



智流形机器人
从 制 造 到 智 造

INSTRUCTION

智流形机器人控制系统

用户手册

版权声明

深圳市智流形机器人技术有限公司版权所有。

深圳市智流形机器人技术有限公司保留所有版权以及相关的知识产权。

声明

深圳市智流形机器人技术有限公司保留在没有预先通知的情况下修改产品或其特性的权利。

深圳市智流形机器人技术有限公司并不承担由于使用产品不当而产生的直接或是间接的伤害或损坏的责任。

联系我们

深圳市智流形机器人技术有限公司

Shenzhen iManifold Robot Technology Ltd.

地址：深圳市宝安区华丰国际机器人产业园 B 栋 410

智云流形科技（江阴）有限公司

iManifold Technology (Jiangyin) Ltd

地址：江苏省无锡市江阴市长山大道 55 号天安数码城 60 幢 102 室

目 录

版权声明	1
声明	1
联系我们	1
1 术语与定义	13
1.1 示教器	13
1.2 按键表示	13
1.3 DH 模型	13
2 安全	15
2.1 安全规范	15
2.2 操作人员安全注意事项	16
2.3 操作过程安全注意事项	17
2.3.1 一般原则	17
2.3.2 紧急处理	18
2.3.3 操作手持示教器	18
2.3.4 安全工具	20
2.3.5 手动模式的安全事项	21
2.3.6 自动模式的安全事项	22
3 安装配线	23
3.1 安装场所和环境	23
3.1.1 安装环境要求	23
3.1.2 安装位置要求	24

3.1.3 安装配线注意事项	24
3.2 控制器安装说明	26
3.2.1 主控及 I/O 单元安装方式	27
3.2.2 主控单元接线说明	28
3.2.3 示教器安装接线	31
3.2.4 IO 模块接线定义	32
3.3 供电电源	34
3.3.1 三相电源	34
3.3.2 噪声滤波器的安装	34
3.3.3 漏电断路器的安装	35
4 手持操作示教器	36
4.1 示教器键位功能	37
4.2 示教器界面	42
4.3 系统应用菜单	44
4.3.1 常用工具栏自定义	45
4.3.2 菜单选项说明	45
4.4 状态栏	47
4.4.1 显示伺服状态	47
4.4.2 工作模式	47
4.4.3 坐标系	48
4.4.4 当前工具号, 工件号	48
4.4.5 速度	48

4.4.6 示教文件回放执行模式	48
4.4.7 当前权限和网络状态	49
4.5 浮动菜单	49
4.6 机器人状态及报警历史查看	51
4.6.1 机器人状态	51
4.6.2 机器人报警历史及操作 log 查看	52
5 快速操作入门	54
5.1 伺服电源接通	54
5.1.1 接通主电源	54
5.1.2 接通伺服电源	55
5.2 伺服电源切断	55
5.2.1 切断伺服电源	55
5.2.2 切断主电源	56
5.3 坐标系的选择	56
5.4 手动速度调整	57
5.5 轴操作	57
5.6 平齐	59
5.7 回零	59
6 工程管理界面	61
6.1 机器人工程结构说明	61
6.2 工程操作	63
7 程序编辑界面	67

7.1 当前打开的程序	67
7.2 历史程序	67
7.3 程序翻页	68
7.4 程序编辑	68
7.5 扩展栏	70
8 变量	74
8.1 全局变量	74
8.1.1 全局位置型变量	74
8.1.2 全局增量位置点	76
8.1.3 全局字符串变量	76
8.1.4 全局数值变量	77
8.2 工程变量	77
8.2.1 工程位置型变量	78
8.2.2 工程增量位置型变量	79
8.2.3 工程数值型变量	79
8.2.4 工程字符串变量	79
8.3 机器人姿态	80
9 坐标系	81
9.1 坐标系系统介绍	81
9.1.1 关节坐标系 — ACS	81
9.1.2 机器人坐标系 — KCS	81
9.1.3 工具坐标系 — TCS	81

9.1.4 世界坐标系 — WCS	82
9.1.5 工件坐标系— PCS	82
9.2 坐标系管理界面	84
9.3 关节坐标系 (ACS)	84
9.4 机器人坐标系 (KCS)	85
9.5 工具坐标系	86
9.5.1 工具坐标系轴动作	86
9.5.2 工具坐标系的标定	87
9.6 工件坐标系 (PCS)	95
9.6.1 工件坐标系轴动作	95
9.6.2 工件坐标系的标定	96
9.7 工作空间监控	101
9.7.1 工作空间监控说明	101
9.7.2 工作空间监控设置	103
10 系统参数设置	111
10.1 系统参数	111
10.2 硬件参数	113
10.3 关节	115
10.4 辅助轴	117
10.5 笛卡尔	118
10.6 规划参数	121
10.7 DH 参数	126

10.8 耦合	127
11 通用 IO	128
11.1 输入信号	128
11.1.1 界面	128
11.1.2 功能说明	128
11.2 输出信号	130
11.2.1 界面	130
11.2.2 功能说明	130
11.3 IO 接线	132
12 远程 IO	133
12.1 界面介绍	133
12.1.1 文件设置	133
12.1.2 远程 IO 输入	134
12.1.3 远程 IO 输出	137
12.1.4 基本设置	139
12.2 单独模式使用	140
12.2.1 应用场景	140
12.2.2 操作步骤	140
12.3 组合模式使用	144
12.3.1 应用场景	144
12.3.2 操作步骤	144
12.4 附加任务使用	148

12.4.1 应用场景	148
12.4.2 操作步骤	148
13 后台任务	153
13.1 后台任务功能简介	153
13.2 编辑后台任务	153
13.3 后台任务配置	154
14 动力学设置	155
14.1 全局参数	155
14.2 动力学关节参数	157
14.3 动力学参数辨识	158
14.3.1 本体辨识	158
14.3.2 摩擦辨识	160
14.3.3 负载辨识	162
14.4 零位标定	165
14.5 文件生成	166
14.5.1 参数设置	167
15 其他功能	170
15.1 零点标定	170
15.2 20 点法	173
15.3 异常处理	175
15.4 权限管理	176
15.4.1 用户切换	176

15.4.2 硬件测试	177
15.4.3 界面风格	178
15.4.4 升级与备份	178
16 指令系统	180
16.1 命令一览	180
16.2 指令规范	181
16.3 运动指令	182
16.3.1 运动指令的处理机制	182
16.3.2 Movj 关节运动	182
16.3.3 Movs 关节样条运动	184
16.3.4 Movl 直线运动	185
16.3.5 Movp 笛卡尔样条运动	187
16.3.6 Movc 圆弧运动	188
16.3.7 MoveExt 外部轴组关节运动	192
16.3.8 MovjUntil 可条件中断 Movj	193
16.3.9 MovlUntil 可条件中断 Movl	194
16.3.10 MovcUntil 可条件中断 Movc	195
16.3.11 MoveExtUntil 可条件中断 MoveExt	196
16.4 运动参数	198
16.4.1 SetDyn 加减速、加加速度时间	198
16.4.2 SetSpeed 全局速度百分比	198
16.4.3 SetCoord 坐标系切换	199

16.4.4 ResetCoord 坐标系切换复位	199
16.4.5 SetTool 工具坐标系切换	199
16.4.6 ResetTool 工具坐标系切换复位	200
16.4.7 SetOriMode 姿态模式设置	200
16.4.8 ResetOriMode 姿态模式复位	201
16.4.9 SetOffset 位置偏置设置	201
16.4.10 ResetOffset 位置偏置取消	202
16.4.11 SetBlendType 设置过渡段类型	202
16.4.12 ResetBlendType 设置过渡段类型	203
16.4.13 GetAxisInput 获取驱动器输入	203
16.5 IO 操作	204
16.5.1 SetDo 输出数字量	204
16.5.2 SetDoAt 运动中输出 IO	205
16.5.3 SetDoDeltaS 运动中输出数字量	206
16.5.4 SetDoDeltaT 运动中输出数字量	207
16.5.5 GetDo 数字量输出读取	208
16.5.6 GetDi 数字量输入获取	208
16.5.7 SetAo 输出模拟量	209
16.5.8 SetAoAt 运动中输出模拟量	210
16.5.9 GetAo 模拟量输出获取	210
16.5.10 GetAi 模拟量输入获取	211
16.5.11 Wait 等待条件完成	211

16.5.12 Pulse 脉冲输出	212
16.6 运动管理	212
16.6.1 Delay 延时	212
16.6.2 WaitExecutor 停止译码	213
16.6.3 tod 获取系统计时	214
16.6.4 SysDate 记录系统时间	214
16.6.5 Pause 暂停	215
16.6.6 Alarm 用户报警	215
16.6.7 GetPoint 保存当前位置	215
16.7 控制流程	216
16.7.1 CALL 调用子程序	216
16.7.2 JUMP—LABEL 跳转—跳转标识	216
16.7.3 IF 条件判断	217
16.7.4 ELSEIF 条件判断	217
16.7.5 WHILE 循环条件判断	218
16.7.6 FOR 循环条件判断	219
16.8 注释及计算	221
16.8.1 /**/多行注释	221
16.8.2 注释/取消注释当前行	221
16.8.3 ASSIGN(=)赋值	221
16.8.4 ADD(+)加法	222
16.8.5 SUB(-)减法	222

16.8.6 MUL(*)乘法	223
16.8.7 DIV(/)除法	223
16.8.8 Expr 高级表达式/语句	224
16.9 数学函数	228
16.10 字符函数	229
16.10.1 strlen(X) 求字符串长度	229
16.10.2 str2num(X)字符串转实数	230
16.10.3 num2str(X)实数转字符串	230
16.10.4 strcat(X,Y)字符串拼接	231
16.10.5 strleft(X,N)取字符串 X 的左边 N 个字符	231
16.10.6 strright(X,N)取字符串 X 的右边 N 个字符	232
16.10.7 strmid(X,M,N)取字符串 X 的第 M 至 N 个字符	232
16.10.8 strsplit(X,Y,N)字符串拆分	233
16.10.9 buildstr 字符串构造	233
16.10.10 parsestr 字符串解析	234
16.11 通讯	234
16.11.1 TcpOpen 打开 tcp/ip 通讯	234
16.11.2 TcpRecv 接收数据	235
16.11.3 TcpSend 发送数据	236
16.11.4 TcpClose 关闭 tcp/ip 连接	237
16.11.5 SerialOpen 打开串口	237
16.11.6 SerialRecv 接收数据	238

16.11.7 SerialSend 发送数据	239
16.11.8 SerialClose 关闭 tcp/ip 连接	240
16.11.9 CanbusOpen 打开 Canbus 通讯	240
16.11.10 CanbusRecv 接收数据	241
16.11.11 CanbusSend 发送数据	242
16.11.12 CanbusClose 关闭 Canbus 连接	242
16.11.13 MBTcpOpen 开启 modbusTcp 通讯	243
16.11.14 MBTcpRead/Write 读/写寄存器	244
16.11.15 MBTcpOpen 关闭 modbusTcp 通讯	245
17 错误信息	246

1 术语与定义

下列术语与定义适用于本文件。

1.1 示教器

能用它对机器人进行编程或使机器人运动，并与控制系统相连的手持式单元。

1.2 按键表示

1. 手持操作示教器上的键用【】表示：例如急停键用【急停】键来表示。
2. 显示屏上的键用{}表示：例如清除键用{清除}键来表示。

1.3 DH 模型

Denavit-Hartenberg (DH) 模型是一种应用于机器人运动学中的建模技术。它通过在每个连杆上建立坐标系，并利用齐次坐标变换来实现连杆之间的坐标转换。

机器人系统中通过 DH 模型来描述系统支持的不同构型的工业机器人。以下是系统支持的多种机器人构型。

		关节配置Axes				应用工艺 (Applications)					
机器人构型 RobotType		Ground	Hand	Aux	Tracking	Arc Welding	Laser Cutting	Palletize	CNC	Vision	Bending
Gantry 	LL-	R--	8	-	-	✓	✓	-	-	-	-
	LL-	RR-	8	-	-	✓	✓	-	-	-	-
	LL-	RRR	8	-	-	✓	✓	-	-	-	-
	LLL	---	8	✓	✓	✓	✓	✓	✓	✓	-
	LLL	R--	8	✓	✓	✓	✓	✓	✓	✓	-
	LLL	RR-	8	✓	✓	✓	✓	✓	✓	✓	✓
	LLL	RRR	8	✓	✓	✓	✓	✓	✓	✓	✓
SCARA 	RRL	R--	8	✓	✓	✓	✓	✓	✓	✓	-
	LRR	R--	8	✓	✓	✓	✓	✓	✓	✓	-
	RLR	R--	8	✓	✓	✓	✓	✓	✓	✓	-
Articulate 	RRR	RRR	8	✓	✓	✓	✓	✓	✓	✓	✓
	RRR	R-R	8	✓	✓	✓	✓	✓	✓	✓	✓
	RRR	-RR	8	✓	✓	✓	✓	✓	✓	✓	✓
	RRR	-R-	8	✓	✓	✓	✓	✓	✓	✓	✓
	RLL	--R	-	✓	✓	✓	✓	✓	✓	✓	-
	RRR	--R	-	✓	✓	✓	✓	✓	✓	✓	-
	RL-	LRR	8	-	-	-	-	✓	-	-	-
	LRL	---	8	-	-	-	-	✓	-	-	-
	LRL	R--	8	-	-	-	-	✓	-	-	-

		关节配置Axes				应用工艺 (Applications)					
机器人构型 RobotType		Ground	Hand	Aux	Tracking	Arc Welding	Laser Cutting	Palletize	CNC	Vision	Bending
Delta 	RR-	---	-	✓	✓	✓	✓	-	-	-	-
	RRR	---	-	✓	✓	✓	✓	-	-	-	-
	RRR	-RR	-	✓	✓	✓	✓	-	-	-	-
	RRR	-R-	-	✓	✓	✓	✓	✓	✓	✓	-
Tripod 	LLL	---	-	✓	✓	✓	✓	✓	✓	✓	-
	RRR	--R	-	✓	✓	✓	✓	✓	✓	✓	-
Customize 	LL-	RR-	8	✓	✓	✓	✓	✓	✓	✓	✓
	LLR	RR-	8	✓	✓	✓	✓	✓	✓	✓	✓
	LLR	RRR	8	✓	✓	✓	✓	✓	✓	✓	✓
Painting 	RRR	RRR	8	✓	✓	✓	✓	✓	✓	✓	-
	RRR	RRR (R)	8	✓	✓	✓	✓	✓	✓	✓	-
Steward 	6-SPS		0	✓	-	-	-	-	-	✓	-
	6-UPU		0	✓	-	-	-	-	-	✓	-

0



2 安全

2.1 安全规范

机器人与其他机械设备的要求通常不同，例如空间运动范围大、快速的操作、手臂的快速运动等，这些都会造成安全隐患。

阅读和理解使用说明书及相关的文件，并遵循各种规则和流程，以免造成人身伤害或设备事故。用户有责任保证机器人系统始终处于安全的操作环境下，并符合和遵守国家及地方有关安全性的法令、法规及相关条例。

遵照法规条例



注意

机器人的示教维护必须遵照下列法规

1. 有关工业安全和健康的法律；
2. 有关工业安全和健康法律的强制性命令；
3. 有关工业安全和健康法律的相应条例；
4. 工业机器人的安全操作 (ISO 10218)。

示教和维修机器人的工作被列入工业安全和健康法律中的“**危险操作**”。

安全管理



注意

1. 示教和维护机器人的人员必须事先经过培训。
2. 根据符合有关法规的具体政策进行安全管理。指定授权的操作者及安全管理
人员，并给予进一步的安全教育。

2.2 操作人员安全注意事项

机器人的最大动作范围内均具有潜在的危险性。

所有机器人行业的相关人员（安全管理员、安装人员、操作人员和维修人员等）必须时刻树立**安全第一**的思想,以确保所有人员的安全为前提进行相关操作。



注意

1. 机器人的安装区域内禁止进行任何的危险作业。
2. 在作业区严格遵守下列条款：
 - 穿着工作服(不穿宽松的衣服);
 - 操作机器人时不许戴手套;
 - 内衣裤、衬衫和领带不要从工作服内露出;
 - 不佩戴大的首饰，如耳环、戒指或垂饰等。
 - 必要时穿戴相应的安全防护用品，如安全帽、安全鞋(带防滑底的)、面罩、防护镜和手套。
3. 未经许可的人员不得接近机器人和其外围的辅助设备。
4. 在设备运转之中，其实机器人看上去已经停止，也有可能是因为机器人在等待启动信号处在即将动作的状态。即使在这样的状态下，也应该将机器人视为正在动作中。为了确保作业人员的安全。应当能以警报灯等的显示或者响声等来切实告知（作业人员）机器人为动作的状态。
5. 绝对不要强制地扳动机器人的轴，否则可能会造成人身伤害或设备损坏。
6. 绝对不要倚靠在机器人电控柜；不要随意地按动操作键。否则可能会造成机器人产生未预料的动作，从而引起人身伤害和设备损坏。

7. 在操作期间，绝不允许非工作人员触动机器人电控柜。否则可能会造成机器人产生未预料的动作，从而引起人身伤害或设备损坏。

2.3 操作过程安全注意事项

2.3.1 一般原则

为了安全操作机器人系统，请遵循下列原则：

1. 如果在保护空间内有工作人员，请手动操作机器人系统。当进入保护空间时，请务必随身携带控制装置（示教器），以便随时控制机器人。
2. 注意旋转或运动的工具，例如切削工具和锯。确保在接近机器人之前，这些工具已经停止运动。
3. 注意工件和机器人系统的高温表面。机器人电机长期运转后温度很高。
4. 注意夹具并确保夹好工件。如果夹具打开，工件会脱落并导致人员伤害或设备损坏。夹具非常有力，如果不按照正确方法操作，也会导致人员伤害。
5. 注意液压、气压系统以及带电部件。即使断电，这些电路上的残余电量也很危险。



危险

6. 当往机器人上安装一个工具时，务必先切断(OFF) 电控柜及所装工具上的电源并锁住其电源开关，而且要挂一个警示牌。安装维护过程中例如接通电源时，可能会造成电击，或使机器人出现非正常运动，从而引起伤害。
7. 绝不要超过机器人的允许范围（机器人的允许范围请参见机器人说明书中的技术规范部分）。否则可能会造成人身伤害或设备损坏。
8. 无论何时如有可能的话，应在作业区外进行示教工作。

9. 当在机器人动作范围内进行示教工作时，则应遵守下列警示：
10. 始终从机器人的前方进行观察；
- 始终按预先制定好的操作程序进行操作；
- 始终具有一个当机器人万一发生未预料的动作而进行躲避的想法，确保您自己在紧急的情况下有退路。

2.3.2 紧急处理



注意

出现下列情况时，应立即按下紧急停止按钮：

1. 操纵机器人运行时，在机器人运行空间内有障碍物可能发生碰撞，或者运行空间内有其他工作人员；
2. 机器人伤害了人或者机器设备。

2.3.3 操作手持示教器

2.3.3.1 操作手持示教器

1. 手持示教器是一款高品质手持式终端，它配备高灵敏度的先进电子设备。为避免操作不当引起的故障或损坏，请在操作时遵循以下说明。
2. 手持示教器仅用于本手册指定的用途。手持示教器按照适用的安全标准进行研发、生产、测试。如果您能遵循本手册中关于安全和使用方面的说明，则在正常情况下，该产品既不会造成人身伤害，也不会致使机器和设备损坏。

2.3.3.2 搬运和清洁

1. 小心搬运。切勿摔打、抛掷或用力撞击。这样会导致破损或故障。
2. 如果受到撞击，务必要验证并确认其安全功能（使动装置和紧急停机）工作正常且未损坏。
3. 设备不使用时，请将其置放于立式壁架上存放，防止意外脱落。
4. 使用和存放手持示教器时始终要确保电缆不会将人绊倒。
5. 切勿使用锋利的物体（例如螺丝刀或笔尖）操作触摸屏。这样可能会使触摸屏受损。用您的手指或触摸笔（根据型号，有些型号的示教器不带触摸笔，一般位于带手持示教器的背面）。
6. 定期清洁触摸屏。灰尘和小颗粒可能会挡住触摸屏造成故障。
7. 切勿使用溶剂、洗涤剂或擦洗海绵清洁手持示教器。请使用软布蘸少量水或中性清洁剂进行清洁。
8. 没有连接 USB 设备时务必盖上 USB 端口的保护盖。如果端口暴露到灰尘中，那么它会中断或发生故障。

2.3.3.3 电缆和电源

1. 打开手持示教器电缆入口区域之前，请关闭电源。否则组件可能受损，或者出现不明信号。
2. 确保任何人员不受电缆羁绊，以免设备跌落在地上。
3. 避免其它物体挤压，以免损坏电缆。
4. 切勿将电缆置于锋利的边缘之上，以免损坏电缆外皮。

2.3.4 安全工具

2.3.4.1 安全保护机制

机器人系统可以配备各种各样的安全保护装置，例如门互锁开关、安全光幕和安全垫等等。最常用的是机器人单元的门互锁开关，打开此装置可暂停机器人。客户使用的时候需要在机器人控制系统中配置对应的开关信号，将硬件接入的 I/O 端口对应到专用信号上。请参考本手册的 I/O 设置内容。

2.3.4.2 安全监控

紧急停止和安全保护装置受到监控，因此控制器可检测到任何故障，并且在问题解决之前机器人会停止。

2.3.4.3 内置安全停止功能

控制器连续监控硬件和软件功能。如果检测到任何问题或错误，机器人将停止操作，直到问题解决。

2.3.4.4 限制机器人工作范围

机器人工作范围可以通过机械停止、软件功能或结合这二者进行限制。请参阅本手册功能相关的说明文档，了解机器人系统的配置方法。

2.3.5 手动模式的安全事项

2.3.5.1 手动模式简介

在手动模式下机械手的移动处于人工控制下。必须按下三位使动装置来启动机械手的电机，也就是允许移动。

手动模式适用于在创建和验证程序以及调试操纵器系统。

2.3.5.2 操作速度

1. 在手动减速模式下操纵器的移动被限制在 250 mm/s（可以通过设置功能设定该功能是否启用）。
2. 在手动模式下操作操纵器时，操作人员距操纵器很近。操纵工业操纵器可能会产生危险，因此应以可控方式进行。
3. 手动模式下，操纵器以预设速度移动，但必须处于人工控制下。手动模式应仅用于所有人员都位于安全保护空间之外时，而且操作人员必须经过特殊训练，深知潜在的危險。

2.3.5.3 三位使动装置

1. 在手动模式下，机器人的电机由手持示教器上的三位使动装置启动。这样，只有按下使动装置机器人才能移动。
2. 这种“止——动”功能在手动全速模式中步进调试程序时也适用。
3. 三位使动装置设计独特，您必须将按钮按下一半才能启动机器人电机。如果按钮未按下或者完全按下，机器人均不会移动。

2.3.6 自动模式的安全事项

2.3.6.1 自动模式简介

1. 自动模式用于在生产中运行机器人程序。
2. 在自动模式下三位使动装置的安全功能会停用，以便机械手在没有人工干预的情况下移动。

2.3.6.2 处理过程干扰

1. 过程干扰不仅会影响特定的操纵器单元，而且会影响整个系统链，即使问题起因于某个特定单元。
2. 事件链可能会导致操作单个操纵器单元时无法获知危险操作，因此须特别注意这种干扰。执行所有补救措施的人员必须熟知整个生产线，而不仅仅是发生故障的操纵器。

3 安装配线

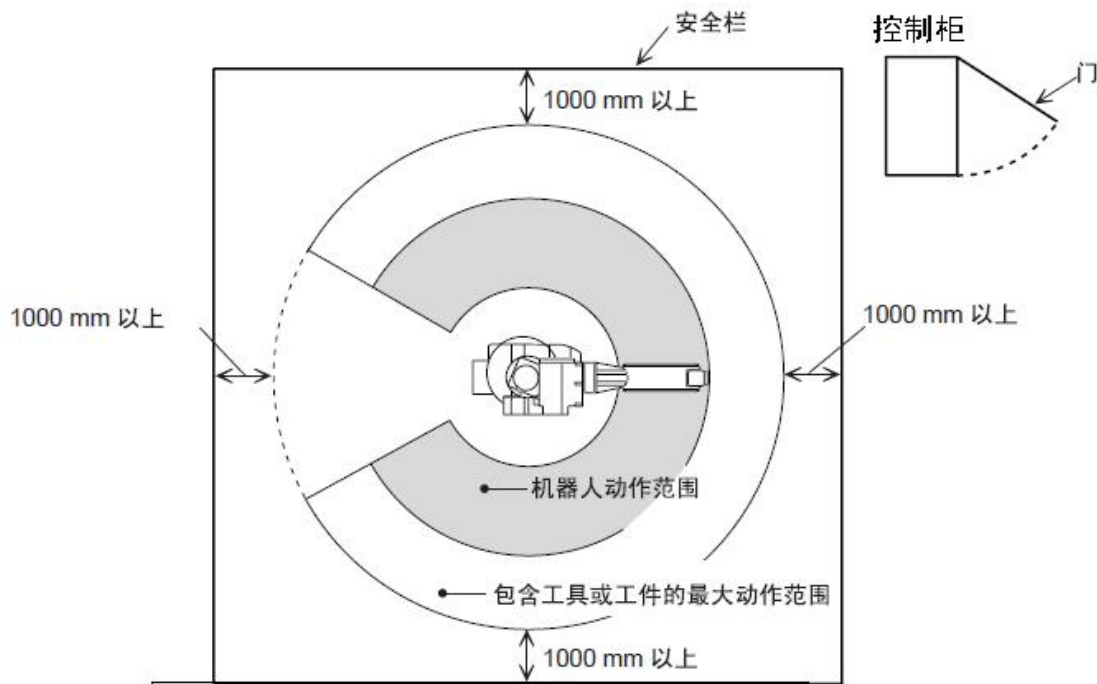
3.1 安装场所和环境

3.1.1 安装环境要求

安装**电控柜**前安装地点必须符合下列条件：

1. 操作期间其环境温度应在 0°至 45°C(32°至 113°F)之间；
2. 搬运及维修期间应为-10°至 60°C(14°至 140°F) ；
3. 湿度必须低于结露点(相对湿度 10% 以下)；
4. 灰尘、粉尘、油烟、水较少的场所；
5. 作业区内不允许有易燃品及腐蚀性液体和气体；
6. 对电控柜的振动或冲击能量小的场所(振动在 0.5G 以下) ；
7. 附近应无大的电器噪音源（如气体保护焊(TIG)设备等）；
8. 没有与移动设备(如叉车)碰撞的潜在危险。

3.1.2 安装位置要求



图表 1 安装位置要求

1. **电控柜**应安装在机器人动作范围之外(安全栏之外)。
2. **电控柜**应安装在能看清机器人动作的位置。
3. **电控柜**应安装在便于打开门检查的位置。
4. 安装**电控柜**至少要距离墙壁 500mm，以保持维护通道畅通。

3.1.3 安装配线注意事项



危险

1. 系统必须电气接地。设备不接地会发生火警或电击，以致造成人身伤害。
2. 系统接地前，应关闭电源并锁住主电源开关。否则可能会造成电击和人身伤害。
3. 在切断电源后的 5 分钟内，不要接触电控柜内的任何基板。电源切断后电容器会储存电能，故无论何时对基板进行操作均应小心，不遵守此警告可能会引起电击。

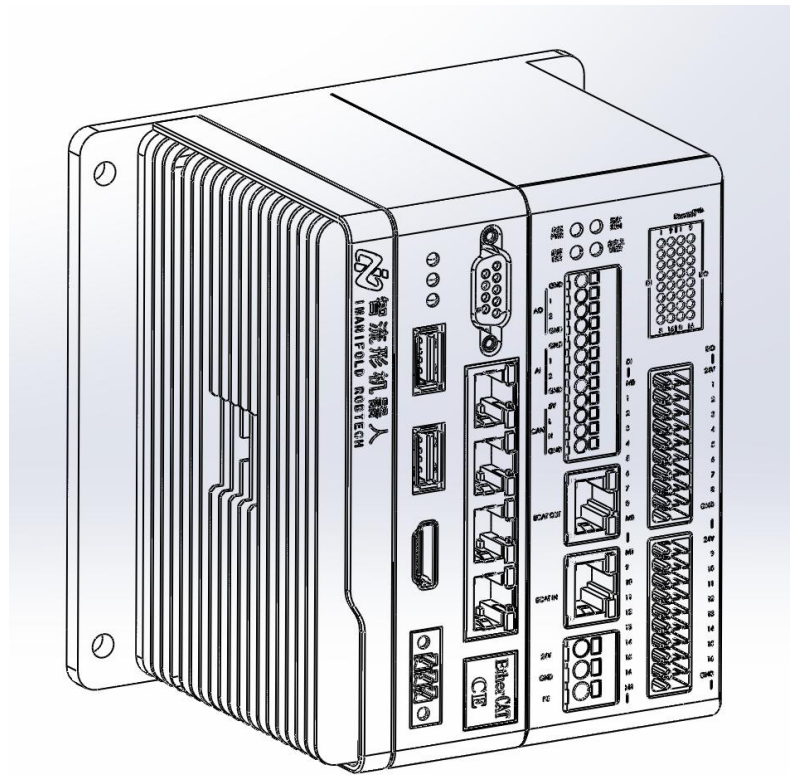
4. 门不关闭则打不开电源，即安全联锁装置阻止打开电源。否则可能会造成火警或电击。
5. 在配线期间，电控柜处于紧急停止的模式下，所发生的任何事由用户负责。一旦配线完成后需要做一次操作检查。否则可能会造成人身伤害或机械故障。



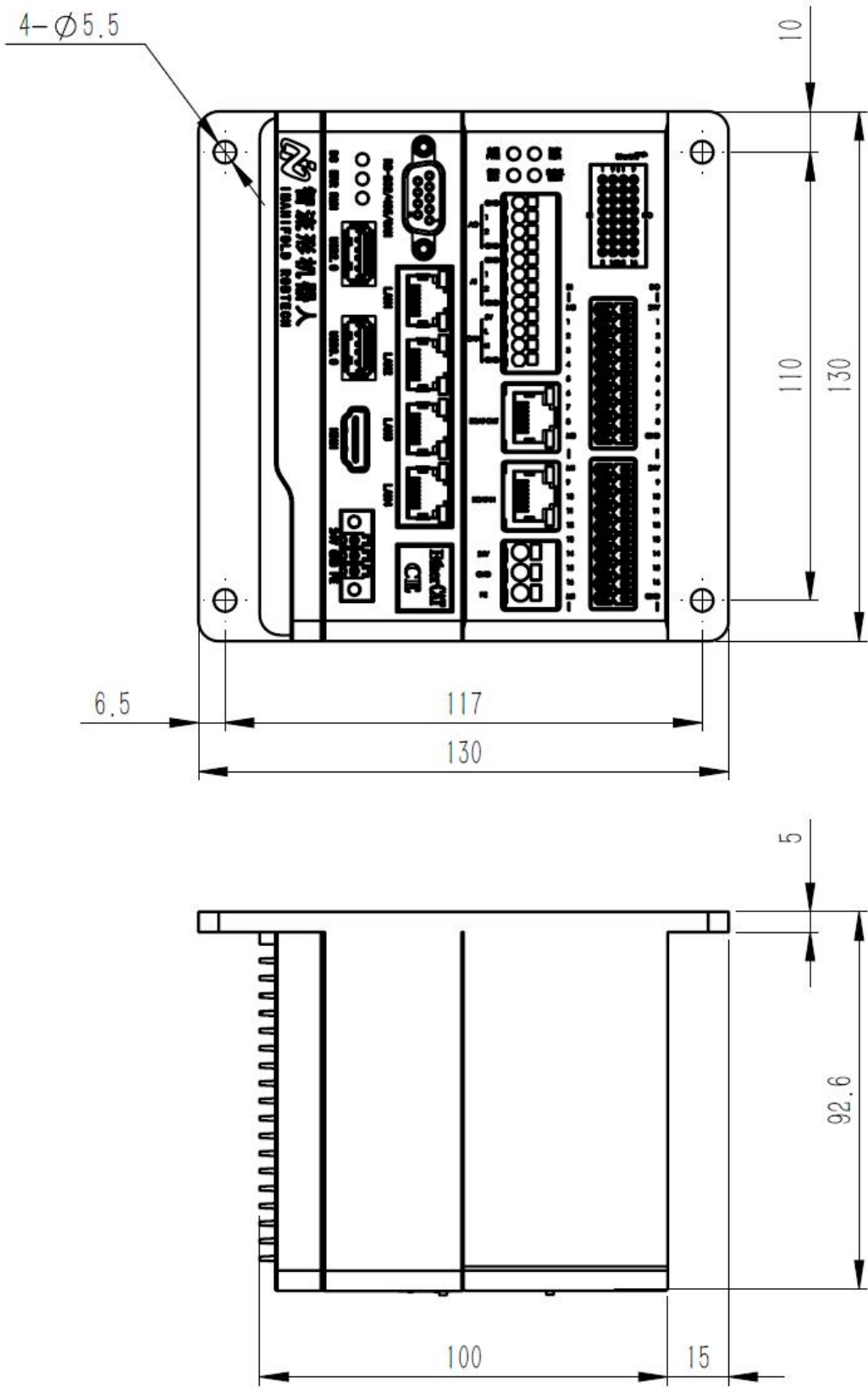
注意

6. 不要爬在电控柜的顶上。否则可能会造成人身伤害或机械故障。
7. 配线须由经授权的工作人员来进行。不正确的配线可能会引起火警或电击。
8. 请按照说明书中规定的额定容量进行配线。不正确的配线可能会引起火警或机械损坏。
9. 确认各电路接线安全牢固。电路接线不牢固会引起火警或电击。
10. 不要用手去直接触摸基板。集成电路(IC)基板可能会由于静电而发生故障。

3.2 控制器安装说明

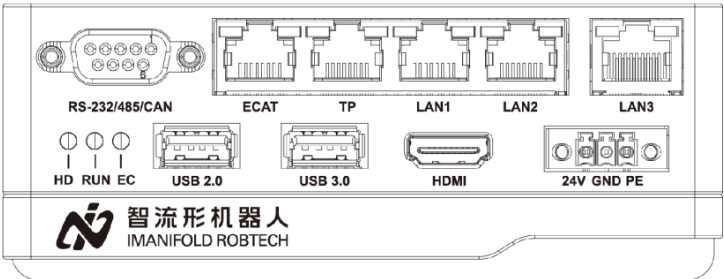


3.2.1 主控及 I/O 单元安装方式



3.2.2 主控单元接线说明

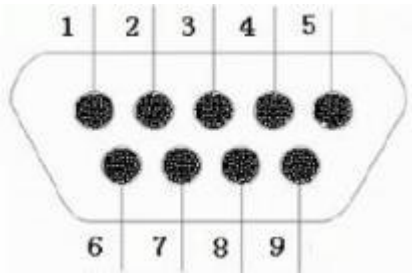
3.2.2.1 主控单元接口说明



图表 2 控制器接口

接口名称	说明
+24V（一定要连接）	给 24V 供电
GND	直流电源地
PE	接可靠性接地及屏蔽端子
TP(接示教器)（一定要连接）	另一端连接手持盒网口，默认 IP： 192.168.100.3
Ecat（一定要连接）	同EtherCAT驱动器电缆连接（标准 网线接口）
LAN1（选用）	默认 IP：192.168.101.3
LAN2（选用）	默认 IP：192.168.102.3
LAN3（选用）	默认 IP：192.168.103.3
RS232/CAN/485 组合口（选 用）	供外部通讯使用
USB2.0/USB3.0（选用）	外接 USB 键盘鼠标或者 U 盘
HDMI（选用）	外接 HDMI 接口显示器

3.2.2.2 RS-232/485/CAN 接口定义



图表 3 串口

1	VCCS_CAN	CAN 电源 24V
2	RS232_RX	串口 232 通信接收
3	CAN_L	CAN 低电平（新版本控制器取消CAN,将CAN接口设计到了I/O模块上）
4	RS485	串口 485 通信低电平
5	GND	接地
6	CAN_GND	CAN 接地（新版本控制器取消CAN,将CAN接口设计到了I/O模块上）
7	RS232_TX	串口 232 通信发送
8	CAN_H	CAN 高电平（新版本控制器取消CAN,将CAN接口设计到了I/O模块上）
9	RS485+	串口 485 通信高电平

3.2.2.3 控制器状态指示灯说明

3.2.2.3.1 状态指示灯 HD:

显示硬盘的读写状态。如果状态指示灯亮起,表示正在进行硬盘写操作。

该状态指示灯由控制器的硬件系统控制,不受上层程序操作。

3.2.2.3.2 状态指示灯 RUN:

状态指示灯 RUN 上电后保持常亮状态 (EtherCAT 驱动未加载),当 EtherCAT 驱动程序启动后,驱动程序将控制该状态指示灯, EtherCAT 驱动操作下的状态指示灯 RUN 显示状态如下:

1. 驱动处于空闲状态 (等待应用程序请求主站) 亮 1s + 灭 1 秒, 交替进行, 即慢闪
2. 总线通讯断开: 亮 300ms + 灭 200ms + 亮 300ms + 灭 1200 毫秒, 交替进行, 即快闪 2 次
3. 应用程序请求主站进入 OP 模式 (PDO 交互通讯) 亮 100ms + 灭 100ms, 交替进行, 即快闪

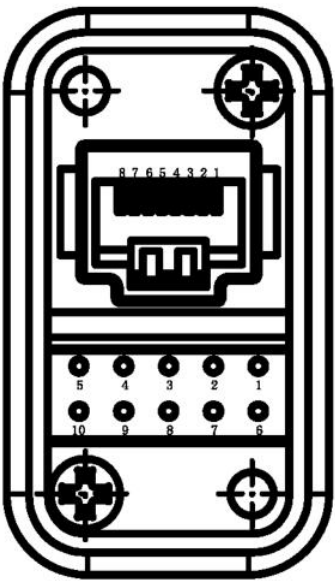
3.2.2.3.3 状态指示灯 EC:

状态指示灯 EC 上电后保持常亮状态, EtherCAT 接口程序启动后将自动操作该状态指示灯, 自动操作模式下

1. 系统未注册: 亮 200ms + 灭 200ms + 亮 800ms + 灭 800ms, 交替进行, 即快闪+慢闪
2. 网络断开或网卡状态异常: 亮 200ms + 灭 200ms + 亮 200ms + 灭 1400ms, 交替进行, 即快闪 2 次
3. 主站请求忙碌中 (主站正在扫描从站): 亮 1s + 灭 1 秒, 交替进行, 即慢闪

4. 主站 Ready (PDO 通讯已建立)：亮 100ms + 灭 100ms，交替进行，即快闪
5. 其它状态 (主站配置过程中)：亮 200ms + 灭 200ms + 亮 200ms + 灭 200ms+ 亮 200ms + 灭 1000ms，交替进行，即快闪 3 次应用程序退出后，状态指示灯 ERR 显示不确定，可能常亮，也可能常灭。

3.2.3 示教器安装接线



图表 4 示教器航插

示教器外接线缆定义：

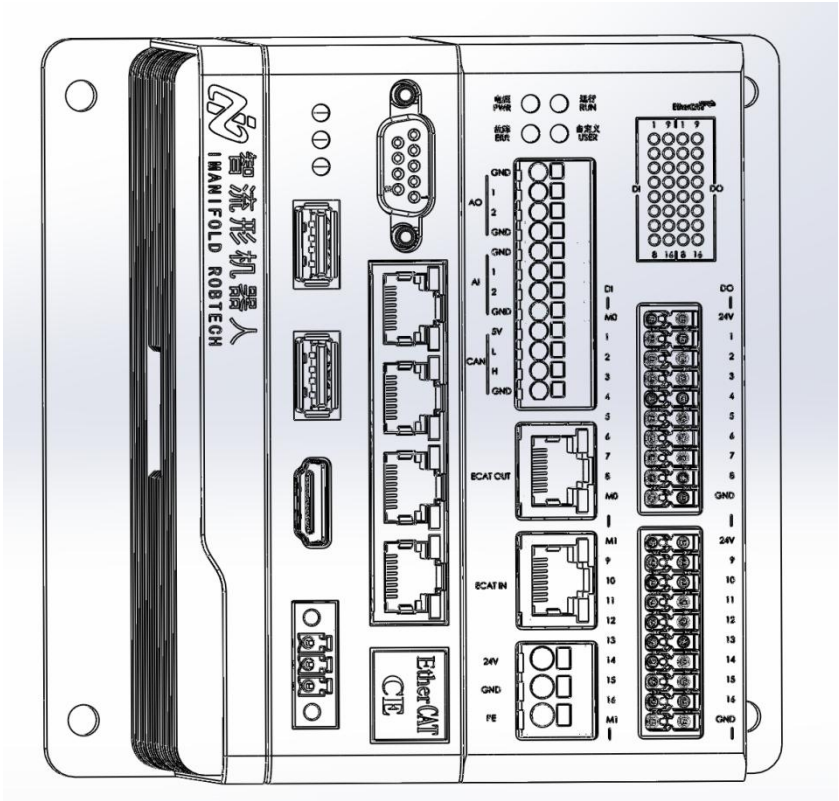
1	红	24V+	接 24V 电源+
2	白红	0V	接 24V 电源 0V
3	黑	PE	接地
4	绿	EMG1-1	急停端子 1
5	黑绿	EMG1-2	急停端子 1
6	蓝	EMG2-1	急停端子 2

7	黑蓝	EMG2-2	急停端子 2
---	----	--------	--------

(具体以线号为准)

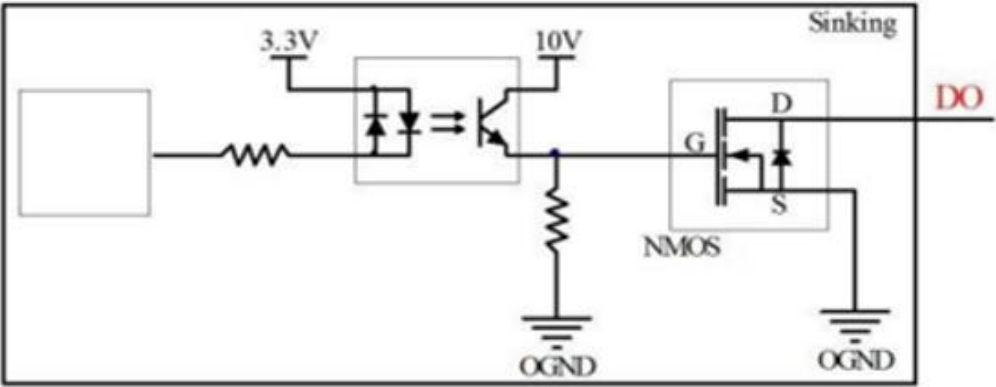
手持盒必须接 24V 供电接线，急停端子可以根据需要选择使用，急停端子提供两组接线输出，连通手持示教器急停信号，默认常闭，按下手持示教器急停按钮，打开。

3.2.4 IO 模块接线定义

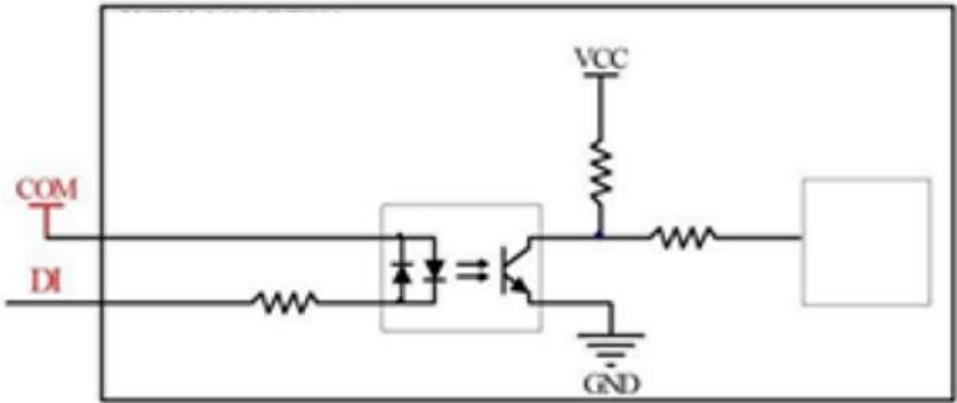


焊接机器人专用EtherCAT从站I/O模块

额定电源：24VDC,+-20%				
IO类别	通道数	信号类型	信号电平	备注
DI	16	兼容PNP/NPN	0V/24V,500mA	"1" 信号电平：15~24VDC； "0" 信号电平：0-5VDC；
DO	16	NPN	0V,500mA	"1" 信号电平：24VDC； "0" 信号电平：Max:1.5VDC,10kΩ负载；
AI	2	电压	0-10V	单极、12BIT、二线制，输入阻抗500K欧；
AO	2	电压	0-10V	单极、12BIT、二线制，最小负载阻抗1K欧，最大电流10mA；
CAN	1	信号	5V	CAN-5V,CAN-L,CAN-H,CAN-GND



图表 5 漏型（底边）输出电路示意图



图表 6 通用输入电路示意图

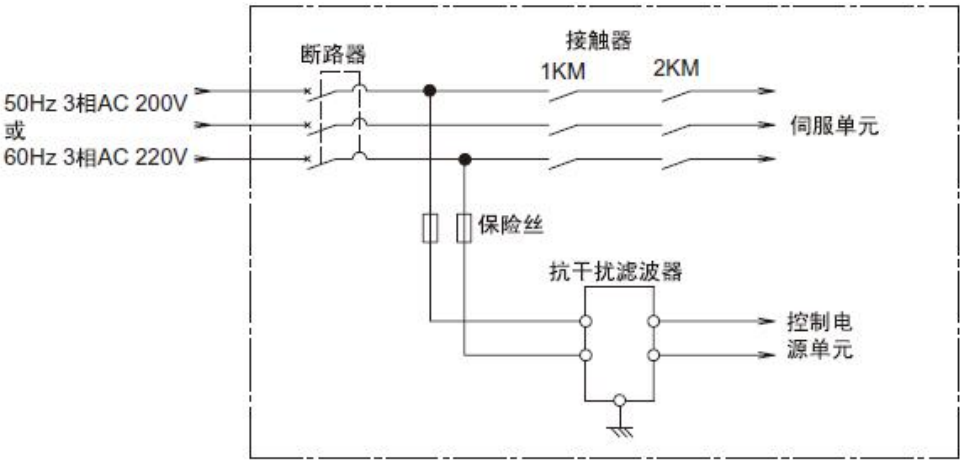
3.3 供电电源

3.3.1 三相电源

三相电源是由交流 200V,50Hz 或 交流 220V,50/60Hz 组成。

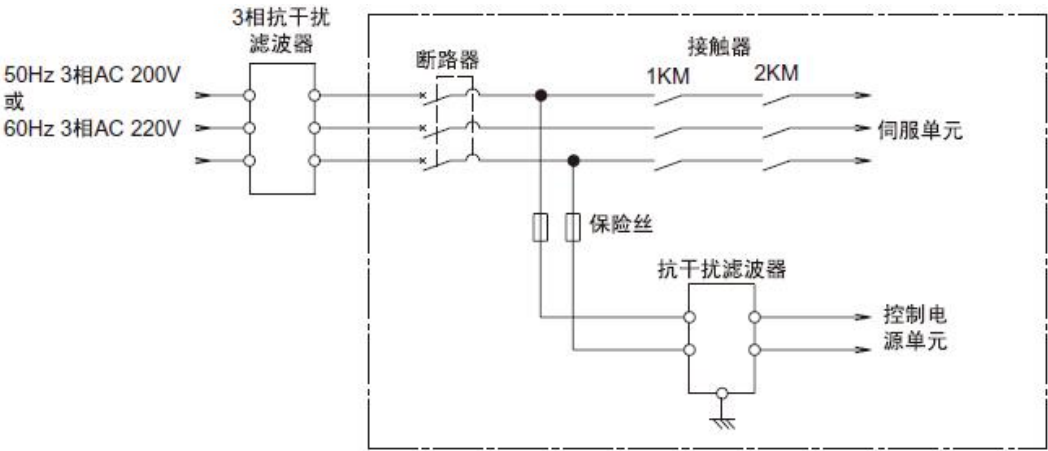
当存在有临时性的电源频率中断或电压下降时，停电处理电路将产生动作使得伺服电源切断。

将电控柜电源连接到一个电压波动小的稳定输入电源上去。



图表 7 电源的输入连接

3.3.2 噪声滤波器的安装

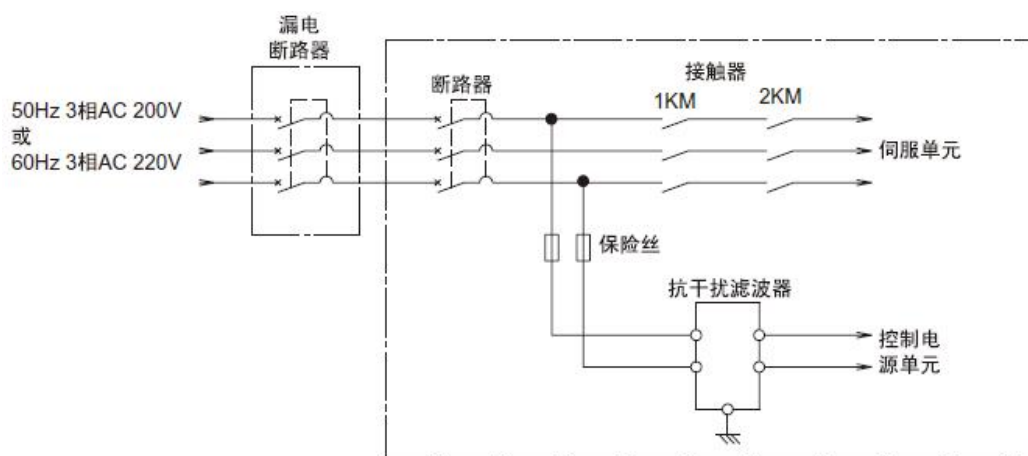


图表 8 三相噪声滤波器的连接

3.3.3 漏电断路器的安装

如果给电控柜电源连接漏电断路器，需使用可防止高频的漏电断路器，它能防止整流器的高频漏电流引起的误动作。

即使安装了漏电断路器，NX100的整流器仍存在有高频电流泄漏的可能性，但此种电流的泄漏没有危险性。



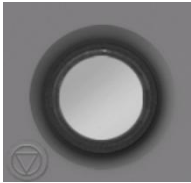
图表 9 漏电断路器的连接

4 手持操作示教器

本章介绍机器人手持操作示教器布局及按键功能。



4.1 示教器键位功能

序号	名称	按键	功能
1	急停		按下此键，伺服电源切断。 切断伺服电源后，手持操作示教器的【伺服准备指示灯】熄灭，屏幕上显示急停信息。 故障排除后，可打开急停键，急停旋钮打开后方可继续接通伺服电源。 此键按下后将不能打开伺服电源。 打开急停键方法：顺时针旋转至急停键弹起，伴随“咔”的声音，此时表示【急停按钮】已打开。
2	暂停		按下此键，机器人暂停运行。 此键在任何模式下均可使用。 示教模式下：此灯被按下时灯亮，此时机器人不能进行轴操作。 回放模式下：此键指示灯按下一次后即可进入暂停模式，此时暂停指示灯亮起，机器人处于暂停状态。按下手持操作示教器上的【启动】按钮，可使机器人继续工作。
3	启动		按下此按钮，机器人开始回放运行。 回放模式运行中，此指示灯亮起。

			通过专用输入的启动信号使机器人开始回放运行时，此指示灯亮起。 按下此按钮前必须把模式旋钮设定到回放模式；确保手持操作示教器【伺服准备】指示灯亮起。
4	模式旋钮		可选择回放模式、示教模式或远程模式。 示教（TEACH）：示教模式 可用手持操作示教器进行轴操作和编辑（在此模式中，外部设备发出的工作信号无效）。 回放（PLAY）：回放模式 可对示教完的程序进行回放运行。 远程（REMOTE）：远程模式 可通过外部 TCP/IP 协议、IO 进行启动示教程序操作。
5	三段开关		按下此键，伺服电源接通。 操作前必须先把模式旋钮设定在示教模式→点击手持操作示教器上【伺服准备】键（【伺服准备】指示灯处于闪烁状态）→轻轻握住三段开关，伺服电源接通（【伺服准备】指示灯处于常亮状态）。 此时若用力握紧，则伺服电源切断。 如果不按手持操作示教器上的【伺服准备】键，即使轻握【三段开关】，伺服电源也无法接通。
6	手脉旋钮		滚轮上下旋转切换下拉框内容、工程/程序及程序行，向下按表示确认选择。

7	伺服准备		按下此键，伺服电源有效接通。 由于急停等原因伺服电源被切断后，用此键有效地接通伺服电源。 回放模式和远程模式时，按下此键后， 【伺服准备】指示灯亮起，伺服电源被接通。 示教模式时，按下此键后，【伺服准备】指示灯闪烁，此时轻握手持操作示教器上【三段开关】，【伺服准备】指示灯亮起，表示伺服电源被接通。
8	上档		可与其他键同时使用。
9	菜单		按下此键，打开系统应用菜单 程序：程序管理、文件编辑 变量：全局变量、全局位置、全局增量、工程变量、工程位置、工程增量 机器人：坐标系管理、零位标定、异常处理 设置：权限管理、系统参数 自定义：工艺

10	连锁		辅助键，与其他键同时使用。 此键必须使用在示教模式下。
11	功能键 1 状态显示		按下此键，打开状态显示窗口。 查看机器人当前位置、数字量 IO、模拟量、多任务、速度、变量、工作空间及部分工艺参数。
12	功能键 2 浮动菜单		按下此键，打开浮动菜单 浮动菜单可以切换坐标系、切换回放模式、切换机器人/变位机、切换速度及部分工艺菜单。
13	轴操作键		对机器人各轴进行操作的键。 此键组必须使用在示教模式下。 可以按住两个或更多的键，操作多个轴。 机器人按照选定坐标系和手动速度运行，在进行轴操作前，请务必确认设定的坐标系和速度是否适当。 操作前需确认机器人手持操作示教器上的【伺服准备】指示灯亮起。
14	前进后退		伺服电源接通状态下，按住此键时，机器人按示教的程序点轨迹单步（逆向）运行。 此键必须使用在示教模式下。

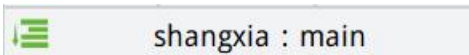
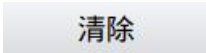
15	加减速速度		机器人示教模式运行速度的设定键。 低速度档位有 1%、 2%、 5%，往上每次按键+/-5%。 被设定的速度显示在状态栏。 【上档】 + 【V+】 / 【V-】 可以调整回放模式速度。

4.2 示教器界面



图表 10 示教器界面

界面中菜单、按钮、标识功能：

	每个菜单和子菜单都显示在系统应用菜单，通过按下手持操作示教器上【菜单】键，或点击界面左上角的LOGO 按钮，显示系统应用菜单。
	显示当前伺服、运行模式、坐标系状态。
	显示当前工具坐标系、工件坐标系序号； 显示速度，前者是示教模式速度，后者是回放模式速度
	显示执行循环模式（循环、单次、单步），当前工程名及程序名
	示教器连接状态、当前权限、当前坐标系和其他工艺状态
	通过此按钮可以查看界面 log，历史报警信息
	进行错误和操作提示或报警，无报警显示正常，显示运动速度
	清除报警按钮
	常用应用工具栏(可自定义)

4.3 系统应用菜单



图表 11 菜单界面

4.3.1 常用工具栏自定义

系统菜单栏有两种操作模式，点击或可相互切换，在拖动模式可以选择常用的界面图标，通过拖动放到常用工具栏，便于快速切换界面。

图标	功能	说明
	点击模式	该模式下，点击菜单栏其他小图标可进入对应界面
	拖动模式	该模式下，拖动可以拖动菜单栏小图标到常用应用工具栏 操作方法： 1. 切换到拖动模式 2. 按住小图标拖到常用应用工具栏，松开图标 

4.3.2 菜单选项说明

图标	名称	功能
	工程管理	1. 新建工程； 2. 工程管理：导入、导出、重命名、另存为、删除选择的工程； 3. 新建程序：在选择工程下新建一个程序； 4. 程序管理：导入、导出、重命名、另存为、删除选择的程序； 5. 打开程序；

	文件编辑	对当前工程下的所选程序进行编辑。
	全局变量	全局数值变量和全局字符串，可在不同的工程中供程序编辑时使用。
	全局位置	可以标定位置型变量，可在不同的工程中供程序文件编辑时使用。
	全局增量	可以写入位置型增量，可在不同的工程中供程序文件编辑时使用。
	工程变量	工程数值变量和工程字符串， 只能在当前打开的工程下编辑使用
	工程位置	可以标定位置型变量，只能在当前打开的工程下编辑使用
	工程增量	可以写入位置型增量，只能在当前打开的工程下编辑使用
	坐标系管理	标定、修改世界坐标系、工具坐标系和工件坐标系。 设置当前使用的坐标系，对应序号在界面右上角状态栏中显示
	零点标定	对机器人的零点进行标定、手动修改、绝对编码器清零
	异常处理	轴仿真设置 轴运动限制开关
	HOME 坐标	设置原位置坐标（回零）

	后台任务	后台任务设置，预留功能
	权限管理	设置管理员权限，不同权限存在不同的操作内容。 版本信息
	系统参数	设置：系统参数、硬件参数、关节、辅助轴、笛卡尔、 规划参数、DH 参数
	未定义	工艺参数设置

4.4 状态栏

4.4.1 显示伺服状态

	伺服关
	伺服开

4.4.2 工作模式

显示机器人的工作模式，通过手持操作示教器上的模式旋钮切换。

	示 教：机器人处于示教工作模式下。
	回 放：机器人处于回放工作模式下。
	远 程：机器人处于远程工作模式下。

4.4.3 坐标系

显示被选择的坐标系，通过界面浮动菜单的{坐标系}选择。浮动菜单通过手持操作示教器上的【fn2】键弹出。

	关节坐标系
	机器人坐标系
	工件坐标系
	世界坐标系
	工具坐标系

4.4.4 当前工具号，工件号

方便用户确定当前使用的工具（工件）序号，程序内部使用一个具有 32 个元素的工具（工件）坐标系数数据队列，默认 1 号为不使用工具（工件），2--32 号坐标系队列元素为可编辑的队列元素。

4.4.5 速度

显示当前速度，通过按手持操作示教器上的【V+】或【V-】键加减速。前者为示教速度，后者为回放速度

4.4.6 示教文件回放执行模式

	无限循环：不停的循环执行示教程序；
---	-------------------

	单次循环：只执行一遍示教程序；
	单行运行：每次只执行一条教程序。

4.4.7 当前权限和网络状态

4.4.7.1 权限等级

在系统应用菜单-权限管理中更改

		
操作模式	管理模式	出厂设置

4.4.7.2 网络状态

	
示教器在线	示教器离线

4.5 浮动菜单

浮动菜单通过手持操作示教器上的【fn2】键弹出/关闭。快捷切换功能。

第一级	第二级	功能说明
	 jog 坐标系切换	切换到关节坐标系
		切换到机器人坐标系
		切换到工件坐标系
		切换到世界坐标系
		切换到工具坐标系
	 回放程序执行模式	无限循环
		单次执行
		单行执行
	 jog 轴切换	切换到机器人轴 (此时手持【J1】 - 【J8】 代表机器人的轴)
		切换到辅助轴 (此时手持【J1】 - 【J8】 代表外部辅助轴)

	速度百分比切换 1% 5% 25% 50% 75% 100%	切换当前模式的速度百分比
--	---	--------------

4.6 机器人状态及报警历史查看

4.6.1 机器人状态

状态	说明
正常	! ⏸ *****正常***** 清除 机器人处于正常运行状态，此时可操作机器人上伺服运动，执行示教程序等操作。
伺服准备	! ✓ *****伺服准备***** 清除 机器人处于伺服上电状态
执行中	! ✓ *****执行中***** 0000.00mm 清除 机器人正在运动或者正在执行示教程序，后面的数字表示当前机器人末端的直线速度
黄色警告	! ⚠ "TCP socket error (Connection timed out)" 清除 机器人处于报警状态，该状态不会导致机器人下伺服或者急停，需要操

	作员根据报警内容检查，排除报警原因
红色报警	 emergency stop 清除 机器人处于严重报警状态，该状态会导致机器人急停并下私服，在报警未解除前机器人不能进入伺服准备状态

有错误信息或报警时，中间显示区变为黄色或红色并显示错误报警信息。按下右边的{清除}键，可清除错误。

4.6.2 机器人报警历史及操作 log 查看

可以通过左边的按钮可查看出现过的所有报警信息记录和界面 log（如下图）。

项目	说明
界面 log	记录历史界面操作及报警 界面Log报警历史  2020-12-01 10:18:56 QSqlDatabasePrivate::database: unable to open database: "Can't connect to MySQL server on '192.168.0.30' (10060) QMYSQL: Unable to connect"  2020-12-01 10:18:38 "TCP socket error (Connection timed out)."  2020-12-01 10:17:56 "TCP socket error (Connection timed out)."  2020-12-01 10:17:14 "TCP socket error (Connection timed out)."  2020-12-01 10:16:32 "TCP socket error (Connection timed out)."  2020-12-01 10:15:50 "TCP socket error (Connection timed out)." 删除log
报警历史	记录系统报警历史



5 快速操作入门

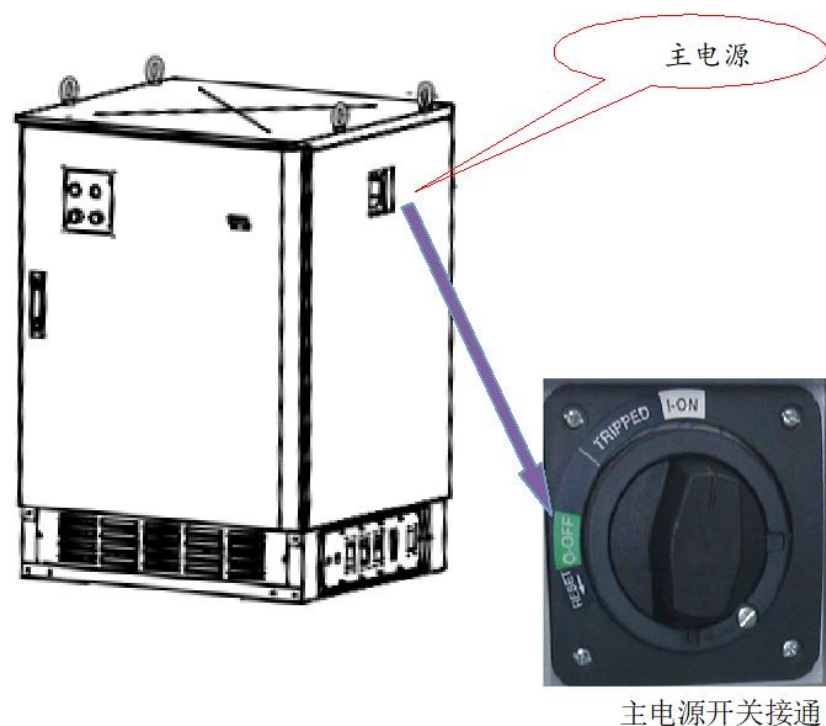
5.1 伺服电源接通



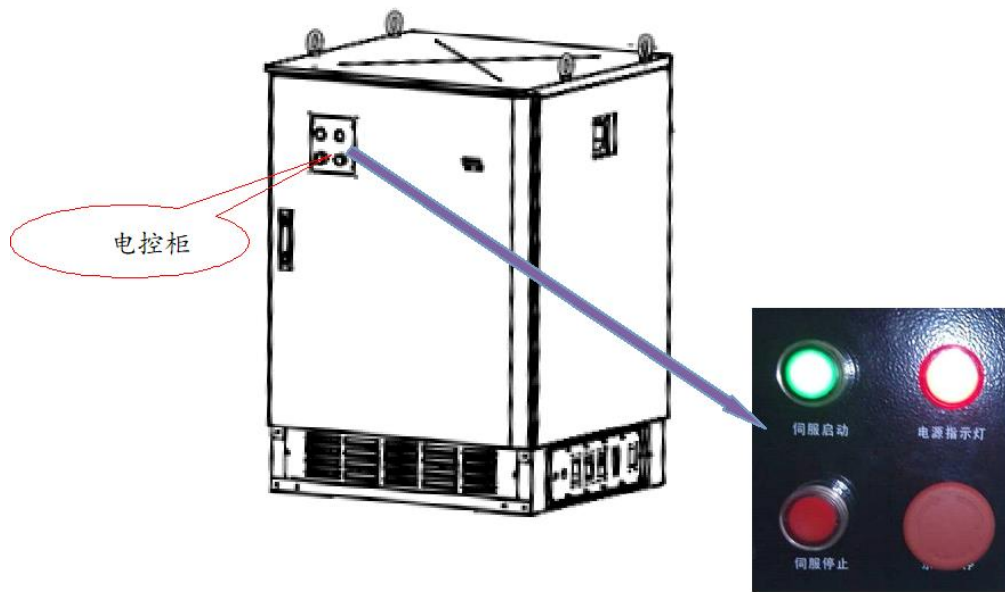
打开上电控柜上的主电源开关时，应确认在机器人动作范围内无任何人员。忽视此提示可能会发生与机器人的意外接触而造成人身伤害。如有任何问题发生，应立即按下急停键，急停键位于电控柜前门的左上方。

5.1.1 接通主电源

把电控柜侧板上的主电源开关扳转到接通(ON)的位置，此时主电源接通。按下电控柜面板上的绿色伺服启动按钮。



图表 12 主电源开关



图表 13 电控柜面板

5.1.2 接通伺服电源

示教模式、回放模式和远程模式的伺服电源接通步骤是不同的。

1. 示教模式下：

按下手持操作示教器上的【伺服准备】键，这时【伺服准备】指示灯处于闪烁状态；轻握手持操作示教器背面的【三段开关】，这时手持操作示教器上的【伺服准备】指示灯长亮，表示伺服电源接通。

2. 回放和远程模式下：

按下手持操作示教器上的【伺服准备】键，这时手持操作示教器上的【伺服准备】指示灯长亮，表示伺服电源接通。

5.2 伺服电源切断

5.2.1 切断伺服电源

示教模式和回放模式、远程模式的伺服电源切断步骤是不同的。

1. 示教模式下:

释放或用力握紧手持操作示教器背面的【三段开关】，这时手持操作示教器上的【伺服准备】指示灯熄灭，表示伺服电源切断。

2. 回放和远程模式下

再次按下手持操作示教器上的【伺服准备】键，这时手持操作示教器上的【伺服准备】指示灯熄灭，表示伺服电源切断。

3. 按下电控柜面板上的急停键

一旦伺服电源切断，则制动装置启动，机器人就被制动而不能再进行任何操作。可在任何模式（示教模式、回放模式或远程模式）下的任何时候进入紧急停止状态。

5.2.2 切断主电源

切断伺服电源后，再切断主电源。

把电控柜侧门上的主电源开关扳转至切断(OFF)的位置，则主电源被切断，如下图所示：



图表 14 旋钮开关

5.3 坐标系的选择

在示教模式下，通过浮动菜单选择机器人运动坐标系。

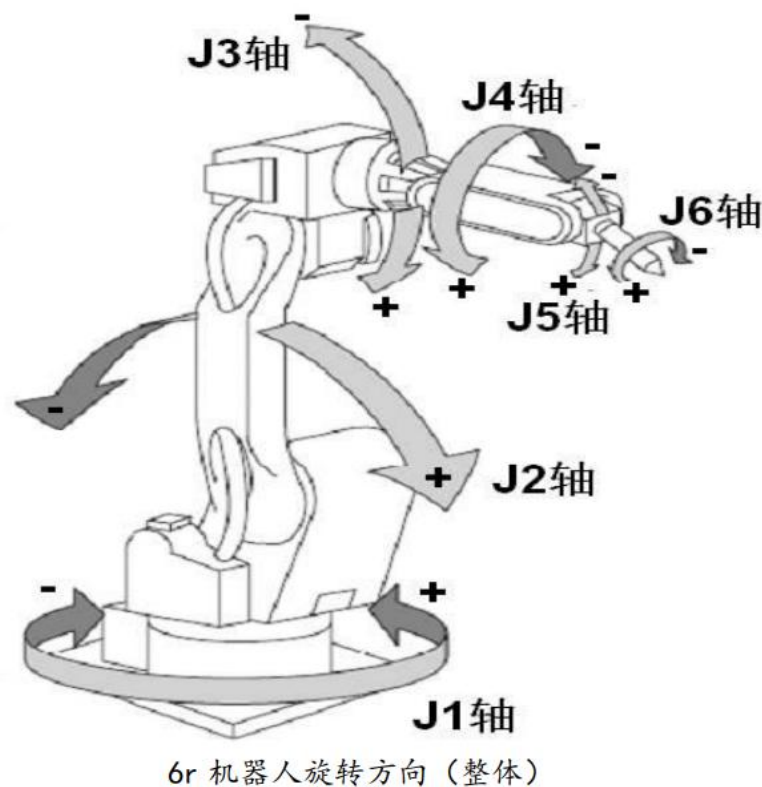
5.4 手动速度调整

示教模式下，修改点动 JOG 机器人运动速度：

按手持操作示教器上【V+】键或【V-】键，每按一次，手动速度 ± 5 ，通过状态区右上角的速度显示来确认。（第一个值为示教速度）

5.5 轴操作

示教模式下，按下**轴操作键**，机器人各轴可移动至所希望的位置，各轴的运动根据所选坐标系而变化。



每个轴只能在按住**轴操作键**时运动。



1. 操作机器人前请先确认急停键可以正常工作：按下电控柜上的急停键，【伺服

准备】指示灯熄灭说明急停键正常。如果机器人不能在紧急情况下停止，则可能会引起机械的损坏。

2. 当在机器人动作范围内进行示教工作时，则应遵守下列警示：

- 始终从机器人的前方进行观察；
- 始终按预先制定好的操作程序进行操作；
- 始终具有一个当机器人万一发生未预料的动作而进行躲避的想法；
- 确保您自己在紧急的情况下有退路。

不适当地和不认真地操作机器人会造成伤害。

3. 在执行下列操作前，确认在机器人动作范围内应无任何人员，并确保您自己处在一个安全的位置区内。

- 接通电控柜的电源时。
- 用手持操作示教器操作机器人时。
- 回放时。
- 远程时。

如果机器人与进入动作范围内的任何人员发生碰撞，将会造成人身伤害。

4. 示教**机器人**前先执行下列检查步骤，如发现问题则应立即更正，并确认其他所有必须作的工作均已完成。

- 查机器人的运动有无异常的问题；
- 检查外部电缆的绝缘及遮盖物是否损坏。

伺服电源接通后（按下【伺服准备】键后，握住【三段开关】，此时伺服指示灯常亮），通过按手持操作示教器上的每个轴操作键，使机器人的每个轴产生所需

的动作。下图表明了每个轴在关节坐标系下的动作示意。

请操作机器人前注意关节运动速度状态，通过高低速按键调节至适当速度。

5.6 平齐

平齐功能只能在示教模式下使用。

平齐功能是绕某一个轴的旋转运动的旋转分量自动平齐到 90 度的整数倍的位置，例如 -180° ， -90° ， 0° ， 90° ， 180° 等。限制条件为欧拉角必须在 15 度范围内。

当角度小于 15 度范围时，按手持操作示教器上**连锁键 + 【A+】键**，选定坐标系的绕 X 轴的旋转分量平齐到 90° 的整数倍；**连锁键 + 【B+】键** 选定坐标系的绕 Y 轴的旋转分量平齐到 90° 的整数倍；**连锁键 + 【C+】键** 选定坐标系的绕 Z 轴的旋转分量平齐到 90° 的整数倍。

最终通过浮动菜单选择机器人运动坐标系来确认是否到达 90° 的整数倍位置。

5.7 回零

示教模式下，设置原位置坐标，上伺服后，按手持操作示教器上**连锁键 + 前进键**可运动到该点。

该点应记录为机器人在任何位置均可安全到达的位置。

机器人HOME点位置

注意：

1.示教模式下，上伺服后，按【连锁】+【前进】可运动到该点；
2.该点应记录为机器人在任何位置均可安全到达的位置。

J1	0.0000	J5	0.0000
J2	0.0000	J6	0.0000
J3	0.0000	J7	0.0000
J4	0.0000	J8	0.0000

修改

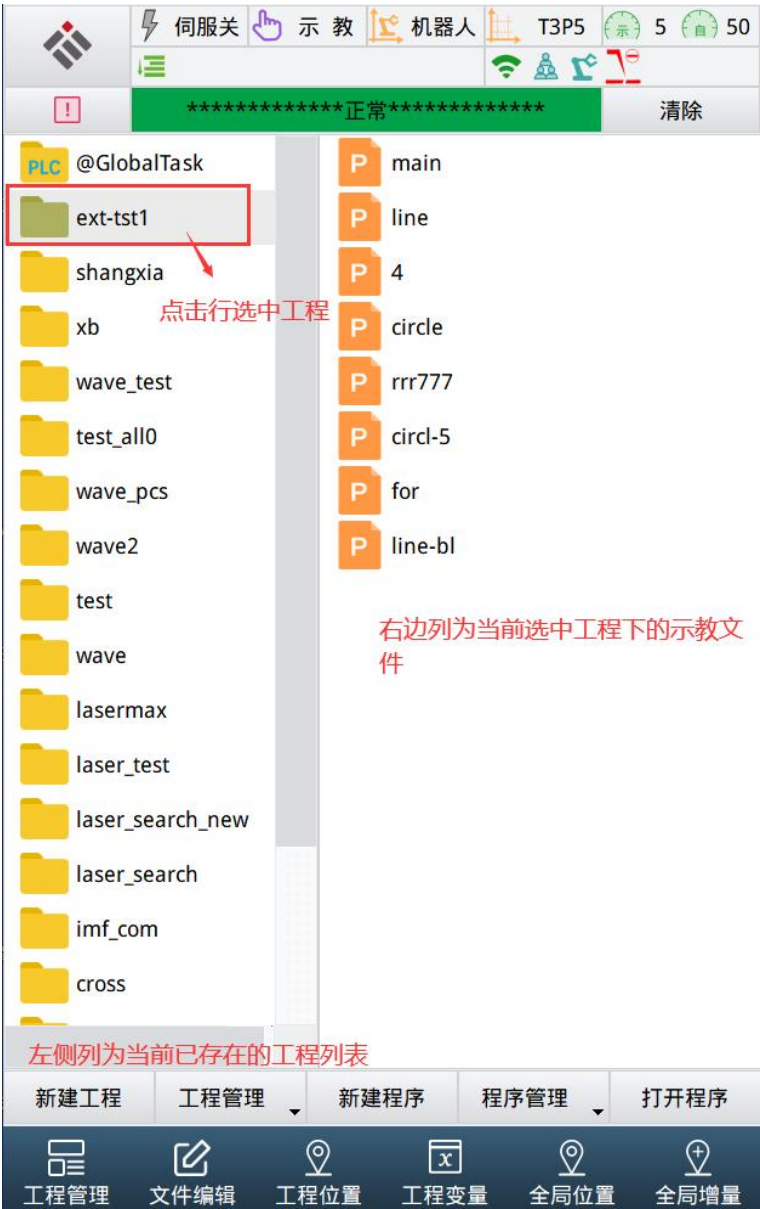
记录

图表 15 HOME 点设置

若未设置 HOME 点位置，则默认为机器人出厂零点位置。

6 工程管理界面

在主菜单选择或者点击常用工具栏的【工程管理】进入工程管理界面。

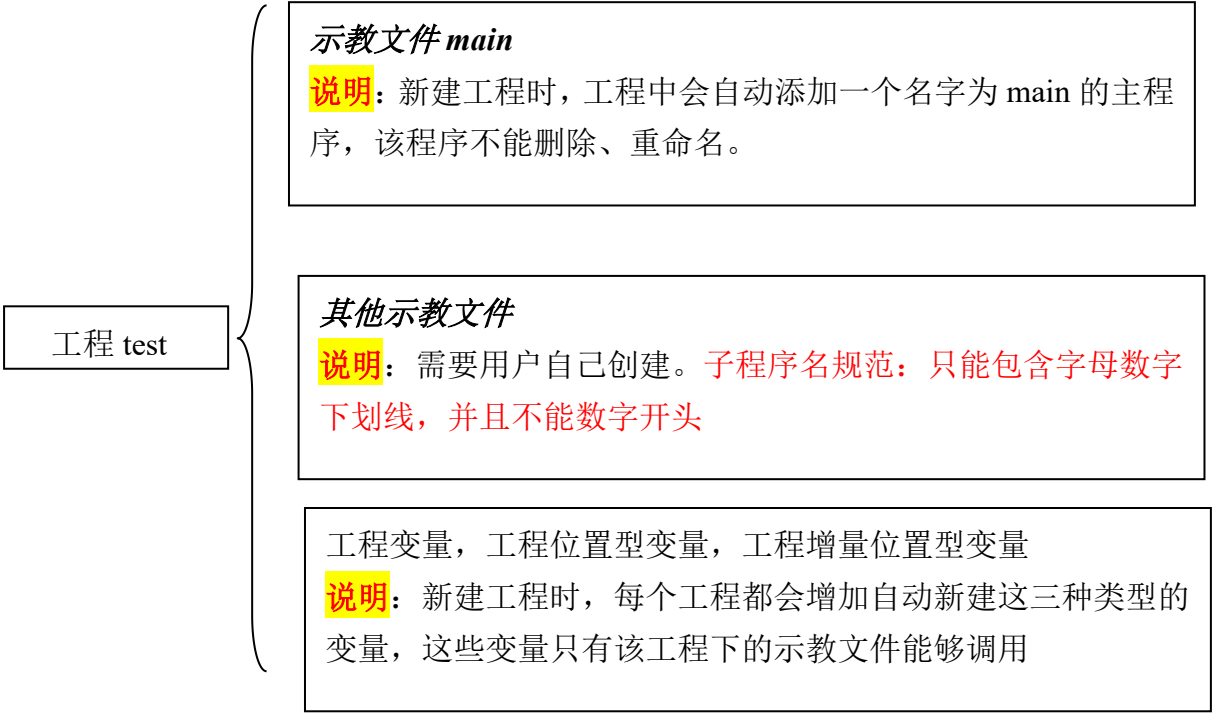


图表 16 工程管理界面

6.1 机器人工程结构说明

机器人程序以工程为单位，每位工程可以有多个示教文件，同时每个工程都包含只有该工程内的示教文件才能使用的变量。以名称为“test”的工程为例结构下图：

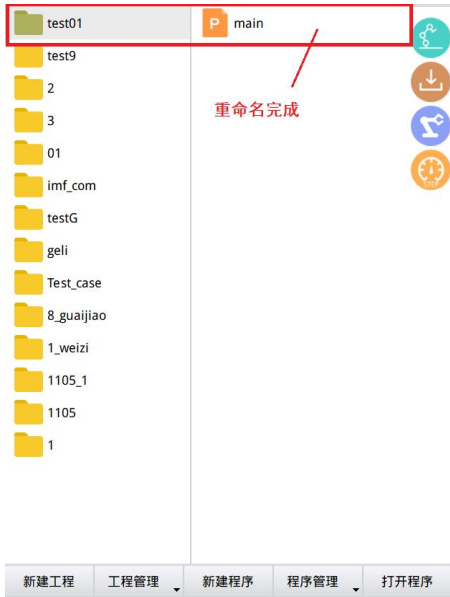
- 示教文件 main 和其他示教文件均可执行；
- **示教文件main和其他示教文件的区别**: 示教文件main中可以调用其他示教文件；其他示教文件可相互调用，但是不能调用示教文件main。

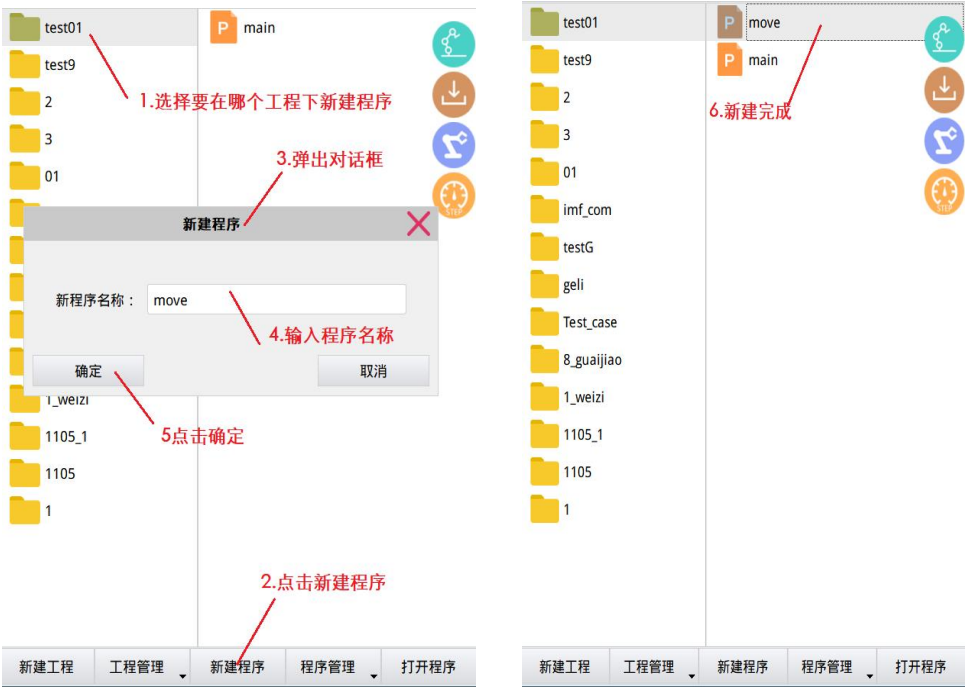


6.2 工程操作

以下所讲的是能够在机器人示教模式下进行的程序编辑操作。

项目	说明
1. 新建工程	
2.工程 重命名	

	
3.工程 另存为	另存为是将整个工程重新复制一份，操作步骤参考 工程重命名
4.工程删除	删除工程会将工程下的所有示教文件及变量文件都删除 

5.新建程序	
6.程序重命名	程序重命名需要先选中需要操作的程序，然后选择【程序管理】，点击【重命名】其他操作步骤查看工程重命名
7.程序另存为	程序另存为需要先选中需要操作的程序，然后选择【程序管理】，点击【另存为】其他操作步骤查看工程重命名
8.程序删除	程序删除需要先选中需要操作的程序，然后选择【程序管理】，点击【删除】其他操作步骤查看工程删除

打开程序



1.选择要打开的程序

2.点击打开程序

新建工程 工程管理 新建程序 程序管理 打开程序



test01 : main

*****正常*****

1 PROGRAM

2 ENDPROGRAM

当前打开的程序:

当前打开的程序为test01工程下的main程序

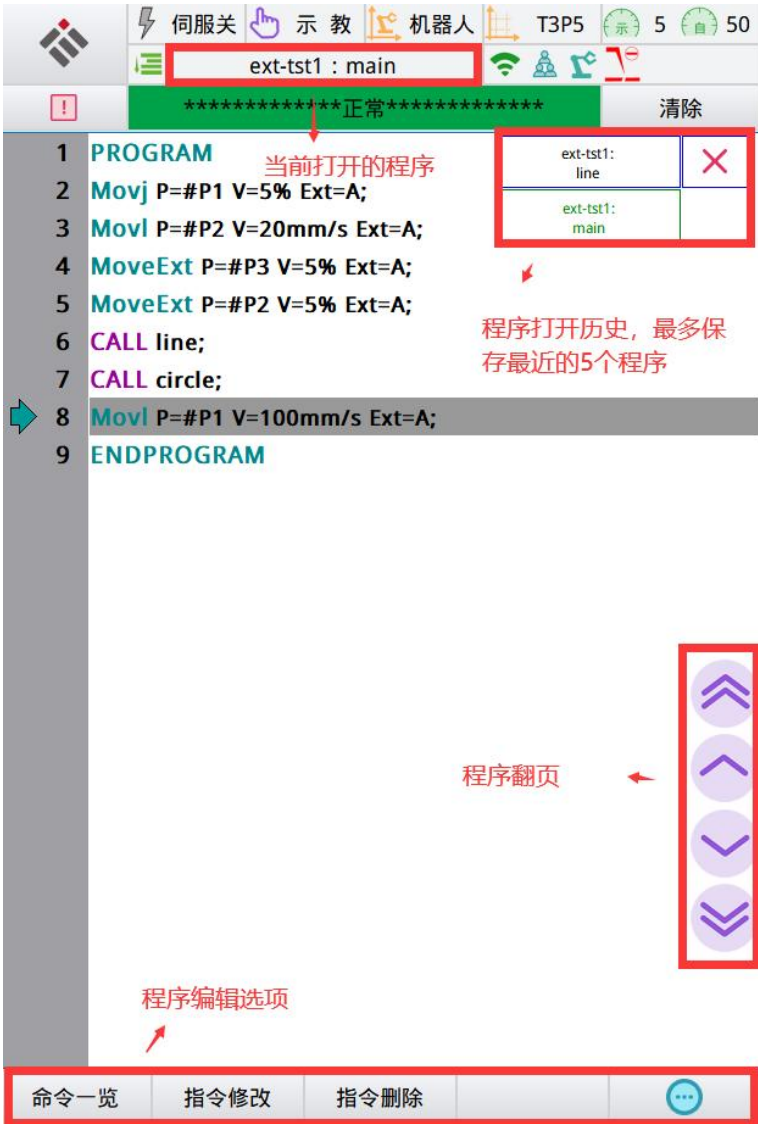
命令一览 指令修改 指令删除

当前打开的程序只有两行

点击打开后进入 程序编辑 界面

7 程序编辑界面

点击菜单栏的或者点击常用工具栏的【文件编辑】进入程序编辑界面



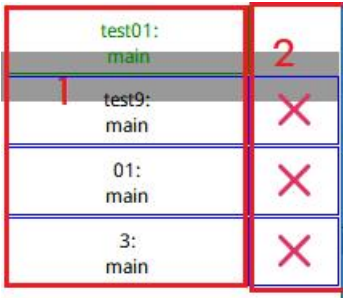
图表 17 程序编辑界面

7.1 当前打开的程序

当前打开的程序名称会显示在状态栏，例如 test01: main，test01 为工程名称，main 为该工程下的示教文件名称。

7.2 历史程序

界面会显示最近打开过的 5 个文件名称



图表 18 历史程序界面

项目	功能说明
1.快速切换历史程序	点击左边 1 列的对应行，可快速切换到对应程序， 绿色行为当前打开的程序
2.删除历史记录	点击右边 2 列的对应  ，可删除对应行的历史程序记录

7.3 程序翻页

图标	功能说明
	到达程序顶端
	向上翻页
	向下翻页
	到达程序末端

7.4 程序编辑

功能	功能说明
----	------

1.示教指令
插入
(向选中行的
下一行插入新
指令)

0.选中插入行
(向选中行的下一行插入新指令)

1.点击命令一览

2.选择指令类型

3.选择要插入的指令名称

4.弹出指令编辑框

5.编辑指令参数

6.点击确认插入

1.点击命令一览

2.选择指令类型

3.选择要插入的指令名称

4.弹出指令编辑框

5.编辑指令参数

6.点击确认插入

6.插入成功

7.点击保存，保存文件

1.点击命令一览

2.选择指令类型

3.选择要插入的指令名称

4.弹出指令编辑框

5.编辑指令参数

6.点击确认插入

6.插入成功

7.点击保存，保存文件

69

2.指令修改

1.选中要修改的行

3.弹出指令编辑框

4.编辑指令参数

2.点击指令修改

5.点击确认修改

6.点击保存文件

7.支持Movj/Movl/Movs/Movp运动指令直接切换

4. 删除指令

（删除单独某行时，点击该行即选中）

1.选中要删除的行

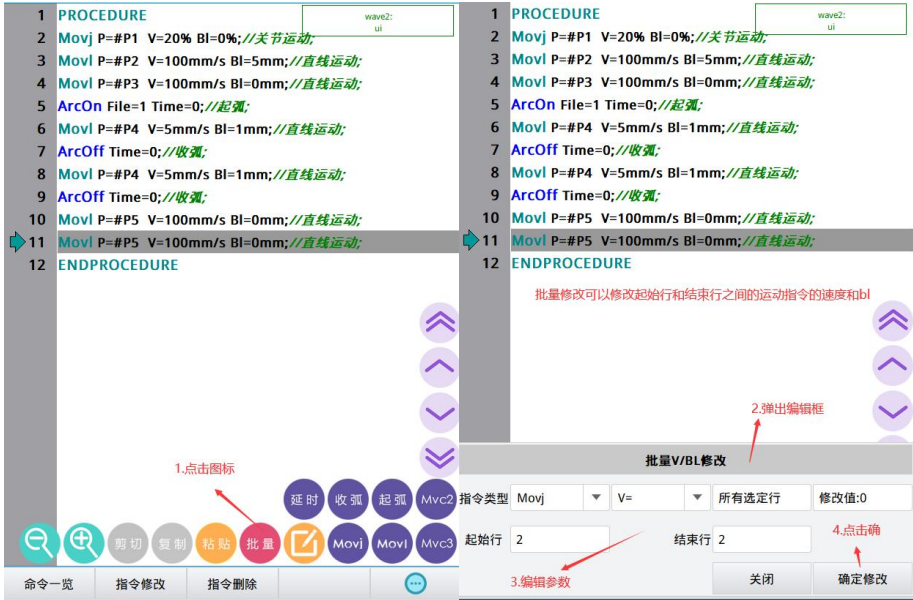
2.点击指令删除，会出现确认删除选项

3.确认删除

7.5 扩展栏

点击，会出现扩展操作栏，扩展栏各按键功能说明如下

图标	功能说明
	程序显示字体缩小

	程序显示字体放大
	剪切选中的行
	复制选中的行
	粘贴程序到选中行的后面
	批量修改，批量修改运动指令的速度和 bl  批量修改可以修改起始行和结束行之间的运动指令的速度和bl 1.点击图标 2.弹出编辑框 3.编辑参数 4.点击确定
	变量查看与修改：该功能用于快速查看和编辑当前选中行指令内所以用到的变量（包括全局变量和工程变量）

	1 PROCEDURE 2 Movj P=#P3 V=80% BI=0%; 3 //方; 4 Movj P=#P37 V=80%; 5 Movl P=#P38 V=1000mm/s; 6 Movl P=#P39 V=1000mm/s; 7 Movl P=#P40 V=1000mm/s; 8 Movl P=#P41 V=1000mm/s; 9 Movl P=#P42 V=1000mm/s; 10 Movl P=#P43 V=1000mm/s; 11 Movj P=#P44 V=80%; 12 Movl P=#P45 V=1000mm/s; 13 Movl P=#P46 V=1000mm/s; 14 Movl P=#P47 V=1000mm/s; 15 Movl P=#P48 V=1000mm/s; 16 Movl P=#P49 V=1000mm/s; 17 Movl P=#P50 V=1000mm/s; 18 Movj P=#P51 V=80%; 19 Movl P=#P52 V=1000mm/s; 20 Movl P=#P53 V=1000mm/s; 21 Movl P=#P54 V=1000mm/s; 22 Movl P=#P55 V=1000mm/s; 23 Movl P=#P56 V=1000mm/s; 24 shangxue: main ext-ist1: main test_all0: main test_all0: bi_test test_all0: bi0 1.选中查看行 2.点击图标 9 Movj P=#GP1 V=50%; 10 Movl P=#P1 V=20mm/s; 11 Movj P=#GP1 V=30%; 12 Movl P=#P1 V=20mm/s; 13 Movj P=#GP1 V=30%; 14 Movl P=#P1 V=20mm/s; 15 Movj P=#GP1 V=30%; 16 Movl P=#P1 V=20mm/s; test01: main test9: main 01: main 3: main 变量查看与修改 参数选择 #GVar1 2.选择要修改的参数 序号 数值 #GVar1 #GVar1 2.0000 #GP1 3. 点击编辑按键进行修改 关闭 修改 9 Movj P=#GP1 V=50%; 10 Movl P=#P1 V=20mm/s; 11 Movj P=#GP1 V=30%; 12 Movl P=#P1 V=20mm/s; 13 Movj P=#GP1 V=30%; 14 Movl P=#P1 V=20mm/s; 15 Movj P=#GP1 V=30%; 16 Movl P=#P1 V=20mm/s;
--	---

test01: main

test9: main

01: main

3: main

变量查看与修改

参数选择 #GP2

机器人位置

坐标系 机器人坐标系

扩展轴位置

轴组 不生效

X 200.0000 Ext1 0.0000

Y -200.0000 Ext2 0.0000

Z 637.0000 Ext3 0.0000

Roll 180.0000 Ext4 0.0000

Pitch 0.0000 Ext5 0.0000

Yaw 180.0000 Ext6 0.0000

J7 0.0000 Ext7 0.0000

J8 0.0000 Ext8 0.0000

关闭 修改 设为当前

设为当前: 记录机器人当前位置到位置型变量, 坐标系为当前jog的坐标系

	快速插入 Mvc2: 操作及步骤同上，会插入一条 Movc2 运动指令,即圆弧中间点
	快速插入 Mvc3: 操作及步骤同上，会插入一条 Movc3 运动指令,即圆弧目标点
	快速插入 Movs: 操作及步骤同上，会插入一条 Movs 运动指令
	快速插入 Movp: 操作及步骤同上，会插入一条 Movp 运动指令
	快速插入 MoveExt: 操作及步骤同上，会插入一条 MovceExt 运动指令
	快速修改当前位置点: 上伺服状态下，点击该按键会在已有工程位置型变量重新记录位置点，该点的位置为机器人当前位置，坐标系为当前 jog 坐标系
	快速创建替换当前位置点: 上伺服状态下，选择某一行运动指令，点击该按键会在工程位置型变量末尾增加一个位置点并替换当前运动指令，该点的位置为机器人当前位置，坐标系为当前 jog 坐标系

8 变量

8.1 全局变量

所有工程中的示教文件均可访问全局变量，包括  全局位置型变量（#GP）、
 全局增量位置型变量（#GIncP）、 全局数值型变量（#GVar）、 全局字符串变量（#GStr）

8.1.1 全局位置型变量

全局位置点在示教文件中以 #GPX(X 表示序号为 1-1000)的形式调用

项目	操作说明
1.位置点记录	全局位置变量序号 #GP1 ☐ 记录时绑定当前坐标系 机器人位置 注释 brbrtb 坐标系 关节坐标系 姿态 自动获取姿态 工具 1 J1 0.0000 J2 0.0000 J3 0.0000 J4 0.0000 J5 0.0000 J6 0.0000 J7 0.0000 J8 0.0000 扩展轴位置 扩展轴轴组 不生效 Ext1 0.0000 Ext2 0.0000 Ext3 0.0000 Ext4 0.0000 Ext5 0.0000 Ext6 0.0000 Ext7 0.0000 Ext8 0.0000 ☒ 关节位置作为参考 1.选择位置点序号 2.选择坐标系类型: 若选择“记录时绑定当前坐标系” 记录时该点的坐标系为当前jog坐标系; 若不选择,则记录的为坐标系选项 中选择的坐标系; 3.上伺服,点击记录 当前 4.默认勾选关节位置作为参考, 确保能够正确的运动到点 更多 < 1/1000 > 记录当前

2.手动修改位置点	全局位置变量序号 #GP1 ☐ 记录时绑定当前坐标系 机器人位置 注释 brbrtb 坐标系 关节坐标系 姿态 自动获取姿态 工具 1 J1 0.0000 J2 0.0000 J3 0.0000 J4 0.0000 J5 0.0000 J6 6.0000 J7 0.0000 J8 0.0000 扩展轴位置 3.手动编辑数值 扩展轴轴组 不生效 Ext1 0.0000 Ext2 0.0000 Ext3 0.0000 Ext4 0.0000 Ext5 0.0000 Ext6 0.0000 Ext7 0.0000 Ext8 0.0000 1.选择要修改的变量序号 4.点击确定按钮修改完成 2.点击修改按钮 更多 < 1/1000 > 记录当前
3.复制粘贴	全局位置变量序号 #GP1 ☐ 记录时绑定当前坐标系 机器人位置 注释 brbrtb 坐标系 关节坐标系 姿态 自动获取姿态 工具 1 J1 0.0000 J2 0.0000 J3 0.0000 J4 0.0000 J5 0.0000 J6 6.0000 J7 0.0000 J8 0.0000 扩展轴位置 4.点击粘贴按键，粘贴完成 扩展轴轴组 不生效 Ext1 0.0000 Ext2 0.0000 Ext3 0.0000 Ext4 0.0000 Ext5 0.0000 Ext6 0.0000 Ext7 0.0000 Ext8 0.0000 1.选择要复制的变量序号 2.点击复制按键 3.选择要粘贴到的变量序号 更多 < 1/1000 > 记录当前
4.运动到点	按下该按钮，上伺服，点手持【前进】按键即可运动到该位置点 更多 < 1/1000 > 记录当前

5.扩展轴轴组

扩展轴位置

扩展轴轴组 ☒ A ☒ B ☐ C ☐ D

Ext1

0.0000

Ext2

0.0000

Ext3

0.0000

Ext4

0.0000

Ext5

0.0000

Ext6

0.0000

Ext7

0.0000

Ext8

0.0000

1, 点击修改

复制

粘贴

2, 勾选需要联动的轴

选择扩展轴（外部轴组）时，点动运行同起同停。

8.1.2 全局增量位置点

全局增量位置点在示教文件中以 #GIncPX(X 表示序号为 1-1000)的形式调用：

全局增量位置点除了没有记录当前位置和运动到点的功能，其他操作和全局位置点的操作方式相同。

8.1.3 全局字符串变量

全局字符串变量在示教文件中以 #GStrX(X 表示序号为 1-300)的形式调用：

全局数值变量	全局字符串
序 号	数 值
#GStr286	
#GStr287	
#GStr288	
#GStr289	
#GStr290	
#GStr291	
#GStr292	
#GStr293	
#GStr294	
#GStr295	
#GStr296	
#GStr297	
#GStr298	
#GStr299	
#GStr300	

更多

<

20 / 20

>

图表 19 全局字符串变量

修改完变量值后点击即完成修改。

8.1.4 全局数值变量

全局数值变量在示教文件中以 #GVarX(X表示序号为1-300)的形式调用：

全局数值变量		全局字符串	
序 号	数 值	说 明	
#GVar1	0.0000		
#GVar2	0.0000		
#GVar3	0.0000		
#GVar4	0.0000		
#GVar5	0.0000		
#GVar6	0.0000		
#GVar7	0.0000		
#GVar8	0.0000		
#GVar9	0.0000		
#GVar10	0.0000		
#GVar11	0.0000		
#GVar12	0.0000		
#GVar13	0.0000		
#GVar14	0.0000		
#GVar15	0.0000		

更多

<

1 / 20

>



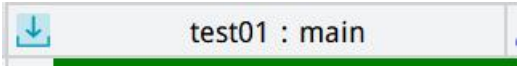
图表 20 全局数值变量

修改完变量值后点击即完成修改。

8.2 工程变量

每个工程下都有一组工程变量，只有该工程下的示教文件才能访问。包括工程位置型变量（#P）、工程增量位置型变量（#IncP）、工程数值型变量（#Var）、工程字符串变量（#Str）。

查看工程变量前，先要打开一个工程，当前打开的工程可在状态栏中获取，如



当前选定的工程即为 test01, 此时进入工程变

量界面显示的为该工程下的变量。

8.2.1 工程位置型变量

工程位置点在示教文件中以 #PX(X 表示序号)的形式调用

项目	操作说明
1.新增变量	
2.其他操作	参考全局位置变量

8.2.2 工程增量位置型变量

伺服关示教关节T0P020 20

test01 : main

*****正常*****清除

工程增量位置序号 #IncP1

机器人位置

注释

坐标系关节坐标系

姿态自动获取姿态工具1

J10.0000

J20.0000

J30.0000

J40.0000

J50.0000

J60.0000

J70.0000

J80.0000

扩展轴位置

扩展轴轴组不生效

Ext10.0000

Ext20.0000

Ext30.0000

Ext40.0000

Ext50.0000

Ext60.0000

Ext70.0000

Ext80.0000

✓✎+📄🔍

清空<1/1>

图表 21 工程增量位置

工程位置点在示教文件中以 #IncpPX(X 表示序号)的形式调用,具体操作参考工程位置型变量

8.2.3 工程数值型变量

工程数值变量在示教文件中以 #VarX(X 表示序号)的形式调用。

和全局数值型变量相比，工程数值型变量需要用户自己点添加，其他操作均相同。

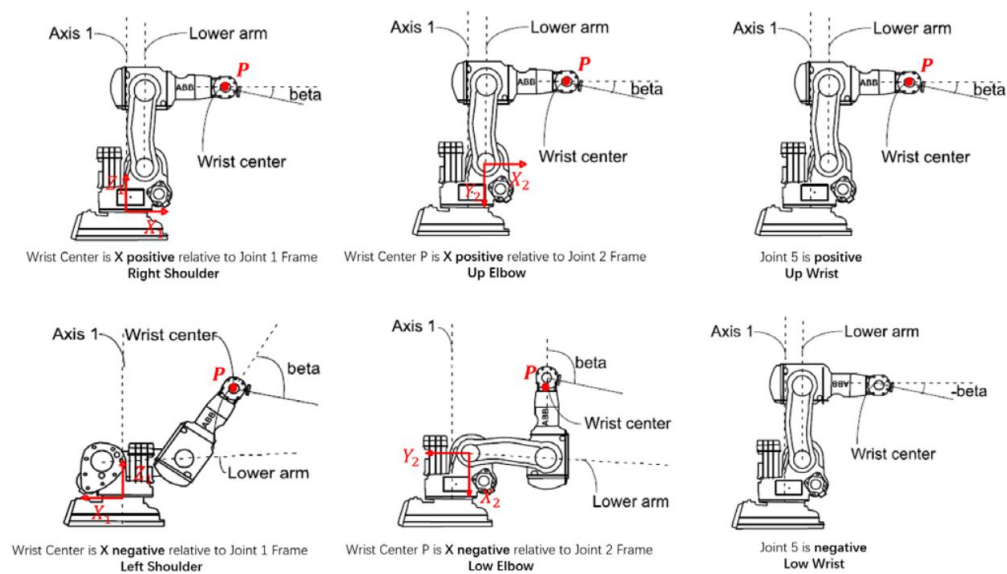
8.2.4 工程字符串变量

工程字符串变量在示教文件中以 #StrX(X 表示序号)的形式调用。

和全局字符串变量相比，工程字符串变量需要用户自己点  添加，其他操作均相同。

8.3 机器人姿态

在位置型变量和增量位置型变量中会用到机器人姿态，对于 6 轴机器人姿态模型如下定义：



图表 22 6 轴机器人姿态定义

9 坐标系

本章分别介绍了

- ACS——关节坐标系
- KCS——机器人坐标系
- WCS——世界坐标系
- TCS——工具坐标系
- PCS——工件坐标系

9.1 坐标系系统介绍

对机器人进行轴操作时，可以使用以下几种坐标系：

9.1.1 关节坐标系 — ACS

Axis Coordinate System，简称为 ACS。关节坐标系是以各轴机械零点为原点所建立的纯旋转的坐标系。机器人的各个关节可以独立的旋转，也可以一起联动。

9.1.2 机器人坐标系 — KCS

Kinematic Coordinate System，简称为 KCS。机器人（运动学）坐标系是用来对机器人进行正逆向运动学建模的坐标系，它是机器人的基础笛卡尔坐标系，也可以称为机器人基础坐标系（Base Coordinate System — BCS）或运动学坐标系，机器人工具末端 TCP 在该坐标系下可以进行沿坐标系 X 轴、Y 轴、Z 轴的移动运动，以及绕坐标系轴 X 轴、Y 轴、Z 轴的旋转运动。

9.1.3 工具坐标系 — TCS

Tool Coordinate System，简称为 TCS。

安装工具：将机器人腕部法兰盘所持工具的有效方向作为工具坐标系 Z 轴，并把工具坐标系的原点定义在工具的尖端点（或中心点）TCP（TOOL CENTER POINT）。

未安装工具：此时工具坐标系建立在机器人法兰盘端面中心点上，Z 轴方向垂直于法兰盘端面指向法兰面的前方。

当机器人运动时，随着工具尖端点 TCP 的运动，工具坐标系也随之运动。用户可以选择在工具坐标系 TCS 下进行示教运动。TCS 坐标系下的示教运动包括沿工具坐标系的 X 轴、Y 轴、Z 轴的移动运动，以及绕工具坐标系轴 X 轴、Y 轴、Z 轴的旋转运动。本机器人系统支持用户保存 32 个自定义的工具坐标系。

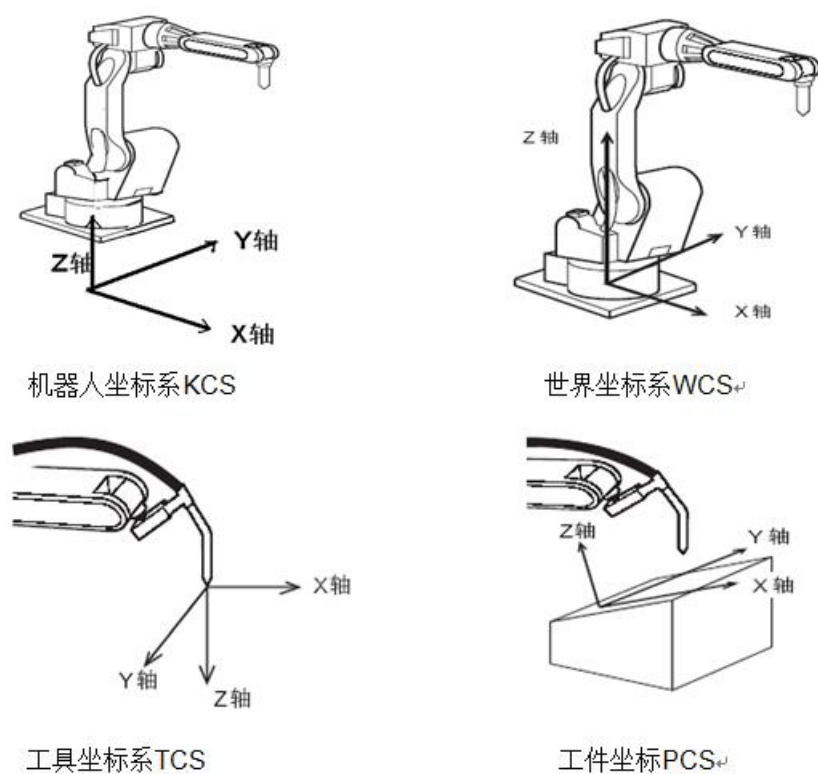
9.1.4 世界坐标系 — WCS

World Coordinate System, 简称为 WCS。世界坐标系也是空间笛卡尔坐标系统。世界坐标系是其它笛卡尔坐标系（机器人运动学坐标系 KCS 和工件坐标系 PCS）的参考坐标系统，运动学坐标系 KCS 和工件坐标系 PCS 的建立都是参照世界坐标系 WCS 来建立的。在默认没有示教配置世界坐标系的情况下，世界坐标系到机器人运动学坐标系之间没有位置的偏置和姿态的变换，所以此时世界坐标系 WCS 和机器人运动学坐标系 KCS 重合。用户可以通过“坐标系管理”界面来示教世界坐标系 WCS。机器人工具末端在世界坐标系下可以进行沿坐标系 X 轴、Y 轴、Z 轴的移动运动，以及绕坐标系轴 X 轴、Y 轴、Z 轴的旋转运动。本机器人系统支持用户保存 32 个自定义的世界坐标系。

9.1.5 工件坐标系 — PCS

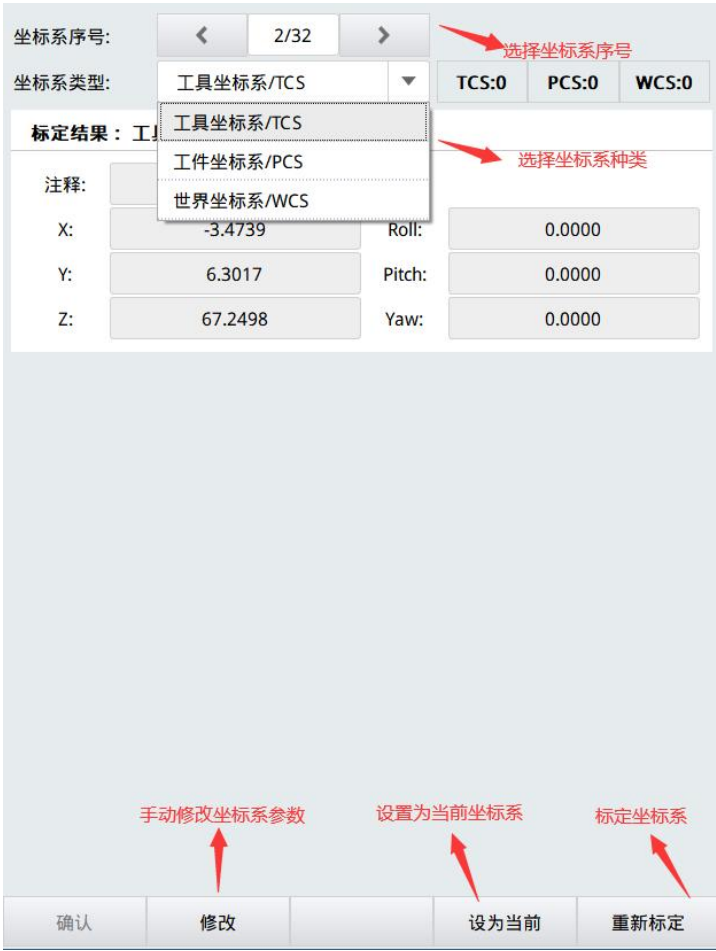
Piece Coordinate System, 简称为 PCS。工件坐标系 PCS 是建立在世界坐标系 WCS 下的一个笛卡尔坐标系。工件坐标系主要是方便用户在一个应用中切换世界坐

标系 WCS 下的多个相同的工件。另外，示教工件坐标系后，机器人工具末端 TCP 在工件坐标系下的移动运动和旋转运动能够减轻示教工作的难度。工件坐标系下可支持用户保存 32 个自定义的工件坐标系。



图表 23 4 种坐标系示意图

9.2 坐标系管理界面



图表 24 坐标系界面

9.3 关节坐标系（ACS）

在示教模式下，坐标系设定为关节坐标系ACS时，机器人的J1、J2、J3、J4、J5、J6各轴分别运动，按轴操作键时各轴的动作情况请参照下表：

轴名称		轴操作键	动作
基本轴	J1 轴		本体左右回旋
	J2 轴		下臂前后运动
	J3 轴		上臂上下运动

腕部轴	J4 轴		上臂带手腕回旋
	J5 轴		手腕上下运动
	J6 轴		手腕回旋

9.4 机器人坐标系（KCS）

在示教模式下，坐标系设定为机器人坐标系 KCS 时，机器人工具末端 TCP 沿 KCS 坐标系的 X、Y、Z 轴平行移动和绕 KCS 坐标系的 X、Y、Z 轴的旋转运动，按住轴操作键时，各轴的动作请参考下表：

轴名称		轴操作键	动作
移动轴	X 轴		沿 KCS 坐标系 X 轴平移运动
	Y 轴		沿 KCS 坐标系 Y 轴平移运动
	Z 轴		沿 KCS 坐标系 Z 轴平移运动
旋转轴	绕 X 轴		绕 KCS 坐标的 X 轴旋转运动
	绕 Y 轴		绕 KCS 坐标的 Y 轴旋转运动

	绕 Z 轴		绕 KCS 坐标的 Z 轴旋转运动
--	-------	--	-------------------

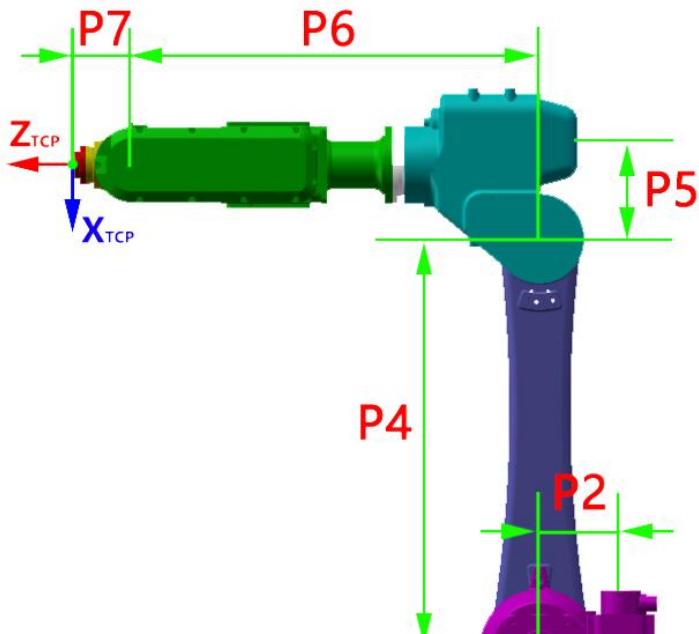
9.5 工具坐标系

工具管理界面主要对机器人末端法兰盘安装的工具进行管理。

9.5.1 工具坐标系轴动作

工具坐标系把机器人腕部法兰盘所握工具的有效方向定为 Z 轴，把坐标系原点定义在工具尖端点或中心点（TOOL CENTER POINT -- TCP），所以工具坐标系的位姿随腕部的运动而发生变化。

未指定工具之前，默认工具坐标系位于机器人法兰盘中心位置，如下图所示。



图表 25 默认工具坐标系

沿工具坐标系的移动，以工具的有效方向为基准，与机器人的位置、姿态无关，所以进行相对于工件不改变工具姿势的平行移动操作时最为适宜。各轴的动作请参

考下表：

轴名称		轴操作键	动作
移动轴	X 轴		沿 TCS 坐标系 X 轴平移运动
	Y 轴		沿 TCS 坐标系 Y 轴平移运动
	Z 轴		沿 TCS 坐标系 Z 轴平移运动
旋转轴	绕 X 轴		绕 TCS 坐标的 X 轴旋转运动
	绕 Y 轴		绕 TCS 坐标的 Y 轴旋转运动
	绕 Z 轴		绕 TCS 坐标的 Z 轴旋转运动

9.5.2 工具坐标系的标定

用户可通过系统应用菜单下的子菜单{坐标系管理}来进入该标定管理界面。工具坐标系 TCS 管理主界面与世界坐标系 WCS 管理主界面相同。

9.5.2.1 参数解释

标定平面：

XY：标定选择三个点 P1-P3 位于 XY 平面，Z 轴方向由 X 轴和 Y 轴根据右手定则叉

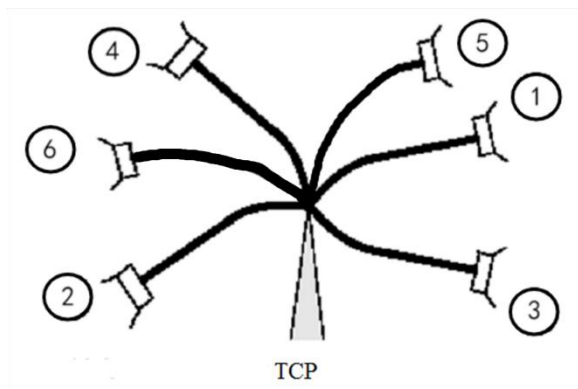
乘得出。

YZ：标定选择三个点 P1-P3 位于 YZ 平面，X 轴方向由 X 轴和 Y 轴根据右手定则叉乘得出。

ZX：标定选择三个点 P1-P3 位于 ZX 平面，Y 轴方向由 Z 轴和 X 轴根据右手定则叉乘得出。

9.5.2.2 偏置 6 点法：标定位置偏移

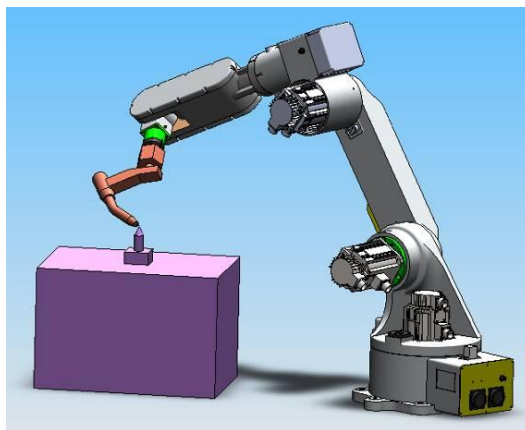
使用{偏置 6 点法}标定时，用待测工具的尖端（中心）点（即 TCP 点）从六个任意不同的方向靠近同一个参照点，参照点可以任意选择，但必须为同一个固定不变的参照点。机器人控制器从六个不同的法兰位置计算出 TCP。机器人 TCP 点运动到参考点的六个法兰位置必须分散开足够的距离，才能使计算出来的 TCP 点尽可能精确。{偏置 6 点法}示意图如下所示：



图表 26 偏置六点法图示

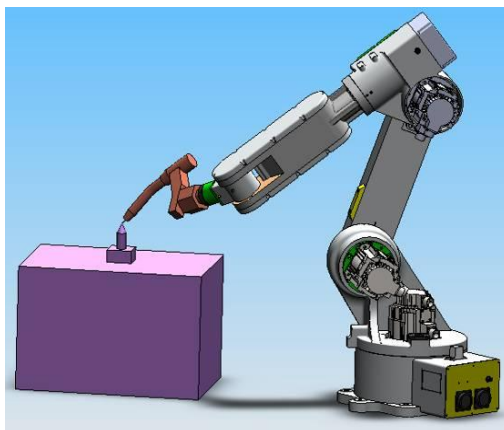
{偏置6点法}示教并计算工具中心点TCP的位置的步骤如下：

第 1 步：选择要刷新的坐标系索引号，选择偏置模式。



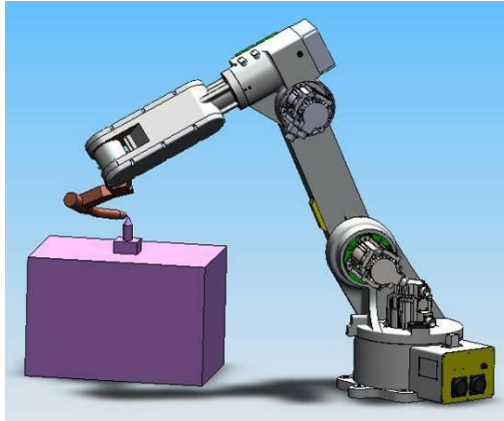
图表 29 记录P2位置点

第 4 步：将待测工具的尖端点 TCP 从第三个方向靠近同一个固定参照点。在伺服电源接通的情况下点击{记录 P3}按钮，记录第三个位置点。P3 点记录完成后{记录 P3}按钮的背景色会变为绿色。



图表 30 记录P3位置点

第 5 步：将待测工具的尖端点 TCP 从第四个方向靠近同一个固定参照点。在伺服电源接通的情况下点击{记录 P4}按钮，记录第四个位置点。P4 点记录完成后{记录 P4}按钮的背景色会变为绿色。



图表 31 记录P4位置点

第 6 步：按照以上步骤继续记录第五第六个位置点，P5、P6 点记录完成后{记录 P5、P6}按钮的背景色会变为绿色。

第7步：{偏置6点法}所需的六个位置点记录完成后点击{计算}按钮，自动计算TCP位置点数据并显示计算结果。

注意，如果偏置六点法中记录了两个或多个相同的位置点，则计算不能成功，程序会报告错误。

第 8 步：点击{保存结果}按钮，保存记录的示教位置点坐标及计算的坐标系数数据，返回到坐标系管理主界面。

第 9 步：点击{设为当前}按钮，将计算的 TCP 工具作为法兰末端工具。到此为止，已完成从工具坐标系计算到切换新计算出来的工具为当前使用的工具的所有步骤。工具坐标系计算并切换成功，现在可以在新的工具下进行机器人的各种运动。

注意，使用偏置六点法只能确定工具尖端（中心）点TCP相对于机器人末端法兰安装面的位置偏移值，当用户需要示教确定工具姿态分量时，需要额外再使用姿态三点法，或者直接使用偏置+姿态的六点法。

9.5.2.3 姿态 3 点法：标定姿态偏移

{姿态 3 点法}示教并计算工具坐标系 TCS 的姿态分量的步骤如下：

第 1 步：选择需要修改或刷新的工具坐标系的序号，并选择姿态三点法工作模

式。

在姿态三点法工作模式下，需要记录三个位置点，即 P1 点、P2 点、P3 点。此外，用户还需要选择示教点所在的平面，如下所示，选择 XY 平面。即示教的 P1 点和 P2 点用来确定工具坐标系的 X 轴的方向，P3 点在工具坐标系 XY 平面的 Y 轴正方向一侧。由于姿态三点法只是确定工具坐标系 TCP 的姿态分量，所以示教的 XY 平面只要求平行于实际工具坐标系 TCS 的 XY 平面即可，并不要求一定是 TCS 的 XY 平面，P1 点是选定的 XY 示教平面的坐标系 X 轴上的一点，并不要求必须是工具尖端（中心）点 TCP（工具坐标系的原点），P2 点和 P3 点也是如此要求。

1.标定方法选择

☐ 偏置+姿态

☐ 偏置

☒ 姿态

☒ XY平面

☐ YZ平面

☐ ZX平面

2.位置点记录

P1

P2

P3

上一步

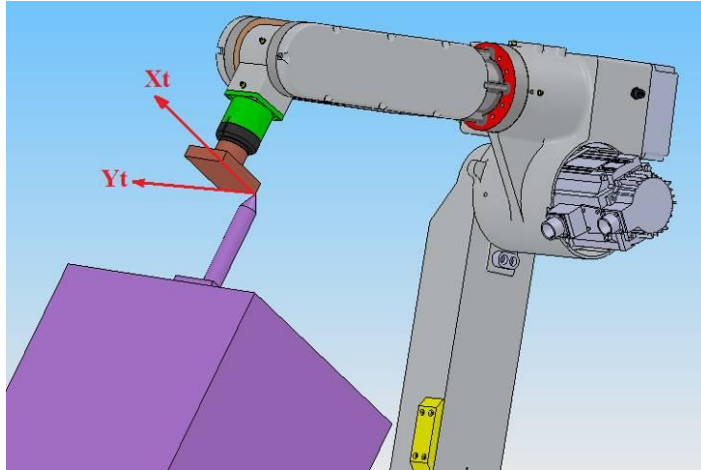
示教点

计算

保存结果

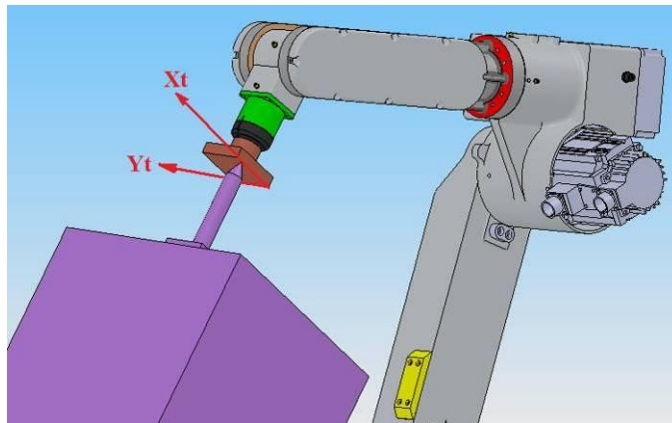
图表 32 选择姿态三点法工作模式

第 2 步：如下图所示，首先记录工具坐标系上 X 轴方向上的第一个点，即 P1 点。



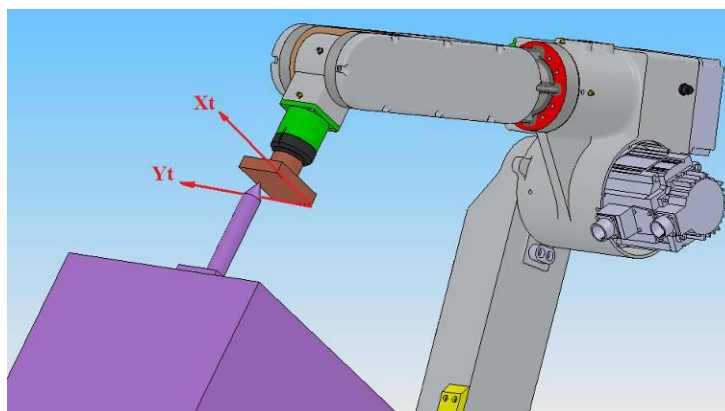
图表 33 记录 P1 点 (X 轴正方向上的第一点 P1)

第 3 步：记录工具坐标系上 X 轴方向上的第二个点，即 P2 点。



图表 34 记录 P2 点 (X 轴正方向上的第二点 P2)

第 4 步：记录工具坐标系上 XY 平面上 Y 轴正方向上的一个点，即 P3 点。



图表 35 记录 P3 点 (XY 平面上 Y 轴正方向上的点)

第 5 步：点击{计算}按钮，程序根据记录的 P1 点、P2 点和 P3 点生成工具坐标系姿态分量数据，并点击{保存结果}更新选中的坐标系序号的工具坐标系 TCS 的姿

态分量。

需要注意的是，采用姿态三点法（需要记录的三个位置点）来确定工具姿态时，这三个位置点只能用笛卡尔空间下的移动运动来示教（即只能走 KCS、WCS、PCS、TCS 下的 XYZ 的移动运动，而不能进行 XYZ 的旋转运动或 ACS 下的单个关节转动运动来示教），不能用有任何姿态的旋转运动来示教，否则，不能计算出工具坐标系的姿态分量，并给出错误警告。

9.5.2.4 偏置 + 姿态 6 点法：标定位置和姿态的偏移

坐标系序号: 2/32

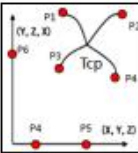
坐标系类型: 工具坐标系/TCS

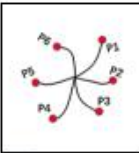
TCS:0 PCS:0 WCS:0

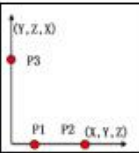
坐标系标定结果

X:	0.0000	Roll:	0.0000
Y:	0.0000	Pitch:	0.0000
Z:	0.0000	Yaw:	0.0000

1. 标定方法选择







☒ 偏置+姿态

☐ 偏置

☐ 姿态

☒ XY平面

☐ YZ平面

☐ ZX平面

2. 位置点记录

P1	P2	P3
P4	P5	P6

上一步

示教点

计算

保存结果

图表 36 工具坐标系标定界面

工具坐标系标定方法有{偏置 6 点法}、{姿态 3 点法}和{偏置 + 姿态 6 点法}，共三种方法可供选择。{偏置 + 姿态 6 点法}可以综合示教六轴关节机器人末端工具 TCP

(TOOL CENTER POINT) 的位置偏移和姿态向量；{偏置 6 点法}只能计算六轴机器人工具末端的 TCP 位置偏移值，不能计算工具的姿态向量；{姿态 3 点法}只能计算六轴机器人工具末端的 TCP 姿态向量，不能计算工具末端 TCP 的位置偏移值。{偏置 + 姿态 6 点法}实际上是{4 点法}和{3 点法}的综合使用，{偏置 + 姿态 6 点法}中记录的前四个位置点使用{4 点法}计算工具末端 TCP 的位置偏移，后三个位置点使用{3 点法}计算工具的 TCP 姿态向量。

9.6 工件坐标系（PCS）

9.6.1 工件坐标系轴动作

在示教模式下，坐标系设定为工件坐标系PCS时，机器人工具末端TCP沿PCS坐标系的X、Y、Z 轴平移运动和绕PCS坐标系的X、Y、Z轴的旋转运动，按住轴操作键时，各轴的动作请参考下表：

轴名称		轴操作键	动作
移动轴	X 轴		沿 PCS 坐标系 X 轴平移运动
	Y 轴		沿 PCS 坐标系 Y 轴平移运动
	Z 轴		沿 PCS 坐标系 Z 轴平移运动
旋转轴	绕 X 轴		绕 PCS 坐标的 X 轴旋转运动
	绕 Y 轴		绕 PCS 坐标的 Y 轴旋转

			运动
	绕 Z 轴		绕 PCS 坐标的 Z 轴旋转 运动

9.6.2 工件坐标系的标定

用户可通过系统应用菜单下的子菜单{坐标系管理}来进入该标定管理界面。

9.6.2.1 标定方法参数解释

标定平面：

XY：标定选择三个点 P1-P3 位于 XY 平面，Z 轴方向由 X 轴和 Y 轴根据右手定则叉乘得出。

YZ：标定选择三个点 P1-P3 位于 YZ 平面，X 轴方向由 X 轴和 Y 轴根据右手定则叉乘得出。

ZX：标定选择三个点 P1-P3 位于 ZX 平面，Y 轴方向由 Z 轴和 X 轴根据右手定则叉乘得出。

模式：

姿态三点法模式 1：以 XY 标定平面为例，P1 为原点位置，P2 为 X 轴正方向上的一点，P3 为 XY 平面上的一点。用这种方法示教的坐标系的原点位于 P1 点，X 轴的正方向从 P1 点指向 P2 点，P3 点位于 Y 轴正方向一侧。YZ 平面和 ZX 平面同理。

姿态三点法模式 2：以 XY 标定平面为例，X 轴上的一点 P1 和另一点 P2，在 Y 轴上示教第三个点 P3。过 P3 点做 P1-P2 连线的垂线，垂足位置处即为坐标系的原点。用这种方法示教的坐标系的 X 轴正方向从 P1 点指向 P2 点，P3 点位于 Y 轴的正半轴上。

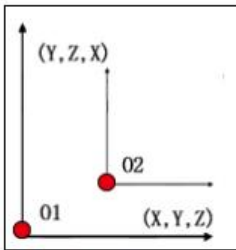
原点偏置：用户采用上述两种方法示教的坐标系偏移到示教记录的 O0 位置点处。

坐标系序号:	<	2/32	>			
坐标系类型:	工件坐标系/PCS		▼	TCS:0	PCS:0	WCS:0

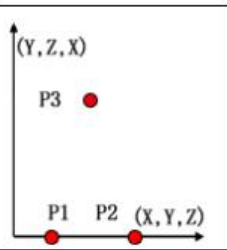
坐标系标定结果

X:	0.0000	Roll:	0.0000
Y:	0.0000	Pitch:	0.0000
Z:	0.0000	Yaw:	0.0000

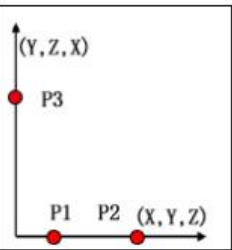
1. 标定方法选择



☒ 原点偏置



☒ 三点法模式1



☐ 三点法模式2

☒ XY平面

☐ YZ平面

☐ ZX平面

2. 位置点记录

记录P1点

记录P2点

记录P3点

记录O点

上一步
示教点
计算
保存结果

图表 37 工件坐标系标定界面

9.6.2.2 如何标定工件一个坐标系

第 1 步：选择坐标系序号和类型

默认 (1) 号坐标系无法修改, 请从 2 号开始选取标定。

第 2 步：选择标定方法

选择标定方法{姿态三点法模式 1}、{姿态三点法模式 2}，详情查考上文“标定方法参数解释”。

此外用户还可以再增加记录一个坐标原点偏置位置点 O0 点。这个位置点是可选项，当用户使用该功能的时候，可以将用户采用上述两种方法示教的坐标系偏移 to 示教记录的 O0 位置点处，而坐标系的姿态保持不变。

用户通过按下界面中的{XY}、{YZ}、{ZX}按钮可以选择示教的坐标系平面。

第 3 步：位置点记录

运动到位置点，点击对应按钮，按钮背景色变为绿色（如下图所示）说明记录成功。注意标定的位置点相互不能太近。

坐标系序号:

<

2/32

>

坐标系类型:

世界坐标系/WCS

▼

TCS:0

PCS:3

WCS:0

坐标系标定结果

X:

0.0000

Roll:

0.0000

Y:

0.0000

Pitch:

0.0000

Z:

0.0000

Yaw:

0.0000

1. 标定方法选择

☐ 原点偏置

☒ 三点法模式1

☐ 三点法模式2

☒ XY平面

☐ YZ平面

☐ ZX平面

2. 位置点记录

记录P1点

记录P2点

记录P3点

上一步

示教点

计算

保存结果

图表 38 工件坐标系标定界面 2

第 4 步：计算并保存结果

点击{计算}, 如果显示成功则点击{保存结果}; 如果失败会在状态栏显示警报, 根据报警提示修改或重新标定。

此外，在位置点记录界面上，用户点击{示教点}按钮，进入示教点管理界面。

在这个界面上，用户可以查看记录的位置点数据，并可以使机器人运动到指定的记录点。首先选中已记录的位置点，然后使机器人处于伺服使能的状态，选择点 P1、P2、P3 及{运动到点}，按住 **【前进】**，机器人按直线运动的模式朝着指定的位置点

运动。点击{上一步}返回标定界面。

坐标系序号:

<2/32>

坐标系类型:

世界坐标系/WCS

TCS:1PCS:17WCS:1

位置点选择

选择点P1

选择点P2

选择点P3

位置点关节坐标

J1:	-0.6579	J5:	89.3373
J2:	-29.0242	J6:	-0.6571
J3:	29.6012	J7:	0.0000
J4:	0.0018	J8:	0.0000

上一步

运动到点

图表 39 示教点查看界面

第 5 步：设为当前

标定完一个工件坐标系后，返回到坐标系管理界面，将标定的坐标系设为当前使用的坐标系。检查状态栏上“ T3P5”是否显示的是设定的坐标系序号，T 为工具坐标系，P 为工件坐标系。

第 6 步：验证坐标系

通过机器人工具末端 TCP 沿 PCS 坐标系的 X、Y、Z 轴平移运动和绕点（尖端）

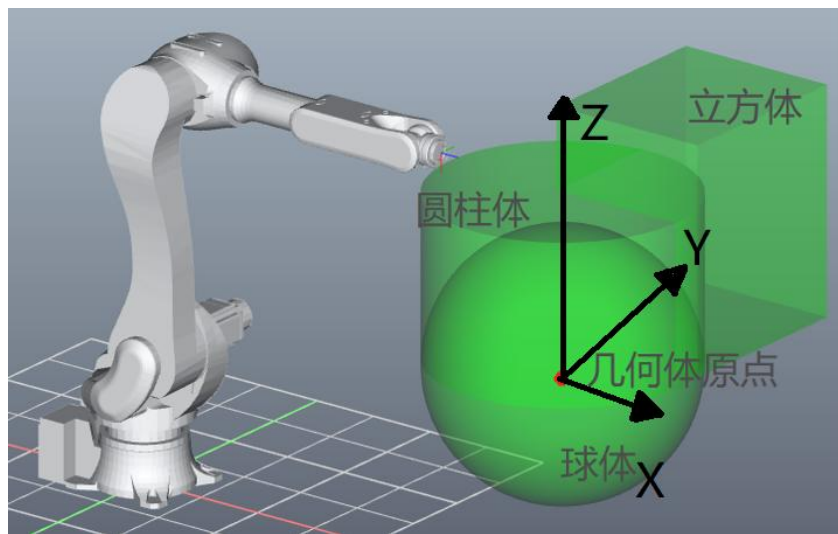
的 PCS 坐标系的 X、Y、Z 轴的旋转运动，来验证标定的坐标系的方向精度。

9.7 工作空间监控

在实际应用场景中，机器人的工作空间往往需要设定相对限制，例如设定工作空间的范围，限制机器人在某一范围内运动，或者设定工作区域的超限范围，限制机器人运动进入某个范围，又或者设定工作的相互干涉范围，例如多台设备需共用同一块空间区域，当机器人运行至干涉范围时，信号可输出给同在干涉范围工作的外围设备，以此进行相互限制，避免碰撞。

机器人系统针对三种类型，开发工作空间监控的功能，其中包括有工作区、障碍区、监控区，且最多可设定 32 个工作空间监控区。

9.7.1 工作空间监控说明



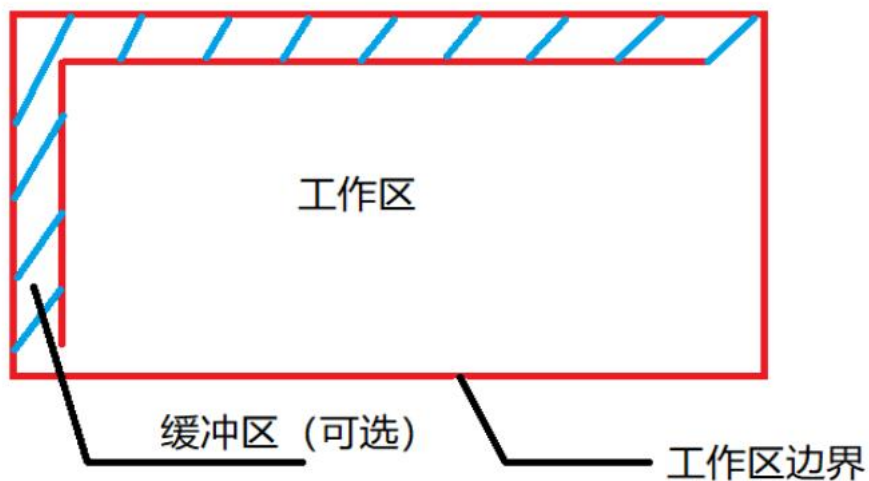
图表 40 工作空间监控示意

9.7.1.1 工作区

设定一个几何体（球体、立方体、圆柱）的工作空间，机器人当前的 TCP 点只能在设定的几何体内运行，一旦超出该工作区域范围，机器人将进行紧急刹车停止

并触发超出工作空间的报警提示，或者已预先设定工作空间缓冲边界，当机器人当前 TCP 点进入缓冲边界区域时，机器人会自动减速到安全速度在缓冲边界内进行运动。

如果机器人当前的 TCP 点未超出被激活的所有的工作区，仍会判断确定机器人在工作区中正常运行，即未超出工作区域。如果机器人当前 TCP 末端都超出已被激活的工作区范围，即超出工作区域，则机器人会进行紧急刹车停止，并触发超出工作空间的报警提示，报警含有对应的工作空间序号。



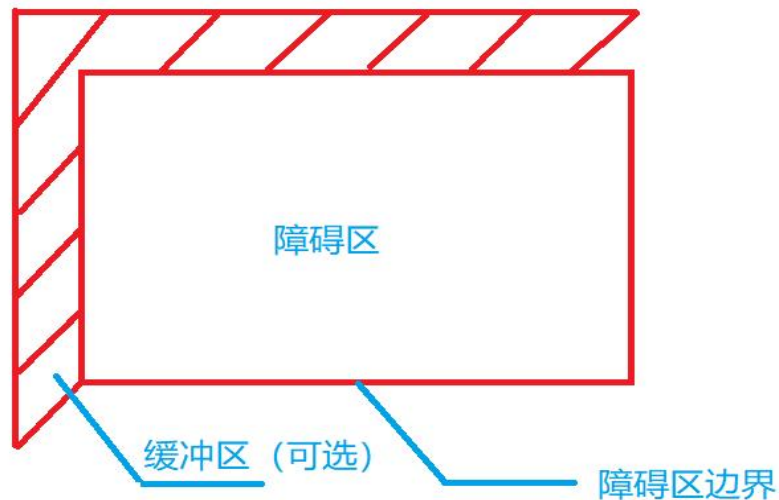
图表 41 工作区示意

9.7.1.2 障碍区

设定一个几何体（球体、立方体、圆柱）的障碍空间，机器人当前的 TCP 点只能在设定的几何体外运行，一旦进入该障碍区域范围，机器人将进行紧急刹车停止并触发已进入障碍区的报警提示，或者已预先设定障碍缓冲边界，当机器人当前 TCP 点进入缓冲边界区域时，机器人会自动减速到安全速度在缓冲边界内进行运动。

如果机器人当前的 TCP 点进入任意一个被激活的障碍区，则会判断确定已进入障碍区内，机器人会进行紧急刹车停止，并触发进入障碍区的报警提示，报警含有

对应的工作空间序号。



图表 42 障碍区示意

9.7.1.3 监控区

设定一个几何体（球体、立方体、圆柱）的监控空间，机器人当前的 TCP 点进入已设定的监控区间时，机器人会根据用户设定进行输入输出 IO 信号触发，例如设定输入为低/高电平触发，输出 IO 信号或者数值；同时会弹出已进入监控区的警告提示，或者已预先设定障碍缓冲边界，当机器人当前 TCP 点进入缓冲边界区域时，机器人会自动减速到安全速度在缓冲边界内进行运动。

如果机器人当前的 TCP 点进入任意一个被激活的监控区，则会判断确定已进入监控区，触发进入障碍区的警告提示；设定在示教模式下的监控区，机器人不会紧急停止运动，在回放模式下，进入监控区机器人会紧急停止运动，报警含有对应的工作空间序号。

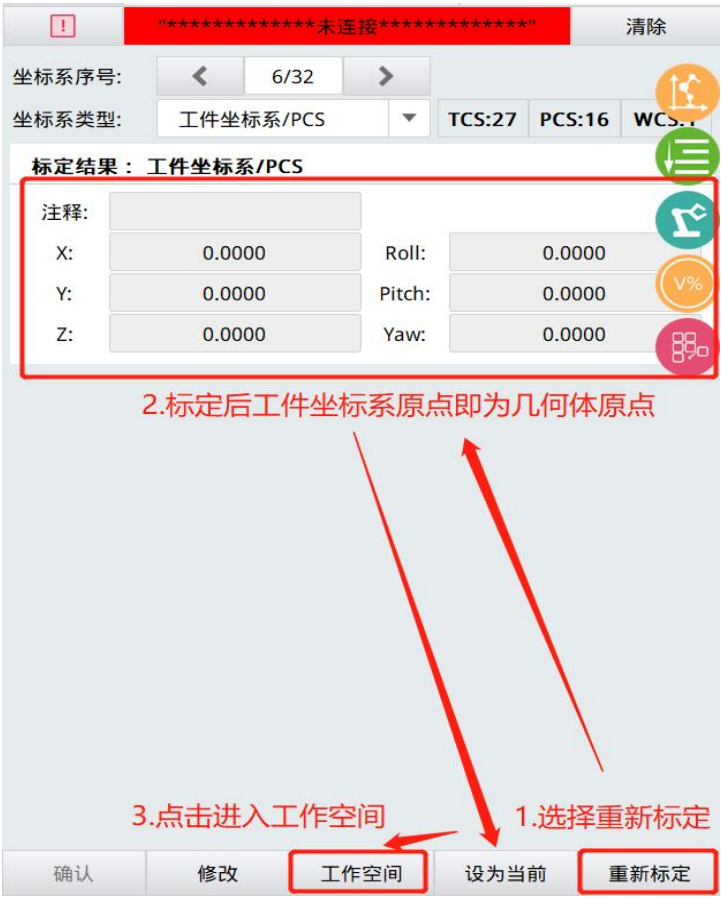
9.7.2 工作空间监控设置

工作监控空间的几何体模型是基于当前绑定的工件坐标系设置，即设定的工件

坐标系原点是设定几何体的原点，也可在功能上设置原点偏置，即几何体原点工件坐标系原点的基础上进行偏置。

9.7.2.1 几何体原点设定

在{菜单}  选择点击{坐标系管理}  来进入坐标系标定界面，并选择坐标系序号。坐标系类型选择{工件坐标系/PCS}，然后进行坐标标定，标定操作步骤说明，请跳转翻阅工件坐标系的标定。



图表 43 几何原点设定

9.7.2.2 全局监控设置

工件坐标系标定后，底下一栏选择点击{工作空间} ，进入如图所示。

注意：首次使用工作空间监控时，需要先点击进入{全局配置}进行启用设置。

坐标系序号: 1(默认)

坐标系类型: 工件坐标系/PCS

TCS:27 PCS:16 WCS:1

1.类型配置

区域类型: 不生效

监控点: 工具末端

工作模式: 自动模式

安全边界(mm): 0.00

2.形状配置(mm)

区域形状: 球体

半径(R): 0.00

未使用: 0.00

未使用: 0.00

3.区域原点偏置(mm)

X: 0.00

y: 0.00

z: 0.00

Roll: 0.00

Pitch: 0.00

Yaw: 0.00

4.IO配置

输出IO模块: 0

输出IO点位: 1

输出有效电平: 低有效

输出GVar序号: 不使用

输入IO模块: 0

输入IO点位: 1

输入有效电平: 低有效

互锁处理: 继续输出

首次使用先到{全局配置}进行设置

坐标系管理 全局配置 保存

坐标系序号: 6/32

坐标系类型: 工件坐标系/PCS

TCS:27 PCS:16 WCS:1

全局监控开关: 不启用

1.全局工作区

IO模块: 0

IO点位: 1

有效电平: 低有效

输出GVar序号: 不使用

2.全局障碍区

IO模块: 0

IO点位: 1

有效电平: 低有效

输出GVar序号: 不使用

3.全局监控区/干涉区

IO模块: 0

IO点位: 1

有效电平: 低有效

输出GVar序号: 不使用

2.点击保存

工作区配置 保存

图表 44 全局监控设定

名称		定义
全局监控开关		工作空间监控的启用/不启用开关
全局工作区	IO模块	机器人当前 TCP 点超出所有激活工作区时，输出对应的 IO 信号。若不使用在输入框内输入-1
	输出Gvar序号	机器人当前 TCP 点超出所有激活工作区时，对应的 Gvar 序号赋值 1。若不使用在输入框内输入-1
	有效电平	可以选择高电平触发或者低电平触发
全局障碍区	IO 模块	机器人当前 TCP 点进入任意一个被激活的障碍区时，输出对应的 IO 信号。若不使用在输入框内输入-1
	输出 Gvar 序号	机器人当前 TCP 点进入任意一个被激活的障碍区时，对应的 Gvar 序号赋值 1。若不使用在输入框内输入-1
	有效电平	可以选择高电平触发或者低电平触发
全局干涉区	IO 模块	机器人当前 TCP 点进入任意一个被激活的干涉区时，输出对应的 IO 信号。若不使用在输入框内输入-1
	输出 Gvar 序号	机器人当前 TCP 点进入任意一个被激活的干涉区时，对应的 Gvar 序号赋值 1。若不使用在输入框内输入-1
	有效电平	可以选择高电平触发或者低电平触发

9.7.2.3 工作空间监控设置

- 1.坐标系序号可选择 1-32；在工作空间配置时，请注意选择的坐标系序号；
- 2.点击保存，数据即生效。

坐标系序号:

<

1(默认)

>

坐标系类型:

工件坐标系/PCS

TCS:27PCS:16WCS:1

1.类型配置

区域类型

不生效

▼

监控点

工具末端

工作模式

自动模式

▼

安全边界(mm)

0.00

2.形状配置(mm)

区域形状

球体

▼

半径(R)

0.00

未使用

0.00

未使用

0.00

3.区域原点偏置(mm)

X: 0.00

y: 0.00

z: 0.00

Roll: 0.00

Pitch: 0.00

Yaw: 0.00

4.IO配置

输出IO模块

0

输出IO点位

1

输出有效电平

低有效

▼

输出GVar序号

不使用

输入IO模块

0

输入IO点位

1

输入有效电平

低有效

▼

互锁处理

继续输出

▼

坐标系管理

全局配置

保存

图表 45 工作空间监控设定

名称		定义
类型配置	区域类型	内含可选四个选项：1.不生效；2.工作区；3.障碍区；4.监控区；
	工作模式	内含可选三种模式： 1.自动模式： 仅在回放模式下生效；

		2.示教模式：仅在示教模式下生效； 3.示教+自动模式：回放和示教模式下生效；	
	监控点	内含可选两个监控点;1.工具末端；2.法兰中心	
	安全边界 (mm)	即缓冲边界，当机器人运行至安全边界时，运行速度会降至安全速度。	
形状配置 (mm)	球体	以当前对应的工件坐标系原点为球体的球心，半径自行设定，限定一个空间区域。	
	立方体	以当前对应的工件坐标系原点为立方体底面顶点，长宽高自行设定，限定一个空间区域。	
	圆柱体	以当前对应的工件坐标系原点为圆柱体底面圆圆心，半径和高度自行设定，限定一个空间区域。	
区域原点偏置 (mm)		几何体原点在当前对应的工件坐标系原点基础上，进行X、Y、Z、绕X、绕Y、绕Z的方向进行偏置。	
	输出 IO 模块	工作区	机器人当前 TCP 点超出被激活的工作区时，输出对应的 IO 信号。 若不使用在输入框内输入-1
		障碍区	机器人当前 TCP 点进入被激活的障碍区时，输出对应的 IO 信号。 若不使用在输入框内输入-1
		干涉区	机器人当前 TCP 点进入被激活的干涉区时，输出对应的 IO 信号。 若不使用在输入框内输入-1

IO配置	输出 Gvar 序号	工作区	机器人当前 TCP 点超出激活工作区时，对应的 Gvar 序号赋值 1。 若不使用在输入框内输入-1
		障碍区	机器人当前 TCP 点进入被激活的障碍区时，对应的 Gvar 序号赋值 1。若不使用在输入框内输入-1
		干涉区	机器人当前 TCP 点进入被激活的干涉区时，对应的 Gvar 序号赋值 1。若不使用在输入框内输入-1
	输入 IO 模块 (只在干涉 区有效)	干涉区	当对应输入 IO 信号被触发时，机器人当前 TCP 点进入被激活的干涉区时，机器人紧急停止运动
	互锁处理 (只在干涉 区有效)		内含两个可选项： 1.继续输出：机器人当前 TCP 点进入被激活的干涉区时，对应输出 IO 信号继续输出； 2.忽略输出：假如输入信号已被触发，机器人当前 TCP 点进入被激活的干涉区时，对应输出 IO 信号不输出；

9.7.2.4 工作空间监控监测

手动按手持盒的查看界面 Fn1，选择{工作空间界面}，可查看到已激活对应的工作空间监控及已被触发对应的工作空间监控。



图表 46 工作空间监控监测界面

10 系统参数设置

点击左上角 Logo 打开{系统应用}，选择{系统参数}进行参数设置。

参数设置完成后点击下方的{保存}保存参数。

选择设置参数，可以通过{复制}{粘贴}按键进行快捷操作。

10.1系统参数



图表 47 系统参数界面

序号	参数名称	参数名称	参数说明
1	远程模式类型	IO 控制	即远程外部 IO 对机器人进行指令控制
		Modbus TCP 控制	上位机通过 Modbus TCP 与机器人进行通讯交互
		Remote TCP 控制	上位机通过 TCP/IP 与机器人进行通讯交互
2	手持连接状态检查	手持掉线不处理	手持盒掉线或曾经掉线后，无任何提示

		手持掉线报警	手持盒掉线后会提示“手持盒掉线或曾经掉线”的报警
3	指令自动加注释	关闭	关闭—表示不启用指令自动加注释功能；
		开启	开启—表示启用指令自动加注释； 编写程序时，设置指令行后会自动加注释
4	Movc2 前进示教检查轨迹	直线运动	示教运行圆指令程序时，设定以直线方式运行到圆中点
		圆弧运动	示教运行圆指令程序时，设定以圆弧方式运行到圆中点
5	碰撞自动松开抱闸	关闭	关闭—表示不使用松抱闸功能；
		开启	开启—表示使用松抱闸功能； 紧急情况下，若不能通过示教器操作机器人，此时可开启此功能，对机器人的四、五、六轴进行松抱闸操作（注意：必须做好安全防护）
6	Setpc 检查	关闭	关闭—表示不使用 Setpc 功能； 开启—表示使用 Setpc 检查；

		开启	每次切换模式后需要在文件编辑界面点击 Setpc 按键指定启动行
7	指定接受物理按键工艺	选择相对应的工艺使用	扩展按键功能受哪个工艺控制

10.2硬件参数

系统参数

硬件参数

关节

辅助轴

笛卡尔

规划参数

硬件参数

轴号选择：

1

▼

复制

粘贴

1. 编码器规划分辨率

0

2. 编码器反馈分辨率

0

3. 读绝对值编码器端口

COM0

▼

4. 编码器类型

NULL

▼

5. 周期读取绝对值编码器

0

6. 电机最大速度(rpm)

0

7. 电机最大电流(A)

0.0000

8. 电机力矩常数(Nm/A)

0.0000

9. 减速比关节端

0.0000

10. 减速比电机端

0.0000

11. 驱动器类型

NULL

▼

图表 48 硬件参数界面

序号	参数名称	参数说明
1	编码器规划分辨率	编码器线数例：17 位编码器的分辨率为 131072
2	编码器反馈分辨率	
3	读绝对值编码器端口	总线式控制值该参数无效
4	编码器类型	默认 GthdEcat
5	周期读取绝对值编码器	定时读取绝对值编码器数据用以刷新机器人的当

		前绝对位置
6	电机最大速度(rpm)	电机最大转速，实际运行过程中不允许超过该转速
7	电机最大电流(A)	电机最大电流，实际运行过程中不允许超过该电流
8	电机力矩常数(Nm/A)	单位电流产生的力矩
9	减速比关节端	电机端转动角度与执行端转动角度的比值（允许值为正负来改变关节运动方向）
10	减速比电机端	
11	驱动器类型	默认 GthdEcat

10.3关节

系统参数

硬件参数

关节

辅助轴

笛卡尔

规划参数

DH

耦合比

关节参数

轴号选择：

1

复制

粘贴

1. 关节类型

2. 关节正限位(deg)

3. 关节负限位(deg)

4. 关节零点偏置(deg)
(实际记录的机器人零位
相对标准模型的偏置)

5. 关节反向

6. 关节JOG最大速度(deg/s)

7. 关节JOG最加速度(deg/s^2)

8. 关节JOG最减速度(deg/s^2)

9. 关节运动最大速度(deg/s)

10. 关节运动最大加速度(deg/s^2)

11. 关节运动最大减速度(deg/s^2)

12. 轴硬件地址

旋转关节(deg)

170

-170

0.0000

无效

20

90

90

166

833

833

1

保存

全部重置

图表 49 关节参数界面

序号	参数名称	参数说明
1	关节类型	不使用； 旋转关节(deg)； 平移关节（mm）；
2	关节正限位	关节运动范围最大值
3	关节负限位	关节运动范围最小值
4	关节零点偏置	用户定义关节零点与系统内部机器人模型零点的 偏差值
5	关节反向	用户定义关节转动方向是否与系统内部机器人模

		型方向相反
6	关节 JOG 最大速度	关节坐标系下点动轴操作键运动速度最大值
7	关节 JOG 最大加速度	关节坐标系下点动轴操作键运动加速度最大值
8	关节 JOG 最大减速度	关节坐标系下点动轴操作键运动减速度最大值
9	关节运动最大速度	MOVJ 指令关节运动速度最大值
10	关节运动最大加速度	MOVJ 指令关节运动加速度最大值
11	关节运动最大减速度	MOVJ 指令关节运动减速度最大值
12	轴硬件地址	总线通讯使用的站号，要和上位机设置匹配

10.4辅助轴

辅助轴参数参照{关节}参数设置，不使用的辅助轴将参数 1{关节类型}选为“未使用”。

系统参数

硬件参数

关节

辅助轴

笛卡尔

规划参数

DH

耦合比

辅助轴参数

轴号选择：Ex1

复制

粘贴

1. 关节类型

不使用

2. 关节正限位(X)

60

3. 关节负限位(X)

-60

4. 关节零点偏置(X)
(实际记录的机器人零位
相对标准模型的偏置)

0.0000

5. 关节反向

无效

6. 关节JOG最大速度(X/s)

10

7. 关节JOG最大加速度(X/s^2)

90

8. 关节JOG最减速度(X/s^2)

90

9. 关节运动最大速度(X/s)

30

10. 关节运动最大加速度(X/s^2)

100

11. 关节运动最大减速度(X/s^2)

100

12. 轴硬件地址

7

保存

全部重置

图表 50 辅助轴参数界面

10.5笛卡尔

笛卡尔参数

1.X方向参数：

正限位(mm)

9999

负限位(mm)

-9999

JOG最大速度(mm/s)

100

JOG最大加速度(mm/s²)

100

JOG最大减速度(mm/s²)

100

2.Y方向参数：

正限位(mm)

9999

负限位(mm)

-9999

JOG最大速度(mm/s)

100

JOG最大加速度(mm/s²)

100

JOG最大减速度(mm/s²)

100

3.Z方向参数：

正限位(mm)

1400

负限位(mm)

358

JOG最大速度(mm/s)

100

JOG最大加速度(mm/s²)

100

JOG最大减速度(mm/s²)

100

图表 51 笛卡尔参数界面 1

4.RX方向参数：

正限位(deg)

360

负限位(deg)

-360

JOG最大速度(deg/s)

20

JOG最大加速度(deg/s²)

100

JOG最大减速度(deg/s²)

100

5.RY方向参数：

正限位(deg)

360

负限位(deg)

-360

JOG最大速度(deg/s)

20

JOG最大加速度(deg/s²)

100

JOG最大减速度(deg/s²)

100

6.RZ方向参数：

正限位(deg)

360

负限位(deg)

-360

JOG最大速度(deg/s)

20

JOG最大加速度(deg/s²)

100

7.末端最大合成速度：

线速度(mm/s)

2000

线加速度(mm/s²)

10000

线减速度(mm/s²)

10000

角速度(deg/s)

360

角加速度(deg/s²)

1500

角减速度(deg/s²)

1500

保存

全部重置

图表 52 笛卡尔参数界面 2

序号	参数名称	参数说明
1	正限位	笛卡尔坐标系下运动范围最大值
2	负限位	笛卡尔坐标系下运动范围最小值
3	JOG 最大速度	笛卡尔坐标系下点动轴操作键运动速度最大值

4	JOG 最大加速度	笛卡尔坐标系下点动轴操作键运动加速度最大值
5	JOG 最大减速度	笛卡尔坐标系下点动轴操作键运动减速度最大值
6	线速度	笛卡尔坐标系下点末端合成直线速度最大值
7	线加速度	笛卡尔坐标系下点末端合成直线加速度最大值
8	线减速度	笛卡尔坐标系下点末端合成直线减速度最大值
9	角速度	笛卡尔坐标系下点末端合成角速度最大值
10	角加速度	笛卡尔坐标系下点末端合成角加速度最大值
11	角减速度	笛卡尔坐标系下点末端合成角减速度最大值

10.6规划参数

系统参数

硬件参数

关节

辅助轴

笛卡尔

规划参数

DH

耦合比

规划参数

1. 规划器类型

Cos

2. 关节平滑窗口

0

3. 轨迹过渡段类型

路径混合

4. 跟随误差(round)

0

5. 示教模式下规划器加加速时间

32

6. 自动运行默认加加速时间

32

7. 自动运行默认速度百分比

1

8. 自动运行默认加加速度百分比

1

9. 自动运行默认减速度百分比

1

10. 示教模式最大关节速度(%)

1

11. 示教模式最大笛卡尔速度(mm/s)

1

12. 奇异点处理

报警停止

13. 上伺服延时(ms)

0

14. 开启示教速度自动调节

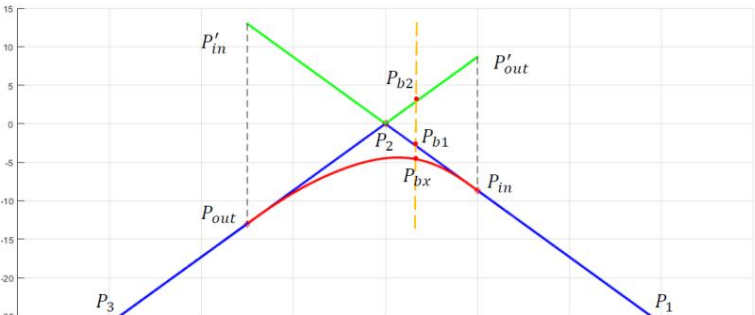
关闭

15. movl/c示教模式最低速度值(mm)

1.00

图表 53 规划参数界面

序号	参数名称	参数说明
1	规划器类型	Cos: Jerk 是 Cos 函数的轨迹规划 Quintic: Jerk 是 5 次曲线的轨迹规划 Door: Jerk 是方波的轨迹规划 SawTooth: Jerk 是锯齿形的轨迹规划 Exp: Jerk 是指数曲线的轨迹规划 Bell: Jerk 是二次曲线的轨迹规划 规划曲线参考下图
2	规划器平滑	输出均值滤波器长度

	窗口	
3	轨迹过渡段类型	Fusion: 路径混合过渡，速度和加速度不突变，轨迹可控，但速度无法恒定，默认选项 Arc: 圆弧过渡，保证过渡段速度恒定，轨迹可控，但会导致加速度突变。不适合关节轨迹和姿态规划。 Spline: 样条过渡，保证过渡段速度恒定，轨迹可控，但会导致加速度突变，突变小于 Arc 过渡。不适合关节轨迹和姿态规划 VelBlend: 速度过渡，即两条运动指令同时规划，进行速度叠加，得到新的位置。轨迹形状和进入 Blend 段时速度相关  Fig: 末端位置路径混合
4	跟 随 误 差 (round)	从开始运动到实际位置的时间段内的位置命令与实际位置的差值
5	示教模式下 规划器的加 加速时间	手动 jog 时候用的加加速度时间，调大了 jog 会比较猛，容易抖
6	自动运行默 认加加速时 间	回放时，如果程序不调用 SetDyn，那么默认的增加速度时间就是用这个值，就是 SetDyn 里的 jerktime

7	自动运行默认速度百分比	开机默认的回放模式运行速度
8	自动运行默认加速度百分比	开机默认的回放模式运行加速度
9	自动运行默认减速度百分比	开机默认的回放模式运行减速度
10	10. 示教模式最大关节速度(%)	示教模式下限制最大关节运动速度
11	示教模式最大笛卡尔速度(mm/s)	示教模式下笛卡尔空间运动最大速度
12	奇异点处理	当机器人经过奇异点时所选择的策略，可选择经过奇异点时报警停止或者减速缓慢经过奇异点
13	上伺服延迟(ms)	增加上伺服的延迟
14	开启示教速度自动调节	示教速度自动调节开关，与 15 项搭配使用
15	Movl/ 示教	提升 Movl/c 设置指令速度低时，使用前进检查下的最低速度值

	模式最低速度 度 值 调 节 (mm)	
--	---------------------------	--

规划器类型图例：

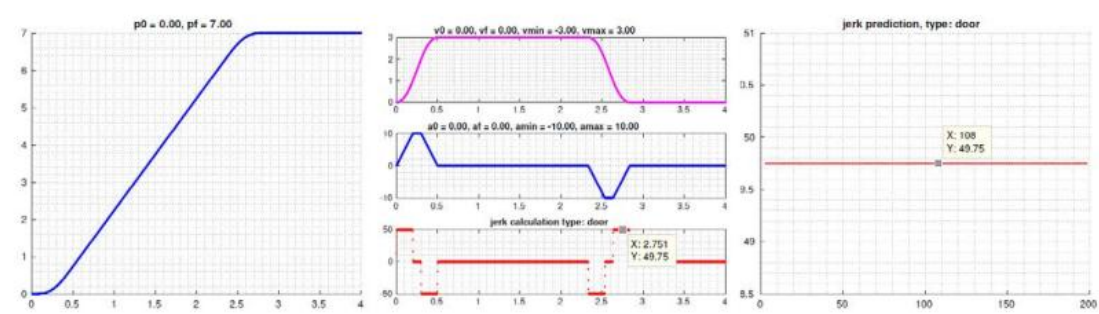


Figure 1: door-shaped jerk

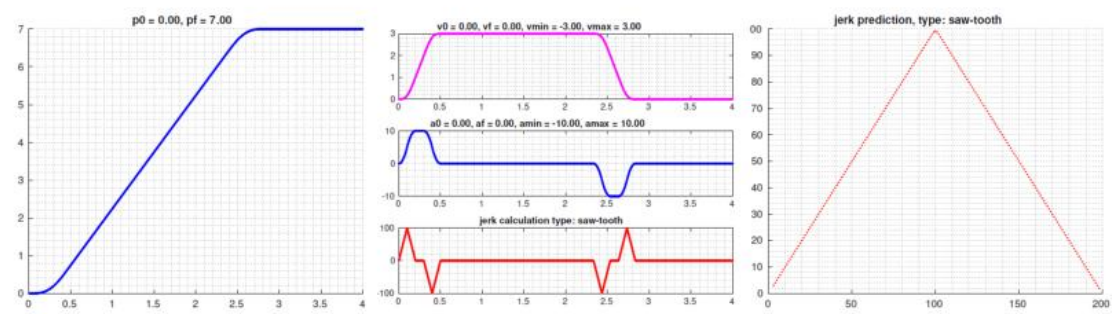


Figure 2: saw-tooth-shaped jerk

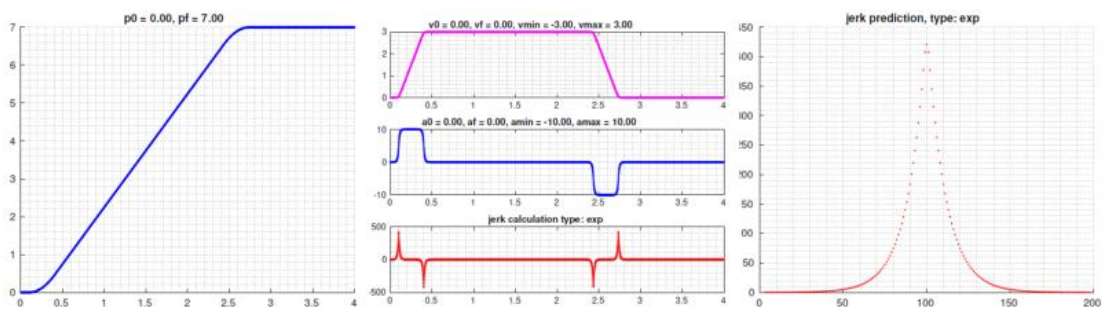


Figure 3: exp-shaped jerk

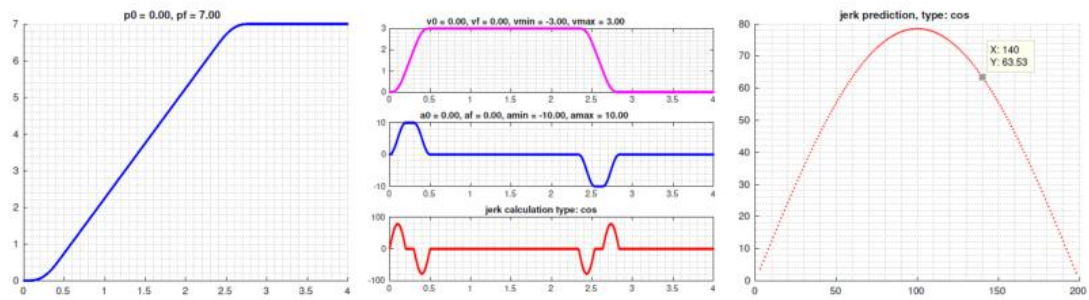


Figure 4: cos-shaped jerk

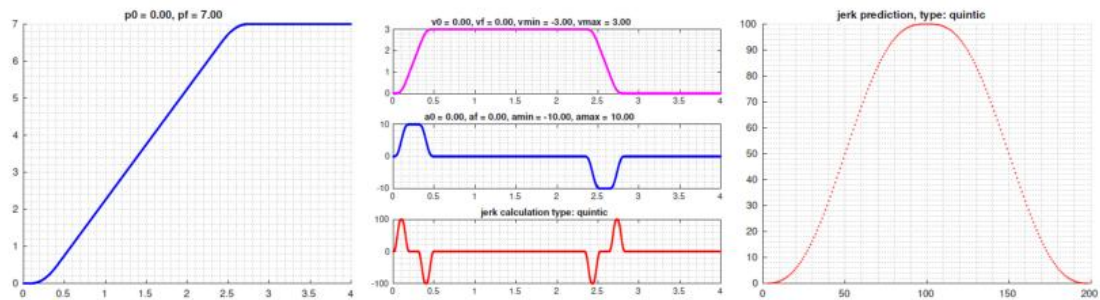


Figure 5: quintic-shaped jerk

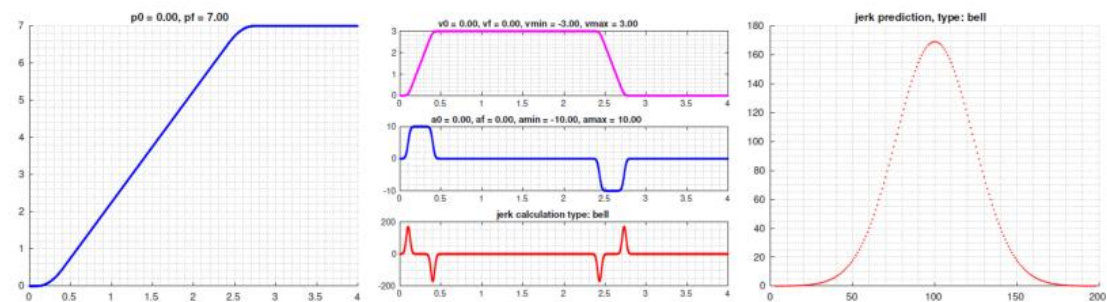
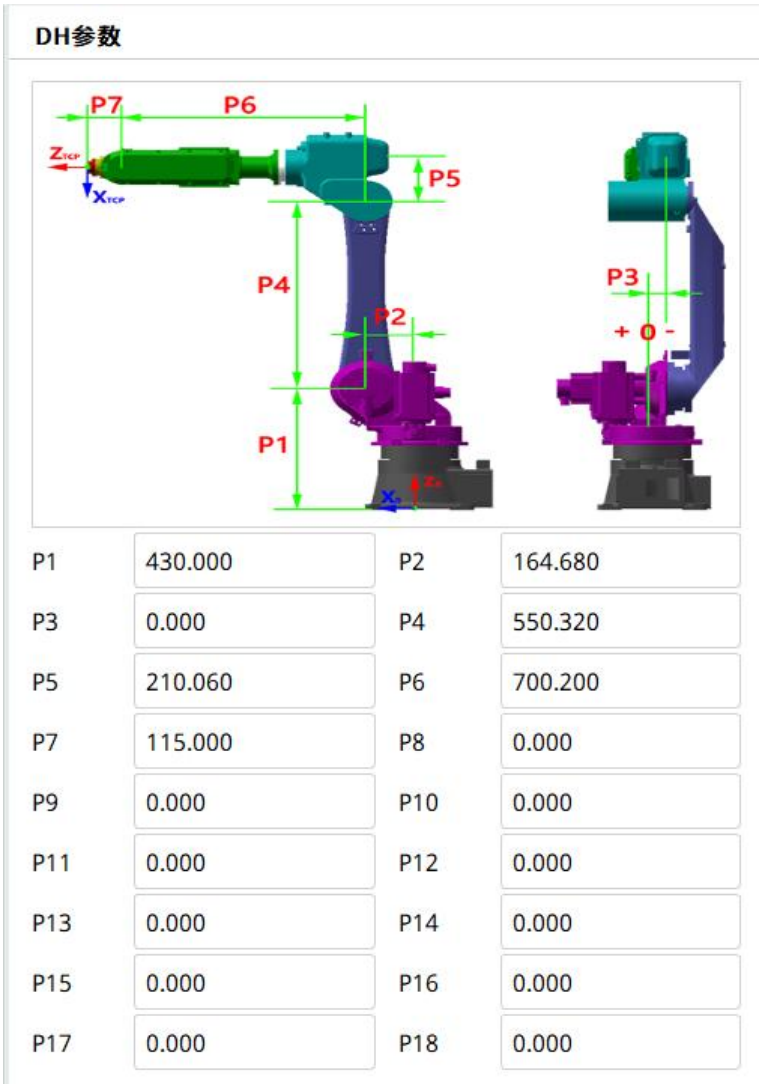


Figure 6: bell-shaped jerk

图表 54 规划曲线

10.7DH 参数



图表 55 DH 参数界面

序号	参数名称	参数说明
1	P1~P7	请依照图示填写各项参数信息
2	P8/P9	轴 4 与轴 5 之间的耦合比
3	P10/P11	表示轴 6 与轴 4 之间的耦合比
4	P12/P13	表示轴 6 与轴 5 之间的耦合比
5	P14~P18	预留

10.8耦合

耦合比

序号	主动轴	从动轴	耦合比
1	0	0	0.00000
2	0	0	0.00000
3	0	0	0.00000
4	0	0	0.00000
5	0	0	0.00000
6	0	0	0.00000
7	0	0	0.00000
8	0	0	0.00000
9	0	0	0.00000
10	0	0	0.00000

保存

全部重置

图表 56 耦合比设置界面

11 通用 IO

通用 IO 主要是配置防碰撞、暂停、重启、报警复位和安全门等通用信号。

11.1 输入信号

11.1.1 界面

数字量输入 数字量输出					
序号	定义	模组	点位	有效值	状态
1	防碰撞	1	0	1	
2	外部禁止运动	1	0	1	
3	暂停	1	0	1	
4	安全门	1	0	1	
5	急停	1	0	1	
6	高温报警	1	0	1	
7	电池报警	1	0	1	

图表 57 IO 输入界面

模组：表示 IO 模块地址 ID，扩展 IO 从站地址 ID 取值从 1 开始。

点位：表示使用该模组的第几位，0 表示不使用，点位取值从 1 开始。

有效值：正常 1 为有效，可以取反。

状态：点位输入状态指示灯，信号触发时显示为绿灯，无信号时显示为灰色。

11.1.2 功能说明

序号	输入输出类型	参数名称	说明
1	数字量输入	防碰撞信号	系统检测到防碰撞信号触发后，紧急停止运动，提示防碰撞信号触发报警。
2	数字量输入	外部禁止机器人	系统检测到外部禁止机器人信号触发后，黄色警告提示，机器人禁止运动。

3	数字量输入	暂停	系统检测到暂停信号触发后，系统暂停执行示教文件。
4	数字量输入	安全门	系统检测到安全门信号触发后，紧急停止运动，提示安全门报警。
5	数字量输入	急停	系统检测到急停信号触发后，紧急停止运动，提示急停报警。
6	数字量输入	高温报警	系统检测到高温报警信号触发后，停止运动，提示高温报警。
7	数字量输入	电池报警	系统检测到电池报警信号触发后，停止运动，提示电池报警。

11.2输出信号

11.2.1 界面

序号	定义	模组	点位	有效值	状态
1	伺服使能	1	0	1	
2	空闲	1	0	1	
3	报警	1	0	1	
4	暂停	1	0	1	
5	工作中	1	0	1	
6	警告	1	0	1	
7	运动中	1	0	1	
8	示教模式	1	0	1	
9	回放模式	1	0	1	
10	远程模式	1	0	1	
11	手持盒暂停	1	0	1	
12	手持盒急停	1	0	1	
13	手持盒启动	1	0	1	
14	三段开关	1	0	1	
15	工作站急停	1	0	1	
16	工作站暂停	1	0	1	

图表 58 IO 输出界面

模组：表示 IO 模块地址 ID，扩展 IO 从站地址 ID 取值从 1 开始。

点位：表示使用该模组的第几位，0 表示不使用，点位取值从 1 开始。

有效值：正常 1 为有效，可以取反。

状态：点位输入状态指示灯，信号触发时显示为绿灯，无信号时显示为灰色。

11.2.2 功能说明

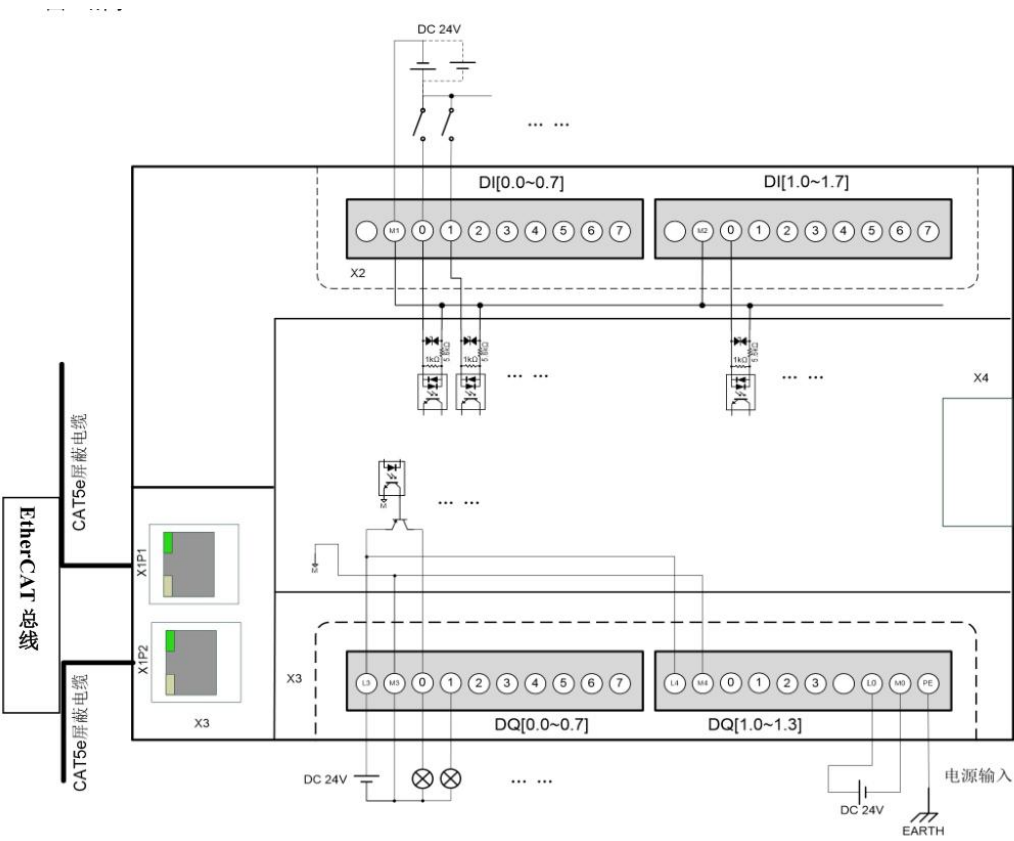
序号	输入输出类型	参数名称	说明
1	数字量输出	伺服使能	机器人使能后输出信号
2	数字量输出	空闲	使能处于待机状态

3	数字量输出	报警（红色报警）	机器人处于红色报警异常状态
4	数字量输出	暂停	机器人处于暂停状态
5	数字量输出	工作中	机器人处于执行示教文件状态。
6	数字量输出	警告（黄色警示）	机器人处于警告提示状态
7	数字量输出	运动中	机器人处于运动中状态
8	数字量输出	示教模式状态	手持示教器处于示教模式时输出信号
9	数字量输出	回放模式状态	手持示教器处于回放模式时输出信号
10	数字量输出	远程模式状态	手持示教器处于远程模式时输出信号
11	数字量输出	手持盒急停输出	手持盒急停信号触发时输出信号。
12	数字量输出	手持盒暂停输出	手持盒暂停信号触发时输出信号。
13	数字量输出	手持盒启动输出	手持盒启动信号触发时输出信号。
14	数字量输出	手持盒三段开关输出	手持盒三段开关使能按钮信号触发时输出信号。
15	数字量输出	工作站急停	收到工作站急停输入信号, 且系统处于工作站急停状态。
16	数字量输出	工作站暂停	收到工作站暂停输入信号, 且系统处于工作站暂停状态。

11.3IO 接线

以 R51C1-EA/pro 为例，具体 IO 接线请参考使用说明书。

R51C1-EA/pro 接线图



图表 59 IO 接线

Ecat：网线连接

电源：24V 供电，如图右下角

输出：L3/L4 (QL0/QL1) 接 24V，M3/M4 (QM0/QM1) 接 0V，输出的是 24V

输入：如果 M1/M2 (IM0/IM1) 接 0V，则需输入 24V；

如果 M1/M2 (IM0/IM1) 接 24V，则需输入 0V；

12 远程 IO

远程 IO 主要应用于远程模式，如远程运行外部启动文件、预约文件、外部启动确认等属于远程操作相关的信号。

12.1 界面介绍

12.1.1 文件设置

文件设置

远程IO输入

远程IO输出

基本设置

文件号	工程名称		程序名称		优先级	完成次数	附加任务
1	io	▼	main	▼	0	0	
2	demo	▼	预约二2	▼	0	0	qingqiang2
3	demo	▼	order1	▼	0	0	qingqiang2
4	io	▼	999	▼	0	0	
5	demo	▼	main	▼	0	0	
6	io	▼	二	▼	0	0	
7		▼		▼	0	0	

图表 60 IO 文件设置

文件号：通过远程输入文件位组合出来的文件号。二进制模式下，只有 1~8 文件号有效，组合模式下，1~255 号文件有效。

工程名称、程序名称：文件号对应的工程名称和程序名称，如果启动文件，将执行文件号对应的工程名称和程序名称的程序。

优先级：显示文件执行的顺序，优先级为 1，为正在执行的程序，在单独模式下，优先级 2~8 为预约中的程序；组合模式下只有优先级为 1 的程序。

完成次数：预约文件程序执行的次数，主要用于计件统计，可以手动修改。

附加任务：当对应文件号完成次数达到设置的执行次数的整数倍时，自动触发执行一次附加任务。

12.1.2 远程 IO 输入

文件设置 远程IO输入 远程IO输出 基本设置					
序号	输入定义	模组	点位	有效值	状态
1	伺服使能	1	4	1	
2	文件位1	1	0	1	
3	文件位2	1	0	1	
4	文件位3	1	0	1	
5	文件位4	1	0	1	
6	文件位5	1	0	1	
7	文件位6	1	0	1	
8	文件位7	1	0	1	
9	文件位8	1	0	1	
10	启动确认	1	0	1	
11	取消预约	1	0	1	
12	报警复位	1	0	1	
13	远程暂停	1	0	1	
14	远程急停	1	0	1	

← 上一页

1 / 1

下一页 →

保存

图表 61 远程 IO 输入

- 模组：表示 IO 模块地址 ID，扩展 IO 从站地址 ID 取值从 1 开始。
- 点位：表示使用该模组的第几位，0 表示不使用，点位取值从 1 开始。
- 有效值：正常 1 为有效，可以取反。
- 状态：点位输入状态指示灯，信号触发时显示为绿灯，无信号时显示为会色。

功能说明：

序号	输入输出类型	参数名称	说 明
----	--------	------	-----

1	远程 IO 输入	伺服使能	远程 IO 控制模式下，系统收到此信号机器人使能。
2	远程 IO 输入	文件位 1	文件号输入位 1，如果是单独模式代表文件位 1；如果是组合模式，表示 8421 编码的第一位。
3	远程 IO 输入	文件位 2	文件号输入位 2，如果是单独模式代表文件位 2；如果是组合模式，表示 8421 编码的第二位。
4	远程 IO 输入	文件位 3	文件号输入位 3，如果是单独模式代表文件位 3；如果是组合模式，表示 8421 编码的第三位。
5	远程 IO 输入	文件位 4	文件号输入位 4，如果是单独模式代表文件位 4；如果是组合模式，表示 8421 编码的第四位。
6	远程 IO 输入	文件位 5	文件号输入位 5，如果是单独模式代表文件位 5；如果是组合模式，表示 8421 编码的第五位。
7	远程 IO 输入	文件位 6	文件号输入位 6，如果是单独模式代表文件位 6；如果是组合模式，表示 8421 编码的第六位。
8	远程 IO 输入	文件位 7	文件号输入位 7，如果是单独模式代表文件位 7；如果是组合模式，表示 8421 编码的第七

			位。
9	远程 IO 输入	文件位 8	文件号输入位 8，如果是单独模式代表文件位 8；如果是组合模式，表示 8421 编码的第八位。
10	远程 IO 输入	启动确认	组合模式下，文件启动信号。单独模式下此信号为无效信号。
11	远程 IO 输入	取消预约	取消所有已预约（包含对应执行或预约文件的附加任务）
12	远程 IO 输入	报警复位	清除报警信号
13	远程 IO 输入	远程暂停	暂停机器人运动
14	远程 IO 输入	远程急停	紧急急停信号

12.1.3 远程 IO 输出

文件设置 远程IO输入 远程IO输出 基本设置					
序号	输出定义	模组	点位	有效值	状态
1	伺服使能	1	0	1	
2	空闲	1	0	1	
3	工作中	1	0	1	
4	暂停	1	0	1	
5	报警	1	0	1	
6	警告	1	0	1	
7	运动中	1	0	1	
8	文件位1	1	0	1	
9	文件位2	1	0	1	
10	文件位3	1	0	1	
11	文件位4	1	0	1	
12	文件位5	1	0	1	
13	文件位6	1	0	1	
14	文件位7	1	0	1	
15	文件位8	1	0	1	
16	启动确认	1	0	1	

← 上一页

1 / 1

下一页 →

保存

图表 62 远程 IO 输出

- 模组：**表示 IO 模块地址 ID，扩展 IO 从站地址 ID 取值从 1 开始。
- 点位：**表示使用该模组的第几位，0 表示不使用，点位取值从 1 开始。
- 有效值：**正常 1 为有效，可以取反。
- 状态：**点位输入状态指示灯，信号触发时显示为绿灯，无信号时显示为会色。

功能说明：

序号	输入输出类型	参数名称	说明
----	--------	------	----

1	远程 IO 输出	伺服使能	机器人使能后输出信号
2	远程 IO 输出	空闲	使能处于待机状态
3	远程 IO 输出	报警（红色报警）	机器人处于红色报警异常状态
4	远程 IO 输出	暂停	机器人处于暂停状态
5	远程 IO 输出	工作中	机器人处于执行示教文件状态。
6	远程 IO 输出	警告（黄色警示）	机器人处于警告提示状态
7	远程 IO 输出	运动中	机器人处于运动中状态
8	远程 IO 输出	文件位 1	文件位 1 处于预约中
9	远程 IO 输出	文件位 2	文件位 2 处于预约中
10	远程 IO 输出	文件位 3	文件位 3 处于预约中
11	远程 IO 输出	文件位 4	文件位 4 处于预约中
12	远程 IO 输出	文件位 5	文件位 5 处于预约中
13	远程 IO 输出	文件位 6	文件位 6 处于预约中
14	远程 IO 输出	文件位 7	文件位 7 处于预约中
15	远程 IO 输出	文件位 8	文件位 8 处于预约中
16	远程 IO 输出	启动确认	远程输入启动确认被按下

12.1.4 基本设置

文件设置

远程IO输入

远程IO输出

基本设置

组合

▼

☐ 模式切换复位预约

☐ 组合模式启动时需确认文件号

图表 63 基本设置

远程 IO 工作模式有两种，1、不使用，2、单独模式，3、组合模式。

文件设置

远程IO输入

远程IO输出

基本设置

不使用

▼

不使用

单独

组合

图表 64 远程 IO 工作模式

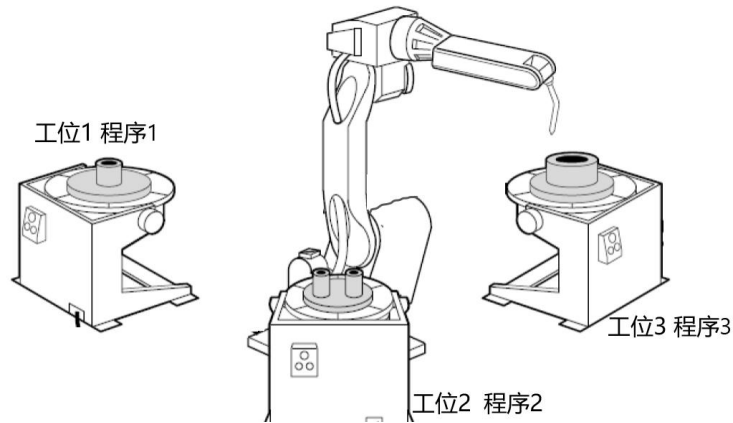
模式切换复位预约和组合模式启动时需确认信号是备用功能。

模式切换复位预约如果勾选，则在手持盒模式切换的时候，会清除所有预约，切换回远程模式时，需重新预约。

组合模式启动时需确认信号如果勾选，组合模式启动时要检测到文件位信号（即对应文件信号（组合）持续输入），启动确认信号，工位文件开始执行后，方可撤销文件位输入信号。

12.2 单独模式使用

12.2.1 应用场景



上图所示的三个工位各自处理不同工件的情况，事先要做如下准备工作：

在工装 1 上，执行工件 1 的程序 1

在工装 2 上，执行工件 2 的程序 2

在工装 3 上，执行工件 3 的程序 3

再现时，切换到“REMOTE”模式，准备好工件 1 后，按下工装 1 上的启动按钮，机器人执行程序 1。在执行程序 1 时，准备好工件 2、工件 3，然后依次按工装 2、工装 3 上的启动按钮。此时，即使程序 1 还在运行中，由于已按了启动按钮，程序 2、程序 3，按顺序进行了作业预约，程序将按照预约顺序被依次执行。

系统最多实现 8 个工位的预约。

12.2.2 操作步骤

12.2.2.1 设置远程工作模式

进入{远程 IO}-{基本设置}-{远程 IO 模式}设置为{单独}模式。

12.2.2.2 设置远程文件名称

步骤一 创建工位程序

进入{主菜单}-{工程管理}-新建 demo 工程，新建工位程序，程序名称为 a1，b2，c3。

步骤二 设置远程文件

进入{远程 IO}界面-选择{文件设置}属性页，设置如下：

文件设置

远程IO输入

远程IO输出

基本设置

文件号	工程名称		程序名称		优先级	完成次数	附加任务
1	demo	▼	a1	▼	0	0	
2	demo	▼	b2	▼	0	0	
3	demo	▼	c3	▼	0	0	
4		▼		▼	0	0	
5		▼		▼	0	0	
6		▼		▼	0	0	
7		▼		▼	0	0	

12.2.2.3 设置远程 IO 输入

接入 ethercat IO，使用输入点 1.1~1.15 进行配置。

远程操作机器人进行 3 个工位工作，文件位 1、文件位 2、文件位 3、使能信号是必须要设置的，取消预约、报警复位、远程暂停、远程急停可以根据需要进行设定。

文件设置	远程IO输入	远程IO输出	基本设置		
序号	输入定义	模组	点位	有效值	状态
1	文件位1	1	1	1	
2	文件位2	1	2	1	
3	文件位3	1	3	1	
4	文件位4	1	0	1	
5	文件位5	1	0	1	
6	文件位6	1	0	1	
7	文件位7	1	0	1	
8	文件位8	1	0	1	
9	伺服使能	1	4	1	
10	启动确认	1	5	1	
11	取消预约	1	6	1	
12	报警复位	1	7	1	
13	远程暂停	1	8	1	
14	远程急停	1	9	1	

12.2.2.4 设置远程 IO 输出

接入 ethercat IO，使用输出点 1.1~1.16 进行配置。

根据外部状态需要设置远程 IO 输出信号。

文件设置	远程IO输入	远程IO输出	基本设置		
序号	输出定义	模组	点位	有效值	状态
1	伺服使能	1	1	1	
2	空闲	1	2	1	
3	工作中	1	3	1	
4	暂停	1	4	1	
5	报警	1	5	1	
6	警告	1	6	1	
7	运动中	1	7	1	
8	文件位1	1	8	1	
9	文件位2	1	9	1	
10	文件位3	1	10	1	
11	文件位4	1	11	1	
12	文件位5	1	12	1	
13	文件位6	1	13	1	
14	文件位7	1	14	1	
15	文件位8	1	15	1	
16	启动确认	1	16	1	

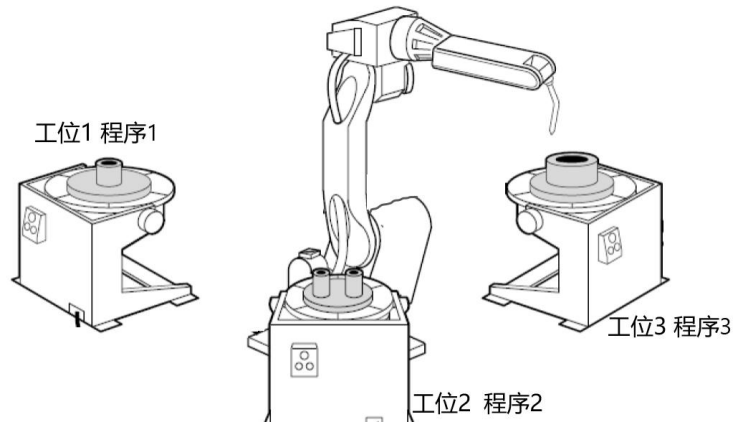
12.2.2.5 开始预约操作

举例：依次执行工位 1、工位 2、工位 3 程序，工作流程：

完成步骤 1-4 的设置后，切换到远程模式——给使能输入信号 1.4——等待机器人正常使能——给工位 1 输入信号 1.1——工位 1 开始工作中，等待执行——给工位 2 输入信号 1.2——工位 2 预约中，等待执行——给工位 3 输入信号 1.3——工位 3 预约中，等待执行。

12.3 组合模式使用

12.3.1 应用场景



上图所示的三个工位各自处理不同工件的情况，事先要做如下准备工作：

在工装 1 上，执行工件 1 的程序 1

在工装 2 上，执行工件 2 的程序 2

在工装 3 上，执行工件 3 的程序 3

只有两个文件位可用。

组合模式系统最多实现 255 个工序文件。

12.3.2 操作步骤

12.3.2.1 设置远程工作模式

进入{远程 IO}-{基本设置}-{远程 IO 模式}设置为{组合}模式。

12.3.2.2 设置远程文件名称

步骤一 创建工位程序

进入{主菜单}-{工程管理}-新建 demo 工程，新建工位程序，程序名称为 a1，

b2, c3。

步骤二 设置远程文件

进入{远程 IO}界面-选择{文件设置}属性页， 设置如下：

文件设置

远程IO输入

远程IO输出

基本设置

文件号	工程名称		程序名称		优先级	完成次数	附加任务
1	demo	▼	a1	▼	0	0	
2	demo	▼	b2	▼	0	0	
3	demo	▼	c3	▼	0	0	
4		▼		▼	0	0	
5		▼		▼	0	0	
6		▼		▼	0	0	
7		▼		▼	0	0	

12.3.2.3 设置远程 IO 输入

接入 ethercat IO， 使用输入点 1.1~1.15 进行配置。

远程操作机器人进行 3 个工位工作， 文件位 1、文件位 2、文件位 3、使能信号是必须要设置的， 取消预约、报警复位、远程暂停、远程急停可以根据需要进行设定。

文件设置	远程IO输入	远程IO输出	基本设置		
序号	输入定义	模组	点位	有效值	状态
1	文件位1	1	1	1	
2	文件位2	1	2	1	
3	文件位3	1	3	1	
4	文件位4	1	0	1	
5	文件位5	1	0	1	
6	文件位6	1	0	1	
7	文件位7	1	0	1	
8	文件位8	1	0	1	
9	伺服使能	1	4	1	
10	启动确认	1	5	1	
11	取消预约	1	6	1	
12	报警复位	1	7	1	
13	远程暂停	1	8	1	
14	远程急停	1	9	1	

12.3.2.4 设置远程 IO 输出

接入 ethercat IO，使用输出点 1.1~1.16 进行配置。

根据外部状态需要设置远程 IO 输出信号。

文件设置	远程IO输入	远程IO输出	基本设置		
序号	输出定义	模组	点位	有效值	状态
1	伺服使能	1	1	1	
2	空闲	1	2	1	
3	工作中	1	3	1	
4	暂停	1	4	1	
5	报警	1	5	1	
6	警告	1	6	1	
7	运动中	1	7	1	
8	文件位1	1	8	1	
9	文件位2	1	9	1	
10	文件位3	1	10	1	
11	文件位4	1	11	1	
12	文件位5	1	12	1	
13	文件位6	1	13	1	
14	文件位7	1	14	1	
15	文件位8	1	15	1	
16	启动确认	1	16	1	

12.3.2.5 开始预约操作

举例：依次执行工位 1、工位 2、工位 3 程序，工作流程：

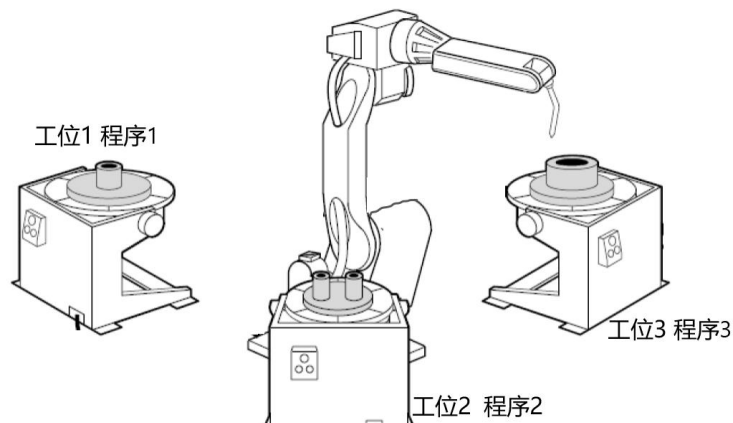
启动工位 1 工作流程：完成步骤 1-4 的设置后，切换到远程模式——给使能输入信号 1.4——等待机器人正常使能——设置文件位 1 信号 1.1——工位 1 预约完成——给启动确认信号 1.5——机器人开始工位 1（demo.a1）工作中，等待执行——工作完成。

启动工位 2 工作流程：完成步骤 1-4 的设置后，切换到远程模式——给使能输入信号 1.4——等待机器人正常使能——给文件位 2 输入信号 1.2——工位 2 预约完成——给启动确认信号 1.5——机器人开始工位 2（demo.b2）工作中，等待执行——工作完成。

启动工位 3 工作流程：完成步骤 1-4 的设置后，切换到远程模式——给使能输入信号 1.4——等待机器人正常使能——给文件位 1 输入信号 1.1，给文件位 2 输入信号 1.2（文件位 1 和文件位 2 组合出来工位 3 文件）——工位 3 预约完成——给启动确认信号 1.5——机器人开始工位 3（demo.c3）工作中，等待执行——工作完成。

12.4 附加任务使用

12.4.1 应用场景



上图所示的二个工位各自处理不同工件的情况，事先要做如下准备工作：

在工装 1 上，执行工件 1 的程序 1

在工装 2 上，执行工件 2 的程序 2

附加任务可基于单独模式和组合模式下进行，系统最多能添加 256 个附加任务。

12.4.2 操作步骤

12.4.2.1 设置远程工作模式

进入{远程 IO}-{基本设置}-{远程 IO 模式}，根据需要可设置为{单独}或者{组合}模式。

12.4.2.2 设置远程文件名称

步骤一 创建工位程序

进入{主菜单}-{工程管理}-新建 demo 工程，新建工位程序，程序名称为 a1，
b2，c3。

步骤二 设置远程文件

进入{远程 IO}界面-选择{文件设置}属性页，设置如下：

文件设置

远程IO输入

远程IO输出

基本设置

文件号	工程名称		程序名称		优先级	完成次数	附加任务
1	demo	▼	a1	▼	0	0	
2		▼		▼	0	0	
3		▼		▼	0	0	
4		▼		▼	0	0	
5		▼		▼	0	0	
6		▼		▼	0	0	

步骤三 设置附加任务

点击对应文件号所在行{附加任务}按钮，进入设置页面，设置如下：

文件设置

远程IO输入

远程IO输出

基本设置

文件号	工程名称		程序名称		优先级	完成次数	附加任务
1	demo	▼	a1	▼	0	0	
2		▼		▼	0	0	
3		▼		▼	0	0	
4		▼		▼	0	0	
5		▼		▼	0	0	
6		▼		▼	0	0	
7		▼		▼	0	0	

附加任务设置

附加任务设置

文件号:

1

工程名称:

demo

程序名称:

a1

☒ 是否启用附加任务

附加任务:

b2

附加任务对应文件号执行次数:

2

修改

取消

启用附加任务功能，需勾选**是否启用附加任务**选项，选择需要添加的**附加任务**，
和**输入附加任务对应文件号执行次数**，点击修改即可完成配置。

文件设置 远程IO输入 远程IO输出 基本设置						
文件号	工程名称	程序名称	优先级	完成次数	附加任务	
1	demo	a1	0	0	b2	
2			0	0		
3			0	0		
4			0	0		
5			0	0		
6			0	0		
7			0	0		

12.4.2.3 设置远程 IO 输入

接入 ethercat IO，使用输入点 1.1~1.15 进行配置。

远程操作机器人进行 3 个工位工作，文件位 1、文件位 2、文件位 3、使能信号是必须要设置的，取消预约、报警复位、远程暂停、远程急停可以根据需要进行设定。

文件设置 远程IO输入 远程IO输出 基本设置					
序号	输入定义	模组	点位	有效值	状态
1	伺服使能	1	1	1	
2	文件位1	1	2	1	
3	文件位2	1	3	1	
4	文件位3	1	4	1	
5	文件位4	1	5	1	
6	文件位5	1	6	1	
7	文件位6	1	7	1	
8	文件位7	1	8	1	
9	文件位8	1	9	1	
10	启动确认	1	10	1	
11	取消预约	1	11	1	
12	报警复位	1	12	1	
13	远程暂停	1	13	1	
14	远程急停	1	14	1	

← 上一页

1 / 1

→ 下一页

 保存

12.4.2.4 设置远程 IO 输出

接入 ethercat IO，使用输出点 1.1~1.16 进行配置。

根据外部状态需要设置远程 IO 输出信号。

文件设置	远程IO输入	远程IO输出	基本设置		
序号	输出定义	模组	点位	有效值	状态
1	伺服使能	1	1	1	
2	空闲	1	2	1	
3	工作中	1	3	1	
4	暂停	1	4	1	
5	报警	1	5	1	
6	警告	1	6	1	
7	运动中	1	7	1	
8	文件位1	1	8	1	
9	文件位2	1	9	1	
10	文件位3	1	10	1	
11	文件位4	1	11	1	
12	文件位5	1	12	1	
13	文件位6	1	13	1	
14	文件位7	1	14	1	
15	文件位8	1	15	1	
16	启动确认	1	16	1	

12.4.2.5 开始附加任务操作

举例：附加任务（文件号 b2）的工作流程：

单独模式/连续模式下，启动两次工位 1（文件号 a1），累积两次工作完成后，系统自动启动附加任务工位 2（文件号 b2）。

注意：每次执行完对应文件号的整数倍次数时，附加任务都会自动启动。

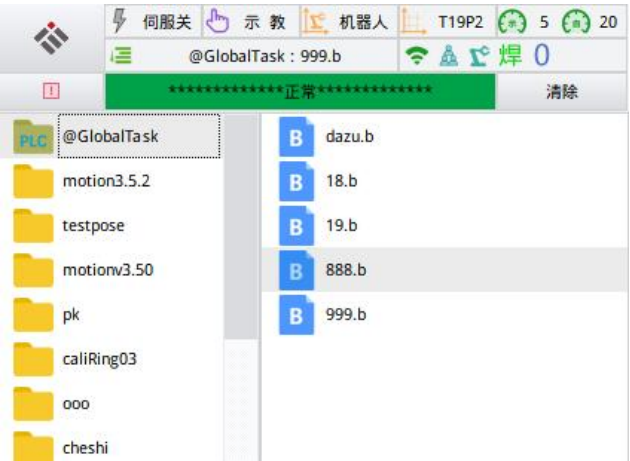
13 后台任务

13.1后台任务功能简介

后台任务功能是内置于控制器的一个虚拟 PLC，可以进行逻辑指令的编写，但是不能处理机器人运动指令。控制器一共内置 10 个后台程序，后台任务可以与主程序并行运行，二者可以进行 IO 信号、总线数据、自定义数据的交互。

示教器控制面板上的开始和停止按钮只能用来控制任务中加载程序的启动与停止，后台任务的开始、暂停及单步，必须在任务管理页进行。

13.2编辑后台任务



图表 65 后台任务文件界面

在工程管理界面中的@GlobalTask 为后台任务编辑文件夹，在此文件夹下创建的文件为后台任务文件，不支持添加机器人运动指令，其余操作与正常程序编辑一致。

13.3后台任务配置



图表 66 后台任务配置界面

- 1. 后台任务的程序在工程"@GlobalTask"中， 若该工程不存在请先创建。
- 2. 最多支持 10 个后台任务同时运行。
- 3. 【首行】 -启动时从首行开始运行【停止行】 -从上次的停止行继续运行。
- 4. 【手持同步】 -随主程序一起运行【开机启动】 -开机后运行。
- 5. 【启用】 -使用该后台任务【不启用】 -不使用。

14 动力学设置

动力学设置页面用来设置机器人动力学相关参数。动力学设置跟机器人拖动示教、碰撞检测等功能有关联。请确保机器人动力学设置合理，否则可能出现上述功能无法正常使用或机器人异常抖动的现象。

14.1全局参数

全局参数

类型

珞石

重力方向(x y z)

0

0

-1

反馈平滑窗口

21

拖动最大速度

0.85000

输出最大扭矩

0.00000

是否开启动力学

开启

前馈开启

关闭

日志开启

开启

启用本体拖动按钮

OFF

拖动模式

OFF

保持伺服

OFF

保存

图表 67 全局参数界面

序号	参数名称	参数说明
1	输入类型	动力学不同的输入反馈参数类型
2	重力方向(XYZ)	决定了机器人运动过程中的受力情况, 如倾斜安装或倒挂安装,需要正确设置重力参数。错误的设置会导致运动异常和报警

3	反馈平滑窗口	输出均值滤波器长度
4	拖动最大速度	拖动模式下运动最大速度
5	输出最大扭矩	拖动模式下运动最大扭矩
6	是否开启动力学	开启动力学功能
7	前馈开启	开启动力学前馈功能
8	日志开启	开启动力学日志
9	启用本体拖动按钮	开启力控传感器按钮
10	拖动模式	进入拖动模式
11	保持伺服	伺服保持

14.2动力学关节参数

电机参数包含额定电流、最大电流、额定力矩、最大力矩、扭矩常数、减速比、力传感器零点等机器人本体相关数据。该页面的参数直接影响运动的准确性请确保机器人电机参数设置合理,否则可能出现功能无法正常使用或机器人参数辨识失败的现象,如果没有机器人厂家的许可和协助,请勿修改。

动力学参数表示各个关节的质量、质心和惯量,可通过本体辨识自动获取动力学参数。

轴 号

1

▼

电机参数

额定电流

5.60000

最大电流

0.00000

额定力矩

0.63000

最大力矩

0.00000

扭矩常数

0.00000

减速比关节

1.00000

减速比电机

120.00000

动力学参数

link质量(kg)

1.00000

link质心(m)

x

0.00000

y

0.00000

z

0.00000

link惯量矩阵(kg*m²)

xx

0.00000

xy

0.00000

xz

0.00000

yy

0.00000

yz

0.00000

zz

0.21095

复制

粘贴

保存

图表 68 电机参数界面

14.3动力学参数辨识

14.3.1 本体辨识

本体辨识操作是通过让机器人执行一系列预设轨迹，采集运动过程中的信息。支持从零获取机器人动力学参数，包括：各关节末端负载、各关节负载惯量参数、各关节一阶摩擦力模型参数。

轴号

1

▼

质心

link质量(kg)

1.00000

link质心(m)

0.00000

0.00000

0.00000

link惯量矩阵(kg*m²)

0.00000

0.00000

0.00000

0.00000

0.00000

0.21095

摩擦力参数

摩擦力类型

一阶摩擦力模型

▼

正向粘性摩擦力

5.69113

负向粘性摩擦力

5.26100

正向哥伦布摩擦力

2.17749

负向哥伦布摩擦力

2.40407

二阶粘性摩擦力

0.00000

工具辨识参数

频率系数(0-1.5)

1.000

起始偏置(度)

0.00

幅值系数(0-1.5)

1.000

全参数辨识

保存

图表 69 本体辨识参数界面

序号	步骤	说明
1	设置频率系数 (0-1.5)	控制辨识过程运动速度 取值范围：0.1（低速模式）至 1.5（高速模式） 建议首次使用采用 0.3 频率验证轨迹，避免运动轨迹存在干涉空间

2	设置起始偏置 (度)	可以对每个轴设置安全避障起始位置 当存在干涉空间时候可设置关节起始偏置避开干涉空间（需要注意正负方向）
3	设置幅值系数 (0-1.5)	定义辨识过程关节运动范围 取值范围：0.1（保守模式）至 1.5（最大行程） 建议首次使用采用 1 幅值验证轨迹，需确保工作空间无干涉
4	点击全参数辨识	手捏三段开关点击全参数辨识开始自动运行预设轨迹， 获取本体参数

14.3.2 摩擦辨识

本体辨识或摩擦力辨识结束后，摩擦力系数中的参数将更新为辨识后的参数。

如果需手动修改该参数，请在修改后，点击"保存"按钮进行保存。但是不建议用户自行修改，擅自修改很可能会造成动力学相关功能异常。同时可通过调整摩擦力系数来调整各个关节的拖动效率。

轴 号

1

摩擦力参数

摩擦力系数(%)

0.750

摩擦力死区(deg/s)

0.02000

抖动控制系数

0.00000

抖动控制翻转周期

1

摩擦力类型

一阶摩擦力模型

正向粘性摩擦力

5.49162

负向粘性摩擦力

5.51060

正向哥伦布摩擦力

1.99404

负向哥伦布摩擦力

2.76316

二阶粘性摩擦力

0.00000

摩擦力辨识

起始偏置(°)

0.00

频率(HZ)

0.500

幅值(°)

15.000

摩擦力辨识

保存

图表 70 摩擦力辨识参数界面

序号	参数名称	参数说明
1	摩擦力系数 (%)	摩擦力补偿的百分比补偿值=系数 × 理论值
2	摩擦力死区 (deg/s)	在死区范围内，摩擦力补偿不生效

3	抖动控制系数	死区内抖动控制的补偿值=系数 x 理论值
4	抖动控制翻转周期	抖动控制的周期=翻转周期 x 规划周期 x2
5	摩擦力类型	实际生效的摩擦力模型，不建议修改
6	正向粘性摩擦力	正运动方向动摩擦力系数
7	负向粘性摩擦力	负运动方向动摩擦力系数
8	正向哥伦布摩擦力	正运动方向静摩擦力系数
9	负向哥伦布摩擦力	正运动方向静摩擦力系数
10	二阶粘性摩擦力	摩擦力的加速度系数

序号	步骤	说明
1	设置频率 (HZ)	控制辨识过程运动速度 取值范围：0.1（低速模式）至 1（高速模式） 建议首次使用采用 0.5 频率验证轨迹，避免运动轨迹存在干涉空间
3	设置起始偏置 (°)	设置安全避障起始位置 当存在干涉空间时候可设置关节起始偏置避开干涉空间（需要注意正负方向）
4	设置幅值 (°)	定义辨识过程关节运动范围 取值范围：0°-360° 建议首次使用采用 20°幅值验证轨迹，需确保工作空间无干涉

5	空载辨识	手捏三段开关并选择需要摩擦力辨识的关节,点击运行辨识程序，等待辨识程序完成
6	计算并保存结果	计算完成出现辨识结果弹窗，点击确认并保存。

14.3.3 负载辨识

负载辨识功能可以方便地计算工具的动力学参数

负载质心

质量(kg)

0.00000

质心(m)

0.00000

0.00000

0.00000

负载惯量矩阵(kg*m²)

0.00000

0.00000

0.00000

0.00000

0.00000

0.00000

负载辨识参数

频率系数(0-1.5)

1.000

轴号

1

▼

起始偏置(度)

0.00

幅值系数(0-1.5)

1.000

不带负载辨识

带负载辨识

计算

保存

图表 71 负载辨识参数界面

序号	步骤	说明
1	设置频率系数 (0-1.5)	控制辨识过程运动速度 取值范围：0.1（低速模式）至 1.5（高速模式）

		建议首次使用采用 0.3 频率验证轨迹，避免运动轨迹存在干涉空间
2	轴号	可以对各个关节设置幅值和起始偏置
3	设置起始偏置 (度)	设置安全避障起始位置 当存在干涉空间时候可设置关节起始偏置避开干涉空间（需要注意正负方向）
4	设置幅值系数 (0-1.5)	定义辨识过程关节运动范围 取值范围：0.1（保守模式）至 1.5（最大行程） 建议首次使用采用 1 幅值验证轨迹，需确保工作空间无干涉
5	空载辨识	手捏三段开关在空载的状态下运行不带工具辨识程序，等待辨识程序完成
6	负载辨识	手捏三段开关装上工具负载，运行带工具辨识程序，等待辨识程序完成
7	计算并保存结果	计算完成出现辨识结果弹窗，点击确认并保存。



注意

- 请务必准确定义新工具的动力学参数，否则将影响机器人的运动性能，严重情况下甚至会造成机器人过载而损坏。
- 支持机器人正装和倒装情况下的负载辨识（倒装需要调整重力方向）。
- 在辨识过程中出现以下情况，将导致辨识过程终止，所有已获得的辨识数据丢失。

失，需要重新辨识：

1. 辨识操作进行到中间时，选择了其他工具或切换到其他界面；
2. 辨识程序运行过程中触发急停或者外部安全停止；
3. 辨识程序运行过程中，从自动模式转换到手动模式会导致辨识失败。



- 辨识程序需要在自动模式下执行，因此各项防护措施均应处于有效状态，外部控制信号可能随时启动机器人，因此请在完成安装且人员全部退到安全区域时再切换到自动模式执行。

14.4 零位标定

力传感器零位标定

1.当前力传感器数据

1:	2458.5	2:	2883	3:	1456	4:	2934.5
5:	2400	6:	2546.5	7:		8:	

2.当前零点文件数据

1	0.00	5	0.00
2	0.00	6	0.00
3	0.00	7	0.10
4	0.00	8	0.10

确认保存

手动修改

3.选择需要记录关节

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

☐ 6

☐ 7

☐ 8

清除记录

记录零点

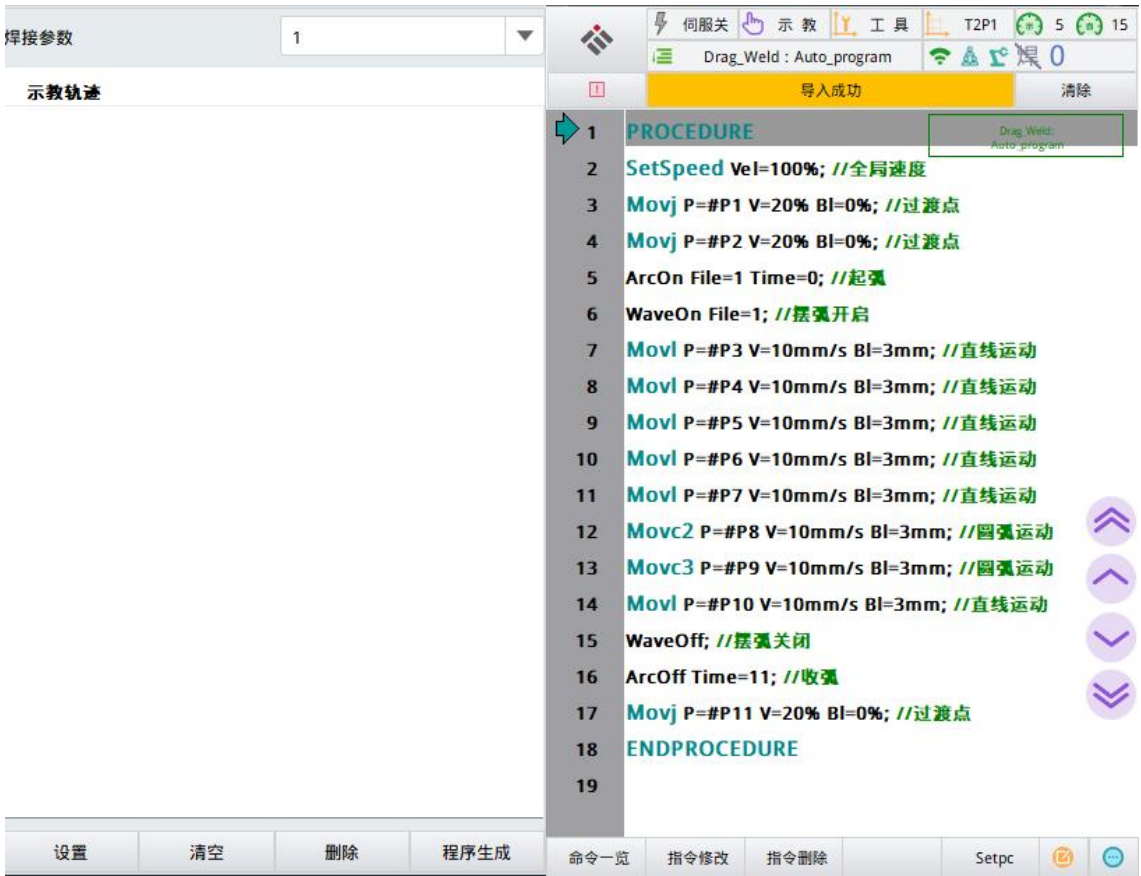
图表 72 零位标定界面

力传感器数据异常或需要重新记录零位时，需要对机器人的每个轴进行力传感器数据标定和软件记录标定。

机械零位标定：通过单轴运动，使每个轴都运行到机械参考零点。

软件记录标定：当 6 个轴都已经通过单轴运动回到机械参考零点之后，需要进入软件的出厂设置中，重新记录零点位置。确保软硬件的零点位置对应。之后每次机器人回零都会回到该位置。

14.5文件生成



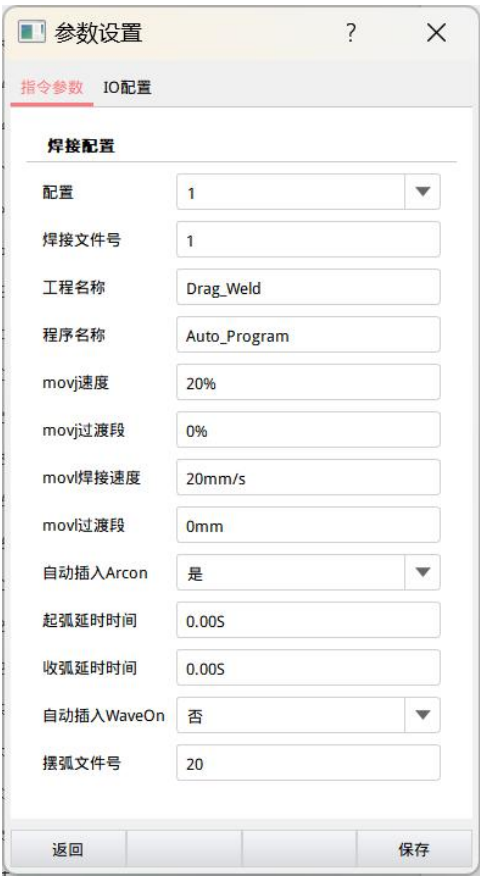
图表 73 文件生成界面

图表 74 文件界面

拖动模式记录点位之前建议提前配置好焊接参数，参数配置完后，再进行拖动示教点位。

拖动模式下通过力传感器（io 信号）记录点位信息，并选择合适的运动指令。
记录完毕后，点击【程序生成】，进入程序界面执行程序。

14.5.1 参数设置



图表 75 焊接参数配置界面

焊接参数配置里包含序号为 1-10 可配置页面，不同序号可设置不同焊接参数内容。配置内容包含：焊接文件号，工程名称，程序名称，movj 速度，movj 过渡段，movl 焊接速度，movl 过渡段，自动插入 Arcon，起弧延时时间，收弧延时时间，自动插入 WaveOn，摆弧文件号，修改完成后保存并可以在主界面调用焊接参数。具体焊接指令说明详见焊接操作说明书。



图表 76 拖动 IO 配置界面

模组：表示 IO 模块地址 ID，扩展 IO 从站地址 ID 取值从 1 开始。

点位：表示使用该模组的第几位，0 表示不使用，点位取值从 1 开始。

有效值：正常 1 为有效，可以取反。

序号	参数名称	有效值	说明
1	Insert_movj	0-输入	插入 movj 指令
2	Insert_movl	0-输入	插入 movl 指令
3	Insert_movc	0-输入	插入 movc 指令
4	delete_last	0-输入	删除最后一条指令

5	arc_on	0-输入	插入焊接开启指令
6	arc_off	0-输入	插入焊接关闭指令
7	wave_on	0-输入	插入摆弧开启指令
8	wave_off	0-输入	插入摆弧关闭指令
9	enable_drag	0-输入	使用本体拖动按钮
10	led_green	1-输出	打开绿色 LED 指示灯
11	led_red	1-输出	打开红色 LED 指示灯
12	led_blue	1-输出	打开蓝色 LED 指示灯

15 其他功能

15.1 零点标定



注意

- 1.没有进行原点位置校准，不能进行示教和回放操作。使用多台机器人的系统，每台机器人都必须进行原点位置校准。
- 2.当关节轴之间存在耦合关系时，例如常见的机器人第五轴和第六轴存在耦合关系，第五轴必须处于零点位置时，第六轴记录的零点数据才会有效，否则，第六轴记录的零点数据是无效的。所以必须在第五轴处于零位的状态下记录第六轴的零位数据。如果不存在耦合关系，则各个轴可以单独标定零位，各自的零位不会影响到其它关节的零位。
- 3.当所有用到的轴（本体轴和辅助扩展轴）都完成零位标定后，零位标定界面上的“全部”指示灯变为绿色，说明机器人已完成零位数据的标定，机器人可以进行笛卡尔空间下的运动。

在清除编码器零点漂移报警之后，需要立刻对机器人的每个轴进行机械零点位置标定和软件记录标定。

机械零位标定：按照下述方法进行，也就是通过单轴运动，使每个轴都运行到机械参考零点。

软件记录标定：当 6 个轴都已经通过单轴运动回到机械参考零点之后，需要进入软件的出厂设置中，重新记录零点位置。确保软件硬件的零点位置对应。之后每次机器人回零都会回到该位置。

伺服关

示教

关节

T0,P0

20 20

2 : 10

*****正常*****

清除

当前绝对值编码器数据

1轴:	5556230	2轴:	0	3轴:	0	4轴:	0
5轴:	0	6轴:	0	7轴:	0	8轴:	6326415
E1轴:	0	E2轴:	0	E3轴:	0	E4轴:	0
E5轴:	0	E6轴:	0	E7轴:	0	E8轴:	0

当前零位状态及零点文件数据

1

绿灯表示该轴零点标定有效

0

3

0

4

0

5

灰色灯表示该轴零点无效,需要重新标定

6

88

7

0

8

0

E1

0

E2

10

E3

11

E4

12

E5

13

E6

14

E7

99

E8

0

手动修改

刷新数据

选择需要记录或者清零的轴

☐ 1轴

☐ 2轴

☐ 3轴

☐ 4轴

☐ 5轴

☐ 6轴

☐ 7轴

☐ 8轴

☐ E1轴

☐ E2轴

☐ E3轴

☐ E4轴

☐ E5轴

☐ E6轴

☐ E7轴

☐ E8轴

绝对编码器清零

记录零点

图表 77 零点标定配置界面

项目	操作步骤
----	------

记录零点



当前绝对值编码器数据

1轴:	0	2轴:	0	3轴:	0	4轴:	0
5轴:	0	6轴:	0	7轴:	0	8轴:	0
E1轴:	0	E2轴:	0	E3轴:	0	E4轴:	0
E5轴:	0	E6轴:	0	E7轴:	0	E8轴:	0

当前零位状态及零点文件数据

1	887	2	0
3	0	4	0
5	0	6	88
7	0	8	0
E1	0	E2	10
E3	11	E4	12
E5	13	E6	14
E7	99	E8	0

1.将机器人对应轴运动到零位后，选择要标定的轴

2.点击记录零点，完成标定

3.标定完成后若状态灯为灰色则表示标定不成功，需要重新标定

选择需要记录或者清零的轴

☐ 1轴 ☒ 2轴 ☒ 3轴 ☒ 4轴 ☐ 5轴 ☐ 6轴

☐ 7轴 ☐ 8轴 ☐ E1轴 ☐ E2轴 ☐ E3轴 ☐ E4轴

☐ E5轴 ☐ E6轴 ☐ E7轴 ☐ E8轴

绝对编码器清零 记录零点

绝对编码器清零

(把选定轴的零

位数据修改为 0)



当前绝对值编码器数据

1轴:	0	2轴:	0	3轴:	0	4轴:	0
5轴:	0	6轴:	0	7轴:	0	8轴:	0
E1轴:	0	E2轴:	0	E3轴:	0	E4轴:	0
E5轴:	0	E6轴:	0	E7轴:	0	E8轴:	0

当前零位状态及零点文件数据

1	887	2	0
3	0	4	0
5	0	6	88
7	0	8	0
E1	0	E2	10
E3	11	E4	12
E5	13	E6	14
E7	99	E8	0

1.点击要清除零位的轴

2.点击清零按钮

选择需要记录或者清零的轴

☐ 1轴 ☒ 2轴 ☒ 3轴 ☒ 4轴 ☐ 5轴 ☐ 6轴

☐ 7轴 ☐ 8轴 ☐ E1轴 ☐ E2轴 ☐ E3轴 ☐ E4轴

☐ E5轴 ☐ E6轴 ☐ E7轴 ☐ E8轴

绝对编码器清零 记录零点

手动修改零点
(不推荐)

伺服关

示教

关节

TOP0

20 20

2 : 10

清除

*****正常*****

当前绝对编码器数据

1轴:	5556230	2轴:	0	3轴:	0	4轴:	0
5轴:	0	6轴:	0	7轴:	0	8轴:	6326415
E1轴:	0	E2轴:	0	E3轴:	0	E4轴:	0
E5轴:	0	E6轴:	0	E7轴:	0	E8轴:	0

当前零位状态及零点文件数据

1	887	2	0
3	0	4	0
5	0	6	88
7	0	8	0
E1	0	E2	10
E3	11	E4	12
E5	13	E6	14
E7	0	E8	0

1.点击手动修改

2.修改对应轴的零点数据

3.点击确认保存，修改完成

确认保存

手动修改

刷新数据

选择需要记录或者清零的轴

☐ 1轴

☐ 2轴

☐ 3轴

☐ 4轴

☐ 5轴

☐ 6轴

☐ 7轴

☐ 8轴

☐ E1轴

☐ E2轴

☐ E3轴

☐ E4轴

☐ E5轴

☐ E6轴

☐ E7轴

☐ E8轴

绝对编码器清零

记录零点

15.220 点法

20 点法用于补偿零点和工具坐标系。

记录点位

1
记录点

2
计算

3
补偿

第一步：记录点位(V3.0)

记录点状态：绿灯-已记录，灰灯-未记录
相邻点各关节角度变化应尽量大

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

3.检查记录是否成功

14.记录完成，点位后点击下一个点位记录

点位序号: < 1 >

J1: -14.5892 J2: 5.7312

J3: -110.5506 J4: 31.6207

J5: 62.0428 J6: -27.5768

4.记录完成后可通过运动到点检查到位精度

1.对准尖端后点击记录点位

清除当前

清除全部

运动到点

记录

	机器人不同姿态用机器人工具末端去对尖端并记录位置点，各位置点关节角度变化尽可能大一些
计算结果	1 记录点 2 计算 3 补偿 第二步：计算 输入耦合比，点【计算】获取结果 5,6轴 0.00000 4,5轴 0.00000 4,6轴 0.00000 2,3轴 0.00000 1.关节零位偏置(°) 存在耦合比时输入耦合比没有可不填 J1: 0.0000 J2: 1.0768 J3: 0.2607 J4: 0.0178 J5: -0.3482 J6: 0.0000 2.TCP(mm) X: 119.3136 Y: -1.0771 Z: 414.5863 平均误差越小越好 3.误差 平均误差(mm): 0.6866 最大误差(mm): 1.4081 计算
选择补偿位置 (手动补偿)	1 记录点 2 计算 3 补偿 第三步：补偿 注意： 1.补偿会修改零位,补偿后,之前记录的点位已经失效; 2.若要重复使用之前的点位,请先手动将零位恢复到补偿前的原始值; 3.若要使用补偿前记录的点进行计算,请先手动将零位恢复到补偿前的原始值。 选择补偿工具坐标系序号(2-32)： 32 手动补偿

15.3异常处理



图表 78 异常配置界面

序号	参数名称	参数说明
1	当前轴仿真状态	将机器人对应轴设置为仿真状态，绿灯表示生效，灰色表示正常。
2	轴运动限制	对机器人轴运动限位做限制，绿灯表示生效，灰色表示正常。
3	取消碰撞报警	取消碰撞报警，绿灯表示生效，灰色表示正常。
4	取消外部IO急停检查	机器人外部IO急停触发时，对机器人外部急停IO做屏蔽10s，若外部急停IO未恢复则会继续报警，绿灯表示生效，灰色表示正常。

15.4权限管理

15.4.1 用户切换

登录：出厂设置登录成功


出厂设置


管理模式


操作模式

密码：

.....

登录

密码修改

☐ 新密码：出厂设置

☐ 新密码：管理模式

确认

清空

开机配置

☒ 开机出厂设置

☐ 开机管理模式

☐ 开机操作模式

确认

图表 79 用户换切换界面

15.4.2 硬件测试



图表 80 系统版本界面

序号	参数名称	参数说明
1	Ping 测试	检测示教器与控制器通讯是否正常
2	触摸标定	对手持触摸屏进行校准, 校准程序执行完后手持会自动重启。
3	背光测试	调节手持 lcd 背光亮度 自动屏保（打开/关闭）
4	手持硬件测试	{打开硬件测试}
5	修改控制器 IP	修改 lan1,lan2（lan3）网口 IP
6	查看控制器 IP	查看 lan1,lan2（lan3）网口 IP

15.4.3 界面风格

界面风格

默认

style 1

style 2

字号：

16

语言(重启生效)：

中文

确认

取消

图表 81 界面风格设置界面

设置界面字体、字号和显示语言。

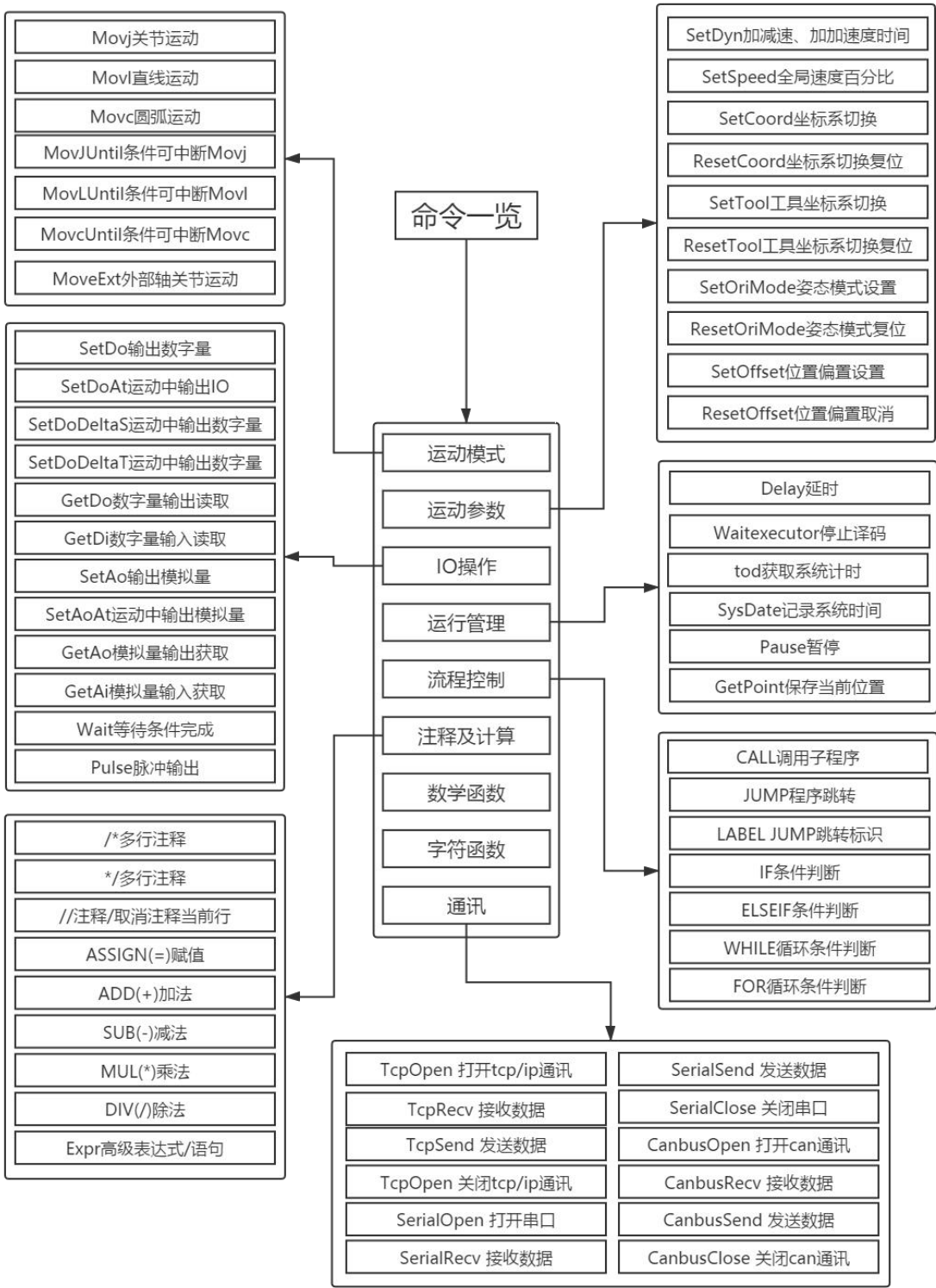
15.4.4 升级与备份

项目	操作步骤
系统升级/导入示教文件	用户切换 硬件测试 界面风格 升级与备份 备份与恢复 1. 界面版本1.14.14 - 2.0.7 2. 软件更新(更新后需重启) 手持盒更新 logo替换 控制器更新 消息提示： Task v4.15.7_20231106 ECAT_20231103,Master_20231024,Net_202308120601037 libmotion_r_v3.4.35 softPlc v0.6.5 imf_communication2.3.4_20230714 imf_data_server1.0.4 imf_remote_modbus2.0.1_20231012 imf_remote_tcp2.1.4_20230825 Import0.3.0 Task v4.15.7_20231106 ECAT_20231103,Master_20231024,Net_202308120601037 3. 文件备份 ☒ 机器人零点 ☒ 后台任务 ☒ 示教工程 ☒ yyy 注意：若备份内容较多时，请将u盘插到控制器，从控制器上的usb口备份，否则花费时间可能较久。 手持USB口 刷新列表 备份选中文件 备份全部 点击控制器更新

界面升级	手持盒界面 UI 升级需要在“升级与备份”界面点击“手持盒更新”，插入 U 盘后选择相对应的升级文件，升级完成后切换“硬件测试”界面点击“打开硬件测试”，再选择“返回机器人界面”即可；
备份文件	1. 点击“备份全部”备份全部文件； 2. 勾选需要备份的文件，点击“备份选中文件”备份选中文件

16 指令系统

16.1 命令一览



图表 82 命令一览

16.2指令规范

表示规则：

- [] :表示中括号里的内容出现 0 或者 1 次，表示可选
- | :表示“或”，两者选其一
- = :表示输入赋值
- => :表示输出赋值
- #表示数值，Var 表示变量，Str 表示字符串
- *浅蓝倾斜字体表示参数设置时需要填入或者选择的数值或变量*

16.3 运动指令

16.3.1 运动指令的处理机制

16.3.1.1 运动指令预处理

机器人并不是执行一行运动指令，机器人移动一个位置。为了提高运动的流畅性，控制系统示教程序解析模块会提前将所有指令编译，并进行预处理，然后根据程序本身逻辑向机器人的运动堆栈中发送若干运动指令，此时可能机器人连第一个位置都没有到达。要理解应用程序的执行与机器人运动是相互独立。同时，系统提供暂停解析指令，来方便用户根据实际情况中断程序的提前编译模式。

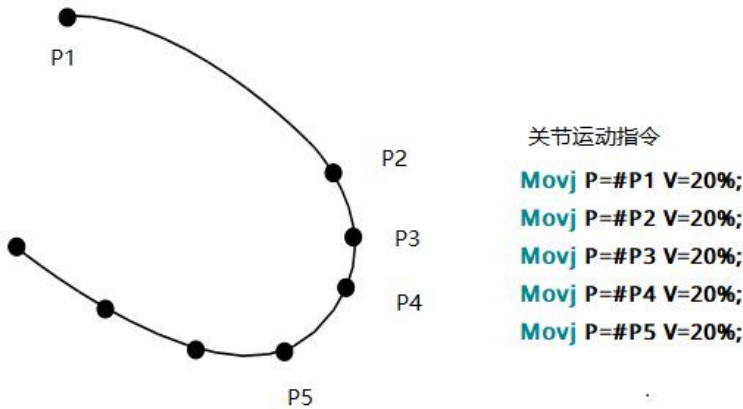
16.3.1.2 运动指令轨迹过渡

两条运动指令的衔接处，因为前后两条移动指令的速度变化不是光滑的，即上一段轨迹速度必须减为 0，才能继续下一段轨迹。这样就毫无必要地增加了周期时间。通过设置过渡段，可大大减少运动时间。当此 Blend 值设置为“不使用”时，机器人在轨迹的每一个点都严格到达。不过当 Blend 值设置为#Var 或%时，轨迹将在每一点附近混合，机器人不严格到达这些点。

16.3.2 Movj 关节运动

Movj 运动轨迹及示例:

Movj 指令用来编制点到点的运动（关节运动）。为了优化运动速度，在起点和终点之间，工具中心沿着一条有系统自定义的**曲线**运动。点与点之间位置通过关节插补。



图表 83 Movj 运动轨迹及指令

Movj 关节运动			
1.目标点	#P	▼	1
2.速度	%	▼	20.00
3.过渡段	不使用	▼	
4.轴组	☐ A	☐ B	
	☐ C	☐ D	

图表 84 Movj 参数设置

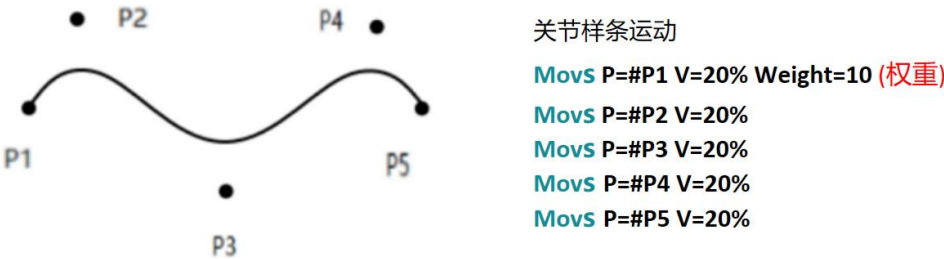
指令格式	Movj P= <i>Point</i> V= <i>Speed</i> % [Bl= <i>Blend</i> %] [Ext= [<i>A-D</i>]]	
示例	Movj P=#P1 V=20% Bl=20% Ext=A;	
参数说明	目标点	<i>Point</i> :可以是关节空间也可以是笛卡尔空间点
	速度	<i>Speed</i> : 关节空间末端合成速度，速度= % * 关节空间末端合成最大速度。取值为 1 至 100，默认值为 20。
	过渡段	<i>Blend</i> : 关节空间过渡半径，过渡半径= % * 关节空间轨迹总长度。
	轴组	Ext 需要同步运动的外部轴组号，可多选。对于 Movj 只存在机器人和外部轴的同起同停的同步运动规划，不存在严格的位置跟随规划。外部轴的速度为机器人运动

		速度和外部轴允许最大速度两者的较小值，不需要单独设定外部轴自己的速度。
注意事项		

16.3.3 Movs 关节样条运动

Movs 运动轨迹及示例:

Movs 指令在关节空间做 b 样条运动。



图表 85 Movs 运动轨迹及指令

Movs 关节样条运动		
1.目标点	#P	1
2.速度	%	20.00
3.权重	1.0	
4.Type	不使用	
5.过渡段	不使用	

图表 86 Movs 参数设置

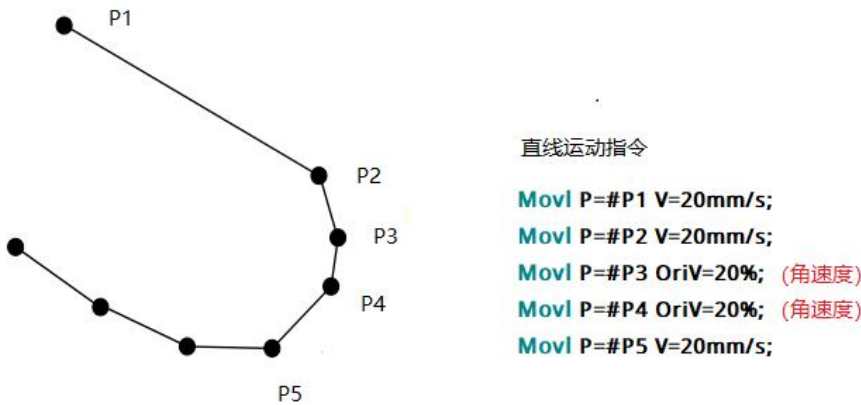
指令格式	Movs P= <i>Point</i> V= <i>Speed</i> % [Weight= [<i>1-10</i>]][Type= <i>end</i> ,Bl= <i>Blend</i> %]	
示例	Movs P=#P1 V=20% Weight=3 Type=end Bl=20%;	
参数说明	目标点	<i>Point</i> : 可以是关节空间也可以是笛卡尔空间点
	速度	<i>Speed</i> : 关节空间末端合成速度，速度= % * 关节空间

		末端合成最大速度。取值为 1 至 100，默认值为 20。
	权重	<i>Weight</i> : 到点精度，取值越高表示到点精度越高，取值为 1 至 10，默认值为 1。
	Type	End 取消样条运动控制点，变成到达目标点
	过渡段	<i>Blend</i> : 关节空间过渡半径，过渡半径= % * 关节空间轨迹总长度。
注意事项	Type 选择 end 之后，Movs 运动会与 Movj 运动相同	

16.3.4 Movl 直线运动

Movl 运动轨迹及示例

笛卡尔空间直线运动，包含姿态跟随和规划。



图表 87 Movl 运动轨迹及指令

Movl 直线运动

1.目标点

#P

▼

5

2.速度

●

线速度(Speed)

○

角速度(OriSpeed)

mm/s

▼

100.00

3.过渡段

mm

▼

0

4.轴组

☐A

☐B

☐C

☐D

5.OriMode

Standard

▼

6.OriDir

Min

▼

图表 88 Movl 参数设置

指令格式	Movl P=<i>Point</i> (V=<i>Speed</i> mm/s [Bl=<i>Blend</i> mm]) (OriV = <i>OriSpeed</i> % [OriBl =% <i>OriBlend</i>]) [Ext= [<i>A-D</i>]]	
示例	Movl P=#P1 V=20mm/s Bl=10mm Ext=A; Movl P=#P1 OriV=20% OriBl=10% Ext=A;	
参数说明	目标点	<i>Point</i> :可以是关节空间也可以是笛卡尔空间点。
	速度	<i>Speed</i> : 笛卡尔空间末端路径合成速度，单位：mm/s <i>OriSpeed</i> : 笛卡尔空间末端姿态合成速度百分比，速度 = % * 姿态末端合成最大速度。
	过渡段	<i>Blend</i> : 笛卡尔空间过渡半径，单位：mm; <i>OriBlend</i> : 笛卡尔空间末端姿态过渡百分比，过渡角度 = % * 姿态空间轨迹总角度。
	轴组	需要同步运动的外部轴组号，可多选。同步运动是否为位置跟随取决于位置点记录的对应 PCS 是否是随着对应外部轴一起运动。
	OriMode	Standard,PathFrame,JointSpace
	OirDir	姿态规划，默认 Min,根据需求给为 Max

注意事项	1.设置 Speed 和 Blend 后，路径规划，姿态进行跟随。 2.设置 OriSpeed 和 OriBlend 后，姿态进行规划，路径跟随。
------	--

16.3.5 Movp 笛卡尔样条运动

Movp 运动轨迹及示例

笛卡尔空间样条运动，包含位置权重，姿态跟随和规划。



图表 89 Movp 运动轨迹及指令

Movp 笛卡尔样条运动		
1.目标点	#P	1
2.速度	mm	20.00
3.位置权重	1.0	
4.姿态权重	1.0	
5.Type	不使用	

图表 90 Movp 参数设置

指令格式	Movp P= <i>Point</i> (V= <i>Speed</i> mm/s [Bl= <i>Blend</i> mm]) (PosWeight= <i>PosBlend</i> % [OriWeight=% <i>OriBlend</i>])
示例	Movp P=#P1 V=20mm/s PosWeight=2 OriWeight=2

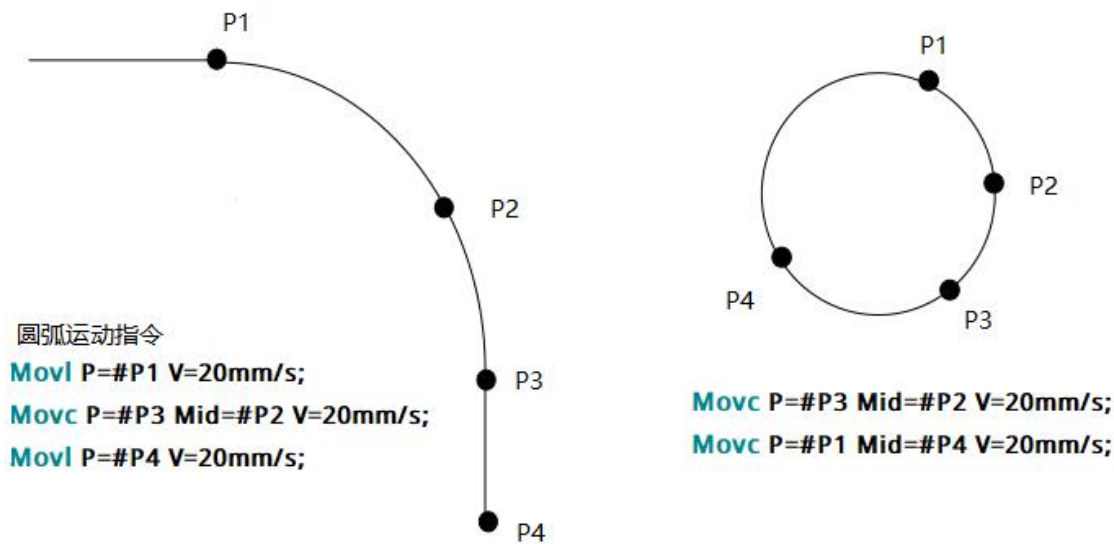
	Movp P=#P1 V=20mm/s PosWeight=2 Type=End Bl=10mm	
参数说明	目标点	<i>Point</i> :可以是关节空间也可以是笛卡尔空间点。
	速度	<i>Speed</i> : 笛卡尔空间末端路径合成速度，单位： mm/s
	过渡段	<i>Blend</i> : 笛卡尔空间过渡半径，单位： mm；
	位置权重	<i>PosWeight</i> : 笛卡尔空间末端位置到点精度，取值越高表示到点精度越高，取值为 1 至 10，默认值为 1。
	姿态权重	<i>OriWeight</i> : 笛卡尔空间末端姿态到点精度，取值越高表示到点精度越高，取值为 1 至 10，默认值为 1。
	Type	End 取消样条运动控制点，变成到达目标点
	过渡段	<i>Blend</i> : 笛卡尔空间过渡半径，单位： mm；
注意事项	Type 选择 end 之后， Movp 运动会与 Movl 运动相同	

16.3.6 Movc 圆弧运动

16.3.6.1 Movc 圆弧运动

Movc 运动轨迹及示例

笛卡尔空间圆弧运动，包含姿态跟随。在圆周运动中，工具中心点沿由 3 点定义的弧线运动，在起点、中点和终点之间采用圆弧插补。



图表 91 Movc 运动轨迹及指令

圆弧指令需要 2 个位置点，分别代表圆弧的中间点与结束点，而机器人的当前位置或上一个动作的结束位置为圆弧起点。一个完整的圆需要 4 个位置点，通过 2 个 Movc 指令实现。

Movc 圆弧运动			
1.目标点	#P	▼	1
中间点	#P	▼	1
2.速度	mm/s	▼	20.00
3.过渡段	不使用	▼	
4.圆弧角度	自动	▼	
5.轴组	☐ A ☐ B ☐ C ☐ D		
6.OriMode	Standard	▼	
7.OriDir	Min	▼	

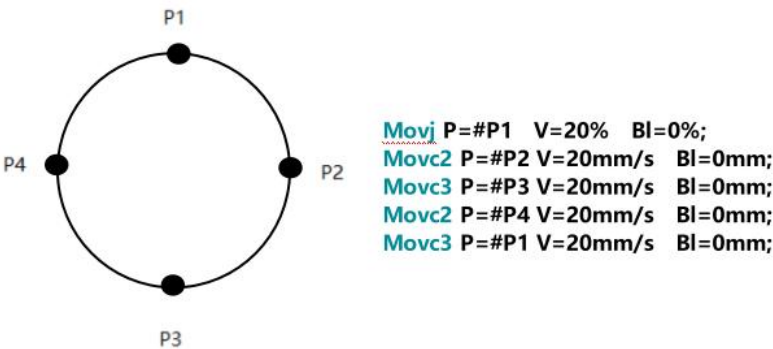
图表 92 Movc 参数设置

指令格式	Movc P= <i>EndPoint</i> Mid= <i>MidPoint</i> V= <i>Speed</i> mm/s [Bl = <i>Blend</i> mm] [Deg = <i>Degree</i>] [Ext = [<i>A-D</i>]]	
示例	Movc P=#P1 Mid=#P2 V=20mm/s Bl=10mm Deg=108 Ext=A;	
参数说明	目标点	<i>EndPoint</i> : 圆弧末端点，可以是关节空间也可以是笛卡尔空间点；

	中间点	<i>MidPoint</i> : 圆弧中间点，可以是关节空间也可以是笛卡尔空间点。
	速度	<i>Speed</i> : 笛卡尔空间末端路径合成速度，单位：mm/s。
	过渡段	<i>Blend</i> : 笛卡尔空间过渡半径，单位：mm。
	圆弧角度	<i>Degree</i> : 实际运动的圆弧角度，譬如一个整圆，默认是三点构成的圆弧角度，支持负数，可选。
	轴组	Ext: 需要同步运动的外部轴组号，可多选，同 MovL。
	OriMode	Standard,PathFrame,JointSpace
	OirDir	姿态规划，默认 Min,根据需求给为 Max
注意事项	圆弧轨迹，姿态规划，路径跟随的情况不合理，不支持。 圆弧起始点姿态绕法线旋转运动，无法运动到终点的姿态。	

16.3.6.2 Movc2 Movc3 组合圆弧运动

Movc2 Movc3 运动轨迹及示例



图表 93 Movc2 Movc3 运动轨迹及指令

Movc2 Movc3 指令做为圆弧的两个位置点，分别代表圆弧的中间点与结束点

Movc2 圆弧中点			Movc3 圆弧端点		
1.目标点	#P	6	1.目标点	#P	7
2.速度	mm/s	20.00	2.速度	mm/s	20.00
3.过渡段	mm	0	3.过渡段	mm	0
4.圆弧角度	自动		4.圆弧角度	自动	
5.轴组	☐ A ☐ B ☐ C ☐ D		5.轴组	☐ A ☐ B ☐ C ☐ D	
6.OriMode	Standard		6.OriMode	Standard	
7.OriDir	Min		7.OriDir	Min	
单行注释			单行注释		

图表 94 Movc2 Movc3 参数设置

指令格式	Movc2 P=<i>Point</i> V=<i>Speed</i> mm/s [Bl = <i>Blend</i> mm] [Deg = <i>Degree</i>] [Ext = [<i>A-D</i>]] Movc3 P=<i>Point</i> V=<i>Speed</i> mm/s [Bl = <i>Blend</i> mm] [Deg = <i>Degree</i>] [Ext = [<i>A-D</i>]]	
示例	Movc2 P=#P1 V=20mm/s Bl=10mm Deg=108 Ext=A; Movc3 P=#P2 V=20mm/s Bl=10mm Deg=108 Ext=A;	
参数说明	目标点	<i>Point</i> : 目标点，可以是关节空间也可以是笛卡尔空间点；
	速度	<i>Speed</i> : 笛卡尔空间末端路径合成速度，单位：mm/s。
	过渡段	<i>Blend</i> : 笛卡尔空间过渡半径，单位：mm。
	圆弧角度	<i>Degree</i> : 实际运动的圆弧角度，譬如一个整圆，默认是三点构成的圆弧角度，支持负数，可选。
	轴组	Ext: 需要同步运动的外部轴组号，可多选，同 MovL。
	OriMode	Standard,PathFrame,JointSpace
	OirDir	姿态规划，默认 Min,根据需求给为 Max

注意事项	圆弧轨迹，姿态规划，路径跟随的情况不合理，不支持。 圆弧起始点姿态绕法线旋转运动，无法运动到终点的姿态。 Movc2 Movc3 的 <i>Speed</i> 和 <i>Blend</i> 仅在 movc2 修改生效, movc3 修改参数无意义。
------	---

16.3.7 MoveExt 外部轴组关节运动

MoveExt 外部轴组关节运动，机器人会根据设置参数选择是否跟随运动。

MoveExt 外部轴组关节运动

1.目标点

#P

1

2.速度

%

20.00

3.过渡段

%

0

4.轴组

☐ A

☐ B

☐ C

☐ D

图表 95 MoveExt 参数设置

指令格式	MoveExt P= <i>Point</i> V= <i>Speed</i> % [Bl = <i>Blend</i> %] [Ext = [<i>A-D</i>]]	
示例	MoveExt P=#P1 V=20% Ext=A	
参数说明	目标点	<i>Point</i> : 目标点，仅仅使用外部轴相关的部分。
	速度	<i>Speed</i> : 关节空间末端合成速度百分比，速度 = % * 关节空间末端合成最大速度。
	过渡段	<i>Blend</i> : 关节空间过渡半径百分比，过渡半径 = % * 关节空间轨迹总长度。
	轴组	Ext: 需要同步运动的外部轴组号，取值为 A,B,C,D。如果未选取 Ext，表示所有外部轴一起运动。机器人同步运动是否为位置跟随外部轴取决于位置点记录的 PCS 是否是随着外部轴一起运动。

注意事项	最多支持 4 个外部轴组同时运动
------	------------------

16.3.8 MovjUntil 可条件中断 Movj

关节运动,运动过程中自定义布尔表达式若变为 True, 停止当前指令或直接执行下一条指令；否则运动到位。可以锁存触发时的位置，包括目标坐标系位置和关节位置。该类型指令**不支持过渡参数的设置,不支持同步运动**。

MovjUntil 可条件中断关节运动			
1.目标点	#P	▼	1
2.速度	%	▼	20
3.中断条件	0		
4.输出变量	#Var	▼	不使用

图表 96 MovjUntil 参数设置

指令格式	MovjUntil P= <i>Point</i> V= <i>Speed</i> % Cond= <i>Condition?</i> =>ReachEnd	
示例	MovjUntil P=#P1 V=20% Cond=(#Var1<<5) =>#Var10;	
参数说明	目标点	<i>Point</i> : 目标点，可以是关节空间也可以是笛卡尔空间点。支持距离目标点距离条件，（距离是基于运动的目标坐标系，未叠加摆动）
	速度	<i>Speed</i> : 关节空间末端合成速度，单位：degree/s。
	中断条件	<i>Condition?</i> ：自定义布尔表达式。例如#Var1<<5。
	输出变量	ReachEnd：若 Condition 为 False，则到达目标点，ReachEnd =True; 若到达目标点前，Condition 为 True，则表示中途离开或者停止，没有到达目标点，则 ReachEnd=False。 ReachEnd 是一个 BoolVar。
注意事项		

16.3.9 MovlUntil 可条件中断 Movl

MovlUntil 笛卡尔空间直线运动,运动过程中布尔表达式若变为 True,停止当前指令或直接执行下一条指令；否则运动到位。可以锁存触发时的位置，包括目标坐标系位置和关节位置。该类型指令不支持过渡参数的设置,不支持同步运动。

MovlUntil 可条件中断直线运动

1.目标点

#P

▼

1

2.速度

☒ 线速度(Speed)

☐ 角速度(OriSpeed)

mm/s

▼

20

3.中断条件

#Var1<<5

4.输出变量

#Var

▼

10

图表 97 MovlUntil 参数设置

指令格式	MovlUntil P= <i>Point</i> (V= <i>Speed</i> mm/s) (Ori= <i>OriSpeed</i> %) Cond= <i>Condition?</i> =>ReachEnd	
示例	MovlUntil P=#P1 V=20mm/s Cond=(#Var1<<5) =>#Var10; MovlUntil P=#P1 OriV=20% Cond=(#Var1<<5) =>#Var10;	
参数说明	目标点	<i>Point</i> : 目标点，可以是关节空间也可以是笛卡尔空间点。支持距离目标点距离条件，（距离是基于运动的目标坐标系，未叠加摆动）
	速度	<i>Speed</i> : 笛卡尔空间末端路径合成速度，单位：mm/s; <i>OriSpeed</i> : 笛卡尔空间末端姿态合成速度百分比，速度 = % * 姿态末端合成最大速度。
	中断条件	<i>Condition?</i> ：自定义布尔表达式。例如#Var1<<5。
	输出变量	ReachEnd:若 Condition 为 False，则到达目标点，

		ReachEnd =True; 若到达目标点前，Condition 为 True，则表示中途离开或者停止，没有到达目标点，则 ReachEnd=False。ReachEnd 是一个 BoolVar。
注意事项	<i>Speed</i> 和 <i>OriSpeed</i> 二选一，前者表示路径规划，姿态跟随；后者是姿态规划，路径跟随。	

16.3.10 MovcUntil 可条件中断 Movc

MovcUntil 笛卡尔空间圆弧运动，运动过程中布尔表达式若变为 True,停止当前指令或直接执行下一条指令；否则运动到位。可以锁存触发时的位置，包括目标坐标系位置和关节位置。该类型指令不支持过渡参数的设置,不支持同步运动。

MovcUntil 可条件中止圆弧运动			
1.目标点	#P	▼	1
中间点	#P	▼	1
2.速度	mm/s	▼	20
3.圆弧角度	自动	▼	
4.中断条件	0		
5.输出变量	#Var	▼	不使用

图表 98 MovcUntil 参数设置

指令格式	MovcUntil P= <i>EndPoint</i> Mid= <i>MidPoint</i> V= <i>Speed</i> mm/s [Deg = <i>Degree</i>] Cond= <i>Condition?</i> => ReachEnd	
示例	MovcUntil P=#P1 Mid=#P2 V=20mm/s Deg=108 Cond=(#Var1<<5) =>#Var10;	
参数说明	目标点	<i>MidPoint</i> : 圆弧中间点，可以是关节空间也可以是笛卡尔空间点。

	中间点	<i>EndPoint</i> : 圆弧末端点, 可以是关节空间也可以是笛卡尔空间点; 支持距离目标点距离条件, (距离是基于运动的目标坐标系, 未叠加摆动)
	速度	<i>Speed</i> : 笛卡尔空间末端路径合成速度, 单位: mm/s。
	圆弧角度	<i>Degree</i> : 实际运动的圆弧角度, 譬如一个整圆, 默认是三点构成的圆弧角度, 支持负数, 可选。
	中断条件	<i>Condition?</i> : 自定义布尔表达式。例如#Var1<<5。
	输出变量	ReachEnd:若 Condition 为 False, 则到达目标点, ReachEnd =True; 若到达目标点前, Condition 为 True, 则表示中途离开或者停止, 没有到达目标点, 则 ReachEnd=False。 ReachEnd 是一个 BoolVar。
注意事项		

16.3.11 MoveExtUntil 可条件中断 MoveExt

MoveExtUntil 外部轴组关节运动, 运动过程中布尔表达式若变为 True,停止当前指令或直接执行下一条指令; 否则运动到位。可以锁存触发时的位置, 包括目标坐标系位置和关节位置。该类型指令**不支持过渡参数的设置,不支持同步运动**。

MoveExtUntil 可条件中止外部轴组关节运动

1.目标点

#P

1

2.速度

%

20.00

3.过渡段

%

0

4.轴组

☐ A

☐ B

☐ C

☐ D

5.中断条件

0

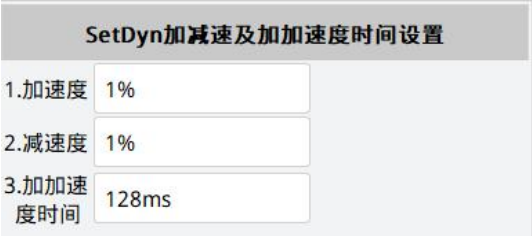
图表 99 MoveExtUntil 参数设置

指令格式	MoveExtUntilMoveExt P= <i>Point</i> V= <i>Speed</i> % [Bl = <i>Blend</i> %] [Ext = <i>A-D</i>] Cond= <i>Condition?</i> =>ReachEnd	
示例	MoveExtUntil P=#P1 V=20% Ext=A Cond=(#Var1<<5) =>#Var10;	
参数说明	目标点	<i>Point</i> : 目标点，可以是关节空间也可以是笛卡尔空间点。支持距离目标点距离条件，（距离是基于运动的目标坐标系，未叠加摆动）
	速度	<i>Speed</i> : 关节空间末端合成速度，单位：degree/s。
	轴组	Ext: 需要同步运动的外部轴组号，取值为 A,B,C,D。如果未选取 Ext，表示所有外部轴一起运动。机器人同步运动是否为位置跟随外部轴取决于位置点记录的 PCS 是否是随着外部轴一起运动。
	中断条件	<i>Condition?</i> : 自定义布尔表达式。例如#Var1<<5。
注意事项	最多支持 4 个外部轴组同时运动	

16.4运动参数

16.4.1 SetDyn 加减速、加加速度时间

加减速速度倍率设置及加加速度时间设置。



图表 100 SetDyn 设置界面

指令格式	SetDyn Acc = #% Dec = #% JerkTime = #	
示例	SetDyn Acc=1% Dec=1% JerkTime=128ms;	
参数说明	加速度	Acc: 全局加速度百分比，接下来所有指令适用。
	减速度	Dec: 全局减速度百分比，接下来所有指令适用。
	加 加 速 度	JerkTime: 加加速度的时间，单位：ms，范围为 32~512
	时 间	毫秒，默认值为 128。
注意事项		

16.4.2 SetSpeed 全局速度百分比

调整本条语句后面的运动指令的速度百分比。



图表 101 SetSpeed 参数设置

指令格式	SetSpeed Vel= #%
示例	SetSpeed Vel=20%;

参数说明	全局速度	全局速度百分比。
注意事项	如果不调用 SetSpeed 指令则程序默认值为 20%。	

16.4.3 SetCoord 坐标系切换

坐标系切换，适用于接下来所有运动指令中，记录在笛卡尔空间下的坐标点，直到遇到下一条 SetCoord 指令，或者遇到 ResetCoord 取消切换。



图表 102 SetCoord 设置

指令格式	SetCoord Type= <i>Type</i> Idx=#	
示例	SetCoord Type=PCS Idx=1;	
参数说明	坐标系	<i>Type</i> :坐标系类型，可选 KCS,PCS, WCS; Idx: 坐标系序号。
注意事项	该指令对 SetCoord 到 ResetCoord 之间的位置点生效。	

16.4.4 ResetCoord 坐标系切换复位

坐标系停止切换，恢复到指令本身记录的坐标系，参考 SetCoord 指令。

16.4.5 SetTool 工具坐标系切换

工具坐标系切换，适用于接下来所有运动指令，直到遇到下一条 SetTool 或者 ResetTool 取消切换。

SetTool工具坐标系切换

1.序号 1

图表 103 SetTool 设置

指令格式	SetTool Index=#	
示例	SetTool Index=1	
参数说明	序号	Index: 工具坐标系序号。
注意事项	该指令对 SetTool 到 ResetTool 之间的位置点生效。	

16.4.6 ResetTool 工具坐标系切换复位

工具停止切换，恢复到指令本身记录的工具号，参考 SetTool 指令。

16.4.7 SetOriMode 姿态模式设置

设置姿态规划模式，对接来下所有笛卡尔空间运动指令生效，直到遇到下一条 SetOriMode 或者 ResetOriMode。

SetOffset 位置点偏置

1.类型

Joint

▼

NULL

▼

2.偏置值

J1: 1.00

J2: 1.00

J3: 1.00

J4: 1.00

J5: 1.00

J6: 1.00

J7: 1.00

J8: 1.00

图表 104 SetOriMode 设置

指令格式	SetOriMode Mode= <i>Standard Unchange Rotate Normal</i>	
示例	SetOriMode Mode=Standard;	
参数说明	模式	Mode: 可选 <i>Standard, Unchange, Rotate, Normal</i> 。

注意事项	该指令对 SetOriMode 到 ResetOriMode 之间的位置点生效。
------	--

16.4.8 ResetOriMode 姿态模式复位

恢复姿态标准跟随模式 Standard。

16.4.9 SetOffset 位置偏置设置

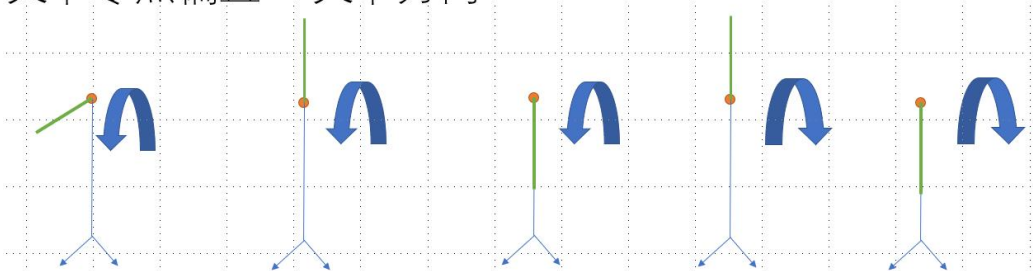
设置关节空间位置点、笛卡尔位置点、外部轴位置点偏置。

SetOffset 关节位置点偏置设置

1.类型	joint	
2.偏置值	J1: 0.00	J2: 0.00
	J3: 0.00	J4: 0.00
	J5: 0.00	J6: 0.00
	J7: 0.00	J8: 0.00

图表 105 SetOffset 参数设置

指令	SetOffset Type = “ <i>Joint</i> ” “ <i>Pose</i> ”(J1 =# J2 =# J3 =# J4 =# J5 =# J6 =# J7 =#	
格式	J8 =#) (X = #Y = #Z = # [Roll = # Pitch = #Yaw = #])	
示例	SetOffset Type=Joint J1=5.00 J2=10.00 J3=0.00 J4=0.00 J5=0.00 J6=0.00 J7=0.00 J8=0.00; SetOffset Type=Pose X=5.00 Y=10.00 Z=0.00 Roll=0.00 Pitch=0.00 Yaw=0.00; SetOffset Type=Pose X=0 Y=0 Z=0 Roll=10 Pitch=10 Yaw=10 SetOffset Type=Pose X=#Gvar1 Y=#Gvar2 Z=#Gvar3 Roll=#Gvar4 Pitch=#Gvar5 Yaw=#Gvar6	
参数	类型	Type: 偏置类型，机器人关节偏置，笛卡尔位置点偏置

说明		NULL：输入类型，支持直接输入点位或变量输入修改点位				
		1. J1,J2,J3,J4,J5,J6,J7,J8 关节位置偏置或者外部轴偏置； 2. X, Y, Z：位置偏移量； 3.Roll, Pitch, Yaw：姿态偏移量。				
偏置值		偏置数值				
注意 事项	关节零点偏置 + 关节方向					
	<table><tr><td>方向：标准+1 零点：水平理论位置</td><td>方向：标准+1 零点：-90度</td><td>方向：标准+1 零点：+90度</td><td>方向：反向-1 零点：-90度</td><td>方向：反向-1 零点：+90度</td></tr></table> 关节方向和零点偏置相对于标准情况 编码器值 → 理论关节角度 → 显示关节角度 = (理论关节角度 - 零点偏置) * 关节方向 显示关节角度 → 理论关节角度 = 显示关节角度 * 关节方向 + 零点偏置 正向运动学输入的是理论关节角度，逆向运动学输出的是理论关节角度 编码器获取的数据，计算当前角度也是理论关节角度 显示关节角度用于界面显示、示教位置		方向：标准+1 零点：水平理论位置	方向：标准+1 零点：-90度	方向：标准+1 零点：+90度	方向：反向-1 零点：-90度
方向：标准+1 零点：水平理论位置	方向：标准+1 零点：-90度	方向：标准+1 零点：+90度	方向：反向-1 零点：-90度	方向：反向-1 零点：+90度		
该指令对 SetOffset 到 ResetOffset 之间的位置点生效。						

16.4.10 ResetOffset 位置偏置取消

关节位置点偏置不再生效，与 SetOffset 配对使用。

16.4.11 SetBlendType 设置过渡段类型



图表 106 SetBlendType 过渡段类型设置

指令格式	SetBlendType Type= <i>Fusion Spline VelBlend</i>	
示例	SetBlendType Type= <i>Fusion</i> ;	
参数说明	模式	Mode: 可选 <i>Fusion</i> , <i>Spline</i> , <i>VelBlend</i> 。
注意事项		

16.4.12 ResetBlendType 设置过渡段类型

设置过渡段类型不再生效，与 SetBlendType 配对使用。

16.4.13 GetAxisInput 获取驱动器输入

GetAxisInput 获取驱动器输入

1.轴号

1

2.位

1

3.输出变量

#GVar

▼

1

图表 107 GetAxisInput 获取驱动器输入

指令格式	GetAxisInput Axis= <i>Axis Number</i> Bit= <i>Bit</i> Value:表示变量输出 <i>#GVar</i> 或 <i>#Var</i> 。	
示例	GetAxisInput Axis=1 Bit=1=>#Var1;	
参数说明	轴号	<i>Axis</i> : 表示机器人关节轴号，最多可支持 16 个轴。
	位	Bit:
	变量	数字量输入值存储在数值型变量中。
注意事项		

16.5IO 操作

16.5.1 SetDo 输出数字量

SetDo数字量输出			
1.模块	0		
2.点位	Idx=	▼	1
3.输出值	数值	▼	0

图表 108 SetDo 设置

指令格式	SetDo Val= <i>Value</i> Grp= <i>Group</i> Idx = # <i>ALL</i>	
示例	SetDo Val=1 Grp=0 Idx=1; SetDo Val=0b1111 Grp=1 Idx=All;	
参数说明	模块	<i>Group</i> :表示第几组数字输出,控制器软件上认为 16 个 IO 为一组。
	点位	Idx.:表示该组中 IO 序号, 如果选择 <i>ALL</i> , 表示该组一起输出。
	输出值	<i>Value</i> :表示数字量输出 <i>0</i> 或 <i>1</i> 。如果选择 <i>ALL</i> , <i>value</i> 是二进制数 <i>0b01100110</i> 。
注意事项	× 15 Del < > Enter ☒ 1 ☒ 2 ☒ 3 ☒ 4 全部置1 ☐ 5 ☐ 6 ☐ 7 ☐ 8 全部置0 ☐ 9 ☐ 10 ☐ 11 ☐ 12 生成 ☐ 13 ☐ 14 ☐ 15 ☐ 16	如果选择 <i>ALL</i> , Val 被理解为二进制。在修改输出值时,生成的是十进制数字, 但显示为 0b 二进制数。下同

16.5.2 SetDoAt 运动中输出 IO

SetDoAt到达目标位置前输出IO			
1.距离	0mm		
2.模块	0		
3.点位	Idx= ▼	1	
4.输出值	数值 ▼	0	
5.延时时间	不使用		

图表 109 SetDoAt

指令格式	SetDoAt Dis= <i>Distance</i> Val= <i>Value</i> Grp= <i>Group</i> Idx = # <i>ALL</i> [Td= <i>Delay</i> ms]	
示例	SetDoAt Dis=7mm Val=1 Grp=1 Idx=1 Td=5ms; SetDoAt Dis=7mm Val=0b1111 Grp=1 Idx=All Td=5ms;	
参数说明	距离	<i>Distance</i> :表示离目标点的距离， 单位 mm。
	模块	<i>Group</i> :表示第几组数字输出,控制器软件上认为 16 个 IO 为一组。
	点位	Idx:表示该组中 IO 序号。如果选择 <i>ALL</i> ， 表示该组一起输出。
	输出值	<i>Value</i> :表示数字量输出 <i>0</i> 或 <i>1</i> 。如果选择 <i>ALL</i> ， <i>value</i> 是二进制数 <i>0b01100110</i> 。
	延时时间	<i>Delay</i> :表示滞后时间， 单位 ms。
注意事项	该指令仅对上一条指令生效。	

16.5.3 SetDoDeltaS 运动中输出数字量

SetDoDeltaS当前运动轨迹上等距离输出IO

1.间隔距离

0mm

2.模块

0

3.点位

Idx=

▼

1

4.输出值

数值

▼

0

5.持续时间

1ms

图表 110 SetDoDeltaS 设置

指令格式	SetDoDeltaS Dis= <i>DeltaS</i> Val= <i>Value</i> Grp= <i>Group</i> Idx = # <i>ALL</i> Width= #ms	
示例	SetDoDeltaS Dis=50mm Val=1 Grp=1 Idx=1 Width=2ms; SetDoDeltaS Dis=50mm Val=0b1111 Grp=1 Idx=All Width=2ms;	
参数说明	距离	<i>DeltaS</i> :表示间隔距离， 单位 mm
	模块	<i>Group</i> :表示第几组数字输出,控制器软件上认为 16 个 IO 为一组。
	点位	Idx:表示该组中 IO 序号。如果选择 <i>ALL</i> ，表示该组一起输出。
	输出值	<i>Value</i> :表示数字量输出 <i>0</i> 或 <i>1</i> 。如果选择 <i>ALL</i> ， <i>value</i> 是二进制数 <i>0b01100110</i> 。
	延时时间	<i>Width</i> :表示信号持续时间， 单位 ms。
注意事项	该指令仅对上一条指令生效。	

16.5.4 SetDoDeltaT 运动中输出数字量

SetDoDeltaT当前运动轨迹上等时间输出IO

1.时间间隔

0ms

2.模块

0

3.点位

Idx=

▼

1

4.输出值

数值

▼

0

5.持续时间

1ms

图表 111 SetDoDeltaT 设置

指令格式	SetDoDeltaT Time= <i>DeltaT</i> Val= <i>Value</i> Group = # Idx = # <i>ALL</i> Width=# ms	
示例	SetDoDeltaT Time=5ms Val=1 Group=1 Idx=1 Width=1ms; SetDoDeltaT Time=5ms Val=0b1111 Group=1 Idx=All Width=1ms;	
参数说明	时间间隔	<i>DeltaT</i> :表示时间间隔，单位 ms。
	模块	<i>Group</i> :表示第几组数字输出,控制器软件上认为 16 个 IO 为一组。
	点位	Idx:表示该组中 IO 序号。如果选择 <i>ALL</i> ，表示该组一起输出。
	输出值	<i>Value</i> :表示数字量输出 <i>0</i> 或 <i>1</i> 。如果选择 <i>ALL</i> ， <i>value</i> 是二进制数 <i>0b01100110</i> 。
	延时时间	Width:表示信号持续时间，单位 ms。
注意事项	该指令仅对上一条指令生效。	

16.5.5 GetDo 数字量输出读取

GetDo获取数字量输出并保存状态到变量

1.模块	0		
2.点位	Idx= ▼	1	
3.变量	#Var ▼	0	

图表 112 GetDo 参数

指令格式	GetDo Grp= <i>Group</i> Idx = # <i>ALL</i> => #Var#	
示例	GetDo Grp=1 Idx=1 =>#Var2; GetDo Grp=1 Idx=All =>#Var2;	
参数说明	模块	<i>Group</i> :表示第几组数字输出,控制器软件上认为16个IO为一组。
	点位	Idx:表示该组中 IO 序号。如果选择 <i>ALL</i> ，表示该组一起输出。
	变量	数字量输出值存储在数值型变量中。
注意事项		

16.5.6 GetDi 数字量输入获取

GetDi获取数字量输入并保存状态到变量

1.模块	0		
2.点位	Idx= ▼	1	
3.变量(#Var)	#Var ▼	0	

图表 113 GetDi 参数

指令格式	GetDi Grp= <i>Group</i> Idx = # => #Var#	
示例	GetDi Grp=1 Idx=1 =>#Var2;	

	GetDi Grp=1 ldx=All =>#Var2;	
参数说明	模块	<i>Group</i> :表示第几组数字输入,控制器软件上认为 16 个 IO 为一组。
	点位	ldx:表示该组中 IO 序号。如果选择 <i>ALL</i> , 表示该组一起输入。
	变量	数字量输入值存储在数值型变量中。
注意事项		

16.5.7 SetAo 输出模拟量

SetAo模拟量输出

1.输出值

数值

▼

0.00

2.点位

0

图表 114 SetAo 参数设置

指令格式	SetAo Val= <i>Value</i> ldx = #	
示例	SetAo Val=7.00 ldx=1;	
参数说明	输出值	<i>Value</i> : 表示模拟量输出值。
	点位	ldx: 表示该组中 IO 序号。
注意事项		

16.5.8 SetAoAt 运动中输出模拟量

SetAoAt距离目标位置指定位置输出模拟量

1.距离	0mm	
2.输出值	数值	0.00
3.点位	0	
4.滞后时间	不使用	

图表 115 SetAoAt 参数设置

指令格式	SetAoAt Dis= <i>Distance</i> Val= <i>Value</i> Idx = # [Td= <i>Delay</i> ms]	
示例	SetAoAt Dis=10mm Val=8.00 Idx=25ms;	
参数说明	距离	<i>Distance</i> : 表示离目标点的距离，单位 mm。
	输出值	<i>Value</i> : 表示模拟量值。
	点位	Idx: 表示该组中 IO 序号。
	滞后时间	<i>Delay</i> : 表示滞后时间，单位 ms。
注意事项	该指令仅对上一条指令生效。	

16.5.9 GetAo 模拟量输出获取

GetAo获取模拟量输出并保存状态到变量

1.点位	0	
2.变量(#Var)	#Var	1

图表 116 GetAo 参数设置

指令格式	GetAo Idx = # => <i>#Var#</i>	
示例	GetAo Idx=1 =>#Var2;	
参数说明	点位	Idx: 表示该组中 IO 序号。
	变量	模拟量输出值存储在变量中。

注意事项	
------	--

16.5.10 GetAi 模拟量输入获取

GetAi获取模拟量输入并保存状态到变量

1.点位

0

2.变量

#Var

▼

1

图表 117 GetAi 参数设置

指令格式	GetAi Idx = # =>#Var#	
示例	GetAi Idx=1 =>#Var2;	
参数说明	点位	Idx: 表示该组中 IO 序号。
	变量	模拟量输入值存储在变量中。
注意事项		

16.5.11 Wait 等待条件完成

Wait 等待条件完成

1.条件

0

2.等待时间

直到条件满足

3.超时输出变量

#Var

▼

不输出变量

图表 118 Wait 参数设置

等待一个布尔表达式

指令格式	Wait Cond= <i>Condition?</i> [Time = # ms => #Var#]	
示例	Wait Cond=(#Var1>>5) Time=5ms =>#Var2;	
参数说明	条件	<i>Condition?</i> : 自定义布尔表达式。

	等待时间	Time:表示等待时间上限, 不设置表示一直等待, 单位 s。
	超时输出变量	如果超时, <i>#Var#</i> 为 True, 否则为 False。
注意事项		

16.5.12 Pulse 脉冲输出

数字量输出置位为 1 并保持一段时间, 然后置零

Pulse 脉冲输出

1.模块

0

2.点位

0

3.持续时间

0ms

图表 119 Pulse 参数设置

指令格式	Pulse Grp= <i>Group</i> Idx = <i>#</i> Width = <i>#</i> ms	
示例	Pulse Grp=1 Idx=1 Width=5ms;	
参数说明	模块	<i>Group</i> :表示第几组数字输入,控制器软件上认为 16 个 IO 为一组。
	点位	Idx:表示该组中 IO 序号。
	持续时间	Width 表示置位时间, 单位 ms。
注意事项		

16.6运动管理

16.6.1 Delay 延时



图表 120 Delay 参数设置

设置延时，增加机器人运动的时间。

指令格式	Delay Time=#ms;	
示例	Delay Time=2000ms;	
参数说明	定时时间	定时时间，单位 ms。
注意事项		

16.6.2 WaitExecutor 停止译码



图表 121 WaitExecutor 停止译码设置

停止示教指令提前译码，等待机器人实际执行到当前指令行再自动开始译码。

指令格式	Delay Dis=#mm;	
示例	WaitExecutor Dis=100mm;	
参数说明	距离目标位置点距离	距离目标位置点距离，单位 mm。
注意事项		

16.6.3 tod 获取系统计时



图表 122 tod 参数设置

指令格式	#Var # = tod();	
示例	#Var1 = tod(); #Var2 = tod(); #Var3 = #Var2 - #Var1; 在 T1 时间记录一个时间，放入变量#Var1 中；在 T2 记录一个时间，放入变量#Var2 中；T1 和 T2 的时间差就等于#Var2 的值减#Var1 的值。	
参数说明	变量	记录的时间存储在#Var#中。

16.6.4 SysDate 记录系统时间



图表 123 SysDate 参数设置

指令格式	SysDate =>#Var#	
示例	SysDate =>#Var1;	
参数说明	变量	Date: 系统时间记录在字符串变量中。

16.6.5 Pause 暂停

停止执行，等待外部信号重新开启

示例	Pause;
----	--------

16.6.6 Alarm 用户报警

Alarm 自定义报警

1.报警等级	Warning	▼
2.报警内容	warring	
3.条件	DO1.1==1	
4.输出变量	#Var	▼ 1

图表 124 Alarm 自定义报警参数设置

通过示教程序主动触发报警下伺服，可以设置触发条件

示例	Alarm Level=Warning Message=“报警内容”Cond=(条件);
----	--

16.6.7 GetPoint 保存当前位置

GetPoint 获取当前位置并保存到位置变量

1.坐标系	KCS	▼
2.位置点	#P	▼ 1

图表 125 GetPoint 参数设置

指令格式	GetPoint <i>Type</i> => P= <i>Point</i>	
示例	GetPoint KCS =>#P1;	
参数说明	坐标系	<i>Type</i> 坐标系型号，KCS,WCS,PCS, ACS。
	位置点	<i>Point</i> 存储位置点。
注意事项		

16.7 控制流程

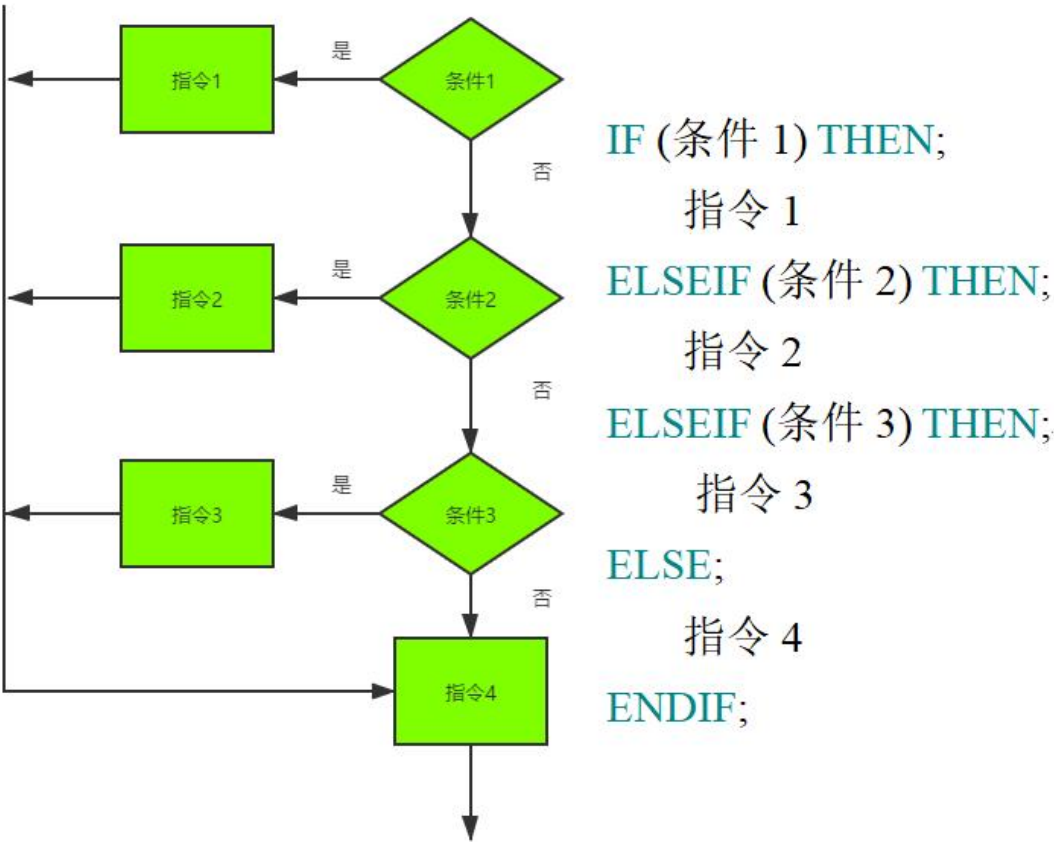
16.7.1 CALL 调用子程序

示例	<pre>1 PROCEDURE 2 Movj P=#P1 V=20%; 3 CALL call; 4 Movl P=#P2 V=20mm/s; 5 ENDPROCEDURE</pre>	当机器人末端运动到点 P1 后，开始执行第 3 行程序：调用名为“call”的子程序，在执行完子程序“call”后，返回原程序继续执行第 4 行程序运动到点 P4。 注意：调用与被调用的程序必须在同一工程下。子程序名：只能包含字母数字下划线，并且不能数字开头。
----	---	--

16.7.2 JUMP—LABEL 跳转—跳转标识

示例	<pre>2 Movj P=#P1 V=20%; 3 JUMP 1; 4 Movj P=#P2 V=20%; 5 LABEL 1; 6 Movj P=#P3 V=20%;</pre>	机器人末端运动到点 P1 后，跳转到标记“1”的第 5 行，往下执行第 6 行运动到点 P3
----	---	---

16.7.3 IF 条件判断



图表 126 IF 流程图

示例	<pre>2 Movj P=#P1 V=20%; 3 IF (#Var1>=5) THEN; 4 Movl P=#P2 V=20mm/s; 5 ELSE; 6 Movl P=#P3 V=20mm/s; 7 ENDIF;</pre>	机器人末端运动到点 P1 后，判断条件（#Var1 的值是否大于等于 5），如果大于等于 5，则运行第 4 行，运动到点 P2；否则运动到点 P3。
----	--	--

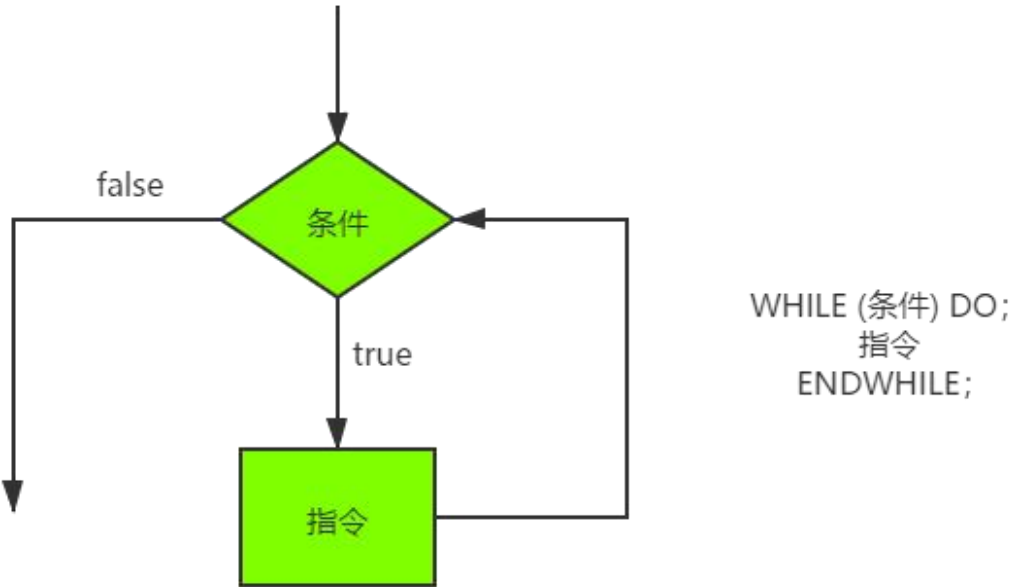
16.7.4 ELSELF 条件判断

参考 IF 流程图。

示例	<pre>2 Movj P=#P1 V=20%; 3 IF (#Var1=5) THEN; 4 Movl P=#P2 V=20mm/s; 5 ELSEIF (#Var1 = 8) THEN; 6 Movl P=#P3 V=20mm/s; 7 ELSE; 8 Movl P=#P4 V=20mm/s; 9 ENDIF;</pre>	机器人末端运动到点 P1 后，判断条件 1（#Var1 的值是否等于 5），如果等于 5，则运行第 4 行，运动到点 P2，结束判断；判断条件 2（#Var1 的值是否等于 8），如果等于 8，则运行第 6 行，运动到点 P3，结束判断；否则运动到点 P4。
----	--	--

16.7.5 WHILE 循环条件判断

WHILE 循环：条件循环，先判断后执行，条件不成立跳出循环



图表 127 WHILE 流程图

示例	<pre>2 Movj P=#P1 V=20%; 3 #Var1 = 1; 4 WHILE (#Var1>=10) DO; 5 Movj P=#P2 V=20%; 6 Movj P=#P3 V=20%; 7 #Var1 = #Var1 + 1; 8 ENDWHILE;</pre>	机器人末端运行到 P1（过渡点），将#Var1 的值设为 1，判断条件 1 的值为 False（不满足条件 1），循环执行程序运动到点 P2、P3、#Var1+1，直到#Var1=10 时满足条件 1 退出循环。机器人末端在 P2、P3 往返共 9 次。
----	---	---

16.7.6 FOR 循环条件判断

FOR 循环

FOR

#Var

▼

1

FROM: 1

TO: 10

STEP: 1

DO

图表 128 FOR 循环

指令格式	<pre>FOR #Var# FROM #1TO #2 STEP #3 DO; 程序 1; ENDFOR;</pre>	默认将#Var#的值设置为#1，当#Var#的值在区间[#1， #2]时循环执行程序 1，循环步数为#3 即#Var#每循环一次加#3（#Var# = #Var#+#3）。
------	---	--

示例 1	<pre>2 Movj P=#P1 V=20%; 3 FOR #Var1 FROM 1 TO 10 STEP 1 DO; 4 Movl P=#P2 V=20mm/s; 5 Movl P=#P3 V=20mm/s; 6 ENDFOR;</pre>	机器人末端先运行到 P1（过渡点），然后在点 P2、P3 间循环跑 10 次后退出 FOR 循环。 （#Var1 的值在第一次循环时设置为 1，每次 +1，直到循环 10 次后 #Var1=11>10,退出循环）
示例 2	<pre>9 Movj P=#P1 V=20%; 10 #Var1 = 1; 11 FOR #Var1 FROM 1 TO 10 STEP 2 DO; 12 Movl P=#P2 V=20mm/s; 13 Movl P=#P3 V=20mm/s; 14 Movc P=#P1 Mid=#P1 V=20mm/s; 15 ENDFOR;</pre>	机器人末端先运行到 P1（过渡点），然后在点 P2、P3 间循环跑 5 次后退出 FOR 循环。 （#Var1 的值在第一次循环时设置为 1，每次 +2，直到循环 5 次后 #Var1=11>10,退出循环）

16.8 注释及计算

16.8.1 `/**/`多行注释

示例	<pre>4 /*FOR #Var1 FROM 1 TO 10 STEP 1 DO; 5 Movl P=#P2 V=20mm/s; 6 Movl P=#P3 V=20mm/s; 7 Movc P=#P1 Mid=#P1 V=20mm/s; 8 ENDFOR;*/</pre>	在起始行插入 <code>{/*多行注释}</code> ，在结束行插入 <code>{*/多行注释}</code> 。取消注释重复操作即可。
----	---	---

16.8.2 注释/取消注释当前行

示例	<pre>//Movj P=#P1 V=20%;</pre>	点击 <code>{//注释/取消注释当前行}</code> 进行注释或取消注释
----	---------------------------------------	--

16.8.3 `ASSIGN(=)`赋值

ASSIGN (=)赋值运算

结果：

#GVar

▼

1

=

#GVar

▼

1

图表 129 赋值运算

指令格式	<code>A = B</code>
示例	<pre>#GVar1 = #Var1; #Var1 = 1.58; #GStr1 = #Str5; #Str1 = "jjkkl"; #GP1 = #GP5;</pre>

	#GIncP1 = #IncP2;
参数说明	将 B 的值赋予给 A; 支持数值变量、字符串变量、位置变量和位置增量变量。

16.8.4 ADD(+)加法

ADD (+)加法

结果：

#GVar

▼

1

=

#Var

▼

1

+

VALUE

▼

1.0000

图表 130 加法运算

指令格式	C=A+B
示例	#GVar1 = #GVar1+1; #GVar1 = #Var1+#Var2; #GVar1 = 1+2;
参数说明	将 A 的值与 B 的值相加后赋予给 C；支持数值变量

16.8.5 SUB(-)减法

SUB (-)减法

结果：

#GVar

▼

1

=

#GVar

▼

1

-

VALUE

▼

1.0000

图表 131 减法运算

指令格式	C=A-B
------	-------

示例	GVar1 = #GVar1-1; #GVar1 = #Var2-#Var1; #Var1 = 5-1;
参数说明	将 A 的值与 B 的值相减后赋予给 C；支持数值变量

16.8.6 MUL(*)乘法

ADD (+)加法

结果：

#GVar

▼

1

=

#Var

▼

1

+

VALUE

▼

1.0000

图表 132 乘法运算

指令格式	C=A*B
示例	#GVar1 = #GVar1*#Var2; #GVar1 = #GVar1*2; #GVar1 = 2*3;
参数说明	将 A 的值与 B 的值相乘后赋予给 C；支持数值变量

16.8.7 DIV(/)除法

ADD (+)加法

结果：

#GVar

▼

1

=

#Var

▼

1

+

VALUE

▼

1.0000

图表 133 加法运算

指令格式	C=A/B
------	-------

示例	#GVar1 = #GVar1/#Var2; #GVar1 = #GVar1/2; #GVar1 = 2/3;
参数说明	将 A 的值除以 B 的值后赋予给 C；支持数值变量

16.8.8 Expr 高级表达式/语句

Expr 高级表达式支持：

- 1.四则运算
- 2.位运算
- 3.布尔运算
- 4.数学函数
- 5.字符串处理



图表 134 高级表达式界面

扩展说明：

运算

%	模(取余操作)
++	自增
--	自减
&	按位与
	按位或
^	按位异或
<<	左移
>>	右移
~	取反

数学函数

ceil()	向上取整
trunc(,)	变量取整
exp()	e 指数函数
log(,)	10 对数函数
mod(,)	求余函数
sqrt(,)	平方根
pow(,)	指数函数
sin()	正弦
cos()	余弦
tan()	正切
acos()	反余弦

asin()	反正弦
atan()	反正切
atan2()	反正切（双变量）
abs()	绝对值

逻辑

&&	逻辑与
	逻辑或
!	逻辑非
>=	大于等于
<=	小于等于
>	大于
<	小于
==	等于
!=	不等于

字符函数

strtovar()	将字符串转换为数值型
vartostr()	将数值型转换为字符串
strcat(,)	字符串拼接
strlen()	计算字符串长度
strcmp(,)	字符串比较

strtok(,)	根据分隔符将字符串分隔成一个个片段
-----------	-------------------

其他

tod()	计时指令
-------	------

中文及大小写切换

点击右下角{切换}，左侧会切换不同的输入按键。



示例：

1_weizi : main

libpng warning: iCCP: known incorrect sR

```
1 PROGRAM
2 SetSpeed Vel=20%;
3 #Var1 = 1;
4 Movj P=#P1 V=20%;
5 LABEL 1;
6 IF (#Var1<=30) THEN;
7     Movl P=#P2 V=2000mm/s;
8     Delay Time=2000ms;
9     Movl P=#P3 V=2000mm/s;
10    Delay Time=2000ms;
11    Movl P=#P4 V=2000mm/s;
12    Delay Time=2000ms;
13    Movl P=#P5 V=2000mm/s;
14    Delay Time=2000ms;
15    Movl P=#P6 V=2000mm/s;
16    Delay Time=2000ms;
17    #Var1 = #Var1 + 1;
18 ELSE;
19     JUMP 2;
20 ENDIF;
21 JUMP 1;
22 LABEL 2;
23 ENDPROGRAM
```

图表 135 高级表达式示例

如第 3 行：给变量#Var 赋值

第 6 行：IF 条件判断语句（自定义布尔表达式）

都可以通过高级表达式输入。

16.9数学函数

abs(X)	取绝对值
sqrt(X)	开根号
exp(X)	自然指数
ln(X)	自然对数
sin(X)	正弦

cos(X)	余弦
tan(X)	正切
asin(X)	反正弦
acos(X)	反余弦
floor(X)	向下取整
ceil(X)	向上取整
round(X)	四舍五入取整
trunc(X)	取整
pow(X,Y)	X 的 Y 次幂
mod(X,Y)	X 对 Y 的余
atan2(X,Y)	反正切

16.10 字符函数

16.10.1 strlen(X) 求字符串长度



图表 136 strlen (X) 求字符串长度

参数说明	计算字符串的长度并保存到一个数值变量中
指令示例	#Str1=asfsdfasg

	#Var1 =strlen(#Str1)
运算结果	#Var1= 9;

16.10.2 str2num(X)字符串转实数

str2num (X)字符串转实数

结果：

#GVar

▼

1

= str2num(

参数1：

#GStr

▼

1

)

图表 137 str2num (X) 字符串转实数

参数说明	将字符串转换成实数变量
指令示例	#Var1=str2num(#Str1)
参数说明	一般用于将收发的数字字符串转换成实数进行计算或赋值

16.10.3 num2str(X)实数转字符串

num2str (X)实数转字符串

结果：

#GStr

▼

1

= num2str(

参数1：

#GVar

▼

1

)

图表 138 num2str (X) 实数转字符串

参数说明	将实数变量转换成字符串
指令示例	#Str1=num2str(#Var1)
参数说明	一般用于将位置点的 x,y,z 等数值转出字符串拼接发送

16.10.4 strcat(X,Y)字符串拼接

strcat (X,Y)X+Y

结果：

#Str

▼

2

= strcat(

参数1：

#Str

▼

3

参数2：

#GStr

▼

4

)

图表 139 strcat(X,Y)字符串拼接

参数说明	将 2 个字符串拼接成 1 个字符串
指令示例	#Str2=asd #Str3=fgh #Str1=strcat(#Str2,#Str3)
运算结果	#Str1=asdfgh

16.10.5 strleft(X,N)取字符串 X 的左边 N 个字符

strleft (X,N)取X的左边N个字符

结果：

#GStr

▼

1

= strleft(

参数1：

#GStr

▼

1

参数2：

0

)

图表 140 strleft(X,N)取左字符

参数说明	取字符串 X 的左边 N 个字符
指令示例	#Str2=asdfgh #Str1=strleft(#Str2,2)

运算结果	#Str1=st
------	----------

16.10.6 strright(X,N)取字符串 X 的右边 N 个字符

strright (X,N)取X的右边N个字符

结果：

#GStr

▼

1

= strright(

参数1：

#GStr

▼

1

参数2：

0

)

图表 141 strright(X,N)取右字符

参数说明	取字符串 X 的右边 N 个字符
指令示例	#Str2=asdfgh #Str1=strright(#Str2,2)
运算结果	#Str1=gh

16.10.7 strmid(X,M,N)取字符串 X 的第 M 至 N 个字符

strmid (X,M,N)取X的第M至N字符

结果：

#GStr

▼

1

= strmid(

参数1：

#GStr

▼

1

参数2：

0

参数3：

0

)

图表 142 strmid(X,M,N)取字符

参数说明	取字符串 X 的第 M 至 N 个字符
------	---------------------

指令示例	<pre>#Str2=asdfgh #Str1=strmid(#Str2,3,4)</pre>
运算结果	<pre>#Str1=df</pre>

16.10.8 strsplit(X,Y,N)字符串拆分

strsplit (X,Y,N)按Y拆分X取第N个结果

结果：

#GStr

▼

1

= strsplit(

参数1：

#GStr

▼

1

参数2：

,

参数3：

0

)

图表 143 strsplit(X,Y,N)字符串拆分

参数说明	将字符串 X 按分隔符 Y 拆分，取第 N 段字符
指令示例	<pre>#Str2=a,s,d,f,g,h #Str1 = strsplit(#Str2, ",", 3);</pre>
运算结果	<pre>#Str1=d</pre>

16.10.9 buildstr 字符串构造

buildstr 字符串构造

结果：

#GStr

▼

1

=buildstr(

)

图表 144 buildstr 字符串构造

参数说明	将格式串 format 中的#Str, #GStr, #Var, #GVar 替换为实际变量并写入到 message 中
指令示例	#Str1 = buildstr("value = #GVar1")
运算结果	#Str1 = "value = 1"(假设#GVar1 = 1)

16.10.10 parsestr 字符串解析

parsestr 字符串解析

parsestr(

1.参数1

2.参数2

)

图表 145 parsestr 字符串解析

参数说明	按照指定格式解析 message,并将 message 中指定的字符转换为变量
指令示例	parse("value = #GVar1", "value = 3.14")
运算结果	#GVar1 = 3.14

16.11 通讯

16.11.1 TcpOpen 打开 tcp/ip 通讯

TcpOpen 开启tcp/ip通讯

1.连接名称

1

2.连接类型

"server"

▼

3.阻塞时间

0ms

4.ip:port

:1

5.输出值

#Var

▼

1

图表 146 TcpOpen 打开 tcp/ip 通讯

指令示例	TcpOpen name="imf" type=""server"" block=1 addr="192.168.100.33:22" =>#Var1;	
参数说明	连接名称	该链接的标识符，后续接收发送操作与此标识符相关；
	连接类型	链接类型， server：创建服务器； client：创建客户端并链接到服务器；
	阻塞时间	指令是否阻塞， 1：阻塞时间（ms）；指令执行的等待时间 0：非阻塞；
	ip:port	链接的 ip 地址：端口；其中，服务器端可以全网口监听，所以创建服务器无需 ip
	输出值	输出值， >=0：socket 维护的链接数； -1：服务器启动中，客户端链接中； -2：失败；

16.11.2 TcpRecv 接收数据

TcpRecv 接收数据

1.连接名称

2.数据保存

#Str

▼

1

3.阻塞时间

0ms

4.输出值

#Var

▼

1

图表 147 TcpRecv 接收数据

指令示例	TcpRecv name="imf" dest=#Str1 block=0 =>#Var1;	
参数说明	连接名称	接收数据链接标识符， 由 TcpOpen 创建；
	数据保存	接收的数据存放的字符串变量；
	阻塞时间	指令是否阻塞， 1： 阻塞时间（ms）； 指令执行的等待时间 0： 非阻塞；
	输出值	输出值， >=0： 收到的字节数； -2： 没有对应的标识符；

16.11.3 TcpSend 发送数据

TcpSend 发送数据

1.连接名称	imf	
2.数据保存	#Str	▼ 1
3.阻塞时间	0ms	
4.输出值	#Var	▼ 1

图表 148 TcpSend 发送数据

指令示例	TcpSend name="imf" src=#Str1 block=0 =>#Var1;	
参数说明	连接名称	发送数据的链接标识符， 由 TcpOpen 创建；
	数据保存	发送的数据存放的字符串变量；
	阻塞时间	指令是否阻塞， 1： 阻塞时间（ms）； 指令执行的等待时间 0： 非阻塞；

	输出值	输出值, >=0: 发送的字节数; -3: 没有对应的标识符;
--	-----	---

16.11.4 TcpClose 关闭 tcp/ip 连接

TcpClose 关闭tcp/ip连接

1.连接名称

图表 149 TcpClose 关闭 tcp/ip 连接

指令示例	TcpClose name="imf";	
参数说明	连接名称	关闭链接标识符，由 TcpOpen 创建;

16.11.5 SerialOpen 打开串口

SerialOpen 打开串口

1.连接名称

2.地址0:rs232 4:rs485

3.波特率▼

4.校验▼n:无校验 o:奇 e:偶

5.停止位

6.阻塞时间

7.输出值▼

图表 150 SerialOpen 打开串口

指令示例	SerialOpen name="imf" addr=4 par="1200,8,n,1" block=0 =>#Var1;	
参数说明	连接名称	打开串口标识符，后续接收发送数据与此标识符相关;
	地址	打开的串口地址，以 0，1，2.....来标记;

		0: rs232; 4: rs485;
	波特率	每秒钟传送的码元符号的个数
	校验	选择检错方式, n: 没有校验检测, o 奇校验, e 偶校验
	停止位	表示传输的结束, 并且提供计算机校正时钟同步的机会。适用于停止位的位数越多, 不同时钟同步的容忍程度越大, 但是数据传输率同时也越慢。
	阻塞时间	指令是否阻塞, 1: 阻塞时间 (ms) ; 指令执行的等待时间 0: 非阻塞;
	输出值	0: 串口打开正常; -1: 失败;

16.11.6 SerialRecv 接收数据

SerialRecv 接收数据

1.连接名称

2.数据保存

#Str

▼

1

3.阻塞时间

0ms

4.输出值

#Var

▼

1

图表 151 SerialRecv 接收数据

示例	SerialRecv name="imf" dest=#Str1 block=0 =>#Var1;	
参数说明	连接名称	接收数据串口标识符, 由 SerialOpen 创建;
	数据保存	接收的数据存放的字符串变量;

	阻塞时间	指令是否阻塞, 1: 阻塞时间 (ms) ; 指令执行的等待时间 0: 非阻塞;
	输出值	输出值, >=0: 收到的字节数; -1: 串口未打开;

16.11.7 SerialSend 发送数据

SerialSend 发送数据

1.连接名称		
2.数据保存	#Str	▼ 1
3.阻塞时间	0ms	
4.输出值	#Var	▼ 1

图表 152 SerialSend 发送数据

指令示例	SerialSend name="imf" src=#Str1 block=0 =>#Var1;	
参数说明	连接名称	发送数据的串口标识符, 由 SerialOpen 创建;
	数据保存	发送的数据存放的字符串变量;
	阻塞时间	指令是否阻塞, 1: 阻塞时间 (ms) ; 指令执行的等待时间 0: 非阻塞;
	输出值	输出值, >=0: 发送的字节数; -1: 串口未打开;

16.11.8 SerialClose 关闭 tcp/ip 连接

SerialClose 关闭tcp/ip连接

1.连接名称

图表 153 SerialClose 关闭 tcp/ip 连接

指令示例	SerialClose name="imf";	
参数说明	连接名称	关闭串口标识符，由 SerialOpen 创建；

16.11.9 CanbusOpen 打开 Canbus 通讯

CanbusOpen 接收数据

1.连接名称

2.地址

0

3.波特率(k)

125

4.阻塞时间

0ms

5.输出值

#Var

1

图表 154 打开 Canbus 通讯

指令示例	CanbusOpen name="imf" addr=1 par=125 block=0 =>#Var1;	
参数说明	连接名称	打开 canbus 标识符，后续接收发送数据与此标识符相关；
	地址	canbus 地址，以 0, 1, 2 来标记，目前只有一个 canbus，填 0；
	波特率	canbus 波特率：125 表示125k，此外还有常用的 1000k，500k，250k 等等；
	阻塞时间	指令是否阻塞， 1：阻塞时间（ms）；指令执行的等待时间 0：非阻塞；

	输出值	0: canbus 打开正常; -1: 失败;
--	-----	----------------------------

16.11.10 CanbusRecv 接收数据

CanbusRecv 接收数据			
1.连接名称			
2.数据保存	#Str	▼	1
3.阻塞时间	0ms		
4.输出值	#Var	▼	1

图表 155 CanbusRecv 接收数据

指令示例	CanbusRecv name="imf" dest=#Str1 block=0 =>#Var1;	
参数说明	连接名称	canbus 标识符，由 CanbusOpen 创建;
	数据保存	收到的数据帧会转换成字符串，以“帧 id，帧负载，标准帧/扩展帧，数据帧/远程帧，数据;”的形式存放，例如：000000FF,08,std,d,09,0A,0B,0C,0D,0E,0F,10;000000FE,8,ext,r,09,0A,0B,0C,0D,0E,0F,10 ，一次收到多帧数据，以";"分割;
	阻塞时间	指令是否阻塞， 1: 阻塞时间（ms）；指令执行的等待时间 0: 非阻塞;
	输出值	输出值， >=0: 收到的 can 帧数; -1: 失败;

16.11.11 CanbusSend 发送数据

CanbusSend 发送数据		
1.连接名称		
2.数据保存	#Str	1
3.阻塞时间	0ms	
4.输出值	#Var	1

图表 156 CanbusSend 发送数据

指令示例	CanbusSend name="imf" src=#Str1 block=0 =>#Var1;	
参数说明	连接名称	canbus 标识符， 由 CanbusOpen 创建；
	数据保存	发送数据帧存储的字符串， 格式同 CanbusRecv
	阻塞时间	指令是否阻塞， 1： 阻塞时间（ms）； 指令执行的等待时间 0： 非阻塞；
	输出值	输出值， >=0： 收到的 can 帧数； -1： 失败；

16.11.12 CanbusClose 关闭 Canbus 连接

CanbusClose 关闭tcp/ip连接	
1.连接名称	

图表 157 CanbusClose 关闭 Canbus 连接

指令示例	CanbusClose name="imf";	
参数说明	连接名称	关闭 canbus 标识符， 由 CanbusOpen 创建；

16.11.13 MBTcpOpen 开启 modbusTcp 通讯

MBTcpOpen 开启modbusTcp通讯

1.连接名称

2.连接类型

server

▼

3.阻塞时间

0ms

4.ip:port

192.168.100.4

:1

5.输出值

#Var

▼

1

图表 158 MBTcpOpen 开启 modbusTcp 通讯

指令示例	MBTcpOpen name="imf" type="server" block=0 addr"192.168.100.3:1"=>#Var1;	
参数说明	连接名称	该链接的标识符，后续接收发送操作与此标识符相关；
	连接类型	链接类型， server：创建服务器； client：创建客户端并链接到服务器；
	ip:port	链接的 ip 地址：端口；其中，服务器端可以全网口监听，所以创建服务器无需 ip
	阻塞时间	指令是否阻塞， 1：阻塞时间（ms）；指令执行的等待时间 0：非阻塞；
	输出值	0：canbus 打开正常； -1：失败；

16.11.14 MBTcpRead/Write 读/写寄存器

MBTcpWrite 写寄存器

指令类型

MBTcpWrite

1.连接名称

2.类型

hold

3.起始地址0x

4.起始变量

#Var

1

5.数量

1

6.阻塞时间

0ms

图表 159 MBTcpRead/Write 读/写寄存器

指令示例	MBTcpWrite name="imf" type="hold" addr=0x10001 var=#Var1 count=1 block=0;	
参数说明	连接名称	该链接的标识符，后续接收发送操作与此标识符相关；
	指令类型	Write：写寄存器； Read：读寄存器；
	起始地址	寄存器的起始地址 0x： 10001；
	类型	寄存器类型 1： hold； 2： input； 3： coil； 4： discrete；
	起始变量	发送的数据存放的数值变量；
	数量	写入变量的数量；
	阻塞时间	指令是否阻塞， 1： 阻塞时间 （ms） ； 指令执行的等待时间 0： 非阻塞；

16.11.15 MBTcpOpen 关闭 modbusTcp 通讯



图表 160 MBTcpOpen 关闭 modbusTcp 通讯

指令示例	MBTtcpClose name="imf";	
参数说明	连接名称	关闭链接标识符，由 MBTtcpOpen 创建；

17 错误信息

错误代码	错误信息	错误分析	解决办法
0001	Hardware slave configuration fails, path is invalid, error code %1 (1: null; 2: length<3; 3: length>4096)	硬件从站配置失败，路径无效，错误码%1 (1: 空指针; 2: 长度小于 3; 3: 长度大于 4096)	内部错误，请升级或还原程序
0002	Hardware initialization fails, error code %1	硬件初始化失败，错误码%1	检查硬件是否上电
0003	Hardware not ready, error code %1 (-1: ethercat disconnected; -2: slave number change; -13: hardware not register; -23: NIC driver abnormal; -33: cable disconnected)	硬件没有准备好，错误码 %1 (-1: 通讯断开; -2: 从站数目变化; -3: 硬件未注册; -23: 网卡驱动异常; -33: 网线断开)	检查 EtherCAT 网络通讯是否正常 检查从站数目是否变化 检查是否注册
0004	Axis %1 is not ready for interpolation, error	轴%1 没有做好插补准备，错误码 %2	检查 EtherCAT 通讯是否正常，断电重启

	code %2		
0005	Hardware slave %1 error, driver error code %2, ref to driver manual	硬件从站%1 错误，从站 错误代码%2，参考具体 的驱动器说明手册	从站错误，参考具体 从站品牌的处理方 法
0006	Read encoder (0-based) %1 fails, error code %2	编码器%1 读取错误，错 误码 %2	检查 EtherCAT 通讯 是否正常，网线是否 断开
0007	Axis %1 servo on fails, error code %2 (-1: ethercat not ready; -4: no power supply; -5: slave error)	轴%1 上伺服失败，错误 代码 %2 (-1: 通讯未准 备好; -4: 未提供强电; -5: 从站错误)	检查 EtherCAT 通讯 是否正常，网线是否 断开，是否上电
0008	Axis %1 servo off fails, error code %2	轴%1 下伺服失败，错误 码 %2	检查 EtherCAT 通讯 是否正常，网线是否 断开，是否上电
0009	Axis %1 clear error fails, error code %2	轴%1 下清除错误失败， 错误码%2	检查 EtherCAT 通讯 是否正常，网线是否 断开，是否上电
000A	Axis %1 get slave error code fails, error code %2	轴%1 获取驱动器错误码 失败，错误码%2	检查 EtherCAT 通讯 是否正常，网线是否 断开，是否上电

000B	Axis %1 get slave state fails, error code %2	轴%1 获取驱动器状态失败, 错误码%2	检查 EtherCAT 通讯是否正常, 网线是否断开, 是否上电
000C	Read profiler (0-based) %1 fails, error code %2	规划器%1 读取错误, 错误码%2	检查 EtherCAT 通讯是否正常, 网线是否断开, 是否上电
000D	Io is not ready, error code %1	IO 未准备就绪, 错误码%1	检查 IO
1001	Software configuration is invalid	软件配置参数无效	检查软件设置参数 参看具体哪个参数 设置失败的错误信息
1002	Software initialization fails	软件初始化失败	检查软件设置参数
1003	Joint reference calibration is invalid	零点参数无效	重新标定零点
1004	Joint %1 positive limit error, current %2 out of [%3, %4]	关节%1 正限位错误, 当前%2, 超出限位[%3, %4]	点动回运动范围后 警告自动清除
1005	Joint %1 negative limit error, current %2 out of [%3, %4]	关节%1 负限位错误, 当前%2, 超出限位[%3, %4]	点动回运动范围后 警告自动清除

1006	Cartesion (XYZABC) %1 positive limit error, current %2 out of [%3, %4]	工作空间%1 正限位错 误, 当前%2, 超出限位 [%3, %4]	点动回运动范围后 警告自动清除
1007	Cartesion (XYZABC) %1 negative limit error, current %2 out of [%3, %4]	工作空间%1 负限位错 误, 当前%2, 超出限位 [%3, %4]	点动回运动范围后 警告自动清除
1008	Joint %1 overspeed, current %2, beyond limit %3	关节%1 超速, 当前%2, 超出限制%3	手动清除错误
1009	%1 linear velocity overspeed, current %2, beyond limit %3	%1 方向线速度超速, 当 前%2, 超出限制%3	手动清除错误
100A	%1 angular velocity overspeed, current %2, beyond limit %3	%1 方向角速度超速, 当 前%2, 超出限制%3	手动清除错误
100B	Motion time out fatal error, current(ns) %1, beyond motion cycle(ns) %2	运动规划超时致命错 误, 当前(纳秒)%1, 超 过运动规划周期(纳 秒)%2	请联系技术支持 或者修改规划周期 和 EtherCAT 周期参 数
100C	Motion time out	运动规划超时警告, 当	请联系技术支持

	warning, current(ns) %1, beyond 80% of motion cycle(ns) %2	前(纳秒)%1, 超过运动 规划周期(纳秒)%2 的 80%	或者修改规划周期 和 EtherCAT 周期参 数
100D	Cannot open premotion log file	无法打开预处理日志文 件	无需处理
100E	Inverse kinematic fatal error, target point (KCS without tool, UUID=%1) is X: %2, Y: %3, Z: %4, A: %5, B: %6, C%7, pose idx: %8, error code %9	运动学逆解致命错误, 目标点(机器人坐标系无 工具,指令行号%1)是 X:%2, Y:%3, Z:%4, A:%5, B:%6, C:%7, 姿态序号:%8, 错误代码%9	检查示教点
100F	Target point is not reachable, target point (KCS without tool, UUID=%1) is: X: %2, Y: %3, Z: %4, A: %5, B: %6, C%7, pose idx: %8, error code %9	目标点无法到达, 目标 点(机器人坐标系无工 具,指令行号%1)是 X:%2, Y:%3, Z:%4, A:%5, B:%6, C:%7, 姿态序号:%8, 错误 代码%9	检查示教点
1010	Movc points are co-linear, command uuid is %1	Movc 的 3 个点共线, 指 令行号为%1	检查示教点

1011	External cannot follow robot, as robot trajectory is too small, uuid = %1	外部轴无法跟随机器人，因为机器人轨迹太短，指令行号%1	重新示教机器人运动轨迹
1012	Start point pose index %1 != End point pose index %2, UUID = %3	起始点姿态序号%1 和目标点姿态序号%2 不一致，指令行号为%3	检查示教点姿态
1013	Start point pose index %1 != Aux point pose index %2, UUID = %3	起始点姿态序号%1 和辅助点姿态序号%2 不一致，指令行号为%3	检查示教点姿态
2001	Invalid parameter: profile ts %1, out of [0.0005, 0.016]	无效参数：规划周期%1，取值范围 [0.0005, 0.016]	检查设置参数
2002	Invalid parameter: profile size %1, out of [32, 512]	无效参数：规划滤波器长度%1，取值范围 [32, 512]	检查设置参数
2003	Invalid parameter: smooth size %1, out of [0, 128]	无效参数：平滑滤波器长度%1，取值范围 [0, 128]	检查设置参数
2004	Invalid parameter: %1	无效参数： %1 编码器实	检查设置参数

	encoder actual resolution %2	实际分辨率%2	
2005	Invalid parameter: %1 encoder feedback resolution %2	无效参数: %1 编码器反 馈分辨率%2	检查设置参数
2006	Invalid parameter: %1 encoder sign %2	无效参数: %1 编码器方 向%2	检查设置参数
2007	Invalid parameter: %1 motor max speed %2	无效参数: %1 编码器最 大转速%2	检查设置参数
2008	Invalid parameter: %1 motor max current %2	无效参数: %1 编码器最 大电流%2	检查设置参数
2009	Invalid parameter: %1 motor max torque %2	无效参数: %1 编码器最 大力矩%2	检查设置参数
200A	Invalid parameter: %1 joint postive limit %2, negative limit %3	无效参数: %1 关节正限 位%2, 负限位%3	检查设置参数
200B	Invalid parameter: %1 joint max velocity %2, max jog velocity %3	无效参数: %1 关节最大 速度%2, 最大点动速 度%3	检查设置参数
200C	Invalid parameter: %1 joint, gear ratio in joint side %1 > gear ratio in	无效参数: %1 关节, 减 速比关节端%1 > 减速 比电机端	检查设置参数

	motor side %2		
200D	Invalid parameter: %1 joint sign %2	无效参数: %1 关节方 向%2	检查设置参数
200E	Invalid parameter: Cartesion %1 postive limit %2, negative limit %3	无效参数: 工作空间%1 正限位%2, 负限位%3	检查设置参数
200F	Invalid parameter: Cartesion %1 max velocity %2, max jog velocity %3	无效参数: 工作空间%1 最大速度%2, 最大点动 速度%3	检查设置参数
2010	Invalid parameter: Cartesion %1 sign %2	无效参数: 工作空间%1 方向%2	检查设置参数
2011	Invalid parameter: Cartesion end linear max velocity %1, max acceleration %2, max decceleration %3	无效参数: 工作空间末 端最大线速度%1, 最大 加速度%2, 最大减速 度%3	检查设置参数
2012	Invalid parameter: Cartesion end angular max velocity %1, max acceleration %2, max	无效参数: 工作空间末 端最大角速度%1, 最大 角加速度%2, 最大角减 速度%3	检查设置参数

	decceleration %3		
2013	Invalid parameter: Joint space end max velocity %1, max acceleration %2, max decceleration %3	无效参数：关节空间末端最大速度%1，最大加速度%2，最大减速度%3	检查设置参数
2014	Invalid parameter: External group %1, joint space end max velocity %2, max acceleration %3, max decceleration %4	无效参数：外部轴组%1，在关节空间末端最大速度%2，最大加速度%3，最大减速度%4	检查设置参数
2015	Invalid parameter: %1 coordinate, index %2	无效参数：%1 坐标系，序号 %2	检查设置参数
2016	Invalid parameter: Joint %1 reference zero pulse, based on referemce current joint is %2	无效参数：关节%1 参考零点，基于参考零点当前角度为%2	检查设置参数
3001	%1: Only works in %2 state	%1: 只能在%2 状态下生效	检查控制器状态
3002	%1: Not works in %2	%1: 在%2 状态下无法生	检查控制器状态

	state	效	
3003	%1: Only works in Teach Mode	%1: 只能在示教模式下生效	检查控制器状态
3004	%1: Hardware is not ready	%1: 硬件未就绪	检查硬件
3005	%1: Command buffer is not ready (full or finished)	%1: 指令缓冲区未就绪 (已满或者发送已经结束)	内部错误 请联系技术支持
3006	%1: Parameter %2 = %3 is invalid	%1: 参数%2 = %3 是无效的	检查设置参数
3007	%1: Parameter %2 = %3 is invalid, valid value is between [%4,%5]	%1: 参数%2 = %3 是无效的, 有效值范围是 [%4, %5]	检查设置参数
3008	%1: Parameter struct is invalid	%1: 结构体参数是无效的	检查设置参数
3009	InitMotionController: Controller has been initialized	InitMotionController: 控制器已经初始化	内部错误 无需处理
300A	StartMotionController: Controller has not been initialized, call imf::InitMotionControll	StartMotionController: 控制器未初始化, 请先调用 imf::InitMotionControlle	内部错误 无需处理

	er first	r	
300B	StartMotionController: Controller has been started	StartMotionController: 控制器已经启动	内部错误 无需处理
300C	%1: Controller has not been started	%1: 控制器未启动	内部错误 无需处理
300D	StartMotionController: Robot model dll path is invalid	StartMotionController: 机器人模型动态库路径 无效	内部错误 升级或还原系统
300E	StartMotionController: Robot model dll dose not contain specific robot type %1	StartMotionController: 机器人模型动态库不包 含指定的机器人类型%1	内部错误 升级或还原系统
300F	StartMotionController: Cannot open %1 controller dll, system error info %2	StartMotionController: 无法打开%1 控制器动态 库, 系统错误信息%2	内部错误 升级或还原系统
3010	StartMotionController: Creating controller with specific type (real/virtual/simulated) fails, system error	StartMotionController: 创建指定控制器类 (实 物/虚拟/仿真) 型失败, 系统错误信息%1	内部错误 升级或还原系统

	info %1		
3011	StartMotionController: Creating %1 posix thread fails, system error info %2	StartMotionController: 创建%1 线程失败, 系统 错误信息%2	内部错误 请联系技术支持
3012	LogMotionDataOn: Motion log path %1 is invalid	LogMotionDataOn: 运 动日志路径%1 是无效的	内部错误 升级或还原系统
3013	Jog: Only works in Teach Mode and Servo On state	Jog: 只在示教模式并且 上伺服状态下生效	检查控制器状态
3014	Jog: Jog coordinate is different from current working coordinate	Jog: 点动的坐标系和当 前生效的坐标系不一致	检查 Jog 发送的坐 标系参数
3015	%1: Immediate motion command only works in Teach Mode, in Standby State and no immediate motion command is executing	%1: 立即运动指令只在 示教模式 (Teach Mode), 伺服准备 (Standby)状态下并且 没有正在执行的立即运 动指令才生效	检查立即指令发送 的条件
3016	%1: Immediate command only works	%1: 立即非运动指令只 在示教模式 (Teach	检查立即指令发送 的条件

	in Teach Mode	Mode)，初始化或伺服准备 (Initial/Standby) 状态下生效	
3017	%1: Immediate command only works in Teach Mode,	%1: 立即非运动指令只在示教模式 (Teach Mode)，初始化或伺服准备 (Initial/Standby) 状态下生效	检查立即指令发送的条件
3018	%1: Velocity %2 is out of range (%3,%4] with unit %5	%1: 速度%2 超出范围 (%3, %4], 单位是%5	检查指令参数
3019	%1: Blend radius %2 is out of range [%3,%4] with unit %5	%1: 过渡段半径%2 超出范围[%3, %4], 单位是%5	检查指令参数
301A	%1: Target point is not reachable, uuid = %2	%1: 目标点无法到达, UUID = %2	检查指令目标点
301B	%1: The first coordinate is system reserved, not allowing modification	%1: 第一个坐标系为系统保留的, 不允许修改	手动清除 无需处理
301C	%1: Only suitable for Real Controller Mode	%1: 只适用于控制器实物模式	内部错误 无需处理
301D	%1: Open dll %2 fails, %3	%1: 打开动态库%2 失	内部错误

		败, %3	升级或还原系统
301E	%1: %2 dll is not loaded	%1: %2 动态库没有加载	内部错误 升级或还原系统
301F	%1: %2 function is not registered, cannot find specified function	%1: %2 函数没有注册, 无法找到指定函数	内部错误 升级或还原系统

错误代码	错误信息	错误分析	解决办法
50000	Emergency stop button is pressed	急停按钮被按下	松开急停
50100	system IO [%1] is triggered	系统 IO[%1]信号触发	
50101	system IO [%1] is triggered	系统 IO[%1]信号触发	
50102	system IO [%1] is triggered	系统 IO[%1]信号触发	

错误代码	错误信息	错误分析	解决办法
20000	undefined data type %1	未定义的数据类型 %1	内部错误, 重启后无法解决, 联系技术支持

20001	no project has been loaded	尚未加载任何工程	请先打开一个工程
21000	mysql exec error %1	MySQL 执行出错 %1	数据库内部错误, 重启后无法解决, 联系技术支持
21001	mysql data format error, missing col %1 or data type not compatible	数据库表格式错误, 缺少 %1 列, 或者其数据格式不兼容	数据库内部错误, 重启后无法解决, 联系技术支持
21002	get failed, id %1 not exist in %2	读取表格%2 错误, id %1 不存在	数据库内部错误, 重启后无法解决, 联系技术支持
21003	set key %1 of table %2 failed, must in range([%3, %4])	根据键 %1 更新表格 %2 失败, 键必须在 [%3, %4]内	数据库内部错误, 重启后无法解决, 联系技术支持
21004	set key %1 of table %2 failed, address out of bounds: offset + size = %3, size of target = %4	根据键 %1 更新表格 %2 失败, 地址越界: offset + size = %3, 结构体大小为 %4	数据库内部错误, 重启后无法解决, 联系技术支持
21005	set key %1 of table %2 failed, key not exist	根据键 %1 更新表格 %2 失败, 键不存在	数据库内部错误, 重启后无法解决, 联系

			技术支持
21006	error loading project %1, not exist	加载工程 %1 失败，不存在	数据库内部错误，重启后无法解决，联系技术支持
21007	error loading project %1, missing main	加载工程 %1 失败，缺失 main	数据库内部错误，重启后无法解决，联系技术支持
21008	user data table %1 not found	用户数据表 %1 不存在	数据库内部错误，重启后无法解决，联系技术支持
22000	modbus error %1	modbus 报警 %1	通讯错误，重启后无法解决，联系技术支持
30000	system init ...	系统初始化中 ...	无需处理
30001	system init timeout, over %1 s	系统初始化超时，超过 %1 s	无需处理
30002	unknown motion mode %1, real will be used.	未知的工作模式 %1，默认采用 real 模式	配置文件错误，重启后无法解决，联系技术支持
30003	no ecat config file, init abort	无法找到 ecat 的配置文件，初始化终止	配置文件错误，重启后无法解决，联系技术支持

31000	api calling error %1	控制器调用接口失败 %1	调用 API 失败, 重启后无法解决, 联系技术支持
31001	motion internal error %1	控制器内部错误: %1	重启后无法解决, 联系技术支持
32000	command %1, missing keyword %2	%1 指令缺失关键字%2	指令格式错误, 检查程序
32001	command %1, unexpected value of keyword %2	%1 指令,对应关键字 %2 的类型错误	指令格式错误, 检查程序
32002	command %1, invalid value of keyword %2	%1 指令,对应关键字 %2 的值非法	指令格式错误, 检查程序
32003	unknown robot command: %1	无法识别的机器人指令: %1	指令格式错误, 检查程序
40000	%1: error parsing IO variable(%2)	%1: 解析 IO 符号(%2)错误	指令格式错误, 检查程序
40001	%1: error parsing String(%2)	%1: 解析字符串(%2)错误	指令格式错误, 检查程序
40002	%1: error parsing operator(%2)	%1: 解析操作符(%2)错误	指令格式错误, 检查程序
40003	%1: error parsing number(%2)	%1: 解析数值(%2)错误	指令格式错误, 检查程序

40004	%1: error parsing integer(%2)	%1: 解析整数(%2)错误	指令格式错误, 检查 程序
40005	%1: error parsing system variable(%2)	%1: 解析系统变量(%2)错 误	指令格式错误, 检查 程序
40006	%1: unbalanced multi-line comment	%1: 多行注释不完整	指令格式错误, 检查 程序
41000	%1: invalid symbol(%2) in expression	%1: 表达式中包含非法 符号 (%2)	指令格式错误, 检查 程序
41001	%1: unknown built-in func %2	%1: 未知的内建函数 %2	指令格式错误, 检查 程序
41002	%1: expression terminated unexpectedly	%1: 表达式异常终止	指令格式错误, 检查 程序
41003	%1: unbalanced parenthesis in expression	%1: 表达式括号不匹配	指令格式错误, 检查 程序
41004	%1: invalid ++/-- in expression	%1: ++/-- 操作符非法	指令格式错误, 检查 程序
41005	%1: undeclared symbol %2	%1: 未定义的符号 %2	指令格式错误, 检查 程序
41100	%1: invalid statement declaration with %2	%1: 非法的语句申明 %2	指令格式错误, 检查 程序

41101	%1: %2 statement missing symbol %3	%1: %2 语句缺失符 号 %3	指令格式错误, 检查 程序
41102	%1: %2 statement missing keyword %3	%1: %2 语句缺失关键 字 %3	指令格式错误, 检查 程序
41180	%1: break/continue only works with in for/while	%1: break/continue 只 在 for/while 中使用	指令格式错误, 检查 程序
41181	%1: for statement loop variable not left-value	%1: for 语句循环变量非 左值, 无法赋值	指令格式错误, 检查 程序
41182	%1: assignment left term not left-value	%1: 赋值语句左边表达 式非左值, 无法赋值	指令格式错误, 检查 程序
41200	%1: %2 all robot cmd admits only one output term	%1: %2 所有机器人指令 只允许有一个输出项	指令格式错误, 检查 程序
41201	%1: %2 output term must be placed in the end for all robot cmd	%1: %2 输出项必须在指 令最后申明	指令格式错误, 检查 程序
41202	%1: %2 missing symbol %3	%1: %2 缺失符号 %3	指令格式错误, 检查 程序
41203	%1: %2 missing keyword %3	%1: %2 缺失键 %3	指令格式错误, 检查 程序
41204	%1: %2 incomplete	%1: %2 键值对不完整	指令格式错误, 检查

	key-value pair detected		程序
41205	%1: %2 key-value pair count check fail, %3 required	%1: %2 键值对数目检查 失败, 需求 %3 对	指令格式错误, 检查 程序
41300	%1: program declartion missing keyword %2	%1: 程序申明缺失关键 字 %2	指令格式错误, 检查 程序
41301	%1: program declartion missing symbol %2	%1: 程序申明缺失符 号 %2	指令格式错误, 检查 程序
41302	%1: program declaration missing name	%1: 程序申明缺失名字	指令格式错误, 检查 程序
41303	%1: function name (%2) differs from file name (%3)	%1: 函数名称(%2)和文件 名称不匹配(%3)	指令格式错误, 检查 程序
41304	%1: function declaration, missing return type	%1: 函数申明缺失返回 值	指令格式错误, 检查 程序
41305	%1: function declaration, invalid return type %2	%1: 函数申明返回值类 型 (%2) 非法	指令格式错误, 检查 程序
41306	%1: function	%1: 函数申明输入变量	指令格式错误, 检查

	declaration, duplicate argument %2	重复 %2	程序
41307	%1: function declaration, invalid argument name %2	%1: 函数申明输入参 数 %2 名称非法	指令格式错误, 检查 程序
41308	%1: function declaration, invalid argument type %2	%1: 函数申明非法的参 数类型 %2	指令格式错误, 检查 程序
41309	%1: wrong argument number: provided = %2, expected = %3	%1: 函数输入参数个数 检查失败: 实际 = %2, 预期 = %3	指令格式错误, 检查 程序
41304	%1: function argument type check failure	%1: 函数输入参数类型 检查失败	指令格式错误, 检查 程序
42000	%1: operands type check fail %2	%1: 操作数类型检查失 败 %2	指令格式错误, 检查 程序
42001	%1: division by 0, %2	%1: 除 0 错误, %2	指令格式错误, 检查 程序
42002	%1: invalid operand value %2	%1: 操作数值非法 %2	指令格式错误, 检查 程序
42003	%1: symbol %2 not exist	%1: 符号 %2 不存在	指令格式错误, 检查 程序
42004	%1: index (%2) exceeds	%1: 数组下标 %2 越	指令格式错误, 检查

	array bounds %3	界 %3	程序
42005	%1: field (%2) not exist in struct	%1: 结构体中不包含 域 %2	指令格式错误, 检查 程序
42100	%1: function %2 arg count check failure, %3 provided, %4 needed	%1: 函数 %2 输入参数 个数检查失败, 需 要 %3, 实际 %4	指令格式错误, 检查 程序
42101	%1: function %2 with the %3-th input null	%1: 函数 %2 的第%3 个 参数为空	指令格式错误, 检查 程序
42102	%1: function %2 division by 0	%1: 函数 %2 除 0 错误	指令格式错误, 检查 程序
42103	%1: function %2 arg type check failiure (%3)	%1: 函数 %2 参数类型 检查失败 (%3)	指令格式错误, 检查 程序
42104	%1: function %2 arg range check failiure %3	%1: 函数 %2 参数类型 范围检查失败 %3	指令格式错误, 检查 程序
42200	%1: %2 passing array as parameter is forbidden, key = %3	%1: %2 禁止传递数组参 数, 键值为 %3	指令格式错误, 检查 程序
42201	%1: %2 group and index of DIO must be integers %3	%1: %2 DIO 的组和编号 都必须是整数 %3	指令格式错误, 检查 程序
42202	%1: %2 index of AIO must be integers %3	%1: %2 AIO 的编号都必 须是整数 %3	指令格式错误, 检查 程序

42203	%1: %2, key %3 must be %4 type	%1: %2 的键值对 %3 必须是 %4 类型	指令格式错误, 检查程序
42300	%1: unknown struct field %2	%1: 未知的系统结构体域 %2	指令格式错误, 检查程序
42301	%1: manipulation of struct field %2 is forbidden	%1: 禁止操作结构体域 %2	指令格式错误, 检查程序
42302	%1: IO variable %2 out of bounds	%1: IO 变量 %2 越界	指令格式错误, 检查程序
42303	%1: undefined coord type %2	%1: 未定义坐标系类型 %2	指令格式错误, 检查程序
43000	file %1 not exist	目标程序(%1)不存在	程序不存在, 检查程序是否存在
43001	duplicate command name %1, last found in %2	指令名称(%1)重复出现, 最后一次在 %2 中	指令格式错误, 检查程序
43002	duplicate command type %1, last found in %2	指令类型(%1)重复出现, 最后一次在 %2 中	指令格式错误, 检查程序
43003	execute %1 is forbidden, type = FUNCTION	禁止执行 %1, 类型为 FUNCTION	指令格式错误, 检查程序

43004	program %1, start line %2 exceeds limit	禁止执行 %1, 启动 行 %2 超限	启动行号超限
60000	invalid registration file	注册文件无效	注册文件错误, 重启 后无法解决, 联系技 术支持
60001	invalid registration code	注册码无效	注册文件错误, 重启 后无法解决, 联系技 术支持
61000	app id = %1 is registered, but cannot be loaded	工艺已经注册, 但是无 法加载, ID = %1	程序文件缺失, 重启 后无法解决, 联系技 术支持
61001	apps %1 and %2 have duplicate id = %3	工艺 %1 和 %2 的内部 ID 冲突, ID = %3	工艺插件错误, 重启 后无法解决, 联系技 术支持
61002	apps %1 and %2 accepts common command %3	工艺 %1 和 %2 指令集 冲突, 指令类型: %3	工艺插件错误, 重启 后无法解决, 联系技 术支持