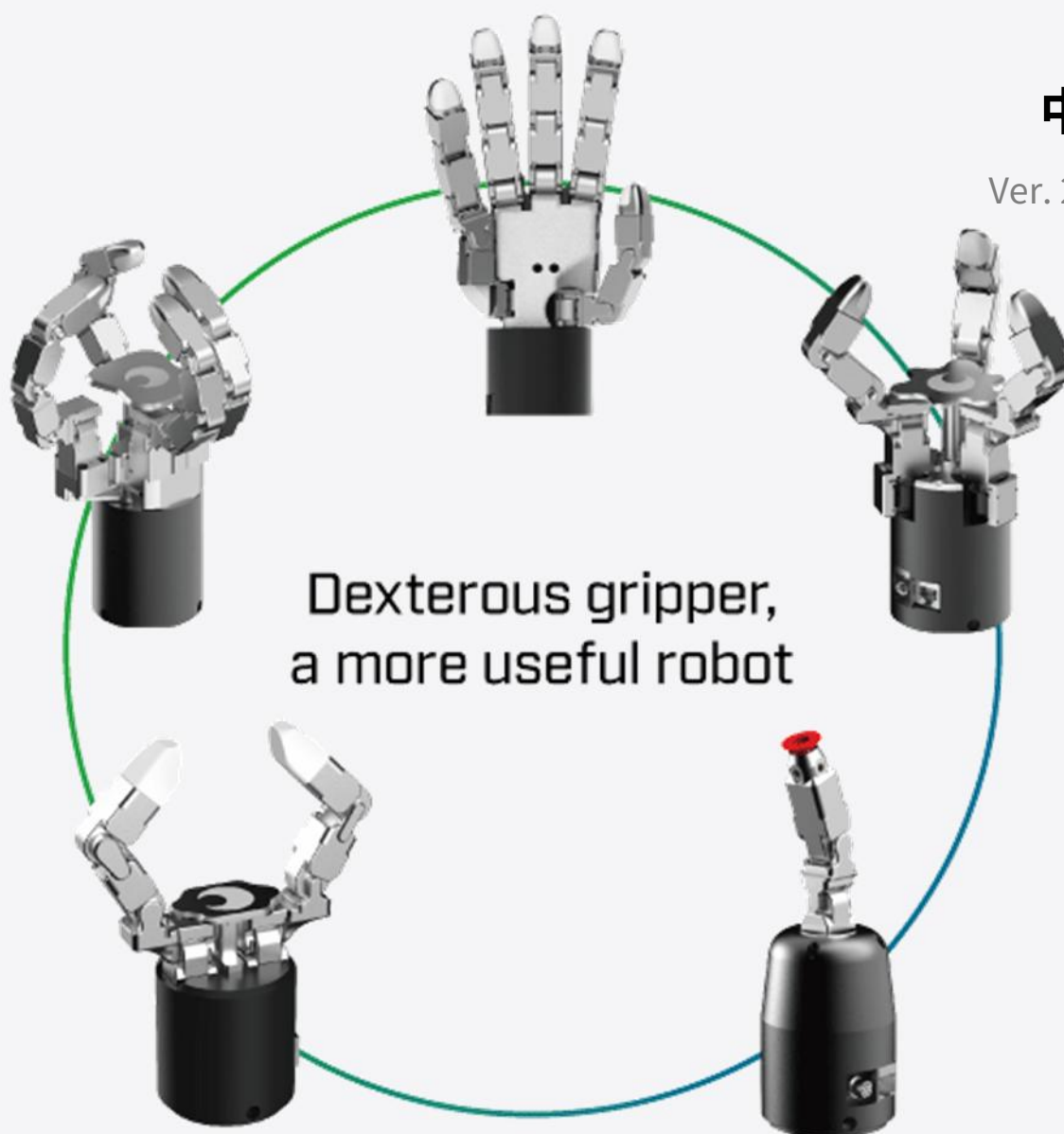


Delto Gripper

Manager 及 SDK 使用说明书

中文

Ver. 2.0.0



INFORMATION

版权

本使用说明书由 Tesollo Inc. 编写，所有版权归 Tesollo Inc. 所有，禁止基于本说明书进行复制、再生产并向第三方分发等行为。

使用说明书修订相关通知

为实现技术改进，本说明书可能会进行修订，特此预先说明。

Manual Version: 2.0.0 中文 (2026.04)

© 2026 Tesollo Inc. All rights reserved

Tesollo Corporation

总部：仁川广域市延寿区 Convensia-daero 165, 26 层

分公司：京畿道光明市 Iljik-ro 43 GIDC Tower, A 栋 26 层

Support T: +82 2 6914 6620

E: support@tesollo.com

H: <http://www.tesollo.com>

目录

第 1 章 说明书信息	12
第 2 章 注意事项	12
第 3 章 入门	13
3.1 安装 DG Manager	13
3.1.1 安装文件位置	13
3.1.2 安装 DG Manager	14
3.2 删除 DG Manager	18
第 4 章 通信连接界面	20
4.1 登录界面	20
4.1.1 界面构成	20
4.1.2 用户模式连接	22
4.1.3 开发者模式连接	26
4.1.4 连接确认	28
4.2 主界面	30
4.2.1 界面构成	30
第 5 章 监控界面	32
5.1 界面构成	32
第 6 章 动作界面	34
6.1 抓取模式	34
6.1.1 界面构成	34
6.1.2 抓取操作	35

6.1.3 抓取数据设置	35
6.1.4 位置控制模式设置	37
6.2 位置控制	38
6.2.1 界面构成	38
6.2.2 位置控制动作	39
6.2.3 示教模式	40
6.3 增益设置	41
6.3.1 界面构成	41
6.3.2 增益模式设置	42
6.3.3 增益设置	43
6.4 电流设置	44
6.4.1 界面构成	44
6.4.2 电流模式设置	44
6.4.3 电流设置	45
第 7 章 配方设置界面	46
7.1 界面构成	46
7.2 位姿配方	47
7.2.1 界面构成	47
7.2.2 位姿配方设置方法	48
7.2.3 位姿配方动作使用方法	49
7.3 增益配方	49
7.3.1 界面构成	50
7.3.2 增益配方设置方法	51
7.3.3 增益配方动作使用方法	52

7.4 抓取配方	52
7.4.1 界面构成	53
7.4.2 抓取配方设置方法	54
7.4.3 抓取配方动作使用方法	55
7.5 混合配方	56
7.5.1 界面构成	56
7.5.2 混合配方设置方法	57
7.5.3 混合配方动作使用方法	59
第 8 章 附加功能界面	60
8.1 设置	60
8.1.1 界面构成	60
8.1.2 Boot Recipe	61
8.1.3 Joint Offset	62
8.1.4 Joint Inpose	64
8.1.5 Moving Inpose	65
8.1.6 Low Pass Filter	66
8.1.7 Torque Limit Mode.....	67
8.1.8 Backup Settings / EEPROM Write	68
8.1.9 Restore Settings	69
8.2 工具	70
8.2.1 界面构成	70
8.2.2 Fingertip Sensor	71
8.2.3 GPIO.....	74
8.2.4 Firmware Download	75

8.2.5 Set IP	77
8.3 Spy	78
8.4 主题	79
第 9 章 SDK 数据结构	79
9.1 Define	79
9.2 枚举类型	81
9.2.1 enum DG_RESULT	81
9.2.2 enum DG_MODEL	81
9.2.3 enum BLEND_MOTION_STATUS	82
9.2.4 enum COMMUNICATION_MODE	82
9.2.5 enum CONTROL_MODE	82
9.2.6 enum GAIN_MODE	82
9.2.7 enum DEVELOPER_MODE_COMMAND	83
9.2.8 enum DEVELOPER_MODE_RECEIVED_DATA_TYPE	83
9.2.9 enum DG_GRASP_MODE	84
9.2.10 enum DG_GRASP_OPTION	86
9.2.11 enum DG_SENSOR_TYPE	86
9.3 结构体	87
9.3.1 struct ReceivedGripperData	87
9.3.2 struct RecipeBlendData	88
9.3.3 struct GripperSystemSetting	88
9.3.4 struct GripperSetting	89
9.3.5 struct RecipePoseData	90
9.3.6 struct RecipeGainData	90

9.3.7 struct RecipeGraspData	91
9.3.8 struct ReceivedFingertipSensorData	91
9.3.9 struct ReceivedGPIOData	92
第 10 章 系统函数	93
10.1 系统设置函数	93
10.1.1 SetGripperSystem	93
10.1.2 SetGripperOption	94
10.1.3 ConnectToGripper	95
10.1.4 DisconnectToGripper	96
10.1.5 SystemStart	97
10.1.6 SystemStop	97
10.1.7 SetIp	98
10.1.8 SetJointEncoderZero	99
10.2 回调	99
10.2.1 CallbackForOnConnected	99
10.2.2 CallbackForOnDisconnected	100
10.2.3 CallbackForOnReceivedGripperData	101
10.2.4 CallbackForOnCommunicationPeriod	101
10.2.5 CallbackForOnReceivedFingertipSensorData	103
10.2.6 CallbackForOnReceivedGPIOData	104
10.3 其他系统函数	105
10.3.1 SetLowPassFilterAlpha	105
10.3.2 SetBootMode	106
10.3.3 SetGPIOOutput	107

10.3.4 SetGPIOOutputAll	108
10.3.5 SetTorqueLimitMode	108
10.3.6 SetBootRecipe	110
10.3.7 EEPROMWrite	111
10.3.8 BackupRecipeData	112
10.3.9 RestoreRecipeData	113
10.4 数据查询	114
10.4.1 GetReceivedGripperData	114
10.4.2 GetCommunicationPeriod	115
10.4.3 GetReceivedFingertipSensorData	115
10.4.4 GetReceivedGPIOData	116
10.4.5 GetLibraryVersion	117
第 11 章 动作设置函数	118
11.1 抓取设置函数	118
11.1.1 SetGraspData	118
11.1.2 SetGraspForce	119
11.2 P 增益设置	121
11.2.1 SetJointGainP	121
11.2.2 SetJointGainPFinger	122
11.2.3 SetJointGainPAll	123
11.2.4 SetJointGainPBase	124
11.3 D 增益设置	125
11.3.1 SetJointGainD	125
11.3.2 SetJointGainDFinger	126

11.3.3 SetJointGainDAll	126
11.3.4 SetJointGainDBase	127
11.4 I 增益设置	129
11.4.1 SetJointGainI	129
11.4.2 SetJointGainIFinger	131
11.4.3 SetJointGainIAll	132
11.4.4 SetJointGainIBase	133
11.5 PID 增益设置	134
11.5.1 SetControlPIDMode	134
11.5.2 SetJointGainPID	135
11.5.3 SetJointGainPIDFinger	136
11.5.4 SetJointGainPIDAll	137
11.5.5 SetJointGainPIDAllEqual	138
11.5.6 SetJointGainPIDBase	139
11.6 动作时间设置	140
11.6.1 SetMotionTimeJoint	140
11.6.2 SetMotionTimeFinger	141
11.6.3 SetMotionTimeAll	142
11.6.4 SetMotionTimeAllEqual	143
11.6.5 SetMotionTimeBase	144
11.7 位置控制模式设置	145
11.7.1 SetPositionModeJoint	145
11.7.2 SetPositionModeFinger	146
11.7.3 SetPositionModeAll	147

11.7.4 SetPositionModeBase	148
11.8 指尖传感器设置	149
11.8.1 SetFingerTipDataZero	149
第 12 章 配方函数	149
12.1 配方位姿函数	150
12.1.1 UpdateRecipeCurrentPoseData	150
12.1.2 UpdateRecipeTargetPoseData	150
12.1.3 LoadRecipePoseData	151
12.2 配方增益函数	153
12.2.1 UpdateRecipeGainData	153
12.2.2 LoadRecipeGainData	154
12.3 配方抓取函数	156
12.3.1 UpdateRecipeGraspData	156
12.3.2 LoadRecipeGraspData	157
12.4 配方混合函数	157
12.4.1 UpdateBlendJoint	158
12.4.2 ClearMoveBlendJoint	158
12.4.3 AddMoveBlendJoint	160
12.4.4 SetMoveBlendJoint	161
12.4.5 StartMoveBlendJoint	163
12.4.6 StopMoveBlendJoint	164
第 13 章 动作函数	165
13.1 ManualTeachMode	165
13.2 StartGraspMotion	166

13.3 位置控制	167
13.3.1 MoveJoint	167
13.3.2 MoveJointFinger	168
13.3.3 MoveJointAll	169
13.3.4 MoveServoJoint	170
13.3.5 MoveJointBase	171
13.4 电流控制	172
13.4.1 SetCurrentControlMode	172
13.4.2 SetTargetCurrentJoint	173
13.4.3 SetTargetCurrentFinger	174
13.4.4 SetTargetCurrentAll	175
13.4.5 SetTargetCurrentBase	176
附录 1 术语表及功能说明	177
附录 2 错误列表	182

第 1 章 说明书信息

- “开发者模式, J1” 等本说明书中使用的专用术语、轴缩写及其他术语, 已收录于“[附录 1 术语表及功能说明](#)”中。如有疑问, 请参考该附录。

- DG Manager 及 SDK 已在 Windows 10 x64、Windows 11 x64、Ubuntu 24.04 x64、Ubuntu 22.04 x64、C++20 环境下完成测试, 在类似的 OS 环境中也可运行。以下为测试所用 PC 规格。

- * CPU: AMD Ryzen 5 5600 6-Core Processor

- * RAM: 16GB

- * Graphic: NVIDIA GeForce RTX 3060

- 用户模式的通信协议为“Modbus RTU”和“Modbus TCP”。开发者模式使用基于 TCP/IP 的专有协议。

第 2 章 注意事项

- 根据 PC 性能及网络环境, 程序可能无法稳定运行。

- DG Manager 可用于操作 TESOLLO 销售的 Delto Gripper 系列。但根据固件和夹爪型号, 支持的功能可能不同或不兼容, 因此使用 DG Manager 及 SDK 时请注意固件版本与型号。

- 若未按照本手册说明操作, 错误操作可能导致误动作。这可能造成夹爪损坏, 并且在某些情况下可能无法享受售后服务。

- 如需其他支持, 请联系 TESOLLO 技术支持邮箱。
(support@tesollo.com)

第 3 章 入门

3.1 安装 DG Manager

3.1.1 安装文件位置

3.1.1.1 Windows

Software > DGManager			
이름	수정한 날짜	유형	크기
 DGManagerInstaller.exe	2025-03-26 오후 7:35	응용 프로그램	261,119KB

在夹爪包装箱内随附的USB中，进入图示路径” Software– DGManagerWindows” 文件夹，可找到名为 “DGManagerInstaller_ver_x_x_x.exe (ver_x_x_x 为版本信息)” 的安装文件。执行该安装文件即可安装 DG Manager。

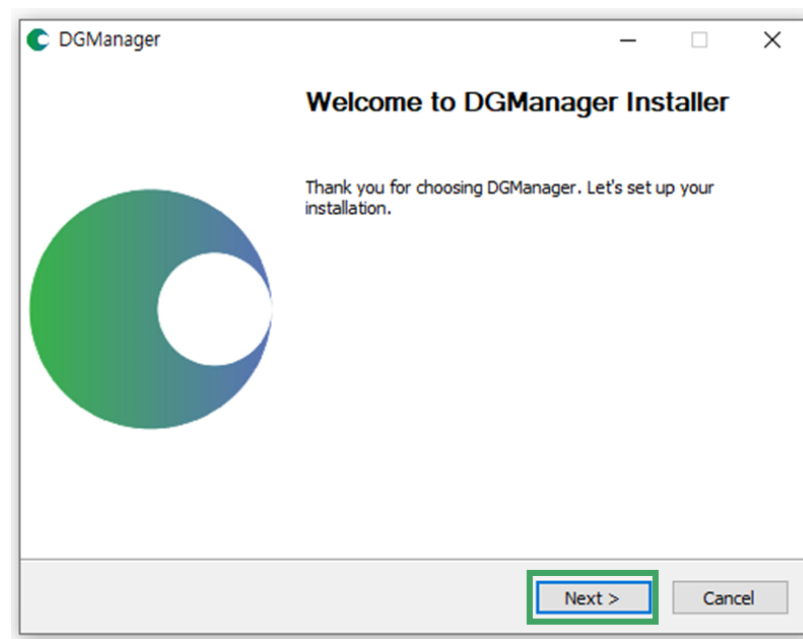
3.1.1.2 Ubuntu 24.04 / 22.04



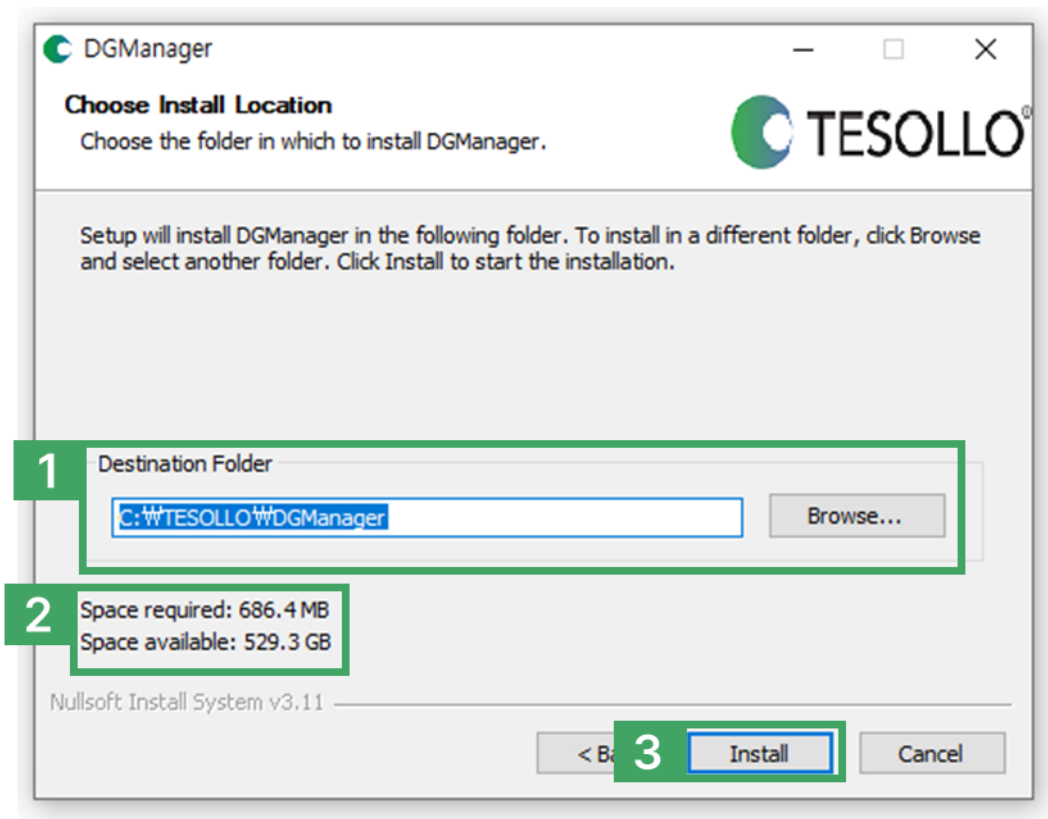
在夹爪包装箱内随附的 USB 中，进入图示路径 “Software - DGManager-Linux” 文件夹，可找到名为 “DGManager_linux_ver_x_x_x.zip (ver_x_x_x 为版本信息)” 的压缩包文件。解压该压缩包后即可运行 DGManager。

3.1.2 安装 DG Manager

3.1.2.1 Windows



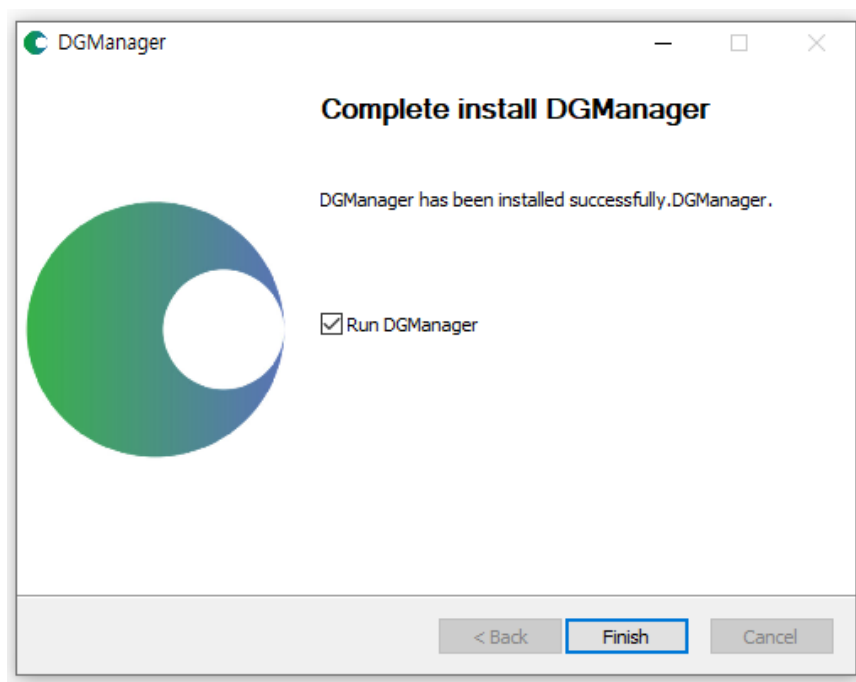
运行程序后，将显示如上图所示的安装界面。点击“Next>”进入下一界面。



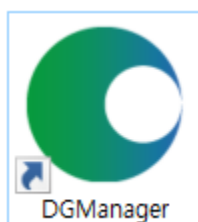
设置文件路径后开始安装。安装前请确认存储空间是否充足。

项目

编号	名称	说明
1	Destination Folder Browse	设置文件安装的文件夹路径
2	容量确认	显示文件容量和安装所需空间
3	Install	开始文件安装



文件正常安装完成后，将显示如上图所示的界面，并在桌面生成 DGManager.exe 快捷方式。



3.1.2.2 Ubuntu 24.04 / 22.04

示例

```
bash
# 解压
unzip DGManager_linux_ver_1_2_0.zip
# 目录进入
cd DGManager-linux-x64/
# 依赖项安装及执行权限授予后执行
. install_run.sh
# 仅授予执行权限后运行（无需安装库）
. run.sh
# 程序执行
./DGManager
```

解压该压缩包后 DGManager 可以运行。

如需安装依赖包并授予执行权限，请先运行 install_run.sh 脚本。

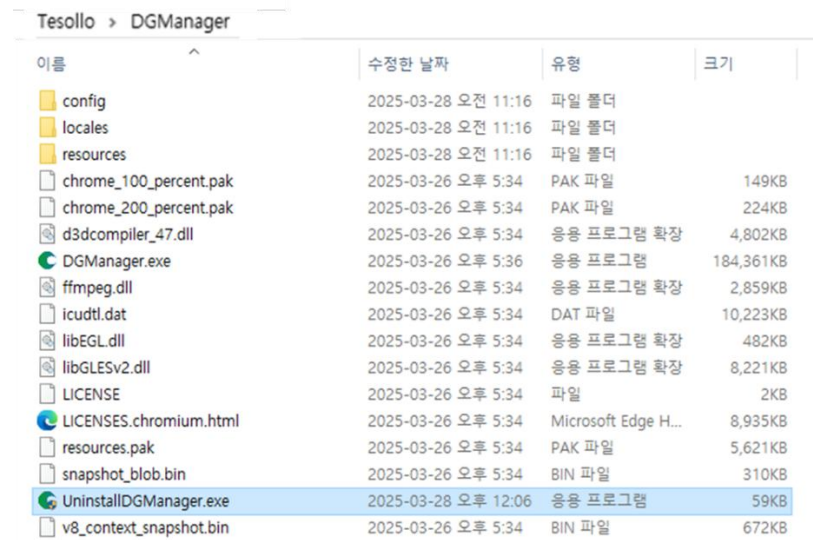
（此过程需要互联网连接。）

如无需安装库文件，仅授予执行权限并运行程序，请执行 run.sh 脚本。

如仅需执行程序，可直接运行 ./DGManager 文件。

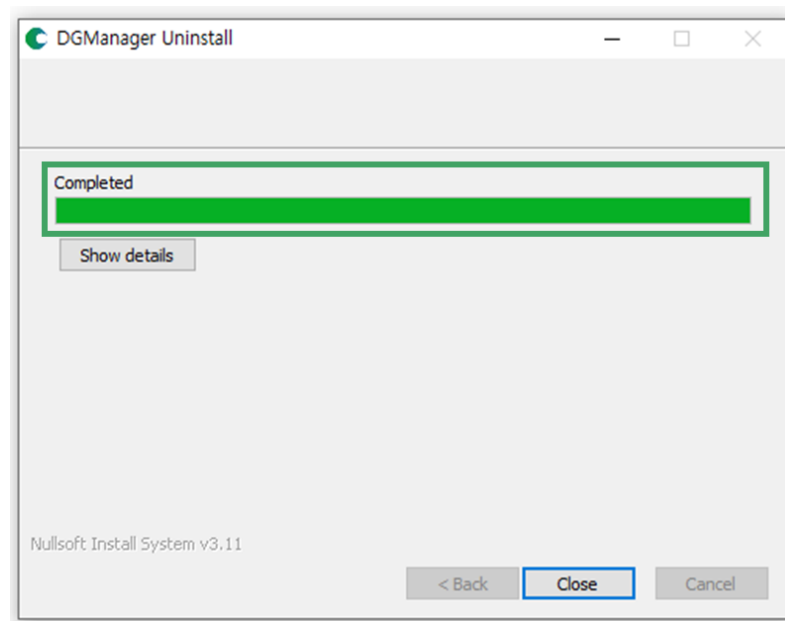
3.2 删除 DG Manager

3.2.1.1 Windows



이름	수정된 날짜	유형	크기
config	2025-03-28 오전 11:16	파일 폴더	
locales	2025-03-28 오전 11:16	파일 폴더	
resources	2025-03-28 오전 11:16	파일 폴더	
chrome_100_percent.pak	2025-03-26 오후 5:34	PAK 파일	149KB
chrome_200_percent.pak	2025-03-26 오후 5:34	PAK 파일	224KB
d3dcompiler_47.dll	2025-03-26 오후 5:34	응용 프로그램 확장	4,802KB
DGManager.exe	2025-03-26 오후 5:36	응용 프로그램	184,361KB
ffmpeg.dll	2025-03-26 오후 5:34	응용 프로그램 확장	2,859KB
icudtl.dat	2025-03-26 오후 5:34	DAT 파일	10,223KB
libEGL.dll	2025-03-26 오후 5:34	응용 프로그램 확장	482KB
libGLESv2.dll	2025-03-26 오후 5:34	응용 프로그램 확장	8,221KB
LICENSE	2025-03-26 오후 5:34	파일	2KB
LICENSES.chromium.html	2025-03-26 오후 5:34	Microsoft Edge H...	8,935KB
resources.pak	2025-03-26 오후 5:34	PAK 파일	5,621KB
snapshot_blob.bin	2025-03-26 오후 5:34	BIN 파일	310KB
UninstallDGManager.exe	2025-03-28 오후 12:06	응용 프로그램	59KB
v8_context_snapshot.bin	2025-03-26 오후 5:34	BIN 파일	672KB

进入文件安装路径后，可找到名为“UninstallDGManager.exe”的文件。执行该文件后，DG Manager 程序将开始卸载。



文件卸载完成后上图如图所示的界面输出。

3.2.1.2 Ubuntu 24.04 / 22.04

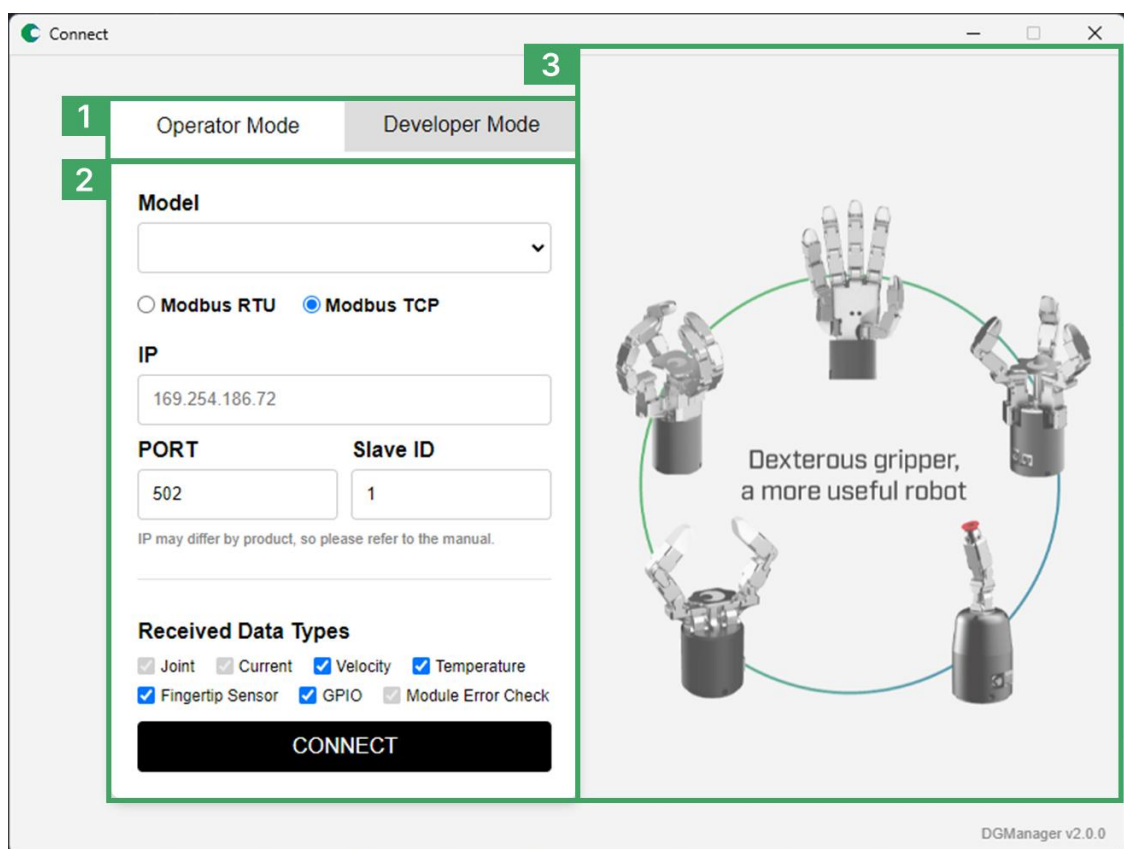
该文件以 ZIP 格式提供，因此在 Linux 中解压后，只需删除相应目录即可移除所有文件。

第 4 章 通信连接界面

4.1 登录界面

4.1.1 界面构成

功能说明



Delto Gripper 支持用户模式和开发者模式。

用户模式(Operator Mode)

该模式使用产品内部控制器进行控制。支持 Modbus Protocol，可通过修改寄存器映射表或使用提供的 UI 轻松控制夹爪。

使用用户模式及 SDK 时，可将“Torque limit mode”设为“off”，以使用“stall torque”。

开发者模式(Developer Mode)

该模式使用开发者自行开发的控制器。可接收夹爪各关节信息，并对各关节进行直接控制。

注意！在用户模式下将 Torque limit mode 设为 off，或在开发者模式下使用 stall torque 时，可能导致产品温度快速上升及产品损坏；发生故障时可能无法享受售后服务。

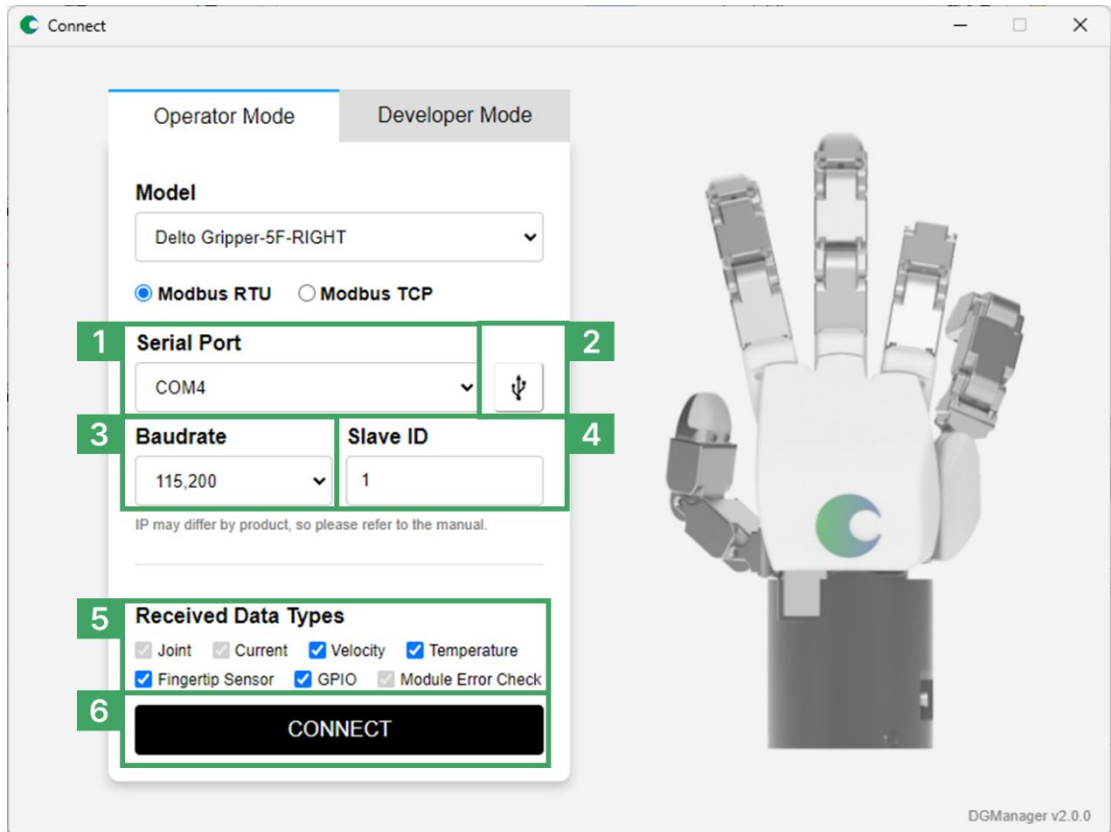
项目

编号	名称	说明
1	控制模式选择	控制模式 4.1.2 用户模式和 4.1.3 开发者模式中 选择
2	连接设置值输入	根据控制及通信方式输入所需的设置值
3	夹爪型号图像	选择夹爪型号图像显示

4.1.2 用户模式连接

4.1.2.1 Modbus RTU 连接设置

功能说明



Modbus RTU 连接示例

当要连接的夹爪型号为DG-3F-M、通信方式为Modbus RTU、连接端口为 COM8、Baudrate 为 115200、Slave ID为1 时，首先在 'Model' 中选择“Delto Gripper-3F-M”。随后选择“Modbus RTU”，点击扫描按钮，在“Serial Port”中选择“COM8”，在“Baudrate”中选择“115,200”，并在“Slave ID”中输入1。最后点击 CONNECT 按钮。通信正常连接后，将切换至主界面。

项目

编号	名称	说明
1	Serial Port	选择夹爪连接的串口
2	Scan	扫描当前连接的串口
3	Baudrate	选择夹爪设置的 Baudrate ☐ 115200
4	Slave ID	Modbus Slave ID 输入 ☐ Slave ID 范围: 1 - 255
5	Received Data Types	选择接收数据类型
6	CONNECT	使用设置值连接夹爪通信

参考

[10.1.1 SetGripperSystem](#)

[10.1.3 ConnectToGripper](#)

* 发生错误时，请参考“附录 2 错误列表”

4.1.2.2 Modbus TCP 连接设置

功能说明

Connect

Operator Mode Developer Mode

Model
Delto Gripper-5F-RIGHT

☐ Modbus RTU ☒ Modbus TCP

1 IP
169.254.186.72

2 PORT
502

3 Slave ID
1

IP may differ by product, so please refer to the manual.

4 Received Data Types
☒ Joint ☒ Current ☒ Velocity ☒ Temperature
☒ Fingertip Sensor ☒ GPIO ☐ Module Error Check

5 CONNECT

DGManager v2.0.0

Modbus TCP 连接示例

当要连接的夹爪型号为 DG-3F-M，设定的 IP 地址为 169.254.186.72、端口为 502、Slave ID 为 1 时，首先在 'Model' 中选择“Delto Gripper-3F-M”。随后选择“Modbus TCP”，并在下方各项目中输入相应值。在“IP”中输入“169.254.186.72”，在“PORT”中输入“502”，在“Slave ID”中输入“1”，最后点击“CONNECT”按钮。通信正常连接后，将切换至主界面。

项目

编号	名称	说明
1	IP	输入用于夹爪通信连接的 IP ☐ 夹爪默认 IP: ' 169.254.186.72' ☐ DG-5F-L 默认 IP: ' 169.254.186.73'
2	PORT	输入用于夹爪通信连接的端口 ☐ 端口范围: 1 - 65535 ☐ 夹爪默认端口: 502
3	Slave ID	Modbus Slave ID 输入 ☐ Slave ID 范围: 1 - 255
4	Received Data Types	选择接收数据类型
5	CONNECT	使用设置值连接夹爪通信

参考

[10.1.1 SetGripperSystem](#)

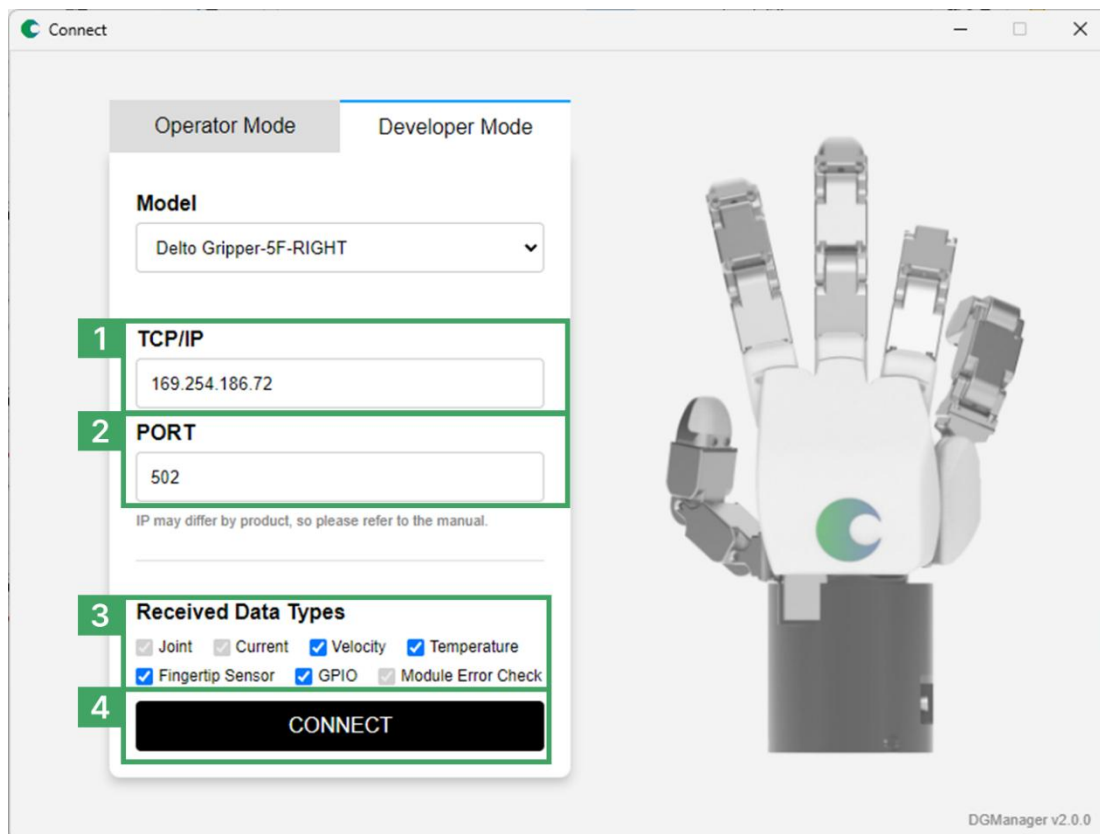
[10.1.3 ConnectToGripper](#)

* 发生错误时，请参考“附录 2 错误列表”

4.1.3 开发者模式连接

4.1.3.1 开发者模式 TCP/IP 设置输入

功能说明



开发者模式 TCP/IP 连接示例

当要连接的夹爪型号为 DG-3F-M、IP 为 169.254.186.72、端口为 502 时，首先在 'Model' 中选择 “Delto Gripper-3F-M”。

随后在 “TCP/IP” 中输入 “169.254.186.72”，在 “PORT” 中输入 502。之后在 Slave ID 栏中输入 1，最后点击 CONNECT 按钮完成连接。

项目

编号	名称	说明
1	TCP/IP	输入用于夹爪通信连接的 IP ☐ 夹爪默认 IP: ' 169.254.186.72' ☐ DG-5F-L 默认 IP: ' 169.254.186.73'
2	PORT	输入用于夹爪通信连接的端口 ☐ 端口范围: 1 ~ 65535 ☐ 夹爪默认端口: 502
3	Received Data Types	选择接收数据类型
4	CONNECT	使用设置值连接夹爪通信

参考

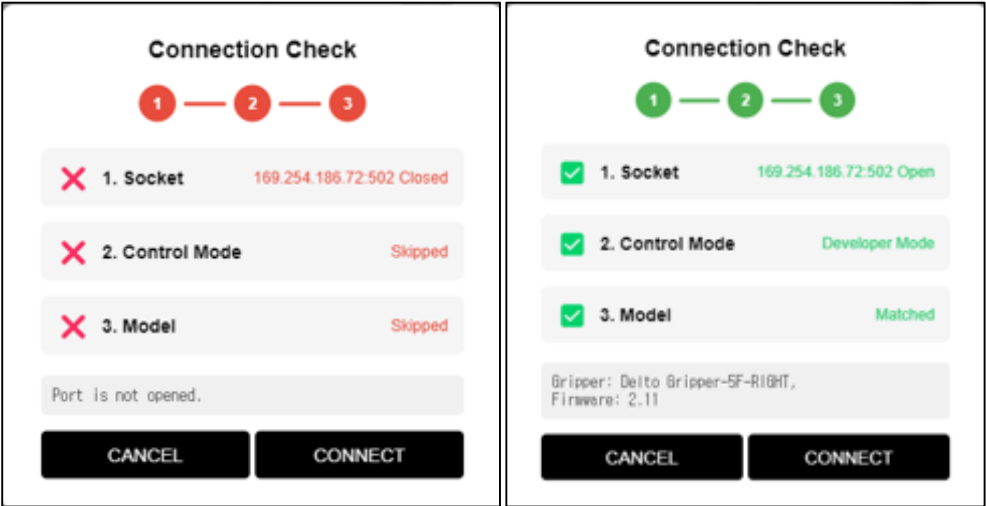
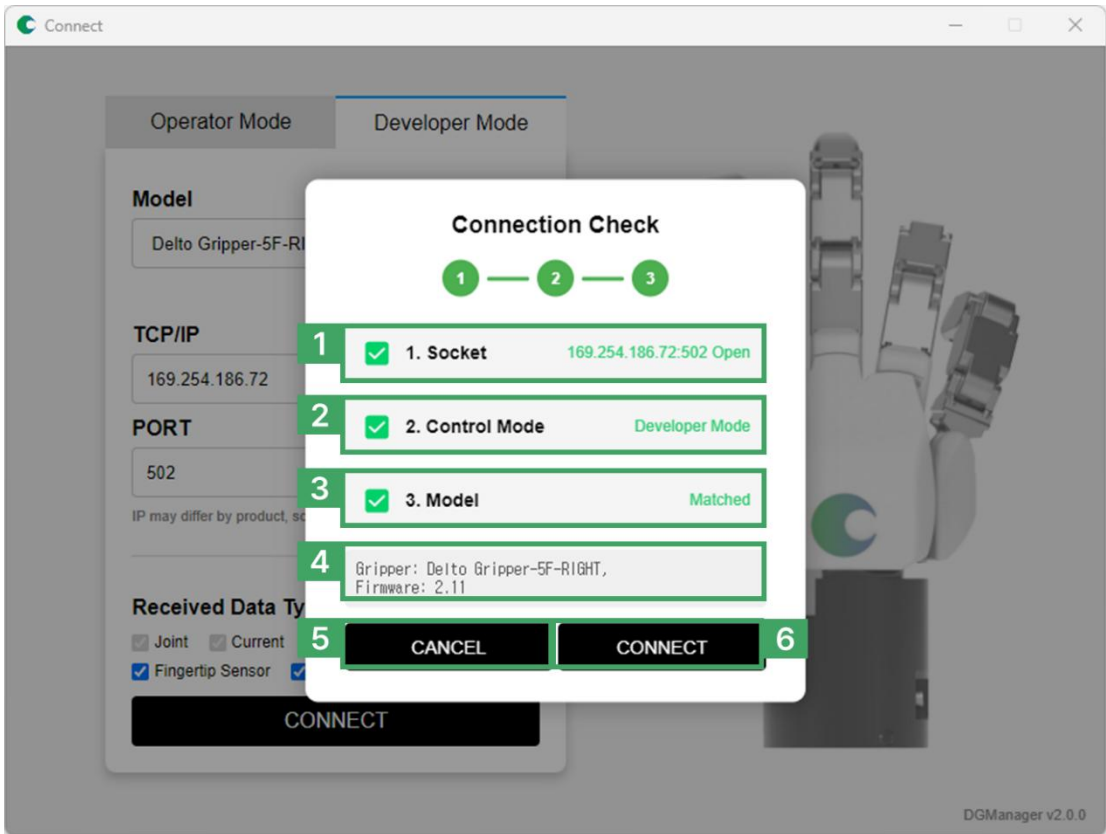
[10.1.1 SetGripperSystem](#)

[10.1.3 ConnectToGripper](#)

* 发生错误时，请参考“附录 2 错误列表”

4.1.4 连接确认

功能说明



连接确认是用于接入 Delto Gripper 前确认当前 Socket 状态、控制模式及型号信息是否正确的步骤。

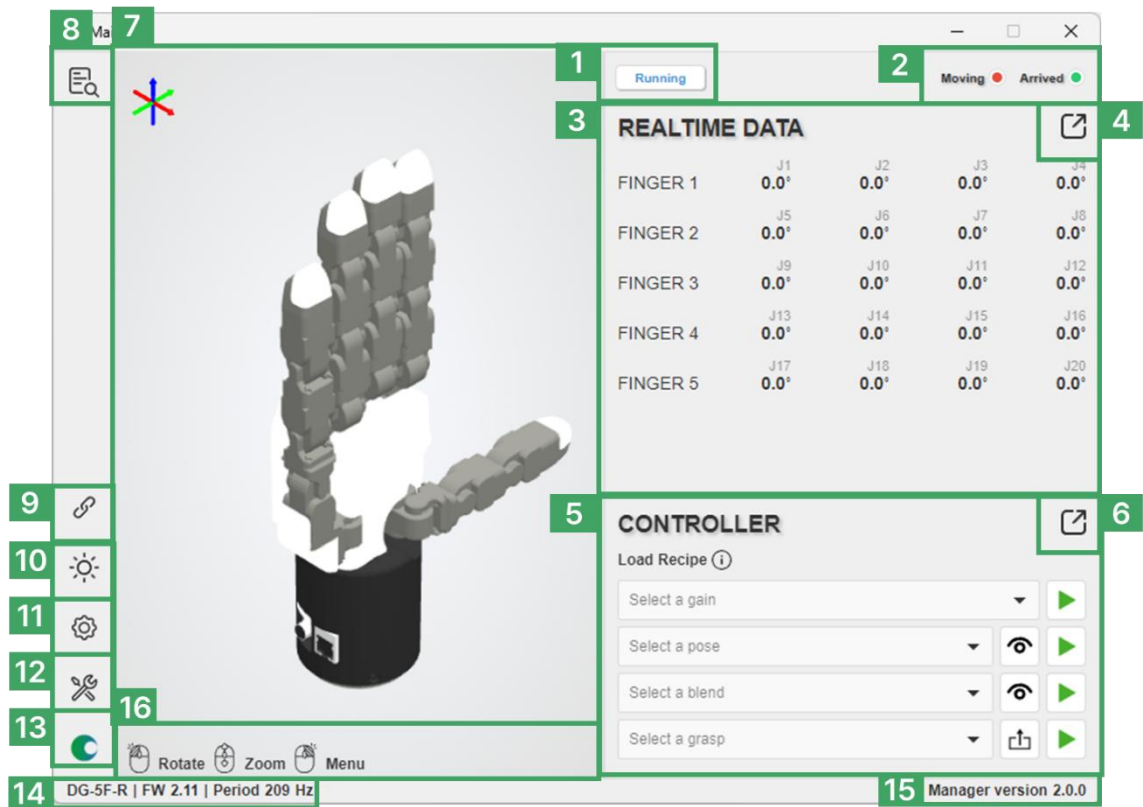
项目

编号	名称	说明
1	Socket	当前设备和夹爪之间 IP 网段确认
2	Control Mode	比较夹爪控制模式与 DGManager 选择的控制模式
3	Model	比较夹爪型号与 DGManager 中选择的型号
4	Log	按项目显示日志
5	CANCEL	当前窗口退出
6	CONNECT	夹爪连接尝试

4.2 主界面

4.2.1 界面构成

功能说明



主界面用于执行夹爪的全部功能。可启动或停止夹爪控制，并通过 3D 仿真查看实时角度信息。此外，还可通过配方预览并执行动作，并使用主题切换、断开连接、信息等附加功能。

在主界面中，可通过“夹爪系统开始/停止”项目轻松进行控制。在“Ready”状态下选择“Start”时，系统将被启用并准备接收各类命令；在“Running”状态下选择“Stop”时，夹爪动作将停止并结束控制。

项目

编号	名称	说明
1	夹爪系统开始/停止	☐ Ready: 夹爪等待状态 ☐ Running: 夹爪系统启用状态 ☐ Start: 夹爪系统启用执行 ☐ Stop: 夹爪系统停止执行
2	当前状态显示	☐ Moving : 当前夹爪正在移动时显示为绿色，停止时显示为红色。 ☐ Arrived : 目标值到达设定范围内时显示为绿色，否则显示为红色。
3	REAL-TIME	各关节的实时位置显示
4	监控界面	第 5 章监控界面参考
5	CONTROLLER	已保存的配方确认，选择配方执行
6	动作，配方界面	第 6 章动作界面 和 第 7 章配方设置界面 打开
7	仿真界面	接收当前夹爪位置并实时渲染 可通过鼠标拖拽关节进行控制
8	Spy 日志界面	8.3 Spy 日志参考
9	夹爪通信连接断开	夹爪通信连接断开
10	主题更改	浅色 / 深色主题之一更改
11	夹爪设置	夹爪 8.1 设置界面 打开
12	夹爪工具	8.2 工具参考
13	程序信息	打开程序信息显示窗口
14	夹爪信息	Model / Firmware Version / Communication Cycle
15	Manager 版本	当前管理器程序版本显示
16	仿真操作按钮	仿真界面操作及关节指定颜色

参考

第 5 章 监控界面

第 6 章 动作界面

第 7 章 配方设置界面

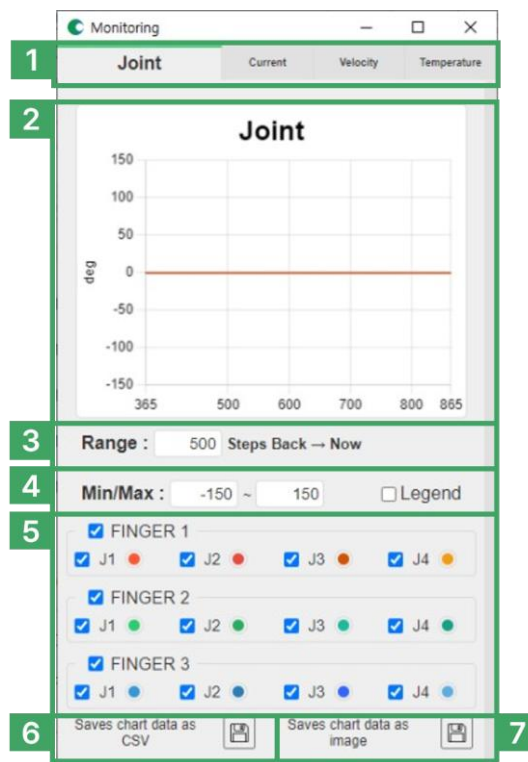
第 8 章 附加功能界面

* 发生错误时，请参考“附录 2 错误列表”

第 5 章 监控界面

5.1 界面构成

功能说明



监控界面可实时显示所选图表项目并保存数据。可选择关节、电流、速度、温度等项目以实时图表查看，并可设置图表的横轴范围（保存的数据数量）和纵轴范围。还可选择要显示在图表中的关节，并将当前图表数据保存为 CSV 文件或 PNG 图像。

项目

编号	名称	说明
1	图表项目选择	要显示的数据选择 ☐ Joint 关节 ☐ Current 电流 ☐ Velocity 速度 ☐ Temperature 温度
2	实时图表界面	将实时接收的数据以图表显示
3	Range	图表的 x 轴的显示数据数量设置
4	Min/Max	图表的 Y 轴的数据范围设置
5	关节选择及显示设置	图表上要显示的关节项选择
6	Saves chart data as CSV	图表当前数据 CSV 文件保存
7	Saves chart data as image	图表当前图像 PNG 文件保存

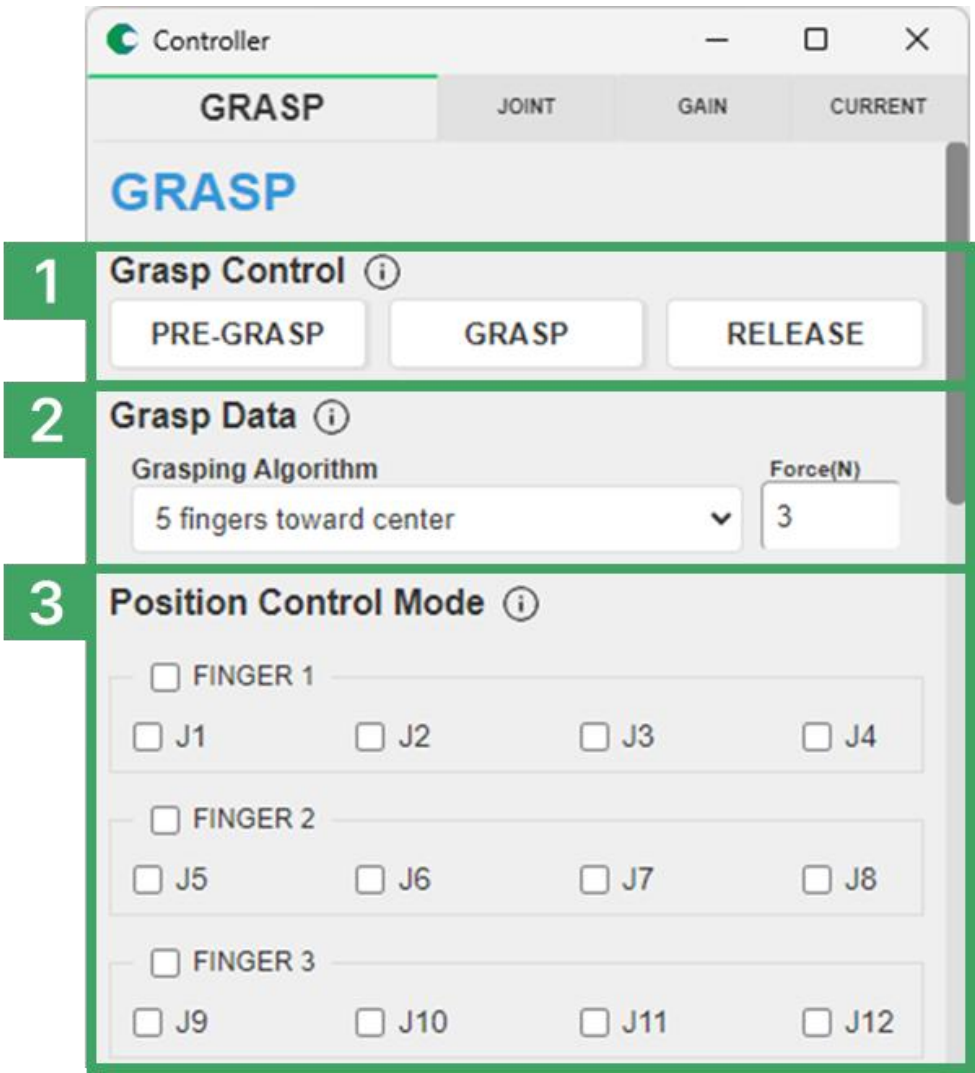
参考

DG-3F-B 不提供 Velocity 和 Temperature 数据

第 6 章 动作界面

6.1 抓取模式

6.1.1 界面构成



编号	名称	说明
1	Grasp Control	GRASP 及 RELEASE 执行
2	Grasp Data	抓取模式及抓取力度设置
3	Position Control Mode	设置各关节的位置控制模式

6.1.2 抓取操作

功能说明



该功能用于执行抓取及释放操作。

项目

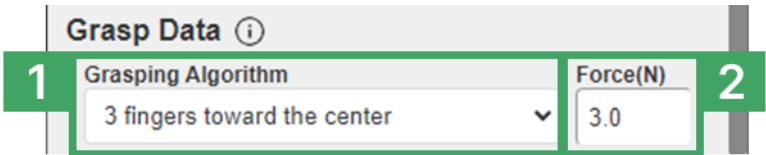
编号	名称	说明
1	PRE-GRASP	抓取的用于测试的示例位姿加载
2	GRASP	根据“6.1.3 抓取数据设置”中设定的值执行抓取
3	RELEASE	释放“6.2.2 位置控制动作”最后目标关节位置值为位置移动执行

参考

* 发生错误时，请参考“附录 2 错误列表”

6.1.3 抓取数据设置

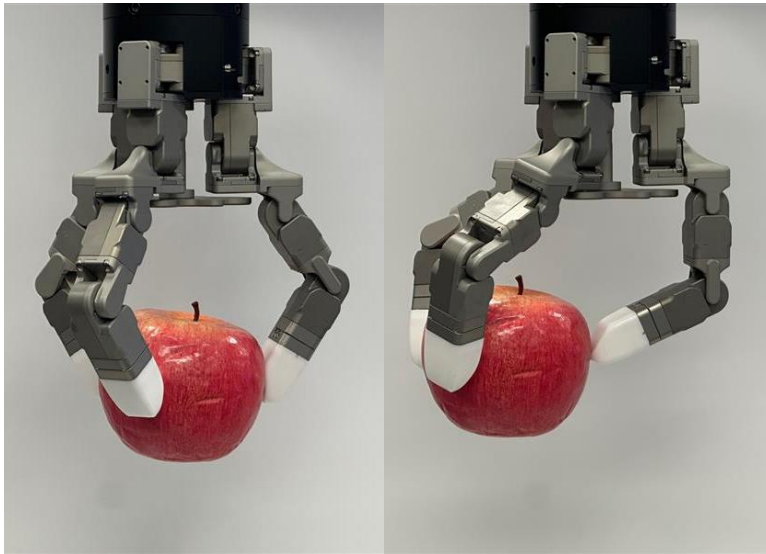
功能说明



该功能用于设置抓取数据。该功能不支持 DG-1F-M 型号。不同型号提供不同的 Grasping Algorithm，并且各 Grasping Algorithm 保存的 PRE-GRASP 位姿不同。各 Grasping Algorithm 的 PRE-GRASP 关节位置信息可通过“6.2.2 位置控制动作”或“8.3 Spy 日志”确认。

抓取默认采用实时力控制抓取算法。力控制抓取算法会使参与抓取的各指尖端部跟随手

指间的中心点。由于该算法实时运行，即使因外部冲击导致夹爪关节位置发生变化，也会在变化后的姿态下尝试维持抓取姿态。特别是在所有关节均解除“6.1.4 位置控制模式设置”中的位置控制模式时，可以更直观地确认该性能。



[即使夹爪关节位置发生变化，也会在当前位置保持抓取姿态]

项目

编号	名称	说明
1	Grasping Algorithm	9.2.9 enum DG_GRASP_MODE 参考
2	Force	抓取力度设置 (单位: N)

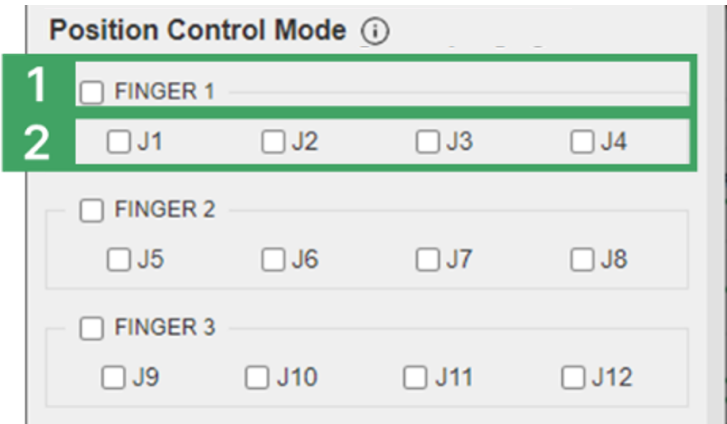
参考

- 11.1 抓取设置函数
- * 不支持的型号: DG-1F-M

* 发生错误时，请参考“附录 2 错误列表”

6.1.4 位置控制模式设置

功能说明



被设置为位置控制模式的关节在执行抓取时不会参与抓取，而是执行位置控制。该功能用于在抓取过程中指定某些关节不移动。

如果勾选 J1、J2 后在“6.1.2 抓取操作”中执行抓取，则第 1 个关节和第 2 个关节将从抓取算法中排除，并仅移动至目标位置值。

项目

编号	名称	说明
1	FINGER 1 - FINGER n	选择按钮时，选择第 n 个手指的所有关节
2	J1 - Jn	选择按钮时，选择第 n 个关节

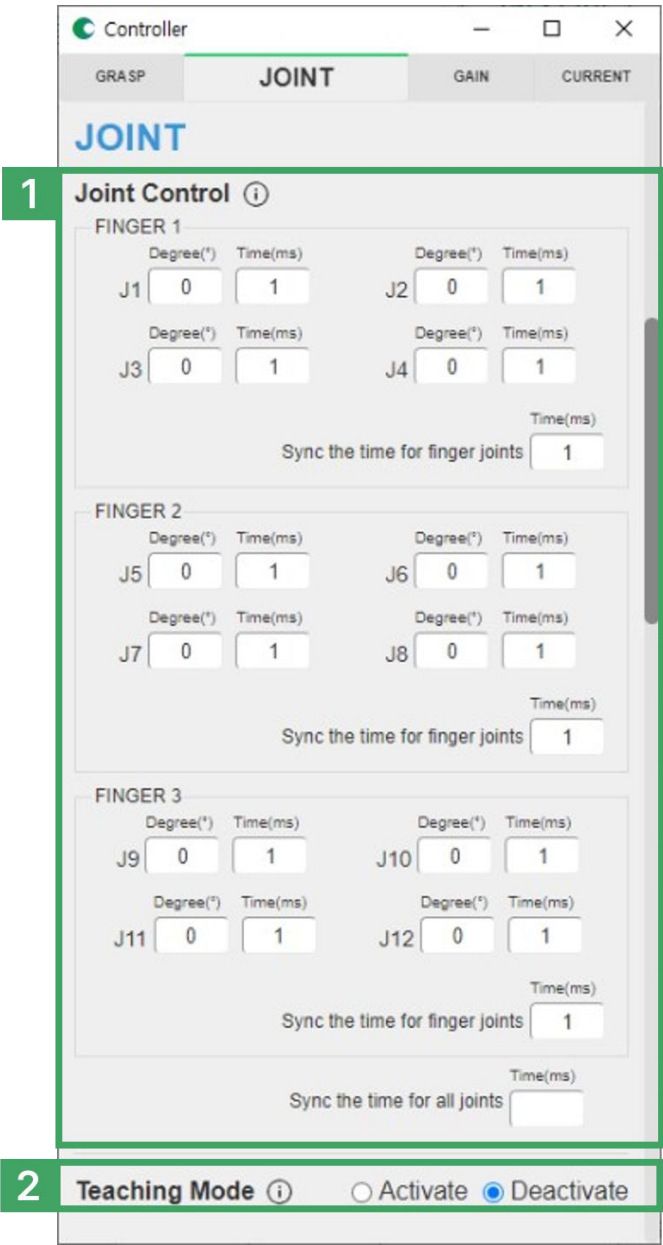
参考

11.7 位置控制模式设置

- * 不支持的型号: DG-1F-M
- * DG-4F-M 的基座关节通过第 17、第 18 号关节控制
- * 发生错误时，请参考“附录 2 错误列表”

6.2 位置控制

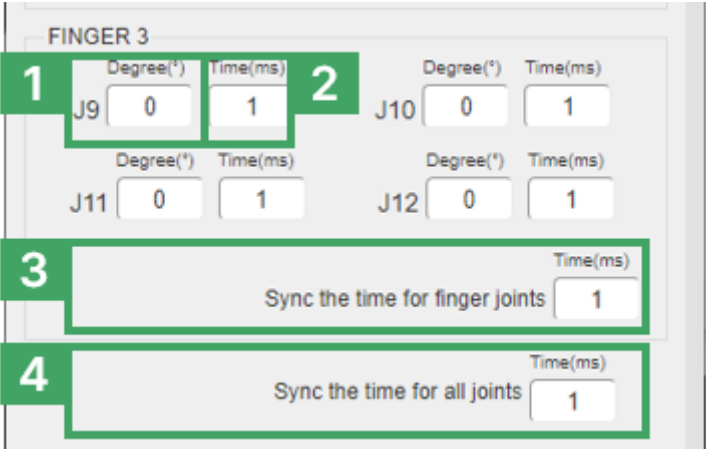
6.2.1 界面构成



编号	名称	说明
1	Joint Control	输入用于关节位置控制的目标值
2	Teaching Mode	设置用于直接示教的示教模式

6.2.2 位置控制动作

功能说明



The screenshot shows the 'FINGER 3' control panel. It contains four numbered steps highlighted with green boxes:

- Step 1:** A table for joint J9 with 'Degree(°)' set to 0 and 'Time(ms)' set to 1.
- Step 2:** A table for joint J10 with 'Degree(°)' set to 0 and 'Time(ms)' set to 1.
- Step 3:** A table for joint J11 with 'Degree(°)' set to 0 and 'Time(ms)' set to 1.
- Step 4:** A table for joint J12 with 'Degree(°)' set to 0 and 'Time(ms)' set to 1.

Below the joint tables, there are two more green boxes:

- Step 3:** A box labeled 'Sync the time for finger joints' with a 'Time(ms)' input set to 1.
- Step 4:** A box labeled 'Sync the time for all joints' with a 'Time(ms)' input set to 1.

位置控制动作是用于移动夹爪关节位置的功能。执行位置移动前需要设置目标值。

目标值单位为‘ degree’ 。位置移动时还可设置到达目标所需时间，移动时间单位为‘ ms’ 。

如果将 J1 的 Time 输入为 1000 ms，并将 degree 值输入为 60°，则 J1 对应的第一个关节将从当前位置在 1000 ms 内移动至 60°。此处设置的“Time”在开发者模式下使用时，可能会因控制器性能而无法得到准确结果；此外，也可能会因关节电流控制值而无法得到准确结果。

项目

编号	名称	说明
1	J1-Jn Degree	目标位置值输入 (单位: degree)
2	J1-Jn Time	输入移动至目标位置值所需的时间 (单位 : ms)
3	Sync the time for finger joints	输入的手指对应的关节项的移动时间输入 (单位: ms)
4	Sync the time for all joints	输入所有关节项的移动时间 (单位: ms)

参考

* 关节和手指数量因型号而异

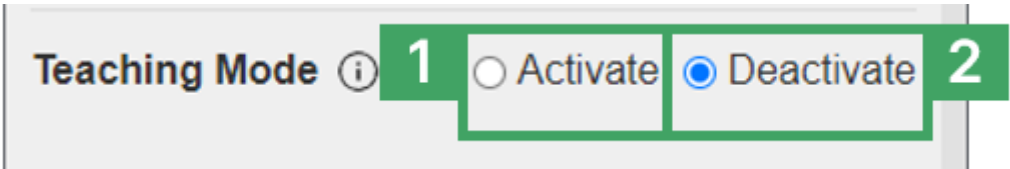
11.6 动作时间设置

11.7 位置控制模式设置

* 发生错误时，请参考“附录 2 错误列表”

6.2.3 示教模式

功能说明



示教模式是用于直接示教夹爪的功能。操作人员可以通过输入夹爪关节目标值进行示教，也可以启用示教模式并通过物理方式将夹爪移动至所需位置。若要使用示教模式，请选择“Activate”按钮；此时夹爪会进入示教模式，可通过物理方式向所需位置推动夹爪。该功能适用于创建用于保存当前位置的位姿配方。

项目

编号	名称	说明
1	Activate	示教模式启用
2	Deactivate	示教模式禁用

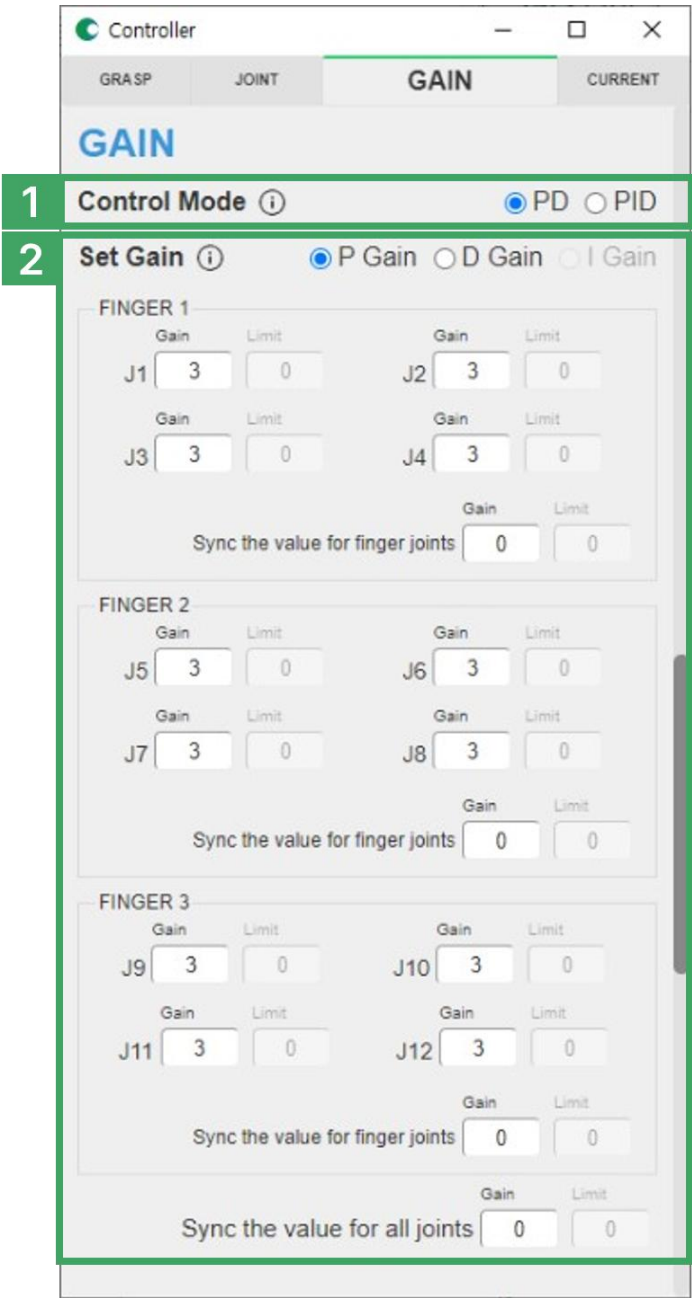
参考

13.1 ManualTeachMode

* 发生错误时，请参考“附录 2 错误列表”

6.3 增益设置

6.3.1 界面构成

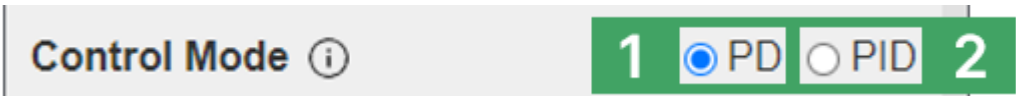


增益设置是根据目的调整夹爪关节位置移动特性的调试过程。根据调试方法，可改善夹爪反驱动性或提高位置精度。

编号	名称	说明
1	Control Mode	PD, PID 增益模式设置
2	Set Gain	各关节每个 PID 增益值设置

6.3.2 增益模式设置

功能说明



Delto Gripper 系列默认采用 PD 控制，因为在物体与夹爪发生碰撞时能够更灵活地应对物理损伤。若需提高位置精度，建议使用 PID 控制模式。

项目

编号	名称	说明
1	PD	设置为 PD 控制
2	PID	设置为 PID 控制

参考

11.5.1 SetControlPIDMode

* 发生错误时，请参考“附录 2 错误列表”

6.3.3 增益设置

功能说明

1

Set Gain ⓘ

P Gain

D Gain

I Gain

2

FINGER 1

Gain

Limit

J1

2

0

Gain

Limit

J2

2

0

Gain

Limit

J3

2

0

Gain

Limit

J4

2

0

3

Gain

Limit

Sync the value for finger joints

0

0

4

Gain

Limit

Sync the value for all joints

0

0

设置关节、手指及全部关节的 PID 增益。

项目

编号	名称	说明
1	Set Gain	选择可设置的增益
2	J1-Jn Gain	设置该关节的增益值
3	Sync the value for finger joints	批量设置该手指关节项的增益值
4	Sync the value for all joints	批量设置所有关节项的增益值

参考

* [关节和手指数量因型号而异](#)

[11.2 P 增益设置](#)

[11.3 D 增益设置](#)

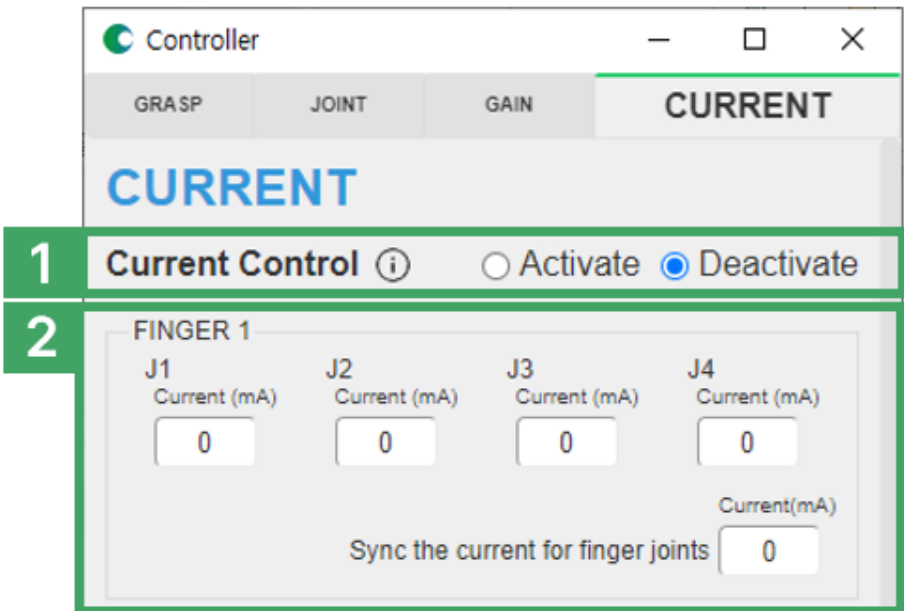
[11.4 I 增益设置](#)

[11.5 PID 增益设置](#)

* 发生错误时，请参考“[附录 2 错误列表](#)”

6.4 电流设置

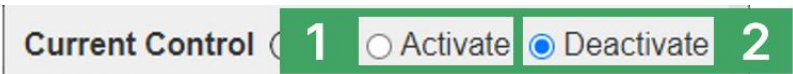
6.4.1 界面构成



使用电流设置时，可对夹爪各关节设置电流控制。在电流控制模式下无法设置 PID 增益值。

6.4.2 电流模式设置

功能说明



启用电流控制模式时，各关节将以电流控制方式动作。

项目

编号	名称	说明
1	Activate	电流控制启用
2	Deactivate	电流控制禁用

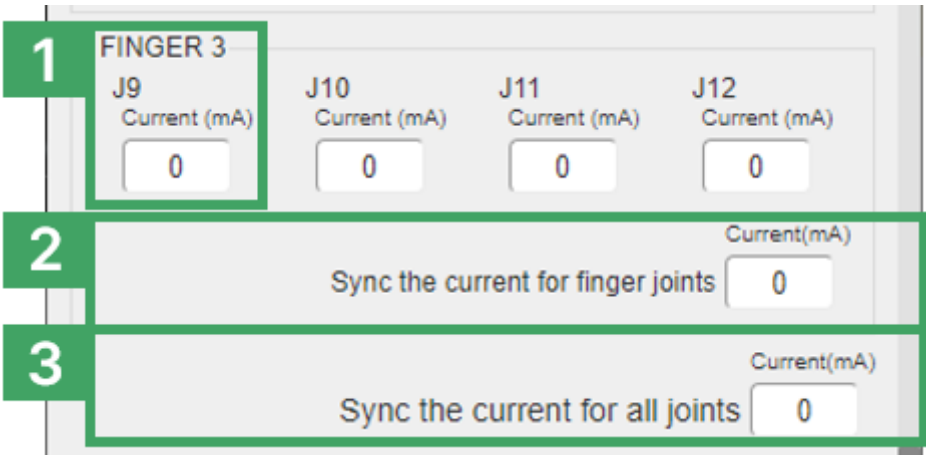
参考

13.4.1 SetCurrentControlMode

* 发生错误时，请参考“附录 2 错误列表”

6.4.3 电流设置

功能说明



电流设置值将在电流控制模式启用时生效。电流值可输入至 M 型号 150 mA、S 型号 360 mA。

项目

编号	名称	说明
1	J1-Jn Current	电流值设置
2	Sync the current for finger joints	批量设置该手指关节项的电流值
3	Sync the current for all joints	批量设置所有关节项的电流值

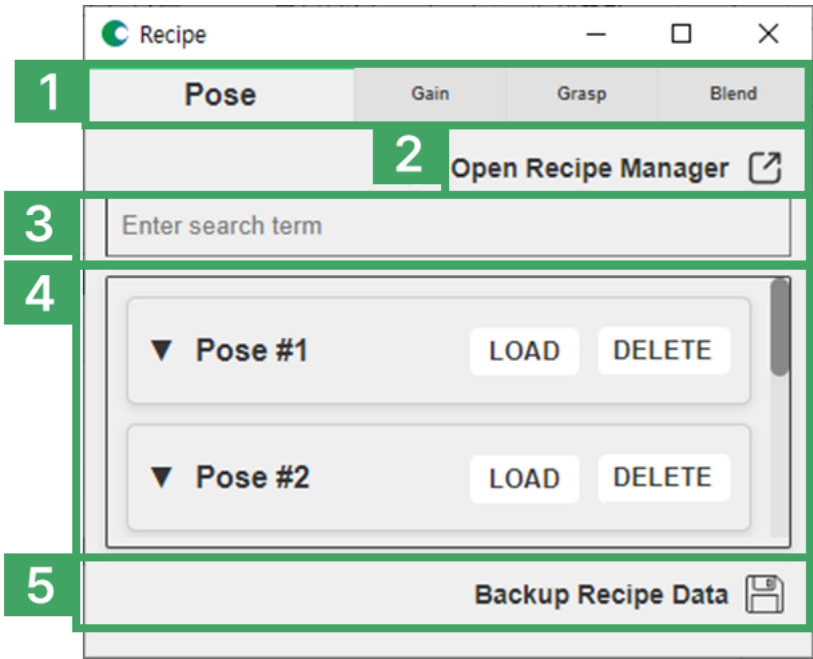
参考

13.4 电流控制

* 发生错误时，请参考“附录 2 错误列表”

第 7 章 配方设置界面

7.1 界面构成



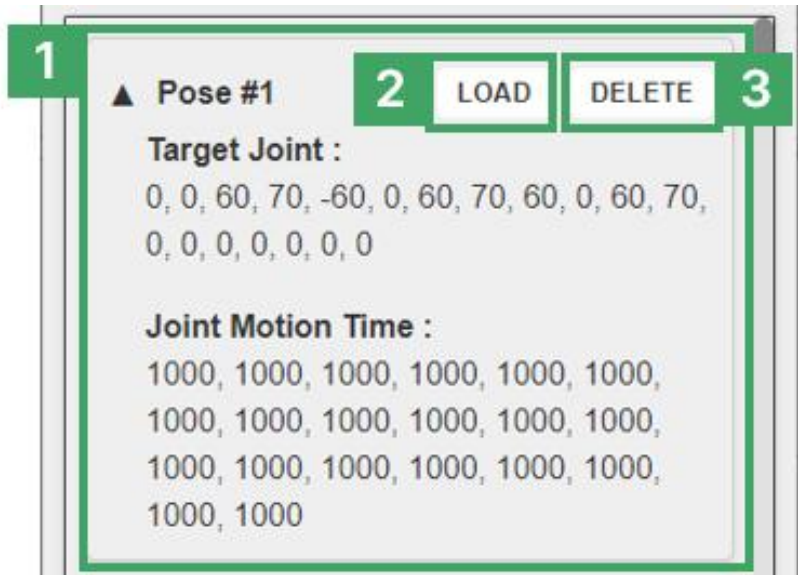
配方设置界面显示当前已保存的配方。可搜索并查看配方，点击 **LOAD** 执行相应配方，或点击 **DELETE** 删除该配方。程序内设置的配方可通过备份功能保存至夹爪或 PC。在用户模式下，可通过 **EEPROM WRITE** 保存配方。执行 **EEPROM WRITE** 后，当前设置的数据将永久保存至夹爪的内存中。即使重新启动夹爪电源，该数据仍会保留，这表示其保存于非易失性存储器中。

项目

编号	名称	说明
1	Recipe Menu	配方项目选择
2	Open Recipe Manager	配方编辑或添加执行管理器打开
3	Search Recipe	用于搜索配方的功能
4	Show Recipe List	已保存的配方项目列表显示
5	Backup Recipe Data	根据使用模式执行配方项目备份 ☐ 用户模式: EEPROM WRITE ☐ 开发者模式: BACKUP RECIPE DATA

7.2 位姿配方

7.2.1 界面构成

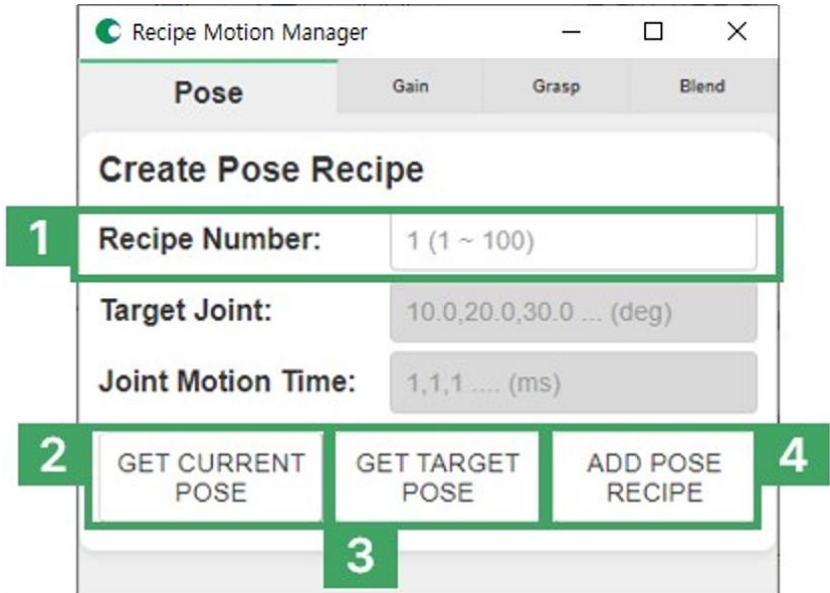


显示当前已保存的位姿配方。位姿配方包括目标关节值和动作时间。点击 **LOAD** 按钮可执行该配方，点击 **DELETE** 按钮可删除该配方。

项目

编号	名称	说明
1	配方信息	配方信息显示 ☐ 目标关节值 ☐ 动作时间
2	LOAD	在管理器界面加载该配方并执行夹爪动作
3	DELETE	删除该配方

7.2.2 位姿配方设置方法



输入可保存的配方编号后，可加载当前关节位置值或目标关节位置值。点击“添加位姿配方”按钮即可保存配方。GET TARGET POSE 会加载在“6.2 位置控制”中输入的 Joint Degree 值。

项目

编号	名称	说明
1	Recipe Number	输入可保存的配方编号。 ☐ 用户模式下最多可保存 100 个。
2	GET CURRENT POSE	当前关节位置及到达时间值获取
3	GET TARGET POSE	目标关节位置及到达时间值获取
4	ADD POSE RECIPE	以设置的配方编号保存位姿配方

参考

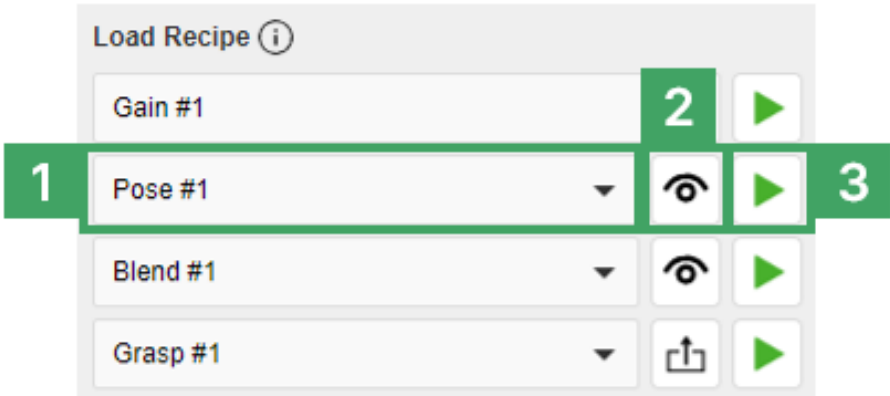
[12.1.1 UpdateRecipeCurrentPoseData](#)

[12.1.2 UpdateRecipeTargetPoseData](#)

* 发生错误时，请参考“附录 2 错误列表”

7.2.3 位姿配方动作使用方法

功能说明



在主界面选择并加载所需的位姿配方。

项目

编号	名称	说明
1	位姿配方选择	位姿配方选择
2	位姿配方预览	通过仿真确认位姿配方动作 按住按钮期间仿真会运行
3	位姿配方加载	位姿配方加载

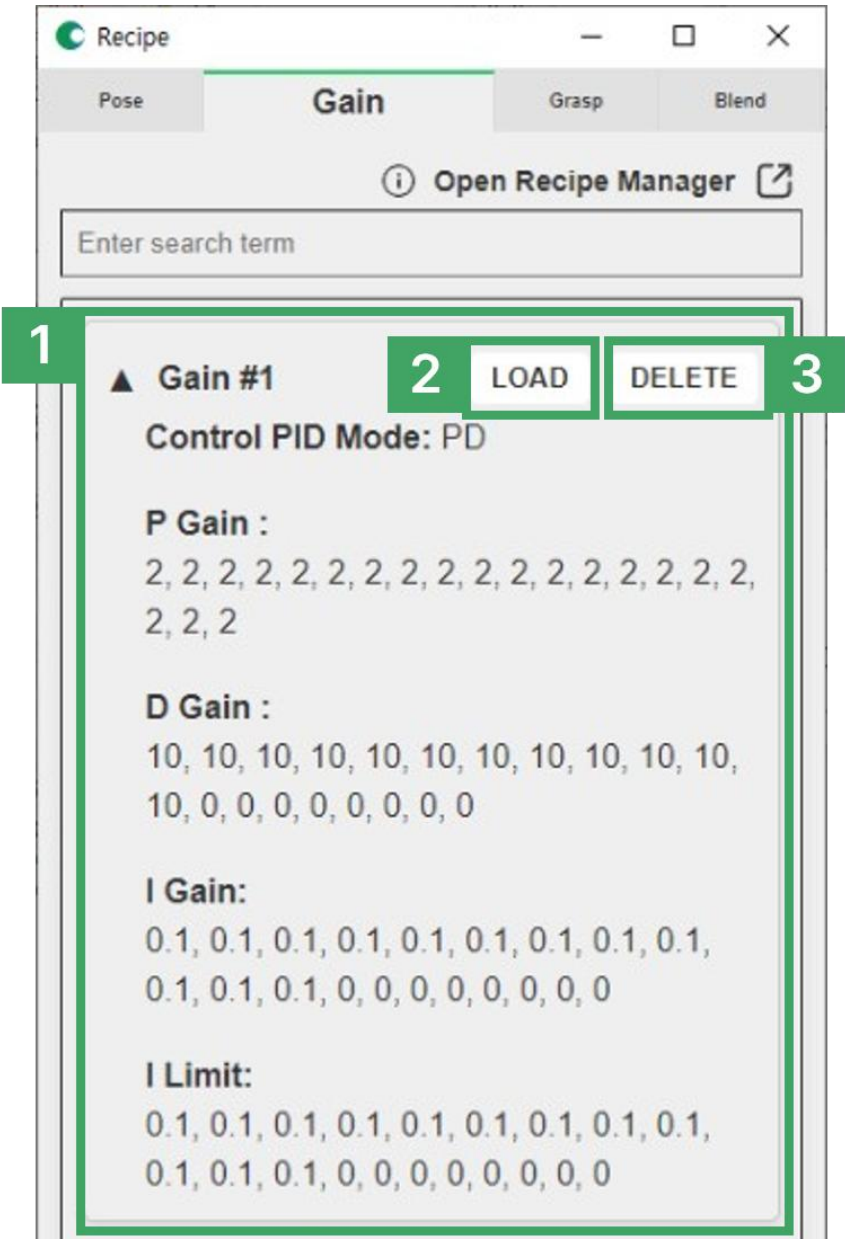
参考

- 12.1.1 UpdateRecipeCurrentPoseData
- 12.1.2 UpdateRecipeTargetPoseData
- 12.1.3 LoadRecipePoseData
- * 发生错误时，请参考“附录 2 错误列表”

7.3 增益配方

7.3.1 界面构成

功能说明



显示当前已保存的增益配方。增益配方包括 P 增益、D 增益、I 增益及误差积分最大值。点击 LOAD 按钮可执行该配方，点击 DELETE 按钮可删除该配方。

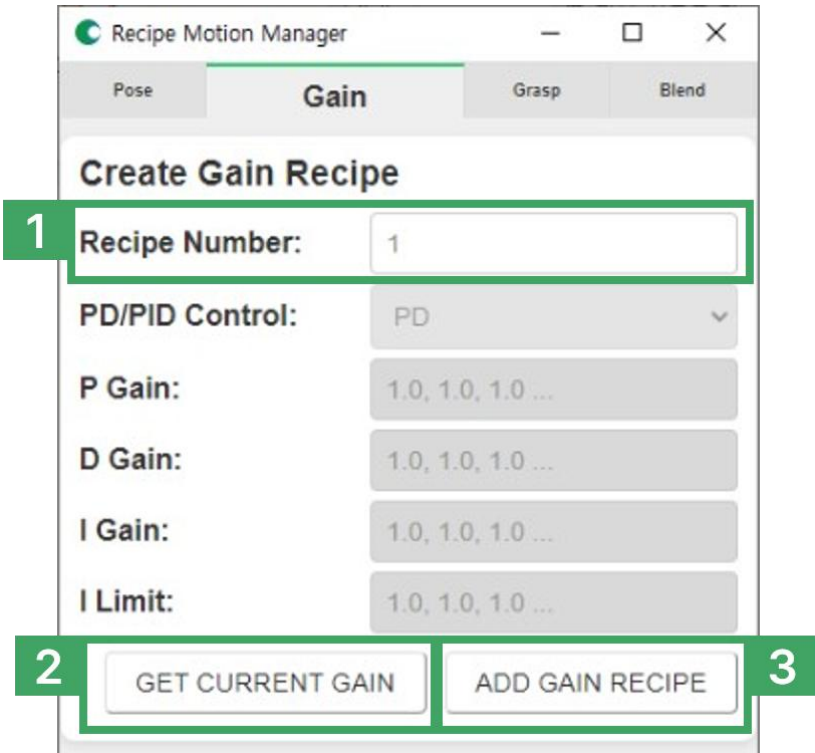
项目

编号	名称	说明
----	----	----

1	配方信息	配方信息显示 ☐ P 增益 ☐ D 增益 ☐ I 增益 ☐ I Limit. 误差积分最大值
2	LOAD	在管理器界面加载该配方并执行夹爪动作
3	DELETE	删除该配方

7.3.2 增益配方设置方法

功能说明



输入所需的配方编号后，可加载当前增益值。点击“添加增益配方”按钮即可保存配方。

项目

编号	名称	说明
----	----	----

1	Recipe Number	输入可保存的配方编号。 用户模式下最多可保存 20 个。
2	GET CURRENT GAIN	获取当前增益值
3	ADD GAIN RECIPE	以设置的配方编号保存增益配方

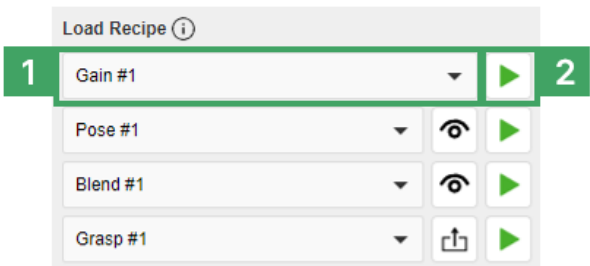
参考

12.2.1 UpdateRecipeGainData

* 发生错误时，请参考“附录 2 错误列表”

7.3.3 增益配方动作使用方法

功能说明



在主界面选择并加载所需的增益配方。

项目

编号	名称	说明
1	增益配方选择	从列表中选择增益配方
2	增益配方加载	加载所选增益配方

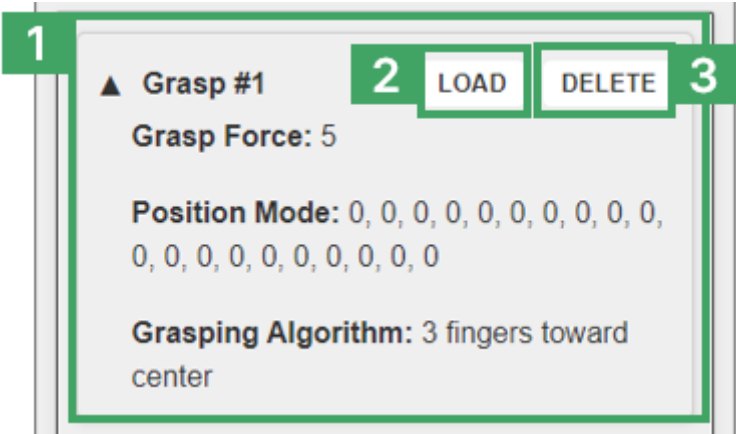
参考

12.2.2 LoadRecipeGainData

* 发生错误时，请参考“附录 2 错误列表”

7.4 抓取配方

7.4.1 界面构成



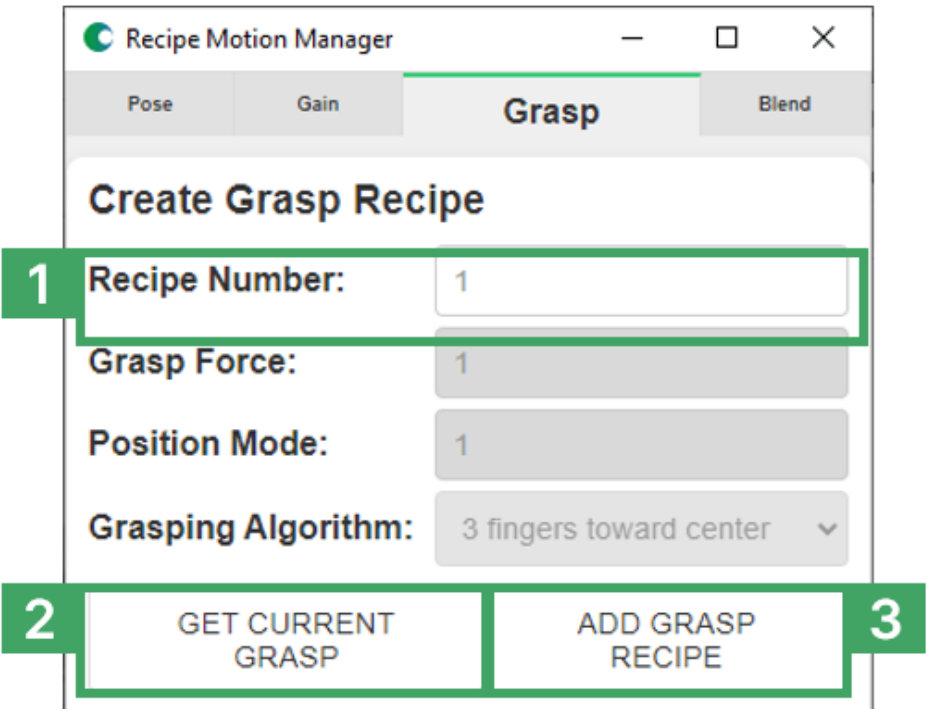
显示当前已保存的抓取配方。抓取配方包括抓取力度设置、位置控制和抓取模式。点击 LOAD 按钮可执行该配方，点击 DELETE 按钮可删除该配方。

项目

编号	名称	说明
1	配方信息	配方信息显示 ☐ Grasp Force ☐ Position Mode ☐ Grasping Algorithm
2	LOAD	在管理器界面加载该配方并执行夹爪动作
3	DELETE	删除该配方

7.4.2 抓取配方设置方法

功能说明



输入可保存的配方编号后，可加载当前抓取设置值。点击“添加抓取配方”按钮即可保存配方。

项目

编号	名称	说明
1	Recipe Number	输入可保存的配方编号 ☐ 用户模式下最多可保存 20 个
2	GET CURRENT GRASP	获取当前抓取值
3	ADD GRASP RECIPE	以设置的配方编号保存抓取配方

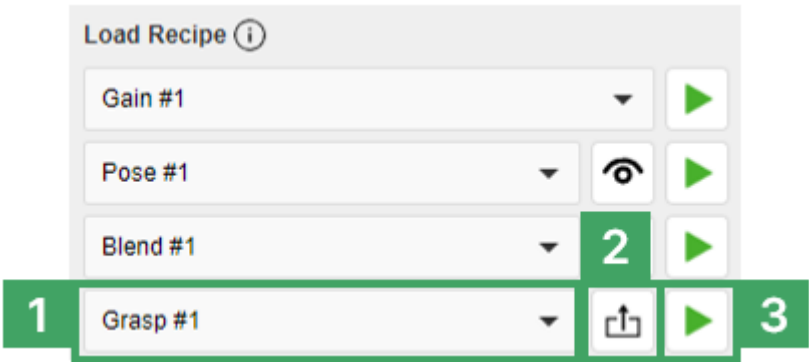
参考

12.3.1 UpdateRecipeGraspData

* 发生错误时，请参考“附录 2 错误列表”

7.4.3 抓取配方动作使用方法

功能说明



在主界面选择并加载执行动作所需的配方。

项目

编号	名称	说明
1	抓取配方选择	从列表中选择可使用的抓取配方
2	LOAD	加载所选抓取配方
3	GRASP	▶: 抓取动作执行 ■: 执行抓取解除

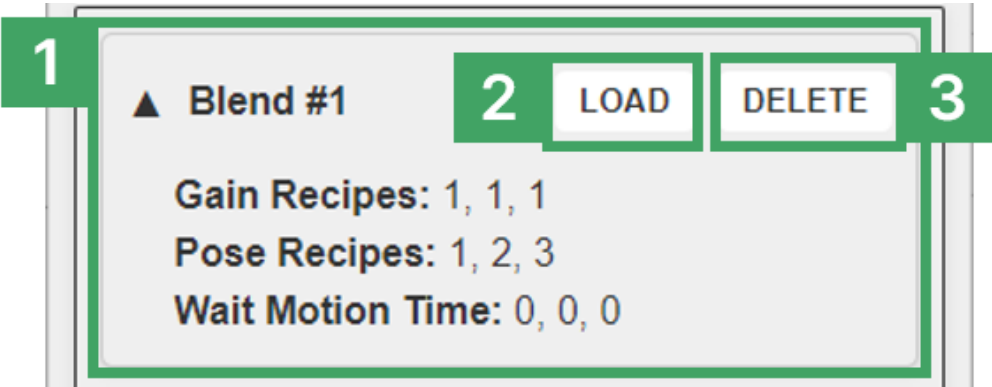
参考

12.3.2 LoadRecipeGraspData

* 发生错误时，请参考“附录 2 错误列表”

7.5 混合配方

7.5.1 界面构成

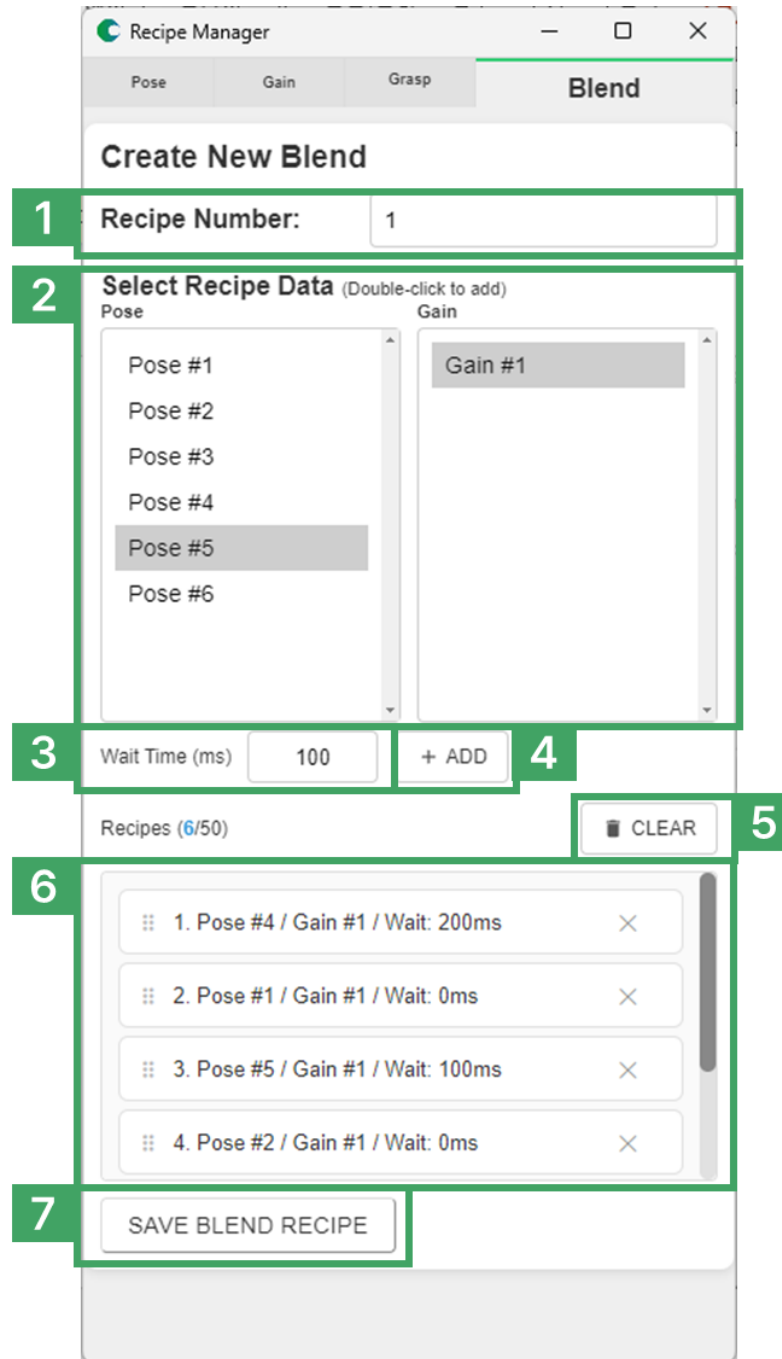


显示当前已保存的混合配方。混合配方包括增益配方、位姿配方和动作时间。点击 LOAD 按钮可加载该配方，点击 DELETE 按钮可删除该配方。

项目

编号	名称	说明
1	混合配方信息	配方信息显示 ☐ Gain Recipes ☐ Pose Recipes ☐ Wait Motion Time(ms)
2	LOAD	当前混合配方加载
3	DELETE	当前选择的混合配方删除

7.5.2 混合配方设置方法



输入可保存的混合配方编号，并设置位姿、增益和等待时间后，点击第 4 项“ADD SELECTED RECIPE”按钮添加混合动作。按上述步骤重复多次，添加所有混合动作项后，点击第 7 项“SAVE BLEND RECIPE”保存混合配方。

项目

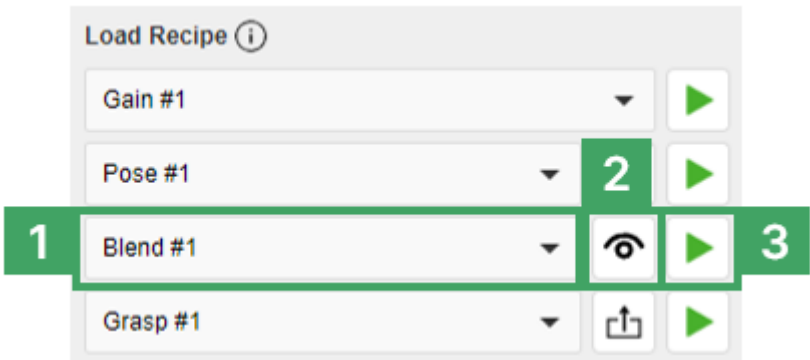
编号	名称	说明
1	Recipe Number	输入可保存的配方编号 ☐ 用户模式下最多可保存 10 个
2	Select Recipe Data	选择位姿配方和增益配方
3	Wait Time	混合运动等待时间设置. 执行当前配方， 并等待输入的时间后执行下一个配方
4	ADD	将选择的位姿、增益配方和等待时间添加 到列表中 ☐ 一个混合最多可保存 50 个配方
5	CLEAR	初始化配方列表
6	Blend List	位姿配方，增益配方，添加位姿配方、增 益配方和动作时间的混合列表
7	SAVE BLEND RECIPE	以设置的配方编号保存混合配方

参考

- [12.4.1 Update Blend Joint](#)
[12.4.2 Clear Move Blend Joint](#)
[12.4.3 Add Move Blend Joint](#)
[12.4.4 Set Move Blend Joint](#)
[12.4.5 Start Move Blend Joint](#)

7.5.3 混合配方动作使用方法

功能说明



在主界面选择并加载所需的混合配方。

项目

编号	名称	说明
1	混合配方选择	从列表中选择混合配方
2	混合配方预览	通过仿真确认位姿配方动作 按住按钮期间仿真会运行
3	混合配方加载	加载所选混合配方

参考

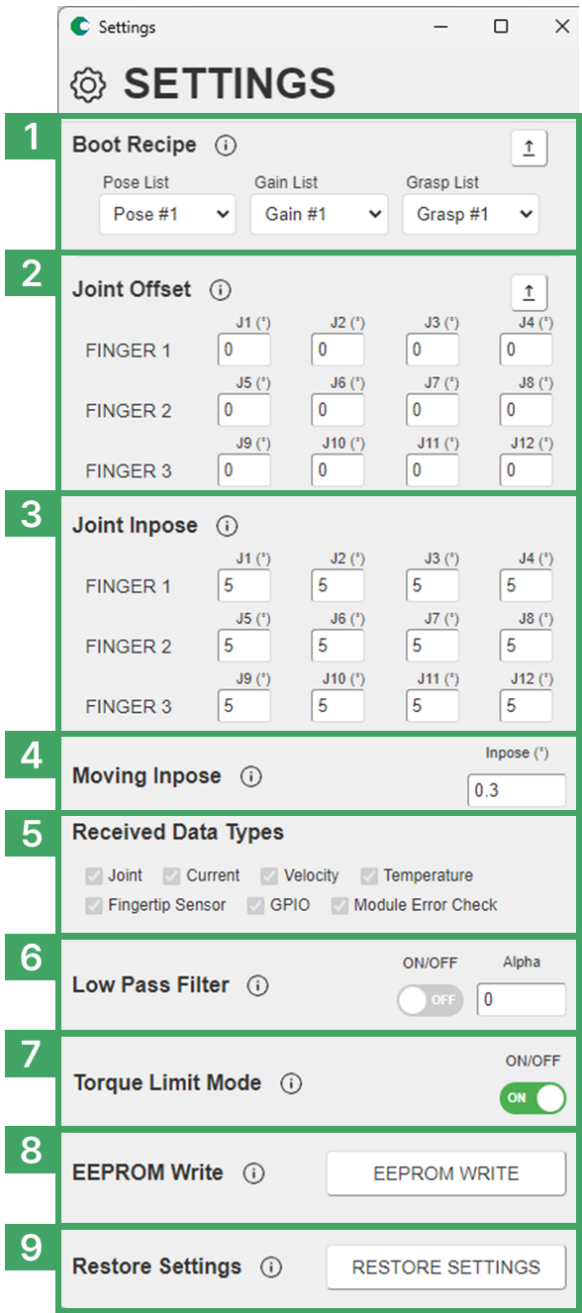
[12.4.5 Start Move Blend Joint](#)

[12.4.6 Stop Move Blend Joint](#)

第 8 章 附加功能界面

8.1 设置

8.1.1 界面构成

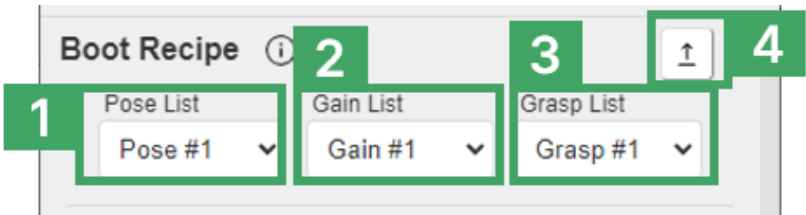


设置界面用于更改夹爪的启动配方、偏移等夹爪设置。

项目

编号	名称	说明
1	Boot Recipe	设置夹爪控制开始时应用的配方
2	Joint Offset	设置夹爪各关节的偏移值
3	Joint Inpose	设置用于确认关节到达的范围
4	Moving Inpose	设置用于确认关节移动的范围
5	Received Data Types	显示从夹爪接收的数据 可在连接前设置
6	Low Pass Filter	低通滤波器设置
7	Torque Limit Mode	设置扭矩限制模式。仅限 M 型号 ☐ 使用时需注意
8	Backup Settings	根据使用模式执行备份 ☐ 用户模式: EEPROM WRITE ☐ 开发者模式: BACKUP SETTINGS
9	Restore Settings	执行恢复

8.1.2 Boot Recipe



启动配方是在夹爪上电后开始控制时应用的配方设置功能。

项目

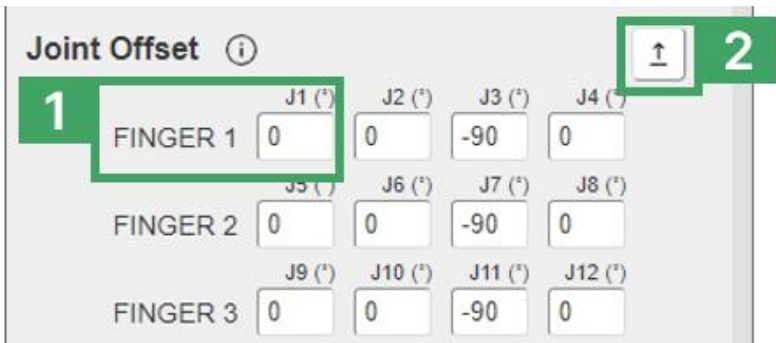
编号	名称	说明
1	位姿启动配方	设置启动时可应用的位姿配方
2	增益启动配方	设置启动时可应用的增益配方
3	抓取启动配方	设置启动时可应用的抓取配方
4	应用按钮	应用设置的启动配方数据

参考

* 发生错误时，请参考“附录 2 错误列表”

8.1.3 Joint Offset

功能说明



关节偏移设置用于在关节物理位置发生偏差时，通过软件方式校准关节零位姿。

项目

编号	名称	说明
----	----	----

1	Joint Offset J1-Jn	n 号关节偏移设置 (单位: degree)
2	应用按钮	设置的 Joint Offset, Joint Inpose, Moving Inpose 批量应用

参考

10.1.2 SetGripperOption

* 发生错误时，请参考“附录 2 错误列表”

8.1.4 Joint Inpose

功能说明

Joint Inpose ⓘ

1

FINGER 1	1	1	1	1
	J1 (°)	J2 (°)	J3 (°)	J4 (°)
FINGER 2	1	1	1	1
	J5 (°)	J6 (°)	J7 (°)	J8 (°)
FINGER 3	1	1	1	1
	J9 (°)	J10 (°)	J11 (°)	J12 (°)

Inpose 用于设置各关节确认是否到达目标位置的范围。

例如，当 J1 设置为 3°，且当前关节位置值为 30° 时，如果执行将目标关节位置值设为 60° 的位置控制，则当当前关节位置值达到 57° 时，系统会判断已到达目标位置。该功能适用于需要连续执行位置移动，并在到达目标关节位置之前执行下一动作的情况。点击“8.1.3 Joint Offset”中的应用按钮后生效。

项目

编号	名称	说明
1	J1-Jn Inpose	n 号关节的到达确认范围设置 (单位: degree)

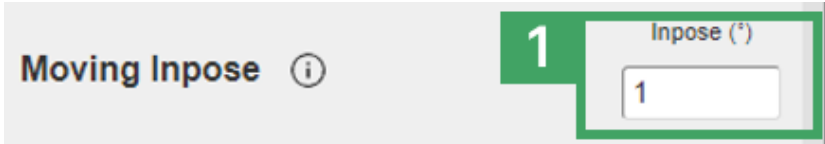
参考

10.1.2 SetGripperOption

* 发生错误时，请参考“附录 2 错误列表”

8.1.5 Moving Inpose

功能说明



Moving Inpose 用于设置判断关节是否发生运动的范围。若 Inpose 值为 1°，当关节的变化量达到 1° 以上时，则判断该关节已发生运动。点击“8.1.3 Joint Offset”中的应用按钮后生效。

项目

编号	名称	说明
1	Inpose	设置关节运动确认范围（单位：degree）

参考

10.1.2 SetGripperOption

* 发生错误时，请参考“附录 2 错误列表”

8.1.6 Low Pass Filter



低通滤波器设置。

项目

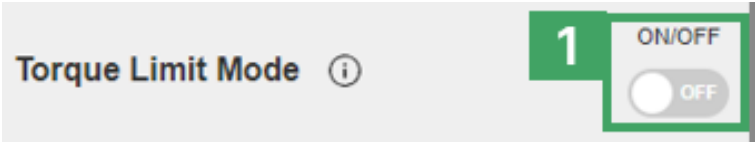
编号	名称	说明
1	ON / OFF	设置是否启用滤波器
2	Alpha	设置滤波器增益值

参考

10.3.1 SetLowPassFilterAlpha 参考
* 发生错误时，请参考“附录 2 错误列表”

8.1.7 Torque Limit Mode

功能说明



Torque Limit Mode 用于解除夹爪的扭矩限制，以使用夹爪提供的最大扭矩而非额定扭矩。该功能仅支持 M 型号。

项目

编号	名称	说明
1	ON/OFF	启用/禁用扭矩限制模式

参考

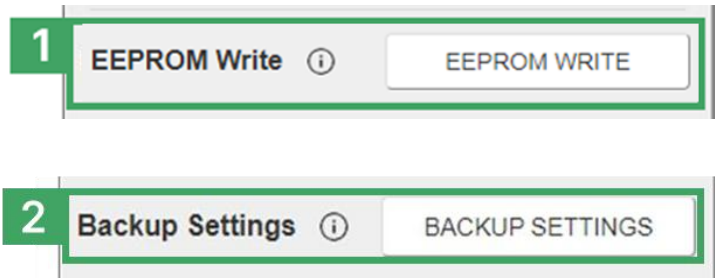
注意！在用户模式下将 Torque limit mode 设为 off，或在开发者模式下使用 stall torque 时，可能导致产品温度快速上升及产品损坏；发生故障时可能无法享受售后服务。

10.3.5 SetTorqueLimitMode 参考

* 发生错误时，请参考“附录 2 错误列表”

8.1.8 Backup Settings / EEPROM Write

功能说明



Backup Settings/EEPROM Write 用于将当前已应用的设置数据保存至 PC 或夹爪。在用户模式下，通信连接时会自动执行备份。将数据保存至 EEPROM 的过程中，如果关闭夹爪电源或断开通信，可能会导致数据损坏，因此需要注意。

DGManager 初次运行时会显示使用方法的指南界面，之后可在“4.2.1 界面构成”中的第 13 项“程序信息显示窗口”中进行确认。

项目

编号	名称	说明
1	EEPROM WRITE	在用户模式下，将当前设置的值保存至夹爪
2	BACKUP SETTINGS	在开发者模式下，将当前设置的数据备份到 PC

参考

10.3.8 BackupRecipeData 参考

* 发生错误时，请参考“附录 2 错误列表”

8.1.9 Restore Settings

功能说明



Restore Settings 用于加载外部路径中的备份文件，并将数据恢复到夹爪。

相关文件用户模式连接时或开发者模式在 ‘ 8.1.8 Backup Settings / EEPROM Write’ 的 Backup Settings 时保存。

保存路径如下：

DGManager-win32-x64(程序路径)\resources\config\{型号名称} \{控制模式}

例) DGManager-win32-x64\resources\config\DG-3F-B\Operator

首次运行 DGManager 时，会显示使用方法指南界面；之后也可在 “4.2.1 界面构成” 中的第 13 项 “程序信息” 显示窗口中确认。

项目

编号	名称	说明
1	RESTORE SETTINGS	使用备份文件恢复数据

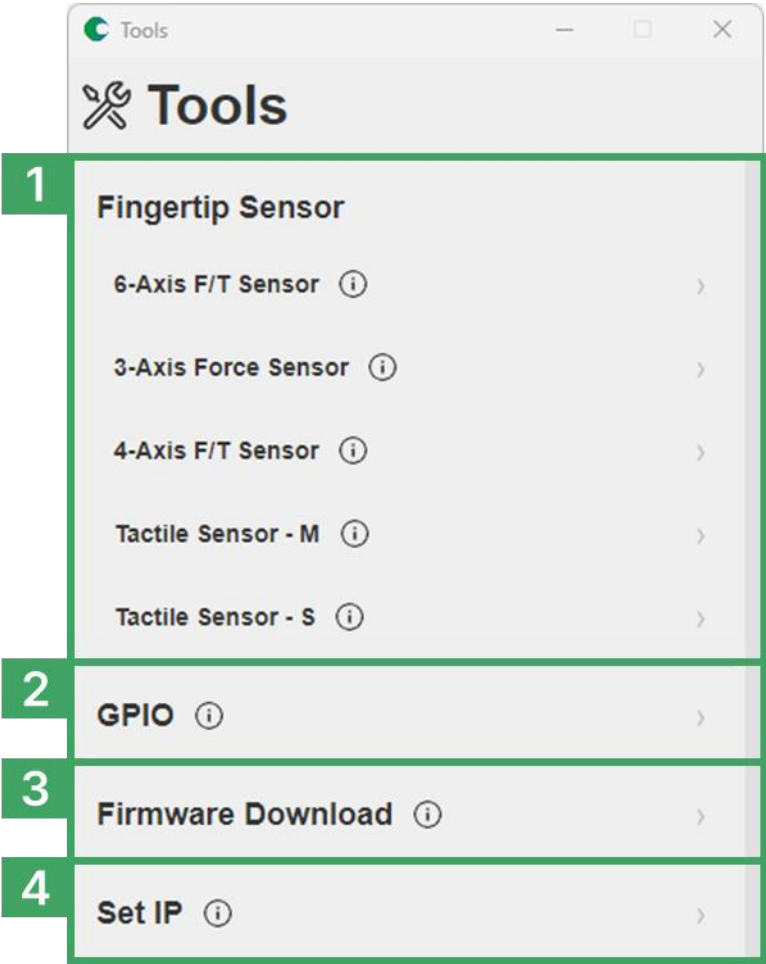
参考

10.3.9 RestoreRecipeData 参考

* 发生错误时，请参考 “附录 2 错误列表”

8.2 工具

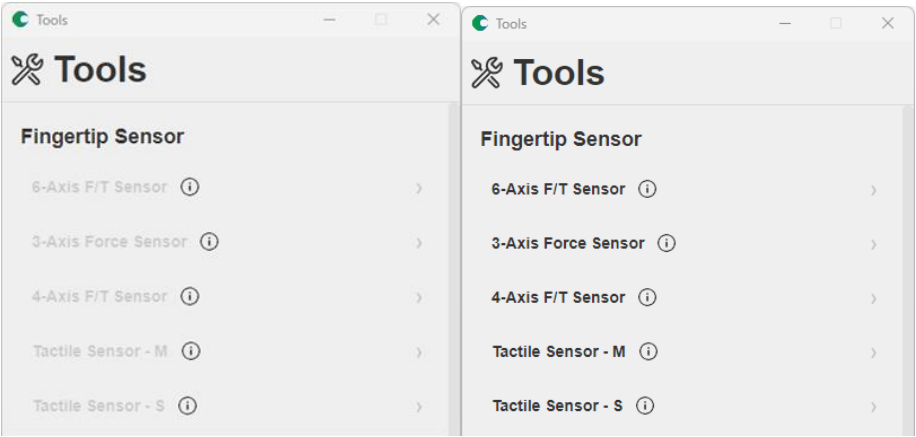
8.2.1 界面构成



编号	名称	说明
1	Fingertip Sensor	指尖传感器数据监控
2	GPIO	打开用于控制夹爪 GPIO 的面板
3	Firmware Download	选择夹爪固件文件并下载至夹爪
4	Set IP	夹爪 IP 和 PORT 设置

8.2.2 Fingertip Sensor

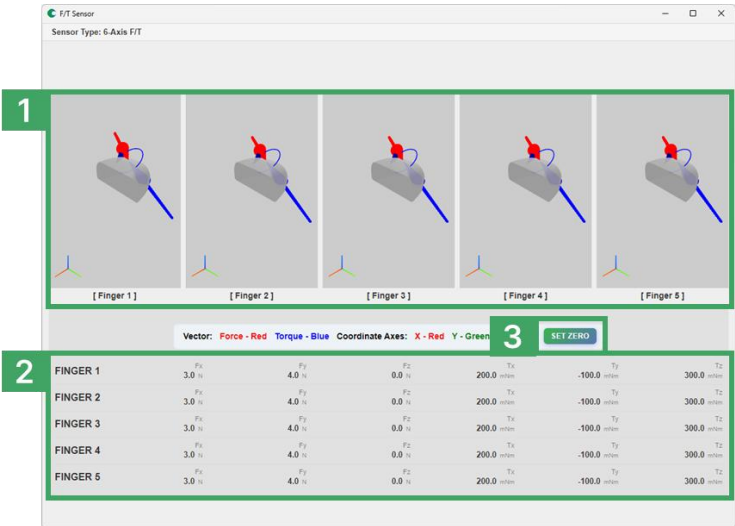
功能说明



Delto Gripper 可安装内置传感器的指尖。根据 M 型号和 S 型号，提供的指尖版本不同。未应用指尖选件的固件无法使用该功能，固件中保存的指尖选件对应的指尖传感器会被启用。

8.2.2.1 6-Axis F/T Sensor

功能说明



可监控 6-Axis Force/Torque 传感器数据，并确认力和扭矩值。
该功能仅支持 M 型号。

项目

编号	名称	说明
1	传感器 3D 可视化	以 3D 方式可视化 FT 传感器数据
2	FT 传感器数据	FT 传感器数据确认 (单位: N, mN· m)
3	FT 传感器初始化	将 FT 传感器数据初始化为 0

参考

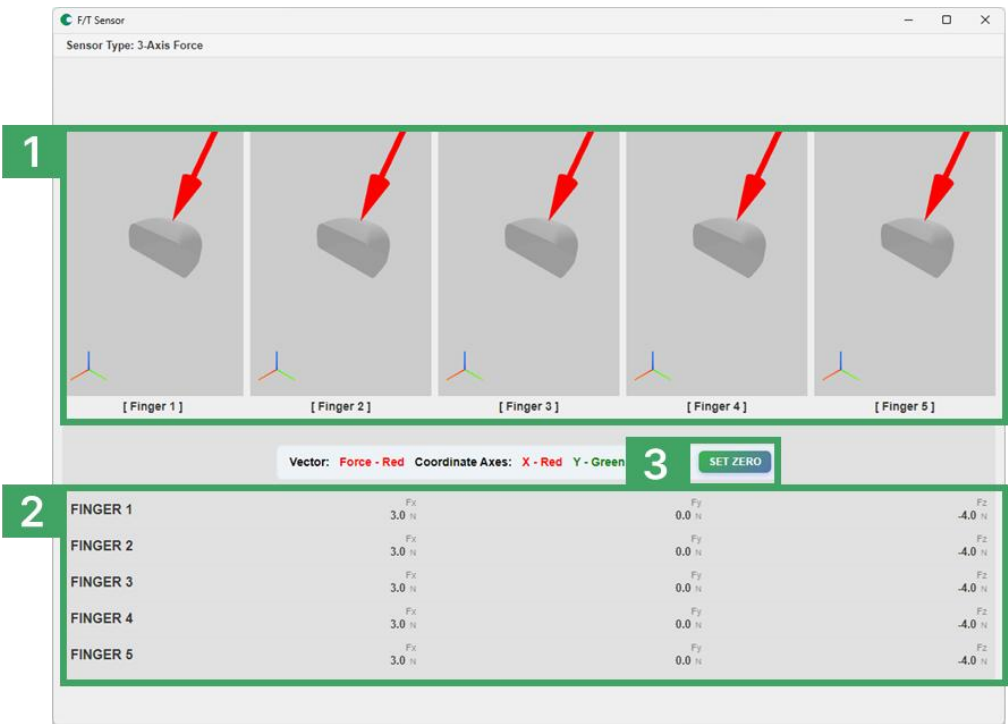
10.2.5 CallbackForOnReceivedFingertipSensorData

11.8.1 SetFingerTipDataZero

* 发生错误时，请参考“附录 2 错误列表”

8.2.2.2 3-Axis Force Sensor

功能说明



可监控 3-Axis Force 传感器数据，并确认力值。该功能仅支持 M 型号。

项目

编号	名称	说明
1	传感器 3D 可视化	以 3D 方式可视化 FT 传感器数据
2	FT 传感器数据	FT 传感器数据确认 (单位: N)
3	FT 传感器初始化	将 FT 传感器数据初始化为 0

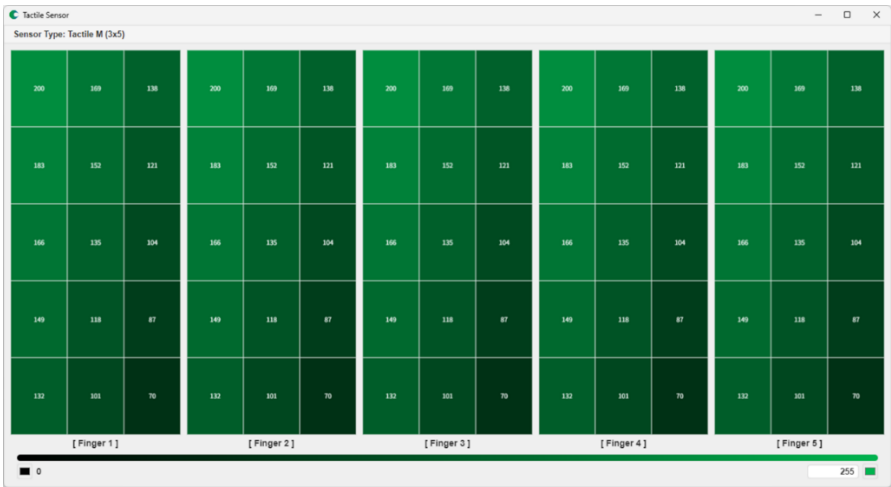
参考

10.2.5 CallbackForOnReceivedFingertipSensorData

* 发生错误时, 请参考“附录 2 错误列表”

8.2.2.3 Tactile Sensor - M/S

功能说明



可监控 Tactile Sensor 数据。M 型号由 3x5 个节点组成, S 型号由 3x6 个节点组成, 可确认各节点的力度。

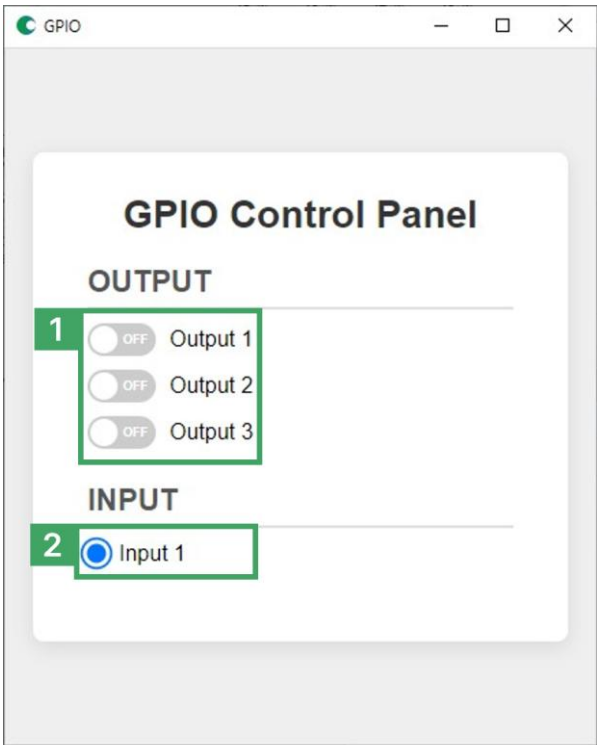
参考

10.2.5 CallbackForOnReceivedFingertipSensorData

* 发生错误时, 请参考“附录 2 错误列表”

8.2.3 GPIO

功能说明



GPIO 可确认夹爪控制器内置的 Output ON/OFF 及 Input 状态。该功能与连接至夹爪电源部的 8PIN 电缆联动，可通过夹爪 GPIO 控制器与外部设备收发 IO 信号。

项目

编号	名称	说明
1	Output	夹爪 Output ON/OFF 执行
2	Input	夹爪 Input 状态显示

参考

[10.2.6 CallbackForOnReceivedGPIOData](#)

[10.3.3 SetGPIOOutput](#)

[10.3.4 SetGPIOOutputAll](#)

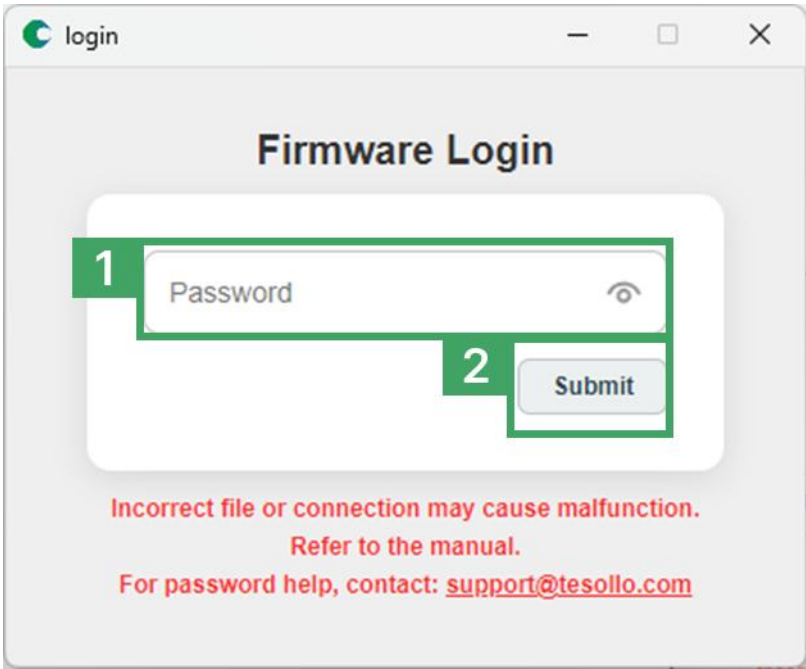
* 发生错误时，请参考“附录 2 错误列表”

8.2.4 Firmware Download

Firmware Download 用于更新夹爪主控制器固件。使用该功能需要密码。如需密码，请联系技术支持邮箱。（support@tesollo.com）

8.2.4.1 登录界面

功能说明



该登录界面用于防止任意固件更新尝试。输入密码后，可切换至固件下载界面。

项目

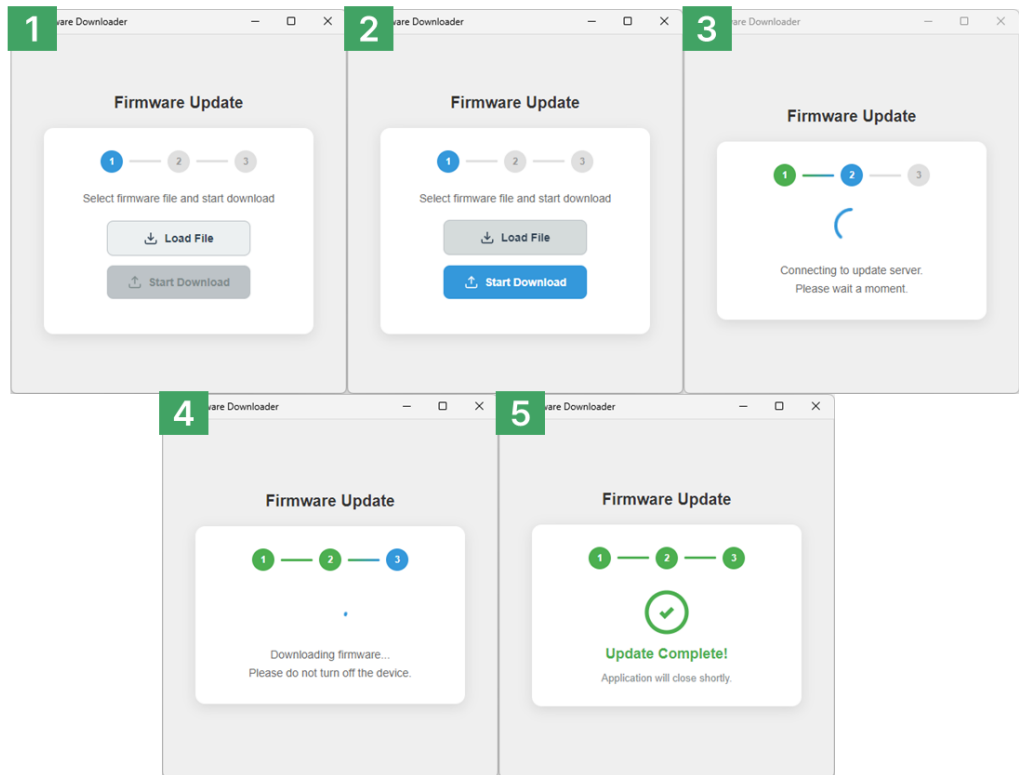
编号	名称	说明
1	密码输入	密码 support@tesollo.com 为参数的
2	Submit	日志的

参考

* 发生错误时，请参考“附录 2 错误列表”

8.2.4.2 固件下载界面

功能说明



显示第 1 个界面后，先选择“Load File”加载可下载的固件文件，然后选择“Start Download”。进入第 3 个界面并最终显示第 5 个界面后，表示固件下载已正常完成。随后夹爪将自动重启；如需重新连接，需要重新运行程序。

项目

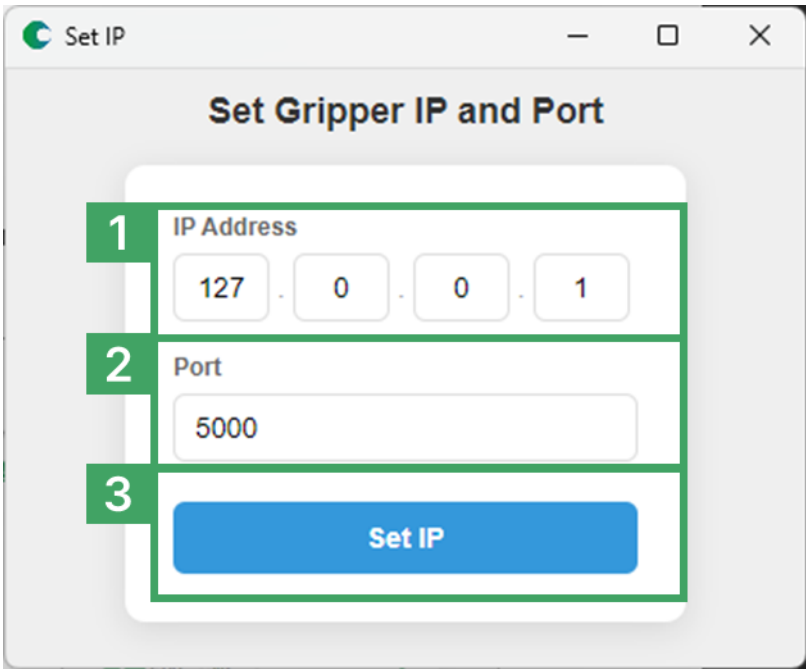
编号	名称	说明
1	LOAD FILE	从存储位置加载固件
2	START DOWNLOAD	固件下载执行. 启动模式进入

参考

10.3.2 SetBootMode
* 发生错误时，请参考“附录 2 错误列表”

8.2.5 Set IP

功能说明



夹爪 IP和 Port 设置。

项目

编号	名称	说明
1	IP	IP 输入
2	Port	Port 输入
3	Set IP	输入的 IP 和 Port 设置

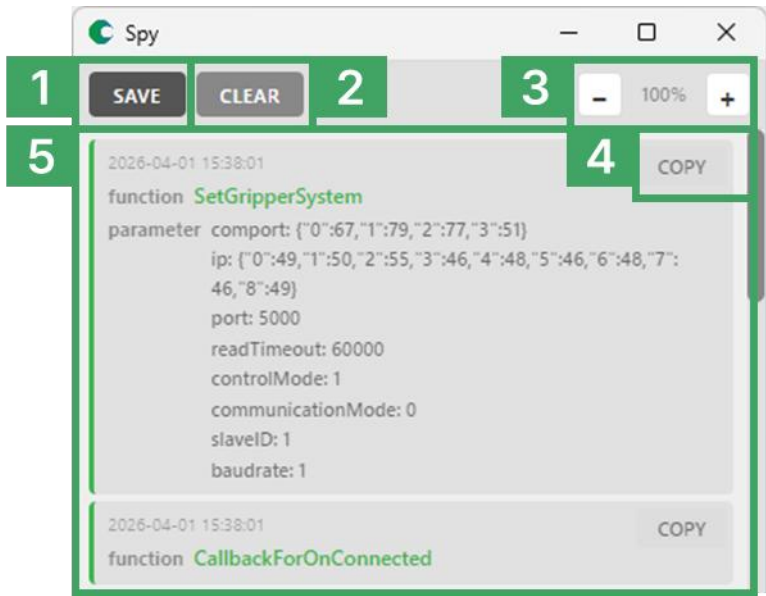
参考

10.1.7 SetIP

* 发生错误时，请参考“附录 2 错误列表”

8.3 Spy

Spy 是用于以日志形式确认使用 DG Manager 过程中所发生事件的界面。为了操作夹爪，在DG Manager中点击按钮时，系统会以函数形式显示与该动作对应的命令。该功能可帮助用户轻松确认操作夹爪的方法。

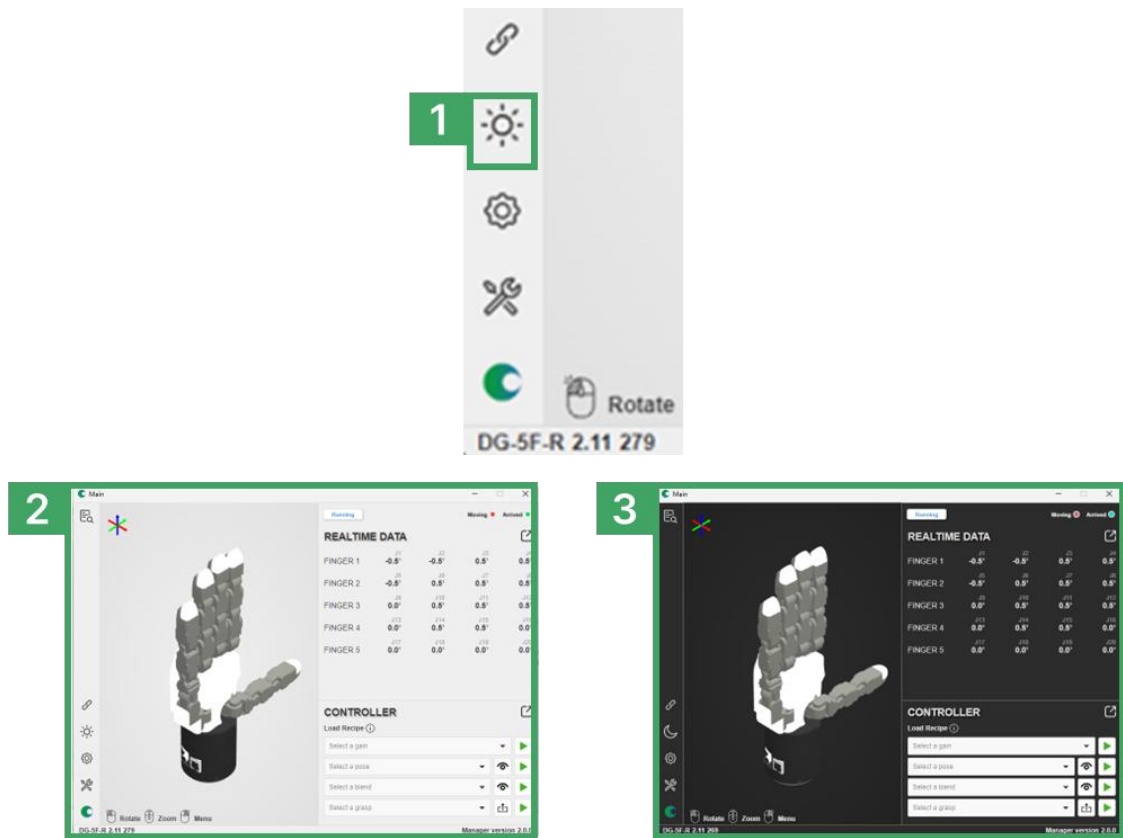


编号	名称	说明
1	SAVE	将 Spy 日志保存为 txt 文件
2	CLEAR	删除当前 Spy 日志的全部内容
3	Font Size	调整 Spy 日志的字体大小
4	COPY	复制日志
5	Display Log	实时显示已执行的 SDK 函数及参数

参考

- 第 10 章系统函数
- 第 11 章动作设置函数
- 第 12 章配方函数
- 第 13 章动作函数

8.4 主题



项目

编号	名称	说明
1	主题更改功能	浅色 / 深色主题更改
2	浅色主题	浅色背景在深色文本使用执行主题
3	深色主题	深色背景在浅色文本使用执行主题

第 9 章 SDK 数据结构

9.1 Define

本章说明 SDK 中定义的常量值。若输入超出这些常量范围的值，可能导致程序错误。

名称	值	说明
MAX_JOINT_COUNT	20	最大关节数量
MAX_FINGER_COUNT	5	最大手指数量
MAX_FINGER_JOINT_COUNT	4	手指每最大关节数量
MAX_BASE_JOINT_COUNT	2	基座最大关节数量
CARTESIAN_COORDINATE_POSE_COUNT	6	笛卡尔坐标系数据数量 (x, Y, Z, Rx, RY, RZ)
MAX_RECEIVED_DATA_SIZE	1024	最大通信数据大小
MAX_GRIPPER_IP_ADDRESS_SIZE	32	夹爪 IP 大小
MAX_GRIPPER_IP_BYTE_LENGTH	4	夹爪 IP 地址区域数量
MAX_COMPORT_NAME_SIZE	32	COM 端口名称大小
MAX_BLEND_COUNT	10	混合动作保存数量
MAX_BLEND_ADD_POSE_COUNT	50	单个混合动作保存配方数量
MAX_RECIPE_POSE_COUNT	100	配方位姿保存数量
MAX_RECIPE_GAIN_COUNT	20	配方增益保存数量
MAX_RECIPE_GRASP_COUNT	20	配方抓取保存数量
MAX_GRIPPER_GPIO_SIZE	4	GPIO 数量(Output 3 个, Input 1 个)
MAX_RECEIVED_DATA_TYPE_COUNT	7	接收可以接收有数据类型数量

9.2 枚举类型

9.2.1 enum DG_RESULT

值	名称	说明
0	DG_RESULT_NONE	正常
其他	ERROR	附录 2 错误列表参考

9.2.2 enum DG_MODEL

夹爪型号设置用于枚举类型常量

值	名称	说明
0x0000	DG_MODEL_NONE	型号预设置
0x1F02	DG_MODEL_DG_1F_M	M 型号 1-Finger
0x2F02	DG_MODEL_DG_2F_M	M 型号 2-Finger
0x3F01	DG_MODEL_DG_3F_B	B 型号 3-Finger
0x3F02	DG_MODEL_DG_3F_M	M 型号 3-Finger
0x4F02	DG_MODEL_DG_4F_M	M 型号 4-Finger
0x5F12	DG_MODEL_DG_5F_LEFT	M 型号 5-Finger 左手
0x5F22	DG_MODEL_DG_5F_RIGHT	M 型号 5-Finger 右手
0x5F14	DG_MODEL_DG_5F_S_LEFT	S 型号 20 自由度 5-Finger 左手
0x5F24	DG_MODEL_DG_5F_S_RIGHT	S 型号 20 自由度 5-Finger 右手
0x5F34	DG_MODEL_DG_5F_S15_LEFT	S 型号 15 自由度 5-Finger 左手
0x5F34	DG_MODEL_DG_5F_S15_RIGHT	S 型号 15 自由度 5-Finger 右手

9.2.3 enum BLEND_MOTION_STATUS

用于确认混合动作状态的枚举常量。

值	名称	说明
0	BLEND_MOTION_STATUS_STOP	混合动作停止状态
1	BLEND_MOTION_STATUS_RUN	混合动作状态
2	BLEND_MOTION_STATUS_COMPLETE	混合动作完成状态

9.2.4 enum COMMUNICATION_MODE

夹爪通信模式设置器用于枚举类型常量

值	名称	说明
0	COMMUNICATION_MODE_ETHERNET	以太网通信连接
1	COMMUNICATION_MODE_RS485	串口通信连接

9.2.5 enum CONTROL_MODE

夹爪控制模式设置器用于枚举类型常量

值	名称	说明
0	CONTROL_MODE_OPERATOR	用户模式
1	CONTROL_MODE_DEVELOPER	开发者模式

9.2.6 enum GAIN_MODE

夹爪 PID 控制模式设置器用于枚举类型常量

值	名称	说明
0	GAIN_MODE_PD	PD 控制模式
1	GAIN_MODE_PID	PID 控制模式

9.2.7 enum DEVELOPER_MODE_COMMAND

夹爪用户模式不支持命令的枚举类型常量

值	名称	说明
1	DEVELOPER_MODE_COMMAND_GET_DATA	设置的数据接收
2	DEVELOPER_MODE_COMMAND_SET_RECEIVED_DATA	接收可以数据设置
5	DEVELOPER_MODE_COMMAND_SET_DUTY	关节占空比输入
6	DEVELOPER_MODE_COMMAND_SET_GPIO	GPIO(Output) 设置
7	DEVELOPER_MODE_COMMAND_SET_IP_ADDRESS	夹爪 IP 及 PORT 设置
8	DEVELOPER_MODE_COMMAND_GET_ID_AND_VERSION	夹爪 ID 及固件版本接收
9	DEVELOPER_MODE_COMMAND_GET_JOINT_ID	夹爪模块上 ID 接收
10	DEVELOPER_MODE_COMMAND_SET_BOOT_MODE	启动模式进入
11	DEVELOPER_MODE_COMMAND_SET_FT_OFFSET_ZERO	指尖传感器偏移设置

9.2.8 enum DEVELOPER_MODE_RECEIVED_DATA_TYPE

用于选择在夹爪开发者模式下可接收的数据类型的枚举常量。

值	名称	说明
1	DEVELOPER_MODE_RECEIVED_DATA_TYPE_JOINT	关节位置信息
2	DEVELOPER_MODE_RECEIVED_DATA_TYPE_CURRENT	关节电流信息
3	DEVELOPER_MODE_RECEIVED_DATA_TYPE_TEMPERATURE	关节温度信息
4	DEVELOPER_MODE_RECEIVED_DATA_TYPE_VELOCITY	关节速度信息
5	DEVELOPER_MODE_RECEIVED_DATA_TYPE_FINGER_FT_SENSOR	F/T 传感器信息
6	DEVELOPER_MODE_RECEIVED_DATA_TYPE_GPIO	GPIO 状态
7	DEVELOPER_MODE_RECEIVED_DATA_TYPE_MODULE_ERROR_CODE	模块错误代码

9.2.9 enum DG_GRASP_MODE

夹爪抓取模式设置用于枚举类型常量

DG_GRASP_MODE_2F DG-2F-M 型号仅使用可用

DG_GRASP_MODE_3F DG-3F-B, DG-3F-M 型号仅使用可用

DG_GRASP_MODE_4F DG-4F-M 型号仅使用可用

DG_GRASP_MODE_5F DG-5F-R, DG-5F-L 型号仅使用可用

值	名称	说明
0	DG_GRASP_MODE_NONE	抓取模式预设置
1	DG_GRASP_MODE_3F_3FINGER	三手指中心点抓取模式
2	DG_GRASP_MODE_3F_2FINGER_1_AND_2	第 1 个, 第 2 个手指中心点抓取模式
3	DG_GRASP_MODE_3F_2FINGER_1_AND_3	第 1 个, 第 3 个手指中心点抓取模式
4	DG_GRASP_MODE_3F_2FINGER_2_AND_3	第 2 个, 第 3 个手指中心点抓取模式
5	DG_GRASP_MODE_3F_3FINGER_PARALLEL	三手指平行抓取模式
6	DG_GRASP_MODE_3F_3FINGER_ENVELOP	三手指包络抓取模式
21	DG_GRASP_MODE_5F_5FINGER	五手指中心点抓取模式
22	DG_GRASP_MODE_5F_3FINGER	三手指中心点抓取模式
23	DG_GRASP_MODE_5F_3FINGER_PARALLEL	三手指平行抓取模式
24	DG_GRASP_MODE_5F_2FINGER_1_AND_2	第 1 个, 第 2 个手指中心点抓取模式
25	DG_GRASP_MODE_5F_2FINGER_1_AND_3	第 1 个, 第 3 个手指中心点抓取模式

26	DG_GRASP_MODE_5F_2FINGER_1_AND_4	第 1 个, 第 4 个手指 中心点抓取模式
27	DG_GRASP_MODE_5F_2FINGER_1_AND_5	第 1 个, 第 5 个手指 中心点抓取模式
28	DG_GRASP_MODE_5F_5FINGER_PARALLEL	五手指平行抓取模式
29	DG_GRASP_MODE_5F_5FINGER_ENVELOP	五手指包络抓取抓取模 式
31	DG_GRASP_MODE_4F_4FINGER	四手指中心点抓取模式
32	DG_GRASP_MODE_4F_4FINGER_PARALLEL	四手指平行抓取模式
33	DG_GRASP_MODE_4F_4FINGER_ENVELOP	四手指包络抓取抓取模 式
34	DG_GRASP_MODE_4F_4FINGER_RIGHT_PARALLEL	右手形状平行抓取模式
35	DG_GRASP_MODE_4F_4FINGER_LEFT_PARALLEL	左手形状平行抓取模式
36	DG_GRASP_MODE_4F_2FINGER_1_AND_2	第 1 个, 第 2 个手指 中心点抓取模式
37	DG_GRASP_MODE_4F_2FINGER_1_AND_4	第 1 个, 第 4 个手指 中心点抓取模式
38	DG_GRASP_MODE_4F_2FINGER_3_AND_4	第 3 个, 第 4 个手指 中心点抓取模式
51	DG_GRASP_MODE_2F_2FINGER	第 1 个, 第 2 个手指 中心点抓取模式
52	DG_GRASP_MODE_2F_2FINGER_ENVELOP	两手指包络抓取抓取模 式

9.2.10 enum DG_GRASP_OPTION

夹爪抓取选项设置用于枚举类型常量（当前不支持功能）

值	名称	说明
0	DG_GRASP_OPTION_NONE	抓取选项预设置
1	DG_GRASP_OPTION_PARALLEL	后续添加计划
2	DG_GRASP_OPTION_FIX_TILT	后续添加计划
3	DG_GRASP_OPTION_SET_TILT	后续添加计划

9.2.11 enum DG_SENSOR_TYPE

指尖传感器类型区分用于枚举类型常量

值	名称	说明
1	DG_SENSOR_TYPE_FT_SENSOR_6_AXIS	6 轴力/扭矩传感器
2	DG_SENSOR_TYPE_FT_SENSOR_3_AXIS	3 轴力/扭矩传感器
3	DG_SENSOR_TYPE_TACTILE_M	M 系列触觉传感器
4	DG_SENSOR_TYPE_FT_SENSOR_4_AXIS	4 轴力/扭矩传感器
5	DG_SENSOR_TYPE_TACTILE_S	S 系列触觉传感器

9.3 结构体

9.3.1 struct ReceivedGripperData

用于确认夹爪提供数据的结构体

数据类型	名称	大小 (byte)	说明
float	joint	80	夹爪关节的位置信息
int	current	80	夹爪关节的电流信息
int	velocity	80	夹爪关节的速度信息 (DG-3F-B 型号支持不支持)
float	temperature	80	夹爪关节的温度信息 (DG-3F-B 型号支持不支持)
int	moving	4	夹爪所有关节的运动状态信息
int	targetArrived	4	夹爪所有关节的目标位置到达状态信息
int	blendMoveState	4	混合关节功能使用时状态信息
int	currentBlendIndex	4	当前执行正在混合索引信息
int	productID	4	夹爪 ID
int	firmwareVersion	4	夹爪固件版本
int	moduleErrorCode	4	模块错误代码

参考资料

数据类型 x MAX_JOINT_COUNT= 80 byte. joint, current, velocity, temperature
9.2.3 enum BLEND_MOTION_STATUS 参考

9.3.2 Struct Recipe Blend Data

混合动作使用用于配方设置结构体

数据类型	名称	大小 (byte)	说明
int	recipePoseNumber	4	配方位姿编号
int	recipeGainNumber	4	配方增益编号
int	blendWaitTime	4	动作完成后等待时间设置
int	number	4	混合动作唯一编号

9.3.3 Struct Gripper System Setting

夹爪通信连接设置用于结构体

数据类型	名称	大小 (byte)	说明
char	comport	32	RS485 连接时 COM 端口设置
char	ip	32	TCP/IP 连接时 IP 设置
int	port	4	TCP/IP 连接时 PORT 设置
int	readTimeout	4	通信超时时间设置
int	controlMode	4	夹爪控制模式设置
int	communicationMode	4	夹爪通信模式设置
int	slaveID	4	Modbus 连接时 Slave ID 设置
int	baudrate	4	RS485 连接时 baudrate 设置

9.3.4 Struct Gripper Setting

夹爪驱动用于默认设置结构体

数据类型	名称	大小 (byte)	说明
float	jointOffset	80	夹爪关节的位置偏移设置
float	jointInpose	80	夹爪关节的到达范围确认用于各也设置
float	tcpInpose	20	指尖位置到达范围确认用于距离设置
float	orientationInpose	20	指尖方向到达范围确认用于各也设置
int	receivedDataType	24	设置接收数据类型
float	movingInpose	4	夹爪关节的运动判断用于各也设置
int	jointCount	4	当前使用正在夹爪型号的关节数量
int	fingerCount	4	当前使用正在夹爪型号的手指数量
DG_MODEL	model	4	当前使用正在夹爪型号设置
int8_t	dutyByteLength	1	当前型号在接收数据大小

参考资料

数据类型 x MAX_JOINT_COUNT= 80 byte. jointOffset, jointInpose
数据类型 x MAX_FINGER_COUNT = 20 byte. tcpInpose, orientationInpose
数据类型 x MAX_RECEIVED_DATA_TYPE_COUNT = 24 byte. receivedDataType
9.2.2 enum DG_MODEL 参考

9.3.5 Struct Recipe Pose Data

配方位姿数据设置结构体

数据类型	名称	大小 (byte)	说明
float	targetJoint	80	目标关节位置值设置
int	jointMotionTime	80	设置到达目标位置所需的执行时间
int	number	4	配方位姿唯一编号
int	mode	4	位置控制模式设置(预使用)

参考资料

数据类型 x MAX_JOINT_COUNT= 80 byte. targetJoint, jointMotionTime

9.3.6 struct Recipe Gain Data

配方增益数据设置结构体

数据类型	名称	大小 (byte)	说明
float	gainP	80	P 增益设置
float	gainD	80	D 增益设置
float	gainI	80	I 增益设置
float	iLimit	80	误差积分最大值设置
int	controlPIDMode	4	PID 控制模式设置
int	number	4	配方增益唯一编号
int	mode	4	位置控制模式设置(预使用)

参考资料

数据类型 x MAX_JOINT_COUNT= 80 byte. gainP, gainD, gainI, iLimit

9.2.6 enum GAIN_MODE 参考

9.3.7 Struct Recipe Grasp Data

配方抓取数据设置结构体

数据类型	名称	大小 (byte)	说明
float	graspForce	4	抓取力度设置
int	positionMode	80	关节抓取模式不是位置控制为固定
int	graspMode	4	抓取模式设置
int	graspOption	4	抓取选项设置（后续添加计划）
int	smoothGrasping	4	平滑抓取模式（后续添加计划）
int	number	4	配方抓取唯一编号

参考资料

数据类型 x MAX_JOINT_COUNT= 80 byte. positionMode

9.2.9 enum DG_GRASP_MODE

9.3.8 Struct Received Fingertip Sensor Data

用于确认夹爪指尖传感器数据的结构体。

数据类型	名称	大小 (byte)	说明
int	sensorType	4	连接的传感器类型
int	attachedFinger	20	传感器附着的手指位置数组
float	forceTorque	120	夹爪指尖传感器 Force / Torque 信息
uint16_t	tactile	180	触觉传感器信息

参考资料

数据类型 x MAX_FINGER_COUNT = 20 byte. attachedFinger

数据类型 x MAX_FINGER_COUNT x 6(传感器数据数量) = 120 byte. forceTorque

数据类型 x MAX_FINGER_COUNT x 18(传感器数据数量) = 180 byte. tactile

9.3.9 Struct Received GPIO Data

用于确认夹爪 GPIO 数据的结构体。

数据类型	名称	大小 (byte)	说明
int	GPIO	16	夹爪 GPIO 数据 (Output 1,2,3 / Input 1)

参考资料

数据类型 x MAX_GRIPPER_GPIO_SIZE = 16 byte. GPIO

第 10 章 系统函数

10.1 系统设置函数

10.1.1 Set Gripper System

功能说明

夹爪和通信连接进行设置值输入执行函数。该函数最优先执行需要，并，该函数执行未未执行时其他所有函数执行。

参数

数据类型	变量	说明
struct	GripperSystemSetting	9.3.3 struct GripperSystemSetting 参考

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
// 用户模式 Modbus RTU 连接时
GripperSystemSetting setting;
setting.communicationMode = COMMUNICATION_MODE_RS485;
setting.controlMode = CONTROL_MODE_OPERATOR;
char port[MAX_COMPORT_NAME_SIZE] = "COM1"; // 连接可用的 COM 端口输入
memcpy(setting.comport, port, MAX_COMPORT_NAME_SIZE);
setting.baudrate = 115200; // 夹爪在设置的 Baudrate 输入
setting.slaveID = 1; // 夹爪在设置的 Slave ID 输入
DG_RESULT result = SetGripperSystem(setting);

// 用户模式 Modbus TCP 连接时
GripperSystemSetting setting;
setting.communicationMode = COMMUNICATION_MODE_ETHERNET;
setting.controlMode = CONTROL_MODE_OPERATOR;
char ip[MAX_GRIPPER_IP_ADDRESS_SIZE] = "169.254.186.72"; // 夹爪 IP 地址输入
memcpy(setting.ip, ip, MAX_GRIPPER_IP_ADDRESS_SIZE);
setting.port = 502; // 夹爪在设置的 Port 输入
setting.slaveID = 1; // 夹爪在设置的 Slave ID 输入
setting.readTimeout= 60000; // 夹爪的通信超时时间输入
DG_RESULT result = SetGripperSystem(setting);

// 开发者模式 TCP/IP 连接时
GripperSystemSetting setting;
setting.communicationMode = COMMUNICATION_MODE_ETHERNET;
setting.controlMode = CONTROL_MODE_DEVELOPER;
char ip[MAX_GRIPPER_IP_ADDRESS_SIZE] = "169.254.186.72"; // 夹爪 IP 地址输入
memcpy(setting.ip, ip, MAX_GRIPPER_IP_ADDRESS_SIZE);
setting.port = 502; // 夹爪在设置的 Port 输入
setting.slaveID = 1; // 夹爪在设置的 Slave ID 输入
DG_RESULT result = SetGripperSystem(setting);
```

10.1.2 SetGripperOption

功能说明

该选项用于设置夹爪动作的默认值。若未执行该选项，将无法控制夹爪。

参数

数据类型	变量	说明
GripperSetting	setting	9.3.4 struct GripperSetting 参考

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
// DG-3F-M 设置时  
float offset[MAX_JOINT_COUNT] = { 0 };  
float inpose[MAX_JOINT_COUNT] = { 0 };  
GripperSetting gs;  
offset[0] = 2; // 第 1 个关节偏移 2也设置  
inpose[2] = 3; // 第 3 个关节的位姿 3也设置  
gs.model = DG_MODEL_DG_3F_M; // 夹爪型号设置  
memcpy(gs.jointOffset, offset, sizeof(offset));  
memcpy(gs.jointInpose, inpose, sizeof(inpose));  
gs.movingInpose = 0.3f;  
DG_RESULT result = SetGripperOption(gs);
```

10.1.3 Connect To Gripper

功能说明

尝试与夹爪建立通信连接。InitializedCallback 函数中包含的回调函数项，可在 “[1.0.2 回调](#)” 中确认详细命令。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
void InitializedCallback(){
    CallbackForOnConnected(*OnConnected);
    CallbackForOnDisconnected(*OnDisconnected);
    CallbackForOnReceivedGripperData(*RecvGripperData);
    CallbackForOnCommunicationPeriod(*OnCountPeriod);
    CallbackForOnReceivedFingertipSensorData(*RecvSensorData);
    CallbackForOnReceivedGPIOData(*RecvGPIOData);
}
InitializedCallback();
DG_RESULT result = ConnectToGripper();
```

10.1.4 Disconnect To Gripper

功能说明

尝试断开与夹爪的通信连接。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++
`DG_RESULT result = DisconnectToGripper();`

10.1.5 System Start

功能说明

启动系统。必须执行该功能后，才能接收夹爪数据并控制夹爪。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++
`DG_RESULT result = SystemStart();`

10.1.6 System Stop

功能说明

系统停止。该功能执行后夹爪的数据接收未且不会，控制也执行。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
DG_RESULT result = SystemStop();
```

10.1.7 Set Ip

功能说明

夹爪 IP和 Port 设置。

参数

数据类型	变量	说明
int*	ip	更改可以 ip 输入
int	port	更改可以 port 输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
int ip[4] = { 169, 254, 186, 72 }; // ip 地址输入
int port = 502; // port 输入
DG_RESULT result = SetIp(ip, port);
```

10.1.8 Set Joint Encoder Zero

功能说明

将夹爪当前位置设置为零点。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
    DG_RESULT result = SetJointEncoderZero();
```

10.2 回调

10.2.1 Call back For On Connected

功能说明

夹爪和的通信连接在关于事件接收。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
static void OnConnected()
{
    std::cout << "Connected gripper server" << std::endl;
}
```

```
DG_RESULT result = CallbackForOnConnected(*OnConnected);
```

10.2.2 CallbackForOnDisconnected

功能说明

夹爪和的通信断开在关于事件接收。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
static void OnDisconnected()  
{  
    std::cout << "Disconnected gripper server" << std::endl;  
}  
  
DG_RESULT result = CallbackForOnDisconnected(*OnDisconnected);
```

10.2.3 Callback For On Received Gripper Data

功能说明

从夹爪接收数据。

参数

数据类型	变量	说明
struct	ReceivedGripperData	9.3.1 struct ReceivedGripperData 参考

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
ReceivedGripperData gripperData;
static void RecvGripperData(const ReceivedGripperData data)
{
    gripperData = data;
}
DG_RESULT result = CallbackForOnReceivedGripperData(*RecvGripperData);
```

10.2.4 Callback For On Communication Period

功能说明

接收夹爪的通信周期。该值表示 1 秒内夹爪与 PC 之间进行通信交互的次数。根据与夹爪连接的控制器规格，该值可能会降低；当该值降低时，可能无法顺畅执行控制。

参数

数据类型	变量	说明
int	period	通信周期

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int countPeriod = 0;
static void OnCountPeriod(const int period)
{
    countPeriod = period;
}
DG_RESULT result = CallbackForOnCommunicationPeriod(*OnCountPeriod);
```

10.2.5 Call back For On Received Fingertip Sensor Data

功能说明

夹爪指尖传感器数据接收。

参数

数据类型	变量	说明
struct	ReceivedFingertipSensorData	指尖传感器数据

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
ReceivedFingertipSensorData sensorData;  
static void RecvSensorData(const ReceivedFingertipSensorData recvSensorData)  
{  
    sensorData = recvSensorData;  
}  
DG_RESULT result = CallbackForOnReceivedFingertipSensorData(*RecvSensorData);
```

10.2.6 Call back For On Received GPIO Data

功能说明

夹爪 GPIO 数据接收。

参数

数据类型	变量	说明
struct	ReceivedGPIOData	GPIO 数据

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
ReceivedGPIOData gpioData;  
static void RecvGPIOData(const ReceivedGPIOData recvGPIOData)  
{  
    gpioData = recvGPIOData;  
}  
DG_RESULT result = CallbackForOnReceivedGPIOData(*RecvGPIOData);
```

10.3 其他系统函数

10.3.1 Set Low Pass Filter Alpha

功能说明

低通滤波器增益设置。设置增益值较低越平滑运动表现。

参数

数据类型	变量	说明
int	isUsed	滤波器使用设置
float	alpha	设置滤波器增益值

返回值

值	说明
DG_RESULT_NONE	正常
DG_RESULT_SYSTEM_SETTING_NOT_PERFORMED	系统设置预执行
DG_RESULT_DATA_IS_NOT_BOOLEAN	0 和 1 仅输入可用
DG_RESULT_DATA_IS_NEGATIVE	负可以值输入加载

示例

C/C++

```
float alpha = 0.1; // 设置增益值
int isUsed = 1; // 滤波器 On/Off
DG_RESULT result = SetLowPassFilterAlpha(isUsed, alpha);
```

10.3.2 Set Boot Mode

功能说明

用于进入固件下载启动模式的功能。进入启动模式需要密码，请联系 support@tesollo.com。

参数

数据类型	变量	说明
char*	password	启动模式进入用于密码输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
DG_RESULT result = SetBootMode(password);
```

10.3.3 Set GPIO Out put

功能说明

夹爪 GPIO 输出设置。

参数

数据类型	变量	说明
int	gpio	输出值设置
int	outputNumber	输出编号设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
int gpio = 1; // output true 输出
int outputNumber= 1; // 输出编号设置
DG_RESULT result = SetGPIOOutput(gpio, outputNumber);
```

10.3.4 Set GPIO Out put All

功能说明

夹爪全部 GPIO 输出设置。

参数

数据类型	变量	说明
int*	gpio	输出值设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int gpio[3] = { 1, 1, 1 }; // output true 输出  
DG_RESULT result = SetGPIOOutputAll(gpio);
```

10.3.5 Set Torque Limit Mode

功能说明

夹爪的扭矩限制模式设置。

参数

数据类型	变量	说明
int	isOn	扭矩限制模式设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int isOn = 0; // 扭矩限制模式断开
DG_RESULT result = SetTorqueLimitMode(isOn);
```

参考

注意！在用户模式下将 Torque limit mode 设为 off，或在开发者模式下使用 stall torque 时，可能导致产品温度快速上升及产品损坏；发生故障时可能无法享受售后服务。

10.3.6 Set Boot Recipe

功能说明

设置夹爪启动时应用的配方。

参数

数据类型	变量	说明
int	recipePoseNumber	位姿配方编号设置
int	recipeGainNumber	位姿增益编号设置
int	recipeGraspNumber	位姿抓取编号设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
DG_RESULT result = SetBootRecipe(1,1,1); // 将第 1 个位姿配方、第 1 个增益配方、第 1 个抓取配方编号设置为启动时应用的配方
```

10.3.7 EEPROM Write

功能说明

将当前设置值保存至 EEPROM。用户模式下保存至夹爪的 EEPROM，开发者模式下保存至 PC 存储。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
DG_RESULT result = EEPROMWrite();// 当前设置的值 EEPROM在保存
```

10.3.8 Back up Recipe Data

功能说明

该功能用于将夹爪中应用的数据备份并保存为 PC 文件。

参数

数据类型	变量	说明
char*	path	备份保存路径

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
char path[] = "C:\\data\\recipe_backup";  
DG_RESULT result = BackupRecipeData(path);
```

10.3.9 Restore Recipe Data

功能说明

功能用户模式(Operator Mode)仅在支持。指过程的路径的备份文件在数据读取连接的夹爪为数据传输。

参数

数据类型	变量	说明
char*	path	恢复文件路径

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
char path[] = "C:\\data\\recipe_restore";
DG_RESULT result = RestoreRecipeData(path);
```

10.4 数据查询

10.4.1 Get Received Gripper Data

功能说明

获取从夹爪接收的数据。

参数

数据类型	变量	说明
struct*	receivedGripperData	接收数据结构体变量

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
ReceivedGripperData receivedGripperData;  
DG_RESULT result = GetReceivedGripperData(&receivedGripperData);
```

相关函数

[10.2.3 Callback For On Received Gripper Data](#)

10.4.2 Get Communication Period

功能说明

夹爪和的通信周期获取.

参数

数据类型	变量	说明
int*	period	通信周期接收用变量

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++ <code>int period; DG_RESULT result = GetCommunicationPeriod(&period);</code>
--

相关函数

10.2.4 Callback For On Communication Period

10.4.3 Get Received Fingertip Sensor Data

功能说明

夹爪手指尖端在附着的力扭矩传感器数据获取.

参数

数据类型	变量	说明
struct*	receivedFingertipData	指尖数据接收结构体变量

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
ReceivedFingertipSensorData receivedFingertipData;  
DG_RESULT result = GetReceivedFingertipSensorData(&receivedFingertipData);
```

相关函数

10.2.5 [Call back For On Received Fingertip Sensor Data](#)

10.4.4 Get Received GPIO Data

功能说明

夹爪 GPIO 数据获取.

参数

数据类型	变量	说明
struct*	receivedGPIOData	GPIO 接收结构体变量

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
ReceivedGPIOData receivedGPIOData;
DG_RESULT result = GetReceivedGPIOData(&receivedGPIOData);
```

相关函数

[10.2.6 Call back For On Received GPIO Data](#)

10.4.5 Get Library Version

功能说明

当前使用正在库版本获取.

参数

数据类型	变量	说明
int*	version	版本信息复制到数组

示例

C/C++

```
int version[3] = {0,0,0}; // 库版本 : major, minor, patch 顺序为保存
GetLibraryVersion(version);
std::cout << "Received Library Version : major : " << version[0]
            << ", minor : " << version[1]
            << ", patch : " << version[2]
            << std::endl;
```

第 11 章 动作设置函数

11.1 抓取设置函数

力控制基于抓取算法设置。 5 指, 3 指, 2 指, 平行, 包络抓取等抓取模式提供, 并且, 抓取选项, 位置控制模式等通过使用多种方式抓取动作执行。

11.1.1 Set Grasp Data

功能说明

为执行抓取动作而设置抓取数据。

参数

数据类型	变量	说明
int	graspMode	抓取模式设置
float	graspForce	抓取力度设置

int	graspOption	抓取选项设置
int	smoothGrasping	平滑抓取使用

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
#define IS_ON 1
DG_GRASP_MODE graspMode = DG_GRASP_MODE_3F_3FINGER; // 三手指中心点抓取模式
float graspForce = 3.0;
int graspOption = DG_GRASP_OPTION_NONE; // 抓取选项预使用
int smoothGrasping = IS_ON; // 平滑抓取使用
DG_RESULT result = SetGraspData(graspMode, graspForce, graspOption,
smoothGrasping);
```

相关函数

13.2 Start Grasp Motion

11.1.2 SetGraspForce

功能说明

更改抓取力度。该函数可在抓取过程中仅调整抓取力度，无需重新进行其他设置。

参数

数据类型	变量	说明
float	graspForce	抓取力度设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++ float graspForce = 5.0; DG_RESULT result = SetGraspForce(graspForce);
--

相关函数

13.2 Start Grasp Motion

11.2 P 增益设置

夹爪 P 增益设置。P 增益设置值较高时目标位置在快速到达可以有，但，夹爪抖动现象可能出现。

11.2.1 Set Joint GainP

功能说明

单个关节的 P 增益值设置。

参数

数据类型	变量	说明
float	gainP	P 增益输入
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float gainP = 2.0;  
int jointNumber = 1;  
DG_RESULT result = SetJointGainP(gainP, jointNumber);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.2.2 Set Joint GainP Finger

功能说明

单个手指关节项的 P 增益值设置。

参数

数据类型	变量	说明
float*	gainP	P 增益输入
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float gainP[MAX_FINGER_JOINT_COUNT] = { 3.0, 3.0, 3.0, 3.0 };  
int fingerNumber = 3;  
DG_RESULT result = SetJointGainPFinger(gainP, fingerNumber);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.2.3 Set Joint GainP All

功能说明

所有关节的 P 增益值设置。

参数

数据类型	变量	说明
float*	gainP	P 增益输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float gainP[MAX_JOINT_COUNT] = { 0 };  
gainP[0] = gainP[1] = 1; // 第 1 个个, 第 2 个个关节增益设置  
gainP[4] = gainP[6] = 2; // 第 5 个个, 第 7 个个关节增益设置  
DG_RESULT result = SetJointGainPAll(gainP);
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.2.4 Set Joint GainP Base

功能说明

基座关节的 P 增益值设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
float*	gainP	P 增益输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainP[MAX_BASE_JOINT_COUNT] = { 0 };  
gainP[0] = 1; // 第 17 个关节, 第 1 个手指对应的基座关节增益设置  
gainP[1] = 2; // 第 18 个关节, 第 4 个手指对应的基座关节增益设置  
DG_RESULT result = SetJointGainPBase(gainP);
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.3 D 增益设置

夹爪 D 增益设置。D 增益 P 增益为的的超调可防止，并且系统的稳定性提高。但设置值高时夹爪抖动现象可能出现。

11.3.1 Set Joint GainD

功能说明

单个关节的 D 增益值设置。

参数

数据类型	变量	说明
float	<code>gainD</code>	D 增益输入
int	<code>jointNumber</code>	关节编号输入

返回值

编号	值	说明
0	<code>DG_RESULT_NONE</code>	正常
0 不是时	<code>ERROR</code>	附录 2 错误列表参考

示例

```
C/C++  
  
float gainD = 2.0;  
int jointNumber = 1;  
DG_RESULT result = SetJointGainD(gainD, jointNumber);
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.3.2 Set Joint GainD Finger

功能说明

单个手指的 D 增益值设置。

参数

数据类型	变量	说明
float*	gainD	D 增益输入
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float gainD[MAX_FINGER_JOINT_COUNT] = { 3.0, 3.0, 3.0, 3.0 };  
int fingerNumber = 3;  
DG_RESULT result = SetJointGainDFinger(gainD, fingerNumber);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.3.3 Set Joint GainD All

功能说明

所有关节的 D 增益值设置。

参数

数据类型	变量	说明
float*	gainD	D 增益输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainD[MAX_JOINT_COUNT] = { 0 };  
gainD[0] = gainD[1] = 1; // 第 1 个个, 第 2 个个关节增益设置  
gainD[4] = gainD[6] = 2; // 第 5 个个, 第 7 个个关节增益设置  
DG\_RESULT result = SetJointGainDAll(gainD);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.3.4 Set Joint GainD Base

功能说明

基座关节的 D 增益值设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
float*	gainD	D 增益输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainD [MAX_BASE_JOINT_COUNT] = { 0 };  
gainD[0] = 1; // 第 17 个关节, 第 1 个手指对应的基座关节增益设置  
gainD[1] = 2; // 第 18 个关节, 第 4 个手指对应的基座关节增益设置  
DG_RESULT result = SetJointGainDBase(gainD);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.4 I 增益设置

设置夹爪的 I 增益和误差积分最大值。I 增益用于使系统对误差减小时的微小变化也能灵敏响应，从而更精确地到达目标值。I 增益值过高时，虽然可快速响应误差，但可能导致夹爪发生抖动；相反，值过低时误差补偿会变慢，误差积分值的恢复也会延迟，从而可能产生振动。误差积分最大值用于防止误差积分值快速累积；该值过低时可能无法充分补偿误差，过高时误差积分值的恢复速度可能变慢。

I 增益会持续累积控制信号，直到误差变为 0。因此，当夹爪无法自行恢复误差时，可能会因关节过载而发生故障。为实现稳定控制，需要将其精确调试为适当的值，使用时也需特别注意。建议以 0.01 为单位调试 I 增益值和误差积分最大值。

11.4.1 Set Joint GainI

功能说明

单个关节的 I 增益值和误差积分最大值设置。

参数

数据类型	变量	说明
float	gainI	I 增益输入
float	iLimit	误差积分最大值设置
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainI = 0.015;  
float iLimit =0.06;  
int jointNumber = 1;  
DG_RESULT result = SetJointGainI (gainI, iLimit, jointNumber);
```

相关函数

[12.2 配方增益函数](#)
[13.3 位置控制](#)

11.4.2 Set Joint GainI Finger

功能说明

单个手指的 I 增益值和误差积分最大值设置。

参数

数据类型	变量	说明
float*	gainI	I 增益输入
float*	iLimit	误差积分最大值设置
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainI[MAX_FINGER_JOINT_COUNT] = { 0.015, 0.015, 0.015, 0.015 };
float iLimit[MAX_FINGER_JOINT_COUNT] = { 0.06, 0.06, 0.06, 0.06 };
int fingerNumber = 1;
DG_RESULT result = SetJointGainIFinger(gainI, iLimit, fingerNumber);
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.4.3 Set Joint GainI All

功能说明

所有关节的 I 增益值和误差积分最大值设置。

参数

数据类型	变量	说明
float*	gainI	I 增益输入
float*	iLimit	误差积分最大值设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++

float gainI[MAX_JOINT_COUNT] = { 0 };
float iLimit[MAX_JOINT_COUNT] = { 0 };
gainI[0] = gainI[1] = 0.015; // 第 1 个个, 第 2 个个关节增益设置
iLimit[0] = iLimit[1] = 0.06; // 第 1 个个, 第 2 个个关节误差积分最大值设置
DG_RESULT result = SetJointGainIAll(gainI, iLimit);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.4.4 Set Joint GainI Base

功能说明

基座关节的 I 增益值和误差积分最大值设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
float*	gainI	I 增益输入
float*	iLimit	误差积分最大值设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float gainI[MAX_JOINT_COUNT] = { 0 };  
float iLimit[MAX_JOINT_COUNT] = { 0 };  
gainI[0] = gainI[1] = 0.015; // 第 17 个, 第 18 个关节对应的基座关节增益设置  
iLimit[0] = iLimit[1] = 0.06; // 第 17 个, 第 18 个关节对应的基座关节误差积分最大值设置  
DG_RESULT result = SetJointGainIAll(gainI, iLimit);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.5 PID 增益设置

11.5.1 Se tControl PID Mode

功能说明

设置夹爪控制器。该函数可将夹爪控制器设置为 PD 模式或 PID 模式。

参数

数据类型	变量	说明
int	mode	增益模式设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int mode = GAIN_MODE_PD; // 设置为 PD 增益模式  
DG_RESULT result = SetControlPIDMode(mode);
```

相关函数

- [12.2 配方增益函数](#)
- [13.3 位置控制](#)

11.5.2 Set Joint Gain PID

功能说明

单个关节的 P, I, D 增益及误差积分最大值设置。

参数

数据类型	变量	说明
float	gainP	P 增益输入
float	gainD	D 增益输入
float	gainI	I 增益输入
float	iLimit	误差积分最大值设置
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
float gainP = 1.0;
float gainD = 5.0;
float gainI = 0.015;
float iLimit = 0.06;
int jointNumber = 1;
DG_RESULT result = SetJointGainPID (gainP, gainD, gainI, iLimit, jointNumber);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.5.3 Set Joint Gain PID Finger

功能说明

单个手指的 P, I, D 增益及误差积分最大值设置。

参数

数据类型	变量	说明
float*	gainP	P 增益输入
float*	gainD	D 增益输入
float*	gainI	I 增益输入
float*	iLimit	误差积分最大值设置
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float gainP[MAX_FINGER_JOINT_COUNT] = { 1.0, 1.0, 1.0, 1.0 };  
float gainD[MAX_FINGER_JOINT_COUNT] = { 5.0, 5.0, 5.0, 5.0 };  
float gainI[MAX_FINGER_JOINT_COUNT] = { 0.015, 0.015, 0.015, 0.015 };  
float iLimit[MAX_FINGER_JOINT_COUNT] = { 0.06, 0.06, 0.06, 0.06 };  
int fingerNumber = 1;  
DG_RESULT result = SetJointGainPIDFinger(gainP, gainD, gainI, iLimit, fingerNumber);
```

相关函数

12.2 配方增益函数
13.3 位置控制

11.5.4 Set Joint Gain PID All

功能说明

所有关节的 P, I, D 增益及误差积分最大值设置。

参数

数据类型	变量	说明
float*	gainP	P 增益输入
float*	gainD	D 增益输入
float*	gainI	I 增益输入
float*	iLimit	误差积分最大值设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
float gainP[MAX_JOINT_COUNT] = { 0 };
float gainD[MAX_JOINT_COUNT] = { 0 };
float gainI[MAX_JOINT_COUNT] = { 0 };
float iLimit[MAX_JOINT_COUNT] = { 0 };
gainP[0] = gainP[1] = 1; // 第 1 个个, 第 2 个个关节增益设置
gainD[0] = gainD[1] = 5; // 第 1 个个, 第 2 个个关节增益设置
gainI[0] = gainI[1] = 0.015; // 第 1 个个, 第 2 个个关节增益设置
iLimit[0] = iLimit[1] = 0.06; // 第 1 个个, 第 2 个个关节误差积分最大值设置
DG_RESULT result = SetJointGainPIDAll(gainP, gainD, gainI, iLimit);
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.5.5 Set Joint Gain PID All Equal

功能说明

所有关节的 P, I, D 增益及误差积分最大值批量设置。

参数

数据类型	变量	说明
float	gainP	P 增益输入
float	gainD	D 增益输入
float	gainI	I 增益输入
float	iLimit	误差积分最大值设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainP = 1.0;
float gainD = 5.0;
float gainI = 0.015;
float iLimit = 0.06;
DG_RESULT result = SetJointGainPIDAllEqual(gainP, gainD, gainI, iLimit);
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.5.6 Set Joint Gain PID Base

功能说明

基座关节的 P, I, D 增益及误差积分最大值设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
float*	gainP	P 增益输入
float*	gainD	D 增益输入
float*	gainI	I 增益输入
float*	iLimit	误差积分最大值设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
float gainP[MAX_BASE_JOINT_COUNT] = { 0 };
float gainD[MAX_BASE_JOINT_COUNT] = { 0 };
float gainI[MAX_BASE_JOINT_COUNT] = { 0 };
float iLimit[MAX_BASE_JOINT_COUNT] = { 0 };
gainP[0] = gainP[1] = 1; // 第 17 个, 第 18 个关节对应的基座关节增益设置
gainD[0] = gainD[1] = 5; // 第 17 个, 第 18 个关节对应的基座关节增益设置
gainI[0] = gainI[1] = 0.015; // 第 17 个, 第 18 个关节对应的基座关节增益设置
iLimit [0] = iLimit [1] = 0.06; // 第 17 个, 第 18 个关节对应的基座误差积分最大值设置
DG_RESULT result = SetJointGainPIDBase (gainP, gainD, gainI, iLimit)
```

相关函数

[12.2 配方增益函数](#)

[13.3 位置控制](#)

11.6 动作时间设置

设置夹爪到达目标位置所需的执行时间。动作时间设置得越短，动作速度越快；设置得越长，动作速度越慢。当动作时间设置得非常短时，夹爪将以最大速度动作。该功能仅在位置控制中生效。动作时间可能会根据控制器规格而有所不同。SDK 通过内部定时器管理动作时间，但在执行 SDK 的控制器环境性能较低时，内部定时器时间可能无法得到保证。此外，根据夹爪运动设置值，较短的动作时间也可能无法得到保证。

11.6.1 Set Motion Time Joint

功能说明

单个关节的位置移动时间设置。

参数

数据类型	变量	说明
int	motionTime	位置移动时间设置(ms)
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int motionTime = 1000;
int jointNumber = 1;
DG_RESULT result = SetMotionTimeJoint(motionTime, jointNumber);
```

相关函数

[12.1 配方位姿函数](#)
[13.3 位置控制](#)

11.6.2 Set Motion Time Finger

功能说明

单个手指的位置移动时间设置。

参数

数据类型	变量	说明
int*	motionTime	位置移动时间设置
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int motionTime[MAX_FINGER_JOINT_COUNT] = { 0 };  
int fingerNumber = 1;  
motionTime[0] = 1000; // 将第 1 个手指的位置移动时间设置为 1 秒  
motionTime[2] = 1500; // 将第 3 个手指的位置移动时间设置为 1.5 秒  
DG_RESULT result = SetMotionTimeFinger(motionTime, fingerNumber);
```

相关函数

[12.1 配方位姿函数](#)
[13.3 位置控制](#)

11.6.3 Set Motion Time All

功能说明

所有关节的位置移动时间分别设置。

参数

数据类型	变量	说明
int*	motionTime	位置移动时间设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int motionTime[MAX_JOINT_COUNT] = { 0 };  
motionTime[0] = 1000; // 将第 1 个关节的位置移动时间设置为 1 秒  
motionTime[2] = 1500; // 将第 3 个关节的位置移动时间设置为 1.5 秒  
DG_RESULT result = SetMotionTimeAll(motionTime);
```

相关函数

[12.1 配方位姿函数](#)

[13.3 位置控制](#)

11.6.4 Set Motion Time All Equal

功能说明

所有关节的位置移动时间批量设置。

参数

数据类型	变量	说明
int	motionTime	位置移动时间设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++ <code>int motionTime = 1500; // 位置移动时间 1.5秒设置</code> <code>DG_RESULT result = SetMotionTimeAllEqual(motionTime);</code>
--

相关函数

12.1 配方位姿函数
13.3 位置控制

11.6.5 Set Motion Time Base

功能说明

基座关节的位置移动时间分别设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
int*	motionTime	位置移动时间设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int motionTime[MAX_JOINT_COUNT] = { 0 };  
motionTime[0] = 1000; // 将第 17 个关节的位置移动时间设置为 1 秒  
motionTime[2] = 1500; // 将第 18 个关节的位置移动时间设置为 1.5 秒  
DG_RESULT result = SetMotionTimeAll(motionTime);
```

相关函数

[12.1 配方位姿函数](#)

[13.3 位置控制](#)

11.7 位置控制模式设置

该选项在抓取动作时应用位置控制模式。被设置为位置控制模式的关节在执行抓取动作时不会参与抓取，而是执行位置控制。该功能用于在抓取过程中指定某些关节不移动，并可实现多种抓取方式。

11.7.1 Set Position Mode Joint

功能说明

单个关节的位置控制模式设置。

参数

数据类型	变量	说明
int	positionMode	位置控制模式使用设置
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
#define IS_ON 1
int positionMode = IS_ON;
int jointNumber = 1;
// 第 1 个关节位置控制仅可用以便设置
DG_RESULT result = SetPositionModeJoint(positionMode, jointNumber);
```

相关函数

12.3 配方抓取函数
13.2 Start Grasp Motion

11.7.2 Set Position Mode Finger

功能说明

单个手指关节项的位置控制模式设置执行

参数

数据类型	变量	说明
int*	positionMode	位置控制模式使用设置
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++

#define IS_OFF 0
#define IS_ON 1
int positionMode[MAX_FINGER_JOINT_COUNT] = { IS_ON, IS_ON, IS_OFF, IS_OFF };
int fingerNumber = 1;
// 第 1 个手指第 1 个, 第 2 个关节位置控制仅可用以便设置
// 第 1 个手指第 3 个, 第 4 个关节位置控制模式断开
DG_RESULT result = SetPositionModeFinger(positionMode, fingerNumber);
```

相关函数

12.3 配方抓取函数
13.2 Start Grasp Motion

11.7.3 Set Position Mode All

功能说明

所有关节项的位置控制模式设置。

参数

数据类型	变量	说明
int*	positionMode	位置控制模式使用设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
#define IS_OFF 0
#define IS_ON 1
int positionMode[MAX_JOINT_COUNT]={ IS_OFF };
positionMode[0] = IS_ON; // 第 1 个关节位置控制仅可用以便设置
positionMode[11] = IS_OFF; // 第 12 个关节位置控制模式断开
DG_RESULT result = SetPositionModeAll(positionMode);
```

相关函数

12.3 配方抓取函数
13.2 Start Grasp Motion

11.7.4 Set Position Mode Base

功能说明

基座关节项的位置控制模式设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
int*	positionMode	位置控制模式使用设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++

#define IS_OFF 0
#define IS_ON 1
int positionMode[MAX_JOINT_COUNT]={ IS_OFF };
positionMode[0] = IS_ON; // 第 17 个关节, 第 1 个手指对应的基座关节位置控制模式 ON
positionMode[11] = IS_OFF; // 第 18 个关节, 第 4 个手指对应的基座关节位置控制模式
OFF

DG_RESULT result = SetPositionModeAll(positionMode);
```

相关函数

12.3 配方抓取函数
13.2 Start Grasp Motion

11.8 指尖传感器设置

夹爪指尖在附着的传感器数据和相关的设置。该功能选项产品手指尖端传感器和一起可使用。

11.8.1 SetFingerTipDataZero

功能说明

将当前接收的指尖数据初始化为 0。再次调用时，将恢复初始化前的原始数据。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++
`DG_RESULT result = SetFingerTipDataZero();`

相关函数

[10.1.2 SetGripperOption](#)

第 12 章 配方函数

12.1 配方位姿函数

12.1.1 Update Recipe Current Pose Data

功能说明

当前夹爪关节位置信息，当前设置的动作时间输入的编号的位姿配方在保存。

参数

数据类型	变量	说明
int	recipeNumber	配方位姿编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float joint[MAX_JOINT_COUNT] = { 0 }; // 夹爪关节数量仪的时间输入  
int motionTime = 500; // 夹爪目标位置到达时间设置  
int recipeNumber = 1; // 配方编号设置  
DG_RESULT result = MoveJointAll(joint);  
if (result == DG_RESULT_NONE) result = SetMotionTimeAllEqual(motionTime);  
if (result == DG_RESULT_NONE) result = UpdateRecipeCurrentPoseData(recipeNumber);
```

相关函数

11.2.1 Set Joint GainP - 11.5.1 SetControl PID Mode
13.3 位置控制

12.1.2 Update Recipe Target Pose Data

功能说明

当前设置的夹爪关节的目标位置信息，动作时间输入的编号的位姿配方在保存。

参数

数据类型	变量	说明
int	recipeNumber	配方位姿编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float joint[MAX_JOINT_COUNT] = { 0 }; // 夹爪关节数量仅的时间输入  
int motionTime = 500; // 夹爪目标位置到达时间设置  
int recipeNumber = 1; // 配方编号设置  
DG_RESULT result = MoveJointAll(joint);  
if (result != DG_RESULT_NONE) return result;  
  
result = SetMotionTimeAllEqual(motionTime);  
if (result != DG_RESULT_NONE) return result;  
  
result = UpdateRecipeTargetPoseData(recipeNumber);  
if (result != DG_RESULT_NONE) return result;
```

相关函数

11.2.1 Set Joint GainP - 11.5.1 SetControl PID Mode
13.3 位置控制

12.1.3 Load Recipe Pose Data

功能说明

输入的编号的位姿配方信息读取动作。

参数

数据类型	变量	说明
int	recipeNumber	配方位姿编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int recipeNumber = 1; // 配方编号设置
DG_RESULT result = LoadRecipePoseData(recipeNumber);
```

相关函数

- 12.1.1 [Update Recipe Current Pose Data](#)
- 12.1.2 [Update Recipe Target Pose Data](#)

12.2 配方增益函数

12.2.1 Update Recipe Gain Data

功能说明

当前设置的 P, I, D 增益, 误差积分最大值及 PID 控制模式输入的编号的增益配方在保存。

参数

数据类型	变量	说明
int	recipeNumber	配方增益编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float gainP = 1; // P 增益设置
float gainD = 5; // D 增益设置
float gainI = 0.015; // I 增益设置
float iLimit = 0.02; // 误差积分最大值设置
int controlPIDMode = GAIN_MODE_PID; // PID 控制模式设置
int recipeNumber = 1; // 配方编号设置
DG_RESULT result = SetJointGainPIDAllEqual(gainP, gainD, gainI, iLimit);
if (result != DG_RESULT_NONE) return result;

result = SetControlPIDMode(controlPIDMode);
if (result != DG_RESULT_NONE) return result;

result = UpdateRecipeGainData(recipeNumber);
if (result != DG_RESULT_NONE) return result;
```

相关函数**12.2.2 Load Recipe Gain Data****12.2.2 Load Recipe Gain Data**

功能说明

输入的编号的增益配方数据读取设置。

参数

数据类型	变量	说明
int	recipeNumber	配方增益编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++ <pre>int recipeNumber = 1; // 配方编号设置 DG_RESULT result = LoadRecipeGainData(recipeNumber);</pre>

相关函数

12.2.1 Update Recipe Gain Data
--

12.3 配方抓取函数

12.3.1 Update Recipe Grasp Data

功能说明

当前设置的抓取选项，抓取模式，平滑抓取模式，位置控制模式，抓取力度设置值输入的抓取配方在保存。

参数

数据类型	变量	说明
int	graspNumber	配方抓取编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
float graspForce = 5; // 抓取力度设置
int graspOption = 1; // 抓取选项设置
int smoothGrasping = 1; // 平滑抓取模式设置
int positionMode[MAX_JOINT_COUNT] = { 0 }; // 夹爪关节数量仅的时间输入
int recipeNumber = 1; // 配方编号设置
DG_GRASP_MODE graspMode = DG_GRASP_MODE_3F_3FINGER; // 抓取模式设置
DG_RESULT result = SetPositionModeAll(positionMode);
if (result != DG_RESULT_NONE) return result;
result = SetGraspData(graspMode, graspForce, graspOption, smoothGrasping);

if (result != DG_RESULT_NONE) return result;
result = UpdateRecipeGraspData(recipeNumber);
```

相关函数

11.1 抓取设置函数

11.7.1 Set Position Mode

11.7.2 Set Position Mode Finger

11.7.3 Set Position Mode All

12.3.2 Load Recipe Grasp Data

功能说明

输入的编号的抓取配方数据读取设置。

参数

数据类型	变量	说明
int	graspNumber	配方抓取编号

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int recipeNumber = 1; // 配方编号设置
DG_RESULT result = LoadRecipeGraspData(recipeNumber);
if (result != DG_RESULT_NONE) return result;

result = StartGraspMotion(1);
if (result != DG_RESULT_NONE) return result;
```

相关函数

12.3.1 Update Recipe Grasp Data

12.4 配方混合函数

12.4.1 Update Blend Joint

功能说明

输入的混合动作数据的位姿配方编号，增益配方编号，动作完成后等待时间混合动作配方在添加。

参数

数据类型	变量	说明
RecipeBlendData	blendData	' 9.3.2 struct RecipeBlendData' 参考

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
RecipeBlendData blendData;
blendData.blendWaitTime = 500; // 动作完成后等待时间设置
blendData.recipeGainNumber = 1; // 增益配方编号设置
blendData.recipePoseNumber = 1; // 位姿配方编号设置
DG_RESULT result = UpdateBlendJoint(blendData);
```

相关函数

[12.1.1 Update Recipe Current Pose Data](#)

[12.1.2 Update Recipe Target Pose Data](#)

[12.2.1 Update Recipe Gain Data](#)

12.4.2 Clear Move Blend Joint

功能说明

当前添加的混合数据列表删除。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++
ClearMoveBlendJoint();

相关函数

- [12.4.1 Update Blend Joint](#)
[12.4.3 Add Move Blend Joint](#)
[12.4.4 Set Move Blend Joint](#)
[12.4.5 Start Move Blend Joint](#)
[12.4.6 Stop Move Blend Joint](#)

12.4.3 Add Move Blend Joint

功能说明

将更新后的混合数据添加到混合列表。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
ClearMoveBlendJoint();

RecipeBlendData blendData;
blendData.recipePoseNumber = 1;
blendData.recipeGainNumber = 3;
blendData.blendWaitTime = 0;

DG_RESULT result = UpdateBlendJoint(blendData);
if (result == DG_RESULT_NONE) {
    DG_RESULT result = AddMoveBlendJoint();
}
```

相关函数

- [12.4.1 Update Blend Joint](#)
- [12.4.2 Clear Move Blend Joint](#)
- [12.4.4 Set Move Blend Joint](#)
- [12.4.5 Start Move Blend Joint](#)
- [12.4.6 Stop Move Blend Joint](#)

12.4.4 Set Move Blend Joint

功能说明

将当前添加的混合列表保存为混合动作列表。

参数

数据类型	变量	说明
int	blendNumber	混合运动编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
ClearMoveBlendJoint();

RecipeBlendData blendData;
blendData.recipePoseNumber = 1;
blendData.recipeGainNumber = 3;
blendData.blendWaitTime = 0;
DG_RESULT result = UpdateBlendJoint(blendData);
if (result != DG_RESULT_NONE)    return result;
result = AddMoveBlendJoint();
if (result != DG_RESULT_NONE)    return result;

blendData.recipePoseNumber = 2;
blendData.recipeGainNumber = 2;
blendData.blendWaitTime = 1000;
DG_RESULT result = UpdateBlendJoint(blendData);
if (result != DG_RESULT_NONE)    return result;
result = AddMoveBlendJoint();
if (result != DG_RESULT_NONE)    return result;

if (result == DG_RESULT_NONE)
{
    int blendNumber = 1;
    result = SetMoveBlendJoint(blendNumber);
}
```

相关函数

12.4.1 [Update Blend Joint](#)

12.4.2 [Clear Move Blend Joint](#)

12.4.3 [Add Move Blend Joint](#)

12.4.5 [Start Move Blend Joint](#)

12.4.6 [Stop Move Blend Joint](#)

12.4.5 Start Move Blend Joint

功能说明

输入的混合编号为混合运动执行。

参数

数据类型	变量	说明
int	blendNumber	混合运动编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int blendNumber = 1;  
DG_RESULT result = StartMoveBlendJoint(blendNumber);
```

相关函数

- [12.4.1 Update Blend Joint](#)
- [12.4.2 Clear Move Blend Joint](#)
- [12.4.3 Add Move Blend Joint](#)
- [12.4.4 Set Move Blend Joint](#)
- [12.4.6 Stop Move Blend Joint](#)

12.4.6 Stop Move Blend Joint

功能说明

当前执行正在混合运动停止。

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
DG_RESULT result = StopMoveBlendJoint();
```

相关函数

[12.4.1 Update Blend Joint](#)

[12.4.2 Clear Move Blend Joint](#)

[12.4.3 Add Move Blend Joint](#)

[12.4.4 Set Move Blend Joint](#)

[12.4.5 Start Move Blend Joint](#)

第 13 章 动作函数

13.1 Manual Teach Mode

功能说明

设置夹爪的手动示教模式。手动示教模式是操作人员可直接设置夹爪位姿的模式，可直接示教所有关节。

参数

数据类型	变量	说明
int	isOn	手动操作模式设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int isOn = 1; // 手动操作模式设置  
ManualTeachMode(isOn);
```

13.2 Start Grasp Motion

功能说明

抓取运动执行。

参数

数据类型	变量	说明
int	isGrasp	抓取运动执行设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
#define IS_ON 1
DG_GRASP_MODE graspMode = DG_GRASP_MODE_3F_3FINGER; // 三手指中心点抓取模式
float graspForce = 3.0;
int graspOption = DG_GRASP_OPTION_NONE; // 抓取选项预使用
int smoothGrasping = IS_ON; // 平滑抓取使用
DG_RESULT result = SetGraspData(graspMode, graspForce, graspOption,
smoothGrasping);
if (result == DG_RESULT_NONE){
    int isGrasp = IS_ON;
    result = StartGraspMotion(isGrasp);
}
```

相关函数

[11.1.1 Set Grasp Data](#)

[11.7 位置控制模式设置](#)

13.3 位置控制

13.3.1 Move Joint

功能说明

单个关节的位置移动执行。

参数

数据类型	变量	说明
float	targetJoint	目标位置值(degree) 输入
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float targetJoint = 20.9; // 目标位置 20.9也设置
int jointNumber = 1;
DG_RESULT result = MoveJoint(targetJoint, jointNumber);
```

13.3.2 Move Joint Finger

功能说明

单个手指关节项的位置移动执行。

参数

数据类型	变量	说明
float*	targetJoint	目标位置值(degree) 输入
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++

float targetJoint[MAX_FINGER_JOINT_COUNT] = { 0 };
targetJoint[0] = 0;
targetJoint[1] = 0;
targetJoint[2] = 60; // 目标位置 60也设置
targetJoint[3] = 60; // 目标位置 60也设置
int fingerNumber = 1;
DG_RESULT result = MoveJointFinger(targetJoint, fingerNumber);
```

相关函数

11.2 P 增益设置
11.6 动作时间设置

13.3.3 Move Joint All

功能说明

所有关节项的位置移动执行。

参数

数据类型	变量	说明
float*	targetJoint	目标位置值(degree) 输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
float targetJoint[MAX_JOINT_COUNT] = { 0 };  
targetJoint[0] = targetJoint[4] = targetJoint[8] = 0;  
targetJoint[1] = targetJoint[5] = targetJoint[9] = 0;  
targetJoint[2] = targetJoint[6] = targetJoint[10] = 60; // 目标位置 60也设置  
targetJoint[3] = targetJoint[7] = targetJoint[11] = 60; // 目标位置 60也设置  
DG_RESULT result = MoveJointAll(targetJoint);
```

13.3.4 Move Servo Joint

功能说明

所有关节项的位置移动实时为执行。该函数动作时间设置在影响不受，立即移动。
该功能开发者模式在可使用。

参数

数据类型	变量	说明
float*	targetJoint	目标位置值(degree) 输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++
float targetJoint[MAX_JOINT_COUNT] = { 0 };
targetJoint[0] = targetJoint[4] = targetJoint[8] = 0;
targetJoint[1] = targetJoint[5] = targetJoint[9] = 0;
targetJoint[2] = targetJoint[6] = targetJoint[10] = 60; // 目标位置 60也设置
targetJoint[3] = targetJoint[7] = targetJoint[11] = 60; // 目标位置 60也设置
DG_RESULT result = MoveServoJoint(targetJoint);

13.3.5 Move Joint Base

功能说明

基座关节项的位置移动执行。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
float*	targetJoint	目标位置值(degree) 输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
float targetJoint[MAX_BASE_JOINT_COUNT] = { 0 };  
targetJoint[0] = 70;  
targetJoint[1] = -70;  
DG_RESULT result = MoveJointBase(targetJoint);
```

13.4 电流控制

各关节电流控制为驱动时使可以有方法说明。电流控制模式在 PID Gain 值设置不可用。

13.4.1 Set Current Control Mode

功能说明

设置夹爪关节的电流控制模式。

参数

数据类型	变量	说明
int	isOn	电流控制模式设置

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int isOn = 1; // 电流控制模式设置
DG_RESULT result = SetCurrentControlMode(isOn);
```

13.4.2 Set Target Current Joint

功能说明

单个关节的目标电流值设置。

参数

数据类型	变量	说明
int	targetCurrent	目标电流值(mA) 输入
int	jointNumber	关节编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int targetCurrent = 50; // 目标电流值 50mA  
int jointNumber = 1; // 第 1 个关节  
DG_RESULT result = SetTargetCurrentJoint(targetCurrent, jointNumber);
```

13.4.3 Set Target Current Finger

功能说明

单个手指关节项的目标电流值设置。

参数

数据类型	变量	说明
int*	targetCurrent	目标电流值(mA) 输入
int	fingerNumber	手指编号输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++  
  
int targetCurrent[4] = { 50,50,50,50 }; // 目标电流值 50mA  
int fingerNumber = 1; // 第 1 个手指  
DG_RESULT result = SetTargetCurrentFinger(targetCurrent, fingerNumber);
```

13.4.4 Set Target Current All

功能说明

全部关节项的目标电流值设置。

参数

数据类型	变量	说明
int*	targetCurrent	目标电流值(mA) 输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

```
C/C++
int targetCurrent[MAX_JOINT_COUNT];
for (int i = 0; i < MAX_JOINT_COUNT; i++)
{
    targetCurrent[i] = 50;
}
DG_RESULT result = SetTargetCurrentAll(targetCurrent);
```

13.4.5 Set Target Current Base

功能说明

基座关节项的目标电流值设置。该函数 DG-4F-M 专用函数。

参数

数据类型	变量	说明
int*	targetCurrent	目标电流值(mA) 输入

返回值

编号	值	说明
0	DG_RESULT_NONE	正常
0 不是时	ERROR	附录 2 错误列表参考

示例

C/C++

```
int targetCurrent[MAX_BASE_JOINT_COUNT];  
targetCurrent[0] = 50; // 第 17 个关节, 第 1 个手指对应的基座关节目标电流值设置  
targetCurrent[1] = 50; // 第 18 个关节, 第 4 个手指对应的基座关节目标电流值设置  
DG_RESULT result = SetTargetCurrentBase(targetCurrent);
```

附录 1 术语表及功能说明

B

Baudrate (波特率)

数据传输的单位 baud 的数据传输速度。

Blend list (混合列表)

配方位姿和配方增益包括的连续位置动作用于列表集合。

Blend motion (混合运动)

混合列表已保存的运动。

Blend move (混合动作)

由配方位姿和配方增益组成的混合列表将按照注册顺序连续执行动作。

C

Cartesian coordinate (笛卡尔坐标系)

3D 空间上在点的位置数学地为表示的坐标系为, x 轴, Y 轴, Z 轴为构成的空间坐标系及直角坐标系。

Communication Period (通信周期)

设备之间在数据收发时间。

COMPORT(COM 端口)

用于串行通信的计算机输入/输出端口。

D

Developer Mode (开发者模式)

用户直接开发的控制器使用的模式。夹爪的各关节的信息接收以各关节直接控制可以有模式。

E

Encoder (编码器)

旋转运动或直线运动执行物体的位置和速度信息前电气信号为输出执行传感器。

F

Fix tilt option (倾斜固定抓取选项)

夹爪当前位姿基准，该选项为抓取动作执行时手指的第 4 个个关节的倾斜保持的状态为抓取动作执行功能。

I

In-hand manipulation (手内操作)

机器人手部抓取物体的位置，方向操作执行控制功能。

Inpose (的位姿)

该范围用于在位置控制移动至目标关节位置值时，确认是否已到达目标位置。主要用于在到达目标位置前执行其他运动。

J

Joint (关节)

1 自由度旋转轴指驱动部。Delto Gripper 系列时 DG-3F-M, DG-3F-B 12 个关节, DG-4F-M 18 个关节, DG-5F-R, DG-5F-L 20 个关节为构成。

L

Low Pass Filter (低通滤波器)

特定频率执行通过和时使滤波器为，接收的数字信号的噪声减少，急剧的变化缓和并平滑结果和输出。

M

Modbus protocol (模式总线协议)

自动化设备之间通信为了开发的工业标准通信协议为主站-从站结构使用并且, RTU, ASCII, TCP/IP 等的多种通信方式支持. Slave ID 通过连接的设备项识别可用.

Modbus RTU (模式总线 RTU)

RS-232, RS-485 等串行通信器为执行协议, 二进制格式的传输方式为高效且快速通信可用.

Modbus TCP TCP (模式总线 TCP)

在 TCP/IP 网络环境中运行的 Modbus Protocol。数据以数据包形式传输, 可通过网络实现快速且灵活的通信。

Module (模块)

独立的部分为分离的最小驱动单位. Delto Gripper 系列各关节一个独立的模块为构成可以, 独立的驱动可用并且, 扩展性及维护容易. 各模块电机, 编码器, 电机驱动器, 减速器等为构成.

O

Operator Mode(用户模式)

使用产品内部控制器执行控制的模式。该模式支持 Modbus Protocol, 可通过修改寄存器映射表或使用提供的 UI 便捷地控制夹爪。

Overshoot (超调)

控制系统在目标值为收敛和过程在暂时为目标值秒和执行现象

P

Parallel option (平行抓取选项)

倾斜固定抓取选项和不同，夹爪的当前位姿和无关无需始终指尖的平面之间相互贴合抓取执行选项为抓取时物体和的接触面最大化可以可以。

PD control (PD 控制)

比例(Proportional)，微分(Derivative) 控制组合的反馈控制系统。
误差在比例的响应和误差的变化率在依据响应结合后速度快速且稳定的控制可用一个完全的误差补偿困难。

PID control (PID 控制)

由比例（Proportional）、积分（Integral）、微分（Derivative）控制组合而成的反馈控制系统。该系统用于在自动化及机器人系统中实现所需目标值，并可实现稳定且精确的控制。

Position control (位置控制)

电机或机械部件的位置调节的控制方式。当前位置基于为目标位置所需移动时使情况位置控制称为，并，PID 控制通过的精确的控制可用。

R

Recipe (配方)

机器人(夹爪) 简便且，高效地控制进行为了主要设置值项预先保存保存的数据的集合。提前配方保存保存后已保存的配方通过控制的便利性提升。

S

Smooth grasping (平滑抓取模式)

物体抓取可以时物体避免损坏较轻地抓取执行模式， 物体在接触后设置的抓取力度为更改。

Socket communication (套接字通信)

网络通过两个设备数据收发执行方式. IP 地址, PORT 编号通过收发可以设备识别.

Software Development Kit (SDK)

用于软件应用程序开发的工具及库集合。用户可通过 SDK 方便地实现夹爪控制功能。

T

TCP/IP

互联网及网络通信用于核心协议为，数据可靠性地传递可以并且. LAN 端口通过有线为网络要连接的可以可以，多个设备高效地连接可用.

W

WSA (Window Socket API)

Windows 操作系统在套接字编程支持执行 API 为客户端和服务端之间的数据通信可用将执行 API.

.

错误列表

值	名称	说明
0	DG_RESULT_NONE	正常
1	DG_RESULT_SYSTEM_SETTING_ NOT_PERFORMED	系统设置完成未不. 系统设置函数执行后执行请
100	DG_RESULT_VALUE_IS_NEGATIVE	输入数据负。
101	DG_RESULT_OVERFLOW_ JOINT_COUNT	输入可用的最大电机编号秒和完成. MAX_JOINT_COUNT = 20
102	DG_RESULT_OVERFLOW_ FINGER_COUNT	输入可用的最大手指编号 秒和完成. MAX_FINGER_COUNT = 5
103	DG_RESULT_OVERFLOW_TCP_COUNT	输入可用的最大坐标系数量 秒和完成. CARTESIAN_COORDINATE_POSE_COUNT = 6
105	DG_RESULT_OVERFLOW_ADD_ BLEND_SIZE	位姿列表最大值。 单个混合动作中最多可添加的位姿数量。 MAX_BLEND_ADD_POSE_COUNT = 50
106	DG_RESULT_OVERFLOW_RECIPE_ BLEND_SIZE	输入可用的最大混合运动编号 秒和完成. MAX_BLEND_COUNT = 10
107	DG_RESULT_BLEND_SIZE_ZERO	已保存的混合运动。
108	DG_RESULT_NOT_SUPPORTED_MODEL	该功能支持执行型号不是.
109	DG_RESULT_DATA_IS_ZERO	输入数据 0。
110	DG_RESULT_DATA_IS_NOT_BOOLEAN	0 和 1 仅可输入.
111	DG_RESULT_NOT_FOUND_MODEL	型号编号查找。
112	DG_RESULT_OVERFLOW_RECIPE_ POSE_COUNT	输入可用的最大配方位姿编号 秒和完成.

		MAX_RECIPE_POSE_COUNT = 100
113	DG_RESULT_OVERFLOW_RECIPE_GAIN_COUNT	输入可用的最大配方增益编号秒和完成. MAX_RECIPE_GAIN_COUNT = 20
114	DG_RESULT_OVERFLOW_RECIPE_GRASP_COUNT	输入可用的最大配方抓取编号秒和完成. MAX_RECIPE_GRASP_COUNT = 20
115	DG_RESULT_INVALID_CONTROL_MODE	无效不系统控制模式。
116	DG_RESULT_INVALID_GRASP_MODE_DATA	输入可用的最大抓取模式编号秒和完成. 9.2.6 enum GAIN_MODE
118	DG_RESULT_OVERFLOW_MAX_BYTE_DATA	输入可用的 byte 最大值(255)秒和完成.
119	DG_RESULT_OVERFLOW_CURRENT_LIMIT	输入可用的最大电流值秒和完成. M 型号: MAX_INPUT_CURRENT = 150 S 型号: MAX_INPUT_CURRENT = 360
120	DG_RESULT_IS_NOT_TORQUE_CONTROL_MODE	扭矩控制模式不是.
121	DG_RESULT_NOT_SUPPORTED_DATA_TYPE	支持不执行不数据接收命令并。
122	DG_RESULT_BACKUP_START	数据备份中。
123	DG_RESULT_INVALID_PASSWORD	错误的密码编号输入。
124	DG_RESULT_OVERFLOW_GPIO_COUNT	输入可用的 GPIO 最大编号秒和完成. MAX_GRIPPER_GPIO_SIZE = 6
125	DG_RESULT_OVERFLOW_GRASP_FORCE	抓取力度输入值秒和完成. MAX_GRASP_FORCE = 20
126	DG_RESULT_OVERFLOW_BLEND_WAIT_TIME	混合等待时间输入值秒和完成. MAX_BLEND_WAIT_TIME = 30000

127	DG_RESULT_NOT_FOUND_ RESTORE_DATA	路径上恢复可以文件。
128	DG_RESULT_RESTORE_START	执行恢复中。
200	DG_RESULT_NOT_ARRIVED	目标位置在到达不执行失败。
201	DG_RESULT_NOT_START_ BLEND_MOVE	混合动作模式不是。
202	DG_RESULT_ALREADY_ BLEND_MOVE_STATE	混合动作中。
203	DG_RESULT_ALREADY_TCP_MOVE	TCP 位置移动中。
204	DG_RESULT_RECIPE_IS_NOT_ JOINT_MODE	位姿配方关节位置控制模式 不是。
205	DG_RESULT_ACTIVATE_CURRENT_ CONTROL_MODE	电流控制模式启用的状态。
206	DG_RESULT_ACTIVATE_ GRASP_MOTION	抓取执行并有状态。
207	DG_RESULT_ACTIVATE_MANUAL_ CONTROL_MODE	手动控制模式启用的状态。
401	DG_RESULT_JOINT_FAULT_1	第 1 个关节在上发现
402	DG_RESULT_JOINT_FAULT_2	第 2 个关节在上发现
403	DG_RESULT_JOINT_FAULT_3	第 3 个关节在上发现
404	DG_RESULT_JOINT_FAULT_4	第 4 个关节在上发现
405	DG_RESULT_JOINT_FAULT_5	第 5 个关节在上发现
406	DG_RESULT_JOINT_FAULT_6	第 6 个关节在上发现
407	DG_RESULT_JOINT_FAULT_7	第 7 个关节在上发现
408	DG_RESULT_JOINT_FAULT_8	第 8 个关节在上发现
409	DG_RESULT_JOINT_FAULT_9	第 9 个关节在上发现
410	DG_RESULT_JOINT_FAULT_10	第 10 个关节在上发现
411	DG_RESULT_JOINT_FAULT_11	第 11 个关节在上发现

412	DG_RESULT_JOINT_FAULT_12	第 12 个关节在上发现
413	DG_RESULT_JOINT_FAULT_13	第 13 个关节在上发现
414	DG_RESULT_JOINT_FAULT_14	第 14 个关节在上发现
415	DG_RESULT_JOINT_FAULT_15	第 15 个关节在上发现
416	DG_RESULT_JOINT_FAULT_16	第 16 个关节在上发现
417	DG_RESULT_JOINT_FAULT_17	第 17 个关节在上发现
418	DG_RESULT_JOINT_FAULT_18	第 18 个关节在上发现
419	DG_RESULT_JOINT_FAULT_19	第 19 个关节在上发现
420	DG_RESULT_JOINT_FAULT_20	第 20 个关节在上发现
421	DG_RESULT_JOINT_FAULT_21	第一个指尖模块在上发现
422	DG_RESULT_JOINT_FAULT_22	两个指尖模块在上发现
423	DG_RESULT_JOINT_FAULT_23	三个指尖模块在上发现
424	DG_RESULT_JOINT_FAULT_24	四个指尖模块在上发现
425	DG_RESULT_JOINT_FAULT_25	五个指尖模块在上发现
500	DG_RESULT SOCK_EXCEPTION	套接字通信在异常发生完成.
501	DG_RESULT SOCK_FAILED_ WSA_START_UP	套接字通信连接失败.
1002	DG_RESULT_NO_GRASP_OBJECT	物体抓取不.
1003	DG_RESULT_ONLY_SUPPORTED_ 3FINGER	DG-3F-B, DG-3F-M 型号专用
1004	DG_RESULT_NOT_SUPPORTED_ CONTROL_MODE_OPERATOR	用户模式支持不执行不功能。
1005	DG_RESULT_NOT_SUPPORTED_ CONTROL_MODE_DEVELOPER	开发者模式支持不执行不功能。
1006	DG_RESULT_NOT_SUPPORTED_PORT_NUM	支持不执行不端口编号。
2000	DG_RESULT_PORT_EXCEPTION	串口通信在异常发生完成.
2001	DG_RESULT_PORT_FAILED_START_UP	串口通信连接失败.
2002	DG_RESULT_PORT_SET_CONFIG_	串口设置失败.

	ERROR	
2003	DG_RESULT_PORT_SET_TIMEOUT_ERROR	串口超时设置失败.
2004	DG_RESULT_PORT_SET_COMMASK_ERROR	串口通信掩码设置失败.
2009	DG_RESULT_DIAGNOSING_SYSTEM	诊断功能执行中。

