WLKATA

Python SDK

用户手册

(技术开发文档)

文档版本：V1.0.2

发布日期：2025/10/22

目录

1. Python SDK 简介 ..... - 2 -
1.1. Python SDK 概述 ..... - 2 -
1.2. PIP 安装 ..... - 4 -
2. Python SDK 控制函数介绍 ..... - 4 -
2.1. 调用库 ..... - 4 -
2.2. 创建机械臂对象(Mirobot) ..... - 5 -
2.3. 设置机械臂地址 ..... - 5 -
2.4. 多功能控制器重启 ..... - 5 -
2.5. 回零 ..... - 6 -
2.6. 机械臂回到初始位置 ..... - 7 -
2.7. 执行离线文件 ..... - 8 -
2.8. 机械臂停止 ..... - 9 -
2.9. 气泵控制 ..... - 11 -
2.10. 夹爪控制 ..... - 12 -
2.11. PWM 控制 ..... - 13 -
2.12. 透传指令 ..... - 14 -
2.13. 固件版本查询 ..... - 14 -
2.14. 获取机械臂全部状态 ..... - 15 -
2.15. 获取机械臂运动状态 ..... - 16 -
2.16. 获取机械臂各轴角度 ..... - 17 -

2.17. 获取机械臂坐标 ..... - 18 -
2.18. 获取机械臂运动模式 ..... - 19 -
2.19. 笛卡尔坐标控制机械臂(Mirobot) ..... - 20 -
2.20. 角度控制机械臂(Mirobot) ..... - 23 -
2.21. 机械臂速度 ..... - 27 -
2.22. 机械臂拓展轴控制 ..... - 27 -
2.23. 创建机械臂对象(E4) ..... - 27 -
2.24. 笛卡尔坐标控制机械臂(E4) ..... - 28 -
2.25. 角度控制机械臂(E4) ..... - 30 -
2.26. 创建机械臂对象(MT4) ..... - 31 -
2.27. 笛卡尔坐标控制机械臂(MT4) ..... - 32 -
2.28. 角度控制机械臂(MT4) ..... - 33 -
2.29. 创建步进电机控制器对象（MS4220） ..... - 35 -
2.30. 设置步进电机控制器地址（MS4220） ..... - 35 -
2.31. 步进电机控制器透传指令（MS4220） ..... - 35 -
2.32. 步进电机控制器重启（MS4220） ..... - 36 -
2.33. 步进电机控制器回零（MS4220） ..... - 36 -
2.34. 步进电机控制器距离移动（MS4220） ..... - 36 -
2.35. 步进电机控制器速度移动（MS4220） ..... - 36 -
2.36. 步进电机状态全部获取（MS4220） ..... - 37 -
2.37. 创建 Mirobot_Serial_GUI 对象 ..... - 38 -
2.38. 调取 Mirobot_GUI ..... - 38 -

2.39. 创建机械臂对象(Haro380) ..... - 39 -

2.40. 笛卡尔坐标控制机械臂(Haro380) ..... - 39 -

2.41. 角度控制机械臂(Haro380) ..... - 41 -

必　读

如实现本手册所示的各种功能，本手册描述的 wlkatapython 版本为 **0.1.1，**且需将多功能控制器固件升级至 **20230710 及以上版本**，Mirobot 机械臂固件升级至 **20231129 及以上版本**，E4 机械臂升级至 **20231031 及以上版本，**MT4 机械臂升级至 **2024082 及以上版本。**Haro380 机械臂本体固件需升级到 **20250804** 版本，多功能控制器需升级到 **20250701** 版本，可下载 **WlkataStudio-Windows-V3.X.X** 配合使用，并对机械臂及多功能控制器进行升级。

更多信息请参照：https://github.com/wlkata/WLKATA-Python-SDK-wlkatapython

1. Python SDK 简介

1.1. Python SDK 概述

本模块提供 Mirobot 机械臂、E4 机械臂、MT4 机械臂、滑轨（RS485）的控制方式，需要注意的是控制端需要连接到多功能控制器进行通信，不能控制端直接连接到机械臂本体。控制端与多功能控制器之间可采用 RS485、UART 进行通信。

本模块是基于 pyserial 库进行通信的，故使用本模块的同时需要加载 pyserial 库。

使用本模块请使用 Python3.9 以上版本，且多功能控制器升级到 20230710 以上版本。

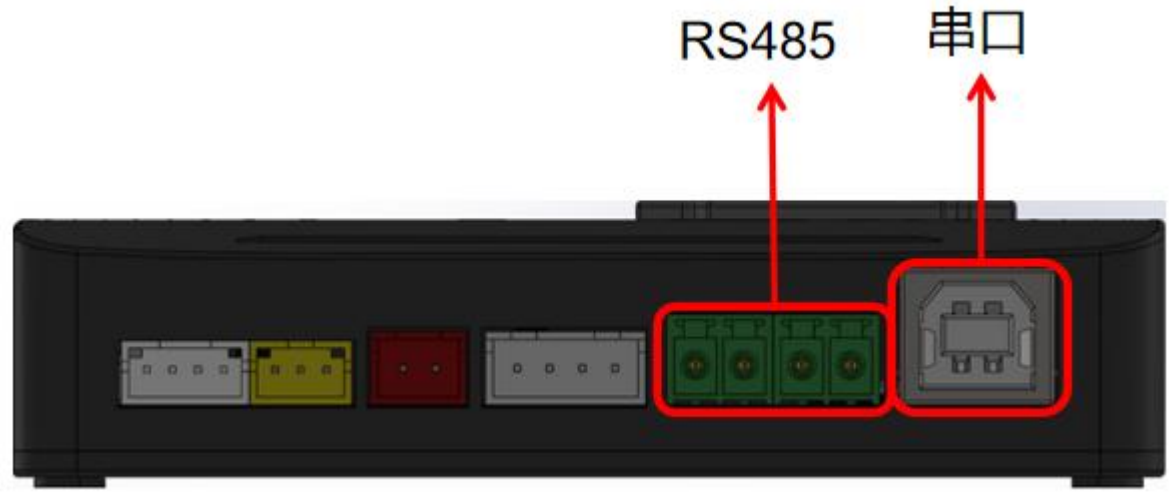
本手册所描述内容对应的模块版本为：

wlkatapython	0.1.1
pyserial	3.5

函数所给出的示例程序都是针对控制端与多功能控制器 RS485 通信进行的。在本文末尾有针对控制端与多功能控制器串口通信的示例程序。

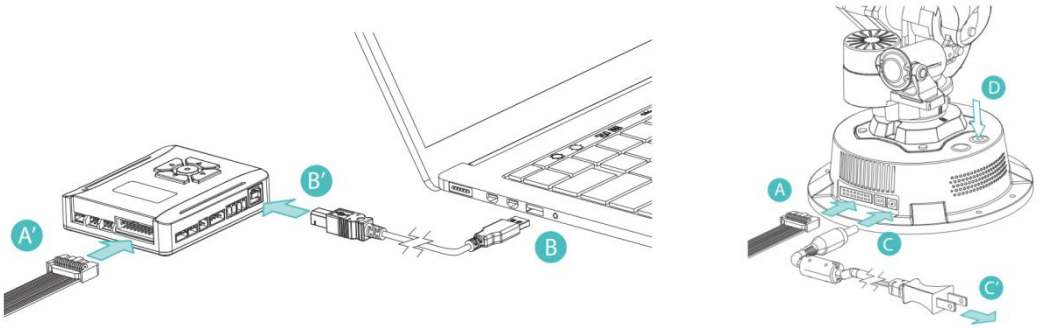
- Mirobot_UART 类主要控制的是 Mirobot 机械臂及其拓展轴。
- E4_UART 类主要控制的是 E4 机械臂及其拓展轴。
- MT4_UART 类主要控制的是 MT4 机械臂及其拓展轴。
- MS4220 类主要控制的是 MS4220 步进电机控制器。

MT4/E4 类是继承自 Mirobot_UART 类，故除笛卡尔坐标控制和角度控制中的参数不同外其他控制函数通用。详细用法见下文。

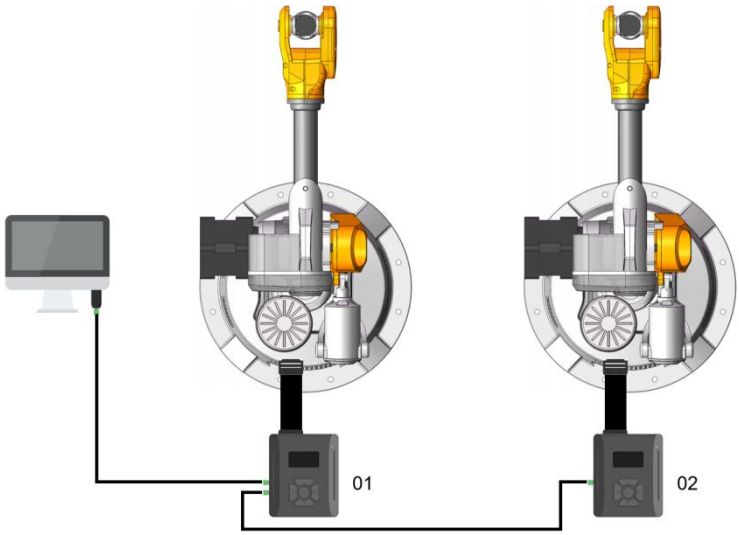


多功能控制器 RS485 设置方法请参照：[多功能控制器用户手册]

UART 接线示意图：



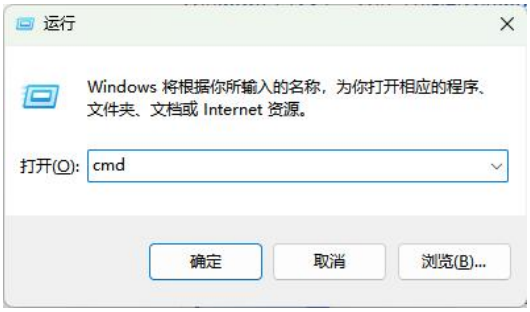
RS485 连接双机械臂接线示意图：



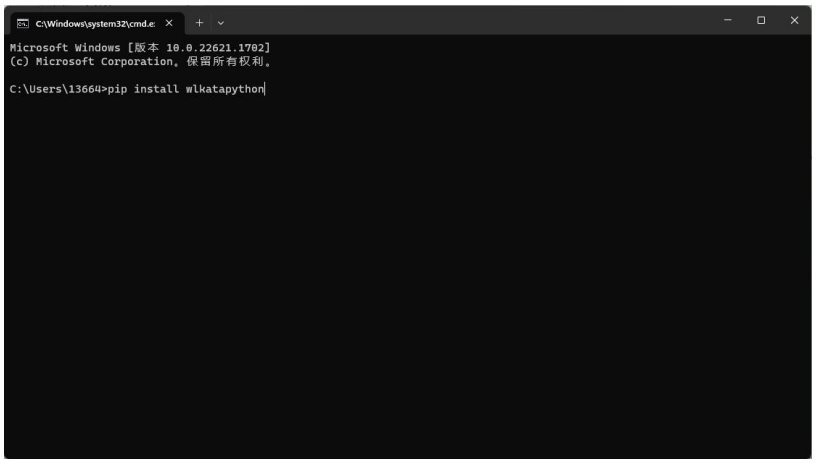
1.2. PIP 安装

- wlkatapython **模块安装**

在 Windows 下使用 win+r 键打开运行窗口，输入 cmd 后点击确认。



在弹出的命令提示符窗口中输入：pip install wlkatapython



点击回车即可自动安装（需电脑联网）。

2. Python SDK 控制函数介绍

2.1. 调用库

函数	import wlkatapython
函数说明	导入 WLKATA 库文件。

参数说明	无
返回值	无
备注	无

2.2. 创建机械臂对象(Mirobot)

函数	Mirobot_UART(block_flag=True,message_flag=True);
函数说明	创建机械臂对象。
参数说明	block_flag:设置成 True，则除透传指令外，其它指令入阻塞模式,反之则进入非阻塞模式。 message_flag：设置成 True,则在终端打印当前发送给设备的信息，以及读取到的设备信息，反之则关闭信息打印。 默认 block_flag 和 message_flag 是关闭状态。

2.3. 设置机械臂地址

函数	init(p, adr);
函数说明	将串口和多功能控制器地址进行绑定。
参数说明	P:串口号; adr:多功能控制器地址。

2.4. 多功能控制器重启

函数	restart();

函数说明	控制多功能控制器重启。
参数说明	无
示例（RS485）	import wlkatapython import serial '''多功能控制器重启''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 mirobot1.restart()#多功能控制器重启 serial1.close()#关闭串口
示例(UART)	import wlkatapython import serial '''多功能控制器重启''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 mirobot1.restart()#多功能控制器重启 serial1.close()#关闭串口

2.5. 回零

函数	homing();
函数说明	机械臂回零操作。

参数说明	无
示例(RS485)	import wlkatapython import serial '''机械臂回零''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)# 新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 mirobot1.homing()#机械臂回零 serial1.close()#关闭串口
示例（UART)	import wlkatapython import serial '''机械臂回零''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)# 新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 mirobot1.homing()#机械臂回零 serial1.close()#关闭串口

2.6. 机械臂回到初始位置

函数	zero();
函数说明	控制机械臂回到初始位置。
参数说明	无

示例(RS485)	import wlkatapython import serial '''机械臂回到初始位置''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)# 新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 mirobot1.zero()#机械臂回到初始位置 serial1.close()#关闭串口
示例（UART)	import wlkatapython import serial '''机械臂回到初始位置''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)# 新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 mirobot1.zero()#机械臂回到初始位置 serial1.close()#关闭串口

2.7. 执行离线文件

函数	runFile(fileName, num=False);
函数说明	执行多功能控制器中的离线文件;
参数说明	fileName,：离线文件名称; num：是否循环执行，False 为只执行 1 次，True 为循环执行该离线文件。该参

	数为选填，不填写默认为 False。
示例(RS485)	import wlkatapython import serial '''执行离线文件''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 mirobot1.runFile("ceshi", False)#循环执行某一文件，True 为循环执行文件，False 为单次执行文件 serial1.close()#关闭串口
示例（UART)	import wlkatapython import serial '''执行离线文件''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 mirobot1.runFile("ceshi", False)#循环执行某一文件，True 为循环执行文件，False 为单次执行文件 serial1.close()#关闭串口

2.8. 机械臂停止

函数	cancellation();
函数说明	机械臂以减速形式停止当前动作，并清理掉缓存动作。

参数说明	无
示例（RS485）	import wlkatapython import serial '''停止当前程序''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 mirobot1.runFile("ceshi",True)#循环执行某一文件，True 为循环执行文件，False 为单次执行文件 time.sleep(5)#等待 5 秒钟 mirobot1.cancellation()#停止当前运行的程序 serial1.close()#关闭串口
示例（UART）	import wlkatapython import serial '''停止当前程序''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 mirobot1.runFile("ceshi",True)#循环执行某一文件，True 为循环执行文件，False 为单次执行文件 time.sleep(5)#等待 5 秒钟 mirobot1.cancellation()#停止当前运行的程序 serial1.close()#关闭串口

2.9. 气泵控制

函数	pump(num);
函数说明	控制气泵吸气、吹气、关闭。
参数说明	num:0—关闭; 1—吸气; 2—吹气。

示例（RS485）	import wlkatapython
	import serial
	import time
	'''控制气泵'''
	serial1 = serial.Serial("COM3", 38400)#设置串口及波特率
	mirobot1
	=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#
	新建 mirobot1 对象
	mirobot1.init(serial1, 1)#设置机械臂地址
	for i in range(0,3):#气泵吸气吹气
	mirobot1.pump(1)#气泵吸气
	time.sleep(3)
	mirobot1.pump(2)#气泵吹气
	time.sleep(3)
	mirobot1.pump(0)#气泵关闭
	serial1.close()#关闭串口

示例（UART）	import wlkatapython import serial import time '''控制气泵'''

	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	mirobot1
	=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#
	新建 mirobot1 对象
	mirobot1.init(serial1, -1)#设置机械臂地址
	for i in range(0,3):#气泵吸气吹气
	mirobot1.pump(1)#气泵吸气
	time.sleep(3)
	mirobot1.pump(2)#气泵吹气
	time.sleep(3)
	mirobot1.pump(0)#气泵关闭
	serial1.close()#关闭串口

2.10. 夹爪控制

函数	gripper(num);
函数说明	控制夹爪张开以及闭合。
参数说明	num:夹爪状态——0 夹爪断电，1 夹爪打开，2 夹爪闭合。
示例（RS485）	import wlkatapython import serial import time '''夹爪控制''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)# 新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 for i in range(0,3):#电动夹爪

	mirobot1.gripper(0)#夹爪断电
	time.sleep(1)
	mirobot1.gripper(1)#夹爪打开
	time.sleep(1)
	mirobot1.gripper(2)#夹爪闭合
	time.sleep(1)
	serial1.close()#关闭串口

示例（UART)	import wlkatapython
	import serial
	import time
	'''夹爪控制'''
	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	mirobot1
	=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象
	mirobot1.init(serial1, -1)#设置机械臂地址
	for i in range(0,3):#电动夹爪
	mirobot1.gripper(0)#夹爪断电
	time.sleep(1)
	mirobot1.gripper(1)#夹爪打开
	time.sleep(1)
	mirobot1.gripper(2)#夹爪闭合
	time.sleep(1)
	serial1.close()#关闭串口

2.11. PWM 控制

函数	pwmWrite(num);

函数说明	向 PWM 口输出信号。
参数说明	num:PWM 输出值。

2.12. 透传指令

函数	sendMsg(string);
函数说明	向机械臂传输指令。
参数说明	String：需要传输的指令。

2.13. 固件版本查询

函数	version();
函数说明	查询机械臂及多功能控制器固件版本。(当前只支持 UART 查询，RS485 暂不支持)
参数说明	无
返回值	机械臂及多功能控制器版本号。

示例（UART)	`import wlkatapython`
	`import serial`
	`'''固件版本查询'''`
	`serial1 = serial.Serial("COM3", 115200)#设置串口及波特率`
	`mirobot1 =wlkatapython.Mirobot_UART()#新建 mirobot1 对象`
	`mirobot1.init(serial1, 1)#设置机械臂地址`
	`print(mirobot1.version())#查询机械臂及多功能控制器信息`
	`serial1.close()#关闭串口`

2.14. 获取机械臂全部状态

函数	getStatus();
函数说明	查询机械臂状态。
参数说明	无
返回值	以字典的形式返回机械臂运动状态、角度模式下各轴角度、坐标模式下的坐标及末端工具状态（气泵+夹爪）、及当前运动模式。

示例（RS485）	`import wlkatapython`
	`import serial`
	`'''获取机械臂全部状态'''`
	`serial1 = serial.Serial("COM3", 38400)#设置串口及波特率`
	`mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象`
	`mirobot1.init(serial1, 1)#设置机械臂地址`
	`print(mirobot1.getStatus())#获取机械臂全部信息`
	`serial1.close()#关闭串口`

示例（UART）	`import wlkatapython`
	`import serial`
	`'''获取机械臂全部状态'''`
	`serial1 = serial.Serial("COM3", 115200)#设置串口及波特率`
	`mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象`
	`mirobot1.init(serial1, -1)#设置机械臂地址`
	`print(mirobot1.getStatus())#获取机械臂全部信息`
	`serial1.close()#关闭串口`

2.15. 获取机械臂运动状态

函数	getState();
函数说明	查询机械臂状态。
参数说明	无
返回值	返回机械臂运动状态。

示例（RS485）	`import wlkatapython`
	`import serial`
	`'''获取机械臂状态'''`
	`serial1 = serial.Serial("COM3", 38400)#设置串口及波特率`
	`mirobot1`
	`=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象`
	`mirobot1.init(serial1, 1)#设置机械臂地址`
	`print(mirobot1.getState())#获取机械臂状态`
	`serial1.close()#关闭串口`

示例（UART）	`import wlkatapython`
	`import serial`
	`'''获取机械臂状态'''`
	`serial1 = serial.Serial("COM3", 115200)#设置串口及波特率`
	`mirobot1`
	`=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象`
	`mirobot1.init(serial1, -1)#设置机械臂地址`
	`print(mirobot1.getState())#获取机械臂状态`
	`serial1.close()#关闭串口`

2.16. 获取机械臂各轴角度

函数	getAngle(num);
函数说明	获取机械臂各轴角度。
参数说明	Num:1/2/3/4/5/6/7，表示 1-6 轴的角度，7 表示传送带的角度。
返回值	返回机械臂各轴角度及传送带的角度。
示例（RS485）	import wlkatapython import serial '''获取机械臂角度''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 for i in range(1,8): print(mirobot1.getAngle(i))#获取机械臂角度,1 到 6 对应 1 到 6 轴角度，7 对应传送带 serial1.close()#关闭串口
示例（UART）	import wlkatapython import serial '''获取机械臂角度''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 for i in range(1,8):

	print(mirobot1.getAngle(i))#获取机械臂角度,1 到 6 对应 1 到 6 轴角度，7 对应传送带 serial1.close()#关闭串口

2.17. 获取机械臂坐标

函数	getcoordinate(num);
函数说明	获取机械臂坐标。
参数说明	Num:1/2/3/4/5/6，分别表示机械臂 X//Y/Z/RX/RY/RZ 坐标。
返回值	返回机械臂 X//Y/Z/RX/RY/RZ 坐标。
示例(RS485)	import wlkatapython import serial '''获取机械臂坐标''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, 1)#设置机械臂地址 for i in range(1,7): print(mirobot1.getcoordinate(i))#获取机械臂坐标，1 到 6 对应 x/y/z/rx/ry/rz 坐标 serial1.close()#关闭串口
示例(UART)	import wlkatapython import serial '''获取机械臂坐标''' serial1 = serial.Serial("COM3", 115200)#设置串口及波特率 mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#

	新建 mirobot1 对象 mirobot1.init(serial1, -1)*#设置机械臂地址* for i in range(1,7): print(mirobot1.getcoordinate(i))*#获取机械臂坐标，1 到 6 对应 x/y/z/rx/ry/rz 坐标* serial1.close()*#关闭串口*

2.18. 获取机械臂运动模式

函数	getmooe();
函数说明	获取机械臂运动模式。
参数说明	无
返回值	0 为角度模式，1 为坐标模式。
示例(RS485)	import wlkatapython import serial '''运动模式查询''' serial1 = serial.Serial("COM3", 38400)*#设置串口及波特率* mirobot1 =wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)*#新建 mirobot1 对象* mirobot1.init(serial1, 1)*#设置机械臂地址* print(mirobot1.getmooe())*#运动模式查询* serial1.close()*#关闭串口*
示例(UART)	import wlkatapython import serial '''运动模式查询''' serial1 = serial.Serial("COM3", 115200)*#设置串口及波特率* mirobot1

	=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新建 mirobot1 对象 mirobot1.init(serial1, -1)#设置机械臂地址 print(mirobot1.getmooe())#运动模式查询 serial1.close()#关闭串口

2.19. 笛卡尔坐标控制机械臂(Mirobot)

函数	writecoordinate(motion,position,x,y,z,a,b,c);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Motion:0—快速运动; 1—直线运动; 2—门型运动。 Position：0—绝对值运动; 1—增量值运动。 x：x 点坐标; y：y 点坐标; z：z 点坐标; a：a 点坐标; b：b 点坐标; c：c 点坐标;
返回值	无
示例（RS485）	import wlkatapython import serial

```
import time

        '''机械臂使用三指柔爪抓取木块'''

coordinate_g=[[198.6 ,0 ,230 ,0 ,0 ,0],
              [198.6 ,0 ,90.7 ,0 ,0 ,0],
              [198.6 ,0 ,70.7 ,0 ,0 ,0],
              [198.6 ,0 ,90.7 ,0 ,0 ,0],
              [152.1 ,-127.6 ,90.4 ,0 ,0 ,-39.9],
              [152.1 ,-127.6 ,70.4 ,0 ,0 ,-39.9],
              [152.1 ,-127.6 ,90.4 ,0 ,0 ,-39.9],
 ]

serial1 = serial.Serial("COM4", 38400)#设置串口及波特率
mirobot1
=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新
建 mirobot1 对象
mirobot1.init(serial1, 1)#设置机械臂地址
if mirobot1.getState()=="Alarm":#如果机械臂处于 Alarm 状态则进行回
零
    mirobot1.homing()
for i in range(0,7):
    while mirobot1.getState()!="Idle":
        print("机械臂 1 状态：",mirobot1.getState())

mirobot1.writecoordinate(0,0,coordinate_g[i][0],coordinate_g[i][1
],coordinate_g[i][2],coordinate_g[i][3],coordinate_g[i][4],coordi
nate_g[i][5])#快速/直线/门型，绝对/相对，x，y，z，a，b，c，
    if i==1:
```

```
            mirobot1.pump(1)  # 气泵吸气
            time.sleep(3)
        elif i==2:
            mirobot1.pump(2)  # 气泵吹气
            time.sleep(3)
        elif i==5:
            mirobot1.pump(1)  # 气泵吸气
            time.sleep(3)
mirobot1.pump(0)  # 气泵关闭
mirobot1.zero()#机械臂回到初始位置
serial1.close()#关闭串口
```

示例（UART）

```
import wlkatapython
import serial
import time

        '''机械臂使用三指柔爪抓取木块'''

coordinate_g=[[198.6 ,0 ,230 ,0 ,0 ,0],
              [198.6 ,0 ,90.7 ,0 ,0 ,0],
              [198.6 ,0 ,70.7 ,0 ,0 ,0],
              [198.6 ,0 ,90.7 ,0 ,0 ,0],
              [152.1 ,-127.6 ,90.4 ,0 ,0 ,-39.9],
              [152.1 ,-127.6 ,70.4 ,0 ,0 ,-39.9],
              [152.1 ,-127.6 ,90.4 ,0 ,0 ,-39.9],
 ]

serial1 = serial.Serial("COM4", 115200)#设置串口及波特率
mirobot1
=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新
```

```
建 mirobot1 对象
mirobot1.init(serial1, -1)#设置机械臂地址
if mirobot1.getState()=="Alarm":#如果机械臂处于 Alarm 状态则进行回
零
    mirobot1.homing()
for i in range(0,7):
    while mirobot1.getState()!="Idle":
        print("机械臂 1 状态：",mirobot1.getState())

mirobot1.writecoordinate(0,0,coordinate_g[i][0],coordinate_g[i][1
],coordinate_g[i][2],coordinate_g[i][3],coordinate_g[i][4],coordi
nate_g[i][5])#快速/直线/门型，绝对/相对，x，y，z，a，b，c，
    if i==1:
        mirobot1.pump(1)  # 气泵吸气
        time.sleep(3)
    elif i==2:
        mirobot1.pump(2)  # 气泵吹气
        time.sleep(3)
    elif i==5:
        mirobot1.pump(1)  # 气泵吸气
        time.sleep(3)
mirobot1.pump(0)  # 气泵关闭
mirobot1.zero()#机械臂回到初始位置
serial1.close()#关闭串口
```

2.20. 角度控制机械臂(Mirobot)

函数	writeangle(position,x,y,z,a,b,c);

函数说明	笛卡尔坐标控制机械臂。
参数说明	Position：0—绝对值运动； 1—增量值运动。 x：1 轴角度； y：2 轴角度； z：3 轴角度； a：4 轴角度； b：5 轴角度； c：6 轴角度；
返回值	无

示例(RS485)	`import wlkatapython`
	`import serial`
	
	`coordinate_g=[[30 ,0 ,0 ,0 ,0 ,0],`
	`[-30 ,0 ,0 ,0 ,0 ,0],`
	
	`[0 ,30 ,0 ,0 ,0 ,0],`
	`[0 ,-30 ,0 ,0 ,0 ,0],`
	
	`[0, 0, 30, 0, 0, 0],`
	`[0 ,0 ,-30 ,0 ,0 ,0],`
	
	`[0, 0, 0, 30, 0, 0],`
	`[0, 0, 0, -30, 0, 0],`
	
	`[0 ,0 ,0 ,0 ,30 ,0],`

```
                [0 ,0 ,0 ,0 ,-30 ,0],


                [0 ,0 ,0 ,0 ,0 ,30],
                [0 ,0 ,0 ,0 ,0 ,-30],


 ]
serial1 = serial.Serial("COM4", 38400)#设置串口及波特率
mirobot1
=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新
建 mirobot1 对象
mirobot1.init(serial1, 1)#设置机械臂地址
if mirobot1.getState()=="Alarm":#如果机械臂处于 Alarm 状态则进行回
零
    mirobot1.homing()
for i in range(0,12):
    while mirobot1.getState()!="Idle":
        print("机械臂 1 状态：",mirobot1.getState())
mirobot1.writeangle(0,coordinate_g[i][0],coordinate_g[i][1],coord
inate_g[i][2],coordinate_g[i][3],coordinate_g[i][4],coordinate_g[
i][5])#快速/直线/门型，绝对/相对，x，y，z，a，b，c，
mirobot1.zero()#机械臂回到初始位置
serial1.close()#关闭串口
```

示例(UART)

```
import wlkatapython
import serial


coordinate_g=[[30 ,0 ,0 ,0 ,0 ,0],
                [-30 ,0 ,0 ,0 ,0 ,0],


                [0 ,30 ,0 ,0 ,0 ,0],
```

```
                [0 ,-30 ,0 ,0 ,0 ,0],

                [0, 0, 30, 0, 0, 0],
                [0 ,0 ,-30 ,0 ,0 ,0],

                [0, 0, 0, 30, 0, 0],
                [0, 0, 0, -30, 0, 0],

                [0 ,0 ,0 ,0 ,30 ,0],
                [0 ,0 ,0 ,0 ,-30 ,0],

                [0 ,0 ,0 ,0 ,0 ,30],
                [0 ,0 ,0 ,0 ,0 ,-30],


 ]
serial1 = serial.Serial("COM4", 115200)#设置串口及波特率
mirobot1
=wlkatapython.Mirobot_UART(block_flag=True,message_flag=True)#新
建 mirobot1 对象
mirobot1.init(serial1, -1)#设置机械臂地址
if mirobot1.getState()=="Alarm":#如果机械臂处于 Alarm 状态则进行回
零
    mirobot1.homing()
for i in range(0,12):
    while mirobot1.getState()!="Idle":
        print("机械臂 1 状态：",mirobot1.getState())
mirobot1.writeangle(0,coordinate_g[i][0],coordinate_g[i][1],coord
inate_g[i][2],coordinate_g[i][3],coordinate_g[i][4],coordinate_g[
i][5])#快速/直线/门型，绝对/相对，x，y，z，a，b，c，
```

	mirobot1.zero()#机械臂回到初始位置 serial1.close()#关闭串口

2.21. 机械臂速度

函数	speed(num);
函数说明	机械臂速度设置。
参数说明	num：机械臂速度，取值范围 0~2000。
返回值	无

2.22. 机械臂拓展轴控制

函数	writeexpand(motion, position, d);
函数说明	控制机械臂拓展轴运动。
参数说明	Motion:0—快速运动； 1—直线运动。 Position：0—绝对值运动； 1—增量值运动。 d:移动多少距离。
返回值	无

2.23. 创建机械臂对象(E4)

函数	E4_UART(block_flag=True,message_flag=True);

函数说明	创建机械臂对象。
参数说明	block_flag:设置成 True，则除透传指令外，其它指令入阻塞模式,反之则进入非阻塞模式。 message_flag：设置成 True,则在终端打印当前发送给设备的信息，以及读取到的设备信息，反之则关闭信息打印。 默认 block_flag 和 message_flag 是关闭状态。

2.24. 笛卡尔坐标控制机械臂(E4)

函数	writecoordinate(motion,position,x,y,z,a);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Motion:0—快速运动； 1—直线运动； 2—门型运动。 Position：0—绝对值运动； 1—增量值运动。 x：x 点坐标； y：y 点坐标； z：z 点坐标； a：a 点坐标；
返回值	无
示例(RS485)	`import wlkatapython`

	import serial
	import time
	'''E4 机械臂笛卡尔坐标运动'''
	serial1 = serial.Serial("COM3", 38400)#设置串口及波特率
	E4_1 =wlkatapython.E4_UART(block_flag=True,message_flag=True)#新建 E4_1 对象
	E4_1.init(serial1, 1)#设置机械臂地址
	E4_1.restart()
	time.sleep(3)
	E4_1.homing()
	E4_1.writecoordinate(0,1,20,20,20)
	E4_1.writecoordinate(0,1,-20,-20,-20)
	serial1.close()#关闭串口

示例(UART)	import wlkatapython
	import serial
	import time
	'''E4 机械臂笛卡尔坐标运动'''
	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	E4_1 =wlkatapython.E4_UART(block_flag=True,message_flag=True)#新建 E4_1 对象
	E4_1.init(serial1, -1)#设置机械臂地址
	E4_1.restart()
	time.sleep(3)
	E4_1.homing()
	E4_1.writecoordinate(0,1,20,20,20)
	E4_1.writecoordinate(0,1,-20,-20,-20)
	serial1.close()#关闭串口

2.25. 角度控制机械臂(E4)

函数	writeangle(position,x,y,z,a);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Position：0—绝对值运动； 1—增量值运动。 x：1 轴角度； y：2 轴角度； z：3 轴角度； a：4 轴角度；
返回值	无

示例(RS485)	import wlkatapython
	import serial
	import time
	
	'''E4 机械臂角度坐标运动'''
	serial1 = serial.Serial("COM3", 38400)#设置串口及波特率
	E4_1 =wlkatapython.E4_UART(block_flag=True,message_flag=True)#新建 E4_1 对象
	E4_1.init(serial1, 1)#设置机械臂地址
	E4_1.restart()
	time.sleep(3)
	E4_1.homing()
	E4_1.writeangle(1,20,20,20)
	E4_1.writeangle(1,-20,-20,-20)
	serial1.close()#关闭串口

示例(UART)	

```
import wlkatapython
import serial
import time


'''E4 机械臂角度坐标运动'''
serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
E4_1 =wlkatapython.E4_UART(block_flag=True,message_flag=True)#新建
E4_1 对象
E4_1.init(serial1, -1)#设置机械臂地址
E4_1.restart()
time.sleep(3)
E4_1.homing()
E4_1.writeangle(1,20,20,20)
E4_1.writeangle(1,-20,-20,-20)
serial1.close()#关闭串口
```

2.26. 创建机械臂对象(MT4)

函数	MT4_UART(block_flag=True,message_flag=True);
函数说明	创建机械臂对象。
参数说明	block_flag:设置成 True，则除透传指令外，其它指令入阻塞模式,反之则进入非阻塞模式。 message_flag：设置成 True,则在终端打印当前发送给设备的信息，以及读取到的设备信息，反之则关闭信息打印。 默认 block_flag 和 message_flag 是关闭状态。

2.27. 笛卡尔坐标控制机械臂(MT4)

函数	writecoordinate(motion,position,x,y,z,a);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Motion:0—快速运动; 1—直线运动; 2—门型运动。 Position：0—绝对值运动; 1—增量值运动。 x：x 点坐标; y：y 点坐标; z：z 点坐标; a：a 点坐标;
返回值	无

示例(RS485)	import wlkatapython
	import serial
	import time
	'''MT4 机械臂笛卡尔坐标运动'''
	serial1 = serial.Serial("COM3", 38400)#设置串口及波特率
	MT4_1 =wlkatapython.MT4_UART(block_flag=True,message_flag=True)#新建 MT4_1 对象
	MT4_1.init(serial1, 1)#设置机械臂地址
	MT4_1.restart()
	time.sleep(3)
	MT4_1.homing()

	MT4_1.writecoordinate(0,1,20,20,20) MT4_1.writecoordinate(0,1,-20,-20,-20) serial1.close()#关闭串口

示例(UART)	import wlkatapython
	import serial
	import time
	'''MT4 机械臂笛卡尔坐标运动'''
	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	MT4_1 =wlkatapython.MT4_UART(block_flag=True,message_flag=True)#新建 MT4_1 对象
	MT4_1.init(serial1, -1)#设置机械臂地址
	MT4_1.restart()
	time.sleep(3)
	MT4_1.homing()
	MT4_1.writecoordinate(0,1,20,20,20)
	MT4_1.writecoordinate(0,1,-20,-20,-20)
	serial1.close()#关闭串口

2.28. 角度控制机械臂(MT4)

函数	writeangle(position,x,y,z,a);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Position：0—绝对值运动； 1—增量值运动。 x：1 轴角度； y：2 轴角度； z：3 轴角度；

	a：4 轴角度；
返回值	无

示例(RS485)	import wlkatapython
	import serial
	import time
	
	'''MT4 机械臂角度坐标运动'''
	serial1 = serial.Serial("COM3", 38400)#设置串口及波特率
	MT4_1 =wlkatapython.MT4_UART(block_flag=True,message_flag=True)#新建 MT4_1 对象
	MT4_1.init(serial1, 1)#设置机械臂地址
	MT4_1.restart()
	time.sleep(3)
	MT4_1.homing()
	MT4_1.writeangle(1,20,20,20)
	MT4_1.writeangle(1,-20,-20,-20)
	serial1.close()#关闭串口

示例(UART)	import wlkatapython
	import serial
	import time
	
	'''MT4 机械臂角度坐标运动'''
	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	MT4_1 =wlkatapython.MT4_UART(block_flag=True,message_flag=True)#新建 MT4_1 对象
	MT4_1.init(serial1, -1)#设置机械臂地址
	MT4_1.restart()
	time.sleep(3)

	MT4_1.homing() MT4_1.writeangle(1,20,20,20) MT4_1.writeangle(1,-20,-20,-20) serial1.close()#关闭串口

2.29. 创建步进电机控制器对象（MS4220）

函数	MS4220_UART(block_flag=True,message_flag=True);
函数说明	创建步进电机控制器对象。
参数说明	无

2.30. 设置步进电机控制器地址（MS4220）

函数	init(p, adr);
函数说明	将串口和步进电机控制器地址进行绑定。
参数说明	P:串口号; adr:步进电机控制器地址。

2.31. 步进电机控制器透传指令（MS4220）

函数	sendMsg(string);
函数说明	向步进电机控制器传输指令。
参数说明	String：需要传输的指令。

2.32. 步进电机控制器重启（MS4220）

函数	restart();
函数说明	步进电机控制器重启。
参数说明	无

2.33. 步进电机控制器回零（MS4220）

函数	homing();
函数说明	步进电机控制器回零。（此功能需要在步进电机控制器安装回零传感器）
参数说明	无

2.34. 步进电机控制器距离移动（MS4220）

函数	move_position(position, num,speed=50);
函数说明	控制步进电机以指定速度移动指定距离。一般安转了回零传感器才会使用这个指令。
参数说明	Position：需要移动的距离，取值范围根据步进电机安装到的设备不同而改变，如安装到传送带和安装到出料机上取值范围是不同的，且这里的数值是步进电机的脉冲数。

2.35. 步进电机控制器速度移动（MS4220）

函数	move_speed(num);

函数说明	步进电机以指定的速度移动。
参数说明	num：步进电机运动及速度设置，取值范围-100~100。负值为反方向转动，正值为正方向转动，数值应为整数。
返回值	无

示例(RS485)	`import wlkatapython`
	`import serial`
	`import time`
	`'''MS4220 步进电机控制器运动'''`
	`serial1 = serial.Serial("COM3", 38400)#设置串口及波特率`
	`MS4220_1 =wlkatapython.MS4220_UART(block_flag=True,message_flag=True)# 新建 MS4220_1 对象`
	`MS4220_1.init(serial1, 10)#设置传动带地址`
	`for i in range(10):`
	`  ms4220.speed(100)`
	`  time.sleep(2)`
	`  ms4220.speed(-100)`
	`  time.sleep(2)`
	`  ms4220.speed(0)`
	`  time.sleep(2)`
	`serial1.close()#关闭串口`

2.36. 步进电机状态全部获取（MS4220）

函数	getStatus();
函数说明	获取步进电机控制器的全部状态。
参数说明	无
返回值	<" State" :" Idle" ," WPos" :0," S0" :0," S1" :0," S2" : 0>
示例(RS485)	State 后面跟的是步进电机控制的状态，有 Idle,Run 等状态。

	WPos 后面接的的当前绝对位置。 S0,S1,S2 后面跟的是 3 个传感器的状态。

2.37. 创建 Mirobot_Serial_GUI 对象

函数	Mirobot_Serial_GUI();
函数说明	创建 Mirobot_Serial_GUI 对象
参数说明	无
返回值	无

2.38. 调取 Mirobot_GUI

函数	Mirobot_Serial_GUI();
函数说明	创建 Mirobot_Serial_GUI 对象
参数说明	无
返回值	无
示例	import wlkatapython Mirobot_GUI=wlkatapython.Mirobot_Serial_GUI() Mirobot_GUI.Mirobot_GUI()

2.39. 创建机械臂对象(Haro380)

函数	Harobot_UART(block_flag=True,message_flag=True);
函数说明	创建机械臂对象。
参数说明	block_flag:设置成 True，则除透传指令外，其它指令入阻塞模式,反之则进入非阻塞模式。 message_flag：设置成 True,则在终端打印当前发送给设备的信息，以及读取到的设备信息，反之则关闭信息打印。 默认 block_flag 和 message_flag 是关闭状态。

2.40. 笛卡尔坐标控制机械臂(Haro380)

函数	writecoordinate(motion,position,x,y,z,a,b,c);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Motion:0—快速运动； 1—直线运动； 2—门型运动。 Position：0—绝对值运动； 1—增量值运动。 x：x 点坐标； y：y 点坐标； z：z 点坐标；

	a：rx 点坐标； b：ry 点坐标； c：rz 点坐标；
返回值	无

示例(RS485)	import wlkatapython
	import serial
	import time
	'''harobot1 机械臂笛卡尔坐标运动'''
	serial1 = serial.Serial("COM3", 38400)#设置串口及波特率
	harobot1=wlkatapython.Harobot_UART(block_flag=True,message_flag=True)#新建 E4_1 对象
	harobot1.init(serial1, 1)#设置机械臂地址
	harobot1.restart()
	time.sleep(3)
	harobot1.homing()
	harobot1.writecoordinate(0,1,20,20,20,0,0,0)
	harobot1.writecoordinate(0,1,-20,-20,-20,0,0,0)
	serial1.close()#关闭串口

示例(UART)	import wlkatapython
	import serial
	import time
	'''E4 机械臂笛卡尔坐标运动'''
	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	harobot1
	=wlkatapython.Harobot_UART(block_flag=True,message_flag=True)#新建 E4_1 对象
	harobot1.init(serial1, -1)#设置机械臂地址
	harobot1.restart()

	time.sleep(3) harobot1.homing() harobot1.writecoordinate(0,1,20,20,20,0,0,0) harobot1.writecoordinate(0,1,-20,-20,-20,0,0,0) serial1.close()#关闭串口

2.41. 角度控制机械臂(Haro380)

函数	writeangle(position,x,y,z,a);
函数说明	笛卡尔坐标控制机械臂。
参数说明	Position：0—绝对值运动； 1—增量值运动。 x：1 轴角度； y：2 轴角度； z：3 轴角度； a：4 轴角度；
返回值	无
示例(RS485)	import wlkatapython import serial import time '''E4 机械臂角度坐标运动''' serial1 = serial.Serial("COM3", 38400)#设置串口及波特率 harobot1 =wlkatapython.Harobot_UART(block_flag=True,message_flag=True)#新建 E4_1 对象

	harobot1.init(serial1, 1)#设置机械臂地址
	harobot1.restart()
	time.sleep(3)
	harobot1.homing()
	harobot1.writeangle(1,20,20,20,0,0,0)
	harobot1.writeangle(1,-20,-20,-20,0,0,0)
	serial1.close()#关闭串口

示例(UART)	import wlkatapython
	import serial
	import time
	
	'''E4 机械臂角度坐标运动'''
	serial1 = serial.Serial("COM3", 115200)#设置串口及波特率
	harobot1
	=wlkatapython.Harobot_UART(block_flag=True,message_flag=True)#新建 E4_1 对象
	harobot1.init(serial1, -1)#设置机械臂地址
	harobot1.restart()
	time.sleep(3)
	harobot1.homing()
	harobot1.writeangle(1,20,20,20,0,0,0)
	harobot1.writeangle(1,-20,-20,-20,0,0,0)
	serial1.close()#关闭串口

版权所有© 勤牛创智科技有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部， 并不得以任何形式传播。

免责申明

在法律允许的最大范围内，本手册所描述的产品（含其硬件、软件、固件等）均“按照现状”提供，可能存在瑕疵、错误或故障，勤牛创智不提供任何形式的明示或默示保证，亦不对使用本手册或使用本公司产品导致的任何特殊、偶然或间接的损害进行赔偿。在使用本产品前详细阅读本使用手册及网上发布的相关技术文档并了解相关信息，确保在充分了解产品相关知识的前提下使用本产品。

本产品的使用者有责任确保遵循相关国家的切实可行的法律法规，确保在勤牛创智机械臂的使用中不存在任何重大危险。

版本修订说明

时间	版本号	修订记录
2023.6.21	V1.000 版	初稿
2023.12.05	V1.007 版	增加 E4 和传送带控制
2024.04.16	V1.009 版	修改舵机夹爪控制方式，增加 Mirobot 机械臂调试 GUI 插件，该插件可使用串口或 RS485 调试机械臂笛卡尔坐标，并保存成 TXT 或 Gcode 文件。
2024.05.08	V1.010 版	增加 RS485 接线示意图
2025.01.02	V1.011 版	增加 MT4 控制
2025.10.22	V1.0.2	增加函数阻塞，增加 Haro380 控制

北京勤牛创智科技有限公司

网址：cn.wlkata.com

官方公众号