



# UFACTORY XARM

## 直线滑轨用户手册



使用前请仔细阅读本手册

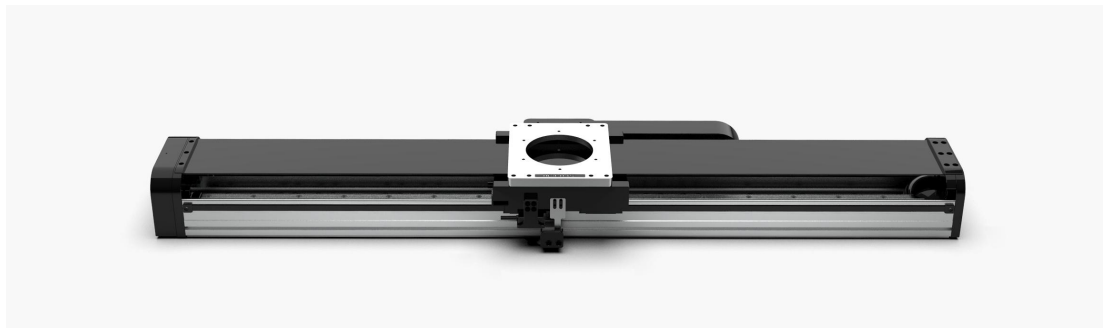
V 2.3.0

## 目录

1. 总体介绍 .....	3
1.1. UFactory 直线滑轨简要介绍 .....	3
1.2. 直线滑轨型号 .....	3
1.3. AC Control Box Pro 控制器 .....	4
1.4. 安全 .....	4
2. 安装 .....	7
2.1. 发货清单 .....	8
2.2. 机械安装 .....	8
3. 直线滑轨的控制方式 .....	11
3.1. 用 xArm Studio 控制直线滑轨 .....	11
3.2. 用 Python-SDK 控制直线滑轨 .....	13
3.3. 指示灯 .....	13
3.4. 直线滑轨 IO .....	14
4. 报警与处理 .....	15
5. UFactory 直线滑轨技术规格 .....	17
6. 售后服务 .....	18
附录 .....	19
附录 1-用 Modbus RTU 通讯协议控制直线滑轨 .....	19
1.1 简要介绍 .....	19
1.2 读取直线滑轨寄存器 .....	19
1.3 写入直线滑轨寄存器 .....	20
1.4 Modbus RTU 示例 .....	20

# 1. 总体介绍

## 1.1. UFactory 直线滑轨简要介绍

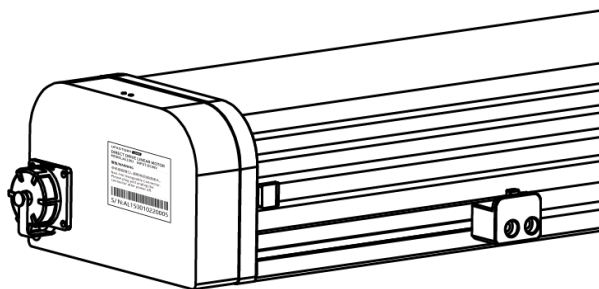


UFactory 直线滑轨

UFactory 直线滑轨需要配合 AC Control Box Pro 控制器使用，能够支持和引导运动部件，按给定的方向平稳地做直线运动，一定程度上增大机械臂的工作范围。

## 1.2. 直线滑轨型号

UFactory 直线滑轨有两个型号，可根据 SN 划分。SN 位于直线滑轨端板处，见下图。



直线滑轨原点：位置-0。

直线滑轨的速度范围是：1 到 1000 (mm/s)。

直线滑轨的位置数值范围：

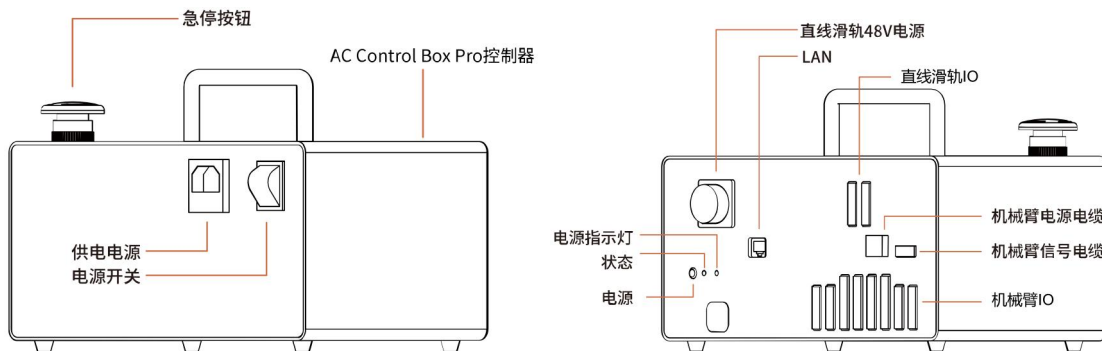
AL0700：0 到 700 (mm)

AL1000：0 到 1000 (mm)

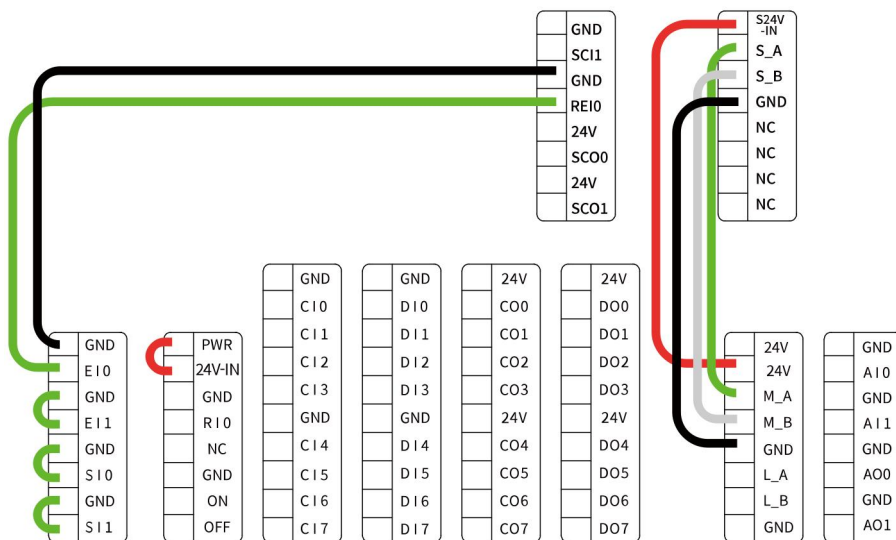
AL1500：0 到 1500 (mm)

### 1.3. AC Control Box Pro 控制器

UFactory 直线滑轨必须配合 AC Control Box 控制器使用。



控制器 IO 连接如下图，出厂时已连接好。



### 1.4. 安全

操作员在使用 UFactory 直线滑轨之前必须已阅读并理解手册中的所有说明。

#### 1.4.1. 警告

在操作机器人之前，必须正确安装好直线滑轨。

请勿安装或操作已损坏或缺少零件的直线滑轨。

切勿为直线滑轨通交流电。

确保所有接线端子稳定连接在机械臂和直线滑轨两端。

请始终使用建议的安装方式。

在初始化直线滑轨之前，请确保没有杂物在机械臂和直线滑轨路径中。

根据您的应用情况，相应设置直线滑轨的速度和位置。

## 注意

术语“操作员”是指负责在 UFactory 直线滑轨上进行以下任何操作的任何人：

- 安装
- 控制
- 维护
- 检查
- 编程
- 退役

本文档说明了 UFactory 直线滑轨从安装到运行再到使用的整个生命周期的一般操作。

本文档中的图形和照片是代表性的示例，它们与交付的产品之间可能存在差异。

### 1.4.2. 风险评估和最终应用

UFactory 直线滑轨用于工业机器人，最终应用中使用的机器人、直线滑轨和任何其他设备必须进行风险评估。机器人集成商的责任是确保遵守所有本地安全措施和规定。根据不同的应用，可能存在需要采取额外保护/安全措施的风险，例如，直线滑轨操作的工件可能对操作员具有固有的危险。

### 1.4.3. 有效性和责任

始终遵守有关自动化安全和通用机器安全的本地和国家法律，法规和指令。

本设备只能在其技术数据范围内使用。产品的任何其他使用均被视为不当和意外使用。

对于因任何不当使用或不当使用引起的任何损坏，UFACTORY 将不承担任何责任。

## 2. 安装

以下小节将指导您完成 UFactory 直线滑轨的安装和常规设置。

- (1) 发货清单
- (2) 机械安装部分

### 警告

安装之前：

阅读并理解与 UFactory 直线滑轨有关的安全说明。

根据发货清单和订单验证包裹。

备有需求中列出的所需零件。

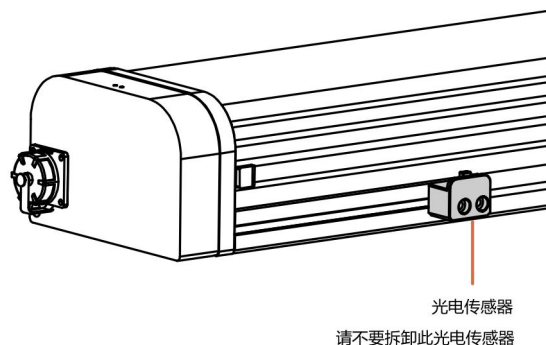
安装时：

满足环境条件。

在牢固地固定住直线滑轨并清除危险区域之前，请勿操作直线滑轨或打开电源。

请不要拆卸直线滑轨上的光电传感器，否则会丢失直线滑轨原点位置。

请勿让强磁体靠近磁栅。



## 2.1. 发货清单

UFactory 直线滑轨套件通常包括以下物品（如下图所示）：

UFactory 直线滑轨\*1（包括机械臂供电电缆\*1、机械臂通信电缆\*1）

直线滑轨供电电缆\*1

AC Control Box Pro 控制器\*1

AC Control Box Pro 控制器电源电缆\*1

网线\*1

M6\*20 杯头内六角螺丝 (28 个) 和 M6 弹垫 (28 个)

M5\*12 杯头内六角螺丝 (5 个) 和 M5 弹垫 (5 个)

5MM L 型扳手\*1

4MM L 型扳手\*1

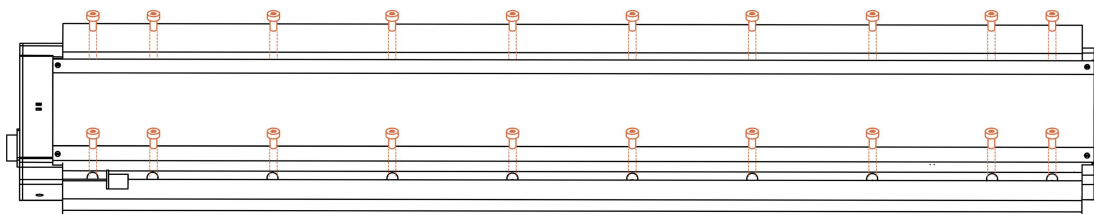
USB 转 485 线缆\*1

## 2.2. 机械安装

UFactory 直线滑轨通过单根设备电缆直接与 AC Control Box Pro 控制器连接。设备电缆为直线滑轨提供 48V 电源，并实现了与机械臂控制器的串行 RS485 通信。

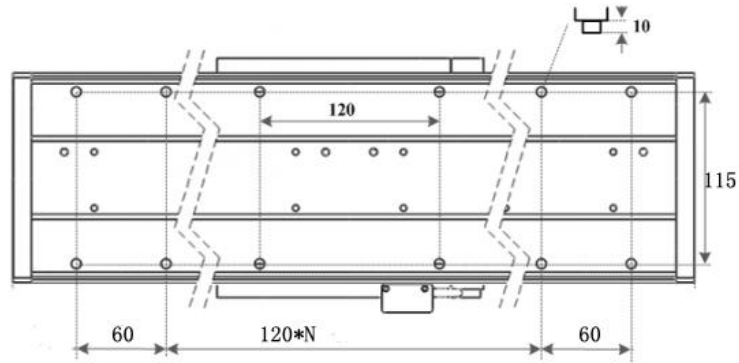
安装直线滑轨流程：

1. 确定直线滑轨及机械臂工作空间（避免碰到其他设备或障碍物）；直线滑轨上有 20 个  $\phi 6.6$  螺丝孔，用于将直线滑轨固定在工作台上。包装内有 28 颗 M6\*20 螺丝，请用其中 20 颗 M6\*20 螺丝将直线滑轨固定。请水平方向安装直线滑轨，



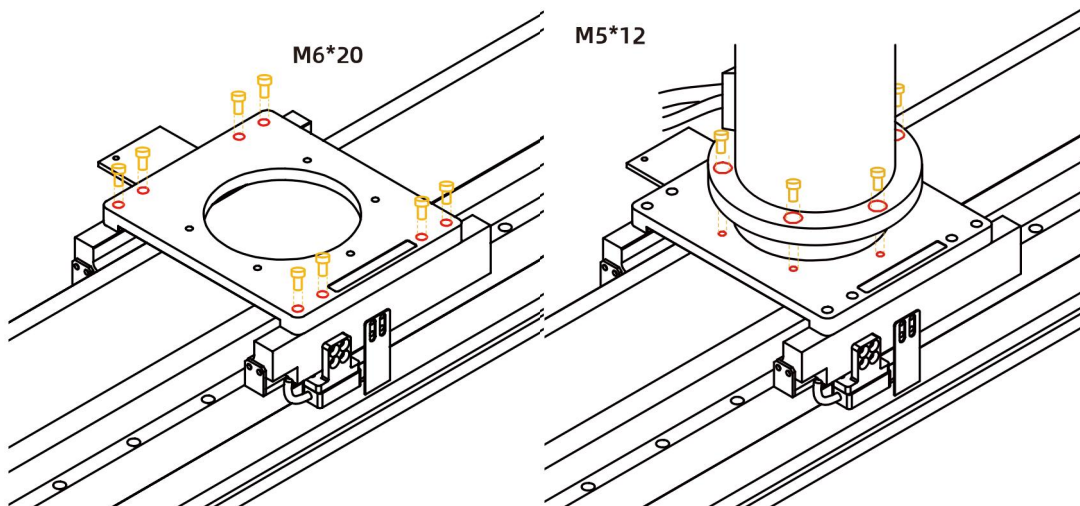


切勿垂直方向安装；



直线滑轨螺丝孔尺寸（单位：mm）

2. 用 8 颗 M6\*20 螺丝将底座板固定于直线滑轨上。



3. 用 5 颗 M5\*12 螺丝将机械臂固定于直线滑轨底座板上

4. 线缆连接：

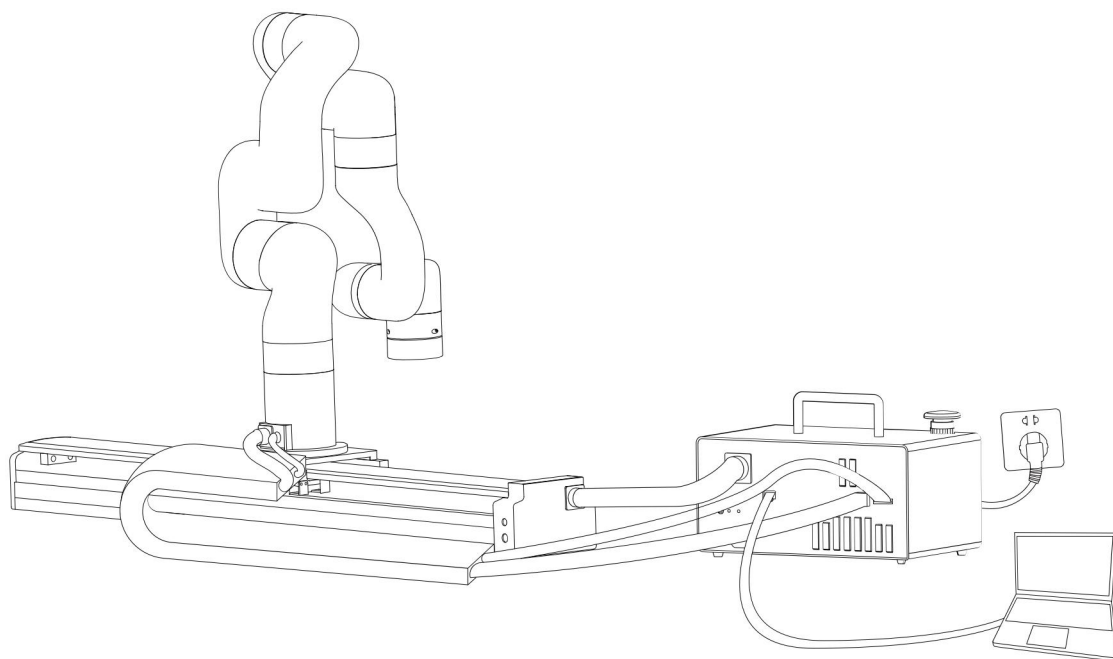
1) 将直线滑轨供电电缆、网线插入 AC Control Box Pro 控制器，接口具备防呆功能；

2) 将机械臂供电电缆和机械臂通信电缆接头插入机械臂接口，接口具备防呆功能；

3) 将机械臂供电电缆、通信电缆插入 AC Control Box Pro 控制器，接口具备防呆功能；

5. 打开控制器电源开关，松开急停按钮

6. xArmStudio-设置-工具设置-直线滑轨，点击“初始化”按钮回零并初始化滑轨。



**注意：**

1. 连接所有线缆时控制器上的急停开关一定要处于按下状态，机械臂电源指示灯熄灭，避免热插拔引起机械臂故障；
2. 因直线滑轨没有刹车设计，必须水平方向安装直线滑轨；

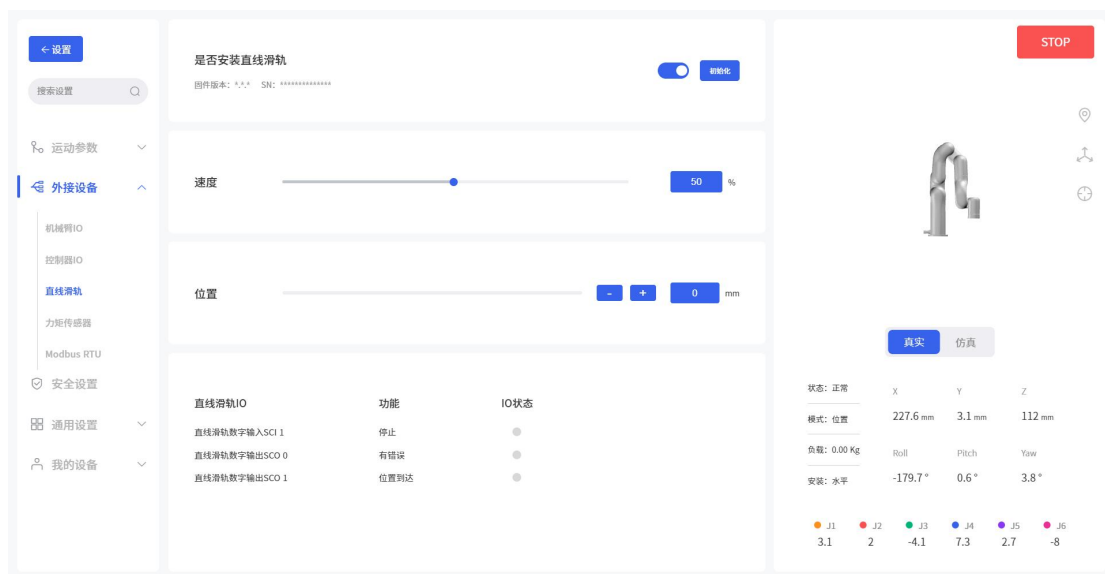
## 3. 直线滑轨的控制方式

### 3.1. 用 xArm Studio 控制直线滑轨

#### 3.1.1 设置直线滑轨

进入【设置】-【外接设备】-【直线滑轨】

- 是否安装直线滑轨：打开
- 点击“初始化”按钮



1. 在直线滑轨设置界面，打开“是否安装直线滑轨”选项。
2. 在直线滑轨设置界面，直线滑轨每次通电后，点击“初始化”按钮，对直线滑轨进行使能初始化并回零。初始化完成后，才能控制滑轨。

#### 3.1.2 控制直线滑轨

1. 在直线滑轨设置界面控制滑轨

可通过进度条、+/-键、速度百分比输入框来调整直线滑轨的速度和位置。

是否安装直线滑轨  初始化

固件版本: \*.\*.\* SN: \*\*\*\*\*

---

速度  %

---

位置  mm

直线滑轨IO	功能	IO状态
直线滑轨数字输入SCI 1	停止	●
直线滑轨数字输出SCO 0	有错误	●
直线滑轨数字输出SCO 1	位置到达	●

2.可以通过观察直线滑轨的 IO 查看对应功能的 IO 状态情况。

3. 通过 Blockly 来控制直线滑轨

### UF Linear Motor Blockly

这段程序的作用：执行此程序，可控制直线滑轨从原点到最远点以最高速度往复运动 10 次。

点击【修改】按钮，弹出快捷按钮调整位置



注意：

1) 当直线滑轨首次上电时，直线滑轨需要完成初始化后才能使用。

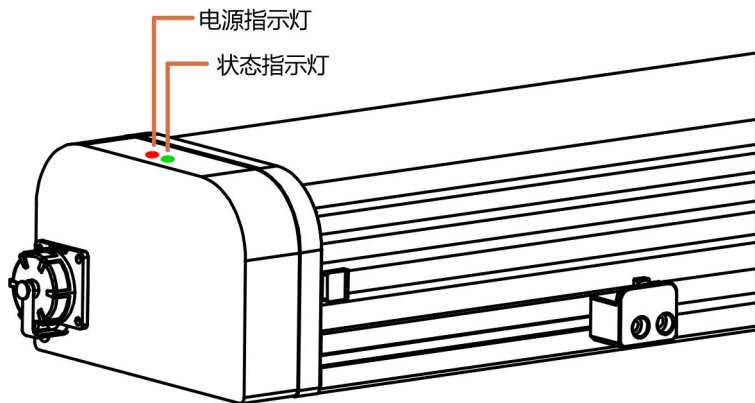
### 3.2. 用 Python-SDK 控制直线滑轨

对于使用 Python-SDK 控制直线滑轨的详细内容请见点击下面的链接查看：

[https://github.com/xArm-Developer/xArm-Python-SDK/blob/master/example/wrapper/common/9000-set\\_linear\\_track.py](https://github.com/xArm-Developer/xArm-Python-SDK/blob/master/example/wrapper/common/9000-set_linear_track.py)

### 3.3. 指示灯

直线滑轨指示灯位于直线滑轨端板处。



1. 电源指示灯：上电时，亮红灯
2. 状态指示灯：绿灯常亮表示直线滑轨正常运行；绿灯闪烁表示直线滑轨出现故障。

### 3.4. 直线滑轨 IO

直线滑轨配置三路 IO，一路输入 IO，两路输出 IO。

SCI 1：直线滑轨急停，低电平有效。

SCO 0：输出高电平，表示直线滑轨有错误。

SCO 1：输出高电平，定位结束表示位置已到达

## 4. 报警与处理

报警处理方式步骤如下：

1. 若遇到软件报错，请参照[报警处理方式](#)
2. 若无效，请对直线滑轨重新上电。按下 AC Control Box Pro 控制器上的急停按钮，5 秒后重新释放急停按钮，点击 xArmStudio“初始化”按钮使直线滑轨回零并使能。

若多次重新上电无效后请寻找 UFACTORY 团队支持。

软件报错代码	报警代码	报警处理
T9	0x09	直线滑轨电流检测异常 请通过控制器上的紧急停止按钮重启机械臂
T11	0x0B	直线滑轨电流过大 请点击“清除错误”清除直线滑轨报错
T12	0x0C	直线滑轨速度过大 请点击“清除错误”清除直线滑轨报错
T14	0x0E	直线滑轨位置指令过大 请点击“清除错误”清除直线滑轨报错
T15	0x0F	直线滑轨 EEPROM 读写错误 请点击“清除错误”清除直线滑轨报错
T20	0x14	直线滑轨驱动 IC 硬件异常 请点击“清除错误”清除直线滑轨报错
T21	0x15	直线滑轨驱动 IC 初始化异常 请点击“清除错误”清除直线滑轨报错
T23	0x17	直线滑轨电机位置偏差过大 请检查直线滑轨运动是否受阻，如直线滑轨运动未受阻，请点击“确认”清除直线滑轨报错
T25	0x19	直线滑轨指令超软件限位 请检测直线滑轨指令是否设置超出软件限制
T26	0x1A	直线滑轨反馈位置超限软件限位
T33	0x21	直线滑轨驱动器过载
T34	0x22	直线滑轨电机过载
T36	0x24	直线滑轨驱动器类型错误 请点击“清除错误”清除直线滑轨波按错
上表中未出现的报警代码：直线滑轨异常。如频繁出现，请联系技术支持。		

xArm-Python-SDK 报警处理方式:

在用 Python 库设计直线滑轨运动规划时,如果直线滑轨出现故障,需要手动清除错误。清除错误后,使能直线滑轨。

Python 库清除错误步骤: (如下接口,详细说明请查看 GitHub)

1. 清除错误: `clean_linear_track_error`
2. 首次上电需要先初始化并使能直线滑轨:

`set_linear_track_enable`

`set_linear_track_back_origin`



## 5. UFactory 直线滑轨技术规格

名称	说明
持续推力	62N
峰值推力	160N
最大速度	1000mm/s
行程范围	AL0700: 0-700mm AL1000: 0-1000mm AL1500: 0-1500mm
最大负载重量	200kg
通信方式	RS-485
通信协议	Modbus RTU
可编程参数	位置, 速度
反馈	位置
尺寸 ( L * W * H )	AL0700: 1092.6*213*114.6mm AL1000: 1382*214*115mm AL1500: 1884.6*213*114.6mm
重量	AL0700: 20kg AL1000: 24kg AL1500: 30.4kg
控制器尺寸 ( L * W * H )	262*185*176mm
控制器重量	4.44Kg

## 6. 售后服务

### 1. 售后政策：

对于产品的质量保证以及维修和退换货的详情，见官网的售后政策：

<https://www.cn.ufactory.cc/warranty>

### 2. 售后服务流程：

（1）联系技术支持（support@ufactory.cc），确认产品需要寄回维修，确定需要寄回的部件。

（2）我司根据售后政策，判定产品保修状况，付费或免费维修。

（3）维修、测试完成后，我们会将产品寄回，一般情况下，整个维修流程大约需要 1-2 周。

### 注意：

1. 当需要将产品寄回我司进行维修时，需要将产品用包装箱打包好，避免在运输过程中发生不必要的碰撞，导致直线滑轨受损。

## 附录

### 附录 1-用 Modbus RTU 通讯协议控制直线滑轨

#### 1.1 简要介绍

直线滑轨默认为标准 Modbus RTU 协议，默认波特率 2Mbps，直线滑轨 ID 为 0x01。目前支持的功能码有：0x03/0x06/0x10。

直线导轨常用的地址有：0x0100(直线滑轨使能)、0x0700(直线滑轨位置命令)、0x0303(直线滑轨运行速度)、0x0A0A(直线滑轨回零)、0x0404(直线滑轨回零速度)、0x004F(获取直线滑轨回零完成状态)。

如果需要存 EEPROM，将通讯地址和 0x1000 进行‘或’运算。

例如：写伺服运行模式到 EEPROM 中。通讯地址（0x0A0B）需改为 0x1A0B

注：直线导轨每次上电后需要初始化一次

#### 1.2 读取直线滑轨寄存器

读取保持寄存器			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	<b>Address</b>
	寄存器数量	2 Bytes	<b>N*</b>
	Modbus CRC 16	2 Bytes	<b>CRC*</b>
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Byte	<b>N*x2</b>
	寄存器值	<b>N*x2 Bytes</b>	<b>Value</b>
	Modbus CRC16	2 Bytes	<b>CRC*</b>

注：N\* = 寄存器数量

Address= 寄存器起始地址（见下面列表）

CRC\* = 循环冗余校验

	寄存器起始地址	寄存器值	
获取直线滑轨状态	0x0000	2bytes	定位结束: 0x0000      运动状态: 0x0001 停止状态: 0x0002 急停和断使能, 都属于停止状态。
获取直线滑轨位置	0x0702	4bytes	位置取值范围: 0xFFFFFFFFB-0x00000320
获取直线滑轨错误	0x000F	2bytes	有错误: 其他返回值都代表有错误 (除 0 以外) 无错误: 0x0000

### 1.3 写入直线滑轨寄存器

写入寄存器			
<b>请求指令格式</b>			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
	字节数	1 Byte	N*x2
	寄存器	N*x2 Bytes	Value
	Modbus CRC 16	2 Bytes	CRC*
<b>响应指令格式</b>			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
	Modbus CRC 16	2 Bytes	CRC*

注: N\* = 寄存器数量

Address= 寄存器起始地址 (见下面列表)

CRC\* = 循环冗余校验

### 1.4 Modbus RTU 示例

#### 1. 使能直线滑轨

使能直线滑轨			
请求指令格式			
Modbus RTU 数据	直线导轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01, 0x00
	寄存器数量	2 Bytes	0x00, 0x01
	字节数	1 Byte	0x02
	寄存器	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0x1D, 0x00
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01, 0x00
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0x00, 0xAC

### 2. 设置直线滑轨位置

设置直线滑轨位置			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x07, 0x00
	寄存器数量	2 Bytes	0x00, 0x02
	字节数	1 Byte	0x04
	寄存器:直线导轨位置 200MM	4 Bytes	0x00, 0x1E, 0x84, 0x80
	Modbus CRC16	2 Bytes	0x7B, 0x62
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x07, 0x00
	寄存器数量	2 Bytes	0x00, 0x02
	Modbus CRC16	2 Bytes	0x40, 0x25

### 3. 设置直线滑轨速度

设置直线滑轨速度			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x03, 0x03
	寄存器数量	2 Bytes	0x00, 0x01
	字节数	1 Byte	0x02

	寄存器（设置速度为 6000r/min）	2 Bytes	0x17, 0x70
	Modbus CRC16	2 Bytes	0xFD, 0xFA
<b>响应指令格式</b>			
<b>Modbus RTU 数据</b>	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x03, 0x03
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0xF1, 0x14

4. 设置直线滑轨回零

<b>设置直线滑轨回零</b>			
<b>请求指令格式</b>			
<b>Modbus RTU 数据</b>	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x06
	寄存器起始地址	2 Bytes	0x0A, 0x0A
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0xFD, 0xFA
<b>响应指令格式</b>			
<b>Modbus RTU 数据</b>	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x0A, 0x0A
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0xFD, 0xFA

5. 设置直线滑轨回零速度

<b>设置直线滑轨回零速度</b>			
<b>请求指令格式</b>			
<b>Modbus RTU 数据</b>	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x04, 0x04
	寄存器数量	2 Bytes	0x00, 0x01
	字节数	1 Byte	0x02
	寄存器设置速度为 3000r/min	2 Bytes	0x0B, 0xB8
	Modbus CRC16	2 Bytes	0xFD, 0xFA
<b>响应指令格式</b>			
<b>Modbus RTU 数据</b>	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x04, 0x04
	寄存器数量	2 Bytes	0x00, 0x01

	Modbus CRC16	2 Bytes	0xF1, 0x14
--	--------------	---------	------------

6. 获取直线滑轨回零是否完成

获取直线滑轨回零是否完成			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x0A, 0x25
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0XB5, 0xDD
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Bytes	0x02
	寄存器值	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0x79, 0x84

7. 监控光电传感器与第一个 Z 相（离开光电传感器后遇到的第一个 Z 相）的距离

监控光电传感器与第一个 Z 相（离开光电传感器后遇到的第一个 Z 相）的距离			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x0A, 0x28
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0xE4, 0x1D
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Bytes	0x02
	寄存器值	2 Bytes	0x06, 0xF2
	Modbus CRC16	2 Bytes	0x3A, 0x61

注：这个的寄存器值是要在回零完成后，才能获取。可以通过移动光电传感器改变此值。此值最大值是 4000，建议最好大于 1000。

8. 定位结束范围

设置定位结束范围			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x0A, 0x0B

	寄存器数量	2 Bytes	0x00, 0x01
	字节数	1 Byte	0x02
	寄存器设置范围为 1000	2 Bytes	0x03, 0xE8
	Modbus CRC16	2 Bytes	0x0C, 0xC0
<b>响应指令格式</b>			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x0A, 0x0B
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0x73, 0xD3

9. 软急停

<b>触发软急停</b>			
<b>请求指令格式</b>			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x0A, 0x0E
	寄存器数量	2 Bytes	0x00, 0x01
	字节数	1 Byte	0x02
	寄存器	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0xCC, 0xBE
<b>响应指令格式</b>			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x0A, 0x0E
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0x63, 0xD2

10. 直线滑轨 SN 区域

寄存器地址从 0x0B10 - 0x0B1F

<b>获取直线滑轨 SN</b>			
<b>请求指令格式</b>			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x0B, 0x10
	寄存器数量	2 Bytes	0x00, 0x02
	Modbus CRC16	2 Bytes	0XB5, 0xDD



响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Bytes	0x04
	寄存器值	2 Bytes	
	Modbus CRC16	2 Bytes	

11. 获取输入 SCI 状态

获取输入 SCI 状态			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x0A, 0x26
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0x86, 0x17
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Byte	0x02
	寄存器值	2 Bytes	**
	Modbus CRC16	2 Bytes	**

12. 获取输出 SCO 状态

获取输出 SCO 状态			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x0A, 0x27
	寄存器数量	2 Bytes	0x00, 0x01
	Modbus CRC16	2 Bytes	0xD7 0xD7
响应指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Bytes	0x02
	寄存器值	2 Bytes	**
	Modbus CRC16	2 Bytes	**

13. 获取状态区域

起始地址 0x0A20

获取状态区域			
请求指令格式			
Modbus RTU 数据	直线滑轨 ID	1 Byte	0x01

	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x0A, 0x20
	寄存器数量	2 Bytes	0x00, 0x08
	Modbus CRC16	2 Bytes	0x46 0x1E
<b>响应指令格式</b>			
<b>Modbus RTU 数据</b>	直线滑轨 ID	1 Byte	0x01
	功能码	1 Byte	0x03
	字节数	1 Bytes	0x10
	寄存器值	16 Bytes	**
	Modbus CRC16	2 Bytes	**

寄存器值 - 16 Bytes: 00 00 0F A0 00 00 00 00 01 00 01 00 02 00 02

1-4 bytes: 00 00 0F A0 , 相对零点的当前位置(单位: 脉冲个数)

5-6 bytes: 00 00, 获取直线滑轨状态(与地址 0x0000 相同)

7-8 bytes: 00 00, 错误码

9-10 bytes: 00 01, 使能状态

11-12 bytes: 00 01, 回零状态

12-14 bytes: 00 02, SCI 状态

15-16 bytes: 00 02, SCO 状态