm_Tip_Init

Set_Arm_Tip_Init(block)	
描述	该函数用于设置机械臂末端参数为初始值。
参数	（1）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.4.10. 设置碰撞等级 Set_Collision_Stage

Set_Collision_Stage(stage,block)	
描述	该函数用于设置机械臂动力学碰撞等级，等级 0~8，等级越高，碰撞检测越灵敏，同时也更容易发生误碰撞检测。机械臂上电后默认碰撞等级为 0，即不检测碰撞。
参数	（1）stage 等级：0~8，0-无碰撞，8-碰撞最灵敏 （2）block False-非阻塞，发送后立即返回；True-阻塞，等待控制器返回设置成功指令
返回值	成功返回：0；

	失败返回：错误码，查询 API 错误类型。

5.4.11. 查询碰撞等级 Get_Collision_Stage

Get_Collision_Stage(stage)	
描述	该函数用于查询机械臂动力学碰撞等级，等级 0~8，数值越高，碰撞检测越灵敏，同时也更易发生误碰撞检测。机械臂上电后默认碰撞等级为 0，即不检测碰撞。
参数	（1）stage 碰撞等级，等级：0~8
返回值	成功返回：(0, stage) stage：碰撞等级，等级：0~8； 失败返回：错误码，查询 API 错误类型。

5.4.12. 设置关节零位补偿角度 Set_Joint_Zero_Offset

Set_Joint_Zero_Offset(offset,block)	
描述	该函数用于设置机械臂各关节零位补偿角度，一般在对机械臂零位进行标定后调用该函数。
参数	（1）offset 关节 1~7 零位补偿角度数组，角度单位：度 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。
备注	该指令用户不可自行使用，必须配合测量设备进行绝对精度补偿时方可使

	用，否则会导致机械臂参数错误！

5.5. 机械臂末端接口板

5.5.1. 查询末端接口板软件版本号 Get_Tool_Software_Version

Get_Tool_Software_Version(version)	
描述	该函数用于查询末端接口板软件版本号
参数	
返回值	成功返回：(0, version) version：末端接口板软件版本号； 失败返回：错误码，查询 API 错误类型。

5.6. 工具坐标系设置

5.6.1. 标定点位 Auto_Set_Tool_Frame

Auto_Set_Tool_Frame(point_num, block)	
描述	该函数用于六点法自动设置工具坐标系（标记点位），机械臂控制器最多只能存储 10 个工具坐标系信息，在建立新的工具坐标系之前，请确认工具坐标系数量没有超过限制，否则建立新工具坐标系无法成功。
参数	（1）point_num 1~6 代表 6 个标定点。 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；

	失败返回：错误码，查询 API 错误类型。
备注	机械臂控制器最多只能存储 10 个工具信息，在建立新的工具之前，请确认工具数量没有超过限制，否则建立新工具无法成功。

5.6.2. 生成工具坐标系 Generate_Auto_Tool_Frame

Generate_Auto_Tool_Frame(name,payload,x,y,z, block)	
描述	该函数用于六点法自动设置工具坐标系(生成坐标系)，机械臂控制器最多只能存储 10 个工具坐标系信息，在建立新的工具坐标系之前，请确认工具坐标系数量没有超过限制，否则建立新工具坐标系无法成功。
参数	（1）name 工具坐标系名称，不能超过十个字节。 （2）payload 新工具坐标系执行末端负载重量　单位 kg （3）x,y,z 新工具坐标系执行末端负载位置　位置 x，y，z　单位 mm （4）block False-非阻塞，发送后立即返回；　True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。
备注	控制器只能存储十个工具坐标系，超过十个控制器不予响应，请在标定前查询已有工具坐标系。

5.6.3. 手动设置工具坐标系 Manual_Set_Tool_Frame

Manual_Set_Tool_Frame(name, pose,payload,x,y,z, block)	
描述	该函数用于手动设置工具坐标系。
参数	（1）name 工具坐标系名称，不能超过十个字节。 （2）pose 新工具坐标系执行末端相对于机械臂法兰中心的位姿 （3）payload 新工具坐标系执行末端负载重量　单位 kg （4）x,y,z 新工具坐标系执行末端负载位置 位置 x，y，z 单位 mm （5）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	控制器只能存储十个工具坐标系，超过十个控制器不予响应，请在标定前查询已有工具。

5.6.4. 切换当前工具坐标系 Change_Tool_Frame

Change_Tool_Frame(name, block)	
描述	该函数用于切换当前工具坐标系
参数	（1）name 目标工具坐标系名称

	（2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.6.5. 删除指定工具坐标系 Delete_Tool_Frame

Delete_Tool_Frame(name, block)	
描述	该函数用于删除指定工具坐标系
参数	（1）name 要删除的工具坐标系名称 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	删除坐标系后，机械臂将切换到机械臂法兰末端工具坐标系。

5.6.6. 修改工具坐标系 Update_Tool_Frame

Update_Tool_Frame(name, pose,payload,x,y,z, block)	
描述	该函数用于修改指定工具坐标系。
参数	（1）name 工具坐标系名称，不能超过十个字节。 （2）pose 新工具坐标系执行末端相对于机械臂法兰中心的位姿 （3）payload

	新工具坐标系执行末端负载重量　单位 kg （4）x,y,z 新工具坐标系执行末端负载位置 位置 x，y，z 单位 mm
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.6.7. 设置工具坐标系的包络参数 Set_Tool_Envelope

Set_Tool_Envelope(envelop_list: ToolEnvelopeList)	
描述	该函数用于设置指定工具坐标系的包络参数
参数	（1）envelop_list 包络参数列表，每个工具最多支持 5 个包络球，可以没有包络
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

示例	# 给“test_2”工具坐标系设置三个包络球
	envelope1 = ToolEnvelope("L", 0.01, 0.1, 0.01, 0.010)
	envelope2 = ToolEnvelope("R", 0.01, 0.1, 0.010, 0.010)
	envelope3 = ToolEnvelope("C", 0.01, 0.1, 0.010, 0.010)
	envelopelist = ToolEnvelopeList("test_2", [envelope1, envelope3, envelope2], 3)
	print(robot.Set_Tool_Envelope(envelopelist))

5.6.8. 获取指定工具坐标系的包络参数 Get_Tool_Envelope

Get_Tool_Envelope(tool_name) -> (int, dict)	
描述	该函数用于获取指定工具坐标系包络参数列表
参数	（1）tool_name 指定工具坐标系名称

返回值	成功返回：(0, envelope_list) envelope_list：包络参数列表，每个工具最多支持 5 个包络球，可以没有包络； 失败返回：错误码，查询 API 错误类型。

示例	# 查询并打印“test_2”工具坐标系的包络参数字典
	print(robot.Get_Tool_Envelope("test_2")[1])
	输出结果：{'tool_name': 'test_2', 'List': [{'name': 'L', 'radius': 0.01, 'x': 0.1, 'y': 0.01, 'z': 0.01}, {'name': 'C', 'radius': 0.01, 'x': 0.1, 'y': 0.01, 'z': 0.01}, {'name': 'R', 'radius': 0.01, 'x': 0.1, 'y': 0.01, 'z': 0.01}], 'count': 3}

5.7. 工具坐标系查询

5.7.1. 获取当前工具坐标系 Get_Current_Tool_Frame

Get_Current_Tool_Frame()	
描述	该函数用于获取当前工具坐标系
参数	
返回值	成功返回：(0, tool) tool 返回的工具坐标系结构体； 失败返回：错误码，查询 API 错误类型。

5.7.2. 获取指定工具坐标系 Get_Given_Tool_Frame

Get_Given_Tool_Frame(name)	
描述	该函数用于获取指定工具坐标系
参数	（1）name 指定的工具坐标系名称

返回值	成功返回：(0, tool) tool 返回的工具坐标系参数； 失败返回：错误码，查询 API 错误类型。

5.7.3. 获取所有工具坐标系名称 Get_All_Tool_Frame

Get_All_Tool_Frame()	
描述	该函数用于获取所有工具坐标系名称
参数	
返回值	成功返回：(0, names, len) names：返回的工具坐标系名称数组 len：返回工具坐标系数量。 失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *



robot= Arm(RM65, '192.168.1.18')


res= robot.Manual_Set_Tool_Frame('new1', [0.371339, -0.29879, 0.286709, 3.075, -0.358,
-0.652], 1, 0, 0, 0)

# 获取全部工具坐标系名称
print(f'工具坐标系：{self.aa.Get_All_Tool_Frame()}')


# 切换当前工具坐标系
robot.Change_Tool_Frame('Arm_Tip')
# 获取当前工具坐标系

retval, tool = robot.Get_Current_Tool_Frame()

print('当前工具坐标系：', tool.frame_name.name)
```

```
print('当前工具坐标系位置：', tool.pose.position.x, tool.pose.position.y, tool.pose.position.z)
# 获取指定坐标系
Retval, tool = robot.Get_Given_Tool_Frame('new1')

print('指定工具坐标系位置：', tool)

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.8. 工作坐标系设置

5.8.1. 自动设置工作坐标系 Auto_Set_Work_Frame

Auto_Set_Work_Frame(name, point_num, block)	
描述	该函数用于三点法自动设置工作坐标系。
参数	（1）name 工作坐标系名称，不能超过十个字节。 （2）point_num 1~3 代表 3 个标定点，依次为原点、X 轴上任意点、Y 轴上任意点，4 代表生成成坐标系。 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	机械臂控制器最多只能存储 10 个工作坐标系信息，在建立新的工作坐标系之前，请确认工作坐标系数量没有超过限制，否则建立新工作坐标系无法

	成功。

5.8.2. 手动设置工作坐标系 Manual_Set_Work_Frame

Manual_Set_Work_Frame(name, pose, block)	
描述	该函数用于手动设置工作坐标系。
参数	（1）name 工作坐标系名称，不能超过十个字节。 （2）pose 新工作坐标系相对于基坐标系的位姿。 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	控制器只能存储十个工作坐标系，超过十个控制器返回设置失败，请在标定前查询已有工作坐标系。

5.8.3. 切换当前工作坐标系 Change_Work_Frame

Change_Work_Frame(name, block)	
描述	该函数用于切换当前工作坐标系
参数	（1）name 目标工作坐标系名称 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。

返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.8.4. 删除指定工作坐标系 Delete_Work_Frame

Delete_Work_Frame(name, block)	
描述	该函数用于删除指定工作坐标系。
参数	（1）name 要删除的工作坐标系名称 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	删除坐标系后，机械臂将切换到机械臂基坐标系

5.8.5. 修改指定工作坐标系 Update_Work_Frame

Update_Work_Frame(name, pose)	
描述	该函数用于修改指定工作坐标系。
参数	（1）name 要删除的工作坐标系名称 （2）pose 更新工作坐标系相对于基坐标系的位姿
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.9. 工作坐标系查询

5.9.1. 获取当前工作坐标系 Get_Current_Work_Frame

Get_Current_Work_Frame()	
描述	该函数用于获取当前工作坐标系。
参数	
返回值	成功返回：(0, frame) frame：返回的工作坐标系位姿参数； 失败返回：错误码，查询 API 错误类型。

5.9.2. 获取指定工作坐标系 Get_Given_Work_Frame

Get_Given_Work_Frame(name)	
描述	该函数用于获取指定工作坐标系
参数	（1）name 指定的工作坐标系名称
返回值	成功返回：(0, pose) pose：返回的工作坐标系位姿参数； 失败返回：错误码，查询 API 错误类型。

5.9.3. 获取所有工作坐标系名称 Get_All_Work_Frame

Get_All_Work_Frame()	
描述	该函数用于获取所有工作坐标系名称。
参数	
返回值	成功返回：(0, names, len) names：返回的工作坐标系的名称数组 len：返回的工作坐标系的数量； 失败返回：错误码，查询 API 错误类型。

代码示例：

```python
from robotic_arm_package.robotic_arm import *



robot= Arm(RM65, '192.168.1.18')


res= robot.Manual_Set_Work_Frame('new2', [0.27826, -0.37872, 0.28538, 3.0, -0.31, -0.889])

# 获取全部工作坐标系名称
print(f'工作坐标系：{self.aa.Get_All_Work_Frame()}')


# 切换当前工作坐标系
robot.Change_Work_Frame('Base')
# 获取当前工作坐标系

retval, frame= robot.Get_Current_Work_Frame()
if retval == 0:

    print('当前工作坐标系：', frame.frame_name.name)

    print('当前工作坐标系位置：', frame.pose.position.x, frame.pose.position.y,
frame.pose.position.z)

else:

    print(f'获取当前坐标系失败:{retval}')

# 获取指定坐标系
retval, frame= robot.Get_Given_Work_Frame('new2')
if retval == 0:

    print('指定工作坐标系位姿：', frame)

else:

    print(f'获取指定坐标系失败:{retval}')
# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.10. 机械臂状态查询

5.10.1. 获取机械臂当前状态 Get_Current_Arm_State

Get_Current_Arm_State()	
描述	该函数用于获取机械臂当前状态
参数	
返回值	成功返回：(0, joint, pose, Arm_Err, Sys_Err) joint：关节 1~7 角度数组 pose：机械臂当前位姿[x,y,z,rx,ry,rz] Arm_Err：机械臂运行错误代码 Sys_Err：控制器错误代码。 失败返回：错误码，查询 API 错误类型。

5.10.2. 获取关节温度 Get_Joint_Temperature

Get_Joint_Temperature()	
描述	该函数用于获取关节当前温度。
参数	
返回值	成功返回：(0, temperature) temperature：关节 1~7 温度数组； 失败返回：错误码，查询 API 错误类型。

5.10.3. 获取关节电流 Get_Joint_Current

Get_Joint_Current()	
描述	该函数用于获取关节当前电流。
参数	

返回值	成功返回：(0, current)current：关节 1~7 电流数组； 失败返回：错误码，查询 API 错误类型。

5.10.4. 获取关节电压 Get_Joint_Voltage

Get_Joint_Voltage()	
描述	该函数用于获取关节当前电压。
参数	
返回值	成功返回：(0,voltage) voltage：关节 1~7 电压数组； 失败返回：错误码，查询 API 错误类型。

5.10.5. 获取关节当前角度 Get_Joint_Degree

Get_Joint_Degree ()	
描述	该函数用于获取机械臂各关节的当前角度
参数	
返回值	成功返回：(0, joint) joint：关节 1~7 当前角度数组； 失败返回：错误码，查询 API 错误类型。

5.10.6. 获取所有状态 Get_Arm_All_State

Get_Arm_All_State()	
描述	该函数用于获取机械臂所有状态
参数	
返回值	成功返回：(0, joint_state) joint_state：机械臂所有状态； 失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *
```

```
# 连接机械臂
robot = Arm(RM65, "192.168.1.18")

res, joint_state = self.aa.Get_Arm_All_State()

self.textEdit.append(f'机械臂关节温度、电流:{list(joint_state.temperature), list(joint_state.current)}')

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.10.7. 获取轨迹规划计数 Get_Arm_Plan_Num

Get_Arm_Plan_Num()	
描述	该函数用于获取机械臂轨迹规划计数
参数	
返回值	成功返回：(0, plan) plan：查询到的轨迹规划计数； 失败返回：错误码，查询 API 错误类型。

5.11. 机械臂初始位姿

5.11.1. 设置初始位姿角度 Set_Arm_Init_Pose

Set_Arm_Init_Pose(target, block)	
描述	该函数用于设置机械臂的初始位姿角度。
参数	（1）target 机械臂初始位置关节角度数组 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设

	置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.11.2. 获取初始位姿角度 Get_Arm_Init_Pose

Get_Arm_Init_Pose(joint)	
描述	该函数用于获取机械臂初始位姿角度。
参数	
返回值	成功返回：(0, joint) joint：机械臂初始位姿关节角度数组； 失败返回：错误码，查询 API 错误类型。

5.11.3. 设置安装角度 Set_Install_Pose

Set_Install_Pose(x, y, z, block)	
描述	该函数用于设置机械臂安装方式。
参数	（1）x 旋转角，单位 ° （2）y 俯仰角，单位 ° （3）z 方位角，单位 ° （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；

	失败返回：错误码，查询 API 错误类型。

5.11.4. 查询安装角度 Get_Install_Pose

Get_Install_Pose(fx, fy, fz)	
描述	该函数用于查询机械臂安装角度。
参数	
返回值	成功返回：(0, fx, fy, fz) fx：旋转角(out)，单位 ° fy：俯仰角(out)，单位 ° fz：方位角(out)，单位 °； 失败返回：错误码，查询 API 错误类型。

5.12. 机械臂运动规划

5.12.1. 关节空间运动 Movej_Cmd

Movej_Cmd(joint, v, r, trajectory_connect, block)	
描述	该函数用于关节空间运动。
参数	（1）joint 目标关节 1~7 角度数组 （2）v 速度百分比系数，1~100。 （3）r 交融半径百分比系数，0~100。 （4）trajectory_connect

	代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行。 （5）block False-非阻塞，发送后立即返回； True-阻塞，等待机械臂到达位置或者规划失败。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。
注意	trajectory_connect 参数为 1 交融半径才生效，如果为 0 则交融半径不生效

5.12.2. 笛卡尔空间直线运动 Movel_Cmd

Movel_Cmd(pose, v, trajectory_connect, r, block)	
描述	该函数用于笛卡尔空间直线运动。
参数	（1）pose 目标位姿，位置单位：米，姿态单位：弧度 （2）v 速度百分比系数，1~100 （3）r 交融半径百分比系数，0~100。 （4）trajectory_connect 代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行。 （5）block

	False-非阻塞，发送后立即返回； True-阻塞，等待机械臂到达位置或者规划失败。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。
注意	trajectory_connect 参数为 1 交融半径才生效，如果为 0 则交融半径不生效

5.12.3. 笛卡尔空间圆弧运动 Movec_Cmd

Movec_Cmd(pose_via, pose_to, v, loop, trajectory_connect, r, block)	
描述	该函数用于笛卡尔空间圆弧运动
参数	（1）pose_vai 中间点位姿，位置单位：米，姿态单位：弧度 （2）pose_to 终点位姿，位置单位：米，姿态单位：弧度 （3）v 速度比例 1~100，即规划速度和加速度占机械臂末端最大角速度和角加速度的百分比 （4）r 交融半径百分比系数，0~100。 （5）loop 规划圈数，目前默认 0。 （6）trajectory_connect 代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一

	条轨迹一起规划，当为 1 时，轨迹不会立即执行 （7）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
注意	trajectory_connect 参数为 1 交融半径才生效，如果为 0 则交融半径不生效

5.12.4. 关节角度 CANFD 透传 Movej_CANFD

Movej_CANFD(joint, follow, expand)	
描述	该函数用于角度不经规划，直接通过 CANFD 透传给机械臂，使用透传接口时，请勿使用其他运动接口。
参数	（1）joint 关节 1~7 目标角度数组，单位° （2）follow 是否高跟随，true--高跟随，false--低跟随 （3）expand 扩展关节目标位置，单位°，不需要时传入 0 即可
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	透传周期越快，控制效果越好，越平顺。基础系列 WIFI 和网口模式透传周期最快 20ms，USB 和 RS48 模式透传周期最快 10ms。高速网口的透传周期最快也可到 10ms，不过在使用该高速网口前，需要使用指令打开配置。另外 I 系列有线网口周期最快可达 5ms。

	用户使用该函数时请做好轨迹规划，轨迹规划的平滑成都决定了机械臂的运行状态，帧与帧之间关节的角度差不能超过 10°，并保证关节规划的速度不超过 180°/s，否则关节不会响应。 由于该模式直接下发给机械臂，不经控制器规划，因此只要控制器运行正常并且目标角度在可达范围内，机械臂立即返回成功指令，此时机械臂可能仍在运行；若有错误，立即返回失败指令。

5.12.5. 位姿 CANFD 透传 Movep_CANFD

Movep_CANFD(pose, follow)	
描述	该函数用于角度不经规划，直接通过 CANFD 透传给机械臂，使用透传接口是，请勿使用其他运动接口。
参数	（1）pose 位姿（优先采用四元数表达） （2）follow 是否高跟随，true--高跟随，false--低跟随
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接机械臂
robot = Arm(RM65, "192.168.1.18")

joint = [20, 20, 70, 30, 90, 10]

zero = [0, 0, 0, 0, 0, 0]
```

```
ret = robot.Movej_Cmd(joint, v=20, r=0)

if ret:

    print(f'运动到起点失败:{ret}')

else:

    print(f'运动到位:{ret}')

res, joint, pose, arm_err, sys_err = robot.Get_Current_Arm_State()


# 周期 20ms，低跟随位姿透传

for i in range(50):

    pose[0] += 0.001

    robot.Movep_CANFD(pose, False)

    time.sleep(0.02)



robot.Movej_Cmd(zero, 20, 0)

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.12.6. 计算环绕运动位姿 MoveRotate_Cmd

MoveRotate_Cmd(rotateAxis, rotateAngle,choose_axis, v, r, trajectory_connect, block)	
描述	该函数用于计算环绕运动位姿并按照结果运动。
参数	（1）rotateAxis 旋转轴： 1：x 轴， 2：y 轴， 3：z 轴

	（2）rotateAngle 旋转角度： 旋转角度， 单位(度) （3）choose_axis 指定计算时使用的坐标系 （4）v 速度 （5）r 交融半径 （6）trajectory_connect 代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行 （7）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.7. 沿工具端位姿移动 MoveCartesianTool_Cmd

MoveCartesianTool_Cmd(Joint_Cur, movelengthx,movelengthy, movelengthz, m_dev, v, r, trajectory_connect, block)	
描述	该函数用于沿工具端位姿运动。
参数	（1）Joint_Cur 当前关节角度 （2）movelengthx

	沿 X 轴移动长度，米为单位 （3）Movelengthy 沿 Y 轴移动长度，米为单位 （4）movelengthz 沿 Z 轴移动长度，米为单位 （5）m_dev 机械臂型号 （6）v 速度 （7）r 交融半径 （8）trajectory_connect 代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行 （9）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.8. 快速急停 Move_Stop_Cmd

Move_Stop_Cmd(block)	
描述	该函数用于突发状况，机械臂以最快速度急停，轨迹不可恢复。

参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.9. 暂停当前规划 Move_Pause_Cmd

Move_Pause_Cmd(block)	
描述	该函数用于轨迹暂停，暂停在规划轨迹上，轨迹可恢复。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.10. 继续当前轨迹 Move_Continue_Cmd

Move_Continue_Cmd(block)	
描述	该函数用于轨迹暂停后，继续当前轨迹运动
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.11. 清除当前轨迹 Clear_Current_Trajectory

Clear_Current_Trajectory(block)	
描述	该函数用于清除当前轨迹，必须在暂停后使用，否则机械臂会发生意外！！!!
参数	（1）block

	False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.12. 清除所有轨迹 Clear_All_Trajectory

Clear_All_Trajectory(block)	
描述	该函数用于清除所有轨迹，必须在暂停后使用，否则机械臂会发生意外！
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.12.13. 关节空间运动 Movej_P_Cmd

Movej_P_Cmd(pose, v, r, trajectory_connect, block)	
描述	该函数用于关节空间运动到目标位姿
参数	（1）pose 目标位姿，位置单位：米，姿态单位：弧度。 **注意**：该目标位姿必须是机械臂当前工具坐标系相对于当前工作坐标系的位，用户在使用该指令前务必确保，否则目标位姿会出错！！！ （2）v 速度百分比系数，1~100 （3）r 交融半径百分比系数，0~100。 （4）trajectory_connect

	代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行 （5）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
注意	该运动暂不支持轨迹交融

5.12.14. 样条曲线运动 Moves_Cmd

Moves_Cmd(pose, v, r, trajectory_connect, block)	
描述	该函数用于样条曲线运动到目标位姿
参数	（1）pose 目标位姿，位置单位：米，姿态单位：弧度。 （2）v 速度百分比系数，1~100 （3）r 交融半径百分比系数，0~100。 （4）trajectory_connect 代表是否和下一条运动一起规划，0 代表立即规划，1 代表和下一条轨迹一起规划，当为 1 时，轨迹不会立即执行，样条曲线运动需至少连续下发三个点位，否则运动轨迹为直线 （5）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设

	置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
注意	该运动暂不支持轨迹交融

5.13. 机械臂示教

5.13.1. 关节示教 Joint_Teach_Cmd

Joint_Teach_Cmd(num, direction, v, block)	
描述	该函数用于关节示教，关节从当前位置开始按照指定方向转动，接收到停止指令或者到达关节限位后停止。
参数	（1）num 示教关节的序号，1~7 （2）direction 示教方向，0-负方向，1-正方向 （3）v 速度比例 1~100，即规划速度和加速度占关节最大线转速和加速度的百分比 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.13.2. 位置示教 Pos_Teach_Cmd

Pos_Teach_Cmd(type, direction, v, block)

描述	该函数用于当前坐标系下（默认为当前工作坐标系下，调用 5.13.5 切换示教运动坐标系 Set_Teach_Frame 可切换为工具坐标系），笛卡尔空间位置示教。机械臂在当前工作坐标系下，按照指定坐标轴方向开始直线运动，接收到停止指令或者该处无逆解时停止。
参数	（1）type 示教类型 0:x 轴方向 1：y 轴方向 2：z 轴方向 （2）direction 示教方向，0-负方向，1-正方向 （3）v 速度百分比系数，1~100 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.13.3. 姿态示教 Ort_Teach_Cmd

Ort_Teach_Cmd(type, direction, v, block)	
描述	该函数用于当前坐标系下（默认为当前工作坐标系下，调用 5.13.5 切换示教运动坐标系 Set_Teach_Frame 可切换为工具坐标系），笛卡尔空间末端姿态示教。机械臂在当前工作坐标系下，绕指定坐标轴旋转，接收到停止指令或者该处无逆解时停止。
参数	（1）type 示教类型 0:rx 轴方向 1：ry 轴方向 2：rz 轴方向

	（2）direction 示教方向，0-负方向，1-正方向 （3）v 速度比例 1~100，即规划速度和加速度占机械臂末端最大角速度和角加速度的百分比 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.13.4. 示教停止 Teach_Stop_Cmd

Teach_Stop_Cmd(block)	
描述	该函数用于示教停止。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.13.5. 切换示教运动坐标系 Set_Teach_Frame

Set_Teach_Frame(type, block)	
描述	该函数用于切换示教运动坐标系。
参数	（1）type 0： 基座标运动, 1： 工具坐标系运动 （2）block

	False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.13.6. 获取示教运动坐标系 Get_Teach_Frame

Get_Teach_Frame()	
描述	该函数用于切换示教运动坐标系。
参数	
返回值	成功返回：(0, type)； type：0-基座标运动, 1-工具坐标系运动 失败返回：错误码，查询 API 错误类型。

5.14. 机械臂步进

5.14.1. 关节步进 Joint_Step_Cmd

Joint_Step_Cmd(num, step, v, block)	
描述	该函数用于关节步进。关节在当前位置下步进指定角度。
参数	（1）num 关节序号，1~7 （2）step 步进的角度 （3）v 速度比例 1~100，即规划速度和加速度占指定关节最大关节转速和关节加速度的百分比

	（4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.14.2. 位置步进 Pos_Step_Cmd

Pos_Step_Cmd(type, step, v, block)	
描述	该函数用于当前坐标系下（默认为当前工作坐标系下，调用 5.13.5 切换示教运动坐标系 Set_Teach_Frame 可切换为工具坐标系），位置步进。机械臂末端在当前工作坐标系下，朝指定坐标轴方向步进指定距离，到达位置返回成功指令，规划错误返回失败指令。
参数	（1）type 示教类型 0:x 轴方向 1：y 轴方向 2：z 轴方向 （2）step 步进的距离，单位 m，精确到 0.001mm （3）v 速度百分比系数，1~100 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.14.3. 姿态步进 Ort_Step_Cmd

Ort_Step_Cmd(type, step, v, block)

描述	该函数用于当前坐标系下（默认为当前工作坐标系下，调用 5.13.5 切换示教运动坐标系 Set_Teach_Frame 可切换为工具坐标系），姿态步进。机械臂末端在当前工作坐标系下，绕指定坐标轴方向步进指定弧度，到达位置返回成功指令，规划错误返回失败指令。
参数	（1）type 示教类型 0:rx 轴方向 1：ry 轴方向 2：rz 轴方向 （2）step 步进的弧度，单位 rad，精确到 0.001rad （3）v 速度比例 1~100，即规划速度和加速度占机械臂末端最大角速度和角加速度的百分比 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15. 控制器配置

5.15.1. 获取控制器状态 Get_Controller_State

Get_Controller_State()	
描述	该函数用于获取控制器状态。
参数	
返回值	成功返回：(0, voltage, current, temperature, sys_err)

	voltage：返回的电压 current：返回的电流 temperature：返回的温度 sys_err：控制器运行错误代码； 失败返回：错误码，查询 API 错误类型。

5.15.2. 设置 WiFi AP 模式设置 Set_WiFi_AP_Data

Set_WiFi_AP_Data(wifi_name, password)	
描述	该函数用于控制器 WiFi AP 模式设置，非阻塞模式，机械臂收到后更改参数，蜂鸣器响后代表更改成功，控制器重启，以 WIFI AP 模式通信。
参数	（1）wifi_name 控制器 wifi 名称 （2）password wifi 密码
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.3. 设置 WiFi STA 模式设置 Set_WIFI_STA_Data

Set_WIFI_STA_Data(router_name, password)	
描述	该函数用于控制器 WiFi STA 模式设置，非阻塞模式，机械臂收到后更改参数，蜂鸣器响后代表更改成功，控制器重启，以 WIFI STA 模式通信。
参数	（1）router_name 路由器名称 （2）password 路由器 Wifi 密码

返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.4. 设置 UART_USB 接口波特率 Set_USB_Data

Set_USB_Data(baudrate)	
描述	该函数用于控制器 UART_USB 接口波特率设置非阻塞模式，机械臂收到后更改参数，然后立即通过 UART-USB 接口与外界通信。 该指令下发后控制器会记录当前波特率，断电重启后仍会使用该波特率对外通信。
参数	（1）baudrate 波特率：9600，19200，38400，115200 和 460800，若用户设置其他数据，控制器会默认按照 460800 处理。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.5. 设置 RS485 配置 Set_RS485

Set_RS485(baudrate)	
描述	该函数用于控制器设置 RS485 配置。 该指令下发后，若 Modbus 模式为打开状态，则会自动关闭，同时控制器会记录当前波特率，断电重启后仍会使用该波特率对外通信。
参数	（1）baudrate 波特率：9600，19200，38400，115200 和 460800，若用户设置其他数据，控制器会默认按照 460800 处理。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.6. 设置机械臂电源 Set_Arm_Power

Set_Arm_Power (cmd, block)

描述	该函数用于设置机械臂电源。
参数	（1）cmd true-上电， false-断电 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.7. 获取机械臂电源 Get_Arm_Power_State

Get_Arm_Power_State (power)	
描述	该函数用于获取机械臂电源
参数	
返回值	成功返回：(0,power) power：获取到的机械臂电源状态：1-上电，0-断电； 失败返回：错误码，查询 API 错误类型。

5.15.8. 读取机械臂软件版本 Get_Arm_Software_Version

Get_Arm_Software_Version(plan_version, ctrl_version, kernal1, kernal2, product_version)	
描述	该函数用于读取机械臂软件版本
参数	
返回值	成功返回：(0,plan_version,ctrl_version, kernal1, kernal2, product_version) plan_version：读取到的用户接口内核版本号 ctrl_version：实时内核版本号 kernal1：实时内核子核心 1 版本号

	kernal2：实时内核子核心 2 版本号 product_version：机械臂型号，仅 I 系列机械臂支持[-I]； 失败返回：错误码，查询 API 错误类型。

5.15.9. 获取控制器的累计运行时间 Get_System_Runtime

Get_System_Runtime (day, hour, min, sec)	
描述	读取控制器的累计运行时间。
参数	
返回值	成功返回：(0, day, hour, min, sec) day：天，hour：小时，min：分，sec：秒； 失败返回：错误码，查询 API 错误类型。

5.15.10. 清空控制器累计运行时间 Clear_System_Runtime

Clear_System_Runtime(block)	
描述	该函数用于清空控制器累计运行时间。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.11. 获取关节累计转动角度 Get_Joint_Odom

Get_Joint_Odom(odom)	
描述	该函数用于读取关节的累计转动角度。
参数	
返回值	成功返回：(0, odom)

	odom：各关节累计的转动角度值； 失败返回：错误码，查询 API 错误类型。

5.15.12. 清除关节累计转动角度 Clear_Joint_Odom

Clear_Joint_Odom(block)	
描述	该函数用于清空关节累计转动角度
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.13. 配置高速网口 Set_High_Speed_Eth

Set_High_Speed_Eth (num, block)	
描述	该函数用于配置高速网口。
参数	（1）num 0-关闭 1-开启 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.15.14. 设置高速网口网络配置 Set_High_Ethernet--基础系列

Set_High_Ethernet(ip, mask, gateway)	
描述	该函数用于设置高速网口网络配置[配置通讯内容]。

参数	（1）ip 网络地址 （2）mask 子网掩码 （3）gateway 网关
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.15.15. 获取高速网口网络配置 Get_High_Ethernet--基础系列

Get_High_Ethernet(ip, mask, gateway, mac)	
描述	该函数用于获取高速网口网络配置[配置通讯内容]
参数	
返回值	成功返回：(0, ip, mask, gateway, mac) ip：网络地址 mask：子网掩码 gateway：网关 mac：MAC 地址； 失败返回：错误码，查询 API 错误类型。

5.15.16. 保存参数 Save_Device_Info_All--基础系列

Save_Device_Info_All()	
描述	该函数用于保存所有参数
参数	

返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.15.17. 配置有线网卡 IP 地址 Set_NetIP--I 系列

Set_NetIP(ip)	
描述	该函数用于配置有线网卡 IP 地址[-I]
参数	（1）ip 网络地址
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.15.18. 查询有线网卡网络信息 Get_Wired_Net--I 系列

Get_Wired_Net(ip, mask, mac)	
描述	该函数用于查询有线网卡网络信息[-I]
参数	
返回值	成功返回：(0, ip, mask, mac) ip：网络地址 mask：子网掩码 mac：MAC 地址； 失败返回：错误码，查询 API 错误类型。

5.15.19. 查询无线网卡网络信息 Get_Wifi_Net--I 系列

Get_Wifi_Net()	
描述	该函数用于查询无线网卡网络信息[-I]
参数	
返回值	成功返回：(0, wifi_net)

	wifi_net：无线网络信息结构体 失败返回：错误码，查询 API 错误类型。

5.15.20. 恢复网络出厂设置 Set_Net_Default--I 系列

Set_Net_Default()	
描述	该函数用于恢复网络出厂设置[-I]
参数	
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.15.21. 清除系统错误代码 Clear_System_Err

Clear_System_Err(block)	
描述	该函数用于清除系统错误
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0； 失败返回：错误码，查询 API 错误类型。

5.15.22. 读取机械臂软件信息 Get_Arm_Software_Info

Get_Arm_Software_Info()	
描述	该函数用于读取机械臂软件信息
参数	
返回值	成功返回：(0,software_info) software_info：机械臂软件信息结构体；

	失败返回：错误码，查询 API 错误类型。

5.15.23. 设置机械臂模式(仿真/真实)Set_Arm_Run_Mode

Set_Arm_Run_Mode(mode)	
描述	该函数用于设置机械臂模式(仿真/真实)。
参数	（1）mode 模式 0:仿真 1:真实
返回值	成功返回：(0,software_info) software_info：机械臂软件信息结构体； 失败返回：错误码，查询 API 错误类型。

5.15.24. 获取机械臂模式(仿真/真实)Get_Arm_Run_Mode

Get_Arm_Run_Mode()	
描述	该函数用于获取机械臂模式(仿真/真实)。
参数	
返回值	成功返回：(0,mode) mode：模式 0:仿真 1:真实； 失败返回：错误码，查询 API 错误类型。

5.16. IO 配置

机械臂具有 IO 端口，基础系列数量和分类如下所示：

数字输出：DO	4 路，可配置为 0~12V
数字输入：DI	3 路，可配置为 0~12V
模拟输出：AO	4 路，输出电压 0~10V
模拟输入：AI	4 路，输入电压 0~10V

I 系列数量和分类如下所示：

数字 IO：DO/DI 复用	4 路，可配置为 0~24V

5.16.1. 设置数字 IO 模式 Set_IO_Mode--I 系列

Set_IO_Mode(io_num, io_mode)	
描述	该函数用于设置数字 IO 模式[-I]。
参数	（1）io_num IO 端口号，范围：1~4 （2）io_mode 模式，0-通用输入模式，1-通用输出模式、2-输入开始功能复用模式，3-输入暂停功能复用模式，4-输入继续功能复用模式，5-输入急停功能复用模式，6-输入进入电流环拖动复用模式，7-输入进入力只动位置拖动模式，8-输入进入力只动姿态拖动模式，9-输入进入力位姿结合拖动复用模式，10-输入外部轴最大软限位复用模式，11-输入外部轴最小软限位复用模式
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.16.2. 设置数字 IO 输出状态 Set_DO_State

Set_DO_State(io_num, state, block=True)	
描述	该函数用于配置指定 IO 输出状态。
参数	（1）io_num 指定 IO 通道号，范围 1~4 （2）state 参数 true-高， false-低

	（3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.16.3. 查询指定 IO 状态 Get_IO_State--I 系列

Get_IO_State(num)	
描述	该函数用于查询指定 IO 状态。
参数	（1）num 指定 IO 通道号，范围 1~4
返回值	成功返回：(0, state,mode) state：IO 状态 mode：0-通用输入模式，1-通用输出模式、2-输入开始功能复用模式，3-输入暂停功能复用模式，4-输入继续功能复用模式，5-输入急停功能复用模式，6-输入进入电流环拖动复用模式，7-输入进入力只动位置拖动模式，8-输入进入力只动姿态拖动模式，9-输入进入力位姿结合拖动复用模式，10-输入外部轴最大软限位复用模式，11-输入外部轴最小软限位复用模式[-I]； 失败返回：错误码，查询 API 错误类型。

5.16.4. 查询数字 IO 输出状态 Get_DO_State--基础系列

Get_DO_State(num)	
描述	该函数用于获取数字 IO 输出状态。
参数	（1）num

	指定 IO 通道号，范围 1~4
返回值	成功返回：(0, state) state：指定数字 IO 通道返回的状态，1-高，0-低； 失败返回：错误码，查询 API 错误类型。

5.16.5. 查询数字 IO 输入状态 Get_DI_State--基础系列

Get_DI_State(num)	
描述	该函数用于获取数字 IO 输入状态。
参数	（1）num 指定 IO 通道号，范围 1~3
返回值	成功返回：(0, state) state：指定数字 IO 通道返回的状态，1-高，0-低； 失败返回：错误码，查询 API 错误类型。

5.16.6. 设置模拟 IO 输出状态 Set_AO_State--基础系列

Set_AO_State(io_num, voltage, block=True)	
描述	该函数用于设置模拟 IO 输出状态。
参数	（1）num 指定通道号，1~4 （2）state IO 输出电压，分辨率 0.001V，范围：0~10000，代表输出电压 0v~10v （3）block False-非阻塞，发送后立即返回； True-阻塞，等待机械臂到

	达位置或者规划失败。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.16.7. 查询模拟 IO 输出状态 Get_AO_State--基础系列

Get_AO_State(num)	
描述	该函数用于获取模拟 IO 输出状态。
参数	（1）num 指定 IO 通道号，范围 1~4
返回值	成功返回：(0, state) state：IO 输出电压，分辨率 0.001V，范围：0~10000，代表输出电压 0v~10v； 失败返回：错误码，查询 API 错误类型。

5.16.8. 查询数字 IO 输入状态 Get_AI_State--基础系列

Get_AI_State(num)	
描述	该函数用于获取模拟 IO 输入状态。
参数	（1）num 指定 IO 通道号，范围 1~4
返回值	成功返回：(0, state) state：IO 输入电压，分辨率 0.001V，范围：0~10000，代表输出电压 0v~10v； 失败返回：错误码，查询 API 错误类型。

5.16.9. 查询所有 IO 输入状态 Get_IO_Input

Get_IO_Input()

描述	该函数用于查询所有 IO 输入状态。
参数	
返回值	成功返回：(0, DI_state, AI_voltage) DI_state：数字 IO 输入状态数组地址，1-高，0-低 AI_voltage：模拟 IO 输入通道 1~4 输入电压数组； 失败返回：错误码，查询 API 错误类型。

5.16.10. 查询所有 IO 的输出状态 Get_IO_Output

Get_IO_Output()	
描述	该函数用于查询所有 IO 输出状态。
参数	
返回值	成功返回：(0, DO_state, AO_voltage) DO_state：数字 IO 输出通道 1~4 状态数组地址，1-高，0-低 AO_voltage：模拟 IO 输出通道 1~4 输处电压数组； 失败返回：错误码，查询 API 错误类型。

5.16.11. 设置电源输出 Set_Voltage--I 系列

Set_Voltage(voltage_type, start_enable)	
描述	该函数用于设置控制器端电源输出。
参数	（1）voltage_type 电源输出类型，范围：0~3(0-0V，2-12V，3-24V) （2）start_enable true-开机启动时即输出此配置电压，false-取消开启启动配置电压
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.16.12. 获取电源输出 Get_Voltage--I 系列

Get_Voltage()	
描述	该函数用于获取控制器端电源输出。
参数	
返回值	成功返回：(0, voltage_type) voltage_type：电源输出类型，范围：0~3(0-0V，2-12V，3-24V)； 失败返回：错误码，查询 API 错误类型。

5.17. 末端工具 IO 配置

机械臂末端工具端具有 IO 端口，数量和分类如下所示：

电源输出	1 路，可配置为 0V/5V/12V/24V
数字 IO	2 路，输入输出可配置 输入：参考电平 12V~24V 输出：5~24V，与输出电压一致
通讯接口	1 路，可配置为 RS485

5.17.1. 设置工具端数字 IO 输出状态 Set_Tool_DO_State

Set_Tool_DO_State(num, state, block)	
描述	该函数用于配置工具端指定数字 IO 输出状态。
参数	（1）num 指定数字 IO 输出通道号，范围 1~2 （2）state 参数 true-高， false-低

	（3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.17.2. 设置工具端数字 IO 模式 Set_Tool_IO_Mode

Set_Tool_IO_Mode(num, state,block)	
描述	该函数用于设置数字输入输出 IO 模式。
参数	（1）num 指定数字输入通道号，范围 1~2 （2）state 模式，0-输入状态，1-输出状态 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.17.3. 查询工具端数字 IO 状态 Get_Tool_IO_State

Get_Tool_IO_State()	
描述	该函数用于查询工具端数字 IO 状态。
参数	
返回值	成功返回：(0, IO_Mode, IO_state) IO_Mode：数字 IO 通道模式(范围 1~2)， 0-输入模式，1-输出模式 IO_state：数字 IO 通道当前输入状态(范围 1~2)，1-高电平，0-低电平；

	失败返回：错误码，查询 API 错误类型。

5.17.4. 设置工具端电源输出 Set_Tool_Voltage

Set_Tool_Voltage(type, block)	
描述	该函数用于设置工具端电源输出。
参数	（1）type 电源输出类型，0-0V，1-5V，2-12V，3-24V （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.17.5. 获取工具端电源输出 Get_Tool_Voltage

Get_Tool_Voltage(voltage)	
描述	该函数用于获取工具端电源输出。
参数	（1）voltage 读取回来的电源输出类型：0-0V，1-5V，2-12V，3-24V
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.18. 末端手爪控制（选配）

睿尔曼机械臂末端配备了因时机器人公司的 EG2-4C2 手爪，为了便于用户操作手爪，机械臂控制器对用户开放了手爪的 API 函数（手爪控制 API 与末端 modbus 功能互斥。

5.18.1. 配置手爪的开口度 Set_Gripper_Route

Set_Gripper_Route(min_limit, max_limit, block)	
描述	该函数用于配置手爪的开口度。
参数	（1）min_limit 手爪开口最小值，范围：0~1000，无单位量纲 （2）Max_limit 手爪开口最大值，范围：0~1000，无单位量纲 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.18.2. 设置夹爪松开到最大位置 Set_Gripper_Release

Set_Gripper_Release (speed, block,timeout)	
描述	该函数用于控制手爪以指定速度张开到最大开口处
参数	（1）speed 手爪松开速度 ，范围 1~1000，无单位量纲 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。 （3）timeout 设置返回超时时间，阻塞模式生效，单位：秒
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.18.3. 设置夹爪夹取 Set_Gripper_Pick

Set_Gripper_Pick(speed, force, block,timeout)	
描述	该函数用于控制手爪以设定的速度去夹取，当手爪所受力矩大于设定的力矩阈值时，停止运动。
参数	（1）speed 手爪夹取速度 ，范围：1~1000，无单位量纲 （2）force 手爪夹取力矩阈值，范围 ：50~1000，无单位量纲 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。 （4）timeout 设置返回超时时间，阻塞模式生效，单位：秒
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.18.4. 设置夹爪持续夹取 Set_Gripper_Pick_On

Set_Gripper_Pick_On(speed, force, block,timeout)	
描述	该函数用于控制手爪以设定的速度去持续夹取，当手爪所受力矩大于设定的力矩阈值时，停止运动。之后当手爪所受力矩小于设定力矩后，手爪继续持续夹取，直到再次手爪所受力矩大于设定的力矩阈值时，停止运动。
参数	（1）speed 手爪夹取速度 ，范围：1~1000，无单位量纲 （2）force

	手爪夹取力矩阈值，范围 ：50~1000，无单位量纲 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。 （4）timeout 设置返回超时时间，阻塞模式生效，单位：秒
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.18.5. 设置夹爪到指定开口位置 Set_Gripper_Position

Set_Gripper_Position (position, block,timeout)	
描述	该函数用于控制手爪到达指定开口度位置。
参数	（1）position 手爪指定开口度，范围 ：1~1000，无单位量纲 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。 （3）timeout 设置返回超时时间，阻塞模式生效，单位：秒
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.18.6. 获取夹爪状态 Get_Gripper_State

Get_Gripper_State ()	
描述	该函数用于获取夹爪状态。
参数	

返回值	成功返回：(0,gripper_state) gripper_state：夹爪状态结构体； 失败返回：错误码，查询 API 错误类型。
备注	此接口需升级夹爪最新固件方可使用。

5.19. 拖动示教及轨迹复现

睿尔曼机械臂采用关节电流环实现拖动示教，拖动示教及轨迹复现的配置函数如下所示。

5.19.1. 进入拖动示教模式 Start_Drag_Teach

Start_Drag_Teach (block)	
描述	该函数用于控制机械臂进入拖动示教模式
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.2. 退出拖动示教模式 Stop_Drag_Teach

Stop_Drag_Teach (block)	
描述	该函数用于控制机械臂退出拖动示教模式
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.3. 拖动示教轨迹复现 Run_Drag_Trajectory

Run_Drag_Trajectory (block)	
描述	该函数用于控制机械臂复现拖动示教的轨迹，必须在拖动示教结束后才能使用，同时保证机械臂位于拖动示教的起点位置。若当前位置没有位于轨迹复现起点，请先调用运动到轨迹起点函数，否则会返回报错信息。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.4. 拖动示教轨迹复现暂停 Pause_Drag_Trajectory

Pause_Drag_Trajectory (block)	
描述	该函数用于控制机械臂在轨迹复现过程中的暂停。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.5. 拖动示教轨迹复现继续 Continue_Drag_Trajectory

Continue_Drag_Trajectory (block)	
描述	该函数用于控制机械臂在轨迹复现过程中暂停之后的继续，轨迹继续时，必须保证机械臂位于暂停时的位置，否则会报错，用户只能从开始位置重新复现轨迹。
参数	（1）block

	False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.6. 拖动示教轨迹复现停止 Stop_Drag_Trajectory

Stop_Drag_Trajectory (block)	
描述	该函数用于控制机械臂在轨迹复现过程中停止，停止后，不可继续。若要再次轨迹复现，只能从第一个轨迹点开始。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.7. 运动到轨迹起点 Drag_Trajectory_Origin

Drag_Trajectory_Origin (block)	
描述	轨迹复现前，必须控制机械臂运动到轨迹起点，如果设置正确，机械臂将以20%的速度运动到轨迹起点。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.8. 复合模式拖动示教 Start_Multi_Drag_Teach

Start_Multi_Drag_Teach(mode,singular_wall,block)	
描述	该函数用于复合模式拖动示。

参数	（1）mode 拖动示教模式 0-电流环模式，1-使用末端六维力，只动位置，2-使用末端六维力 ，只动姿态，3-使用末端六维力，位置和姿态同时动 （2）singular_wall 仅在六维力模式拖动示教中生效，用于指定是否开启拖动奇异墙，0 表示关闭拖动奇异墙，1 表示开启拖动奇异墙 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.19.9. 保存拖动示教轨迹 Save_Trajectory

Save_Trajectory(filename)	
描述	该函数用于保存拖动示教轨迹，在拖动示教结束后调用。
参数	（1）filename 轨迹要保存路径及名称，例: c:/rm_test.txt
返回值	成功返回：(0, num) num：轨迹点数； 失败返回：错误码，查询 API 错误类型。

5.19.10. 设置力位混合控制 Set_Force_Postion

Set_Force_Postion(sensor, mode, direction, N, block)	
描述	该函数用于设置力位混合控制。在笛卡尔空间轨迹规划时，使用该功能可保证机械臂末端接触力恒定，使用时力的方向与机械臂运动方向不能在同一方向。开启力位混合控制，执行笛卡尔空间运动，接收到运动完成反馈后，需

	要等待 2S 后继续下发下一条运动指令。
参数	（1）sensor 0-一维力；1-六维力 （2）mode 0-基坐标系力控；1-工具坐标系力控 （3）direction 力控方向；0-沿 X 轴；1-沿 Y 轴；2-沿 Z 轴；3-沿 RX 姿态方向；4-沿 RY 姿态方向；5-沿 RZ 姿态方向 （4）N 力的大小，单位 N，精确到 0.1N （5）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
注意	在进行力的操作之前，如果未进行力数据标定，可使用清空一维力、六维力数据接口对零位进行标定。

5.19.11. 结束力位混合控制 Stop_Force_Postion

Stop_Force_Postion (block)	
描述	该函数用于结束力位混合控制.
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。

返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接机械臂
robot = Arm(RM65, "192.168.1.18")

joint = [0, -20, -70, 0, -90, 0]

zero = [0, 0, 0, 0, 0, 0]



ret = robot.Movej_Cmd(joint, 20)

if ret:

    print(f'运动到起点失败:{ret}')

else:

    print(f'运动到位:{ret}')

res, joint1, pose, aerr1, serr1 = robot.Get_Current_Arm_State()

res2, joint2, poseto, aerr2, serr2 = robot.Get_Current_Arm_State()



poseto[0] += 0.05    # x 轴增加 0.05m

pose[1] += 0.05      # y 轴增加 0.05m


print(f'pose:{pose},poseto:{poseto}')

robot.Stop_Force_Postion()

robot.Set_Force_Postion(1, 1, 2, 5)    # 设置使用六维力，工具坐标系 Z 轴方向，5N 大小的力控
```

```
time.sleep(1)

for i in range(5):

    robot.Movel_Cmd(poseto, 5)

    time.sleep(2)

    robot.Movel_Cmd(pose, 5)

    time.sleep(2)

robot.Stop_Force_Postion()


# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.20. 末端六维力传感器的使用（选配）

睿尔曼 RM-65F 机械臂末端配备集成式六维力传感器，无需外部走线，用户可直接通过 API 对六维力进行操作，获取六维力数据。

5.20.1. 获取六维力数据 Get_Force_Data

Get_Force_Data()	
描述	该函数用于获取当前六维力传感器得到的力和力矩信息，若要周期获取力数据，周期不能小于 50ms。
参数	
返回值	成功返回：(0, force, zero_force, work_zero, tool_zero) force：返回的力和力矩数组，数组 6 个元素，依次为 Fx，Fy，Fz， Mx，My， Mz。其中，力的单位为 N；力矩单位为 Nm。

	zero_force：传感器坐标系下系统受到的外力数据 work_zero：当前工作坐标系下系统受到的外力数据 tool_zero：当前工具坐标系下系统受到的外力数据； 失败返回：错误码，查询 API 错误类型。

5.20.2. 清空六维力数据 Clear_Force_Data

Clear_Force_Data(block)	
描述	该函数用具清空六维力数据，即后续获得的所有数据都是基于当前数据的偏移量。。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.20.3. 设置六维力重心参数 Set_Force_Sensor

Set_Force_Sensor ()	
描述	设置六维力重心参数，六维力重新安装后，必须重新计算六维力所收到的初始力和重心。分别在不同姿态下，获取六维力的数据，用于计算重心位置。该指令下发后，机械臂以 20%的速度运动到各标定点，该过程不可中断，中断后必须重新标定。 **重要说明：**必须保证在机械臂静止状态下标定。
参数	
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.20.4. 手动标定六维力数据 Manual_Set_Force

Manual_Set_Force (type,joint)	
描述	该手动标定流程，适用于空间狭窄工作区域，以防自动标定过程中机械臂发生碰撞，用户手动标定六维力时，需要选择四个点位的数据，连续调用函数四次，机械臂开始自动沿用户设置的目标运动，并在此过程中计算六维力重心
参数	（1）type 点位，依次调用四次发送 1~4； （2）joint 关节角度
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.20.5. 退出标定流程 Stop_Set_Force_Sensor

Stop_Set_Force_Sensor (block)	
描述	在标定六/一维力过程中，如果发生意外，发送该指令，停止机械臂运动，退出标定流程。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接机械臂
```

```
robot = Arm(RM65, "192.168.1.18")

# 手动标定六维力重心
print(f'手动设置六维力重心参数：{robot.Manual_Set_Force(1, [0, 0, 0, 0, 90, 0])}')

time.sleep(0.5)

print(f'手动设置六维力重心参数：{robot.Manual_Set_Force(2, [0, 0, 0, 0, 0, -90])}')

time.sleep(0.5)

print(f'手动设置六维力重心参数：{robot.Manual_Set_Force(3, [0, 0, 0, 0, 0, 90])}')

time.sleep(0.5)

print(f'手动设置六维力重心参数：{robot.Manual_Set_Force(4, [0, 0, 0, 0, -90, 0])}')

time.sleep(0.5)

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.21. 末端五指灵巧手控制（选配）

睿尔曼 RM-65 机械臂末端配备了五指灵巧手，可通过 API 对灵巧手进行设置。

5.21.1. 设置灵巧手手势序号 Set_Hand_Posture

Set_Hand_Posture (posture_num, block)	
描述	设置灵巧手手势序号，设置成功后，灵巧手按照预先保存在 Flash 中的手势运动。
参数	（1）posture_num 预先保存在灵巧手内的手势序号，范围：1~40

	（2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.21.2. 设置灵巧手动作序列序号 Set_Hand_Seq

Set_Hand_Seq (seq_num, block)	
描述	设置灵巧手动作序列序号，设置成功后，灵巧手按照预先保存在 Flash 中的动作序列运动。
参数	（1）seq_num 预先保存在灵巧手内的动作序列序号，范围：1~40 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.21.3. 设置灵巧手角度 Set_Hand_Angle

Set_Hand_Angle(angle, block)	
描述	设置灵巧手角度，灵巧手有 6 个自由度，从 1~6 分别为小拇指，无名指，中指，食指，大拇指弯曲，大拇指旋转。
参数	（1）angle 手指角度数组，6 个元素分别代表 6 个自由度的角度。范围：0~1000。另外，-1 代表该自由度不执行任何操作，保持当前状态 （2）block

	False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.21.4. 设置灵巧手各关节速度 Set_Hand_Speed

Set_Hand_Speed (speed, block)	
描述	设置灵巧手各关节速度
参数	（1）speed 灵巧手各关节速度设置，范围：1~1000 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.21.5. 设置灵巧手各关节力阈值 Set_Hand_Force

Set_Hand_Force (force, block)	
描述	设置灵巧手各关节力阈值。
参数	（1）force 灵巧手各关节力阈值设置，范围：1~1000，代表各关节的力矩阈值（四指握力 0~10N，拇指握力 0~15N）。 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.22. 末端传感器-一维力（选配）

睿尔曼机械臂末端接口板集成了一维力传感器，可获取 Z 方向的力，量程 200N，准度 0.5%FS。

5.22.1. 查询一维力数据 Get_Fz

Get_Fz()	
描述	该函数用于查询末端一维力数据。
参数	
返回值	成功返回：(0, fz, work_fz, tool_fz) fz：原始数据 单位：N zero_fz：传感器坐标系下系统外受力数据 单位：N work_fz：当前工作坐标系下系统外受力数据 单位：N tool_fz：当前工具坐标系下系统外受力数据 单位：N。 失败返回:错误码，查询 API 错误类型。
备注	第一帧指令下发后，开始更新一维力数据，此时返回的数据有滞后性；请从第二帧的数据开始使用。若周期查询 Fz 数据，频率不能高于 40Hz。

5.22.2. 清空一维力数据 Clear_Fz

Clear_Fz(block)	
描述	该函数用于清零末端一维力数据。清空一维力数据后，后续所有获取到的数据都是基于当前的偏置。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设

	置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.22.3. 自动标定末端一维力数据 Auto_Set_Fz

Auto_Set_Fz()	
描述	该函数用于自动标定末端一维力数据。一维力重新安装后，必须重新计算一维力所受到的初始力和重心。分别在不同姿态下，获取一维力的数据，用于计算重心位置，该步骤对于基于一维力的力位混合控制操作具有重要意义
参数	
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.22.4. 手动标定末端一维力数据 Manual_Set_Fz

Manual_Set_Fz(joint1,joint2)	
描述	该函数用于手动标定末端一维力数据。一维力重新安装后，必须重新计算一维力所受到的初始力和重心。该手动标定流程，适用于空间狭窄工作区域，以防自动标定过程中机械臂发生碰撞，用户可以手动选取 2 个位姿下发，当下发完后，机械臂开始自动沿用户设置的目标运动，并在此过程中计算一维力重心
参数	（1）joint 点位 1 关节角度 （2）joint2 点位 2 关节角度
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23. Modbus 配置

睿尔曼机械臂在控制器的航插和末端接口板航插处，各有 1 路 RS485 通讯接口，这两个 RS485 端口可通过接口配置为标准的 ModbusRTU 模式。然后通过接口对端口连接的外设进行读写操作。

注意：控制器的 RS485 接口在未配置为 Modbus RTU 模式的情况下，可用于用户对机械臂进行控制，这两种模式不可兼容。若要恢复机械臂控制模式，必须将该端口的 Modbus RTU 模式关闭。Modbus RTU 模式关闭后，系统会自动切换回机械臂控制模式，波特率 460800BPS，停止位 1，数据位 8，无检验。

同时，I 系列控制器支持 modbus-TCP 主站配置，可配置使用 modbus-TCP 主站，用于连接外部设备的 modbus-TCP 从站。

5.23.1. 设置通讯端口 Modbus RTU 模式 Set_Modbus_Mode

Set_Modbus_Mode (port,baudrate,timeout,block)	
描述	该函数用于配置通讯端口 Modbus RTU 模式。机械臂启动后，要对通讯端口进行任何操作，必须先启动该指令，否则会返回报错信息。 另外，机械臂会对用户的配置方式进行保存，机械臂重启后会自动恢复到用户断电之前配置的模式。
参数	（1）port 通讯端口，0-控制器 RS485 端口为 RTU 主站，1-末端接口板 RS485 接口为 RTU 主站，2-控制器 RS485 端口为 RTU 从站 （2）baudrate 波特率，支持 9600，115200，460800 三种常见波特率

	（3）timeout 超时时间，单位百毫秒。对 Modbus 设备所有的读写指令，在规定的超时时间内未返回响应数据，则返回超时报错提醒。超时时间若设置为 0，则机械臂按 1 进行配置。 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.2. 关闭通讯端口 Modbus RTU 模式 Close_Modbus_Mode

Close_Modbus_Mode (port , block)	
描述	该函数用于关闭通讯端口 Modbus RTU 模式。
参数	（1）port 通讯端口，0-控制器 RS485 端口为 RTU 主站，1-末端接口板 RS485 接口为 RTU 主站，2-控制器 RS485 端口为 RTU 从站 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.3. 配置连接 ModbusTCP 从站 Set_Modbustcp_Mode--I 系列

Set_Modbustcp_Mode (ip, port, timeout)	
描述	该函数用于配置连接 ModbusTCP 从站。
参数	（1）ip

	从机 IP 地址 （2）port 端口号 （3）timeout 超时时间，单位秒。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.4. 配置关闭 ModbusTCP 从站 Close_Modbustcp_Mode--I 系列

Close_Modbustcp_Mode ()	
描述	该函数用于配置关闭 ModbusTCP 从站。
参数	
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.5. 读线圈 Get_Read_Coils

Get_Read_Coils (port, address, num, device)	
描述	该函数用于读线圈。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 线圈起始地址 （3）num 要读的线圈的数量，该指令最多一次性支持读 8 个线圈数据，即返回的数据不会超过一个字节

	（4）device 外设设备地址
返回值	成功返回：(0, coils_data) coils_data：返回线圈状态； 失败返回：错误码，查询 API 错误类型。

5.23.6. 读离散输入量 Get_Read_Input_Status

Get_Read_Input_Status(port,address,num,device)	
描述	读离散输入量。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 数据起始地址 （3）num 要读的数据的数量，该指令最多一次性支持读 8 个离散量数据，即返回的数据不会超过一个字节 （4）device 外设设备地址
返回值	成功返回：(0, coils_data) coils_data：返回离散量； 失败返回：错误码，查询 API 错误类型。

5.23.7. 读保持寄存器 Get_Read_Holding_Registers

Get_Read_Holding_Registers(port,address,device)	
描述	该函数用于读保持寄存器。该函数每次只能读 1 个寄存器，即 2 个字节的数据，不可一次性读取多个寄存器数据。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 数据起始地址 （3）device 外设设备地址
返回值	成功返回：(0,coils_data) coils_data：返回寄存器数据； 失败返回：错误码，查询 API 错误类型。

5.23.8. 读输入寄存器 Get_Read_Input_Registers

Get_Read_Input_Registers(port,address,device)	
描述	该函数用于读输入寄存器。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 数据起始地址

	（3）device 外设设备地址
返回值	成功返回：(0,coils_data) coils_data：返回寄存器数据； 失败返回：错误码，查询 API 错误类型。

5.23.9. 写单圈数据 Write_Single_Coil

Write_Single_Coil(port,address,data,device,block)	
描述	该函数用于写单圈数据。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 线圈起始地址 data 要写入线圈的数据 （3）device 外设设备地址 （4）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.10. 写单个寄存器 Write_Single_Register

Write_Single_Register(port,address,data,device,block)	
描述	该函数用于写单个寄存器。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 寄存器起始地址。 （3）data 要写入寄存器的数据。 （4）device 外设设备地址 （5）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.11. 写多个寄存器 Write_Registers

Write_Registers(port,address,num,single_data, device, block)	
描述	该函数用于写多个寄存器。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备

	（2）address 寄存器起始地址 （3）num 写寄存器个数，寄存器每次写的数量不超过 10 个 （4）single_data 要写入寄存器的数据数组，类型：byte （5）device 外设设备地址 （6）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.12. 写多圈数据 Write_Coils

Write_Coils(port,address,num, coils_data,device,block)	
描述	该函数用于写多圈数据。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 线圈起始地址。 （3）num 写线圈个数，每次写的数量不超过 160 个

	（4）coils_data 要写入线圈的数据数组，类型：byte。若线圈个数不大于 8，则写入的数据为 1 个字节；否则，则为多个数据的数组。 （5）device 外设设备地址 （6）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.23.13. 读多圈数据 Get_Read_Multiple_Coils

Get_Read_Multiple_Coils(port,address,num,device)	
描述	该函数用于读多圈数据。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 线圈起始地址 （3）num 8< num <= 120 要读的线圈的数量，该指令最多一次性支持读 120 个线圈数据， 即 15 个 byte （4）device 外设设备地址

返回值	成功返回：(0,coils_data) coils_data：返回线圈状态； 失败返回：错误码，查询 API 错误类型。

5.23.14. 读多个保持寄存器 Read_Multiple_Holding_Registers

Read_Multiple_Holding_Registers(port, address,num, device)	
描述	该函数用于读多个保持寄存器。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，3-控制器 ModbusTCP 设备 （2）address 寄存器起始地址 （3）num 2 < num < 13 要读的寄存器的数量，该指令最多一次性支持读 12 个寄存器数据， 即 24 个 （4）device 外设设备地址
返回值	成功返回：(0,coils_data) coils_data：返回寄存器数据；失败返回：错误码，查询 API 错误类型。

5.23.15. 读多个输入寄存器 Read_Multiple_Input_Registers

Read_Multiple_Input_Registers(port, address,num, device)	
描述	该函数用于读多个保持寄存器。
参数	（1）port 通讯端口，0-控制器 RS485 端口，1-末端接口板 RS485 接口，

	3-控制器 ModbusTCP 设备 （2）address 寄存器起始地址 （3）num 2 < num < 13 要读的寄存器的数量，该指令最多一次性支持读 12 个寄存器数据， 即 24 个 （4）device 外设设备地址
返回值	成功返回：(0,coils_data) coils_data：返回寄存器数据；失败返回：错误码，查询 API 错误类型。

5.24. 升降机构

睿尔曼机械臂可集成自主研发升降机构。

5.24.1. 升降机构速度开环控制 Set_Lift_Speed

Set_Lift_Speed (speed)	
描述	升降机构速度开环控制。
参数	（1）speed 升降机速度百分比，-100 ~100 speed<0：升降机构向下运动 speed>0：升降机构向上运动 Speed=0: 升降机构停止运动

返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.24.2. 设置升降机构高度 Set_Lift_Height

Set_Lift_Height(height,speed,block)	
描述	该函数用于设置升降机构高度。
参数	（1）height 目标高度，单位 mm，范围：0~2600 （2）speed 升降机速度百分比，1~100 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.24.3. 获取升降机构状态 Get_Lift_State

Get_Lift_State()	
描述	该函数用于获取升降机构状态。
参数	
返回值	成功返回：(0,height,current,err,mode)] height：当前升降机构高度，单位：mm，精度：1mm，范围：0~2300 current：当前升降驱动电流，单位：mA，精度：1mA err：升降驱动错误代码，错误代码类型参考关节错误代码； mode:：当前升降状态，0-空闲，1-正方向速度运动，2-正方向位置运动，3-负方向速度运动，4-负方向位置运动

	失败返回：错误码，查询 API 错误类型。

5.25. 透传力位混合控制补偿

针对睿尔曼带一维力和六维力版本的机械臂，用户除了可直接使用示教器调用底层的力位混合控制模块外，还可以将自定义的轨迹以周期性透传的形式结合底层的力位混合控制算法进行补偿。

在进行力的操作之前，如果未进行力数据标定，可使用清空一维力、六维力数据接口对零位进行标定。

5.25.1. 开启透传力位混合控制补偿模式 Start_Force_Position_Move

Start_Force_Position_Move (block)	
描述	开启透传力位混合控制补偿模式。
参数	（1）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.25.2. 力位混合控制补偿透传模式（关节角度）Force_Position_Move_Joint

Force_Position_Move_Joint(joint,sensor,mode,dir,force, follow)	
描述	该函数用于力位混合控制补偿透传模式（关节角度）。
参数	（1）joint 目标关节角度 （2）sensor 所使用传感器类型，0-一维力，1-六维力

	（3）mode 模式，0-沿基坐标系，1-沿工具端坐标系 （4）dir 力控方向，0~5 分别代表 X/Y/Z/Rx/Ry/Rz，其中一维力类型时默认方向为 Z 方向 （5）force 力的大小 单位 0.1N （6）follow 是否高跟随
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	备注 1：该功能只适用于一维力传感器和六维力传感器机械臂版本 备注 2：透传周期越快，力位混合控制效果越好。基础系列 WIFI 和网口模式透传周期最快 20ms，USB 和 RS485 模式透传周期最快 10ms。高速网口的透传周期最快也可到 10ms，不过在使用该高速网口前，需要使用指令打开配置。另外 I 系列有线网口周期最快可达 5ms

5.25.3. 力位混合控制补偿透传模式（位姿）Force_Position_Move_Pose

Force_Position_Move_Pose(pose,sensor,mode,dir,force, follow)	
描述	该函数用于力位混合控制补偿透传模式（位姿）。
参数	（1）pose 当前坐标系下目标位姿，位姿中包括姿态欧拉角和姿态四元数，四元数合理情况下，优先使用姿态四元数 （2）sensor

	所使用传感器类型，0-一维力，1-六维力 （3）mode 模式，0-沿基坐标系，1-沿工具端坐标系 （4）dir 力控方向，0~5 分别代表 X/Y/Z/Rx/Ry/Rz，其中一维力类型时默认方向为 Z 方向 （5）force 力的大小 单位 0.1N （6）follow 是否高跟随
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。
备注	1、该功能只适用于一维力传感器和六维力传感器机械臂版本 2、透传周期越快，力位混合控制效果越好。基础系列 WIFI 和网口模式透传周期最快 20ms，USB 和 RS485 模式透传周期最快 10ms。高速网口的透传周期最快也可到 10ms，不过在使用该高速网口前，需要使用指令打开配置。另外 I 系列有线网口周期最快可达 5ms。 3、透传开始的起点务必为机械臂当前位姿，否则可能会力控补偿失败或机械臂无法运动

5.25.4. 关闭透传力位混合控制补偿模式 Stop_Force_Position_Move

Stop_Force_Position_Move(block)	
描述	该函数用于关闭透传力位混合控制补偿模式。
参数	（1）block

	False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0；失败返回：错误码，查询 API 错误类型。

5.26. 算法工具接口

针对睿尔曼机械臂，提供正解、逆解等工具接口。

算法接口可单独使用，也可连接机械臂使用。连接机械臂使用时，为保证算法所用的参数是机械臂当前的数据，需调用获取工作、工具坐标系，获取安装角度等接口同步信息，否则算法将使用 API 与机械臂创建连接时机械臂的坐标系、安装角度等数据计算。

5.26.1. 初始化算法依赖数据 Algo_Init_Sys_Data

Algo_Init_Sys_Data(dMode, bType)	
描述	初始化算法依赖数据(不连接机械臂时调用， 连接机械臂会自动调用)。
参数	（1）dMode 机械臂型号，RobotType 结构体 （2）bType 传感器型号，SensorType 结构体

5.26.2. 设置算法的安装角度 Algo_Set_Angle

Algo_Set_Angle(x, y, z)	
描述	设置算法的安装角度参数。
参数	（1）x X 轴安装角度，单位度。

	（2）y Y 轴安装角度，单位度。 （3）z Z 轴安装角度，单位度。

5.26.3. 获取算法的安装角度 Algo_Get_Angle

Algo_Get_Angle()	
描述	设置算法的安装角度参数。
返回值	(x, y, z) x：X 轴安装角度，单位度。 y：Y 轴安装角度，单位度。 z：Z 轴安装角度，单位度。

5.26.4. 设置算法工作坐标系 Algo_Set_WorkFrame

Algo_Set_WorkFrame(coord_work)	
描述	设置算法工作坐标系。
参数	（1）coord_work 工作坐标系数据

5.26.5. 获取当前工作坐标系

Algo_Get_Curr_WorkFrame()	
描述	获取算法工作坐标系。
返回值	coord_work：工作坐标系数据

5.26.6. 设置算法工具坐标系 Algo_Set_ToolFrame

Algo_Set_ToolFrame(coord_tool)	
描述	设置算法工具坐标系和负载。
参数	（1）coord_tool 工具坐标系数据

5.26.7. 获取算法当前工具坐标系

Algo_Get_Curr_ToolFrame()	
描述	设置算法工具坐标系和负载。
返回值	coord_tool：坐标系数据

5.26.8. 正解 Algo_Forward_Kinematics

Algo_Forward_Kinematics(joint)	
描述	用于睿尔曼机械臂正解计算。
参数	（1）joint 关节 1 到关节 7 角度，单位度。
返回值	返回求解得目标位姿[x, y, z, rx, ry, rz]

5.26.9. 逆解 Algo_Inverse_Kinematics

Algo_Inverse_Kinematics(q_in, q_pose,q_out, flag)	
描述	用于睿尔曼机械臂逆解计算。
参数	（1）q_in 上一时刻关节角，单位度。 （2）q_pose 目标位姿。

	（3）flag 姿态参数类别：0-四元数；1-欧拉角
返回值	成功返回：(0, q_out) q_out：输出的关节角度，单位度； 计算失败返回：CALCULATION_FAILED；

5.26.10. 计算平移、旋转运动位姿 Algo_PoseMove

Algo_PoseMove(poseCurrent, deltaPosAndRot, frameMode)	
描述	用于计算 Pos 和 Rot 沿某坐标系有一定的位移和旋转角度后，所得到的位姿数据。
参数	（1）poseCurrent 当前位姿，输入 position 和 euler。 （2）deltaPosAndRot 沿轴位移和绕轴旋转数组（dx，dy，dz，rotx, roty, rotz），位置移动单位：m，旋转单位：度 （3）frameMode 坐标系模式选择。 0：计算相对于工作坐标系平移、旋转后的位姿，当工作坐标系为 0 时，即为计算相对基坐标系的位姿； 1：计算相对于工具坐标系平移、旋转后的位姿
返回值	返回经平移、旋转后的位姿[x, y, z, rx, ry, rz]

5.26.11. 计算环绕运动位姿 RotateMove

Algo_RotateMove(curr_joint, rotate_axis,rotate_angle, choose_axis)	
描述	用于计算环绕运动位姿。

参数	（1）curr_joint 当前关节角度，单位度。 （2）rotate_axis 旋转轴：1：x 轴， 2：y 轴，3：z 轴 （3）rotate_angle 旋转角度：旋转角度，单位（度） （4）choose_axis 指定计算时使用的坐标系
返回值	返回求解得目标位姿[x, y, z, rx, ry, rz]

5.26.12. 末端位姿转成工具位姿 end2tool

Algo_End2Tool(eu_end)	
描述	末端位姿转成工具位姿。即为：机械臂末端在基坐标系下的位姿，转化成工具坐标系末端在工作坐标系下的位姿
参数	（1）eu_end 基于世界坐标系和默认工具坐标系的末端位姿
返回值	工具位姿[x, y, z, rx, ry, rz]

5.26.13. 工具位姿转末端位姿 tool2end

Algo_Tool2End(eu_tool)	
描述	末端位姿转成工具位姿。即为：工具坐标系末端在工作坐标系下的位姿，转换成机械臂末端在基坐标系下的位姿
参数	（1）eu_tool 基于工作坐标系和工具坐标系的末端位姿

返回值	末端位姿[x, y, z, rx, ry, rz]

5.26.14. 四元数转欧拉角 quaternion2euler

Algo_Quaternion2Euler(qua)	
描述	四元数转欧拉角 。
参数	（1） qua 四元数
返回值	欧拉角

5.26.15. 欧拉角转四元数 euler2quaternion

Algo_Euler2Quaternion(eu)	
描述	欧拉角转四元数。
参数	（1）eu 欧拉角
返回值	四元数

5.26.16. 欧拉角转旋转矩阵 euler2matrix

Algo_Euler2Matrix(eu)	
描述	欧拉角转旋转矩阵。
参数	（1）eu 欧拉角
返回值	旋转矩阵

5.26.17. 位姿转旋转矩阵 pos2matrix

Algo_Pos2Matrix(state)

描述	位姿转旋转矩阵。
参数	（1）state 位姿
返回值	旋转矩阵

5.26.18. 旋转矩阵转位姿 matrix2pos

Algo_Matrix2Pos(matrix)	
描述	旋转矩阵转位姿。
参数	（1）matrix 旋转矩阵
返回值	位姿

5.26.19. 基坐标系转工作坐标系 Base_To_WorkFrame

Algo_Base2WorkFrame(matrix, state)	
描述	基坐标系转工作坐标系。
参数	（1）matrix 工作坐标系在基坐标系下的矩阵 （2）state 工具端坐标在基坐标系下位姿
返回值	工作坐标系下的位姿

5.26.20. 工作坐标系转基坐标系 WorkFrame_To_Base

Algo_WorkFrame2Base(matrix, state)	
描述	工作坐标系转基坐标系。
参数	（1）matrix

	工作坐标系在基坐标系下的矩阵 （2）state 工具端坐标在工作坐标系下位姿
返回值	基坐标系下的位姿

5.26.21. 计算沿工具坐标系运动位姿 Cartesian_Tool

Algo_Cartesian_Tool(curr_joint, move_lengthx,move_lengthy, move_lengthz)	
描述	计算沿工具坐标系运动位姿。
参数	（1）curr_joint 当前关节角度，单位度 （2）move_lengthx 沿 X 轴移动长度，米为单位 （3）move_lengthy 沿 Y 轴移动长度，米为单位 （4）move_lengthz 沿 Z 轴移动长度，米为单位
返回值	基坐标系下的位姿

5.26.22. 设置算法关节最大限位 Set_Algo_Joint_Max_Limit

Algo_Set_Joint_Max_Limit(joint_limit)	
描述	设置算法关节最大限位。
参数	（1）Joint_limit 关节的最大限位数组

5.26.23. 获取算法关节最大限位 Get_Algo_Joint_Max_Limit

Algo_Get_Joint_Max_Limit()	
描述	获取算法关节最大限位。
返回值	joint_limit：关节的最大限位数组

5.26.24. 设置算法关节最小限位 Set_Algo_Joint_Min_Limit

Algo_Set_Joint_Min_Limit(joint_limit)	
描述	设置算法关节最大限位。
参数	（1）Joint_limit 关节的最小限位数组

5.26.25. 获取算法关节最小限位

Algo_Get_Joint_Min_Limit()	
描述	获取算法关节最大限位。
返回值	joint_limit：关节的最小限位数组

5.26.26. 设置算法关节最大速度 Set_Algo_Joint_Max_Speed

Algo_Set_Joint_Max_Speed(joint_slim_max)	
描述	设置算法关节最大速度。
参数	（1）Joint_slim_max 关节的最大速度数组

5.26.27. 获取算法关节最大速度

Algo_Get_Joint_Max_Speed()	
描述	获取算法关节最大速度。
返回值	Joint_slim_max：关节的最大速度数组

5.26.28. 设置算法关节最大加速度 Set_Algo_Joint_Max_Acc

Algo_Set_Joint_Max_Acc(joint_alim_max)	
描述	设置算法关节最大加速度。
参数	（1）Joint_alim_max 关节的最大加速度数组

5.26.29. 获取算法关节最大加速度 Get_Algo_Joint_Max_Acc

Algo_Get_Joint_Max_Acc()	
描述	获取算法关节最大加速度。
返回值	joint_alim_max：返回的关节的最大加速度数组

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 不连接机械臂的情况下，调用算法接口
dmode = RobotType.RM65

rbt_type = SensorType.B

Arm.Algo_Init_Sys_Data(dmode, rbt_type)

# 设置算法的安装角度为 Y 轴 90 °
Arm.Algo_Set_Angle(0, 90, 0)

# 设置算法的工作坐标系
coord_work = FRAME()

coord_work.frame_name.name = "123".encode()

coord_work.pose.position.x = 1

coord_work.pose.position.y = 12

coord_work.pose.position.z = 123
```

```
coord_work.pose.euler.rx = 0.5

coord_work.pose.euler.ry = 1

coord_work.pose.euler.rz = 1.5

Arm.Algo_Set_WorkFrame(coord_work)

curr_pose = Arm.Algo_Get_Curr_WorkFrame()

print(f"当前工作坐标系：{curr_pose.frame_name.name, curr_pose.pose.position.x,
curr_pose.pose.position.y, curr_pose.pose.position.z, curr_pose.pose.euler.rx,
curr_pose.pose.euler.ry, curr_pose.pose.euler.rz}")

# 计算正解结果

joint = [0, 0, 90, 0, 90, 0]

compute_pose = Arm.Algo_Forward_Kinematics(joint)

print(f'正解：{compute_pose}')

# 计算逆解（使用四元数表示姿态）
tar = [0, -0.342, 0.2, 1.571, 0, 0, 1]

q_in = [-35.28, 112.283, 88.039, -130.327, 11.468, 73.635]

q_out = Arm.Algo_Inverse_Kinematics(q_in, tar, 0)

print(f'逆解·：{q_out}')

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.27. 在线编程

5.27.1. 文件下发 Send_TrajectoryFile

```
Send_TrajectoryFile(send_params)
```

描述	在线编程文件下发。
参数	（1）send_params 文件下发参数
返回值	成功返回：(0, err_line) err_line：有问题的工程行数。 失败返回：错误码，查询 API 错误类型

5.27.2. 轨迹规划中改变速度比例系数 Set_Plan_Speed

Set_Plan_Speed(speed, block)	
描述	该函数用于轨迹规划中改变速度比例系数。
参数	（1）speed 当前进度条的速度数据 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接机械臂
robot = Arm(RM65, "192.168.1.18")

# 文件下发,以 20%速度运行

ret = robot.Send_TrajectoryFile("H:/program1.txt", 20, 0, 0)

# 改变当前进度条的速度数据
speed = random.randint(1, 99)
```

```
print(f"改变速度为{speed}：{robot.Set_Plan_Speed(speed)}")

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.28. 机械臂状态主动上报

5.28.1. 设置主动上报配置 Set_Realtime_Push

Set_Realtime_Push(cycle=-1, port=-1, enable=True, force_coordinate=-1, ip=None, joint_speed=-1, lift_state=-1, expand_state=-1)	
描述	该函数用于设置主动上报接口配置。以下参数可分开设置，均为可选字段
参数	（1）cycle 设置广播周期，为 5ms 的倍数 （2）port 设置广播的端口号 （3）enable 设置使能，是否使能主动上上报，默认使能主动上报 （4）force_coordinate 系统外受力数据的坐标系，0 为传感器坐标系 1 为当前工作坐标系 2 为当前工具坐标系 （5）ip 自定义的上报目标 IP 地址 （6）joint_speed

	关节速度。1：上报；0：关闭上报；-1：不设置，保持之前的状态 （7）lift_state 升降关节信息。1：上报；0：关闭上报；-1：不设置，保持之前的状态 （8）expand_state 扩展关节信息（升降关节和扩展关节为二选一，优先显示升降关节）1：上报；0：关闭上报；-1：不设置，保持之前的状态
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.28.2. 获取主动上报配置 Get_Realtime_Push

Get_Realtime_Push()	
描述	该函数用于获取主动上报接口配置。
参数	
返回值	成功返回：(0, cycle, port, enable, force_coordinate, ip) cycle：获取广播周期，为 5ms 的倍数 port：获取广播的端口号 enable：获取使能，是否使能主动上上报 force_coordinate：系统外受力数据的坐标系，0 为传感器坐标系 1 为当前工作坐标系 2 为当前工具坐标系 ip：自定义的上报目标 IP 地址； 失败返回：错误码，查询 API 错误类型

5.28.3. 机械臂状态主动上报 Realtime_Arm_Joint_State

Realtime_Arm_Joint_State(RobotStatusListener RobotStatuscallback)	
描述	该函数该函数使用 UDP 协议监听本机广播的端口号，接收机械臂状态广播数据。可注册回调函数来处理机械臂状态信息。
参数	（1）RobotStatuscallback 用于接收机械臂状态广播回调函数。

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接机械臂
robot = Arm(RM65, "192.168.1.18")

# 关闭主动上报接口

ret = robot.Set_Realtime_Push(enable=False)

# 设置广播周期 100ms，端口号 8089，目标 IP 为"192.168.1.20"
ret = robot.Set_Realtime_Push(cycle=20, port=8089, ip="192.168.1.20")

# 查询主动上报配置

error_code, cycle, port, enable, force_coordinate, ip = robot.Get_Realtime_Push()

# 注册机械臂状态主动上报回调函数
def robotstatus (data):

    print("RobotStatus RobotStatus RobotStatus")

    print("当前角度:", data.joint_status.joint_position[0], data.joint_status.joint_position[1],

          data.joint_status.joint_position[2])

    print("当前力:", data.force_sensor.force[0], data.force_sensor.force[1],
data.force_sensor.force[2])

    print("err:", data.errCode)
```

```
robotstatus = RealtimePush_Callback(robotstatus )

robot.Realtime_Arm_Joint_State(robotstatus)

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.29. 通用扩展关节

5.29.1. 关节速度环控制 Expand_Set_Speed

Expand_Set_Speed(speed, block)	
描述	扩展关节速度环控制。
参数	（1）speed -50 表示最大速度的百分之五十反方向运动 （2）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.29.2. 关节位置环控制 Expand_Set_Pos

Expand_Set_Pos(pos, speed, block)	
描述	该函数用于扩展关节位置环控制。
参数	（1）pos

	升降关节精度 1mm 旋转关节精度 0.001° （2）speed 50 表示最大速度的百分之五十,且速度必须大于 0 （3）block False-非阻塞，发送后立即返回； True-阻塞，等待控制器返回设置成功指令。
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.29.3. 扩展关节状态获取 Expand_Get_State

Expand_Get_State()	
描述	该函数用于获取扩展关节状态。
参数	
返回值	成功返回：(0,pos, err_flag, current, mode) pos：当前升降机构高度，单位：mm，精度：1mm，如果是旋转关节则为角度 单位度，精度 0.001° err：升降驱动错误代码，错误代码类型参考关节错误代码 current：当前升降驱动电流，单位：mA，精度：1mA mode：当前升降状态，0-空闲，1-正方向速度运动，2-正方向位置运动，3-负方向速度运动，4-负方向位置运动。 失败返回：错误码，查询 API 错误类型

5.30. 在线编程存储列表（I 系列）

5.30.1. 查询在线编程程序列表 Get_Program_Trajectory_List

Get_Program_Trajectory_List(page_num=0, page_size=0, vague_search=None)	
描述	查询在线编程程序列表。
参数	（1）page_num 页码（全部查询时此参数传 0） （2）page_size 每页大小（全部查询时此参数传 0） （3）vague_search 模糊搜索 （传递此参数可进行模糊查询）
返回值	成功返回：(0, program_list) program_list：符合条件的在线编程列表。 失败返回：错误码，查询 API 错误类型

5.30.2. 查询当前在线编程文件的运行状态 Get_Program_Run_State

Get_Program_Run_State(cycle_num)	
描述	该函数用于查询当前在线编程文件的运行状态。
参数	（1）cycle_num 循环指令数量
返回值	成功返回：(0, state) state：在线编程运行状态结构体 失败返回：错误码，查询 API 错误类型

5.30.3. 运行指定编号在线编程 Set_Program_ID_Start

Set_Program_ID_Start(id, speed=0, block=True)	
描述	该函数用于运行指定编号在线编程。

参数	（1）id 运行指定的 ID，1-100，存在轨迹可运行 （2）speed 1-100，需要运行轨迹的速度，按照存储的速度运行则传入 0 （3）block 0-非阻塞，开始运行后返回；1-阻塞，等待在线编程程序运行结束返回
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.30.4. 删除指定 ID 的轨迹 Delete_Program_Trajectory

Delete_Program_Trajectory(id)	
描述	该函数用于删除指定 ID 的轨迹。
参数	（1）id 指定需要删除的轨迹编号
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.30.5. 修改指定编号轨迹的信息 Update_Program_Trajectory

Update_Program_Trajectory(id, plan_speed, project_name)	
描述	该函数用于修改指定编号轨迹信息。
参数	（1）id 指定在线编程轨迹编号 （2）speed 更新后的规划速度比例 1-100 （3）Project_name

	更新后的文件名称（最大 10 个字节）
返回值	成功返回：0 失败返回：错误码，查询 API 错误类型

5.30.6. 设置 IO 默认运行的在线编程文件编号 Set_Default_Run_Program

Set_Default_Run_Program(id)	
描述	该函数用于设置 IO 默认运行的在线编程文件编号。
参数	（1）id 设置 IO 默认运行的在线编程文件编号，支持 0-100，0 代表取消设置
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.30.7. 获取 IO 默认运行的在线编程文件编号 Get_Default_Run_Program

Get_Default_Run_Program( )	
描述	该函数用于获取 IO 默认运行的在线编程文件编号。
参数	
返回值	成功返回：(0, id) id：IO 默认运行的在线编程文件编号，支持 0-100，0 代表无默认 失败返回：错误码，查询 API 错误类型

代码示例：

```
from robotic_arm_package.robotic_arm import *


# 连接机械臂
robot = Arm(RM65, "192.168.1.18")

# 模糊查询，查询名称有“2”的在线编程轨迹

ret = robot.Get_Program_Trajectory_List(vague_search='2')
print(f'查询结果：{ret[0]},列表长度：{ret[1].total_size}，符合条件的程序列表元素：
```

```
{ret[1].list[0].trajectory_name}')

# 非阻塞阻塞运行 id 为 2 的程序

ret = robot.Set_Program_ID_Start(2, block=False)

# 获取程序运行状态，运行的程序中包含 3 个循环指令

ret = robot.Get_Program_Run_State(3)
print(f'查询结果{ret[0]}，运行状态{ret[1]}，运行 id{ret[2]}，运行行数{ret[3]},循环行数{ret[4]}，循环
状态{ret[5]}')

# 删除 id 为 1 的程序

ret = robot.Delete_Program_Trajectory(1)

# 断开连接
robot.RM_API_UnInit()

robot.Arm_Socket_Close()
```

5.31. 全局路点（I 系列）

5.31.1. 新增全局路点 Add_Global_Waypoint

Add_Global_Waypoint(waypoint)	
描述	该函数用于新增全局路点。
参数	（1）waypoint 新增全局路点参数结构体（无需输入新增全局路点时间）
返回值	成功返回：0 失败返回：错误码，查询 API 错误类型
示例	point_name = "p1" waypoint = Waypoint(point_name, [18.44, -10.677, -124.158, -15,

	-71.455, 0], [0.186350, 0.062099, 0.2, 3.141, 0, 1.569], 'World', 'Arm_Tip') print(robot.Add_Global_Waypoint(waypoint))

5.31.2. 更新全局路点 Update_Global_Waypoint

Update_Global_Waypoint(waypoint)	
描述	该函数用于更新全局路点。
参数	（1）waypoint 更新全局路点参数（无需输入新增全局路点时间）
返回值	成功返回：0 失败返回：错误码，查询 API 错误类型
示例	point_name = "p1" waypoint = Waypoint(point_name, [30, -10.677, -110, -15, -71.455, 0], [0.186350, 0.062099, 0.2, 3.141, 0, 1.569], 'World', 'Arm_Tip') print(robot.Update_Global_Waypoint(waypoint))

5.31.3. 删除全局路点 Delete_Global_Waypoint

Delete_Global_Waypoint(name)	
描述	该函数用于删除全局路点。
参数	（1）name 全局路点名称
返回值	成功返回：0。失败返回：错误码，查询 API 错误类型

5.31.4. 查询多个全局路点 Get_Global_Point_List

Get_Global_Point_List(page_num=0, page_size=0, vague_search=None)

描述	该函数用于查询多个全局路点。
参数	（1）page_num 页码（0 代表全部查询） （2）page_size 每页大小（0 代表全部查询） （3）vague_search 模糊搜索 （传递此参数可进行模糊查询）
返回值	成功返回：(0, point_list) point_list：符合条件的全局路点列表。 失败返回：错误码，查询 API 错误类型
示例	print(robot.Get_Global_Point_List(vague_search="test")[1]) **输出结果：** {'page_num': 0, 'page_size': 0, 'total_size': 6, 'vague_search': '', 'points_list': [{'point_name': 'test1', 'joint': [18.44, -10.677, -124.158, -15.0, -71.455, 0.0, 0.0], 'pose': [0.17634999752044678, 0.062098998576402664, 0.20000000298023224, 3.1410000324249268, 0.0, 1.569000005722046], 'work_frame': 'World', 'tool_frame': 'Arm_Tip', 'time': '2024-03-26 16:03:15'}]}

5.31.5. 查询指定全局路点 Given_Global_Waypoint

Given_Global_Waypoint(name)	
描述	该函数用于查询指定全局路点。
参数	（1）name

	指定全局路点名称
返回值	成功返回：(0, point) point：指定全局路点的参数。 失败返回：错误码，查询 API 错误类型
示例	print(robot.Given_Global_Waypoint("test1")[1]) **输出结果：**{'point_name': 'test1', 'joint': [18.44, -10.677, -124.158, -15.0, -71.455, 0.0, 0.0], 'pose': [0.17634999752044678, 0.062098998576402664, 0.20000000298023224, 3.1410000324249268, 0.0, 1.569000005722046], 'work_frame': 'World', 'tool_frame': 'Arm_Tip', 'time': '2024-03-26 16:03:15'}

5.32. 电子围栏和虚拟墙（I 系列）

I 系列机械臂具备电子围栏与虚拟墙功能，并提供了针对控制器所保存的电子围栏或虚拟墙几何模型参数的操作接口。用户可以通过这些接口，实现对电子围栏或虚拟墙的新增、查询、更新和删除操作，在使用中，可以灵活的使用保存在控制器中的参数配置，需要注意的是，目前控制器支持保存的参数要求不超过 10 个。

电子围栏功能通过精确设置参数，确保机械臂的轨迹规划、示教等运动均在设定的电子围栏范围内进行。当机械臂的运动轨迹可能超出电子围栏的界限时，系统会立即返回相应的错误码，并自动中止运动，从而有效保障机械臂的安全运行。需要注意的是，电子围栏目前仅支持长方体和点面矢量平面这两种形状，并且其仅在仿真模式下生效，为用户提供一个预演轨迹与进行轨迹优化的安全环境。

虚拟墙功能支持在电流环拖动示教与力控拖动示教两种模式下，对拖动范围进行精确限制。在这两种特定的示教模式下，用户可以借助虚拟墙功能，确保机械臂的拖动操作不会超出预设的范围。但请务必注意，虚拟墙功能目前支持长方体和球体两种形状，并仅在上述两种示教模式下有效。在其他操作模式下，此功能将自动失效。因此，请确保在正确的操作模式下使用虚拟墙功能，以充分发挥其限制拖动范围的作用。

5.32.1. 新增几何模型参数 Add_Electronic_Fence_Config

Add_Electronic_Fence_Config(config)	
描述	该函数用于新增几何模型参数，最多支持 10 个几何模型。
参数	（1）config 几何模型参数结构体
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.2. 更新几何模型参数 Update_Electronic_Fence_Config

Update_Electronic_Fence_Config(config)	
描述	该函数用于更新几何模型参数，最多支持 10 个几何模型。
参数	（1）config 几何模型参数结构体
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.3. 删除几何模型参数 Delete_Electronic_Fence_Config

Delete_Electronic_Fence_Config(name)

描述	该函数用于更新几何模型参数，最多支持 10 个几何模型。
参数	（1）name 指定几何模型名称
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.4. 查询所有几何模型名称 Get_Electronic_Fence_List_Names

Get_Electronic_Fence_List_Names()	
描述	该函数用于查询所有几何模型名称，最多支持 10 个几何模型。
参数	
返回值	成功返回：(0,names,len) names：几何模型名称列表，长度为实际存在几何模型； len：几何模型名称列表长度； 失败返回：错误码，查询 API 错误类型

5.32.5. 查询指定几何模型参数 Given_Electronic_Fence_Config

Given_Electronic_Fence_Config(name)	
描述	该函数用于查询指定几何模型信息，最多支持 10 个几何模型。
参数	（1）name 指定几何模型名称
返回值	成功返回：(0,config) config：返回指定几何模型参数； 失败返回：错误码，查询 API 错误类型

5.32.6. 查询所有几何模型信息 Get_Electronic_Fence_List_Info

Get_Electronic_Fence_List_Info()	
描述	该函数用于查询所有几何模型信息，最多支持 10 个几何模型。
参数	
返回值	成功返回：(0,list_info,len) list_info：几何模型信息列表，长度为实际存在几何模型； len：几何模型信息列表长度 失败返回：错误码，查询 API 错误类型

5.32.7. 设置电子围栏使能状态 Set_Electronic_Fence_Enable

Set_Electronic_Fence_Enable(enable_state, in_out_side,effective_region)	
描述	该函数用于设置电子围栏使能状态
参数	（1）enable_state true 代表使能，false 代表禁使能 （2）in_out_side 0-机器人在电子围栏内部，1-机器人在电子围栏外部 （3）effective_region 0-针对整臂区域生效
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.8. 获取电子围栏使能状态 Get_Electronic_Fence_Enable

Get_Electronic_Fence_Enable()	
描述	该函数用于获取电子围栏使能状态

参数	
返回值	成功返回：(0,enable_state, in_out_side,effective_region) enable_state：true 代表使能，false 代表禁使能； in_out_side：0-机器人在电子围栏内部，1-机器人在电子围栏外部 effective_region：0-针对整臂区域生效。 失败返回：错误码，查询 API 错误类型

5.32.9. 设置当前电子围栏参数 Set_Electronic_Fence_Config

Set_Electronic_Fence_Config(config)	
描述	该函数用于设置当前电子围栏参数
参数	（1）config 当前电子围栏参数（无需设置电子围栏名称）
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.10. 获取当前电子围栏参数 Get_Electronic_Fence_Config

Get_Electronic_Fence_Config()	
描述	该函数用于获取当前电子围栏参数
参数	
返回值	成功返回：(0,config) config：当前电子围栏参数结构体（返回参数中不包含电子围栏名称）； 失败返回：错误码，查询 API 错误类型

5.32.11. 设置虚拟墙使能状态 Set_Virtual_Wall_Enable

Set_Virtual_Wall_Enable(enable_state, in_out_side,effective_region)

描述	该函数用于设置虚拟墙使能状态
参数	（1）enable_state true 代表使能，false 代表禁使能 （2）in_out_side 0-机器人在虚拟墙内部 （3）effective_region 1-针对末端生效
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.12. 获取虚拟墙使能状态 Get_Virtual_Wall_Enable

Get_Virtual_Wall_Enable()	
描述	该函数用于获取虚拟墙使能状态
参数	
返回值	成功返回：(0,enable_state, in_out_side,effective_region) enable_state：true 代表使能，false 代表禁使能； in_out_side：0-机器人在虚拟墙内部 effective_region：1-针对末端生效。 失败返回：错误码，查询 API 错误类型

5.32.13. 设置当前虚拟墙参数 Set_Virtual_Wall_Config

Set_Virtual_Wall_Config(config)	
描述	该函数用于设置当前虚拟墙参数
参数	（1）config

	当前虚拟墙参数（无需设置虚拟墙名称）
返回值	成功返回：0。 失败返回：错误码，查询 API 错误类型

5.32.14. 获取当前虚拟墙参数 Get_Virtual_Wall_Config

Get_Virtual_Wall_Config()	
描述	该函数用于获取当前虚拟墙参数
参数	
返回值	成功返回：(0,config) config：当前虚拟墙参数结构体（返回参数中不包含虚拟墙名称）； 失败返回：错误码，查询 API 错误类型

5.33. 自碰撞安全检测（I 系列）

I 系列机械臂支持自碰撞安全检测，自碰撞安全检测使能状态下，可确保在轨迹规划、示教等运动过程中机械臂的各个部分不会相互碰撞，需要注意的是，以上自碰撞安全检测功能目前只在仿真模式下生效，用于进行预演轨迹与轨迹优化。

5.33.1. 设置自碰撞安全检测使能状态 Set_Self_Collision_Enable

Set_Self_Collision_Enable(enable_state)	
描述	该函数用于设置自碰撞安全检测使能状态
参数	（1）enable_state true 代表使能，false 代表禁使能
返回值	成功返回：0。

	失败返回：错误码，查询 API 错误类型

5.33.2. 获取自碰撞安全检测使能状态 Get_Self_Collision_Enable

Get_Self_Collision_Enable()	
描述	该函数用于获取自碰撞安全检测使能状态
参数	
返回值	成功返回：(0,enable_state) enable_state：true 代表使能，false 代表禁使能； 失败返回：错误码，查询 API 错误类型