

# DobotSCStudio 机器人 控制软件用户手册

文档版本: V1.0

发布日期: 2020.05.20

深圳市越疆科技有限公司



# 注意事项

#### 版权所有 © 越疆科技有限公司2020。 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。

#### 免责声明

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,越疆不提供任何形式的明示或默示保证,包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证;亦不对使用本手册或使用本公司产品导致的任何特殊、附带、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 在充分了解机械臂及其相关知识的前提下使用机械臂。越疆建议您在专业人员的指导下使用 本手册。该手册所包含的所有安全方面的信息都不得视为Dobot的保证,即便遵循本手册及 相关说明,使用过程中造成的危害或损失依然有可能发生。

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,确保在越疆机械臂的 使用中不存在任何重大危险。

# 越疆科技有限公司

地址: 深圳市南山区留仙大道3370号南山智园崇文区2号楼9-10楼

网址: <u>cn.dobot.cc</u>

文档版本 V1.0(2020-05-20)

用户手册



# 前言

# 目的

本手册介绍了机器人控制软件DobotSCStudio的功能和使用方法,方便用户了解和使用 Dobot机器人。

# 读者对象

本手册适用于:

- 客户工程师
- 销售工程师
- 安装调测工程师
- 技术支持工程师

# 修订记录

时间	修订记录
2020/05/18	第一次发布

# 符号约定

在本手册中可能出现下列标志,它们所代表的含义如下。

符号	说明
<b>企</b>	表示有高度潜在危险,如果不能避免,会导致人员 死亡或严重伤害
	表示有中度或低度潜在危害,如果不能避免,可能 导致人员轻微伤害、机械臂毁坏等情况
▲ 注意	表示有潜在风险,如果忽视这些文本,可能导致机 械臂损坏、数据丢失或不可预知的结果
<b>□</b> 说明	表示是正文的附加信息,是对正文的强调和补充

文档版本 V1.0 (2020-05-20)



# 目录

注	意事項	页	••••••	•••••••••••••••••••••••••••••••••••••••	i
前	言	•••••	••••••	••••••	ii
1.	功能	说明	••••••	••••••	1
	1.1	简介			1
		1.1.1	主界面说明		1
	1.2	设置			2
		1.2.1	运动参数设置		3
		1.2.2	用户坐标系设	置	6
		1.2.3	工具坐标系设	置	9
		1.2.4	回零		
		1.2.5	标定		
		1.2.6	虚拟仿真		
		1.2.7	日志		
		1.2.8	语言切换		
		1.2.9	末端负载		
		1.2.10	网络配置		
	1.3	I/O 监控	호 		
	1.4	远程控制	制		
		1.4.1	远程 I/O		
		1.4.2	远程 Modbus		
	1.5	编程			
		1.5.1	工程说明		
		1.5.2	编程面板说明		
		1.5.3	编程说明		
	1.6	使能			40
	1.7	设置全	局速率		40
	1.8	报警说	明		41
2.	编程	语言	••••••	••••••	
	2.1	算术运	算符		43
	2.2	关系运	算符		43
	2.3	逻辑运	算符		44
	2.4	一般关	键字		44
	2.5	一般符	号		44
	2.6	过程控	制指令		44
	2.7	全局变	量		
	2.8	运动指	令		45
	2.9	运动参	数设置指令		
	2.10	输入/输	出指令		55
	2.11	程序管	理指令		
	2.12	获取位	姿指令		61
	2.13	ТСР			
	2.14	UDP			
	文档)	版本 V1.0	) (2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



	2.15	Modbu	s	69
		2.15.1	Modbus 寄存器说明	69
		2.15.2	指令说明	70
	2.16	ECP		73
	2.17	1 工艺指	令	74
		2.17.1	传送带跟踪指令	74
		2.17.2	码垛指令	76
3.	工艺	指南		81
	3.1	传送带	跟踪	81
		3.1.1	概述	81
		3.1.2	搭建环境	81
		3.1.3	标定传送带	83
		3.1.4	配置传送带	86
		3.1.5	应用案例	96
	3.2	码垛工	艺	
		3.2.1		
		3.2.2	码垛设置	
		3.2.3	应用案例	
4.	典型	应用…	,,, .	
	4.1	Modbu	s 应用	
	4.2	1/0 应月	,, ≈ ∏	
		4.2.1		
			4/15 / F/1FU 4 /	



# 1. **功能说明**

# 1.1 简介

SC系列控制柜配备DobotSCStudio配套使用。支持用户二次开发,并提供多种机械结构的运动学算法,适用于各种工艺应用。

# 1.1.1 主界面说明

DobotSCStudio主界面如图 1.1所示,其界面详细说明如表 1.1所示。



图 1.1 主界面

表	1.1	DobotSCStudio	面板说明
---	-----	---------------	------

编号	说明		
1	工程		
	用户可导入工程,并对工程进行调试,运行等操作		
2	点动		
	在不同坐标系下控制机械臂的运动,仅在手动模式 下有效		
	若在关节坐标系下点动:从上往下表示点动J1、J2、		
	J3、J4、J5、J6		
	若在笛卡尔坐标系下点动:从上往下表示点动X、		
	Y, Z, R (A), B, C		
3	系统		
	用户可在该页面进行系统设置,如网络设置,机器		
文档版本 V1.0(2020-05-20) 用户	手册 版权所有 © 越疆科技有限公司		



编号	说明			
	人参数设置,坐标系设置,工艺包设置等			
4	单击图标可切换手动模式和自动模式			
	<ul> <li>• • • : 手动模式。点动机械臂、编程或设置参</li> </ul>			
	<ul> <li>数等</li> <li>●:自动模式。监控运动轨迹、I/O端口或调 试运行程序等</li> </ul>			
5	• 手动模式下仅表示机械臂电机使能状态			
	• 自动模式下单击此按钮控制电机使能			
	详细说明请参见1.6 使能			
6	机械臂报警信息查询			
	当机械臂有报警时,该图标变为红色			
	可 在 操 作 面 板 查 看 具 体 的 报 警 信 息 并 将 DobotSCStudio切换至手动模式后清除报警			
7	单击图标可设置机械臂全局速率比例,详细请参见 1.7 设置全局速率			
8	用户模式选择			
	<ul> <li>观察员:查看全局变量、I/O状态、机器姿态、</li> <li>位置信息、报警信息等</li> </ul>			
	• 操作人员:在监控人员权限的基础上,还可基于现有的程序脚本文件操作,无需编程			
	• 程序员:在操作人员权限的基础上,还可编程、 示教			
	• 管理员: 可设置或修改任何参数			
	根据使用者的角色,在操作面板选择用户模式并登 录。其中,监控人员无需密码可直接登录			
	默认密码: admin,用户可在管理员模式下进入"配 置>基础配置>用户模式"页面下修改密码			
9	急停开关			
	机械臂运行过程中出现突发情况时可按下急停开 关,使控制柜伺服驱动断电,机械臂紧急停止			
5	欢迎页面可查看软件说明书,应用案例等			

# 1.2 设置

机械臂示教和再现前,需对机械臂进行一系列设置,包括运动参数设置、语言选择、用 户模式、工艺包设置等。

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



#### 1.2.1 运动参数设置

机械臂在示教或再现时可设置各坐标系的运动速度和加速度以及再现时门型运动相关 参数,设置完成后请单击"保存"。单击"≤≤>配置>机器人配置"进入机器人参数界面。

关节点动参数:设置点动时各关节轴的最大速度和最大加速度。六轴机械臂关节点 • 动参数如图 1.2所示。

关节点动参数           坐标系点动参数           坐标系点动参数           坐标系再现参数           二           1型参数	点动关节参数			默认(保存)
关节再现参数	」,2≶) 到塹ቢ	100.00	J1加速度(°/s2)	500.00
	J2速度(°/s)	100.00 *	J2加速度(°/s2)	500.00
	]3速度(°/s)	100.00 Å	J3加速度(°/s2)	500.00
	J4速度(°/s)	100.00 *	J4加速度(°/s2)	500.00 ×
	]5速度(°/s)	25.00 Å	J5加速度(°/s2)	25.00
	J6速度(°/s)	26.00 *	J6加速度(°/s2)	26.00

图 1.2 点动关节参数

□□说明

如果机械臂为水平多关节机械臂(SCARA四轴),则J5、J6相关参数不需设置。

机械臂点动时各关节实际运动速度与最大速度关系如下所示:

- 各关节点动速度=各关节最大速度\*全局速率
- 各关节点动加速度=各关节最大加速度\*全局速率

□□说明

- 全局速率可在主界面进行设置,详细请参见1.7 设置全局速率。 •
- 指令设置的百分比可在编程时调用速度指令设置,详细请参见2.9 运动参数 • 设置指令。
- 坐标系点动参数:设置点动时各笛卡尔坐标轴的最大速度和最大加速度。六轴机械 臂坐标系点动参数如图 1.3所示。

文档版本 V1.0 (2020-05-20)



关节点动参数				
上 坐标系点动参数		默认(保存)		
业标系再现参数	点动坐标系参数			
门型参数				
关节再现参数	X速度(nm/z)	48.00	↓ ▼ X加速度(nm/s2)	201.00
	Y速度(nm/s)	50.00	Y加速度(nn/s2)	200.00
	Z速度(nm/s)	50.00	↓ ▼ Z加速度(nn/s2)	200.00 *
	Rx速度(nm/s)	50.00	Rx加速度(nn/s2)	200.00
	Ry速度(nm/s)	1.00	Ry加速度(nn/m2)	36.00 *
	Rz速度(nm/s)	1.00	Rz加速度(nn/s2)	44.00 *

图 1.3 点动坐标系参数

#### □□说明

如果机械臂为水平多关节机械臂(SCARA四轴),则Rx表示R轴,Ry、Rz相关参数 不需设置。

机械臂点动时各笛卡尔坐标轴实际运动速度与最大运动速度关系如下所示:

- 各笛卡尔坐标轴点动速度=各笛卡尔坐标轴最大速度\*全局速率
- 各笛卡尔坐标轴点动加速度=各笛卡尔坐标轴最大加速度速度\*全局速率
- 示教再现关节参数:设置再现时各关节轴的最大速度和最大加速度。六轴机械臂示教再现关节参数如图 1.4所示。

	关节点动参数										
<u>i</u>	坐标系点动参数										
	坐标系再现参数	示教	示教再现关节参数							孚	
6	门型参数										
<u>^</u>	关节再现参数		J1速度(°/s)	540.00	, J	1加速度(°/s2)	15000.00	÷ J	1加加速度(°/s3)	25002.00	*
			J2速度(°/s)	540.00	J	2加速度(°/s2)	15000.00	÷ J	2加加速度(°/s3)	50000.00	*
			J3速度(°/s)	600.00	J	3加速度(°/s2)	25000.00	≜ Ţ	3加加速度(°/s3)	80000.00	*
			J4速度(°/s)	1350.00	, J	4加速度(°/s2)	25000.00	÷ J	4加加速度(°/s3)	25000.00	*
			J5速度(°/s)	21.00	, J	5加速度(°/s2)	65.00	÷ J	5加加速度(°/s3)	88.00	*
			J6速度(°/s)	1.00 🗘	, J	6加速度(°/s2)	65.00	÷ J	6加加速度(°/s3)	96.00	*

图 1.4 示教再现关节参数

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



□□说明

如果机械臂为水平多关节机械臂(SCARA四轴),则J5、J6相关参数不需设置。

机械臂再现时各关节实际运动速度与最大速度关系如下所示:

- 各关节再现速度=各关节最大速度\*全局速率\*指令设置的百分比
- 各关节再现加速度=各关节最大加速度\*全局速率\*指令设置的百分比
- 各关节再现加加速度=各关节最大加加速度\*全局速率\*指令设置的百分比
- 示教再现坐标系参数:设置再现时各笛卡尔坐标轴的最大速度和最大加速度。六轴 机械臂示教再现坐标系参数如图 1.5所示。

	关节点动参数						
	坐标系点动参数						
<u>Ľ</u>	坐标系再现参数						
	门型参数	示教再现坐标系参数	t		(	默认	(保存)
<u>^</u>	关节再现参数						
				_			
		XYZ速度(mm/s)	1000.00	* *	RxRyRz <u>速度</u> (mm/s)	997.00	* *
		XYZ加速度(mm/s2)	80000.00	4. 7	RxRyRz加速度(mm/s2)	50000.00	 ▼
		Z¥Z加加速度(mm/s3)	80000.00	4. W	RxRyRz加加速度(mm/s3)	50000.00	 ▼

#### 图 1.5 示教再现关节坐标系参数

#### □□说明

如果机械臂为水平多关节机械臂(SCARA四轴),则RxRyRz表示R轴。

机械臂再现时各笛卡尔坐标轴实际运动速度与最大速度关系如下所示:

- 各笛卡尔坐标轴再现速度=各笛卡尔坐标轴最大速度\*全局速率\*指令设置的百分比
- 各笛卡尔坐标轴再现加速度=各笛卡尔坐标轴最大加速度\*全局速率\*指令设置 的百分比
- 各笛卡尔坐标轴再现加加速度=各笛卡尔坐标轴最大加加速度\*全局速率\*指令 设置的百分比
- 弧形再现参数:再现时若机械运动模式为 "Jump" 时,需设置起始点抬升高度、结



束点下降高度以及抬升最大高度。

DobotSCStudio支持设置10组Jump相关参数,设置并勾选任一组参数后可在编程时 调用,如图 1.6所示。

> 关节点动参	数 示教	な再现门想	型参数 <b>(</b> 重	单位:毫米	)			(默认)	保存
坐标系点动参	数	更能 No.	0 j	起始高度	5.00	结束高度	54.00	最大太升高度	101.00
坐标系再现参	·数 ✓ 1	更能 No.	1 j	起始高度	8.00	结束高度	97.00	最大太升高度	6.00
	<sup>*</sup> 数	吏能 No.	2 j	起始高度	6.00	结束高度	19.00	最大太升高度	92.00
		更能 No.	3 j	起始高度	7.00	结束高度	31.00 ÷	最大太升高度	17.00
	<b>⊻</b> 1	更能 No.	4 j	起始高度	7.00	结束高度	56.00	最大太升高度	81.00
	<b>✓</b> 1	更能 No.	5 j	起始高度	7.00	结束高度	31.00	最大太升高度	49.00
	<b>⊻</b> 1	更能 No.	6 j	起始高度	7.00	结束高度	52.00	最大太升高度	14.00
	<b>∠</b> 1	更能 No.	7 j	起始高度	7.00	结束高度	79.00	最大太升高度	63.00
	<b>∠</b> 1	走能 No.	8 j	起始高度	7.00	结束高度	96.00	最大太升高度	95.00
	<b>⊻</b> 1	史能 No.	9 j	起始高度	7.00	结束高度	85.00	最大太升高度	21.00

#### 图 1.6 弧形再现参数

## 1.2.2 **用户坐标系设置**

当工件的位置发生变化或机械臂的运行程序需要在多个同类型的加工系统中重复使用时,此时需要设置用户坐标系,使所有路径都跟随用户坐标同步更新,极大的简化了示教编程。当前系统支持10个用户坐标系,用户坐标系0为基坐标系,不可更改。

# ⚠注意

建立用户坐标系时,请确保参考坐标系为默认坐标系,即增加用户坐标系时 DobotSCStudio界面的用户坐标系图标为用户坐标系: 0。

# 1.2.2.1 四轴用户坐标系设置

四轴用户坐标系采用两点示教法生成:将机械臂移动至任意两点A(x1,y1,z1)和B(x2,y2,z2),其中A点作为原点,AB两点之间的连线确定用户坐标系X轴正方向,然后根据 右手定则确定Y轴和Z轴方向,如图 1.7所示。





图 1.7 用户坐标系两点示教法

本节以建立用户坐标系1为例进行说明。

# 前提条件

- 机械臂已上电。
- DobotSCStudio已切换至手动模式。

# 操作步骤

步骤1 单击"三>配置>坐标系设置>四轴用户坐标系"进入"用户坐标系"界面,

如图 1.8所示。



图 1.8 用户坐标系界面

**步骤2** 使能机械臂,并点动机械臂至任意一点,在"第一个点"区域单击"获取第一个点"获取第一个点的坐标。

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



步骤3 点动机械臂至另一点,在"第二个点"区域单击"获取第二个点"获 取第二个点的坐标。

步骤4 单击"保存",用户坐标系1参数自动生成。

步骤5 在点动界面单击"坐标系选择", 选择"坐标系:1"。

此时可在用户坐标系1下进行示教、编程等操作。

# 1.2.2.2 **六轴用户坐标系设置**

六轴用户坐标系采用三点示教法生成:将机械臂移动至任意三点A(x1,y1,z1),**B**(x2,y2,z2)和**C**(x3,y3,z3)。其中A点作为原点,AB两点之间的连线确定用户坐标系X轴正方向,C点沿X轴做垂线确定Y轴正方向,根据右手法则,确定Z轴方向,如所示。



图 1.9 三点示教法

本节以创建用户坐标系1为例进行说明。

# 前提条件

- 机械臂已上电。
- DobotSCStudio已切换至手动模式。

# 操作步骤

步骤1 单击" 三>配置>坐标系设置>六轴用户坐标系"。

进入"用户坐标系"界面,如图 1.10所示。





图 1.10 用户坐标系界面

#### □□说明

Rx、Ry、Rz为工具中心点TCP绕选定的用户坐标系X、Y、Z轴旋转的角度。

**步骤 2** 使能机械臂,并点动机械臂至任意一点,在"P1"页签单击"获取第一个点"获取第一个点的坐标。

步骤3 使能机械臂,并点动机械臂至第二个任意点,在"P2"页签单击"获 取第二个点"获取第二个点的坐标。

步骤4 使能机械臂,并点动机械臂至第三个任意点,在"P3"页签单击"获 取第三个点"获取第三个点的坐标。

步骤5 单击"添加"和"保存",用户坐标系1参数自动生成。

步骤6 在点动界面单击"坐标系选择", 选择"坐标系: 1"。

此时可在用户坐标系1下进行示教、编程等操作。

#### 1.2.3 工具坐标系设置

当机械臂末端安装了工具(如焊枪、喷嘴、夹具等)后,为了编程和机械臂运行的需要, 此时需要设置工具坐标系。例如,利用多个夹具同时搬运多个工件,可将每个夹具设置为独 立的工具坐标系以提高搬运效率。

当前系统支持10个工具坐标,工具坐标系0为默认工具坐标系,即不使用工具,不可更改。

# 

建立工具坐标系时,请确保参考坐标系为默认坐标系,即增加工具坐标系时 DobotSCStudio界面的工具坐标系图标为工具坐标系: 0。



# 1.2.3.1 **四轴工具坐标系设置**

四轴工具坐标系采用两点示教法生成:机械臂末端安装工具后,调整工具的位姿,使TCP (Tool Center Point)以两种不同的姿态对准空间中同一点(参考点),获取工具的位置偏移, 生成工具坐标系,如图 1.11所示。



图 1.11 工具坐标系两点示教法

本节以建立工具坐标系1为例进行说明。

# 前提条件

- 机械臂已上电。
- DobotSCStudio已切换至手动模式。

# 操作步骤

- 步骤1 在机械臂末端安装夹具,本节不做详细说明。
- **步骤 2** 单击"<sup>■</sup>>配置>坐标系设置>四轴工具坐标系"进入"工具坐标系"界面,如 图 1.12所示。

	12	四轴田白坐标系	I.	具坐标系	Ā			
		TROUP THAT		х	Y	Z	Rx	
		四轴工具坐标系	0		0.0000	0.0000	0.0000	
			1	123.0000	0.0000	0.0000	0.0000	
								4轴
								X: 0.0000 \$ Y: 0.0000 \$
								Z: 0.0000 \$ R: 0.0000 \$
								· · · · · · · · · · · · · · · · · · ·
								X. 0.000 v Y. 0.000 v
								Z: 0.0000 , R: 0.0000 ,
	20/0							添加保存 删除
	1018	*坐称						
文档版	本 V1.0(	2020-05-2	0)			用户	手册	版权所有 © 越疆科技有限公司



#### 图 1.12 工具坐标系界面

**步骤 3** 使能机械臂,并点动机械臂,使TCP(Tool Center Point)以第一种姿态对准参考点(空间中的一点),并在"第一个点"区域单击"获取第一个点"获取第一个点的坐标。

步骤4 使能机械臂,点动J4轴,切换至第二种姿态,再点动其他轴使TCP以 当前姿态对准参考点,并在"第二个点"区域单击"获取第二个点"获取第二 个点的坐标。

步骤5 单击"添加"和"保存",计算得到工具坐标系1的参数。

步骤6 在点动界面单击"坐标系选择", 选择"坐标系:1"。

此时可在工具坐标系1下进行示教、编程等操作。

结果验证

将机械臂的TCP点对准工件固定点,使能机械臂后点动R轴。若机械臂能够以不同姿态 绕工件旋转,则说明工具坐标系建立成功。

#### 1.2.3.2 六轴工具坐标系设置

六轴工具坐标系采用三点示教法 "TCP+ZX" 生成: 机械臂末端安装工具后,调整工具的位姿,使TCP(Tool Center Point)以三种不同的姿态对准空间中同一点(参考点),获取工具位置偏移。再根据另外三点(A、B、C)获取工具姿态偏移,如图 1.13所示。



图 1.13 工具坐标系三点示教法 "TCP+ZX"

本节以建立工具坐标系1为例进行说明。



### 前提条件

- 机械臂已上电。
- DobotSCStudio已切换至手动模式。

#### 操作步骤

- 步骤1 在机械臂末端安装夹具,本节不做详细说明。
- 步骤2 单击"5]>配置>坐标系设置>六轴工具坐标系。

进入"工具坐标系"界面,如图 1.14所示。



图 1.14 工具坐标系界面

□□说明

Rx、Ry、Rz为工具中心点TCP绕选定的工具坐标系X、Y、Z轴旋转的角度。

步骤3 使能机械臂,并点动机械臂,使TCP(Tool Center Point)以第一种姿态对准参考点(空间中的一点),并在"位置"页签的"P1"区域单击"获取 第一个点"获取第一个点的坐标。

步骤4 使能机械臂,并点动机械臂,使TCP以第二种姿态对准参考点,并在 "位置"页签的"P2"区域单击"获取第二个点"获取第二个点的坐标。

步骤5 使能机械臂,并点动机械臂,使TCP以第三种姿态对准参考点,并在 "位置"页签的"P3"区域单击"获取第三个点"获取第三个点的坐标。

步骤 6 使能机械臂,并点动机械臂,使TCP以垂直姿态对准参考点,在"姿态"页签的"P1"区域单击"获取第一个点"获取第四个点的坐标(A点)。 步骤 7 使能机械臂,并正向点动Z轴,使机械臂移动至另一点,在"姿态"

页签的"P2"区域单击"获取第二个点"获取第五个点的坐标(B点)。 该步骤确定Z轴正方向。

文档版本 V1.0 (2020-05-20)

版权所有 © 越疆科技有限公司



**步骤 8** 使能机械臂,并正向点动X轴,使机械臂移动至C点(不能与A、B两 点在同一直线上),在"姿态"页签的"P3"区域单击"获取第三个点"获取 第六个点的坐标。 该步骤确定X轴正方向,然后根据右手定则确定Y轴方向。 步骤 9 单击"添加"和"保存",计算得到工具坐标系1的参数。 步骤 10 在点动界面单击"坐标系选择",选择"坐标系:1"。 此时可在工具坐标系1下进行示教、编程等操作。

#### 1.2.4 回零

当更换机械臂电机、减速机等传动部件或者与工件发生碰撞等情况下,机械臂的零点位 置发生变化,此时需对机械臂进行回零操作。

# 1.2.4.1 水平多关节机械臂(SCARA 四轴)回零

#### 前提条件

机械臂已正常上电。

操作步骤

步骤1 机械臂下使能状态下,将机械臂置于出厂位置,如图 1.15所示。



图 1.15 出厂位置

□□说明

文档版本 V1.0(2020-05-20)



在机械臂关节相互转动的两个结构件上各有一个键槽,当转动机械臂关节使两个 键槽对应重合时,此位置为相应关节的出厂位置,如图 1.16所示。



图 1.16 回零线槽

**步骤 2** 顺时针旋转Z轴丝杠上的机械挡块至极限位置然后再逆时针旋转180° 至360°,防止电机堵转。机械挡块如图 1.17所示。



图 1.17 机械挡块位置

□□说明



机械挡块是为了防止机械臂运行至机械限位区域外,以免给机械臂和操作人员带 来伤害。

- 步骤3 将DobotSCStudio切换至手动模式,并使能机械臂。
- 步骤4 单击" ■>配置>机器人设置>回零"。

进入回零界面,如图 1.18所示。



图 1.18 回零界面

**步骤 5** 在"回零"界面单击

此时界面的消息窗口会显示"回零成功!"。

回零过程中机械臂不会运动,会将当前位置设置为零点位置。回零后最终位置 如图 1.19所示。





图 1.19 零点位置

# 1.2.4.2 垂直多关节机械臂(六轴)回零

# 前提条件

机械臂已上电。

## 操作步骤

步骤1 机械臂下使能状态下,将机械臂置于出厂位置,出厂位置如图 1.20所示。





图 1.20 出厂位置

□□说明

在机械臂关节相互转动的两个结构件上各有一个键槽,当转动机械臂关节使两个 键槽对应重合时,此位置为相应关节的出厂位置,如图 1.21所示。





图 1.21 回零线槽

步骤2 将DobotSCStudio切换至手动模式,使能机械臂。

步骤3 在"设置"界面单击"回零"。

进入回零界面,如图 1.18所示。



图 1.22 回零界面

步骤4 在"回零"界面单击 , 进行回零操作。

文档版本 V1.0(2020-05-20)



此时界面的消息窗口会显示"回零成功!"。

回零过程中机械臂不会运动,会将当前位置设置为零点位置。回零后最终位置如图 1.23所示。



图 1.23 零点位置

# 1.2.5 标定

机械臂出厂已进行过标定,若用户在实际应用过程中对机械臂精度要求较高,此时需要 重新进行标定,以提高绝对定位精度。

水平多关节机械臂(SCARA四轴):正常情况下,机械臂以左右手方向切换至同一点,切换前后J2轴关节坐标为轴对称。若切换前后J2轴关节坐标变为非对称,则导致机械臂无法以左右手方向到达同一点,绝对定位精度降低。此时需通过补充J2轴关节角度使左右手切换前后关节坐标变为对称,以提高绝对定位精度。

本节以水平多关节机械臂(SCARA四轴)为例说明如何进行标定。

# 前提条件

- 机械臂已上电。
- 机械臂末端已安装标定工装。

#### 操作步骤



步骤1 将DobotSCStudio切换至手动模式。

步骤2 单击"3>配置>机器人设置>标定"。

进入标定界面,如图 1.24所示。

0 <sup>0</sup> 回零	左手	「右手
🕞 标定		
末端负载		
	第一步: 使用左手姿态 移动机械臂至固定位置。	第二步: 使用右手姿态 移动机械臂至固定位置。
	第一个点: 0.0000 章	第二个点: 0.000 章 (

图 1.24 标定界面

步骤3 使机械臂处于使能状态。

□□说明

步骤 4~步骤 9均需在使能状态下执行。

步骤4 点动机械臂,使机械臂以左手方向移动至标定盘上的一点。

步骤5 在"左手"区域单击"获取P1",此时显示左手方向J2轴的坐标。

步骤6 点动机械臂使其抬升一定高度。

步骤7 点动机械臂使其以右手方向移动至步骤4中相同的位置,如图1.25 所示。





图 1.25 左右手标定

步骤8 在"右手"区域单击"获取P2",此时显示右手方向J2轴的坐标。

步骤9 单击"标定"。

如果标定成功,消息窗口会显示"标定成功!"。

# 1.2.6 虚拟仿真

用户点动或运行机械臂时,可通过虚拟仿真界面实时查看机械臂运动状况。

#### 1.2.7 日志

用户可以通过查看日志了解机械臂历史操作。可根据用户操作、控制器错误、伺服错误 三类日志进行日志的筛选。点击"重置"可清除日志。



图 1.26 日志

文档版本 V1.0(2020-05-20) 用户	<sup>白</sup> 手册 版权所有	ĩ©	越疆科技有限公司
--------------------------	----------------------	----	----------



# 1.2.8 语言切换

单击"配置>基础配置>语言"进入语言切换界面,用户可切换为中文或者英文两种语言。

## 1.2.9 末端负载

末端负载参数一般无需设置,此处不做多说明。

#### 1.2.10 网络配置

Dobot机器人控制系统可通过Ethernet接口与外部设备通信,支持TCP、UDP或Modbus协议。其默认IP地址为192.168.5.1。在实际应用中,如果采用TCP或UDP协议,则Dobot机器人控制系统可作为客户端或服务端,如视觉场景;如果采用Modbus协议,则Dobot机器人控制系统作为Modbus从站,外部设备作为Modbus主站。

用户可在"<sup>1</sup>>配置 > 网络配置"页面修改IP地址,如图 1.27所示。其中,Dobot机器 人控制系统的IP地址必须与外部设备的IP地址在同一网段,且不冲突。

● 手动设 使用下面的Ⅱ	置IP地址 P地址	£:			
IP地址:	192		168	5	1
子网掩码:	255		255	255	0
默认网关:	192		168	5	1

#### 图 1.27 IP 地址设置

Dobot机器人控制系统与外部设备连接如图 1.28和图 1.29所示。

- 如果Dobot机器人控制系统与外部设备直接连接或通过交换机连接,则需勾选"手动设置IP地址",修改IP地址、子网掩码以及默认网关后单击"保存"。
- 如果Dobot机器人控制系统与外部设备通过路由器,则勾选"自动获得IP地址"自动分配IP地址,然后单击"保存"。

<u>小</u>注意

通过路由器连接时请勿将网线接入"WAN"接口。

文档版本 V1.0(2020-05-20)

版权所有 © 越疆科技有限公司





图 1.28 Dobot 机器人控制系统与外部设备直连



图 1.29 Dobot 机器人控制系统与外部设备通过路由器或交换机连接

# 1.3 I/O 监控

单击" => 配置>IO监控"进入I/O监控页面如图 1.30所示。



_IO输出				_IO输入———			
01:0	02:0	03:0	04:0	01:0	02:0	03:0	04:0
05:0	06:0	07:0	08:0	05:0	06:0	07:0	08:0
09:0	10:0	11:0	12:0	09:0	10:0	11:0	12:0
				13:0	14:0	15:0	16:0
13:0	14:0	15:0	16:0	17:0	18:0	19:0	20:0
-继电器输出 01:0	02	2:0	03:0	21:0	22:0	23:0	24:0
04:0	05	5:0			重	置	

图 1.30 I/O 监控页面

I/O监控具有两个功能:输出和监控。

- 输出:在手动模式下可以设置输出的状态。
- 监控:可以查看输入、输出的真实状态,且在自动模式下安全保护,仅支持查看输入输出状态,不能修改。

# 1.4 远程控制

外部设备可以通过不同的远程控制模式如远程I/O模式、远程Modubs模式下发指令控制 机械臂(即对示教好的程序文件进行控制运行)。当需要设置远程控制模式时,需通过 DobotSCStudio设置,且机械臂处于下使能状态,否则会触发报警。

- 远程控制模式切换不需重新启动机器人控制系统。
- 无论机器人控制系统处于哪种模式,硬件上的急停开关始终有效。
- 若机械臂在远程控制模式下运行,此时不能切换为其他远程控制模式,需退 出当前远程模式后再切换,即先停止运行再切换。
- 若机械臂在上使能状态,则不能触发远程控制,会触发报警。请在下使能状态下触发远程控制。

# 1.4.1 远程 I/O

当远程控制模式为远程I/O时,外部设备可通过远程I/O来控制机器人,如工位预约应用场景。Dobot机器人控制系统特定的I/O接口定义如表 1.2所示。

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司
-----------------------	------	-----------------



# 表 1.2 特定 I/O 接口定义说明

I/O接口	说明
输入(用于外部控制)	
DI 11	清除报警
DI 12	继续运行
DI 13	暂停运行,但不退出远程I/O模式
DI 14	停止运行,退出远程I/O模式
DI 15	进入远程I/O模式,开始运行
DI 16	急停,退出远程I/O模式
输出(用于显示状态)	
DO 13	准备好状态
DO 14	暂停状态
DO 15	报警状态
DO 16	运行状态

<u>▲</u>注意

所有输入信号为沿触发,低电平变为高电平有效。

# 前提条件

- 己准备好远程运行的工程文件,且能正确运行。
- 已通过I/O接口连接外部设备,特定I/O接口说明如表 1.2所示。
- 机械臂已上电。

□□说明

本节不对如何连接外部设备以及如何使用外部设备进行详细说明。

# 操作步骤

步骤1 单击" ड→2 同置>远程控制"。

进入远程控制页面,如图 1.31所示。



)		- Modbus				_程
10 输入 ———		_ coils			- 現名・ Leece	
清除报警: 11	Low	清除报警:	5 Low		-1±-11.  00000	
开始: 15	Low 🔻	开始: 「	0 Low	-	22222	
暂停: 13	Low 🔻	暂停:	1 Low		44444 5555	
恢复: 12	Low 🔻	恢复: 「	2 Low	<b>_</b>	5666 DDDD	
停止: 14	Low 🔻	停止: 「	3 Low		DefaultPro	
强制停止: 16	Low 🔻	强制停止:	4 Low	<b>_</b>	2	
10 输出———		_ inBits			3	
准备: 13	Low 🔻	准备: 「	1 Low	-	_4 5	
暂停: 14	Low 🔻	暂停:	2 Low	<b>_</b>	6 7	
报警: 15	Low 🔻	报警: 「	4 Low	-	-' aaaaa	
运行: 16	Low 🔻	运行: 「	3 Low	<b>_</b>	ae allioint	

图 1.31 远程控制页面

**步骤 2** 在"远程控制模式"区域勾选"IO",并选择待远程运行的工程,并 单击"保存"。

弹出"保存成功,控制模式为IO"窗口。

此时,除了急停按钮,其他按键无效。

**步骤3** 在外部设备上触发启动信号,此时机械臂会按照选择的工程文件进行运行。

若触发停止信号,则退出远程I/O模式。

# 1.4.2 远程 Modbus

当远程控制模式为远程Modbus时,外部设备可通过远程Modbus来控制机器人,如工位预约应用场景。Modbus寄存器详细说明请参见2.15.1 Modbus寄存器说明。其特定功能如表 1.3所示。

寄存器(以PLC为例)	寄存器(Dobot机器人控制系统)	说明
线圈寄存器		
00001	0	进入远程Modbus模式,开始运行
00002	1	暂停运行,但不退出远程Modbus 模式
00003	2	继续运行
00004	3	停止运行,退出远程Modbus模式

表	1.3	Modbus 寄存器特定功能说明	Modbus	
---	-----	------------------	--------	--

文档版本 V1.0 (2020-05-20)

版权所有 © 越疆科技有限公司



寄存器(以PLC为例)	寄存器(Dobot机器人控制系统)	说明	
00005	4	急停,退出远程Modbus模式	
00006	5	清除报警	
触点寄存器			
10001	0	自动退出	
10002	1	准备好状态	
10003	2	暂停状态	
10004	3	运行状态	
10005	4	报警状态	

# 前提条件

- 已准备好远程运行的工程文件,且能正确运行。
- 已通过Ethernet接口将外部设备与机械臂连接。可直接连接或通过路由器连接,请 根据实际情况选择。

其中,Dobot机器人控制系统和外部设备的IP地址需在同一网段。Dobot机器人控制系统的IP地址可在"配置 > 网络配置"中查询并修改;端口为502,不能修改。

• 机械臂已上电。

□□说明

本节不对如何连接外部设备以及如何使用外部设备进行详细说明。

# 操作步骤

步骤1 单击" >配置>远程控制"。

进入远程控制页面,如图 1.32所示。



• IO						(	🛾 Modbus	
IO —				, [	Modbus			选择脱机的工程
清除报警:	11	Low	$\overline{\nabla}$		清除报警:	5	Low 🔻	工程名: cccc
开始: 「	15	Low	$\overline{\nabla}$		开始: 【	0	Low 🔻	22222
暂停:	13	Low	$\overline{\nabla}$		暂停:	1	Low 🔻	5555
恢复: 「	12	Low	Ŧ		恢复: 【	2	Low 🔻	6666 DDDDD
停止: 「	14	Low	$\overline{\mathbf{v}}$		停止:	3	Low 🔻	DefaultPro
强制停止:	16	Low	~		强制停止:	4	Low 🔻	
┌ IO 输出				1	_ inBits			_3
准备:	13	Low	$\overline{\mathbf{v}}$		准备:	1	Low 🔻	5
暂停: 「	14	Low	Ŧ		暂停:	2	Low 🔻	_6 _7
报警: 「	15	Low	Ŧ		报警:	4	Low 🔻	
运行: 「	16	Low	Ŧ		运行:	3	Low 🔻	allioint
								加载 保存

图 1.32 远程控制页面

**步骤 2** 在"远程控制模式"区域勾选"Modbus",并选择待远程运行的工程, 并单击"保存"。

弹出"保存成功,控制模式为Modbus"窗口。

此时,除了急停按钮,其他按键无效。

**步骤3** 在外部设备上触发启动信号,此时机械臂会按照选择的工程文件进行运行。

若触发停止信号,则退出远程Modbus模式。

# 1.5 编程

#### 1.5.1 **工程说明**

SC系列控制柜编程以工程形式来管理,包含了存点信息、全局变量、程序文件,工程结构如图 1.33所示。



图 1.33 工程结构

# 1.5.2 编程面板说明

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司
-----------------------	------	-----------------



用户编写程序时,需将DobotSCStudio切换至手动模式,其编程面板如图 1.34所示,详 细说明如表 1.4所示。



图 1.34 编程面板

表 1.4 编程面板说明

编号	说明
1	工程结构文件
	• 示教点:示教存点,详细请参见1.5.3.2 示教存 点
	<ul> <li>全局变量:定义全局函数、全局点位、全局变量</li> </ul>
	<ul> <li>线程:多任务代码文件。任务个数与新建工程</li> <li>时设置的"线程数"有关,最多可同时执行5</li> <li>个任务</li> </ul>
2	编程常用按钮,详细说明请参见表 1.5
3	程序窗口
4	程序运行按钮,详细说明请参见表 1.7
5	调试结果显示

编程常用按钮说明如表 1.5所示。

文档版本 V1.0 (2020-05-20)



# 表 1.5 编程常用按钮说明

图标	说明
	保存已编写的程序
◆	撤销编写的代码
►	重做已编写的程序
	复制选中的代码
×	剪切选中的代码
	粘贴已选中的代码
F <sup>*</sup>	常用API指令库,详细说明请参见2 编程语言
<b>()</b>	用于代码注释
প <b>্</b>	常用运算指令和过程控制指令,详细说明请参见2 <i>编程语言</i>

# 1.5.3 编程说明

本节以水平多关节机械臂(SCARA四轴)为例进行编程说明,其编程流程如图 1.35所示。







# 1.5.3.1 创建工程

# 前提条件

- 机械臂已上电。
- DobotSCStudio已切换至手动模式。

# 操作步骤

	<u></u>	٢		
步骤1	単击して	)进入编程页面,	如图	1.36所示。



图 1.36 编程页面

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司


**步骤 2** 单击 . 进入创建工程的页面,在创建工程页面输入工程名。也可选择工程模版。单击"确认",编程页面会自动生成工程文件。

🧿 Dialog					?	×
(选择—	个模板 P Empty	P Array	DynamicTracking	ElapsedTime	P MoveJ	
	Pallet	TCPClient	TCPServer	UDPClient	UDPServer	
Standa	ard project ter	nplate				
名称	My project				*.prj	~
					确认取	肖

图 1.37 创建工程

步骤3 根据实际需求创建多个任务,如图 1.38所示。单击"线程"右键选择"新建线程",最多支持创建5个线程。



图 1.38 创建工程



步骤4 (可选)导入已有工程的存点。

若用户需复用己有工程中的存点信息,可在创建工程页面右键单击"点数据",单击"导入点数据",然后选中工程后单击确认,如图 1.39所示。

打开	_		×
20200403			- 1
AP0059			- 1
Array			- 1
Array1			- 1
DefaultPro			
DynamicTracking			
DynamicTracking1			
DynamicTracking2			
DynamicTracking23			
ElapsedTime			
ElapsedTime3			
Empty			
Empty2			
MoveJ			
MoveJ1			
		确认	取消

图 1.39 导入已有工程的存点信息

## 1.5.3.2 示教存点

前提条件

- 已创建工程或已导入工程。
- DobotSCStudio已切换至手动模式。

#### 操作步骤

创建工程后需在"示教点"页面进行示教存点,以便在编写程序时调用。若在创建工程 时已导入已有工程的存点信息,则可跳过该操作,编写程序时直接调用存点即可。

步骤1 使能机械臂。

步骤2 单击Jog按键,将机械臂移动至一点。

步骤3 双击"点数据",在"点数据"页面单击 → 新增存点信息。

"点数据"页面会显示存点信息,如图 1.40所示。

其中,"Arm"为机械臂的方向。"Tool"为工具坐标系,"User"为用户坐标系。



点	数据 🗵 全局	設置 区 sr	cO 🔣 srci	×					
ſ	<b>3</b>		Ð - O	★ ▼					
	No.	х	Y	Z	R	Arm	Tool	User	Load
1	P1	400.0000	0.0000	0.0000	0.0000	Right 🝷	No.0 -	No.0 -	No.0 -

## 图 1.40 四轴存点信息

#### 表 1.6 示教按钮说明

按钮	说明
€	添加示教点
-0	删除示教点
	覆盖点。选中示教点,点动机械臂至一点后,单击该图标可覆盖选中的示教点
<b>-</b> ?*	运行至示教点。勾选示教点,长按该按钮可使机械臂运行到勾选的示教点位置
R	保存示教点
	上一页
₹	下一页

若用户需修改存点信息,则双击对应项修改即可,如图 1.41所示。

٠



	2	$\Theta^{-} $						
0.	x	Y	z	R	Arm	Tool	User	Load
40	0.0000	1.0000	120.0000	0.0000	Right -	No.0 ~	No.0 -	No.0
	40	400.0000	400.0000 1.0000	400.0000 1.0000 120.0000	400.0000 1.0000 120.0000 0.0000	400.0000 1.0000 120.0000 0.0000 Right *	400.0000 1.0000 120.0000 0.0000 Right Vo.0 V	400.0000 1.0000 120.0000 0.0000 Right Vo.0 Vo.0 Vo.0 V

### 图 1.41 修改存点信息

• 若用户需覆盖当前点,则选中当前点后,单击

步骤4 重复步骤 2和步骤 3新增存点。

步骤 5 单击 , 保存新增的存点。

□□说明

如果机械臂为垂直多关节机械臂(六轴),则存点页面如图 1.42所示。其中,R、D、N、Cfg表示机械臂相关参数。

F		* I		€ 🗠	Θ 1	•	F										
	No.	х	Y	Z	Rx	Ry	Rz	F		[	)	ſ	N	Cfg	Tool	User	Load
1	P1	400.00	1.0000	120.00	0.0000	0.0000	0.0000	-1	*	-1	Ŧ	-1	~	0	No.0 🔻	No.0 🔻	No.0 🔻
2	P2	-8.4353	105.02	-136.0	164.77	0.0000	0.0000	-1	-	-1	Ŧ	-1	Ŧ	0	No.0 🔻	No.0 🔻	No.0 -

#### 图 1.42 六轴存点页面

#### 1.5.3.3 编写程序

### 前提条件



- 己创建工程或导入工程。
- 已有存点信息。

#### 操作步骤

SC系列控制柜采用Lua语言编程,封装了常用的运动指令以及基本的运算和逻辑指令,详细请参见2 编程语言。

假设已存示教点 "P1"和 "P2"。我们在线程 "src0"页签编写一个简单的程序示例指导 用户如何调用指令,其程序如图 1.43所示。该示例通过关节插补模式使机械臂在 "P1"和 "P2"点之间循环运动。

点数排	R 🖸 全局変量 🗵 src0 🗵 src1 🗵
	▶ 🗲 🛠 🐔 🖬 💬 Ft 😋
1	while ( true )
2	do
3	Go (P1)
4	Go (P2)
5	end
6	
7	
8	
9	
_	

图 1.43 Lua 程序

步骤1 单击" ♥ >Syntax",在语法界面双击击 <sup>while</sup> 调用循环指令,并将循环条件 设置为"true"。



图 1.44 while 指令



- 步骤2 在while循环do...end中添加运动指令。
- 1. 单击 "**F**求 <sub>> MOVE</sub>"。

显示运动指令列表,如图 1.45所示。

<b>F</b> ≵ <b>℃</b>	E
	+ 10
	- MOVE
	СР
	Arch
	Speed
	LimZ
	SpeedS
	Accel
	AccelS
	Go
	MoveJ
	Move
	Arc3
	Jump
	Circle3
	GetPose
	GetAngle

图 1.45 运动指令列表

2. 在"MOVE"中选择一条运动指令。

此时会弹出该运动指令的参数设置页面。

以Go指令为例,我们可以设置该指令所要运行的点。

3. 在Go指令的参数设置页面的"第一个参数"区域选择"P1",单击"插入"。 表示通过Go模式运动至P1点。

Go 第一个参数 point: P1 ▼	
- 第二个参数 □ OPTION:	
	插入取消

图 1.46 调用 Go 指令

如果用户想指定运动速度、臂方向,则可在Go指令的参数设置页面的"第二个参数"区域进行设置,如图 1.47所示



ARM =	Left 💌	✔ Speed=	50		
Vser=	0 -	🖌 Accel=	20		
🗹 Tool=	0 -	运行=	0		
✓ CP =	0	End=	0		
SYNC =	1	ZLimit=	0		
确认取消					

图 1.47 设置可选参数

4. 换行,重复执行2。

5. 在Go指令的参数设置页面的"第一个参数"区域选择"P2",单击"插入"。 表示通过Go模式运动至P2点。

#### □□说明

如果用户需通过断点调试程序,则需在编写程序时设置断点。单击对应行编号附近的位置即可设置断点,如图 1.47所示。



图 1.40 以且例



此时,一个简单的程序已经编写完成。

- 1.5.3.4 运行程序
  - 步骤1 将机械臂切换至自动模式。

步骤2

单击🌥 🌂	使能机械臂。

此时,编程界面如图 1.49所示。

```
文档版本 V1.0(2020-05-20) 用户手册
```





	- 古教援 🛛 全局容量 🔍 srol 🖾 srol 🖾	194709 Log 性所因
<ul> <li>【】 工作空间</li> </ul>		
<ul> <li>My project</li> </ul>	1 while ( true )	+ 10
- 🖬 结程	2 do	- MOVE
are0	<b>3</b> Go (P1)	
SICO	4 Go (P2)	
🔤 src1	6	
€ 点数据	7	
	8	
	9	Accel
		Go
		MoveJ
		Move
		GetPose
		GetAngle

图 1.49 编程界面

其中,图 1.49中所示的编程运行按钮说明如表 1.7所示。

表 1.7 编程运行按钮说明

图标	说明
2	构建程序
	可检测代码是否正确
	一键运行程序
	单击后, ▶ <sub>变为</sub> <b>□</b> , <sub>程序开始运行</sub>
	如需暂停运行,则单击 即可
	开始运行程序
<b>₽</b> 3K	单击一次,表示启动调试, 🗽 变为
	单击两次,表示开始运行程序, 变为
	aik .
	如需暂停运行,则单击 即可
	停止程序运行
文档版本 V1.0(2020-05-20) 用户	Ⅰ手册 版权所有 © 越疆科技有限公司



图标	说明
۹	单步执行程序 仅当 变为 <b>1</b> , 才能开始单步执行
	监控 调试程序时可实时监控调试过程



• 如果在编写程序时设置了断点,则程序运行至断点处的前一个位置后停止。如



- 手动模式下电机使能:手动模式下单击示教界面的使能图标 (并不不定),若 (本) 变为
   ,此时可点动机械臂。
- 自动模式下电机使能:自动模式下单击示教界面的使能图标,若不变为
   ,此时可通过运行程序控制机械臂运动。

## 1.7 设置全局速率

如果用户需全局修改机械臂点动和再现的运动速度,可在DobotSCStudio界面单击 ,在操作面板以1%、5%或10%递增或递减全局速率,也可拖动滑动条修改全局速率, 如图 1.50所示。

文档版本 V1.0(2020-05-20)

用户手册



		高级
~自动模式速度比例 ———	提示:根据当前模式设置速度比例!	
-1 -5 -10	0	50% +1 +10

图 1.50 修改全局速率

点动或再现时,各轴(关节坐标系和笛卡尔坐标轴)速度和加速度计算方法如下所示:

- 各轴点动速度=各轴设置的点动最大速度\*全局速率
- 各轴点动加速度=各轴设置的点动最大加速度\*全局速率
- 各轴再现速度=各轴设置的再现最大速度\*全局速率\*指令中设置的百分比
- 各轴再现加速度=各轴设置的再现最大加速度\*全局速率\*指令中设置的百分比
- 各轴再现加加速度=各轴设置的再现最大加加速度\*全局速率\*指令中设置的百分比
   □说明
  - 各轴点动或再现的最大速度、加速度或加加速度可在"配置"界面进行设置, 详细请参见1.2.1 运动参数设置。
  - 指令设置的百分比可在编程时调用速度函数指令设置,详细请参见2.9 运动
     参数设置指令。

## 1.8 报警说明

当点动或存点的方法不正确或机械臂使用不当时,比如使机械臂限位或处于奇异点,会 触发报警。

如果机械臂运行过程中触发报警,DobotSCStudio界面的报警图标 此时可在报警界面查看报警信息,如图 1.51所示。

清除报警方法说明:

- 如果是限位报警,则DobotSCStudio切换至手动模式并使能,在关节坐标系下反向 点动限位的轴即可清除报警。
- 如果是其他报警,则DobotSCStudio切换至手动模式,然后在DobotSCStudio报警界

面单击 📠 清除报警说明。若无法清除报警,则重新启动机械臂。

文档版本 V1.0 (2020-05-20)

会变为



欢迎页面 🗵 整体速率 🗵 报警信息 🗵

	编号	类型	级别	原因	解决办法	
1	0x11	控制器错误	0	逆解算无解		
				1		
						ſ
						۱ <u> </u>

## 图 1.51 报警界面



# 2. 编程语言

Lua语言是SC系列控制柜采用的编程语言,用户可根据实际工艺需求,使用Lua语言在 DobotSCStudio上灵活的编写相应的控制程序。SC系列控制柜还封装了Dobot API指令供用户 直接调用,简化了编程。本节描述常用的编程指令供用户参考。

## 2.1 算术运算符

指令符号	说明
+	加法运算
-	减法运算
*	乘法运算
	浮点除法运算
//	向下取整除法运算
%	取余除法运算
٨	指数运算
&	与运算
	或运算
~	异或运算
<<	左移运算
>>	右移运算

#### 表 2.1 算术运算符

## 2.2 关系运算符

## 表 2.2 关系运算符

指令符号	说明
	等于
~=	不等于
<=	小于等于
>=	大于等于
<	小于
>	大于



# 2.3 逻辑运算符

表 2.3 逻辑运算符

指令符号	说明
or	逻辑或
not	逻辑非
and	逻辑与

## 2.4 一般关键字

表 2.4 一般关键字

指令符号	说明
break	跳出循环
local	定义局部变量, 仅在定义的脚本中有效
nil	变量为空值
return	函数调用返回
enter	换行

## 2.5 一般符号

表 2.5 一般符号

指令符号	说明
#	求table数组长度

## 2.6 过程控制指令

### 表 2.6 过程控制指令

指令符号	说明	
ifthenelseifthenelseend	if条件分支指令	
whiledoend	while循环控制指令	
fordoend	for循环控制指令	
repeat until()	repeat循环控制指令	

# 2.7 全局变量

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



机械臂全局变量可在"global.lua"文件中定义,包括定义全局函数、全局点位、全局变量等。

• 全局函数:

function exam()

print("This is an example")

end

- 全局点位
  - 以四轴为例:定义笛卡尔坐标点位,臂方向右手方向,工具坐标系和用户坐标 系都为0。

 $P = \{armOrientation = "right", coordinate = \{400,0,0,0\}, tool = 0, user = 0\}$ 

以六轴为例,定义关节坐标点位,臂方向R=1,D=-1,N=0,第六轴角度标识 Cfg=1,工具坐标系和用户坐标系都为0。

 $P = \{armOrientation = \{1, 1, -1, 1\}, joint = \{20, 10, 22, 2.14, 0.87, 3.85\}, tool = 0, user = 0\}$ 

全局变量

•

flag = 0

•

## 2.8 运动指令

表	2.7	运动指令
表	2.7	运动指令

指令符号	说明
Go	从当前位置以点到点方式运动至笛卡尔坐标系下 的目标位置
MoveJ	从当前位置以点到点方式运动至目标关节角度
Move	从当前位置以直线方式运动至笛卡尔坐标系下的 目标位置
Arc3	从当前位置以圆弧插补方式移动至笛卡尔坐标系 下的目标位置
Jump	<ul> <li>如果机械臂为水平多关节机械臂(SCARA四轴),从当前位置以Go运动方式进行门型运动,使机械臂移动至目标位置</li> <li>如果机械臂为垂直多关节机械臂(六轴),从当前位置以Move运动方式进行门型运动,使机械臂移动至目标位置</li> </ul>
Circle3	从当前位置以整圆插补方式移动至笛卡尔坐标系 下的目标位置
RP	笛卡尔坐标系下增加X、Y、Z方向上的偏移量并返 回一个新的坐标点



指令符号	说明
RJ	关节坐标系下增加每个轴的偏移量,并返回一个新 的坐标点
MoveR	从当前位置以直线方式运动至笛卡尔坐标系下的 偏移位置
GoR	从当前位置以点到点方式运动至笛卡尔坐标系下 的偏移位置
MoveJR	从当前位置以点到点方式运动至关节偏移位置

<u>▲</u>注意

所有指令中的可选参数在编程时可根据实际情况进行设置。如果设置,则只对当 前指令有效。

原型	Go(P,"ARM= <i>Left</i> User=1 Tool=2 CP=1 Speed=50 Accel=20 SYNC=1 ")
描述	从当前位置以点到点方式运动至笛卡尔坐标系下的目标位置
参数	必选参数: P,表示目标点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡 尔坐标点位 可洗参数.
	<ul> <li>ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或 Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效</li> <li>User,表示田户坐标系,取值范围,0~9</li> </ul>
	<ul> <li>Tool: 表示工具坐标系,取值范围: 0~9</li> </ul>
	• CP: 运动时设置平滑过渡,取值范围: 0~100
	• Speed: 运动速度比例,取值范围: 1~100
	• Accel: 运动加速度比例,取值范围: 1~100
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回
示例	1、 机械臂以默认设置运动至P1点
	Go(P1)
	2、 水平多关节机械臂(SCARA四轴)以右手方向,50%的速度,50%的加速度运动至P1 点
	Go(P1," ARM=Right Speed=50 Accel=50")
	3、 自定义笛卡尔坐标点位P1,水平多关节机械臂(SCARA四轴)以右手方向运动至P1 点

表 2.8 Go 指令



local P1 = {armOrientation = "right", coordinate={20,15,52,0}}
Go(P1)

### 表 2.9 MoveJ 指令

原型	MoveJ(P,"CP=1 Speed=50 Accel=20 SYNC=1")
描述	从当前位置以点到点方式运动至目标关节角度
参数	必选参数: P, 表示目标关节角度, 不能通过"示教点"页面导入, 需先定义关节点位
	可选参数:
	• CP: 运动时设置平滑过渡,取值范围: 0~100
	• Speed: 运动速度比例,取值范围: 1~100
	• Accel: 运动加速度比例,取值范围: 1~100
	• SYNC: 同步标识, 取值范围: 0或1。SYNC=0表示异步执行, 调用后立即返回, 但
	不关注指令执行情况;SYNC=1表示同步执行,调用后,待指令执行完才返回
示例	local $P = {joint={0,-0.0674194,0,0}}$
	MoveJ(P)
	自定义关节坐标点点位P,水平多关节机械臂(SCARA四轴)以默认设置运动至P点

表 2.10 Move 指令

原型	Move(P,"ARM= <i>Left</i> User=1 Tool=2 CP=1 SpeedS=50 AccelS=20 SYNC=1")
描述	从当前位置以直线方式运动至笛卡尔坐标系下的目标位置
参数	必选参数: P,表示目标点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡 尔坐标点位 可洗参数:
	• ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或 Right。若机械臂为垂直多关节机械臂(六轴),该参数无效
	• User: 表示用户坐标系, 取值范围: 0~9
	• Tool: 表示工具坐标系,取值范围: 0~9
	• CP: 运动时设置平滑过渡,取值范围: 0~100
	• SpeedS: 运动速度比例,取值范围: 1~100
	• AccelS: 运动加速度比例,取值范围: 1~100
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回
示例	1、 机械臂以默认设置运动至P1点
	Move(P1)
	2、 水平多关节机械臂(SCARA四轴)以左手方向,50%的速度,20%的加速度运动至P1

文档版本 V1.0(2020-05-20)

版权所有 © 越疆科技有限公司



点
Move(P1," ARM=Left SpeedS=50 AccelS=20")
3、 自定义笛卡尔坐标点位P1,水平多关节机械臂(SCARA四轴)以默认设置运动至P1 点
local P1 = {coordinate={20,15,52,0}} Move(P1)

原型	Arc3(P1,P2, "ARM= <i>Left</i> User=1 Tool=2 CP=1 SpeedS=50 AccelS=20 SYNC=1")
描述	从当前位置以圆弧插补方式移动至笛卡尔坐标系下的目标位置
	该指令需结合其他运动指令确定圆弧起始点
参数	必选参数:
	• P1,表示圆弧中间点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡尔 坐标点位
	• P2: 表示圆弧结束点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡尔 坐标点位
	可选参数:
	• ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或 Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效
	• User: 表示用户坐标系, 取值范围: 0~9
	• Tool: 表示工具坐标系, 取值范围: 0~9
	• CP: 运动时设置平滑过渡,取值范围: 0~100
	• SpeedS: 运动速度比例,取值范围: 1~100
	• AccelS: 运动加速度比例, 取值范围: 1~100
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回
示例	While true do
	Go(P1)
	Arc3(P2,P3)
	end
	机械臂循环从P1点以圆弧的方式经过P2点运动至P3点

### 表 2.12 Jump 指令

原型	Jump(P,"ARM= <i>Left</i> User=1 Tool=2 Speed=50 Accel=20 Arch=1 SYNC=1")
	Jump(P,"ARM=Left User=1 Tool=2 Speed=50 Accel=20 Start=10 ZLimit=80 End=50 SYNC=1")



描述	• 如果机械臂为水平多关节机械臂(SCARA四轴),从当前位置以Go运动方式进行门型运动,使机械臂移动至目标位置
	• 如果机械臂为垂直多关节机械臂(六轴),从当前位置以Move运动方式进行门型运动,使机械臂移动至目标位置
参数	必选参数: P,表示目标点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡 尔坐标点位。且P点高度不能高于ZLimit,否则会触发JUMP参数错误报警
	可选参数:
	• ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或 Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效
	• User: 表示用户坐标系, 取值范围: 0~9
	• Tool: 表示工具坐标系,取值范围: 0~9
	• Speed: 运动速度比例,取值范围: 1~100
	• Accel: 运动加速度比例,取值范围: 1~100
	• Arch: 设置Jump门型参数编号,取值范围: 0~9
	• Start: 起始点高度
	• ZLimit: 最大抬升高度
	• End: 结束点下降高度
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回
示例	Jump(P1): 机械臂以Jump模式运动至P1点

起始点和目标点的高度均不能大于ZLimit,否则会触发JUMP参数错误报警。

原型	Circle3(P1,P2, Count, "User=1 Tool=2 CP=1 SpeedS=50 AccelS=20 SYNC=1")	
描述	从当前位置以整圆插补方式移动至笛卡尔坐标系下的目标位置	
	该指令需结合其他运动指令确定圆弧起始点	
参数	必选参数:	
	• P1,表示整圆中间点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡尔 坐标点位	
	• P2: 表示整圆结束点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡尔 坐标点位	
	• Count: 整圆个数,取值范围: 1~999	

表 2.13 Circle3 指令

文档版本 V1.0 (2020-05-20)

版权所有 © 越疆科技有限公司

٦



	可选参数:		
	• ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或		
	Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效		
	• User: 表示用户坐标系, 取值范围: 0~9		
	• Tool: 表示工具坐标系,取值范围: 0~9		
	• CP:运动时设置平滑过渡,取值范围: 0~100		
	• SpeedS: 运动速度比例,取值范围: 1~100		
	• AccelS: 运动加速度比例,取值范围: 1~100		
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但		
	不关注指令执行情况;SYNC=1表示同步执行,调用后,待指令执行完才返回		
示例	Go(P1)		
	Circle3(P2,P3,1)		
	机械臂从P1点以整圆的方式经过P2点运动至P3点		

表 2.14 RP 指令

原型	RP(P1, {OffsetX, OffsetY, OffsetZ})	
描述	笛卡尔坐标系下增加X、Y、Z方向上的偏移量并返回一个新的笛卡尔坐标点	
	除MoveJ外,其他运动指令均支持运行至该点	
参数	<ul> <li>P1,表示当前笛卡尔坐标点,可从"示教点"页面获取,也可自定义点位,但只支持 笛卡尔坐标点位</li> <li>OffsetX,OffsetY,OffsetZ:笛卡尔坐标系下X轴、Y轴、Z轴方向上的偏移 单位:毫米</li> </ul>	
返回	笛卡尔坐标点	
示例	P2=RP(P1, {50,10,32})	
	Move(P2)或 Move(RP(P1, {50,10,32}))	

表 2.15 RP 指令

原型	RJ(P1, {Offset1, Offset2, Offset3, Offset4, Offset5, Offset6})	
描述	关节坐标系下增加每个轴的偏移量,并返回一个新的关节坐标点	
	仅支持MoveJ运动指令运行至该点	
参数	• P1,表示当前关节点位,不能通过"示教点"页面导入,需先定义	
	<ul> <li>Offset1~Offset6:关节坐标系下J1轴 ~ J6轴方向上的偏移。若机械臂为水平多关节机 械臂(SCARA四轴),则Offset5、Offset6无效</li> <li>单位:度</li> </ul>	
返回	关节坐标点	



示例	以水平多关节机械臂为例:
	local P1 = $\{joint=\{0,-0.0674194,0,0\}\}$
	P2=RJ(P1, {60,50,32,30})
	MoveJ(P2)或 MoveJ(RJ(P1, {60,50,32,30}))

表 2.16	GoR	指令
--------	-----	----

原型	GoR({OffsetX, OffsetY, OffsetZ},"ARM= <i>Left</i> User=1 Tool=2 CP=1 Speed=50 Accel=20 SYNC=1 ")		
描述	从当前位置以点到点方式运动至笛卡尔坐标系下的偏移位置		
参数	<ul> <li>必选参数: OffsetX, OffsetY, OffsetZ: 笛卡尔坐标系下X轴、Y轴、Z轴方向上的偏移 单位:毫米</li> <li>可选参数:</li> <li>ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或 Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效</li> <li>User: 表示用户坐标系,取值范围: 0~9</li> <li>Tool: 表示工具坐标系,取值范围: 0~9</li> <li>Speed: 运动速度比例,取值范围: 1~100</li> <li>Accel: 运动加速度比例,取值范围: 1~100</li> <li>SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况: SYNC = 1表示同步执行,调用后立即返回,但</li> </ul>		
示例	Go(P1) GoR({10,10,10},"Accel=100 Speed=100 CP=100")		

表 2.17 MoveJR 指令

原型	MoveJR({Offset1, Offset2, Offset3, Offset4, Offset5, Offset6}," CP=1 Speed=50 Accel=20 SYNC=1")		
描述	从当前位置以点到点方式运动至关节偏移角度		
参数	<ul> <li>必选参数: Offset1~Offset6: 关节坐标系下J1轴 ~ J6轴方向上的偏移。若机械臂为水平多 关节机械臂(SCARA四轴),则Offset5、Offset6无效</li> <li>单位: 度</li> <li>可选参数:</li> <li>CP: 运动时设置平滑过渡,取值范围: 0~100</li> <li>Speed: 运动速度比例,取值范围: 1~100</li> <li>Accel: 运动加速度比例,取值范围: 1~100</li> <li>SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况: SYNC = 1表示同步执行,调用后,待指令执行完才返回</li> </ul>		
 文档版本	L 版权所有 © 越疆科技有限公司		



原型	MoveJR({Offset1, Offset2, Offset3, Offset4, Offset5, Offset6}," CP=1 Speed=50 Accel=20 SYNC=1")
示例	Go(P1) MoveJR({20,20,10,0},"SYNC=1")

表	2.18	Move 指令

原型	MoveR({OffsetX, OffsetY, OffsetZ},"ARM= <i>Left</i> User=1 Tool=2 CP=1 SpeedS=50 AccelS=20 SYNC=1")	
描述	从当前位置以直线方式运动至笛卡尔坐标系下偏移位置	
参数	必选参数:必选参数:OffsetX,OffsetY,OffsetZ: 笛卡尔坐标系下X轴、Y轴、Z轴方向上的偏移 单位:毫米	
	可选参数: <ul> <li>ARM:机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或</li> </ul>	
	Right。若机械臂为垂直多关节机械臂 (六轴), 该参数无效	
	• User: 表示用户坐标系, 取值范围: 0~9	
	• Tool: 表示工具坐标系,取值范围: 0~9	
	• SpeedS: 运动速度比例,取值范围: 1~100	
	• AccelS: 运动加速度比例,取值范围: 1~100	
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回	
示例	Go(P1)	
	MoveR({20,20,20},"AccelS=100 SpeedS=100 CP=100")	

# 2.9 运动参数设置指令

表 2.19 运动参数设置指令	表 2	2.19	运动参数设置指令
-----------------	-----	------	----------

指令符号	说明
Accel	设置Go、Jump、MoveJ、GoR、MoveJR运动加速度 比例
AccelS	设置Move、Arc3、Circle3、MoveR运动加速度比例
Speed	设置Go、Jump、MoveJ、GoR、MoveJR运动速度比例
SpeedS	设置Move、Arc3、Circle3、MoveR运动速度比例
Arch	设置Jump门型参数索引(起始点抬升高度、最大抬 升高度、结束点下降高度)
文档版本 V1.0(2020-05-20) 用户	「手册 版权所有 © 越疆科技有限公司



指令符号	说明
СР	运动时设置平滑过渡
LimZ	设置Jump模式最大抬升高度

#### 表 2.20 Accel 指令

原型	Accel(R)	
描述	设置Go、Jump、MoveJ、GoR、MoveJR运动加速度比例	
参数	R: 百分比, 取值范围1~100	
示例	Accel(50)	
	Go(P1)	
	表示机械臂以50%的加速度比例运动至P1点	

## 表 2.21 AccelS 指令

原型	AccelS(R)	
描述	设置Move、Arc3、Circle3、MoveR运动加速度比例	
参数	R: 百分比, 取值范围1~100	
示例	AccelS(20)	
	Move(P1)	
	表示机械臂以20%的加速度比例运动至P1点	

## 表 2.22 Speed 指令

原型	Speed(R)	
描述	设置Go、Jump、MoveJ、GoR、MoveJR运动速度比例	
参数	R: 百分比, 取值范围1~100	
示例	Speed(20)	
	Go(P1)	
	表示机械臂以20%的速度比例运动至P1点	

## 表 2.23 SpeedS 指令

原型	SpeedS(R)
描述	设置Move、Arc3、Circle3、MoveR运动速度比例
参数	R: 百分比, 取值范围1~100

文档版本 V1.0(2020-05-20)

版权所有 © 越疆科技有限公司



原型	SpeedS(R)
示例 SpeedS(20)	
	Move(P1)
	表示机械臂以20%的速度比例运动至P1点

## 表 2.24 Arch 指令

原型	Arch(Index)	
描述	设置Jump门型参数索引(起始点抬升高度、最大抬升高度、结束点下降高度)	
	Jump门型参数需在DobotSCStudio"配置>机器人参数>门形参数"中设置,详细请参见 1.2.1 运动参数设置	
参数	Index: 门型参数索引,取值范围: 0~9	
	需在"设置 > 弧形再现参数"勾选对应索引后才能生效	
示例	Arch(1)	
	Jump(P1)	

## 表 2.25 CP 指令

原型	CP(R)	
描述	设置平滑过渡比例,即从起始点经过中间点到达终点时,经过中间点是以直角方式过渡运 是以曲线方式过渡,如图 2.1所示	
	仅对Go、Move、Arc3、Circle3、GoR、MoveR有效	
参数	R: 平滑过渡比例。取值范围: 0~100	
	"0"表示运动时不开启平滑过渡	
示例	CP(50)	
	Move(P1)	
	Move(P2)	
	Move(P3)	
	机械臂从P1运动至P3点经过P2点时以50%平滑过渡	





表 2.26 LimZ 指令

原型	LimZ(zValue)	
描述	设置Jump模式最大抬升高度	
参数	zValue: 最大抬升高度,不能超过机械臂Z轴极限位置	
示例	LimZ(80)	
	Jump(P," Start=10 ZLimit=LimZ End=50")	

# 2.10 输入/输出指令

表 2.27 输入/输出指令

指令符号	说明
DI	读取数字输入端口状态
DO	设置数字输出端口状态(队列指令)
DOExecute	设置数字输出端口状态(立即指令)
MoveIO	机械臂在Move运动模式下运动时并行设置数字输 出端口状态
GoIO	机械臂在Go运动模式下运动时并行设置数字输出 端口状态

#### □□说明

Dobot机器人控制系统支持两种类型的指令: 立即指令与队列指令。

- 立即指令: Dobot控制器在收到指令后立即处理该指令,而不管当前控制器是 否在还在处理其他指令。
- 队列指令: Dobot控制器在收到指令后会将该指令放入控制器内部的指令队列中, Dobot控制器将顺序执行指令。

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



#### 表 2.28 数字输入指令

原型	DI(index)	
描述	读取输入端口状态	
参数	index: 数字输入索引, 取值范围: 1~16	
返回	<ul> <li>当设置了索引时,DI(index)返回对应的index的电平(ON/OFF)</li> <li>当没有参数时,返回所有index的电平并以table形式存储 假如local di=DI(),table存储形式为{num = 24 value = {0x55, 0xAA, 0x52}},则可通过 di.num,di.value[n]等来获取对应index的值</li> </ul>	
示例	if (DI(1))==ON then Move(P1) end 数字输入端口1为高电平时机械臂以直线运动方式运动至P1点	

表 2.29 数字输出指令(立即指令)

原型	DOExecute(index,ON   OFF)
描述	设置数字输出端口状态(立即指令)
参数	<ul> <li>index:数字输出索引,取值范围:1~24</li> <li>ON/OFF:数字输出端口状态,ON:高电平;OFF:低电平</li> </ul>
示例	DOExecute(1,OFF) 将数字输出端口1设置为低电平

#### 表 2.30 数字输出指令(队列指令)

原型	DO(index,ON   OFF)
描述	设置数字输出端口状态(队列指令)
参数	<ul> <li>index:数字输出索引,取值范围:1~24</li> <li>ON/OFF:数字输出端口状态,ON:高电平;OFF:低电平</li> </ul>
示例	DO(1,ON) 将数字输出端口1设置为高电平

原型	MovelO(P, {{Mode, Distance, Index, Status}}, "ARM= <i>Left</i> User=1 Tool=2 CP=1 SpeedS=50 AccelS=20 SYNC=1")
描述	机械臂在Move运动模式下运动时并行设置数字输出端口状态

表 2.31	以 Move 模式运动时并行数字输出指令	(队列指令)
--------	----------------------	--------



参数	必选参数:	
	• P,表示目标点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡尔坐标 点位	
	• Mode: 设置Distance模式。0: Distance为距离百分比; 1: Distance为离起始点或目标 点的距离	
	• Distance: 运行指定的距离	
	若Mode为0,则Distance表示起始点与目标点之间距离的百分比,取值范围: 0~100	
	若Mode为1,则Distance表示离起始点或目标点的距离	
	• 若Distance取值为正,则表示离起始点的距离	
	• 若Distance取值为负,则表示离目标点的距离。	
	• Index: 数字输出索引,取值范围: 1~24	
	• Status: 数字输出状态,取值范围: 0或1	
	可选参数:	
	• ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或	
	Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效	
	• User: 表示用户坐标系, 取值范围: 0~9	
	• Tool: 表示工具坐标系, 取值范围: 0~9	
	• CP:运动时设置平滑过渡,取值范围: 0~100	
	• Speed: 运动速度比例, 取值范围: 1~100	
	• Accel: 运动加速度比例, 取值范围: 1~100	
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回	
示例	DO(1,ON)	
	将数字输出端口1设置为高电平	

#### 表 2.32 以 Go 模式运动时并行数字输出指令(队列指令)

原型	GoIO(P, {{Mode, Distance, Index, Status}}, "ARM= <i>Left</i> User=1 Tool=2 CP=1 Speed=50 Accel=20 SYNC=1")	
描述	机械臂在Go运动模式下运动时并行设置数字输出端口状态	
参数	<ul> <li>必选参数:</li> <li>P,表示目标点,可从"示教点"页面获取,也可自定义点位,但只支持笛卡尔坐标点位</li> <li>Mode:设置Distance模式。0:Distance为距离百分比;1:Distance为离起始点或目标点的距离</li> <li>Distance:运行指定的距离 若Mode为0,则Distance表示起始点与目标点之间距离的百分比,取值范围:0~100</li> </ul>	



	若Mode为1,则Distance表示离起始点或目标点的距离
	• 若Distance取值为正,则表示离起始点的距离
	• 若Distance取值为负,则表示离目标点的距离。
	• Index: 数字输出索引,取值范围: 1~24
	• Status: 数字输出状态,取值范围: 0或1
	可选参数:
	• ARM: 机械臂方向。若机械臂为水平多关节机械臂(SCARA四轴),则设置为Left或
	Right。若机械臂为垂直多关节机械臂(六轴),则该参数无效
	• User: 表示用户坐标系, 取值范围: 0~9
	• Tool: 表示工具坐标系,取值范围: 0~9
	• CP: 运动时设置平滑过渡,取值范围: 0~100
	• Speed: 运动速度比例,取值范围: 1~100
	• Accel: 运动加速度比例,取值范围: 1~100
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但
	不关注指令执行情况;SYNC=1表示同步执行,调用后,待指令执行完才返回
示例	DO(1,ON)
	将数字输出端口1设置为高电平

# 2.11 程序管理指令

表 2.33	程序管理指令
--------	--------

指令符号	说明
Wait	运动暂停时间
Sleep	通用脚本暂停时间
Pause	程序运行暂停
ResetElapsedTime	开始计时
ElapsedTime	结束计时
System	获取当前时间

### 表 2.34 指令延时时间

原型	Wait( <i>time</i> )
描述	运动暂停时间
参数	time: 延时时间。单位: 毫秒



示例	Go(P1)
	Wait(1000)
	机械臂执行至P1点后延时1000毫秒

## 表 2.35 程序等待时间

原型	Sleep( <i>time</i> )
描述	通用脚本暂停时间
参数	time: 延时时间。单位: 毫秒
示例	while true do
	Speed(100)
	Go(P1)
	sleep(3)
	Speed(100)
	Accel(40)
	Go(P2)
	sleep(3)
	end

## 表 2.36 程序运行暂停指令

原型	Pause()		
描述	程序运行暂停指令		
	当程序运行至该指令后,程序暂停运行,此时程序运行页面中编程按钮 U 变为 U 。 若需继续运行,则单击 U 即可		
参数	无		
示例	while true		
	do		
	Go(P1)		
	Go(P2)		
	Pause()		
	Go(P3)		
	Go(P4)		
	end		
	机械臂执行至P2点后暂停运行		
文档版本	↓ V1.0(2020-05-20) 用户手册 版权所有 © 越疆科技有限公司		

文档版本 V1.0 (2020-05-20)

版权所有 © 越疆科技有限公司



原型	ResetElapsedTime()
描述	待此指令前所有指令执行完成后开始计时,需结合ElapsedTime()一起使用
	如计算脚本中某段代码执行完成所耗时间
参数	无
返回	无
示例	Go(P2, " Speed=100 Accel=100")
	ResetElapsedTime()
	for i=1,10 do
	Jump(P1, " Speed=100 Accel=100 Start=0 End=0 ZLimit=185")
	Jump(P2, " Speed=100 Accel=100 Start=0 End=0 ZLimit=185")
	end
	<pre>print (ElapsedTime())</pre>
	Sleep(1000)

#### 表 2.37 开始计时指令

#### 表 2.38 结束计时指令

原型	ElapsedTime()
描述	结束计时,返回时间差。需结合ResetElapsedTime()一起使用
参数	无
返回	返回时间差,单位:毫秒
示例	Go(P2, " Speed=100 Accel=100")
	ResetElapsedTime()
	for i=1,10 do
	Jump(P1, "Speed=100 Accel=100 Start=0 End=0 ZLimit=185")
	Jump(P2, " Speed=100 Accel=100 Start=0 End=0 ZLimit=185")
	end
	<pre>print (ElapsedTime())</pre>
	Sleep(1000)

#### 表 2.39 获取当前时间

原型	Systime()		
描述	获取当前时间		
参数	无		
文档版本 V1.0 (2020-05-20)		用户手册	版权所有 © 越疆科技有限公司



返回	返回当前时间
示例	Go(P2, " Speed=100 Accel=100")
	local time1=Systime()
	for i=1,10 do
	Jump(P1, " Speed=100 Accel=100 Start=0 End=0 ZLimit=185")
	Jump(P2, " Speed=100 Accel=100 Start=0 End=0 ZLimit=185")
	end
	local time2=Systime()
	local time = time2 - time1
	Sleep(1000)

# 2.12 获取位姿指令

原型	GetPose()	
描述	获取笛卡尔坐标系下机械臂的实时位姿	
	如果设置了用户坐标系或工具坐标系,则获取的位姿为当前坐标系下的位姿	
参数	无	
返回	机械臂当前位置的笛卡尔坐标值	
示例	local currentPose = GetPose()	
	获取当前位置	
	local liftPose = {armOrientation = left , coordinate = {currentPose.coordinate[1], currentPose.	
	coordinate[2], currentPose. coordinate[3], currentPose. coordinate[4] }, tool = currentPose.tool,	
	user = currentPose.user}	
	原地抬升	
	Go(liftPose,"Speed=100 Accel=100")	
	Go(P1)	

表 2.40 位姿指令(1)

表 2.41 位姿指令(2)

原型	GetAngle()
描述	获取关节坐标系下机械臂的实时位姿
参数	无
返回	机械臂当前位置的关节坐标值

文档版本 V1.0(2020-05-20)

版权所有 © 越疆科技有限公司



示例	local armPose
	local joint = GetAngle()
	获取当前位姿
	if joint.joint[2] > 0 then
	armPose = "right"
	else
	armPose = "left"
	end
	local liftPose = {armOrientation = armPose, joint = {joint.joint[1], joint.joint[2], joint.joint[3],
	joint.joint[4]}, tool = 0, user = 0}

## 2.13 **TCP**

原型	err, socket = TCPCreate( <i>isServer</i> , <i>IP</i> , <i>port</i> )	
描述	创建TCP网络	
	仅支持单连接	
参数	isServer: 是否创建服务器。0: 表示创建客户端: 1: 表示创建服务端	
	IP: 服务端IP地址, 需与客户端IP地址在同一网段, 且不冲突	
	port: 服务端端口	
	机械臂作为服务端时,"port"不能设置为502或8080,否则会与Modbus默认端口或流水线	
	跟踪中使用的端口冲突,导致创建TCP网络失败	
返回	err:	
	0: 创建TCP网络成功	
	1: 创建TCP网络失败	
	socket: 创建的socket对象	
示例	请参见程序 2.1和程序 2.2	

#### 表 2.42 创建 TCP 网络指令

#### 表 2.43 TCP 连接指令

原型	TCPStart(socket, timeout)
描述	TCP连接功能
参数	socket: socket对象 timeout: 等待超时时间。单位: 秒。如果为0,则一直等待连接。如果不为0,则超过设定 的时间后退出连接

文档版本 V1.0(2020-05-20)

版权所有 © 越疆科技有限公司



返回	• 0: TCP网络连接成功	
	• 1: 输入参数错误	
	• 2: socket对象不存在	
	• 3: 设置超时时间错误	
	• 4: 若机械臂作为客户端,则说明连接错误;若机械臂作为服务端,则说明接收错误	
示例	请参见程序 2.1和程序 2.2	

#### 表 2.44 TCP 接收数据指令

原型	err, Recbuf = TCPRead(socket, timeout, type)
描述	机械臂作为客户端接收来自服务端的数据 机械臂作为服务端时接收来自客户端的数据
参数	socket: socket对象 timeout: 接收超时时间,单位: 秒。若timeout为0或者不设置,则该指令为阻塞式读取, 即接收完数据后程序才往下执行;若不为0,则超过设定的timeout后,程序继续往下执行, 即不考虑数据是否接收完,继续往下执行 type: 缓存类型。若设置为空,则RecBuf缓存格式为table形式,如果设置为"string",则 RecBuf缓存字符串形式
返回	err: 0: 接收数据成功 1: 接收数据失败 Recbuf: 接收数据缓存区
示例	请参见程序 2.1和程序 2.2

## 表 2.45 TCP 发送数据指令

原型	TCPWrite(socket, buf, timeout)
描述	机械臂作为客户端时发送数据给服务端
	机械臂作为服务端时发送数据给客户端
参数	socket: socket对象
	buf: 发送的数据
	timeout: 超时时间,单位: 秒。若timeout为0或者不设置,则该指令为阻塞式读取,即发
	送完数据后程序才往下执行;若不为0,则超过设定的timeout后,程序继续往下执行,即
	个考虑数据走召接收元,继续往下执行 ————————————————————————————————————
返回	0: 发送数据成功
	1: 发送数据失败
示例	请参见程序 2.1和程序 2.2
文档版本	V1.0(2020-05-20) 用户手册 版权所有 © 越疆科技有限公司



#### 表 2.46 关闭 TCP 指令

原型	TCPDestroy(socket)
描述	关闭TCP功能
参数	socket: socket对象
返回	0: 关闭TCP成功
	1: 关闭TCP失败
示例	请参见程序 2.1和程序 2.2

⚠注意

- 当前TCP连接仅支持连接单个客户端。连接时请启动服务端后再连接客户端;
   断开连接时请先关闭客户端,以免服务端端口未及时释放导致再次连接时连接失败。
- 机械臂作为服务端时,IP地址可在"配置 > 网络配置"页面查询并修改,端口不能设置为502或8080,否则会与Modbus和流水线跟踪中的默认端口冲突,导致创建TCP网络失败。

#### 程序 2.1 TCP 服务端示例

local ip="192.168.5.1"	//机械臂的 IP 地址作为服务端的 IP 地址
local port=6001	//服务端端口
local err=0	
local socket=0	
err, socket = TCPCreate(true, ip, port)	
if $err == 0$ then	
err = TCPStart(socket, 0)	
if $err == 0$ then	
local RecBuf	
while true do	
TCPWrite(socket, "tcp server test"	") //服务端发送数据至客户端
err, RecBuf = TCPRead(socket,0,	"string") //服务端接收客户端发送的数据
if $err == 0$ then	
Go(P1)	//服务端接收到数据后开始执行运动指令
Go(P2)	
print(Recbuf)	
else	
文档版本 V1.0(2020-05-20)	用户手册 版权所有 © 越疆科技有限公司



print("Read error ".. err) break end end else print("Create failed ".. err) end TCPDestroy(socket) else print("Create failed ".. err)

end

1±/] <sup>-</sup> 2		
local ip="192.168.5.25"	//外部设备如相机的 IP 地址作为服务端的 IP 地	址
local port=6001	//服务端端口	
local err=0		
local socket=0		
err, socket = TCPCreate(false, ip, port)		
if $err == 0$ then		
err = TCPStart(socket, 0)		
if $err == 0$ then		
local RecBuf		
while true do		
TCPWrite(socket, "tcp client test	") //客户端发送数据至服务站	荷
TCPWrite(socket, {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07})		
err, RecBuf = TCPRead(socket, 0	)) //客户端接收服务端发送的	数据
if $err == 0$ then		
Go(P1)	//客户端接收到数据后开始执行运动	指令
Go(P2)		
print(Recbuf)		
else		
print("Read error " err)		
break		
end		
end		
else		
文档版本 V1.0(2020-05-20)	用户手册 版权所有 © 越疆科技有限2	公司

#### 程序 2.2 TCP 客户端示例



print("Create failed ".. err)

end

TCPDestroy(socket)

else

print("Create failed ".. err)

end

## 2.14 **UDP**

### 表 2.47 创建 UDP 网络指令

原型	err, socket = UDPCreate( <i>isServer</i> , <i>IP</i> , <i>port</i> )
描述	创建UDP网络
	仅支持单连接
参数	isServer: 是否创建服务器。0: 表示创建客户端; 1: 表示创建服务端
	IP: 服务端IP地址, 需与客户端IP地址在同一网段, 且不冲突
	port: 服务端端口
	机械臂作为服务端时,"port"不能设置为502或8080,否则会与Modbus默认端口或流水线
	跟踪中使用的端口冲突,导致创建UDP网络失败
返回	err:
	0: 创建UDP网络成功
	1: 创建UDP网络识别
	socket: 创建的socket对象
示例	请参见程序 2.3和程序 2.4

### 表 2.48 UDP 接收数据指令

原型	err, Recbuf = UDPRead(socket, timeout, type)	
描述	机械臂作为客户端接收来自服务端的数据 机械臂作为服务端时接收来自客户端的数据	
参数	socket: socket对象 timeout: 超时时间,单位:秒。若timeout为0或者不设置,则该指令为阻塞式读取,即接 收完数据后程序才往下执行;若不为0,则超过设定的timeout后,程序继续往下执行,即 不考虑数据是否接收完,继续往下执行 type:缓存类型。若设置为空,则RecBuf缓存格式为table形式,如果设置为"string",则 RecBuf缓存字符串形式	



返回	err:
	0: 接收数据成功
	1: 接收数据失败
	Recbuf: 接收数据缓存区
示例	请参见程序 2.3和程序 2.4

#### 表 2.49 UDP 发送数据指令

原型	UDPWrite(socket, buf, timeout)
描述	机械臂作为客户端时发送数据给服务端
	机械臂作为服务端时发送数据给客户端
参数	socket: socket对象 buf: 发送的数据 timeout: 超时时间,单位: 秒。若timeout为0或者不设置,则该指令为阻塞式读取,即发 送完数据后程序才往下执行;若不为0,则超过设定的timeout后,程序继续往下执行,即 不考虑数据是否发送完,继续往下执行
返回	<ul><li>0: 发送数据成功</li><li>1: 发送数据失败</li></ul>
示例	请参见程序 2.3和程序 2.4

⚠注意

- 当前UDP连接仅支持连接单个客户端。连接时请启动服务端后再连接客户端;
   断开连接时请先关闭客户端,以免服务端端口未及时释放导致再次连接时连接失败。
- 机械臂作为服务端时,IP地址可在"配置 > 网络配置"页面查询并修改,端口不能设置为502或8080,否则会与Modbus和流水线跟踪中的默认端口冲突,导致创建UDP网络失败.

### 程序 2.3 UDP 服务端示例

local ip="192.168.5.1"	//机械臂的	IP 地址作为服务端的 IP 地址
local port=6201	//服务端端口	1
local err=0		
local socket=0		
err, socket = UDPCreate(true, ip, port)		
if $err == 0$ then		
local RecBuf		
while true do		
文档版本 V1.0 (2020-05-20)	用户手册	版权所有 © 越疆科技有限公司


	UDPWrite(socket, "udp server test")	//服务端发送数据至客户端
	err, RecBuf = UDPRead(socket, 0)	//服务端接收客户端发送的数据
	if $err == 0$ then	
	Go(P1)	//服务端接收到数据后开始执行运动指令
	Go(P2)	
	print(Recbuf)	
	else	
	print("Read error " err)	
	break;	
	end	
ei	nd	
else		
p	rint("Create failed " err)	
end		

# 程序 2.4 UDP 客户端示例

local ip="192.168.1.25"		//外部设备的 IP 地址作为服务端的 IP 地址
local port=6200		//服务端端口
local err=0		
local socket=0		
err, socket = UDPCreate(false, ip, port)		
if $err == 0$ then		
local buf		
while true do		
UDPWrite(socket, "udp client test")		//客户端发送数据至服务端
UDPWrite(socket, {0x01, 0x02, 0x02	3, 0x04, 0x05, 0z	x06, 0x07})
err, RecBuf = UDPRead(socket, 0)		//客户端接收服务端发送的数据
if $err == 0$ then		
Go(P1)		//客户端接收到数据后开始执行运动指令
Go(P2)		
print(Recbuf)		
else		
print("Read error " err)		
break		
文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



	end
	end
else	
	print("Create failed " err)
end	

# 2.15 Modbus

### 2.15.1 Modbus 寄存器说明

Modbus是一种串行通信协议,通过此协议,Dobot机器人控制系统可与外部设备进行通信。其中,外部设备如PLC作为Modbus主站,Dobot机器人控制系统作为Modbus从站。

Modbus协议的数据主要分为四类:线圈状态、离散量输入、输入寄存器、保持寄存器, 基于Dobot机器人控制系统的内存空间,我们定义了线圈、触点(离散输入)、输入、保持寄 存器,用于外部设备与机器人控制系统进行数据交互。其中,每个寄存器地址为4096个。详 细说明如下所示。

• 线圈寄存器

线圈寄存器地址(PLC 为例)	线圈寄存器地址 (Dobot 机器人控制系统)	数据类型	说明
00001	0	位	启动
00002	1	位	暂停
00003	2	位	继续
00004	3	位	停止
00005	4	位	急停
00006	5	位	清除报警
00007~0999	6~998	位	系统预留
01001~04096	999~4095	位	用户自定义

#### 表 2.50 线圈寄存器地址说明

• 触点寄存器

表	2.51	触点寄存器地址说明
		MAX.( H) 13 HH - C-12 00.73

触点寄存器地址(PLC 为例)	触点寄存器地址(Dobot 机器人控制系统)	数据类型	说明
10001	0	位	自动退出



触点寄存器地址(PLC 为例)	触点寄存器地址(Dobot 机器人控制系统)	数据类型	说明
10002	1	位	准备好状态
10003	2	位	暂停状态
10004	3	位	运行状态
10005	4	位	报警状态
10006~10999	5~998	位	系统预留
11000~14096	999~4095	位	用户自定义

• 输入寄存器

### 表 2.52 输入寄存器地址说明

输入寄存器地址(PLC 为例)	输入寄存器地址(Dobot 机器人控制系统)	数据类型	说明
30001~34096	0~4095	字	系统预留

• 保持寄存器

### 表 2.53 保持寄存器地址说明

保持寄存器地址(PLC 为例)	保持寄存器地址(Dobot 机器人控制系统)	数据类型	说明
40001~41000	0~999	字	系统预留
41001~44095	1000~4095	字	用户自定义

# 2.15.2 指令说明

### 表 2.54 读线圈功能指令

原型	GetCoils(addr, count)	
描述	读取Modbus从站线圈寄存器地址的值	
参数	addr: 线圈寄存器起始地址。取值范围: 0~4095 count: 连续读取线圈寄存器的地址个数。取值范围: 0~4096 - addr	
返回	线圈寄存器地址的值,存储在table中。table中第一个值对应线圈寄存器起始地址的值 返回的数据类型:位	

文档版本 V1.0(2020-05-20)



示例	从地址0开始连续读取5个线圈的值
	Coils = GetCoils(0,5)
	返回结果:
	Coils={1,0,0,0,0}
	结合表 2.50, 表示机械臂处于启动状态

### 表 2.55 写线圈功能指令

原型	SetCoils(addr, count, table)
描述	设置Modbus从站线圈寄存器地址的值
	对于Modbus从站线圈寄存器地址0~5,该指令不可用
参数	addr: 线圈寄存器起始地址。取值范围: 6~4095
	count: 连续写入线圈寄存器的地址个数。取值范围: 0~4096 - addr
	table: 线圈寄存器地址的值。数据类型: 位
返回	无
示例	从地址1024开始,连续写5个线圈
	local Coils = $\{0,1,1,1,0\}$
	setCoils(1024, #coils, coils)

# 表 2.56 读离散输入功能指令

原型	GetInBits(addr, count)	
描述	读取Modbus从站触点寄存器地址的值	
参数	addr: 触点寄存器起始地址地址。取值范围: 0~4095 count: 连续读取触点寄存器的地址个数。取值范围: 0~4096-addr	
返回	触点寄存器地址的值,存储在table中。table中第一个值对应线圈寄存器起始地址的值 返回的数据类型:位	
示例	从地址0开始连续读取5个地址的值 inBits = GetInBits(0,5) 返回结果: inBits = {0,0,0,1,0} 结合表 2.51,表示机械臂处于运行状态	

# 表 2.57 读输入寄存器功能指令

原型 GetinRegs(addr, count, type)
---------------------------------

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



描述	按照指定的数据类型,读取Modbus从站输入寄存器地址的值		
参数	addr: 输入寄存器起始地址。取值范围: 0~4095		
	count: 读取输入寄存器的地址个数。取值范围: 0~4096-addr		
	type: 数据类型		
	• 如果为空,默认读取16位无符号整数(2个字节,占用1个寄存器)		
	• "U16":读取16位无符号整数(2个字节,占用1个寄存器)		
	• "U32":读取32位无符号整数(4个字节,占用2个寄存器)		
	• "F32":读取32位单浮点数(4个字节,占用2个寄存器)		
	• "F64":读取64位双精度浮点数(8个字节,占用4个寄存器)		
返回	输入寄存器地址的值,存储在table中。table中第一个值对应线圈寄存器起始地址的值		
示例	示例1:从地址2048开始读取一个16位无符号整数		
	data = GetInRegs(2048,1)		
	示例2:从地址2048开始读取一个32位无符号整数		
	data = GetInRegs(2048, 1, "U32")		

表 2.58 读保持寄存器功能指令

原型	GetHoldRegs(addr, count, type)		
描述	按照指定的数据类型,读取Modbus从站保持寄存器地址的值		
参数	addr: 保持寄存器的起始地址。取值范围: 0~4095		
	count: 读取保持寄存器的地址个数。取值范围: 0~4096-addr		
	type: 数据类型		
	• 如果为空,默认读取16位无符号整数(2个字节,占用1个寄存器)		
	• "U16":读取16位无符号整数(2个字节,占用1个寄存器)		
	• "U32":读取32位无符号整数(4个字节,占用2个寄存器)		
	• "F32":读取32位单精度浮点数(4个字节,占用2个寄存器)		
	• "F64":读取64位双精度浮点数(8个字节,占用4个寄存器)		
返回	保持寄存器地址的值,存储在table中。table中第一个值对应线圈寄存器起始地址的值		
示例	示例1:从地址2048开始读取一个16位无符号整数		
	data = GetHoldRegs(2048,1)		
	示例2:从地址2048开始读取一个32位无符号整数		
	data = GetInRegs(2048, 1, "U32")		

表:	2.59	写保持寄存器功能指令
----	------	------------

原型	SetHoldRegs(addr,	count, table, type)	
文档版本	V1.0 (2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



描述	设置保持寄存器地址的值		
参数	addr: 保持寄存器的起始地址。取值范围: 0~4095		
	count: 写入保持寄存器的地址个数。取值范围: 0~4096-addr		
	table: 保持寄存器地址的值		
	type: 数据类型		
	• 如果为空,默认写入16位无符号整数(2个字节,占用1个寄存器)		
	• "U16": 写入16位无符号整数(2个字节,占用1个寄存器)		
	• "U32": 写入32位无符号整数(4个字节,占用2个寄存器)		
	• "F32": 写入32位单精度浮点数(4个字节,占用2个寄存器)		
	• "F64": 写入64位双精度浮点数(8个字节,占用4个寄存器)		
返回	无		
示例	示例 1: 从地址 2048 开始,写入一个 16 位无符号整数		
local data = {6000} SetHoldRegs(2048, #data, data, "U16") 示例2:从地址2048开始,写入一个64位双精度浮点数			
			local data = $\{95.32105\}$
			SetHoldRegs(2048, #data, data, "F64")

# 2.16 ECP

ECP(External Control Point)即外部控制点,是在外部固定工具的顶端的加工点上定义 机械臂位置和方向的坐标点,如图 2.2所示。设置ECP后,机械臂会握住工件围绕ECP点按 照指定的轨迹移动,例如缝纫场景。



文档版本 V1.0(2020-05-20)

版权所有 © 越疆科技有限公司



# 表 2.60 开启外部控制点指令

原型	ECP( <i>isOn</i> )
描述	开启外部控制点功能
参数	isOn: 是否开启ECP。ON: 开启。OFF: 关闭
返回	无
示例	请参见程序 2.5

#### 表 2.61 设置外部控制点指令

原型	ECPSet(point)	
描述	设置外部控制点	
	该指令需在开启了ECP后才有效	
参数	point: 示教点	
返回	无	
示例	P1={armOrientation="left", coordinate={630,-32, 95.5,90}, tool=0, user=0}	
	ECPSet(P1)	

#### 程序 2.5 ECP 示例

```
while true do
```

```
ECPSet(P1)
ECP(ON)
```

```
Move(P2, "SpeedS=80 CP=10")
```

```
Move(P3, "SpeedS=80 CP=10")
```

```
Arc3(P4,P5, "SpeedS=80 CP=10")
```

```
Move(P6, "SpeedS=80 CP=10")
```

ECP(OFF)

end

# 2.17 工艺指令

# 2.17.1 传送带跟踪指令

### 表 2.62 设置传送带参数指令

原型	CnvVison(CnvID)		
描述	设置传送带编号,建立跟踪	际列	
参数	CnvID: 传送带编号		
文档版本 V1.0(2020-05-20)		用户手册	版权所有 © 越疆科技有限公司



返回	0:无错误 1:有错误
示例	CnvVison(1)sss
	即把编号为1的传送带信息(分辨率、起始位置、方向和边界等)下发至机器人控制系统

# 表 2.63 获取传送带工件状态指令

原型	GetCnvObject( <i>CnvID</i> , <i>ObjID</i> )
描述	从指定的传送带上获取指定工件的数据,检测是否进入抓取区域
参数	CnvID: 传送带编号
	ObjID: 工件编号
返回	工件状态: 是否检测到工件。取值范围: true 或 false 工件分类属性
	工件坐标(x,y,r)
示例	$P111 = \{0,0,0\}$
	while true do
	flag,typeObject,P111 = GetCnvObject(0,0)
	if flag == true then
	break
	end
	Sleep(20)
	end

# 表 2.64 设置偏差指令

原型	SetCnvPointOffset(xOffset,yOffset)
描述	设置传送带用户坐标系下X、Y方向的偏移
参数	xOffset: X轴方向偏移 yOffset: Y轴方向偏移 单位: 毫米
返回	<ul><li>0:无错误</li><li>1:有错误</li></ul>

表	2.65	设置时间补偿指令
---	------	----------

	原型	SetCnvTimeCompensation ( <i>time</i> )
--	----	--



描述	设置时间补偿
	该指令用于补偿拍照触发带来的时间延时导致传送带运动方向的抓取位置偏移
参数	time: 时间偏差, 单位: 毫秒
返回	<b>0:</b> 无错误
	1: 有错误

# 表 2.66 同步传送带指令

原型	SyncCnv ( <i>CnvID</i> )
描述	同步指定的传送带
	SyncCnv (CnvID)与StopSyncCnv (CnvID)之间的运动指令仅支持Move指令
参数	CnvID: 传送带编号
返回	<b>0:</b> 无错误

### 表 2.67 停止同步传送带指令

原型	StopSyncCnv ( <i>CnvID</i> )
描述	停止同步传送带
	运行完该条指令后才会继续下发执行该指令后的其他指令
参数	CnvID: 传送带编号
返回	0: 无错误
	1: 有错误

# 2.17.2 码垛指令

### 表 2.68 创建矩阵跺盘指令

原型	Pallet = MatrixPallet ( <i>index</i> , "IsUnstack= <i>true</i> Userframe= <i>1</i> ")
描述	实例化矩阵跺盘变量
参数	index: 创建的矩阵跺型编号 可选参数: IsUnstack: 跺模式。取值范围: true或false。true: 拆跺模式; false: 码垛模式。若不设置, 默认为码垛模式 Userframe: 使用的用户坐标系序号。若不设置,默认为0
返回	实例化的跺盘变量
示例	<pre>myPallet = MatrixPallet(0, "IsUnstack=tsrue Userframe=8")</pre>

文档版本 V1.0(2020-05-20)



### 表 2.69 设置下次操作跺指令

原型	SetPartIndex (Pallet, index)
描述	设置下次操作跺的序号
参数	Pallet: 实例化的跺盘变量
	index: 下次操作跺的序号。起始值: 0
返回	无
示例	local myPallet = MatrixPallet(0, "IsUnstack=true Userframe=8")
	SetPartIndex(myPallet,1)
	表示下次操作跺的序号为2

# 表 2.70 获取当前操作跺序号指令

原型	GetPartIndex (Pallet)	
描述	获取当前操作跺序号	
参数	Pallet: 实例化的跺盘变量	
返回	当前操作跺的序号	
示例	local index=GetPartIndex(myPallet)	
	假设返回值为1,表示当前操作跺的序号为2	

# 表 2.71 设置下次操作层指令

原型	SetLayerIndex (Pallet, index)	
描述	设置下次操作码垛层的序号	
参数	Pallet: 实例化的跺盘变量	
	index: 下次操作码垛层的序号。起始值: 0	
返回	无	
示例	local myPallet = MatrixPallet(1, "IsUnstack=true Userframe=8")	
	SetPartIndex(myPallet,1)	
	表示下次操作的码垛层为第二层	

### 表 2.72 获取当前操作层序号指令

原型	GetLayerIndex (Pallet)					
描述	获取当前操作层序号					
参数	Pallet: 实例化的跺盘变量					
文档版本 V1.0(2020-05-20)		用户手册	版权所有 © 越疆科技有限公司			



返回	当前操作层的序号
示例	local index=GetLayerIndex(myPallet)
	假设返回值为1,表示当前操作码垛层为第二层

# 表 2.73 复位指令

原型	Restet (Pallet)				
描述	复位,恢复初始化状态				
参数	Pallet: 实例化的跺盘变量				
返回	无				
示例	local myPallet = MatrixPallet(0, "IsUnstack=true Userframe=8")				
	Reset(myPallet)				

# 表 2.74 查询码垛或拆跺状态指令

原型	IsDone (Pallet)					
描述	查询码垛或拆跺是否完成					
参数	Pallet: 实例化的跺盘变量					
返回	true: 完成					
	false: 未完成					
示例	Result = IsDone(myPallet)					
	If (result == true)					

# 表 2.75 释放码垛实例指令

原型	Release (Pallet)			
描述	释放码垛实例			
参数	Pallet: 实例化的跺盘变量			
返回	无			
示例	Release(myPalletSS)			

# 表 2.76 Moveln 运动指令

原型	MoveIn (Pallet, "velAB=20 velBC=30 accAB=20 accBC=10 CP=20 SYNC=1")
描述	表示机械臂从当前位置按照配置的码垛路径运动至第一跺目标点

文档版本 V1.0(2020-05-20)



参数	必选参数:					
	Pallet: 实例化的跺盘变量					
	可选参数:					
	• velAB: 安全过渡点与准备点之间的运动速度比例,取值范围: 1~100					
	• velBC: 准备点与第一跺目标点之间的运动速度比例,取值范围: 1~100					
	• accAB: 安全过渡点与准备点之间的运动加速度比例,取值范围: 1~100					
	• accBC: 安全过渡点与准备点之间的运动加速度比例,取值范围: 1~100					
	• CP: 运动时设置平滑过渡,取值范围: 0~100					
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但 不关注指令执行情况; SYNC=1表示同步执行,调用后,待指令执行完才返回					
返回	无					
示例	MoveIn(myPallet, "velAB=90 velBC=50")					

表 2.77 MoveOut 运动指令

原型	MoveOut (Pallet, "velAB=20 velBC=30 accAB=20 accBC=10 CP=20 SYNC=1")				
描述	表示机械臂从当前位置按照配置的拆垛路径运动至安全过渡点				
参数	必选参数:				
	Pallet: 实例化的跺盘变量				
	可选参数:				
	• velAB: 准备点与安全过渡点之间的运动速度比例,取值范围: 1~100				
	• velBC: 第一跺目标点与准备点之间的运动速度比例,取值范围: 1~100				
	• accAB: 准备点与安全过渡点之间的运动加速度比例,取值范围: 1~100				
	• accBC: 第一跺目标点与准备点之间的运动加速度比例,取值范围: 1~100				
	• CP: 运动时设置平滑过渡,取值范围: 0~100				
	• SYNC: 同步标识,取值范围: 0或1。SYNC=0表示异步执行,调用后立即返回,但				
	不关注指令执行情况; SYNC = 1表示同步执行,调用后,待指令执行完才返回				
返回	无				
示例	MoveOut(myPallet, "velAB=90 velBC=50")				

### □□说明

码垛路径和拆跺路径分别如图 2.3和图 2.4所示。其中,A点为安全过渡点(A点), 进跺盘和出跺盘之后都会经过的一个点位,可固定或随层高变化。B点为准备点, 由目标点和设置的偏移计算出来。C点为第一垛的目标点位。





图 2.3 码垛路径



图 2.4 拆跺路径



# 3. **工艺指南**

# 3.1 传送带跟踪

### 3.1.1 概述

传送带跟踪即通过视觉或光电传感器检测到传送带上的物体,机械臂在物体移动过程中 对其进行准确的抓取或其他应用。

### 3.1.2 搭建环境

传送带跟踪通信流程图如图 3.1所示。其中,光电传感器和视觉系统根据实际的检测方 式选择。如果采用光电传感器,则通过DI信号变化检测工件有无。如果采用视觉系统,则通 过DO触发拍照检测,上升沿有效。





一套完整的传送带跟踪工艺环境如图 3.2所示,其中视觉系统和光电传感器根据实际检测方式选择一种即可。



传送带移动方向

图 3.2 传送带跟踪工艺环境

### 3.1.2.1 编码器

编码器用于记录传送带移动距离和工件位置,通过计数器反馈给机器人控制系统,方便 机械臂进行抓取。将编码器通过高速计数器和通讯模块(统称计数器模块)连接至控制柜 EtherCAT接口,简易连接图如图 3.3所示。建议选用E6B2-CWZ1X(1000P/R)编码器和倍福 EL5101高速计数器以及倍福EK1100通讯模块。

文档	镇版本 V1.0(2020-05-20)	用户手册	版权所有 ©	越疆科技有限公司





图 3.3 编码器连接

# 3.1.2.2 光电传感器

光电传感器检测物体有无时会输出不同的电平信号,将光电传感器连接至机器人控制系统I/O模块的DI接口,Dobot机器人控制系统可通过信号边沿检测物体。

# 3.1.2.3 视觉系统

视觉系统通过TCP/IP与机器人控制系统通信,并通过I/O模块的DO接口触发拍照检测。

• 若Dobot机器人控制系统版本号为V3.0.0.20190219121343以前版本,则通过DO16接口触发拍照,简易连接如图 3.4所示。用户可在DobotSCStudio的"配置>基础配置>版本"页面查看控制器版本号。



图 3.4 视觉系统连接(1)

• 若Dobot机器人控制系统版本号为V3.0.0.20190219121343版本及以后版本,则通过 DO10接口触发拍照,简易连接如图 3.5所示。

|--|





图 3.5 视觉系统连接(2)

### 3.1.3 标定传送带

传送带跟踪物体前,需设置标系对传送带进行标定,用于描述传送带与机械臂之间的相 对关系。下述标定过程中,采用用户坐标系进行标定,并假设基坐标系下Y轴正方向与传送 带移动的方向重合。

# 前提条件

- 机械臂已上电。
- 机械臂末端已安装标定针。

### 操作步骤

步骤1 在传送带上贴上标签,如图 3.7所示。





图 3.6 传送带上贴上标签

**步骤 2** 单击"<sup>■</sup>>配置>坐标系设置>四轴用户坐标系"进入"用户坐标系" 界面,如图 1.8所示。

-			х	Y	z	Rx	
7	四轴工具坐标系	0	0.0000	0.0000	0.0000	0.0000	
							User0 x
							4轴
							第一个点
							X: 0.0000 + Y: 0.0000
							Z: 0.0000 ‡ R: 0.0000
							秋取第一下黑
							第二个点
							Х: 0.0000 🐥 Ү: 0.0000
							Z: 0.0000 + R: 0.0000
							万主 ( P在 ) ( PA
48.15							

图 3.7 用户坐标系界面

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



步骤3 使能机械臂,并点动机械臂至传送带上的标签处,在"第一个点"区域单击"获取第一个点",假设为A点,即用户坐标系的原点,如图 3.8所示。



图 3.8 传送带标定

步骤4 控制传送带移动一段距离后停止,标签也随之移动。

步骤5 使能机械臂,并点动机械臂至传送带上的标签处,在"第二个点"区 域单击"获取第二个点,假设为B点。即AB两点确定用户坐标系X轴正方向, 根据右手定则确定用户坐标系的Y轴和Z轴,如图 3.9所示。





### 图 3.9 用户坐标系标定

步骤6 单击"保存",用户坐标系参数自动生成。

假设为用户坐标系1。如果此时R轴坐标为90°,则标定成功。否则,请重新标定。

# 

机械臂与传送带之间的位置关系不同,标定后的R轴坐标也不同。

- 如果传送带移动方向为基坐标下X轴正方向,则R轴坐标为0°。
- 如果传送带移动方向为基坐标下X轴负方向,则R轴坐标为180°或-180°。
- 如果传送带移动方向为基坐标下Y轴正方向,则R轴坐标为90°。
- 如果传送带移动方向为基坐标下Y轴负方向,则R轴坐标为-90°。

### 3.1.4 配置传送带

### 前提条件

- 机械臂已上电。
- 机械臂末端已安装配套的标定针。



- 整个操作过程需在基坐标下进行,且使用配套的标定针。
- 请务必按步骤顺序进行操作,否则导致参数设置失败。

### 操作步骤

步骤1 在"工艺包"界面单击"动态跟踪"。 进入动态跟踪界面,如图 3.10所示。



传送带编号: □ _	预览	基本设置
		编码器
传送带类型: 线性	Y	传感器
用户坐标系: 0	•	
		 保存

图 3.10 动态跟踪界面

# 步骤2 设置传送带属性。

基本参数说明如表 3.1所示。

表 3.1 基本参数说明

参数	说明
传送带编号	传送带编号
	当前不支持设置
传送带类型	传送带型号
	• 直线型
	• 圆盘型
	当前仅支持直线类型,其他类型不支持
编码器通道	编码器通道
	当前不支持设置
用户坐标系	用户坐标系编号
	请根据3.1.3 标定传送带设置的用户坐标系选择编
	号
检测方式	检测方式
	• 传感器:使用传感器检测工件
	• 相机:使用视觉检测工件
	请根据实际跟踪方式选择
文档版本 V1.0(2020-05-20) 用F	□手册 版权所有 © 越疆科技有限公司





1. 在传送带上贴上标签,如图 3.11所示。



### 图 3.11 传送带上贴上标签

使能机械臂,并点动机械臂至传送带上的标签处,单击"1",如图 3.12和
 图 3.13所示。



传送带编号: 0 👻				Ť	1988〕 基本设置 〕
位素	Х	Ÿ	Z	编码器	编码器 使感器
1	0.0000	0.0000	0.0000	0.0000	
2	0.0000	0.0000	0.0000	0.0000	
it	算				
	编码器方向:	◉ 正方向			
	编码器分辨率。	0.0000	脉冲/毫米		
					<b>重置</b> 保存

图 3.12 编码器标定界面



图 3.13 编码器标定

- 3. 控制传送带移动一段距离并停止,标签也随之移动。
- 4. 使能机械臂,并点动机械臂至传送带上的标签处,单击"2"。
- 5. 单击"计算",获取编码器分辨率。

编码器分辨率即传送带移动单位长度(mm)时编码器的脉冲增量。

□□说明

如果"检测方式"为传感器,则执行步骤 4。如果"检测方式"为相机,则执行步骤 5。

文档版本	V1.0	(2020-05-20)
入归风平	V I.U	(2020-03-20)



步骤4 (可选)单击"传感器",标定传感器的起始位置,如图 3.14所示。

後送邢編号: 0 ▼	<b>预</b> 基本设置
- 传感器设置	编码器
触发输入端口: 0 去重距离; 0.0000 毫米	工作边界
触发类型: 上升沿 🔹 开始监听	
← 感器标定	
示教点位置 X 0.0000 毫米	
Y 0.00C 毫米 B 0.0000 脉冲	
计算传感器位置 X (0.0000) 毫米	
Y 0.00C 毫米 B 0.0000 脉冲	重置

图 3.14 传感器标定界面

传感器标定是为了在传感器被工件触发的瞬间,获取工件的位置,以便工件随 传送带移动时,根据坐标的偏移来计算出每一时刻工件所在的用户坐标系下的 位置。

控制传送带移动,当工件经过传感器后使工件停止在机械臂的工作空间内的某 一位置,此时移动机械臂对准该工件中心,获取当前的示教位置,同时机械臂 记录下触发传感器后传送带的移动距离,根据当前的示教位置和传送带移动距 离获取触发传感器时工件的位置,详细步骤如下所示。

1. 在传感器界面设置传感器相关参数。

参数说明如表 3.2所示。

参数	说明
传送带编号	传送带编号
	当前不支持设置
触发输入端口	当前不支持设置
触发类型	传感器被工件触发时输出的信号边沿
	<ul> <li>上升沿</li> </ul>
	<ul> <li>下降沿</li> </ul>
	当前不支持设置
去重距离	去重距离,需根据实际情况设置,一般可设置为
	2mm~5mm
	去重即检测一个有效信号后,在后续一段距离内如
文档版本 V1.0(2020-05-20) 用户	「手册 版权所有 © 越疆科技有限公司



参数	说明
	果再检测到该信号变化,则认为是该信号为无效信
	号, 会自动剔除

2. 单击"开始监听"开始监听。

3. 把一个工件放置在传感器上游,开启传送带,工件随传送带移动,待工件 经过传感器且移动至机械臂工作空间后停止传送带。

4. 使能机械臂,并点动机械臂至工件的中心位置,单击"示教点位置"读取 工件当前的位置。

5. 单击"计算传感器位置",根据工件当前的示教位置和传送带移动距离获 取触发传感器时工件的位置

步骤5 (可选)单击"相机",标定视觉,如图 3.15所示。

视觉标定前,还需在视觉软件设置机械臂的IP地址和端口,以便机械臂与视觉系统通过TCP/IP进行通信,将视觉数据发送至机械臂。其中,IP地址为192.168.5.1,端口为8080。数据发送格式如下:

- 同一帧照片中发现的工件一同发送。
- 工件的数据格式为(x,y,r,classID),其中,classID为工件分类属性。
- 不同工件数据以分号隔开。
  - 一帧数据以"&"结束。 数据格式例如: x1,y1,r1,1;x2,y2,r2,4;x3,y3,r3,2&

①注意

不同的视觉品牌使用的视觉软件不同,视觉的相关设置也不同,本节不描述详细的操作方法,请根据实际情况操作。

传送带编号: 0 👻				预览	基本设置
基本设置标定					编码器
宽度:	3072	高度:	2048		工作边界
触发10口:	0	触发类型:	上升沿 🗸		
拍照间距.	0.0000	毫米 去重距离:	0.0000	毫米	
单位类型: 像素	Ŧ				
					重置

图 3.15 视觉标定界面

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



视觉标定是为了在相机触发拍照后,获取工件在拍照位置的用户坐标系下的坐标,以便工件随传送带移动时,根据坐标的偏移来计算出每一时刻工件所在的用户坐标系下的位置。

将标定板放置于相机视野范围内的传送带上,触发拍照获取标定板上9个点的 像素坐标并记录当前传送带的位置;控制传送带移动使标定板停止在机械臂的 工作空间内的某一位置,此时移动机械臂对准标定板上9个点,分别获取9个点 的示教位置,同时记录当前传送带的位置,根据9个点的示教位置、像素坐标 和传送带移动距离计算拍照位置处的笛卡尔坐标,获取像素坐标与笛卡尔坐标 之间的关系,详细步骤如下所示。

1. 在相机界面的"基本设置"页签设置视觉基本参数。

基本参数说明如表 3.3示。

参数	说明
传送带编号	传送带编号
	当前不支持设置
宽度	相机的像素宽
	默认值: 3072
高度	相机的像素高
	默认值: 2048
触发IO口	当前不支持设置
触发类型	当前不支持设置
拍照间距	触发拍照距离
	即传送带移动给定的距离后,机器人控制系统发出 一个DO信号触发相机拍照
	建议"拍照间距"设置范围: (1-d)/2< Interval < (1-d)
	其中,l表示相机视野在传送移动方向上的宽度,d 表示待抓取工件的最大宽度
去重距离	去重距离,需根据实际情况设置
	去重距离即检测到工件在前后两次拍照中同时被 识别,会自动剔除其中一个
单位类型	当前不支持设置

表 3.3 视觉基本参数设置

2. 单击"标定"。

进入标定界面。

3. 将标定板放置于传送带上且在相机视野范围内,并单击"传送带位置"记 文档版本 V1.0 (2020-05-20) 用户手册 版权所有 © 越疆科技有限公司





录当前的编码器值,标定板如图 3.16所示。

图 3.16 标定板

4. 通过视觉软件触发拍照,在视觉软件上获取标定板上9个标定点的像素坐标,并在标定界面按顺序输入至对应的编号空格位置,如图 3.17所示。



举本攻員	拍摄	x		带位置	0.0	000 Y	相机
	1	0.0000	0.0000	6	0.0000	0.0000	
	2	0.0000	0.0000	7	0.0000	0.0000	
	3	0.0000	0.0000	8	0.0000	0.0000	
	4	0.0000	0.0000	9	0.0000	0.0000	
	5	0.0000	0.0000		Т		
	L		ć	+11			

图 3.17 获取像素坐标

- 5. 单击"下一个"。
- 6. 开启传送带,将标定板随传送带移动至机械臂工作空间后停止传送带,单击"传送带位置"获取当前的编码器值。

7. 使能机械臂,点动机械臂至标定板上对应的9个标定点,并分别单击"1"~"9"获取机械臂对应的笛卡尔坐标,如图 3.18和图 3.19所示。



图 3.18 视觉标定



姜送带编号: 0 v					700 基本设置 ●
基本设置标定					编码器
					相机
	传送	传送带位置		00	
位置 x	У	位置	x	¥	工作边界
1 0.0000	0.0000	6	0.0000	0.0000	]
2 0.0000	0.0000	7	0.0000	0.0000	]
3 0.0000	0.0000	8	0.0000	0.0000	]
4 0.0000	0.0000	9	0.0000	0.0000	]
5 0.0000	0.0000		ħ	茂	
	:	计算			重置
					保存 〕

图 3.19 获取标定点笛卡尔坐标

# ⚠注意

请务必按照视觉像素点的顺序依次点动机械臂对准这9个点获取对应的笛卡尔坐标,以免视觉标定错误。

8. 单击"计算",根据9个标定点的像素坐标、对应的笛卡尔坐标和传送带移动距离计算标定板9个点在拍照处的笛卡尔坐标,获取像素坐标和笛卡尔坐标的对应关系。

步骤6 单击"工作边界"标定边界。



图 3.20 标定边界说明





图 3.21 边界标定界面

边界参数如表 3.4所示。

表 3.4 边界参数设置

参数	说明
传送带编号	传送带编号
	当前不支持设置
工作进入边界	机械臂工作进入边界
拾取下边界	工件拾取区域的下边界
	当工件超出该边界且还未被拾取时将不再抓取
工作离开边界	机械臂工作离开边界
停止距离	工件接近"工作进入边界"时的平滑停止距离
	当前不支持设置

1. 使能机械臂,点动机械臂至开始跟踪的位置,单击"工作进入边界"获取工作进入边界。

- 2. 使能机械臂,点动机械臂至预计可以完成流程的最迟的工件跟踪开始位
- 置,单击"拾取下边界"获取工具拾取区域的下边界。
  - 该位置需要根据传送带的运动速度以及实际经验设置。
- 3. 使能机械臂,点动机械臂至必须结束跟踪的位置(为了防止跟踪至机器人运动区域外),单击"工作离开边界"获取工作离开边界。

步骤7 单击"保存"。

# 3.1.5 应用案例

### 3.1.5.1 传送带动态视觉跟踪抓取示例

传送带动态视觉跟踪需示教6个点:

文档版本 V1.0(2020-05-20)

版权所有 ◎ 越疆科技有限公司



- 等待工件点: P1
- 跟踪工件上方点: P2
- 抓取工件点: P3
- 抬升工件点: P4
- 放置上方点: P5
- 放置点: P6

如果对工件抓取角度有要求,还需设置抓取基准角度。

本章节以无抓取角度要求为例进行说明,存点示意图如图 3.22所示。



图 3.22 存点示意

# 前提条件

- DobotSCStudio上已创建传送带动态视觉跟踪工程,详细请参见1.5.3.1 创建工程。
- 机械臂已连接气泵。
- 机械臂末端已安装夹具。
- DobotSCStudio已切换至手动模式。
- 已安装视觉软件。
- (可选)若安装了偏心工具,则需标定工具坐标系,详细请参见1.2.3.1 四轴工具坐标系,详细请参见1.2.3.1 四轴工具坐标系设置。

# 

- 点动机械臂时请先使能机械臂。
- 不同的视觉品牌使用的视觉软件不同,视觉的设置和模板的建立也不同,本
   节不描述详细的操作方法,请根据实际情况操作。
- P1、P2、P3、P4点的臂方向必须一致。
- P2、P3、P4需在设置的用户坐标系下示教。如果末端安装了偏心工具,还需 先设置工具坐标系,然后在工具坐标系下示教P2、P3、P4。
- P1点关节角度建议: J3约120°左右, J4约0°。

```
文档版本 V1.0(2020-05-20) 用户手册 版权所有 © 越疆科技有限公司
```



#### 操作步骤

步骤1 在基坐标下,点动机械臂至等待工件点P1,并在"示教点"页面单击 ↔ 新 增存点信息。

⚠注意

如果末端安装了偏心工具,则步骤 2和步骤 3需要在工具坐标系下进行。

**步骤2** 在设置的用户坐标系下,点动机械臂至合适的跟踪高度(高于工件的高度),记录下P2点的高度,假设为z1。

步骤3 在设置的用户坐标系下,把工件放置于传送带上,点动机械臂至工件中心,此时记录下P3点的高度,假设为z0。

**步骤 4** 在基坐标系下,在"示教点"页面新增P2,P3,P4点的存点信息,其中,P2=(0,0,z1,0),P3=(0,0,z0,0),P4=(0,0,z1,0)。

**步骤5** 将工件放置于相机视野内,并在视觉软件上建立模板,以便提取工件特征。

请将当前的模板角度设置为0°。

□□说明

如果使用传感器跟踪,则将工件以正确的工位放置(传感器跟踪的工件有固定的工位或方向)。

- **步骤6** (可选)若对抓取角度有要求,则执行该步骤。如果对抓取角度无要求,则跳过该步骤。
- 1. 控制传送带运动,将工件移动至机械臂可抓取的范围,点动机械臂去抓取 工件(抓取点为工件中心),在基坐标系下调整R轴角度为0°。
- 2. 切换至设置的用户坐标系下,记录此时R轴角度,假设为r1。
- 3. 调整R轴角度至合适位置抓取物料,此时记录R轴角度,假设为r2。

因此, P2、P3、P4点R轴角度r为r2-r1。若r>180°,则r=r-360°。若r<-180°,则r=r+360°。若 -180≤r≤180°,则r 不变。

4. 在"示教点"页面修改P2, P3, P4点的存点信息。其中, P2=(0,0,z1,r), P3=(0,0,z0,r), P4=(0,0,z1,r)。

步骤7 在基坐标系下点动机械臂至工件放置点P6,并在"示教点"页面单击

→ 新增存点信息。

步骤8 在基坐标系下点动机械臂至工件放置点上方位置P5,并在"示教点" 页面单击 → 新增存点信息。







- SyncCnv(0)与StopSyncCnv()之间的运动指令仅支持Move指令。
- SyncCnv(0)与StopSyncCnv()之间可支持wait、DI和DO指令。

程序 3.1 传送带跟踪程序

CnvVison(0)	//开启传送带		
DO(1,0)	//通过 DO1 和 DO2 控制气泵启停和状态		
DO(2,0)			
local flag	//是否存在工件标识		
local typeObject	///工件的属性分类		
local point = $\{0,0,0\}$	//出跟踪队列的工件的坐标(X,Y,R)		
while true do			
文档版本 V1.0(2020-05-20)	用户手册 版权所有 © 越疆科技有限公司		



```
Go(P1,"Speed=100 Accel=100 SYNC=1") //机械臂等待点,一般设置在工作空间内
print("Test")
while true do
     flag,typeObject,point = GetCnvObject(0,0) //查询是否有工件,如果有则跳出循环
     if flag == true then
          break
     end
     Sleep(20)
end
                                    //同步传送带,开始跟踪
SyncCnv(0)
Move(P2,"SpeedS=100 AccelS=100")
                                   //跟踪工件上方点
Move(P3,"SpeedS=100 AccelS=100")
                                    //抓取工件点
Wait(100)
DO(1,1)
                                     //开启气泵,抓取工件
DO(2,1)
Wait(100)
Move(P4,"SpeedS=100 AccelS=100 SYNC=1") //抬升工件点
                                     //停止传送带跟踪
StopSyncCnv()
Sleep(20)
Go(P5, "Speed=100 Accel=100")
                               //放置点上方
Go(P6,"Speed=100 Accel=100 SYNC=1") //放置点
Sleep(1000)
DO(1,0)
                                //关闭气泵
DO(2,0,"SYNC=1")
Sleep(1000)
Go(P5,"Speed=100 Accel=100")
```

end

# 3.2 码垛工艺

### 3.2.1 概述

在搬运应用中,有些搬运的物料排列规则且间距均匀,若一一示教各个物料的位置会存 在误差大、效率低的问题。码垛工艺可有效的解决该类问题。

一个完整的码垛工艺流程包括码垛参数设置和调用码垛API编程。用户在DobotSCStudio 上设置码垛相关参数后,生成的配置文件自动导入Dobot机器人控制系统,然后可根据实际 应用场景调用码垛API编写脚本,快速实现码垛搬运作业。

### 3.2.2 码垛设置

码垛参数设置包括基本参数设置、路径点设置。基本参数设置即设置码垛名称、跺个数、 码垛方向以及间距。路径点即跺在码垛和拆跺路径中经过的配置点,包括:

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



- 安全过渡点 (A点): 进跺盘和出跺盘之后都会经过的一个点位, 可固定或随层高变化。
- 准备点(B点):由目标点和设置的偏移计算出来的点位。
- 目标点 (C点): 第一跺的目标点位。

码垛路径和拆跺路径分别如图 3.24和图 3.25所示。



图 3.24 码垛路径



图 3.25 拆跺路径

其中,跺表示需要码放或搬运的工件、产品等。跺盘表示用于放置跺的物品或区域。码 垛表示跺按照配置的跺型放置到跺盘。拆跺指按照配置的跺型将跺从跺盘中取出。跺型表示 在所有跺的摆放方式,当前仅支持矩阵跺型,即只由一种排样(一层中跺的摆放方式)组成, 跺在X、Y、Z方向上等距分布,如图 3.26所示。

文档版本 V1.0(2020-05-20)





### 图 3.26 矩阵跺型

本节描述如何设置码垛参数。

# 前提条件

- 机械臂已上电。
- 机械臂末端已安装吸盘或夹爪套件。
- (可选)已在跺盘建立用户坐标系。用户在示教时可根据实际情况选择坐标系。

### 操作步骤

- 步骤1 在"工艺包"页面单击"矩阵码垛"。
  - 进入码垛设置页面,如图 3.27所示。

文档版本 V1.0(2020-05-20)



码垛			基础设置 托盘 过度点 准备点 选择
编号	变量名	用户	
			名称:
			方问: ->->z *
			X Y Z
			总数: 0 10 10 10 1
			目标点: 0.0000 \$ 0.0000 \$ 0.0000 \$
			添加 替换 删除 保存

### 图 3.27 码垛页面

表 3.5 码垛基本参数说明

参数	说明
名称	码垛名称
方向	码跺方向
	取值: X->Y->Z或Y->X->Z
	本示例选择 "X->Y->Z"
总数	跺在X、Y、Z各个方向上的数量
目标点	跺在X、Y、Z各个方向上的间距

**步骤3** 示教机械臂至第一个跺的位置,并在"托盘"页签设置单击"获取坐标",如图 3.28所示。

用户坐标系编号需与示教时选择的用户坐标系保持一致。

步骤2 在"基础设置"页签设置码垛基本参数。 参数说明如表 3.5所示。


码垛			基础设置 托盘 过度点 准备点 选择
编号	变量名	用户	
			第一个托盘的位置。
			用户坐标: 0 X: 0.0000 ↓ Rx: 0.0000 ↓ Y: 0.0000 ↓ Ry: 0.0000 ↓ Z: 0.0000 ↓ Rz: 0.0000 ↓ 家职坐标 定位
			添加

图 3.28 示教第一个跺的目标点位

**步骤 4** 示教机械臂至过渡点,并在"过渡点"页签设置单击"获取坐标", 如图 3.29所示。

同样,用户坐标系编号需与示教时选择的用户坐标系保持一致。

若勾选"随层高变化",则过渡点自动随层高变化。若不勾选,则为固定点。

码垛			基础设置	托盘	过度点	准备点	选择			
编号	交量名	用户	安全过源 每层都石	寛点,女 下同。	口果检查为	随每一层	的高度	而变化,则	准备点	
			用户坐 X: Y: 乙: 回随层高	会标: 变化	0 0.0000 0.0000 0.0000		Rx: Ry: Rz:		0.0000	
					获取坐标		(	定位		
	刪除托盘		添加	)	替	换	(H	除	保存	$\bigcirc$

图 3.29 示教过渡点

**步骤5** 示教机械臂至第一垛的正上方即准备点,并在"准备点"页签设置单击"获取坐标"。

同样,用户坐标系编号需与示教时选择的用户坐标系保持一致。



步骤6 单击"添加",生成参数配置文件导入Dobot机器人控制系统。

### 3.2.3 应用案例

配置完码垛参数后,可通过调用码垛API编写脚本,实现快速搬运作用。本节以码垛为示例进行说明。

#### 程序 3.2 码垛示例

local MPpick = MatrixPallet(0, "IsUnstack=true Userframe=8")	//定义码垛实例
Reset(MPpick)	//初始化码垛实例
while true do	
MoveIn(MPpick,"velAB=90 velBC=50")	//开始码垛
MoveOut(MPpick)	
result=IsDone(MPpick)	
if (result == true)	//检查码垛是否完成
then	
print("EXIT")	
break	
end	
end	
Release(MPpick)	//释放码垛实例

文档版本 V1.0(2020-05-20)



# 4. **典型应用**

## 4.1 Modbus 应用

Dobot机器人控制系统可通过Modbus协议与外部设备进行通信。其中,外部设备如PLC 作为Modbus主站,Dobot机器人控制系统作为Modbus从站。

本节以PLC为例,通过读写相关寄存器来控制机械臂运行,其接线示意图如图 4.1所示。



图 4.1 典型接线方式

其中,Dobot机器人控制系统和PLC的IP地址需在同一网段。Dobot机器人控制系统的IP 地址可在"配置 > 网络配置"中查询并修改;端口为502,不能修改。

程序流程图如图 4.2所示,对应的程序示例如程序 4.1所示。

□□说明

- 图 4.2中提及的相关寄存器地址均为Dobot机器人控制系统侧的寄存器地址, 对应PLC的相关寄存器地址请参见2.15.1 Modbus寄存器说明。
- 本节仅说明Dobot机器人控制系统侧的程序示例,PLC侧的程序示例不做详细 说明。

文档版本 V1.0(2020-05-20)





图 4.2 程序流程图

程序 4.1 机器人从站程序示例

local data = $\{1\}$		
local data2 = $\{1\}$		
文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



```
local coil = { }
local RST = \{0\}
while true
do
     coil = GetCoils(1000,1)
     if coil[1] == 1 then
         Go(P1,"SYNC=1")
         Sleep(1000)
          SetCoils(1001, 1, data)
          while true do
                 Sleep(1000)
                 data2 = GetHoldRegs(1000,1)
                 if data2[1] == 666 then
                      Go(P2,"SYNC=1")
                      SetHoldRegs(1001,#data,data2,"U16")
                      Sleep(1000)
                      Go(P3,"SYNC=1")
                      SetCoils(1000, 1, RST)
                      SetCoils(1001, 1, RST)
                      SetHoldRegs(1000,1,RST,"U16")
                      \operatorname{coil} = \{0\}
                      data2 = \{0\}
                      break
                 end
            end
     end
end
```

### 4.2 I/O 应用

### 4.2.1 抓取瓶子应用

本节以六轴机械臂为例,描述抓取瓶子的应用示例。完整的工作流程包括瓶子输送、抓 取瓶子放至盒子并合上盒盖、以及盒子输送。瓶子和盒子的输送需由传送带、PLC实现。抓 取瓶子等一系列操作由机械臂完成。本节仅介绍机械臂抓取部分,应用场景如图 4.3所示。





图 4.3 应用场景

机械臂抓取所需点位说明如表 4.1所示,所有点位通过示教得到。

点位	说明
P1	安全位置
P2	抓取瓶子过渡点
Р3	抓取瓶子接近点
P4	抓取瓶子点
P5, P7, P9, P11	放置瓶子上方点(盒子上方点)
P6, P8, P10, P12	放置点
P13	抓取盖子过渡点
P14	抓取盖子点
P15	抓取盖子上方点
P16	盖盖子上方点(盒子上方点)
P17	盖盖子点

表 4.1 点位说明

机械臂每执行一个任务,都会发送信号至PLC标识当前需完成的任务,以便PLC侧进行相关工作,详细信号说明如表 4.2所示。

#### 表 4.2 自定义数字信号说明

数字信号 说明		
输入信号		
DI2	PLC允许抓取瓶子信号	

文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司



数字信号	说明
DI3	PLC允许抓取盒盖信号
输出信号	
DO7	回到安全位置信号
DO8	抓取完一个瓶子信号
D09	盒盖合上信号
DO17	抓取瓶子信号
DO18, DO19	抓取盒盖信号

详细工作流程如图 4.4所示。

文档版本 V1.0 (2020-05-20)





- 图 4.4 抓取瓶子流程图
- 程序 4.2 抓取瓶子 Demo

local bottleNum = 4		
local readyPos=P1Safe position		
local bottleLoadingPos=P4Bottle ready po	sition	
文档版本 V1.0(2020-05-20)	用户手册	版权所有 © 越疆科技有限公司
	111	

```
local bottleUnloadingPosup ={P5, P7, P9, P11}
local bottleUnloadingPos ={P6, P8, P10, P12}
local boxLoadingPos = P14 -- The position of the lid of a box
local boxUnloadingPos = P17 -- The position of the lid
while true do
     for bottleIndex = 1, bottleNum do
           Go(readyPos, "Speed=100 Accel=100")
           DOExecute(7,1)
           Wait(1000)
           while true do --Grab bottle signal
               Sleep(500)
               input = DI(2,1)
               if input == 1 then
                    break
               end
           end
           DOExecute(7,0)
           Wait(500)
           Go(P2, "Speed=100 Accel=100 SYNC=1") -- Transition position
           Move(P3,"Speed=100 Accel=100 SYNC=1") -- Above the bottle
           Move(bottleLoadingPos," Speed=100 Accel=100 SYNC=1")
                                                                       --Grab bottle position
           DOExecute(17,1)
                                                  --Grabbing signal
           Wait(500)
           Move(P3,"Speed=100 Accel=100 SYNC=1") -- Above the bottle
           Go(P1,"Speed=100 Accel=100 SYNC=1") -- Transition position
           Go(bottleUnloadingPosup[bottleIndex], "Speed=100 Accel=100 SYNC=1") -- Above the bottle
           Move(bottleUnloadingPos[bottleIndex],"Speed=100 Accel=100 SYNC=1") --Bottle location
           DOExecute(17,0)
                                                  --Grabbing signal
           Wait(500)
           Move(bottleUnloadingPosup[bottleIndex],"Speed=100 Accel=100 SYNC=1")
           DOExecute(8,1) --Grab the bottle to complete the signal
           Wait(1000)
           DOExecute(8,0)
    end
    Go(readyPos,"Speed=100 Accel=100 SYNC=1") --Safe position
    while true do -- Grab the lid signal
        Sleep(1000)
     文档版本 V1.0 (2020-05-20)
                                                用户手册
                                                                       版权所有 © 越疆科技有限公司
```



```
input = DI(3,1)
   if input == 1 then
        break
   end
end
Go(P13, "Speed=100 Accel=100 SYNC=1") -- Transition position
Move(P15,"Speed=100 Accel=100 SYNC=1") --Grab the lid transition position
Move(boxLoadingPos," Speed=100 Accel=100 SYNC=1") --Grab the lid
DOExecute(18,1) --Grabbing signal
DOExecute(19,1) --Grabbing signal
Wait(500)
Move(P15,"Speed=100 Accel=100 SYNC=1") --Grab the lid transition position
Move(P16,"Speed=100 Accel=100 SYNC=1") --Cover the box and cover it transition positio
Move(boxUnloadingPos,"Speed=10 Accel=10 SYNC=1") --Cover the box and cover it
DOExecute(18,0)
DOExecute(19,0)
Wait(500)
DOExecute(9,1)
Wait(500)
Move(P16,"Speed=100 Accel=100 SYNC=1") --Cover the box and cover it transition positio
Go(readyPos,"Speed=100 Accel=100 SYNC=1") --Safe position
DOExecute(9,0)
Wait(500)
```

end