

Delto Gripper

Manager 및 SDK 사용설명서

한국어

Ver. 2.0.0



INFORMATION

저작권

이 사용 설명서는 (주)테솔로에 의해 작성되었으며, 모든 저작권은 (주)테솔로에 귀속됨으로 해당 설명서를 기반으로 재생산하여 제 3 자에게 배포하는 등의 행위는 금지되어 있음

사용 설명서 개정 관련 고지

기술적 향상을 위해 해당 설명서는 개정될 수 있음을 미리 고지함

Manual Version: 2.0.0 국문 (2026.04)

© 2026 주식회사 테솔로 All rights reserved

주식회사 테솔로

본사: 인천광역시 연수구 컨벤시아대로 165, 26 층

지사: 경기도 광명시 일직로 43 GIDC 타워, A 동 26 층

Support T: +82 2 6914 6620

E: support@tesollo.com

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

목차

제 1 장 설명서 정보	11
제 2 장 유의 사항.....	12
제 3 장 시작하기	13
3.1 DG Manager 설치.....	13
3.1.1 설치 파일 위치.....	13
3.1.2 DG Manager 설치.....	14
3.2 DG Manager 삭제.....	18
제 4 장 통신 연결 화면	20
4.1 로그인 화면.....	20
4.1.1 화면 구성.....	20
4.1.2 사용자 모드 연결.....	22
4.1.3 개발자 모드 연결.....	26
4.1.4 연결 확인.....	28
4.2 메인 화면	30
4.2.1 화면 구성.....	30
제 5 장 모니터링 화면.....	32
5.1 화면 구성	32
제 6 장 동작 화면.....	34
6.1 그래프 모드	34
6.1.1 화면 구성.....	34
6.1.2 그래프 조작	35
6.1.3 그래프 데이터 설정	35
6.1.4 위치 제어 모드 설정.....	37
6.2 위치 제어.....	38

6.2.1 화면 구성.....	38
6.2.2 위치 제어 동작	39
6.2.3 티칭 모드.....	40
6.3 계인 설정.....	41
6.3.1 화면 구성.....	41
6.3.2 계인 모드 설정	42
6.3.3 계인 설정.....	43
6.4 전류 설정	44
6.4.1 화면 구성.....	44
6.4.2 전류 모드 설정	44
6.4.3 전류 설정.....	45
제 7 장 레시피 설정 화면.....	46
7.1 화면 구성.....	46
7.2 포즈 레시피.....	47
7.2.1 화면 구성	47
7.2.2 포즈 레시피 설정 방법	48
7.2.3 포즈 레시피 동작 사용 방법.....	49
7.3 계인 레시피.....	50
7.3.1 화면 구성	50
7.3.2 계인 레시피 설정 방법	51
7.3.3 계인 레시피 동작 사용 방법.....	52
7.4 그랩 레시피.....	53
7.4.1 화면 구성	53
7.4.2 그랩 레시피 설정 방법	54
7.4.3 그랩 레시피 동작 사용 방법	55
7.5 블렌드 레시피.....	56
7.5.1 화면 구성	56

7.5.2 블렌드 레시피 설정 방법	57
7.5.3 블렌드 레시피 동작 사용 방법.....	59
제 8 장 부가 기능 화면	60
8.1 설정	60
8.1.1 화면 구성.....	60
8.1.2 Boot Recipe	61
8.1.3 Joint Offset.....	62
8.1.4 Joint Inpose.....	63
8.1.5 Moving Inpose	63
8.1.6 Low Pass Filter	64
8.1.7 Torque Limit Mode.....	64
8.1.8 Backup Settings / EEPROM Write.....	65
8.1.9 Restore Settings.....	66
8.2 도구	67
8.2.1 화면 구성.....	67
8.2.2 Fingertip Sensor.....	68
8.2.3 GPIO	71
8.2.4 Firmware Download	72
8.2.5 Set IP.....	74
8.3 스파이	75
8.4 테마	76
제 9 장 SDK 자료 구조	77
9.1 Define	77
9.2 열거형	78
9.2.1 enum DG_RESULT	78
9.2.2 enum DG_MODEL	78
9.2.3 enum BLEND_MOTION_STATUS	78

9.2.4 enum COMMUNICATION_MODE.....	79
9.2.5 enum CONTROL_MODE.....	79
9.2.6 enum GAIN_MODE.....	79
9.2.7 enum DEVELOPER_MODE_COMMAND	79
9.2.8 enum DEVELOPER_MODE_RECEIVED_DATA_TYPE.....	80
9.2.9 enum DG_GRASP_MODE	80
9.2.10 enum DG_GRASP_OPTION.....	82
9.2.11 enum DG_SENSOR_TYPE	82
9.3 구조체	83
9.3.1 struct ReceivedGripperData	83
9.3.2 struct RecipeBlendData	83
9.3.3 struct GripperSystemSetting.....	84
9.3.4 struct GripperSetting.....	84
9.3.5 struct RecipePoseData	85
9.3.6 struct RecipeGainData	85
9.3.7 struct RecipeGraspData.....	86
9.3.8 struct ReceivedFingertipSensorData.....	86
9.3.9 struct ReceivedGPIOData	87
제 10 장 시스템 함수	88
10.1 시스템 설정 함수	88
10.1.1 SetGripperSystem.....	88
10.1.2 SetGripperOption	89
10.1.3 ConnectToGripper	90
10.1.4 DisconnectToGripper	90
10.1.5 SystemStart.....	91
10.1.6 SystemStop	91
10.1.7 SetIp	92

10.1.8 SetJointEncoderZero	92
10.2 콜백.....	93
10.2.1 CallbackForOnConnected.....	93
10.2.2 CallbackForOnDisconnected	93
10.2.3 CallbackForOnReceivedGripperData.....	94
10.2.4 CallbackForOnCommunicationPeriod	94
10.2.5 CallbackForOnReceivedFingertipSensorData	95
10.2.6 CallbackForOnReceivedGPIOData.....	96
10.3 기타 시스템 함수	97
10.3.1 SetLowPassFilterAlpha	97
10.3.2 SetBootMode	97
10.3.3 SetGPIOOutput	98
10.3.4 SetGPIOOutputAll.....	99
10.3.5 SetTorqueLimitMode.....	99
10.3.6 SetBootRecipe	100
10.3.7 EEPROMWrite.....	101
10.3.8 BackupRecipeData	101
10.3.9 RestoreRecipeData.....	102
10.4 데이터 조회.....	103
10.4.1 GetReceivedGripperData.....	103
10.4.2 GetCommunicationPeriod.....	103
10.4.3 GetReceivedFingertipSensorData.....	104
10.4.4 GetReceivedGPIOData	104
10.4.5 GetLibraryVersion	105
제 11 장 동작 설정 함수	106
11.1 그랩 설정 함수.....	106
11.1.1 SetGraspData.....	106

11.1.2 SetGraspForce.....	107
11.2 게인 P 설정.....	108
11.2.1 SetJointGainP.....	108
11.2.2 SetJointGainPFinger	109
11.2.3 SetJointGainPAI.....	110
11.2.4 SetJointGainPBase	110
11.3 게인 D 설정.....	112
11.3.1 SetJointGainD	112
11.3.2 SetJointGainDFinger	113
11.3.3 SetJointGainDAI	113
11.3.4 SetJointGainDBase	114
11.4 게인 I 설정.....	115
11.4.1 SetJointGainI.....	115
11.4.2 SetJointGainIFinger	116
11.4.3 SetJointGainIAI.....	117
11.4.4 SetJointGainIBase.....	117
11.5 게인 PID 설정	119
11.5.1 SetControlPIDMode	119
11.5.2 SetJointGainPID	120
11.5.3 SetJointGainPIDFinger.....	121
11.5.4 SetJointGainPIDAI.....	122
11.5.5 SetJointGainPIDAIIEqual.....	123
11.5.6 SetJointGainPIDBase	124
11.6 동작 시간 설정.....	125
11.6.1 SetMotionTimeJoint.....	125
11.6.2 SetMotionTimeFinger.....	126
11.6.3 SetMotionTimeAll.....	127

11.6.4 SetMotionTimeAllEqual.....	128
11.6.5 SetMotionTimeBase.....	129
11.7 위치 제어 모드 설정.....	130
11.7.1 SetPositionModeJoint.....	130
11.7.2 SetPositionModeFinger.....	131
11.7.3 SetPositionModeAll.....	132
11.7.4 SetPositionModeBase.....	133
11.8 핑거팁 센서 설정.....	134
11.8.1 SetFingerTipDataZero.....	134
11.9 전류 I 게인 설정.....	135
11.9.1 SetJointCurrentGainI.....	135
11.9.2 SetJointCurrentGainIFinger.....	136
11.9.3 SetJointCurrentGainIAll.....	137
11.9.4 SetJointCurrentGainIBase.....	138
11.10 전류 P 게인 설정.....	139
11.10.1 SetJointCurrentGainP.....	139
11.10.2 SetJointCurrentGainPFinger.....	140
11.10.3 SetJointCurrentGainPAll.....	141
11.10.4 SetJointCurrentGainPBase.....	142
제 12 장 레시피 함수.....	143
12.1 레시피 포즈 함수.....	143
12.1.1 UpdateRecipeCurrentPoseData.....	143
12.1.2 UpdateRecipeTargetPoseData.....	144
12.1.3 LoadRecipePoseData.....	145
12.2 레시피 게인 함수.....	146
12.2.1 UpdateRecipeGainData.....	146
12.2.2 LoadRecipeGainData.....	147

12.3 레시피 그랩 함수	148
12.3.1 UpdateRecipeGraspData.....	148
12.3.2 LoadRecipeGraspData.....	149
12.4 레시피 블렌드 함수	150
12.4.1 UpdateBlendJoint.....	150
12.4.2 ClearMoveBlendJoint.....	151
12.4.3 AddMoveBlendJoint	152
12.4.4 SetMoveBlendJoint	153
12.4.5 StartMoveBlendJoint	154
12.4.6 StopMoveBlendJoint.....	155
제 13 장 동작 함수	156
13.1 ManualTeachMode.....	156
13.2 StartGraspMotion	157
13.3 위치 제어.....	158
13.3.1 MoveJoint	158
13.3.2 MoveJointFinger	159
13.3.3 MoveJointAll	160
13.3.4 MoveServoJoint.....	161
13.3.5 MoveJointBase	162
13.4 전류 제어.....	163
13.4.1 SetCurrentControlMode	163
13.4.2 SetTargetCurrentJoint	164
13.4.3 SetTargetCurrentFinger	165
13.4.4 SetTargetCurrentAll	166
13.4.5 SetTargetCurrentBase	167
부록 1 용어집 및 기능설명.....	168
부록 2 에러 리스트	172

제 1 장 설명서 정보

- '개발자 모드, J1'과 같이 자체적으로 사용되는 단어, 축약어 및 기타 용어는 '부록 1 용어집 및 기능설명'에 수록되어 있으므로, 혼동되실 경우 참고하시기 바랍니다.

- DG Manager 및 SDK 는 Windows 10 x64, Windows 11 x64, Ubuntu 24.04 x64, Ubuntu 22.04 x64, C++20 환경에서 테스트되었으며, 유사한 계열의 OS 환경에서도 동작이 가능합니다. 아래는 테스트한 PC 사양입니다.

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

- * RAM: 16GB

- * Graphic: NVIDIA GeForce RTX 3060

- 사용자 모드 통신 프로토콜은 'Modbus RTU', 'Modbus TCP' 입니다. 개발자 모드는 TCP/IP 기반 자체 프로토콜을 사용합니다.

제 2 장 유의 사항

- PC 성능 및 네트워크 환경에 따라 프로그램이 원활하게 작동하지 않을 수 있습니다.

- DG Manager는 테솔로에서 판매하고 있는 Delto Gripper 시리즈를 조작하기 위해 사용하실 수 있습니다. 다만 펌웨어와 그리퍼 모델에 따라 지원하는 기능이 상이하거나 호환되지 않을 수 있습니다. 때문에 DG Manager 및 SDK를 사용할 때 펌웨어와 모델에 유의하여 사용하시길 바랍니다.

- 본 매뉴얼에서 안내한 대로 실행하지 않고, 잘못된 조작으로 인하여 오작동이 발생할 수 있습니다. 이는 그리퍼에 손상이 발생할 수 있으며, 경우에 따라 A/S를 받지 못할 수 있습니다.

- 기타 지원이 필요한 경우 테솔로 서포트 메일로 문의 바랍니다.
(support@tesollo.com)

제 3 장 시작하기

3.1 DG Manager 설치

3.1.1 설치 파일 위치

3.1.1.1 Windows

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

그리퍼 포장 박스에 같이 동봉된 USB안에 그림과 같은 경로인 'Software – DGManager-Windows' 폴더로 들어가면 'DGManagerInstaller_ver_x_x_x.exe(ver_x_x_x는 버전 정보)'라는 설치 파일이 있습니다. 해당 설치 파일을 실행하시면 DG Manager를 설치할 수 있습니다.

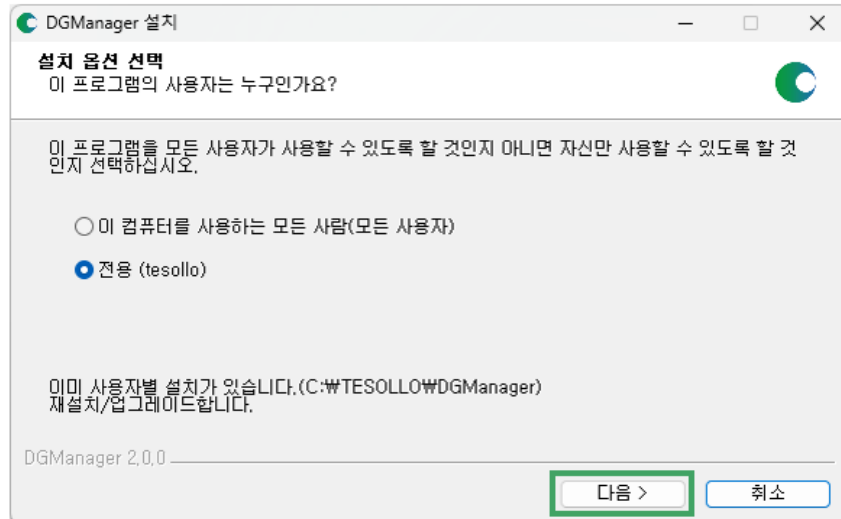
3.1.1.2 Ubuntu 24.04 / 22.04



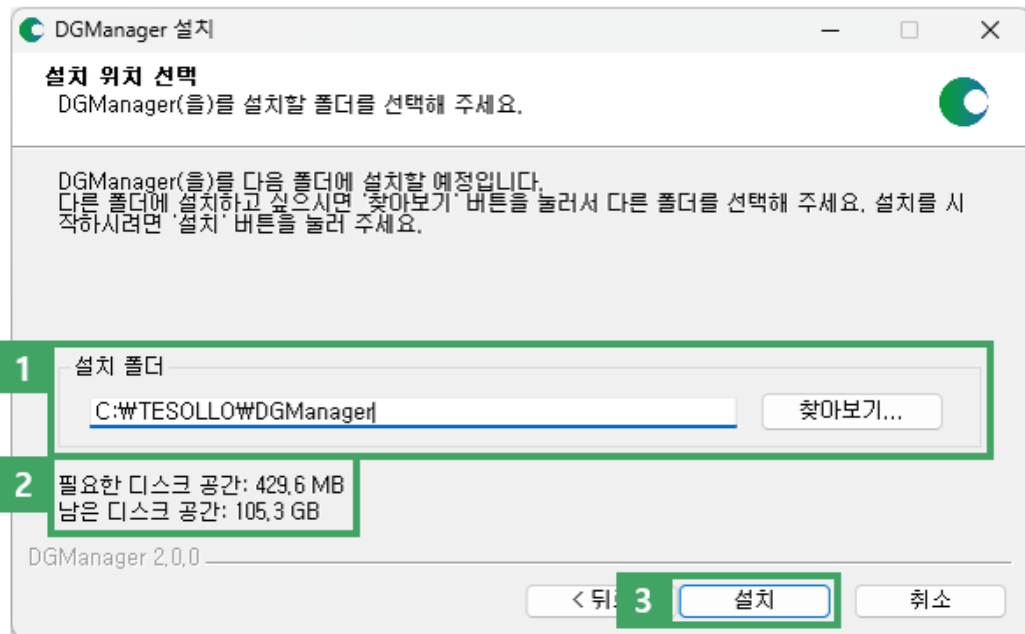
그리퍼 포장 박스에 같이 동봉된 USB안에 그림과 같은 경로인 'Software – DGManager-Linux' 폴더로 들어가면 'DGManager_linux_ver_x_x_x.zip(ver_x_x_x는 버전 정보)'라는 압축 파일이 있습니다. 압축을 푸시면 DGManager를 실행할 수 있습니다.

3.1.2 DG Manager 설치

3.1.2.1 Windows



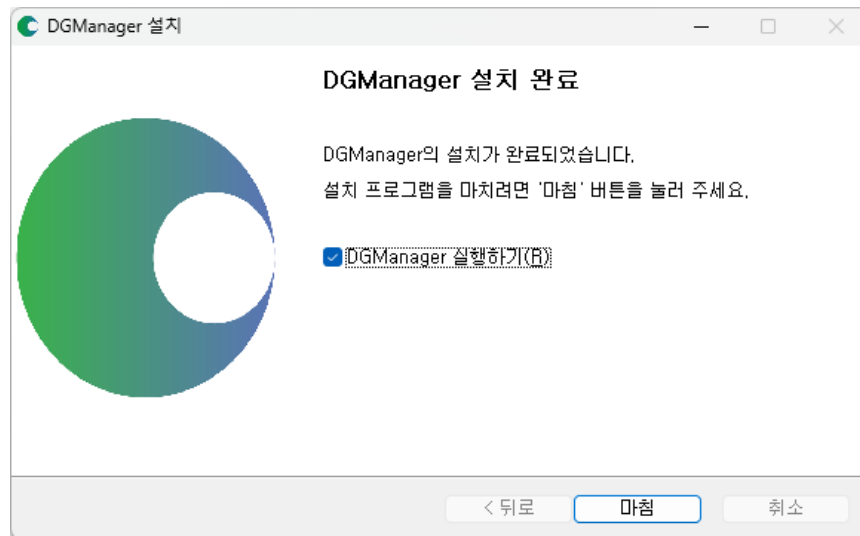
프로그램을 실행시키면 위 그림과 같이 설치파일 화면이 출력 됩니다. 'Next>'를 눌러 다음 화면으로 이동합니다.



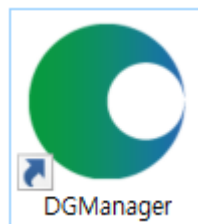
파일 경로를 설정한 후 설치를 시작합니다. 설치 시작전에 저장소에 남은 공간이 충분한지 확인해주세요.

항목

번호	명칭	설명
1	Destination Folder Browse	파일을 설치할 폴더 경로 설정
2	용량 확인	파일 용량과 설치공간 용량 표시
3	Install	파일 설치 시작



정상적으로 파일 설치가 완료되면 위와 같은 화면이 출력되며, 바탕화면에 DGManager.exe 바로가기가 생성됩니다.



3.1.2.2 Ubuntu 24.04 / 22.04

예제

```
bash
# 압축 풀기
unzip DGManager_linux_ver_1_2_0.zip
# 디렉토리 진입
cd DGManager-linux-x64/
# 의존성 설치 및 실행 권한 부여 후 실행
. install_run.sh
# 실행 권한만 부여하고 실행 (라이브러리 설치 없이)
. run.sh
# 프로그램 실행
./DGManager
```

압축을 푸시면 DGManager를 실행하실 수 있습니다.

의존성 패키지 설치 및 실행 권한 부여가 필요한 경우, install_run.sh 스크립트를 먼저 실행하시기 바랍니다.

(이 과정에서는 인터넷 연결이 필요합니다.)

라이브러리 설치 없이 권한만 부여하고 실행하려는 경우에는 run.sh 스크립트를 실행하시면 됩니다.

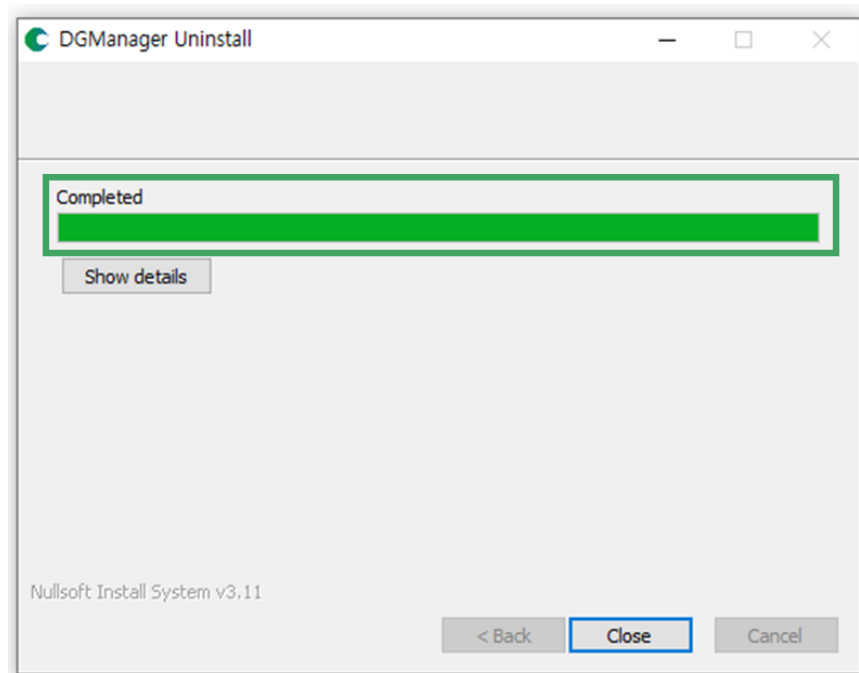
단순히 실행만 하려면 ./DGManager 파일을 직접 실행하시면 됩니다.

3.2 DG Manager 삭제

3.2.1.1 Windows

Tesollo > DGManager			
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

파일 설치 경로로 진입하시면 'UninstallDGManager.exe'라는 파일이 있습니다. 해당 파일을 실행하면 DG Manager 프로그램 제거를 시작합니다.



파일 제거가 완료되면 위 그림과 같은 화면이 출력됩니다.

3.2.1.2 Ubuntu 24.04 / 22.04

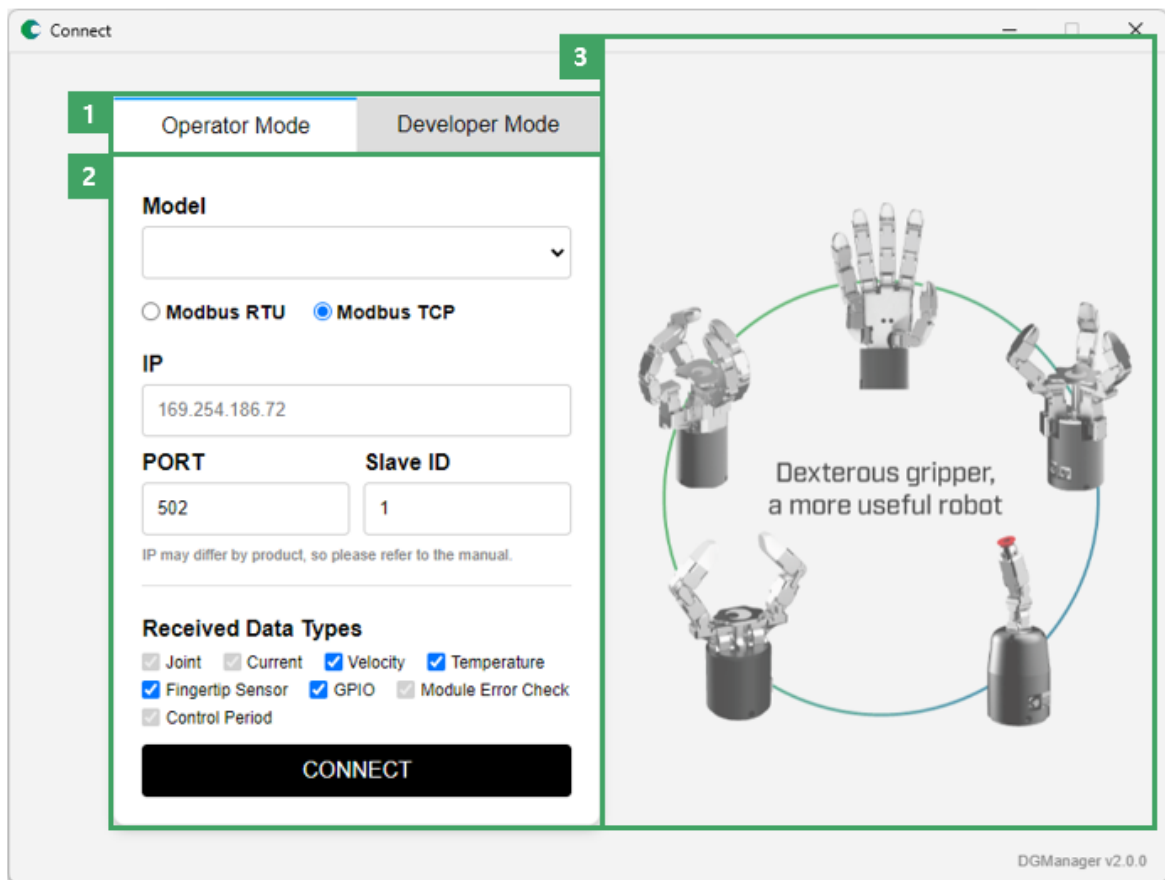
해당 파일은 ZIP 형식으로 공유되므로, 리눅스에서는 압축을 푼 후 단순히 디렉토리를 삭제하시면 모든 파일이 제거됩니다.

제 4 장 통신 연결 화면

4.1 로그인 화면

4.1.1 화면 구성

기능 설명



Delto Gripper는 **사용자 모드**와 **개발자 모드**를 지원합니다.

사용자 모드(Operator Mode)

제품 내부의 제어를 사용하여 제어하는 모드입니다. Modbus Protocol을 지원하며, 단순히 레지스터맵을 변경하거나, 제공되는 UI 조작으로 그리퍼를 쉽게 제어할 수 있습니다.

사용자 모드 및 SDK를 사용하실 경우 'Torque limit mode'를 'off'로 설정해야 'stall torque' 사용이 가능합니다.

개발자 모드(Developer Mode)

개발자가 직접 개발한 제어를 이용하는 모드입니다. 그리퍼의 각 관절의 정보를 수신하여 각 관절을 직접 제어할 수 있는 모드입니다.

주의!! 사용자 모드에서 'Torque limit mode' off 사용 또는 개발자 모드에서 stall torque로 사용 시, 급격한 제품 온도 상승 및 제품 손상 가능성이 있으며, 고장 시 A/S가 어려운 점 주의바랍니다.

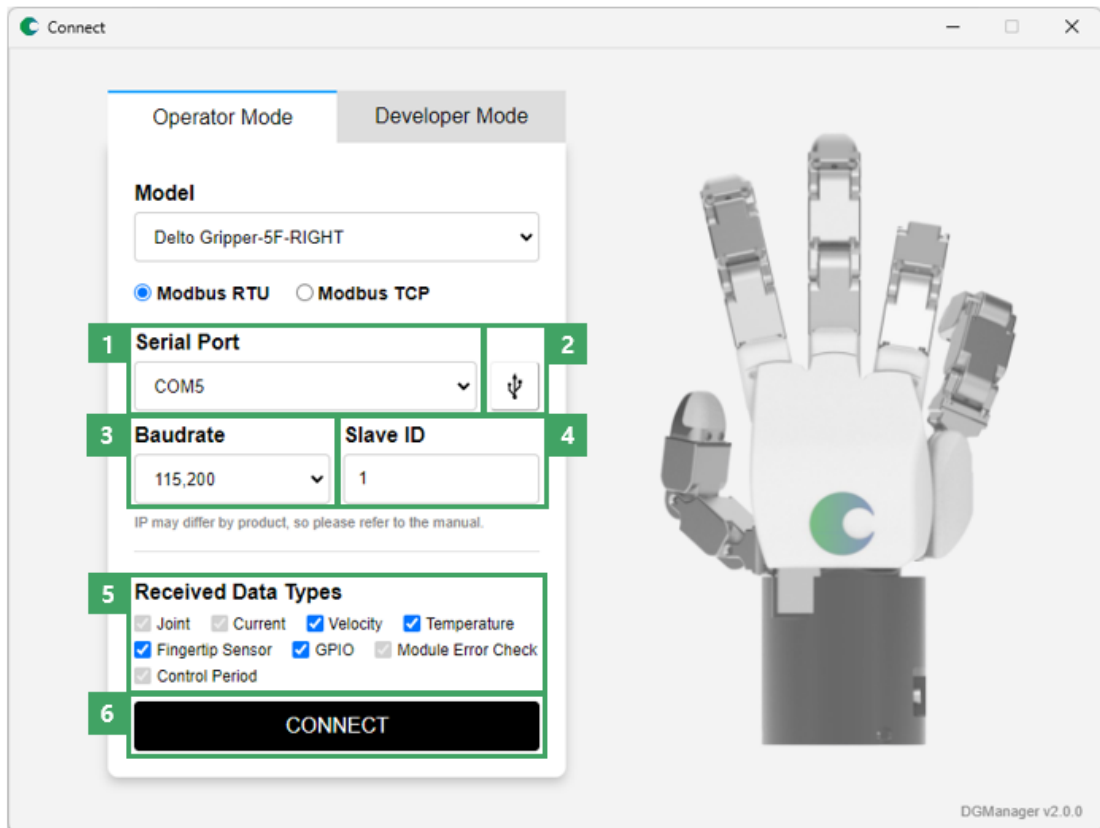
항목

번호	명칭	설명
1	제어 모드 선택	제어 모드를 4.1.2 사용자 모드와 4.1.3 개발자 모드 중 선택
2	연결 설정값 입력	제어 및 통신 방식에 따라 필요한 설정값 입력
3	그리퍼 모델 이미지	선택 그리퍼 모델 이미지 표시

4.1.2 사용자 모드 연결

4.1.2.1 Modbus RTU 연결 설정

기능 설명



Modbus RTU 연결 예시

연결할 그리퍼의 모델이 DG-3F-M, Modbus RTU, COM8 포트 연결, Baudrate 115200, Slave ID 1인 경우, 먼저 'Model'에서 'Delto Gripper-3F-M'를 선택합니다. 이후 'Modbus RTU'를 선택한 뒤 스캔 버튼을 클릭하여 'Serial Port'에서 'COM8'을 선택, 'Baudrate'에서 '115,200'을 선택, 'Slave ID'에 1을 입력합니다. 마지막으로 CONNECT 버튼을 클릭합니다. 정상적으로 통신이 연결이 되면 메인 화면으로 넘어갑니다.

항목

번호	명칭	설명
1	Serial Port	그리퍼 연결된 시리얼 포트 선택
2	Scan	현재 연결된 시리얼 포트 스캔
3	Baudrate	그리퍼 설정된 Baudrate 선택 □ 115200
4	Slave ID	Modbus Slave ID 입력 □ Slave ID 범위: 1 – 255
5	Received Data Types	수신받을 데이터 타입을 선택
6	CONNECT	설정한 값으로 그리퍼 통신 연결

참고

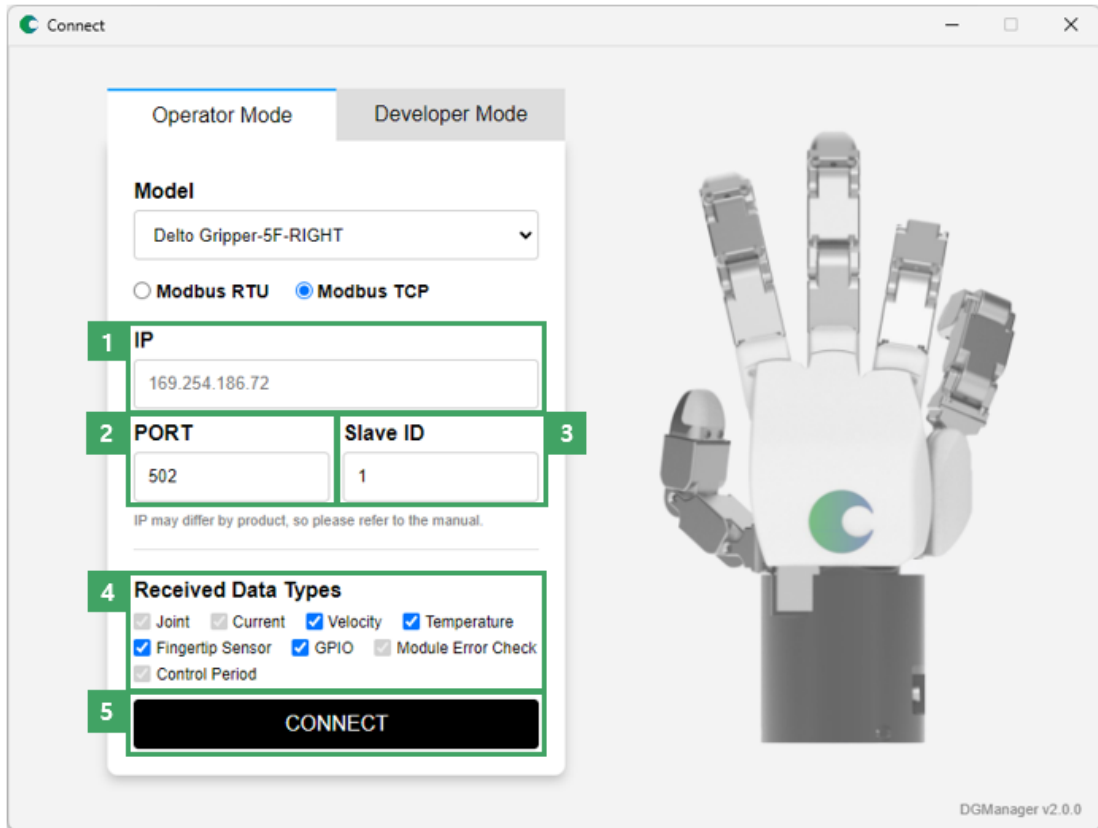
[10.1.1 SetGripperSystem](#)

[10.1.3 ConnectToGripper](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

4.1.2.2 Modbus TCP 연결 설정

기능 설명



Modbus TCP 연결 예시

연결할 그리퍼의 모델이 DG-3F-M, 설정한 IP 주소는 169.254.186.72, 포트 502, Slave ID 1인 경우, 먼저 'Model'에서 'Delto Gripper-3F-M'를 선택합니다. 이후 'Modbus TCP'를 선택한 후 하단의 각 항목에 해당하는 값을 입력합니다. 'IP'에 '169.254.186.72'를 입력, 'PORT'에 '502'를 입력, 'Slave ID'에 '1'을 입력한 후, 마지막으로 'CONNECT' 버튼을 클릭합니다. 정상적으로 통신이 연결이 되면 메인 화면으로 넘어갑니다.

항목

번호	명칭	설명
1	IP	그리퍼 통신 연결을 위한 IP 입력 □ 그리퍼 기본 IP: '169.254.186.72' □ DG-5F-L 기본 IP: '169.254.186.73'
2	PORT	그리퍼 통신 연결을 위한 포트 입력 □ 포트 범위: 1 – 65535 □ 그리퍼 기본 포트: 502
3	Slave ID	Modbus Slave ID 입력 □ Slave ID 범위: 1 – 255
4	Received Data Types	수신받을 데이터 타입을 선택
5	CONNECT	설정된 값으로 그리퍼 통신 연결

참고

10.1.1 SetGripperSystem

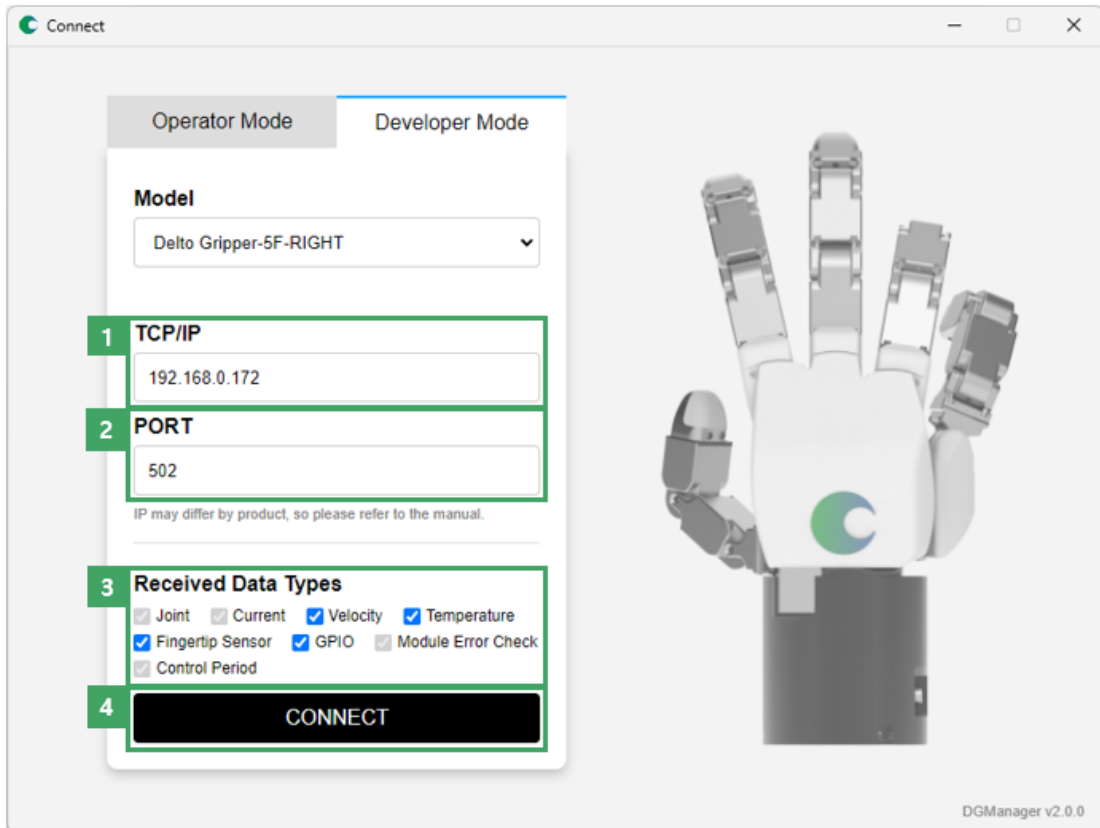
10.1.3 ConnectToGripper

* 에러 발생 시 '부록 2 에러 리스트' 참조

4.1.3 개발자 모드 연결

4.1.3.1 개발자 모드 TCP/IP 세팅 입력

기능 설명



개발자 모드 TCP/IP 연결 예시

연결할 그리퍼의 모델이 DG-3F-M, IP는 169.254.186.72, 포트 502인 경우, 먼저 'Model'에서 'Delto Gripper-3F-M'을 선택합니다. 이후 'TCP/IP'에 '169.254.186.72'를 입력하고, 'PORT'에 502를 입력합니다. 이후 Slave ID란에 1을 입력한 후, 마지막으로 CONNECT 버튼을 클릭하여 연결을 완료합니다.

항목

번호	명칭	설명
1	TCP/IP	그리퍼 통신 연결을 위한 IP 입력 □ 그리퍼 기본 IP: '169.254.186.72' □ DG-5F-L 기본 IP: '169.254.186.73'
2	PORT	그리퍼 통신 연결을 위한 포트 입력 □ 포트 범위: 1 ~ 65535 □ 그리퍼 기본 포트: 502
3	Received Data Types	수신받을 데이터 타입을 선택
4	CONNECT	설정된 값으로 그리퍼 통신 연결

참고

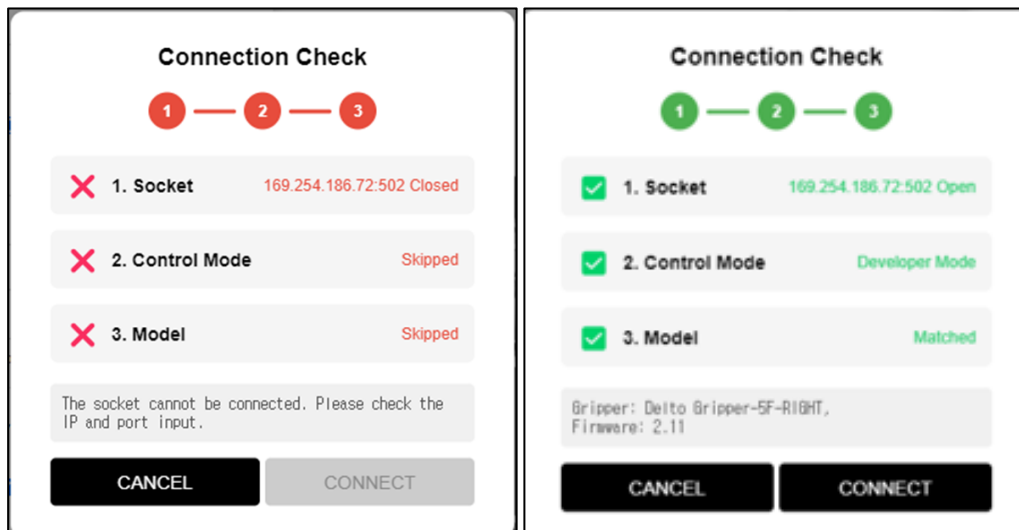
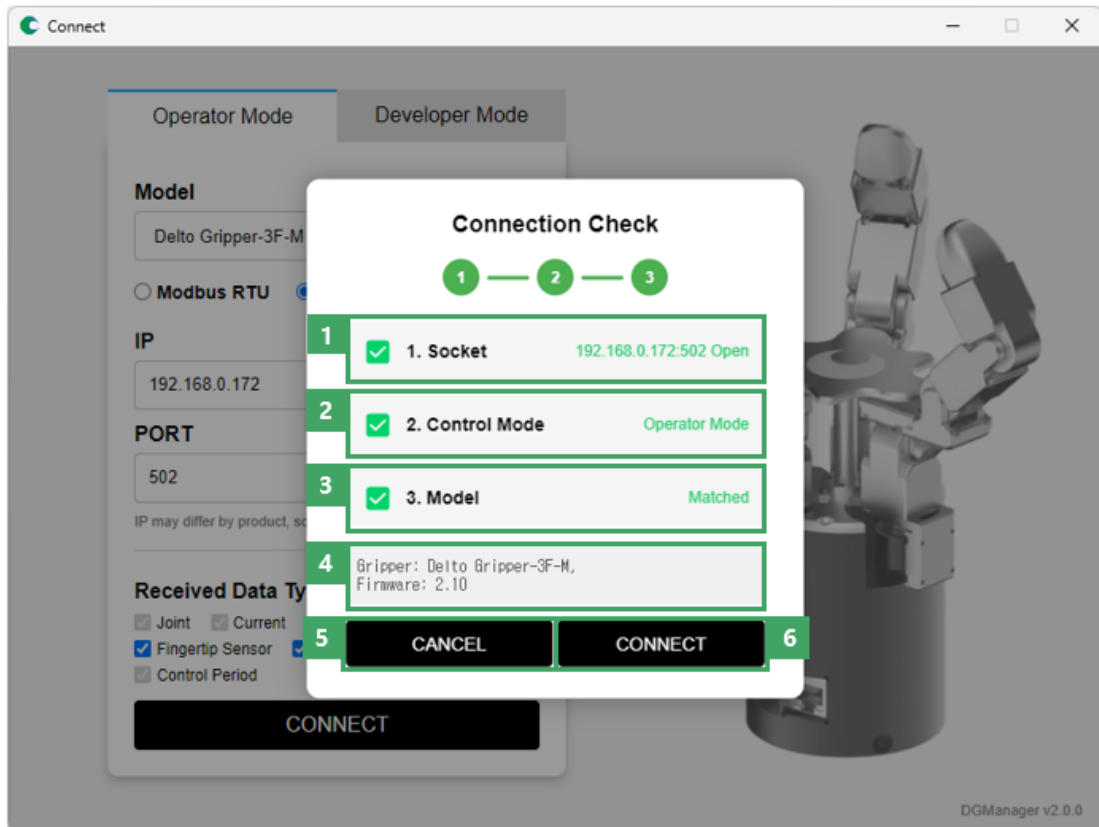
[10.1.1 SetGripperSystem](#)

[10.1.3 ConnectToGripper](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

4.1.4 연결 확인

기능 설명



연결 확인은 Delto Gripper와 접속을 하기위해 현재 Socket 상태, 제어 모드, 모델 정보가 올바른지 확인하는 단계입니다.

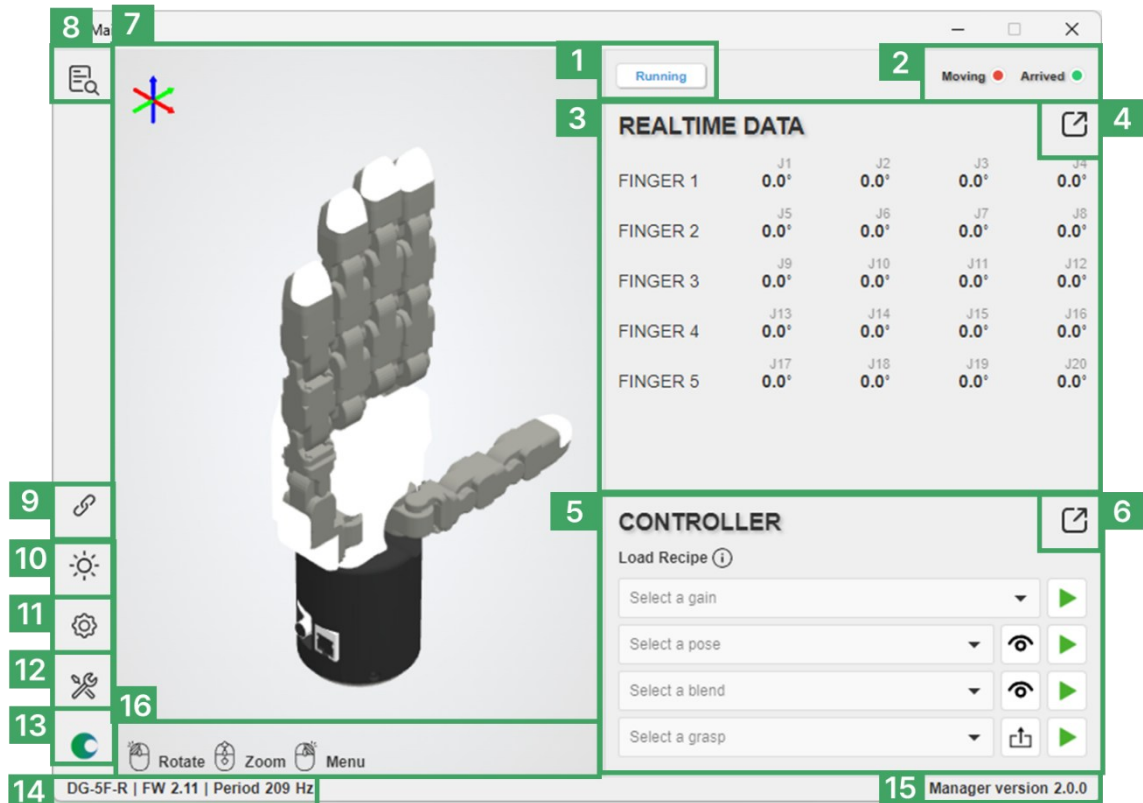
항목

번호	명칭	설명
1	Socket	현재 장치와 그리퍼간 IP 대역 확인
2	Control Mode	그리퍼 제어모드와 DGManager 선택한 제어 모드 비교
3	Model	그리퍼 모델과 DGManager 에 선택한 모델 비교
4	Log	각 항목별 로그 디스플레이
5	CANCEL	현재 창 나가기
6	CONNECT	그리퍼 연결 시도

4.2 메인 화면

4.2.1 화면 구성

기능 설명



메인 화면은 그리퍼의 전체 기능을 실행할 수 있는 화면입니다. 그리퍼 제어를 켜고 끌 수 있으며, 실시간 각도 정보를 3D 시뮬레이션으로 확인할 수 있습니다. 또한, 레시피를 통해 모션을 미리 검토 후 실행할 수 있고, 테마 변경, 연결 해제, 인포메이션 등의 부가기능을 사용할 수 있습니다.

메인 화면에서 그리퍼 시스템 시작/정지 항목을 통해 손쉽게 제어할 수 있습니다. 'Ready' 상태에서 'Start' 선택하면 시스템이 활성화되어 각종 명령을 수신할 준비가 되고, 'Running' 상태에서 'Stop' 상태를 선택하면 그리퍼의 동작이 중지되어 제어를 종료할 수 있습니다.

항목

번호	명칭	설명
1	그리퍼 시스템 시작/정지	☐ Ready: 그리퍼 대기 상태 ☐ Running: 그리퍼 시스템 활성화 상태 ☐ Start: 그리퍼 시스템 활성화 수행 ☐ Stop: 그리퍼 시스템 정지 수행
2	현재 상태 표시	☐ Moving: 현재 그리퍼가 움직이고 있으면 초록색, 정지해 있으면 빨간색으로 표시. ☐ Arrived: 타겟 값에 설정 범위 내에 도달하면 초록색, 아니면 빨간색으로 표시
3	REAL-TIME	각 관절의 실시간 위치를 표시
4	모니터링 화면	제 5 장 모니터링 화면 참조
5	CONTROLLER	저장된 레시피를 확인, 선택 레시피를 실행
6	동작, 레시피 화면	제 6 장 동작 화면 과 제 7 장 레시피 설정 화면 열기
7	시뮬레이션 화면	현재 그리퍼의 위치를 받아 실시간 렌더링 관절을 마우스로 드래그하여 관절 제어 가능
8	스파이 화면	8.3 스파이 참조
9	그리퍼 통신 연결 해제	그리퍼 통신 연결 해제
10	테마 변경	밝은 / 어두운 테마 중 하나로 변경
11	그리퍼 설정	그리퍼 8.1 설정 화면 열기
12	그리퍼 도구	8.2 도구 참조
13	프로그램 정보	프로그램 정보 표시 창 열기
14	그리퍼 정보	Model / Firmware Version / Communication Period
15	Manager 버전	현재 매니저 프로그램 버전 표시
16	시뮬레이션 조작 버튼	시뮬레이션 화면 조작 및 관절 색 지정

참고

[제 5 장 모니터링 화면](#)

[제 6 장 동작 화면](#)

[제 7 장 레시피 설정 화면](#)

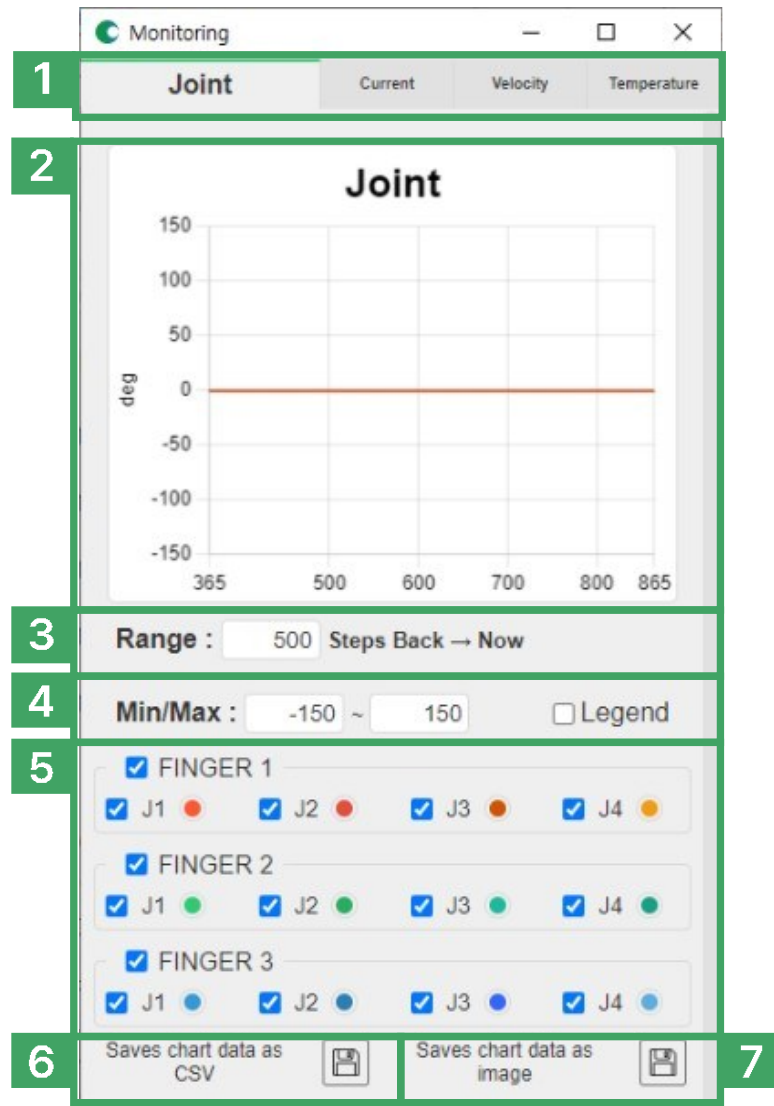
[제 8 장 부가 기능 화면](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

제 5 장 모니터링 화면

5.1 화면 구성

기능 설명



모니터링 화면은 선택한 그래프 항목들을 실시간으로 표시하고, 데이터를 저장할 수 있는 기능을 제공합니다. 관절, 전류, 속도, 온도의 항목을 선택해 실시간 그래프로 확인할 수 있으며, 그래프의 가로 범위(저장할 데이터 개수)와 세로 범위를 설정할 수 있습니다. 또한, 그래프에 표시할 관절을 선택하고, 현재 그래프 데이터를 CSV 파일 또는 이미지 PNG 파일로 저장할 수 있습니다.

항목

번호	명칭	설명
1	그래프 항목 선택	표시할 데이터를 선택 ☐ Joint 관절 ☐ Current 전류 ☐ Velocity 속도 ☐ Temperature 온도
2	실시간 그래프 화면	실시간으로 받는 데이터를 그래프로 표시
3	Range	그래프의 X 축의 표시 데이터 개수를 설정
4	Min/Max	그래프의 Y 축의 데이터 범위 설정
5	관절 선택 및 표시 설정	그래프 상에 표시할 관절들을 선택
6	Saves chart data as CSV	그래프 현재 데이터를 CSV 파일 저장
7	Saves chart data as image	그래프 현재 이미지를 PNG 파일 저장

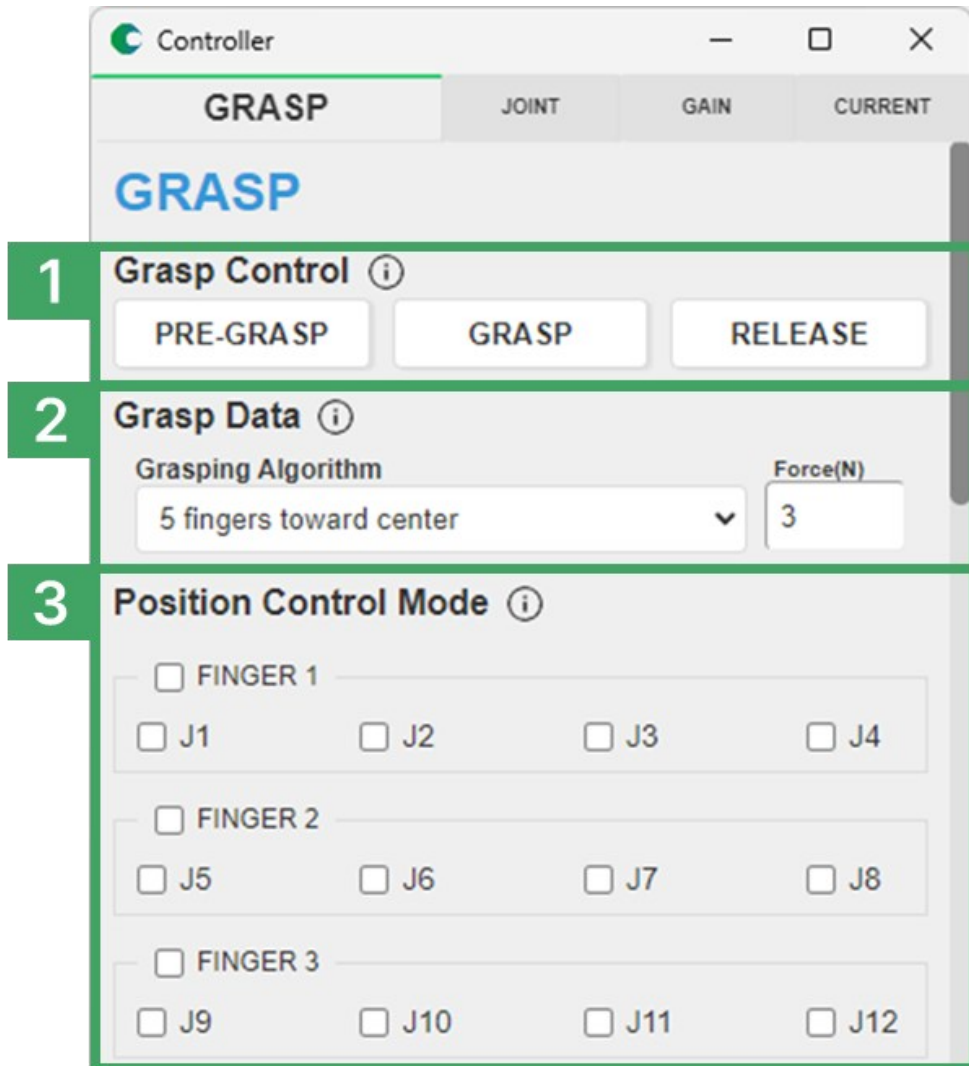
참고

Velocity, Temperature 는 DG-3F-B 에서 제공되지 않음

제 6 장 동작 화면

6.1 그랩 모드

6.1.1 화면 구성



번호	명칭	설명
1	Grasp Control	GRASP 및 RELEASE 수행
2	Grasp Data	그랩 모드 및 그랩 세기 설정
3	Position Control Mode	각 관절을 위치 제어 모드로 설정

6.1.2 그랩 조작

기능 설명



그랩 및 릴리즈를 조작할 수 있는 기능입니다.

항목

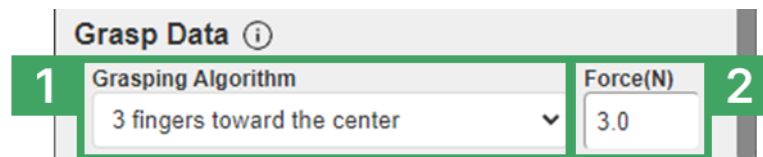
번호	명칭	설명
1	PRE-GRASP	그랩의 테스트를 위한 예시 포즈 불러오기
2	GRASP	그랩은 6.1.3 그랩 데이터 설정 그랩 데이터 설정에서 설정한 값으로 그랩 수행
3	RELEASE	릴리즈는 6.2.2 위치 제어 동작 마지막 목표 관절 위치 값으로 위치 이동 수행

참고

* 에러 발생 시 '부록 2 에러 리스트' 참조

6.1.3 그랩 데이터 설정

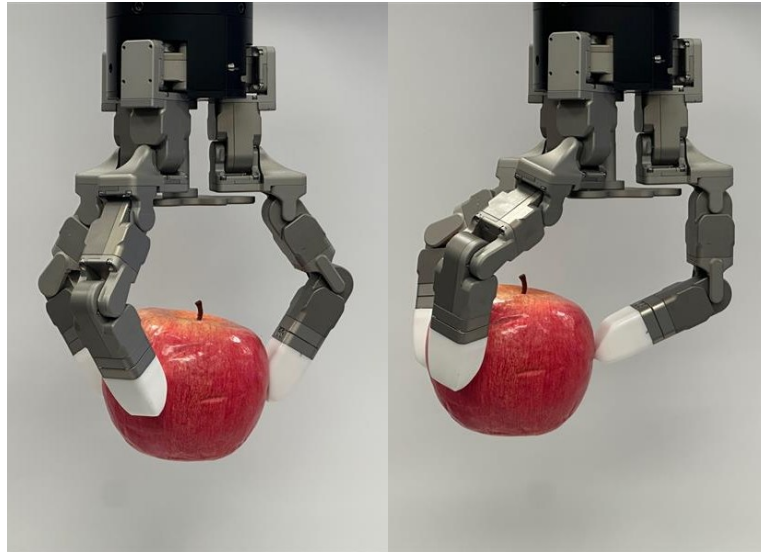
기능 설명



그랩 데이터를 설정할 수 있는 기능입니다. 해당 기능은 DG-1F-M 모델에서는 사용이 불가능 합니다. 각 모델별 Grasping Algorithm을 제공하며, Grasping Algorithm마다 저장된 PRE-GRASP 포즈가 다릅니다. 각 Grasping Algorithm의 PRE-GRASP에 대한 관절 위치 정보는 '6.2.2 위치 제어 동작' 혹은 '8.3 스파이'를 통해 확인하실 수 있습니다.

그랩은 기본적으로 실시간 힘 제어 그랩 알고리즘이 작동합니다. 힘 제어 그랩 알고리즘은 그랩에 사용되는 각 손가락 끝 단이 손가락 간 중심을 추종합니다. 이는 실시간으로 작동되기 때문에 외부 충격으로 그리퍼 관절 위치가 변경되더라도 변경된 자세에서 그랩 자세를 유지하려고 합니다. 특히 모든 관절에 대해 '6.1.4 위치 제어 모드 설정'의 위치 제어 모드를 해제했을

때 이러한 성능을 쉽게 확인하실 수 있습니다.



[그리퍼 관절 위치가 변경되어도 현재 위치에서 그랩 자세를 유지]

항목

번호	명칭	설명
1	Grasping Algorithm	9.2.9 enum DG_GRASP_MODE 참고
2	Force	그랩 세기 설정 (단위: N)

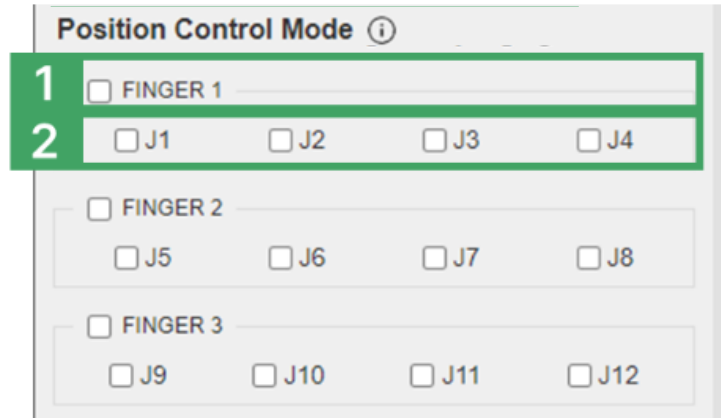
참고

11.1 그랩 설정 함수

- * 사용 불가능한 모델: DG-1F-M
- * 에러 발생 시 '부록 2 에러 리스트' 참조

6.1.4 위치 제어 모드 설정

기능 설명



The image shows a 'Position Control Mode' dialog box with a green border. It contains three sections for FINGER 1, FINGER 2, and FINGER 3. Each section has a header checkbox and a row of joint checkboxes (J1-J4 for FINGER 1, J5-J8 for FINGER 2, and J9-J12 for FINGER 3). The first two rows are highlighted with a green box and numbered 1 and 2.

위치 제어 모드로 설정된 관절은 그랩 수행 시 그랩을 하지 않고 위치 제어를 수행합니다. 이는 그랩 수행 중 특정 관절을 움직이지 않고 싶을 때 사용합니다.

만약 J1, J2를 체크 표시하고 난 후 '6.1.2 그랩 조작'에 그랩을 수행하게 되면, 관절 1번과 관절 2번은 그랩 알고리즘에서 제외되며 목표 위치 값으로 위치 이동만 수행하게 됩니다.

항목

번호	명칭	설명
1	FINGER 1 – FINGER n	버튼 선택 시 n번 손가락에 해당되는 모든 관절을 선택
2	J1 – Jn	버튼 선택 시 n에 해당되는 관절 선택

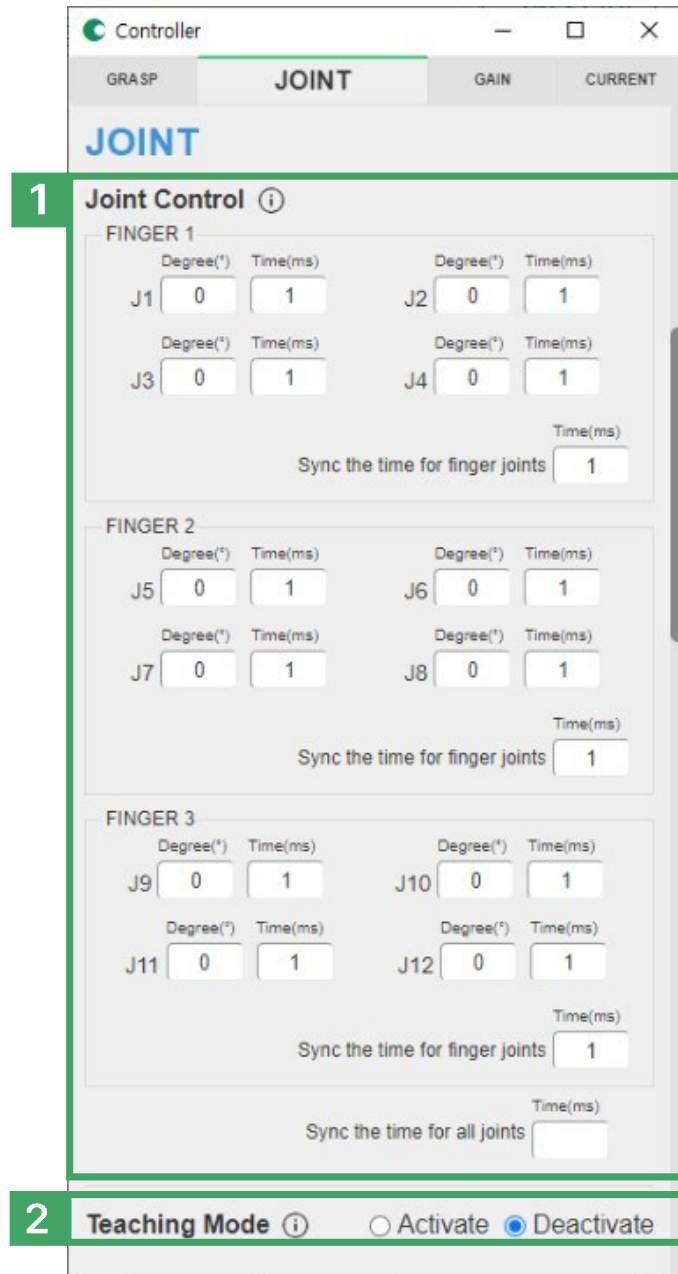
참고

11.7 위치 제어 모드 설정

- * 사용 불가능한 모델: DG-1F-M
- * DG-4F-M의 베이스 관절은 17, 18번 관절로 제어
- * 에러 발생 시 '부록 2 에러 리스트' 참조

6.2 위치 제어

6.2.1 화면 구성



1 Joint Control ⓘ

FINGER 1

Joint	Degree(°)	Time(ms)
J1	0	1
J2	0	1
J3	0	1
J4	0	1

Sync the time for finger joints Time(ms)

FINGER 2

Joint	Degree(°)	Time(ms)
J5	0	1
J6	0	1
J7	0	1
J8	0	1

Sync the time for finger joints Time(ms)

FINGER 3

Joint	Degree(°)	Time(ms)
J9	0	1
J10	0	1
J11	0	1
J12	0	1

Sync the time for finger joints Time(ms)

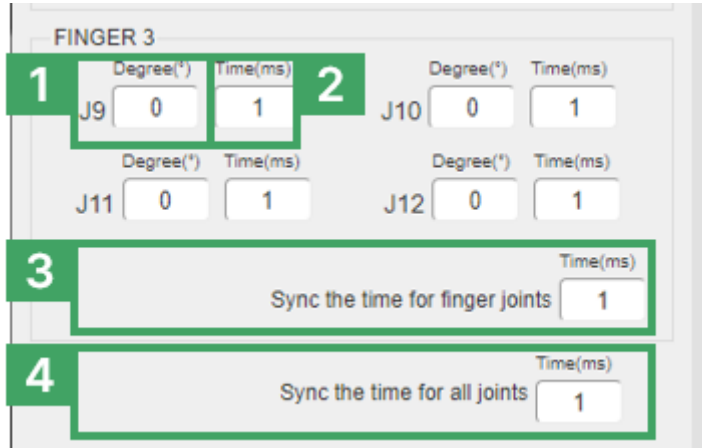
Sync the time for all joints Time(ms)

2 Teaching Mode ⓘ ☐ Activate ☒ Deactivate

번호	명칭	설명
1	Joint Control	관절을 위치제어하기 위한 목표 값 입력
2	Teaching Mode	직접 교시를 위한 티칭 모드 설정

6.2.2 위치 제어 동작

기능 설명



The screenshot shows the 'FINGER 3' control panel with four numbered steps highlighted in green:

- Step 1:** Sets the target position (Degree) and time (ms) for joint J9. Values shown: J9 Degree(°) = 0, Time(ms) = 1.
- Step 2:** Sets the target position (Degree) and time (ms) for joint J10. Values shown: J10 Degree(°) = 0, Time(ms) = 1.
- Step 3:** Sync the time for finger joints. Value shown: 1.
- Step 4:** Sync the time for all joints. Value shown: 1.

위치 제어 동작은 그리퍼의 관절을 위치 이동할 수 있는 기능입니다. 위치 이동을 하기 위해서는 목표 값을 설정해야 합니다. 목표 값의 단위는 'degree'입니다. 또한 위치 이동 시 목표까지 이동할 시간을 설정할 수 있습니다. 이동 시간 단위는 'ms'입니다.

만약 J1의 Time을 1000ms로 입력하고 degree 값을 60°로 입력하게 되면, J1에 해당하는 첫번째 관절은 현재 위치에서 60°까지 1000ms 시간동안 이동하게 됩니다. 여기서 설정하는 'Time'은 개발자 모드에서 사용 시 컨트롤러 성능에 따라 정확한 수치가 나오지 않을 수 있으며, 관절에 전류 제어 값에 따라서도 정확한 수치가 나오지 않을 수 있습니다.

항목

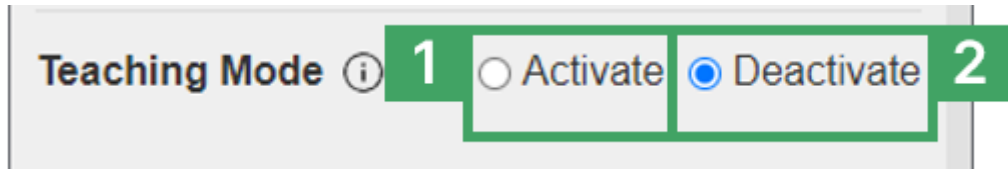
번호	명칭	설명
1	J1-Jn Degree	목표 위치 값 입력 (단위: degree)
2	J1-Jn Time	목표 위치 값까지 이동 시간 입력 (단위: ms)
3	Sync the time for finger joints	입력한 손가락에 해당하는 관절들의 이동 시간 입력 (단위: ms)
4	Sync the time for all joints	모든 관절들의 이동 시간 입력 (단위: ms)

참고

- * 모델에 따라 관절과 손가락 개수 상이
- 11.6 동작 시간 설정
- 11.7 위치 제어 모드 설정
- * 에러 발생 시 '부록 2 에러 리스트' 참조

6.2.3 티칭 모드

기능 설명



티칭 모드는 그리퍼를 직접 교시할 수 있는 기능입니다. 작업자는 그리퍼 관절의 목표값을 입력하여 그리퍼를 티칭할 수 있고, 티칭 모드를 활용하여 직접 교시를 통해 티칭을 수행할 수도 있습니다. 티칭 모드를 사용하기 위해 'Activate' 버튼을 선택하게 되면, 그리퍼는 티칭 모드 상태가 되며 사용자는 그리퍼를 물리적으로 조작하여 원하는 위치로 이동시킬 수 있습니다. 이는 현재 위치를 저장하는 레시피 포즈를 만드는데 유용합니다.

항목

번호	명칭	설명
1	Activate	티칭 모드 활성화
2	Deactivate	티칭 모드 비활성화

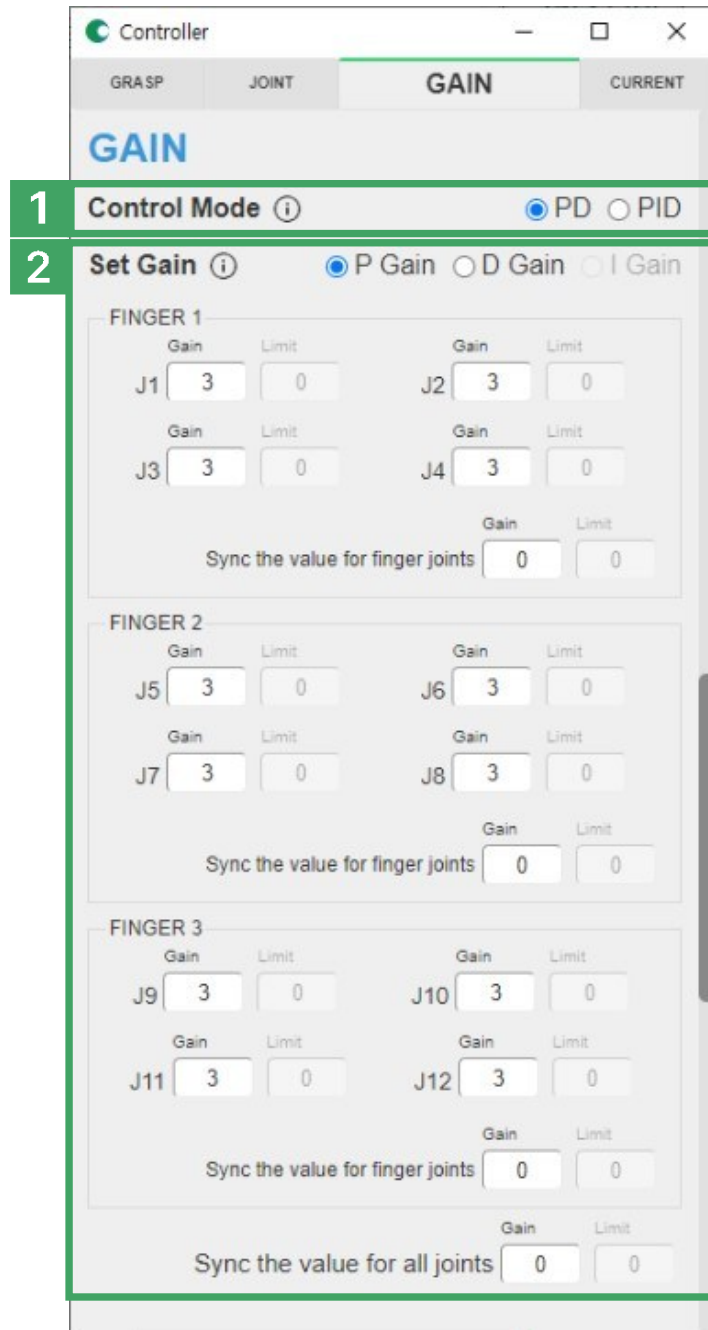
참고

[13.1 ManualTeachMode](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

6.3 게인 설정

6.3.1 화면 구성



Controller

GRASP JOINT **GAIN** CURRENT

GAIN

1 **Control Mode** ☒ PD ☐ PID

2 **Set Gain** ☒ P Gain ☐ D Gain ☐ I Gain

FINGER 1

Joint	Gain	Limit
J1	3	0
J2	3	0
J3	3	0
J4	3	0

Sync the value for finger joints

FINGER 2

Joint	Gain	Limit
J5	3	0
J6	3	0
J7	3	0
J8	3	0

Sync the value for finger joints

FINGER 3

Joint	Gain	Limit
J9	3	0
J10	3	0
J11	3	0
J12	3	0

Sync the value for finger joints

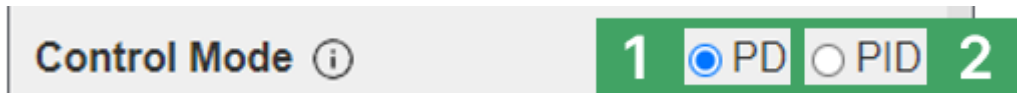
Sync the value for all joints

게인 설정은 그리퍼 관절의 위치 이동을 목적에 맞게 사용하기 위한 튜닝 과정입니다. 튜닝 방법에 따라 그리퍼의 역구동성을 좋게 하거나, 위치 정밀도를 올리는 기능을 선택하실 수 있습니다.

번호	명칭	설명
1	Control Mode	PD, PID 게인 모드 설정
2	Set Gain	각 관절 마다 PID 게인값 설정

6.3.2 게인 모드 설정

기능 설명



게인 모드 설정은 PD, PID 제어 모드를 설정할 수 있습니다. Delto Gripper 시리즈는 기본적으로 PD제어로 동작합니다. 이는 물체와 그리퍼가 충돌 상황에서 물리적인 손상에 유연하게 대응할 수 있기 때문입니다. 반면 위치 정밀도를 높이기 위해서는 PID 제어모드를 사용하는 것이 좋습니다.

항목

번호	명칭	설명
1	PD	PD 제어로 설정
2	PID	PID 제어로 설정

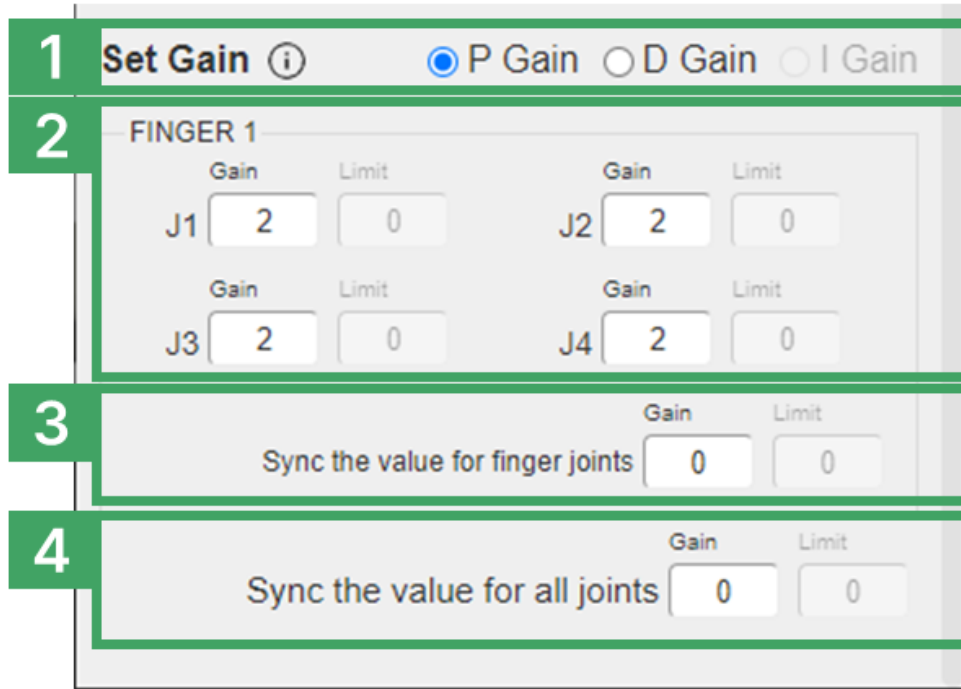
참고

[11.5.1 SetControlPIDMode](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

6.3.3 게인 설정

기능 설명



The screenshot shows the 'Set Gain' interface with four numbered steps highlighted by green boxes:

- 1 Set Gain**: A header bar with a title 'Set Gain' and an information icon. Below it are three radio buttons: 'P Gain' (selected), 'D Gain', and 'I Gain'.
- 2 FINGER 1**: A section for configuring Finger 1. It contains four joint settings (J1, J2, J3, J4). Each joint has a 'Gain' input field (all set to 2) and a 'Limit' input field (all set to 0).
- 3**: A section for synchronizing values for finger joints. It has a label 'Sync the value for finger joints' and two input fields: 'Gain' (set to 0) and 'Limit' (set to 0).
- 4**: A section for synchronizing values for all joints. It has a label 'Sync the value for all joints' and two input fields: 'Gain' (set to 0) and 'Limit' (set to 0).

관절, 손가락, 전체 PID 게인을 설정합니다.

항목

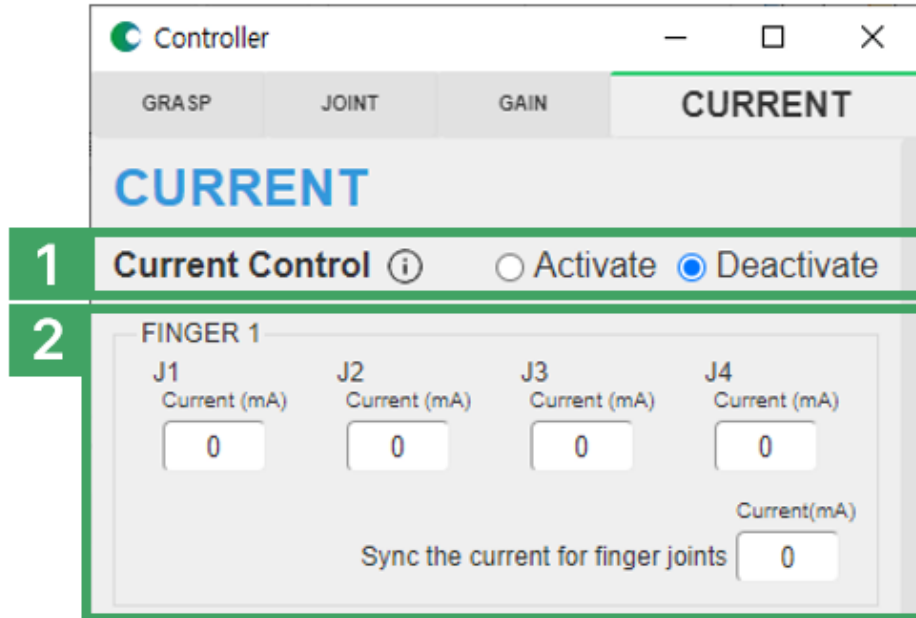
번호	명칭	설명
1	Set Gain	설정할 게인 선택
2	J1-Jn Gain	해당 관절의 게인값을 설정
3	Sync the value for finger joints	해당 손가락 관절들의 게인값을 일괄 설정
4	Sync the value for all joints	모든 관절들의 게인값을 일괄 설정

참고

- * 모델에 따라 관절과 손가락 개수 상이
- 11.2 게인 P 설정
- 11.3 게인 D 설정
- 11.4 게인 I 설정
- 11.5 게인 PID 설정
- * 에러 발생 시 '부록 2 에러 리스트' 참조

6.4 전류 설정

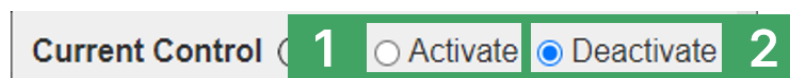
6.4.1 화면 구성



전류 설정을 이용하시면, 각 그리퍼 관절에 대해 전류제어를 설정할 수 있습니다. 전류 제어 모드에서는 PID 게인 값 설정이 불가능 합니다.

6.4.2 전류 모드 설정

기능 설명



전류 제어 모드를 활성화하면 각 관절은 전류 제어로 동작 합니다.

항목

번호	명칭	설명
1	Activate	전류제어 활성화
2	Deactivate	전류제어 비활성화

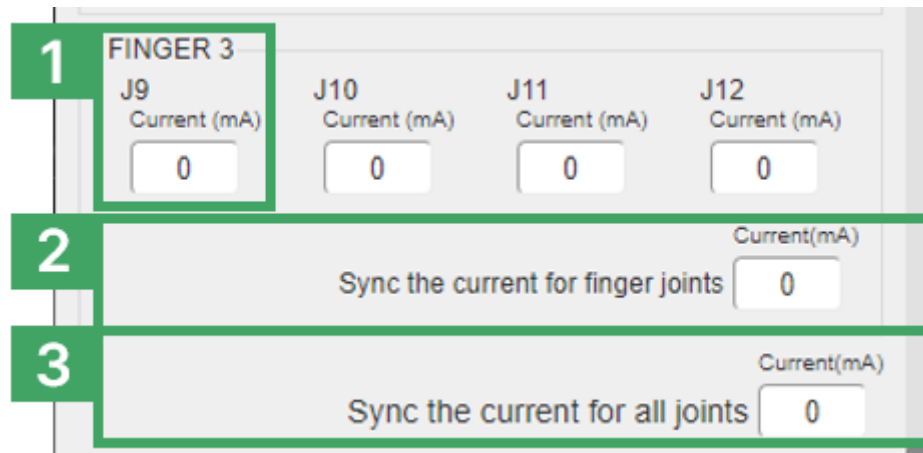
참고

13.4.1 SetCurrentControlMode

* 에러 발생 시 '부록 2 에러 리스트' 참조

6.4.3 전류 설정

기능 설명



The screenshot shows a software interface for setting currents. It is divided into three numbered sections:

- 1 FINGER 3**: Contains four input fields for J9, J10, J11, and J12, each labeled 'Current (mA)' and showing a value of 0.
- 2**: A section labeled 'Sync the current for finger joints' with a 'Current(mA)' input field showing 0.
- 3**: A section labeled 'Sync the current for all joints' with a 'Current(mA)' input field showing 0.

전류 설정값은 전류 제어 모드가 활성화됐을 때 해당 값을 반영합니다. 전류값은 M모델에서는 150mA, S모델에서는 360mA까지 입력이 가능합니다.

항목

번호	명칭	설명
1	J1-Jn Current	전류값 설정
2	Sync the current for finger joints	해당 손가락 관절들의 전류 값을 일괄 설정
3	Sync the current for all joints	모든 관절들의 전류 값을 일괄 설정

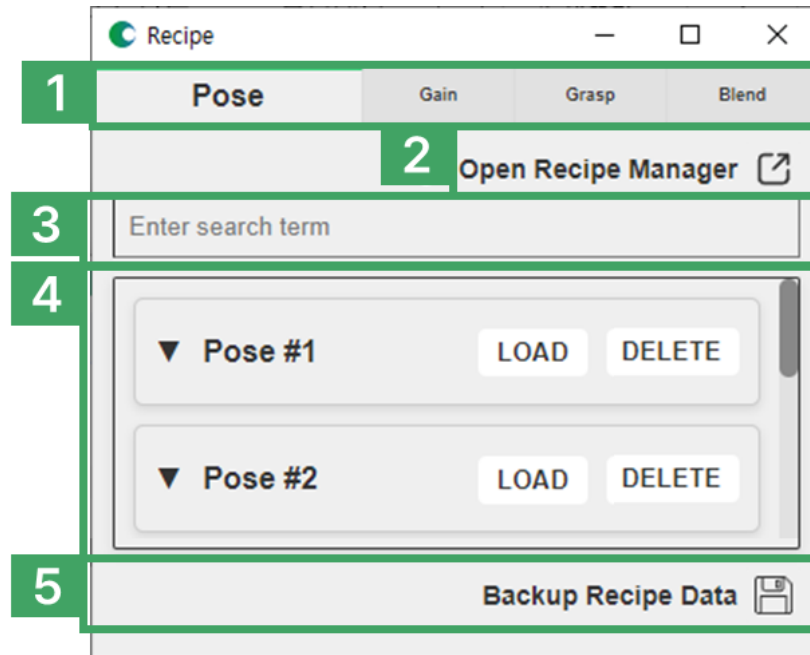
참고

13.4 전류 제어

* 에러 발생 시 '부록 2 에러 리스트' 참조

제 7 장 레시피 설정 화면

7.1 화면 구성



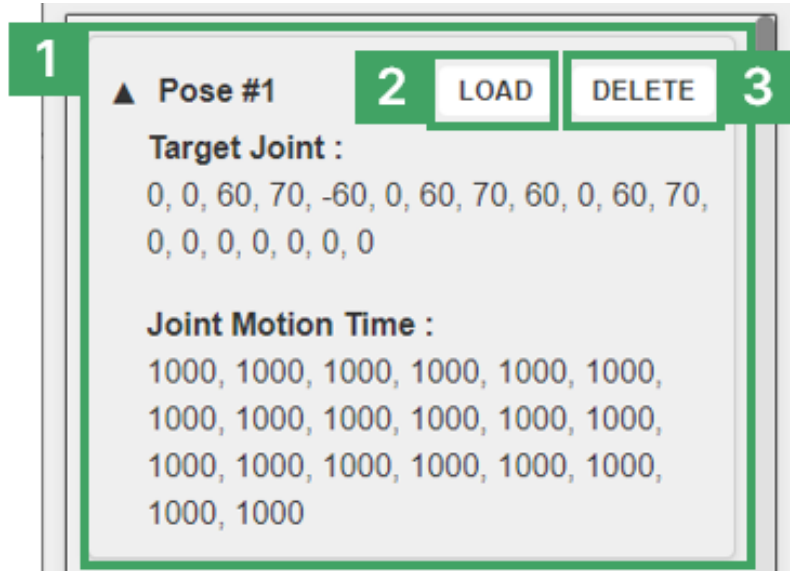
레시피 설정 화면은 현재 저장된 레시피를 표시합니다. 레시피를 검색하여 확인하고, LOAD 버튼을 눌러 해당 레시피를 수행하거나 DELETE를 눌러 해당 레시피를 삭제할 수 있습니다. 프로그램 내 설정된 레시피를 백업 기능을 통해 그리퍼 혹은 PC에 저장할 수 있습니다. 사용자 모드에서는 EEPROM WRITE로 표기되어 있으며, EEPROM WRITE를 수행하면, 현재 설정한 데이터가 그리퍼에 메모리에 영구 저장됩니다. 이는 그리퍼 전원을 재부팅하더라도 남아있는 비휘발성 메모리를 의미합니다.

항목

번호	명칭	설명
1	Recipe Menu	레시피 항목을 선택
2	Open Recipe Manager	레시피를 편집하거나 추가하는 매니저 열기
3	Search Recipe	레시피를 검색하기 위한 기능
4	Show Recipe List	저장된 레시피 항목 리스트 표시
5	Backup Recipe Data	사용하는 모드에 따라 레시피 항목을 백업 ☐ 사용자 모드: EEPROM WRITE ☐ 개발자 모드: BACKUP RECIPE DATA

7.2 포즈 레시피

7.2.1 화면 구성

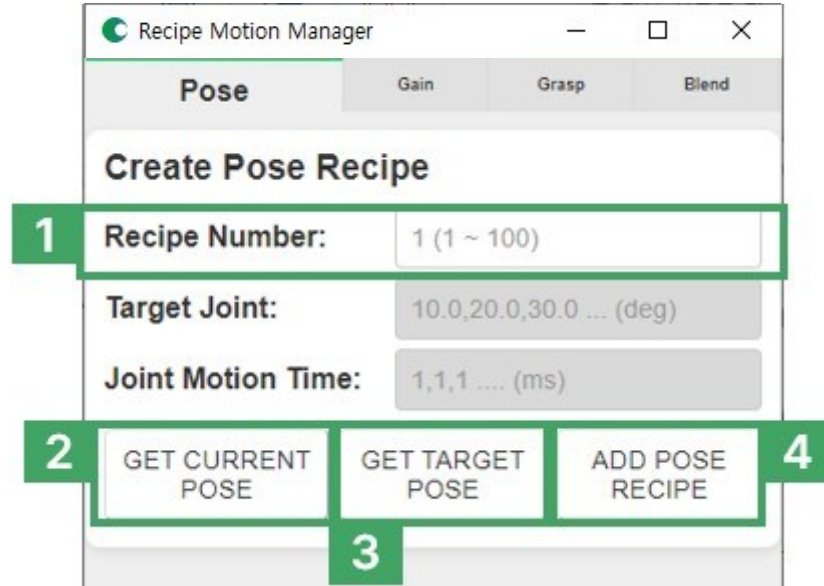


현재 저장된 포즈 레시피를 표시합니다. 포즈 레시피는 목표 관절 값, 동작 시간을 포함하고 있습니다. LOAD 버튼을 눌러 해당 레시피를 수행하거나 DELETE 버튼을 눌러 해당 레시피를 삭제할 수 있습니다.

항목

번호	명칭	설명
1	레시피 정보	레시피 정보 표시 □ 목표 관절 값 □ 동작 시간
2	LOAD	해당 레시피를 매니저 화면에 불러오며 그리퍼를 동작
3	DELETE	해당 레시피를 삭제

7.2.2 포즈 레시피 설정 방법



저장할 레시피 번호를 입력하고, 현재 관절 위치 값, 혹은 목표 관절 위치 값을 불러옵니다. 포즈 레시피 추가 버튼을 눌러 레시피를 저장합니다. GET TARGET POSE는 '6.2 위치 제어'에 입력된 Joint Degree 값을 불러옵니다.

항목

번호	명칭	설명
1	Recipe Number	저장할 레시피 번호 입력. □ 사용자 모드에서는 최대 100 개까지 저장
2	GET CURRENT POSE	현재 관절 위치 및 도착 시간 값 가져오기
3	GET TARGET POSE	목표 관절 위치 및 도착 시간 값 가져오기
4	ADD POSE RECIPE	설정한 레시피 번호로 포즈 레시피 저장

참고

12.1.1 UpdateRecipeCurrentPoseData
 12.1.2 UpdateRecipeTargetPoseData
 * 에러 발생 시 '부록 2 에러 리스트' 참조

7.2.3 포즈 레시피 동작 사용 방법

기능 설명



메인 화면에서 희망하는 포즈 레시피를 선택하고 불러옵니다.

항목

번호	명칭	설명
1	포즈 레시피 선택	포즈 레시피 선택
2	포즈 레시피 미리보기	시뮬레이션으로 포즈 레시피 움직임 확인 버튼을 누르는 동안 시뮬레이션이 동작
3	포즈 레시피 불러오기	포즈 레시피 불러오기

참고

[12.1.1 UpdateRecipeCurrentPoseData](#)

[12.1.2 UpdateRecipeTargetPoseData](#)

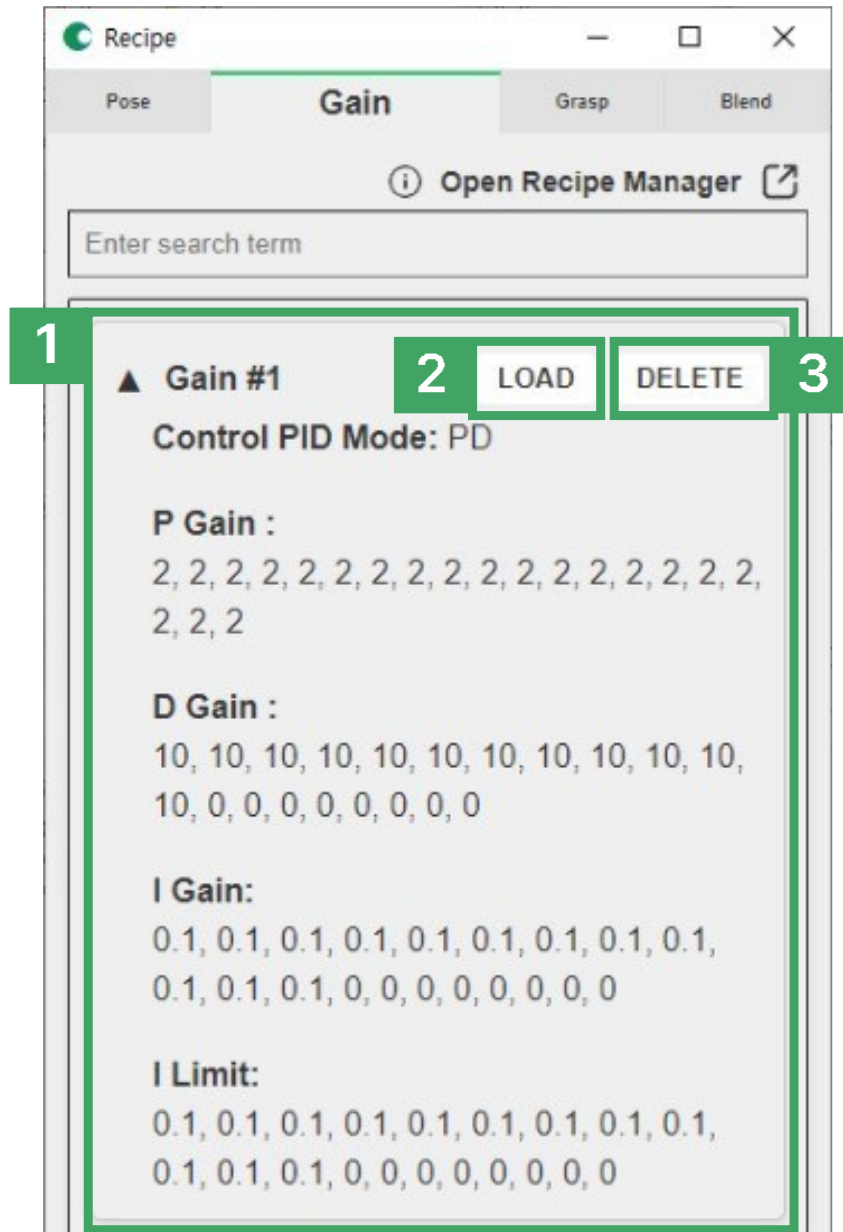
[12.1.3 LoadRecipePoseData](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

7.3 게인 레시피

7.3.1 화면 구성

기능 설명



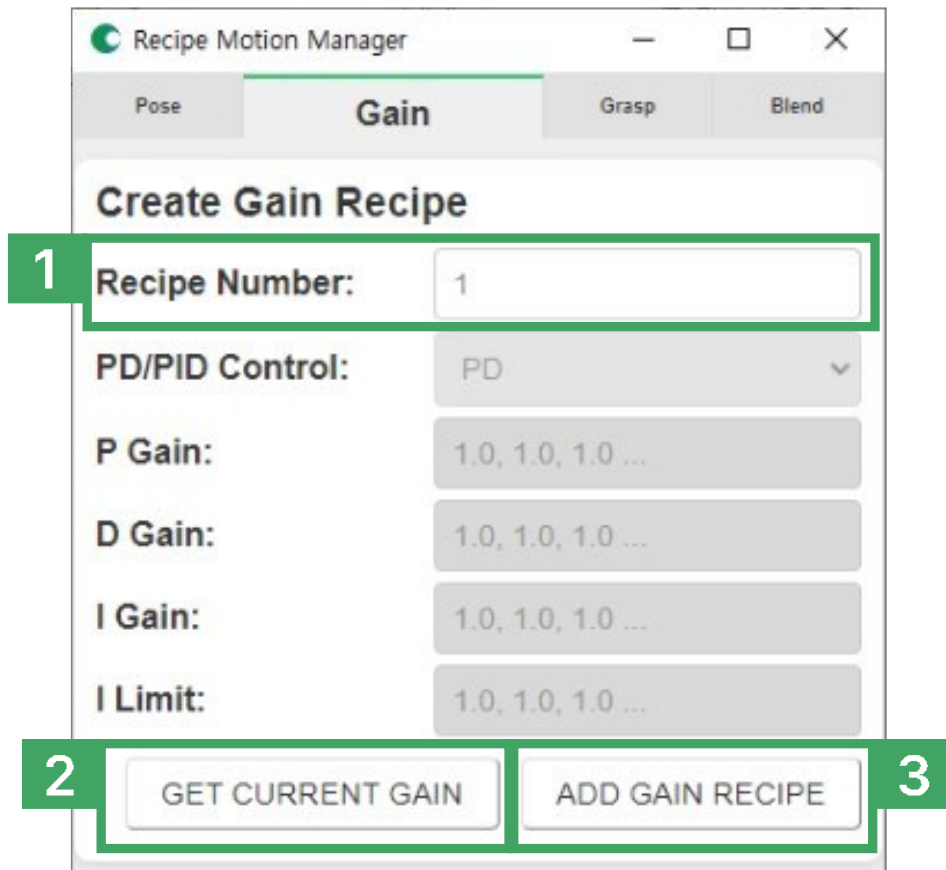
현재 저장된 게인 레시피를 표시합니다. 게인 레시피는 P 게인, D 게인, I 게인, 오차 적분 최대값을 포함합니다. LOAD 버튼을 눌러 해당 레시피를 수행하거나 DELETE를 눌러 해당 레시피를 삭제할 수 있습니다.

항목

번호	명칭	설명
1	레시피 정보	레시피 정보 표시 ☐ P 게인 ☐ D 게인 ☐ I 게인 ☐ I Limit. 오차 적분 최대값
2	LOAD	해당 레시피를 매니저 화면에 불러오며 그리퍼를 동작
3	DELETE	해당 레시피를 삭제

7.3.2 게인 레시피 설정 방법

기능 설명



The screenshot shows the 'Recipe Motion Manager' window with the 'Gain' tab selected. The 'Create Gain Recipe' section contains the following fields and controls:

- Recipe Number:** A text input field with the value '1'. A green box with the number '1' highlights this field.
- PD/PID Control:** A dropdown menu currently set to 'PD'.
- P Gain:** A text input field with the value '1.0, 1.0, 1.0 ...'.
- D Gain:** A text input field with the value '1.0, 1.0, 1.0 ...'.
- I Gain:** A text input field with the value '1.0, 1.0, 1.0 ...'.
- I Limit:** A text input field with the value '1.0, 1.0, 1.0 ...'.
- Buttons:** At the bottom, there are two buttons: 'GET CURRENT GAIN' and 'ADD GAIN RECIPE'. A green box with the number '2' highlights the 'GET CURRENT GAIN' button, and a green box with the number '3' highlights the 'ADD GAIN RECIPE' button.

희망하는 레시피 번호를 입력하고, 현재 게인값을 불러옵니다. 게인 레시피 추가 버튼을 클릭하여 레시피를 저장합니다.

항목

번호	명칭	설명
1	Recipe Number	저장할 레시피 번호 입력. □ 사용자모드에서는 최대 20 개까지 저장.
2	GET CURRENT GAIN	현재 게인값 가져오기
3	ADD GAIN RECIPE	설정된 레시피 번호로 게인 레시피 저장

참고

12.2.1 UpdateRecipeGainData

* 에러 발생 시 '부록 2 에러 리스트' 참조

7.3.3 게인 레시피 동작 사용 방법

기능 설명



메인 화면에서 희망하는 게인 레시피를 선택하고 불러옵니다.

항목

번호	명칭	설명
1	게인 레시피 선택	목록에서 게인 레시피를 선택
2	게인 레시피 불러오기	선택한 게인 레시피 불러오기

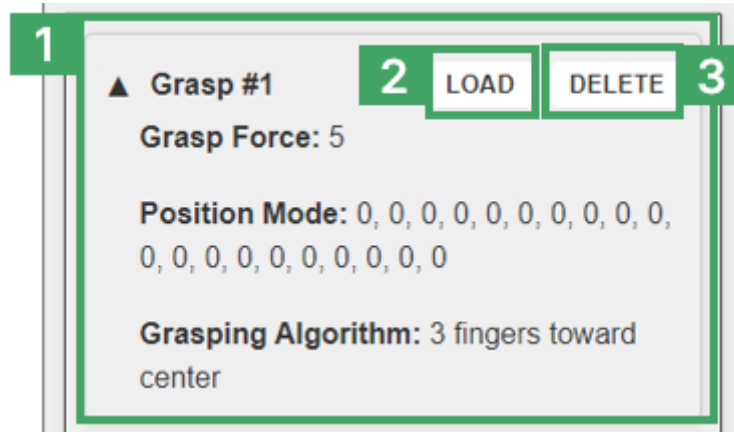
참고

12.2.2 LoadRecipeGainData

* 에러 발생 시 '부록 2 에러 리스트' 참조

7.4 그랩 레시피

7.4.1 화면 구성



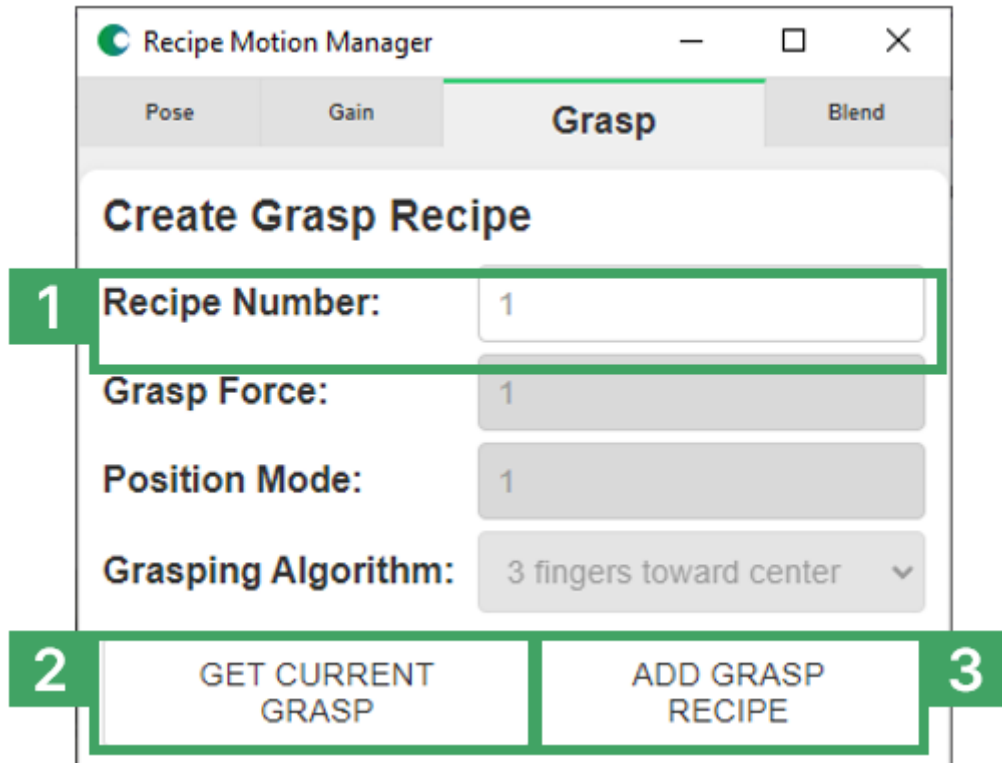
현재 저장된 그랩 레시피를 표시합니다. 그랩 레시피의 경우는 그랩 세기 설정, 위치 제어, 그랩 모드를 포함합니다. LOAD 버튼을 눌러 해당 레시피를 수행하거나 Delete를 눌러 해당 레시피를 삭제할 수 있습니다.

항목

번호	명칭	설명
1	레시피 정보	레시피 정보 표시 ☐ Grasp Force ☐ Position Mode ☐ Grasping Algorithm
2	LOAD	해당 레시피를 매니저 화면에 불러오며 그리퍼를 동작
3	DELETE	해당 레시피를 삭제

7.4.2 그랩 레시피 설정 방법

기능 설명



저장할 레시피 번호를 입력하고, 현재 그랩 설정 값을 불러옵니다. 그랩 레시피 추가 버튼을 클릭하여 레시피를 저장합니다.

항목

번호	명칭	설명
1	Recipe Number	저장할 레시피 번호 입력 □ 사용자모드에서는 최대 20 개까지 저장
2	GET CURRENT GRASP	현재 그랩 값 가져오기
3	ADD GRASP RECIPE	설정한 레시피 번호로 그랩 레시피 저장

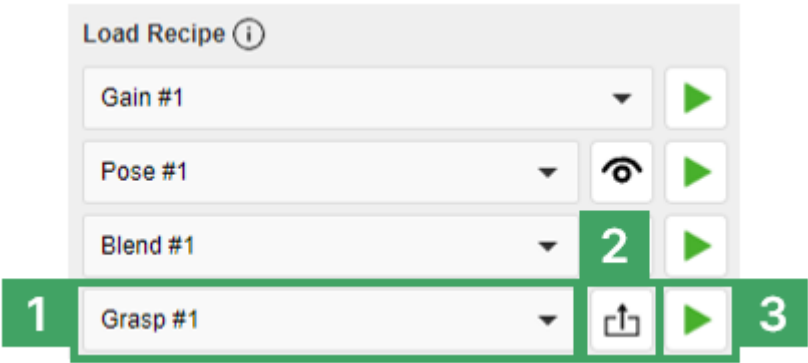
참고

[12.3.1 UpdateRecipeGraspData](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

7.4.3 그랩 레시피 동작 사용 방법

기능 설명



메인 화면에서 동작을 희망하는 레시피를 선택하고 불러옵니다.

항목

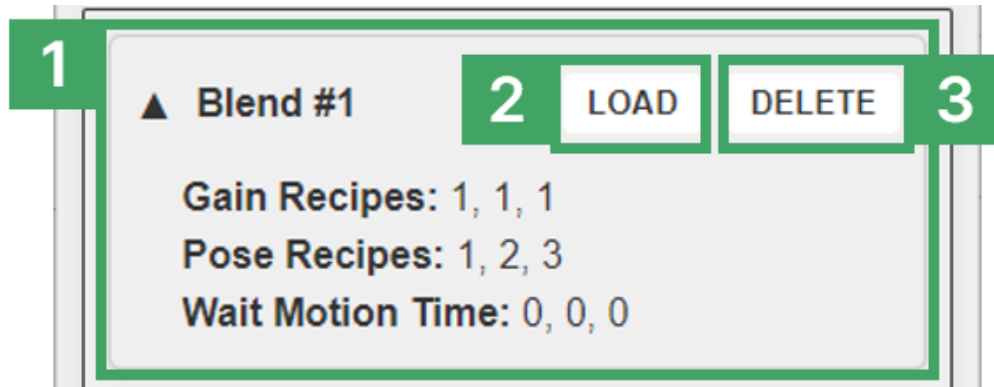
번호	명칭	설명
1	그랩 레시피 선택	목록에서 사용할 그랩 레시피 선택
2	LOAD	선택한 그랩 레시피 불러오기
3	GRASP	: 그랩 동작 수행 : 그랩 해제 수행

참고

12.3.2 LoadRecipeGraspData
* 에러 발생 시 '부록 2 에러 리스트' 참조

7.5 블렌드 레시피

7.5.1 화면 구성

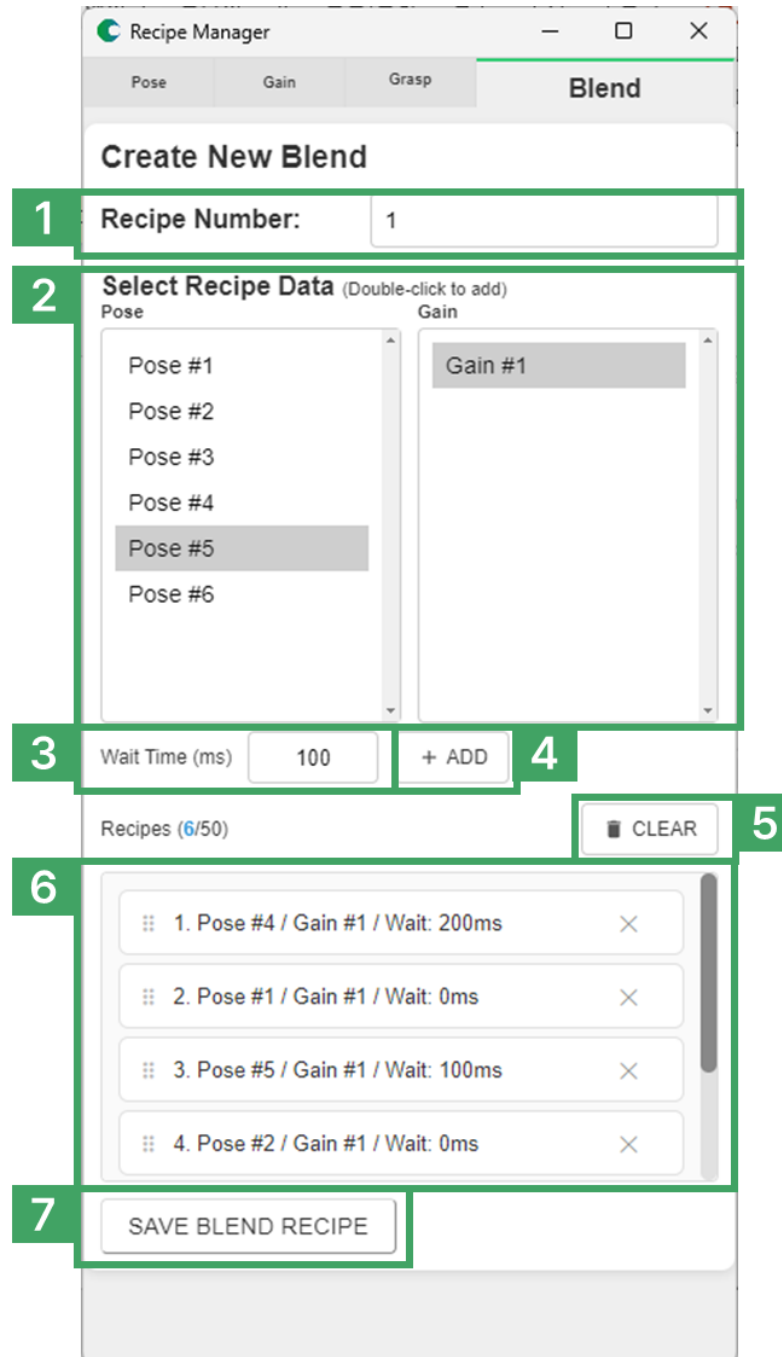


현재 저장된 레시피를 표시합니다. 블렌드 레시피의 경우는 게인 레시피, 포즈 레시피, 동작 시간을 포함하고 있습니다. LOAD 버튼을 눌러 해당 레시피를 불러오거나 DELETE를 눌러 해당 레시피를 삭제할 수 있습니다.

항목

번호	명칭	설명
1	블렌드 레시피 정보	레시피 정보 표시 □ Gain Recipes □ Pose Recipes □ Wait Motion Time(ms)
2	LOAD	현재 블렌드 레시피 불러오기
3	DELETE	현재 선택된 블렌드 레시피 삭제

7.5.2 블렌드 레시피 설정 방법



Recipe Manager

Pose Gain Grasp **Blend**

Create New Blend

1 Recipe Number: 1

2 Select Recipe Data (Double-click to add)

Pose Gain

Pose #1
Pose #2
Pose #3
Pose #4
Pose #5
Pose #6

Gain #1

3 Wait Time (ms) 100 4 + ADD

5 Recipes (6/50) CLEAR

6

1. Pose #4 / Gain #1 / Wait: 200ms ×

2. Pose #1 / Gain #1 / Wait: 0ms ×

3. Pose #5 / Gain #1 / Wait: 100ms ×

4. Pose #2 / Gain #1 / Wait: 0ms ×

7 SAVE BLEND RECIPE

저장할 블렌드 레시피 번호를 지정하고 포즈, 게인, 대기 시간을 설정해주고 4번 항목 'ADD SELECTED RECIPE' 버튼을 눌러 블렌드 모션을 추가합니다. 위 차레를 여러 번 반복하여 블렌드 모션들을 모두 추가하였으면 7번 항목 'SAVE BLEND RECIPE'을 눌러 블렌드 레시피를 저장합니다.

항목

번호	명칭	설명
1	Recipe Number	저장할 레시피 번호 입력 □ 사용자모드에서는 최대 10 개 저장
2	Select Recipe Data	포즈, 게인 레시피 선택
3	Wait Time	블렌드 모션 대기 시간 설정. 현재 레시피를 수행하고 입력된 시간만큼 대기한 후 다음 레시피를 수행
4	ADD	선택한 포즈, 게인 레시피, 대기 시간을 리스트에 추가 □ 하나의 블렌드에 최대 50 개 레시피 저장
5	CLEAR	레시피 리스트 초기화
6	Blend List	포즈 레시피, 게인 레시피, 동작시간이 추가된 블렌드 리스트
7	SAVE BLEND RECIPE	설정한 레시피 번호로 블렌드 레시피 저장

참고

[12.4.1 UpdateBlendJoint](#)
[12.4.2 ClearMoveBlendJoint](#)
[12.4.3 AddMoveBlendJoint](#)
[12.4.4 SetMoveBlendJoint](#)
[12.4.5 StartMoveBlendJoint](#)

7.5.3 블렌드 레시피 동작 사용 방법

기능 설명



메인 화면에서 희망하는 블렌드 레시피를 선택하고 불러옵니다.

항목

번호	명칭	설명
1	블렌드 레시피 선택	목록에서 블렌드 레시피 선택
2	블렌드 레시피 미리보기	시뮬레이션으로 포즈 레시피 움직임 확인 버튼을 누르는 동안 시뮬레이션이 동작
3	블렌드 레시피 불러오기	선택한 블렌드 레시피 불러오기

참고

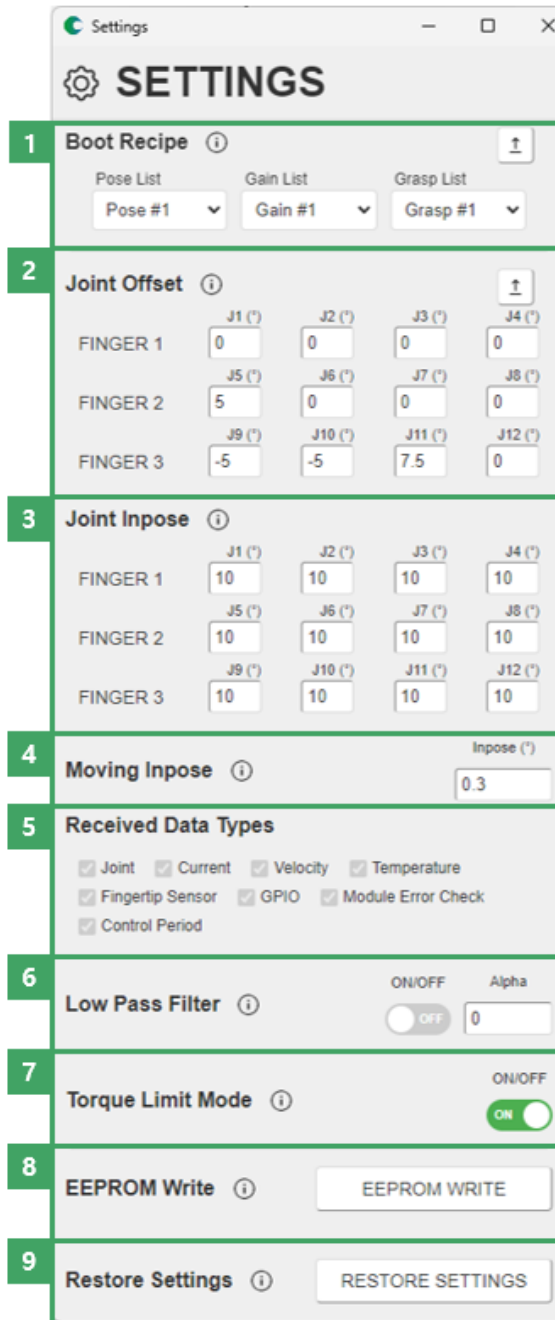
[12.4.5 StartMoveBlendJoint](#)

[12.4.6 StopMoveBlendJoint](#)

제 8 장 부가 기능 화면

8.1 설정

8.1.1 화면 구성



1 Boot Recipe ⓘ

Pose List: Pose #1 ▼ Gain List: Gain #1 ▼ Grasp List: Grasp #1 ▼

2 Joint Offset ⓘ

	J1 (°)	J2 (°)	J3 (°)	J4 (°)
FINGER 1	0	0	0	0
	J5 (°)	J6 (°)	J7 (°)	J8 (°)
FINGER 2	5	0	0	0
	J9 (°)	J10 (°)	J11 (°)	J12 (°)
FINGER 3	-5	-5	7.5	0

3 Joint Inpose ⓘ

	J1 (°)	J2 (°)	J3 (°)	J4 (°)
FINGER 1	10	10	10	10
	J5 (°)	J6 (°)	J7 (°)	J8 (°)
FINGER 2	10	10	10	10
	J9 (°)	J10 (°)	J11 (°)	J12 (°)
FINGER 3	10	10	10	10

4 Moving Inpose ⓘ Inpose (°): 0.3

5 Received Data Types

☒ Joint ☒ Current ☒ Velocity ☒ Temperature
☒ Fingertip Sensor ☒ GPIO ☒ Module Error Check
☒ Control Period

6 Low Pass Filter ⓘ ON/OFF: OFF Alpha: 0

7 Torque Limit Mode ⓘ ON/OFF: ON

8 EEPROM Write ⓘ EEPROM WRITE

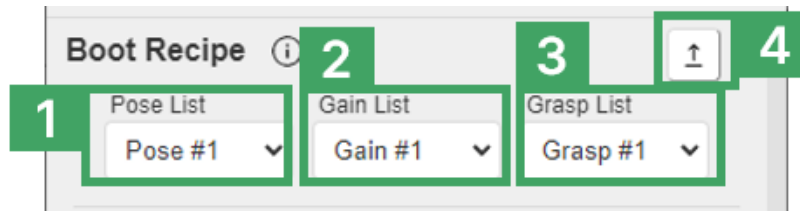
9 Restore Settings ⓘ RESTORE SETTINGS

세팅 화면은 그리퍼의 부트 레시피, 오프셋과 같은 그리퍼 세팅을 변경하는 화면입니다.

항목

번호	명칭	설명
1	Boot Recipe	그리퍼 제어 시작 시 레시피를 설정
2	Joint Offset	그리퍼 각 관절의 오프셋 값 설정
3	Joint Inpose	관절 도착 확인을 위한 범위 설정
4	Moving Inpose	관절 이동 확인을 위한 범위 설정
5	Received Data Types	그리퍼로부터 수신 받는 데이터 표시 연결 전에 설정 가능
6	Low Pass Filter	저역 통과 필터 설정
7	Torque Limit Mode	토크 제한 모드 설정. M 모델 전용 □ 사용 시 주의 필요
8	Backup Settings	사용하는 모드에 따라 백업을 수행 □ 사용자 모드: EEPROM WRITE □ 개발자 모드: BACKUP SETTINGS
9	Restore Settings	복원을 수행

8.1.2 Boot Recipe



부트 레시피는 그리퍼에 전원이 인가된 이후 제어를 시작할 때 적용할 레시피를 설정하는 기능입니다.

항목

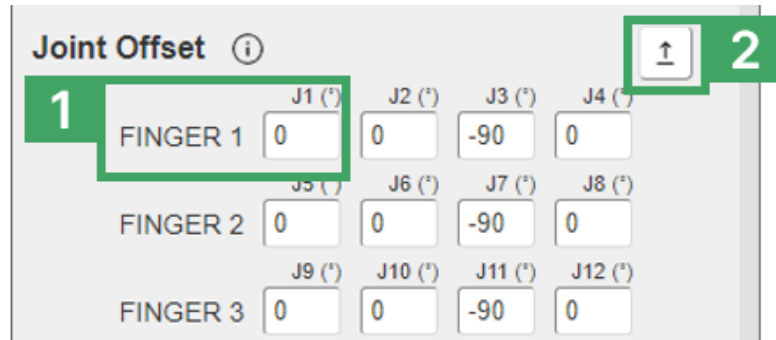
번호	명칭	설명
1	포즈 부트 레시피	부팅 시 설정할 포즈 레시피 설정
2	게인 부트 레시피	부팅 시 설정할 게인 레시피 설정
3	그랩 부트 레시피	부팅 시 설정할 그랩 레시피 설정
4	적용 버튼	설정한 부트 레시피 데이터를 적용

참고

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.3 Joint Offset

기능 설명



	J1 (°)	J2 (°)	J3 (°)	J4 (°)
FINGER 1	0	0	-90	0
	J5 (°)	J6 (°)	J7 (°)	J8 (°)
FINGER 2	0	0	-90	0
	J9 (°)	J10 (°)	J11 (°)	J12 (°)
FINGER 3	0	0	-90	0

관절 오프셋 설정은 관절의 물리적 위치가 어긋났을 경우, 소프트웨어적으로 관절 제로 포즈를 잡기 위해 사용됩니다. 그리퍼를 제로 포즈로 이동했을 때 특정 관절이 제로 포즈와 다르다면, 수동으로 제로 포즈를 만든 후 해당하는 관절 위치의 변화값을 관절 오프셋에 설정합니다.

항목

번호	명칭	설명
1	Joint Offset J1-Jn	n 번 관절 오프셋 설정 (단위: degree)
2	적용 버튼	설정된 Joint Offset, Joint Inpose, Moving Inpose 일괄 적용

참고

[10.1.2 SetGripperOption](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.4 Joint Inpose

기능 설명



Joint Inpose ⓘ

	J1 (°)	J2 (°)	J3 (°)	J4 (°)
1 FINGER 1	1	1	1	1
FINGER 2	1	1	1	1
FINGER 3	1	1	1	1

Joint Inpose는 각 관절별로 목표 위치 값에 도착을 확인하기 위한 범위 설정 기능입니다. 만약 J1에 3도가 설정된 상태로, 현재 관절 위치 값이 30도일 때 목표 관절 위치 값을 60도로 설정한 위치제어를 수행하게 되면, 현재 관절 위치 값이 57도가 되면 목표 위치에 도착했다고 판단합니다. 이 기능은 연속으로 위치이동이 필요할 때, 목표 관절 위치에 도달하기전에 다음 동작을 수행할 때 적합합니다. '8.1.3 Joint Offset'의 적용 버튼을 눌러 적용합니다.

항목

번호	명칭	설명
1	J1-Jn Inpose	n번 관절의 도착 확인 범위 설정 (단위: degree)

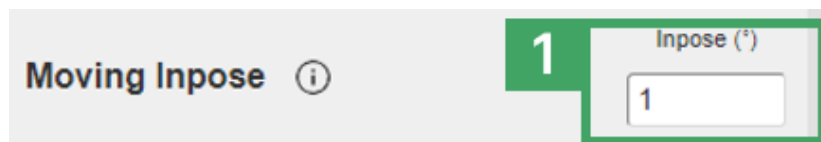
참고

10.1.2 SetGripperOption

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.5 Moving Inpose

기능 설명



Moving Inpose ⓘ

1 Inpose (°)

1

Moving Inpose는 관절의 움직임 확인을 위한 범위를 설정하는 기능입니다. 만약 Inpose 값이 1도 일 경우, 관절의 변화량이 1도 이상일 경우 관절이 움직였다고 판단됩니다.

'8.1.3 Joint Offset'의 적용 버튼을 눌러 적용합니다.

항목

번호	명칭	설명
1	Inpose	관절의 움직임 확인 범위 설정 (단위: degree)

참고

[10.1.2 SetGripperOption](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.6 Low Pass Filter



저역 통과 필터를 설정합니다.

항목

번호	명칭	설명
1	ON / OFF	필터 사용 유무 설정
2	Alpha	필터 게인값 설정

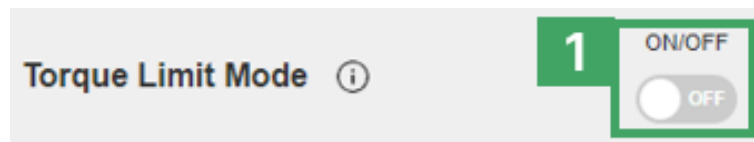
참고

[10.3.1 SetLowPassFilterAlpha](#) 참조

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.7 Torque Limit Mode

기능 설명



Torque Limit Mode는 그리퍼에 토크 제한을 해제하는 기능입니다. 이는 그리퍼에서 제공하는 정격 토크가 아닌 최대 토크를 사용하기 위함입니다. M모델에서만 지원되는 기능입니다.

항목

번호	명칭	설명
1	ON/OFF	토크 제한 모드 사용 유무 설정

참고

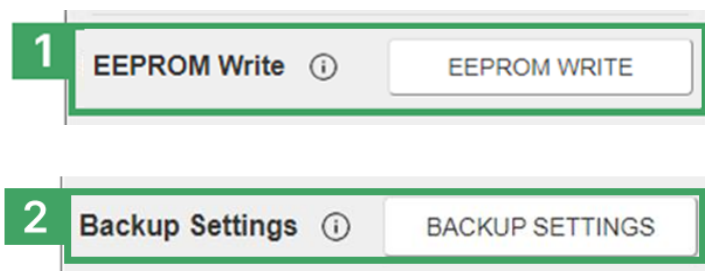
주의!! 사용자 모드에서 'Torque limit mode' off 사용 또는 개발자 모드에서 stall torque 로 사용 시, 급격한 제품 온도 상승 및 제품 손상 가능성이 있으며, 고장 시 A/S 가 어려운 점 주의바랍니다.

10.3.5 SetTorqueLimitMode 참조

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.8 Backup Settings / EEPROM Write

기능 설명



Backup Settings/EEPROM Write는 현재 적용된 설정 데이터를 PC/그리퍼에 저장하는 기능입니다. 사용자 모드에서는 통신이 연결될 때 자동으로 백업이 수행됩니다. EEPROM에 데이터를 저장하는 동안 그리퍼 전원을 끄거나 통신을 끊으면 데이터 손상이 발생할 수 있으므로 주의가 필요합니다.

DGManager 초기 실행 시 사용 방법에 대한 가이드 화면이 표시되며, 이후에는 4.2.1 화면 구성의 13. 프로그램 정보 표시 창에서 확인 가능합니다.

항목

번호	명칭	설명
1	EEPROM WRITE	사용자 모드에서 현재 설정된 값을 그리퍼에 저장
2	BACKUP SETTINGS	개발자 모드에서 현재 설정된 데이터를 PC 로 백업

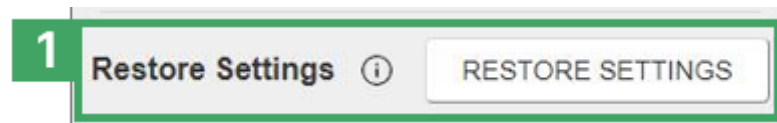
참고

10.3.8 BackupRecipeData 참조

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.1.9 Restore Settings

기능 설명



Restore Settings는 외부 경로의 백업 파일을 불러와 그리퍼에 데이터 복원하는 기능입니다. 관련 파일은 사용자모드 접속 시 혹은 개발자 모드에서 '8.1.8 Backup Settings / EEPROM Write'의 Backup Settings 시에 저장됩니다.

저장되는 경로는 다음과 같습니다:

DGManager-win32-x64(프로그램경로)\resources\config\{모델명}\{제어 모드}

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

DGManager 초기 실행 시 사용 방법에 대한 가이드 화면이 표시되며, 이후에는 4.2.1 화면 구성의 13. 프로그램 정보 표시 창에서 확인 가능합니다.

항목

번호	명칭	설명
1	RESTORE SETTINGS	백업 파일을 이용한 데이터 복원

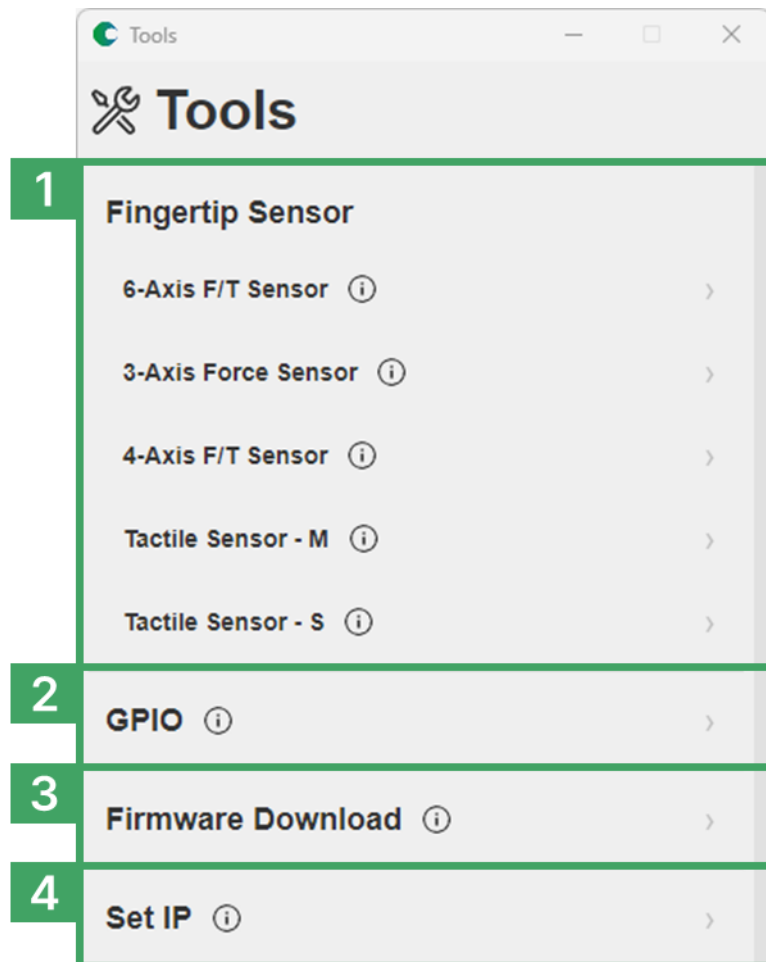
참고

10.3.9 RestoreRecipeData 참조

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2 도구

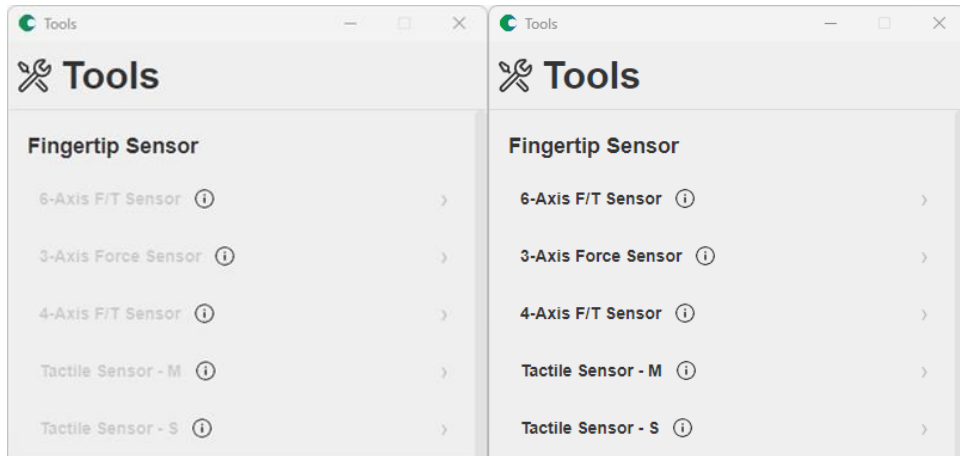
8.2.1 화면 구성



번호	명칭	설명
1	Fingertip Sensor	핑거팁 센서 데이터 모니터링
2	GPIO	그리퍼 GPIO 제어용 패널 열기
3	Firmware Download	그리퍼 펌웨어 파일을 선택하여 그리퍼에 펌웨어 다운로드
4	Set IP	그리퍼 IP 와 PORT 설정

8.2.2 Fingertip Sensor

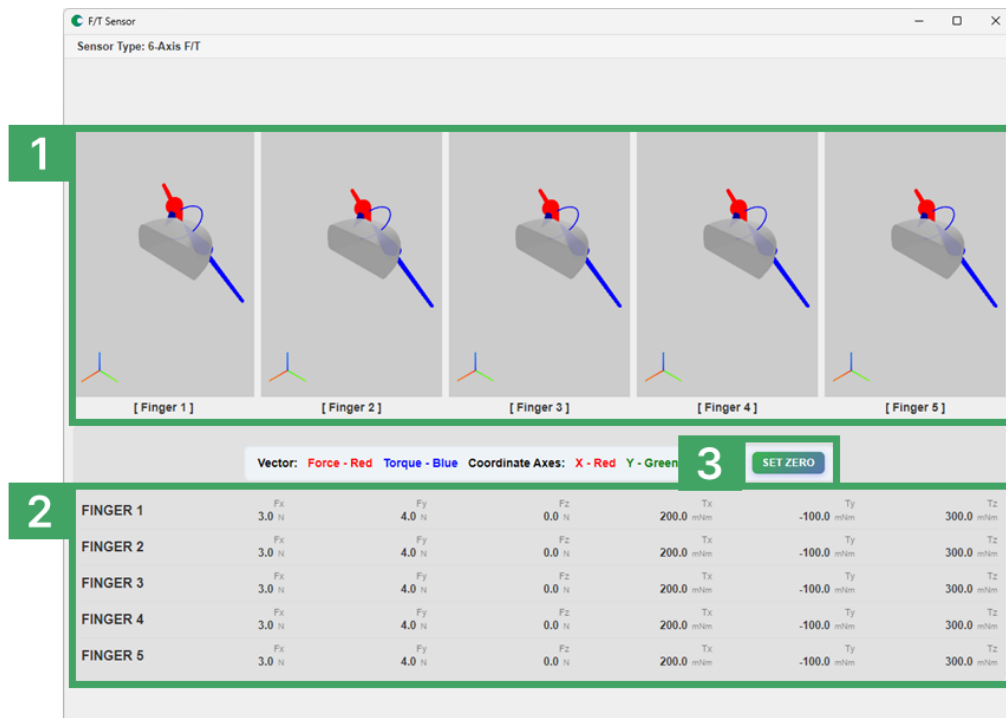
기능 설명



Delto Gripper는 센서가 내장된 핑거팁을 체결할 수 있습니다. 핑거팁은 M모델과 S모델에 따라 제공되는 버전이 다릅니다. 핑거팁 옵션이 적용되지 않은 펌웨어에서는 사용할 수 없으며, 펌웨어에 저장된 핑거팁 옵션에 해당하는 핑거팁 센서가 활성화됩니다.

8.2.2.1 6-Axis F/T Sensor

기능 설명



6-Axis Force/Torque 센서 데이터를 모니터링 할 수 있습니다. 힘과 토크값을 확인할 수 있습니다. M모델에서만 사용 가능 합니다.

항목

번호	명칭	설명
1	센서 3D 시각화	FT 센서 데이터를 3D 로 시각화
2	FT 센서 데이터	FT 센서 데이터를 확인 (단위: N, mN·m)
3	FT 센서 초기화	FT 센서 데이터를 0 으로 초기화

참조

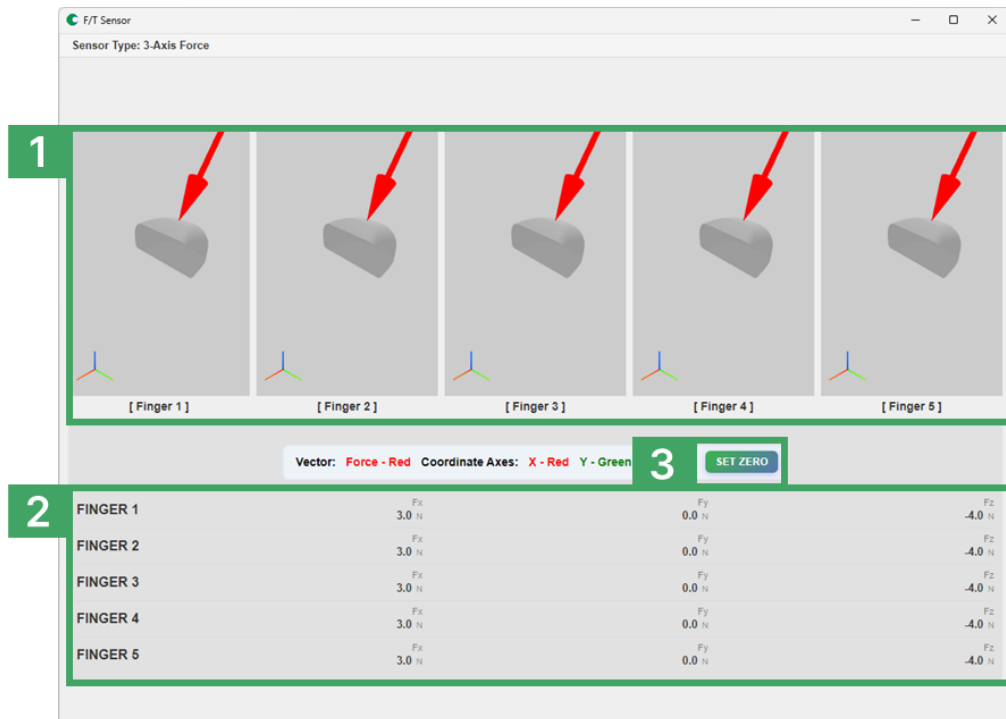
10.2.5 CallbackForOnReceivedFingertipSensorData

오류! 참조 원본을 찾을 수 없습니다. 오류! 참조 원본을 찾을 수 없습니다.

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2.2.2 3-Axis Force Sensor

기능 설명



3-Axis Force 센서 데이터를 모니터링 할 수 있습니다. 힘값을 확인할 수 있습니다. M모델에서만 사용 가능 합니다.

항목

번호	명칭	설명
1	센서 3D 시각화	FT 센서 데이터를 3D 로 시각화
2	FT 센서 데이터	FT 센서 데이터를 확인 (단위: N)
3	FT 센서 초기화	FT 센서 데이터를 0 으로 초기화

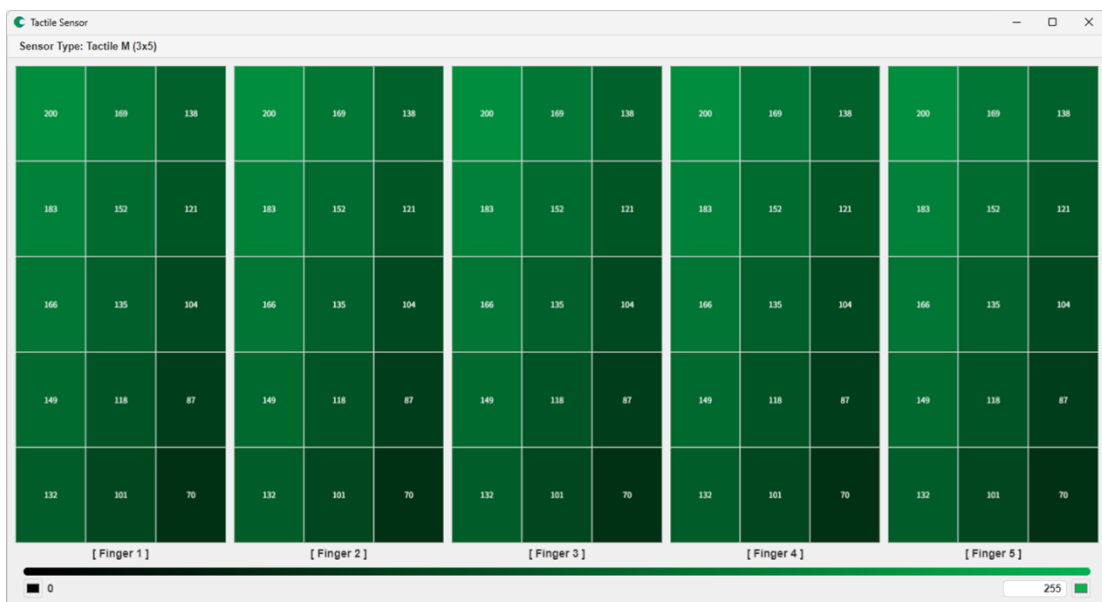
참조

[10.2.5 CallbackForOnReceivedFingertipSensorData](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2.2.3 Tactile Sensor – M/S

기능 설명



Tactile Sensor 데이터를 모니터링 할 수 있습니다. M모델은 3x5개 노드, S모델은 3x6개 노드로 구성되어 있으며, 각 노드마다 세기를 확인할 수 있습니다.

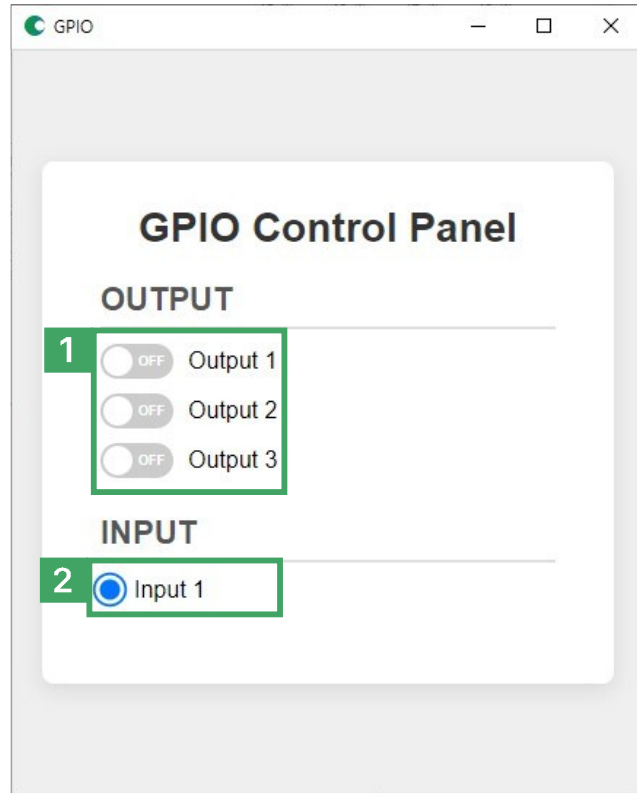
참조

[10.2.5 CallbackForOnReceivedFingertipSensorData](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2.3 GPIO

기능 설명



GPIO는 그리퍼 제어기에 내장되어 있는 Output ON/OFF 및 Input 상태를 확인할 수 있습니다. 이는 그리퍼 전원부에 연결된 8PIN 케이블과 연동 되어, 그리퍼 GPIO 제어기를 통해 외부 장치와 IO 신호를 주고받을 수 있습니다.

항목

번호	명칭	설명
1	Output	그리퍼 Output ON/OFF 수행
2	Input	그리퍼 Input 상태 표시

참조

[10.2.6 CallbackForOnReceivedGPIOData](#)

[10.3.3 SetGPIOOutput](#)

[10.3.4 SetGPIOOutputAll](#)

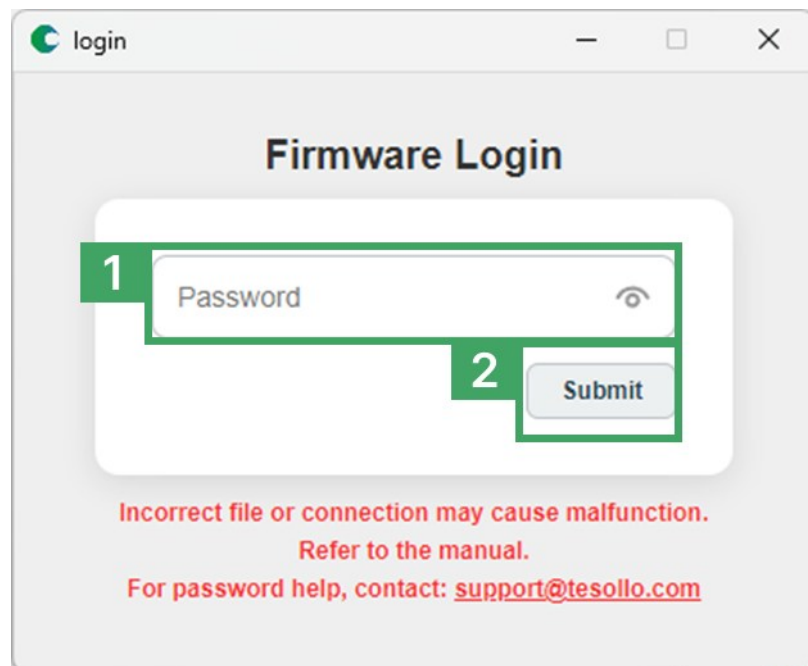
* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2.4 Firmware Download

Firmware Download는 그리퍼의 메인 제어기 펌웨어를 업데이트 하기 위한 기능입니다. 해당 기능을 사용하기 위해서는 비밀번호가 필요합니다. 비밀번호가 필요하시면 서포트 메일로 연락바랍니다. (support@tesollo.com)

8.2.4.1 로그인 화면

기능 설명



The image shows a 'login' window titled 'Firmware Login'. It contains a password input field labeled '1' and a 'Submit' button labeled '2'. Below the input field, there is a red warning message: 'Incorrect file or connection may cause malfunction. Refer to the manual. For password help, contact: support@tesollo.com'.

임의 펌웨어 업데이트 시도를 방지하기 위한 로그인 화면입니다. 패스워드 입력을 통해 펌웨어 다운로드 화면으로 전환할 수 있습니다.

항목

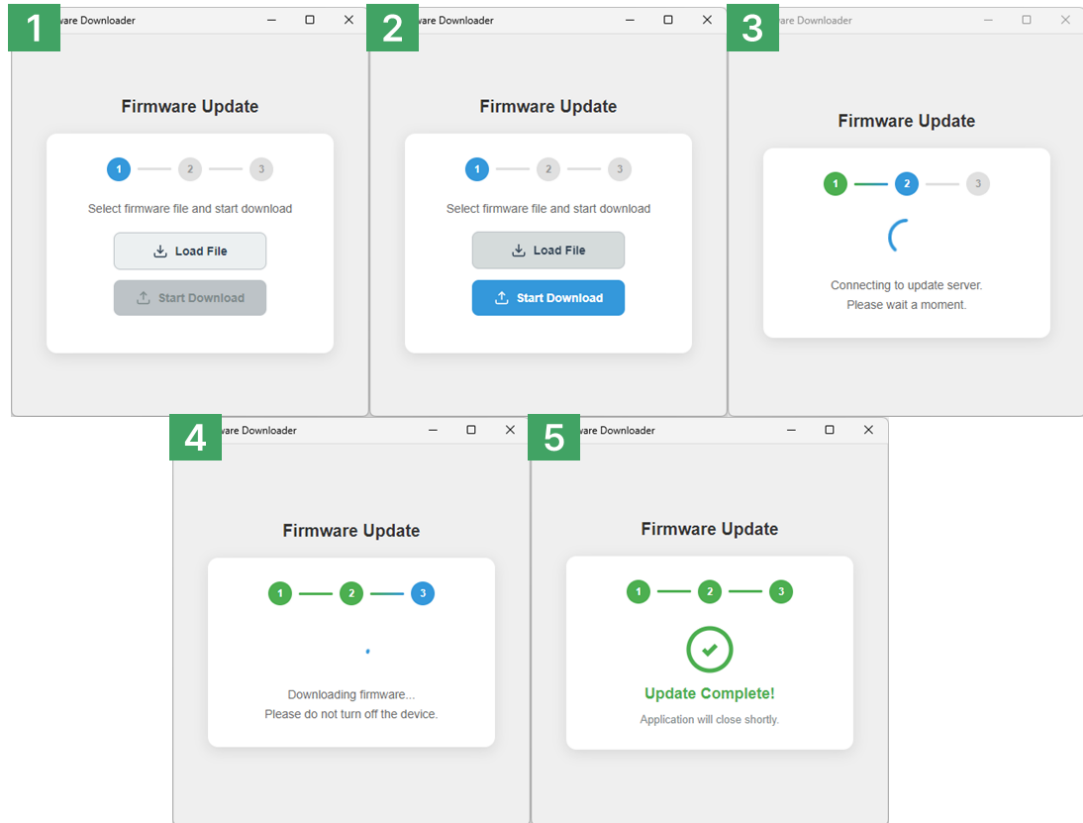
번호	명칭	설명
1	패스워드 입력	패스워드는 support@tesollo.com 으로 문의
2	Submit	로그인

참조

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2.4.2 펌웨어 다운로드 화면

기능 설명



1번 화면이 출력되면 먼저 'Load File'을 선택하여 다운로드할 펌웨어 파일을 불러오고, 'Start Download'를 선택합니다. 이후 3번 화면으로 넘어가면서 최종적으로 5번 화면이 출력되면 정상적으로 펌웨어의 다운로드가 완료됩니다. 이후 그리퍼는 자동으로 재부팅 되며, 재연결을 위해선 프로그램을 재실행 해야 합니다.

항목

번호	명칭	설명
1	LOAD FILE	저장소에서 펌웨어를 불러오기
2	START DOWNLOAD	펌웨어 다운로드를 수행. 부트모드 진입

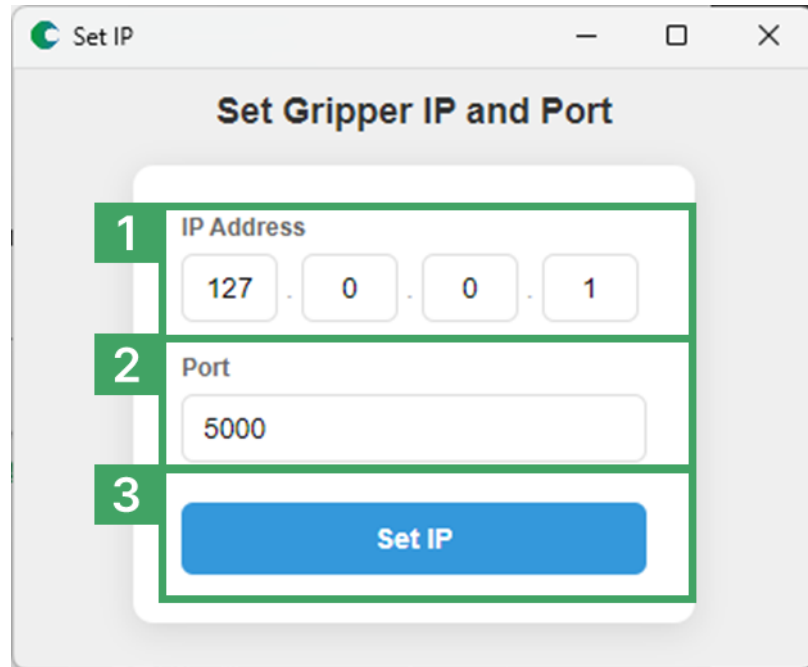
참조

[10.3.2 SetBootMode](#)

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.2.5 Set IP

기능 설명



그리퍼 IP와 Port를 설정합니다.

항목

번호	명칭	설명
1	IP	IP 입력
2	Port	Port 입력
3	Set IP	입력된 IP 와 Port 를 설정

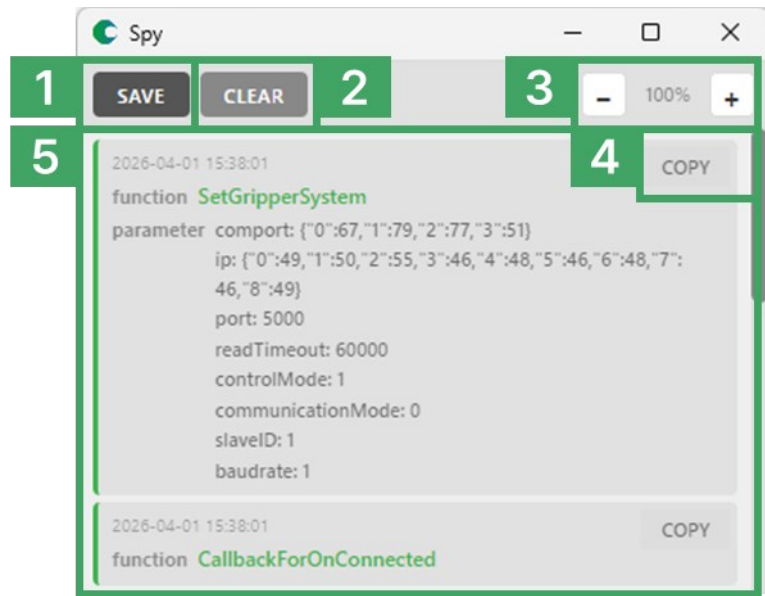
참조

10.1.7 SetI

* 에러 발생 시 '부록 2 에러 리스트' 참조

8.3 스파이

스파이는 DG Manager 를 사용하면서 발생하는 이벤트를 로그로 확인할 수 있는 화면입니다. 그리퍼 조작을 위해 DG Manager 에서 버튼을 누르면, 동작에 해당하는 명령어를 함수 형태로 표시해줍니다. 그리퍼를 조작하기 위한 방법을 쉽게 확인할 수 있는 기능입니다.

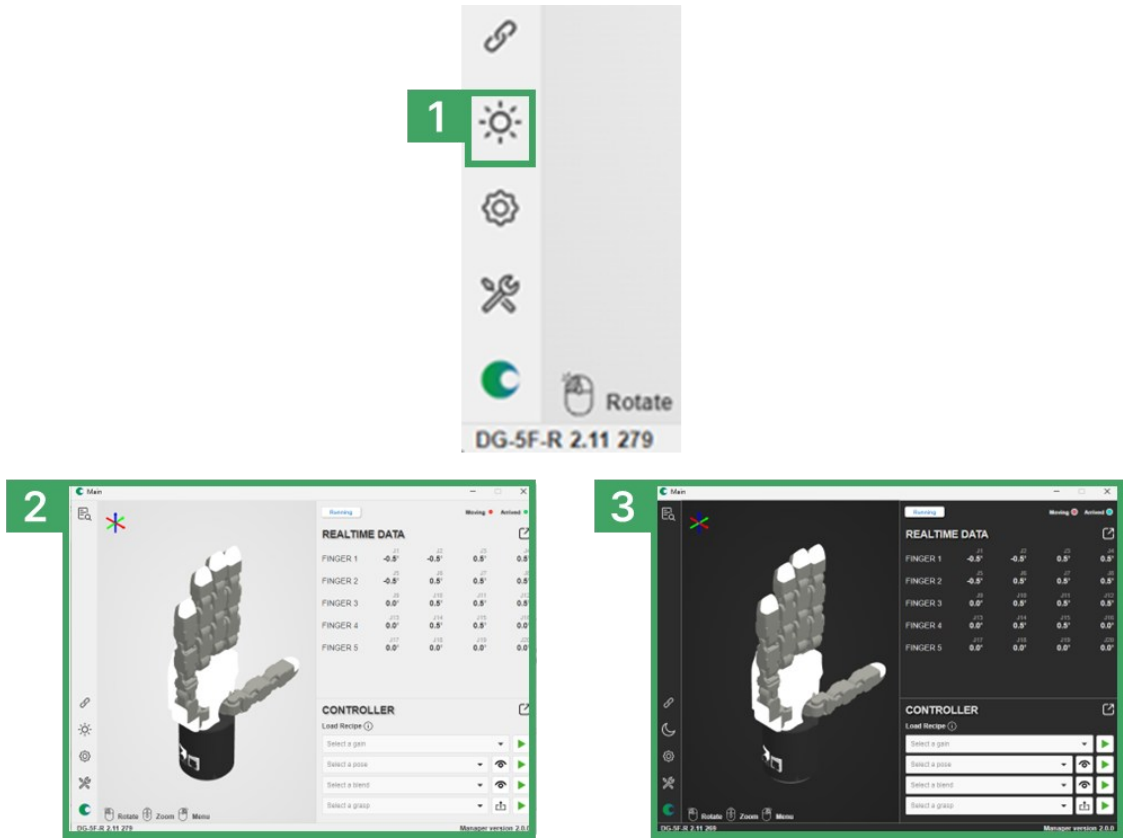


번호	명칭	설명
1	SAVE	스파이 로그를 txt 파일에 저장
2	CLEAR	현재 스파이 로그 전체 내용 삭제
3	Font Size	스파이 로그 글씨 크기 조정
4	COPY	로그 복사
5	Display Log	수행한 SDK 함수와 파라미터를 실시간으로 표시

참고

[제 10 장 시스템 함수](#)
[제 11 장 동작 설정 함수](#)
[제 12 장 레시피 함수](#)
[제 13 장 동작 함수](#)

8.4 테마



항목

번호	명칭	설명
1	테마 변경 기능	밝은 / 어두운 테마 변경
2	라이트 테마	밝은 배경에 어두운 텍스트를 사용하는 테마
3	다크 테마	어두운 배경에 밝은 텍스트를 사용하는 테마

제 9 장 SDK 자료 구조

9.1 Define

본 장은 SDK 에서 사용하기 위해 정의된 상수값을 설명합니다. 해당 상수값에서 벗어난 값을 입력할 경우 프로그램에 오류를 유발할 수 있습니다.

명칭	값	설명
MAX_JOINT_COUNT	20	최대 관절 개수
MAX_FINGER_COUNT	5	최대 손가락 개수
MAX_FINGER_JOINT_COUNT	4	손가락당 최대 관절 개수
MAX_BASE_JOINT_COUNT	2	베이스 최대 관절 개수
CARTESIAN_COORDINATE_POSE_COUNT	6	카테시안 좌표계 데이터 개수 (X, Y, Z, RX, RY, RZ)
MAX_RECEIVED_DATA_SIZE	1024	최대 통신 데이터 사이즈
MAX_GRIPPER_IP_ADDRESS_SIZE	32	그리퍼 IP 사이즈
MAX_GRIPPER_IP_BYTE_LENGTH	4	그리퍼 IP 주소 영역 개수
MAX_COMPORT_NAME_SIZE	32	컴포트 이름 사이즈
MAX_BLEND_COUNT	10	블렌드 동작 저장 개수
MAX_BLEND_ADD_POSE_COUNT	50	한 개 블렌드 동작 저장 레시피 개수
MAX_RECIPE_POSE_COUNT	100	레시피 포즈 저장 개수
MAX_RECIPE_GAIN_COUNT	20	레시피 게인 저장 개수
MAX_RECIPE_GRASP_COUNT	20	레시피 그랩 저장 개수
MAX_GRIPPER_GPIO_SIZE	4	GPIO 개수(Output 3 개, Input 1 개)
MAX_RECEIVED_DATA_TYPE_COUNT	7	수신 받을 수 있는 데이터 타입 개수

9.2 열거형

9.2.1 enum DG_RESULT

값	명칭	설명
0	DG_RESULT_NONE	정상
그 외	ERROR	부록 2 에러 리스트 참조

9.2.2 enum DG_MODEL

그리퍼 모델 설정을 위한 열거형 상수

값	명칭	설명
0x0000	DG_MODEL_NONE	모델 미설정
0x1F02	DG_MODEL_DG_1F_M	M 모델 1-Finger
0x2F02	DG_MODEL_DG_2F_M	M 모델 2-Finger
0x3F01	DG_MODEL_DG_3F_B	B 모델 3-Finger
0x3F02	DG_MODEL_DG_3F_M	M 모델 3-Finger
0x4F02	DG_MODEL_DG_4F_M	M 모델 4-Finger
0x5F12	DG_MODEL_DG_5F_LEFT	M 모델 5-Finger 왼손
0x5F22	DG_MODEL_DG_5F_RIGHT	M 모델 5-Finger 오른손
0x5F14	DG_MODEL_DG_5F_S_LEFT	S 모델 20 자유도 5-Finger 왼손
0x5F24	DG_MODEL_DG_5F_S_RIGHT	S 모델 20 자유도 5-Finger 오른손
0x5F34	DG_MODEL_DG_5F_S15_LEFT	S 모델 15 자유도 5-Finger 왼손
0x5F44	DG_MODEL_DG_5F_S15_RIGHT	S 모델 15 자유도 5-Finger 오른손

9.2.3 enum BLEND_MOTION_STATUS

블렌드 동작에 상태를 확인하기 위한 열거형 상수

값	명칭	설명
0	BLEND_MOTION_STATUS_STOP	블렌드 동작 정지 상태
1	BLEND_MOTION_STATUS_RUN	블렌드 동작 상태
2	BLEND_MOTION_STATUS_COMPLETE	블렌드 동작 완료 상태

9.2.4 enum COMMUNICATION_MODE

그리퍼 통신 모드를 설정하기 위한 열거형 상수

값	명칭	설명
0	COMMUNICATION_MODE_ETHERNET	이더넷 통신 연결
1	COMMUNICATION_MODE_RS485	시리얼 통신 연결

9.2.5 enum CONTROL_MODE

그리퍼 제어 모드를 설정하기 위한 열거형 상수

값	명칭	설명
0	CONTROL_MODE_OPERATOR	사용자 모드
1	CONTROL_MODE_DEVELOPER	개발자 모드

9.2.6 enum GAIN_MODE

그리퍼 PID 제어모드를 설정하기 위한 열거형 상수

값	명칭	설명
0	GAIN_MODE_PD	PD 제어 모드
1	GAIN_MODE_PID	PID 제어 모드

9.2.7 enum DEVELOPER_MODE_COMMAND

그리퍼 사용자 모드 미지원 명령어에 대한 열거형 상수

값	명칭	설명
1	DEVELOPER_MODE_COMMAND_GET_DATA	설정된 데이터 수신
2	DEVELOPER_MODE_COMMAND_SET_RECEIVED_DATA	수신할 데이터 설정
5	DEVELOPER_MODE_COMMAND_SET_DUTY	관절 듀티 입력
6	DEVELOPER_MODE_COMMAND_SET_GPIO	GPIO(Output) 설정
7	DEVELOPER_MODE_COMMAND_SET_IP_ADDRESS	그리퍼 IP 및 PORT 설정
8	DEVELOPER_MODE_COMMAND_GET_ID_AND_VERSION	그리퍼 ID 및 펌웨어 버전 수신
9	DEVELOPER_MODE_COMMAND_GET_JOINT_ID	그리퍼 모듈 이상 ID 수신
10	DEVELOPER_MODE_COMMAND_SET_BOOT_MODE	부트 모드 진입

11	DEVELOPER_MODE_COMMAND_SET_FT_OFFSET_ZERO	핑거팁 센서 오프셋 설정
----	---	------------------

9.2.8 enum DEVELOPER_MODE_RECEIVED_DATA_TYPE

그리퍼 개발자 모드에서 수신 받을 수 있는 데이터 유형을 선택하기 위한 열거형 상수

값	명칭	설명
1	DEVELOPER_MODE_RECEIVED_DATA_TYPE_JOINT	관절 위치 정보
2	DEVELOPER_MODE_RECEIVED_DATA_TYPE_CURRENT	관절 전류 정보
3	DEVELOPER_MODE_RECEIVED_DATA_TYPE_TEMPERATURE	관절 온도 정보
4	DEVELOPER_MODE_RECEIVED_DATA_TYPE_VELOCITY	관절 속도 정보
5	DEVELOPER_MODE_RECEIVED_DATA_TYPE_FINGER_FT_SENSOR	핑거팁 FT 센서 정보
6	DEVELOPER_MODE_RECEIVED_DATA_TYPE_GPIO	GPIO 상태
7	DEVELOPER_MODE_RECEIVED_DATA_TYPE_MODULE_ERROR_CODE	모듈 에러 코드
8	DEVELOPER_MODE_RECEIVED_DATA_TYPE_CONTROL_PERIOD	그리퍼 제어 주기

9.2.9 enum DG_GRASP_MODE

그리퍼 그랩 모드 설정을 위한 열거형 상수

DG_GRASP_MODE_2F는 DG-2F-M 모델만 사용 가능

DG_GRASP_MODE_3F는 DG-3F-B, DG-3F-M 모델만 사용 가능

DG_GRASP_MODE_4F는 DG-4F-M 모델만 사용 가능

DG_GRASP_MODE_5F는 DG-5F-R, DG-5F-L 모델만 사용 가능

값	명칭	설명
0	DG_GRASP_MODE_NONE	그랩 모드 미설정
1	DG_GRASP_MODE_3F_3FINGER	세 손가락 중점 그랩 모드

2	DG_GRASP_MODE_3F_2FINGER_1_AND_2	1 번, 2 번 손가락 중점 그랩 모드
3	DG_GRASP_MODE_3F_2FINGER_1_AND_3	1 번, 3 번 손가락 중점 그랩 모드
4	DG_GRASP_MODE_3F_2FINGER_2_AND_3	2 번, 3 번 손가락 중점 그랩 모드
5	DG_GRASP_MODE_3F_3FINGER_PARALLEL	세 손가락 평행 그랩 모드
6	DG_GRASP_MODE_3F_3FINGER_ENVELOP	세 손가락 감싸쥐기 그랩 모드
21	DG_GRASP_MODE_5F_5FINGER	다섯 손가락 중점 그랩 모드
22	DG_GRASP_MODE_5F_3FINGER	세 손가락 중점 그랩 모드
23	DG_GRASP_MODE_5F_3FINGER_PARALLEL	세 손가락 평행 그랩 모드
24	DG_GRASP_MODE_5F_2FINGER_1_AND_2	1 번, 2 번 손가락 중점 그랩 모드
25	DG_GRASP_MODE_5F_2FINGER_1_AND_3	1 번, 3 번 손가락 중점 그랩 모드
26	DG_GRASP_MODE_5F_2FINGER_1_AND_4	1 번, 4 번 손가락 중점 그랩 모드
27	DG_GRASP_MODE_5F_2FINGER_1_AND_5	1 번, 5 번 손가락 중점 그랩 모드
28	DG_GRASP_MODE_5F_5FINGER_PARALLEL	다섯 손가락 평행 그랩 모드
29	DG_GRASP_MODE_5F_5FINGER_ENVELOP	다섯 손가락 감싸쥐기 그랩 모드
31	DG_GRASP_MODE_4F_4FINGER	네 손가락 중점 그랩 모드
32	DG_GRASP_MODE_4F_4FINGER_PARALLEL	네 손가락 평행 그랩 모드
33	DG_GRASP_MODE_4F_4FINGER_ENVELOP	네 손가락 감싸쥐기 그랩 모드
34	DG_GRASP_MODE_4F_4FINGER_RIGHT_PARALLEL	오른손 모양 평행 그랩 모드
35	DG_GRASP_MODE_4F_4FINGER_LEFT_PARALLEL	왼손 모양 평행 그랩 모드
36	DG_GRASP_MODE_4F_2FINGER_1_AND_2	1 번, 2 번 손가락 중점 그랩 모드

37	DG_GRASP_MODE_4F_2FINGER_1_AND_4	1 번, 4 번 손가락 중점 그랩 모드
38	DG_GRASP_MODE_4F_2FINGER_3_AND_4	3 번, 4 번 손가락 중점 그랩 모드
51	DG_GRASP_MODE_2F_2FINGER	1 번, 2 번 손가락 중점 그랩 모드
52	DG_GRASP_MODE_2F_2FINGER_ENVELOP	두 손가락 감싸쥐기 그랩 모드

9.2.10 enum DG_GRASP_OPTION

그리퍼 그랩 옵션 설정을 위한 열거형 상수 (현재 미지원 기능)

값	명칭	설명
0	DG_GRASP_OPTION_NONE	그랩 옵션 미설정
1	DG_GRASP_OPTION_PARALLEL	추후 추가 예정
2	DG_GRASP_OPTION_FIX_TILT	추후 추가 예정
3	DG_GRASP_OPTION_SET_TILT	추후 추가 예정

9.2.11 enum DG_SENSOR_TYPE

핑거팁 센서 유형 구분을 위한 열거형 상수

값	명칭	설명
1	DG_SENSOR_TYPE_FT_SENSOR_6_AXIS	6 축 힘/토크 센서
2	DG_SENSOR_TYPE_FT_SENSOR_3_AXIS	3 축 힘 센서
3	DG_SENSOR_TYPE_TACTILE_M	M 시리즈 택타일 센서
5	DG_SENSOR_TYPE_TACTILE_S	S 시리즈 택타일 센서

9.3 구조체

9.3.1 struct ReceivedGripperData

그리퍼에서 제공되는 데이터를 확인하기 위한 구조체

자료형	명칭	크기 (byte)	설명
float	joint	80	그리퍼 관절의 위치 정보
int	current	80	그리퍼 관절의 전류 정보
int	velocity	80	그리퍼 관절의 속도 정보 (DG-3F-B 모델은 지원 안함)
float	temperature	80	그리퍼 관절의 온도 정보 (DG-3F-B 모델은 지원 안함)
int	moving	4	그리퍼 모든 관절의 움직임 상태 정보
int	targetArrived	4	그리퍼 모든 관절의 목표 위치 도착 상태 정보
int	blendMoveState	4	블렌드 조인트 기능 사용시 상태 정보
int	currentBlendIndex	4	현재 실행중인 블렌드 인덱스 정보
int	productID	4	그리퍼 ID
int	firmwareVersion	4	그리퍼 펌웨어 버전
int	moduleErrorCode	4	모듈 에러 코드
int	controlPeriod	4	그리퍼 내부 제어주기

참조 자료

자료형 x MAX_JOINT_COUNT = 80 byte. joint, current, velocity, temperature

9.2.3 enum BLEND_MOTION_STATUS 참조

9.3.2 struct RecipeBlendData

블렌드 동작을 사용을 위한 레시피 설정 구조체

자료형	명칭	크기 (byte)	설명
int	recipePoseNumber	4	레시피 포즈 넘버
int	recipeGainNumber	4	레시피 게인 넘버
int	blendWaitTime	4	동작 완료 후 대기 시간 설정
int	number	4	블렌드 동작 고유 번호

9.3.3 struct GripperSystemSetting

그리퍼 통신 연결 설정을 위한 구조체

자료형	명칭	크기 (byte)	설명
char	comport	32	RS485 연결 시 콤포트 설정
char	ip	32	TCP/IP 연결 시 IP 설정
int	port	4	TCP/IP 연결 시 PORT 설정
int	readTimeout	4	통신 타임 아웃 시간 설정
int	controlMode	4	그리퍼 제어모드 설정
int	communicationMode	4	그리퍼 통신모드 설정
int	slaveID	4	Modbus 연결 시 Slave ID 설정
int	baudrate	4	RS485 연결 시 baudrate 설정

9.3.4 struct GripperSetting

그리퍼 구동을 위한 기본 설정 구조체

자료형	명칭	크기 (byte)	설명
float	jointOffset	80	그리퍼 관절의 위치 오프셋 설정
float	jointInpose	80	그리퍼 관절의 도착 범위 확인을 위한 각도 설정
float	tcpInpose	20	손 끝 위치 도착 범위 확인을 위한 거리 설정
float	orientationInpose	20	손 끝 방향 도착 범위 확인을 위한 각도 설정
int	receivedDataType	24	수신 받을 데이터 타입 설정
float	movingInpose	4	그리퍼 관절의 움직임 판단을 위한 각도 설정
int	jointCount	4	현재 사용 중인 그리퍼 모델의 관절 개수
int	fingerCount	4	현재 사용 중인 그리퍼 모델의 손가락 개수
DG_MODEL	model	4	현재 사용 중인 그리퍼 모델 설정
int8_t	dutyByteLength	1	현재 모델에 수신 데이터 사이즈

참조 자료

자료형 x MAX_JOINT_COUNT = 80 byte. jointOffset, jointInpose

자료형 x MAX_FINGER_COUNT = 20 byte. tcpInpose, orientationInpose

자료형 x MAX_RECEIVED_DATA_TYPE_COUNT = 24 byte. receivedDataType

[9.2.2 enum DG_MODEL 참조](#)

9.3.5 struct RecipePoseData

레시피 포즈 데이터 설정 구조체

자료형	명칭	크기 (byte)	설명
float	targetJoint	80	목표 관절 위치값 설정
int	jointMotionTime	80	목표 위치까지 도달하는데 걸리는 시간 설정
int	number	4	레시피 포즈 고유 번호
int	mode	4	위치 제어 모드 설정(미사용)

참조 자료

자료형 x MAX_JOINT_COUNT = 80 byte. targetJoint, jointMotionTime

9.3.6 struct RecipeGainData

레시피 게인 데이터 설정 구조체

자료형	명칭	크기 (byte)	설명
float	gainP	80	P 게인 설정
float	gainD	80	D 게인 설정
float	gainI	80	I 게인 설정
float	iLimit	80	오차 적분 최댓값 설정
int	controlPIDMode	4	PID 제어 모드 설정
int	number	4	레시피 게인 고유 번호
int	mode	4	위치 제어 모드 설정(미사용)

참조 자료

자료형 x MAX_JOINT_COUNT = 80 byte. gainP, gainD, gainI, iLimit

[9.2.6 enum GAIN_MODE 참조](#)

9.3.7 struct RecipeGraspData

레시피 그랩 데이터 설정 구조체

자료형	명칭	크기 (byte)	설명
float	graspForce	4	그랩 세기 설정
int	positionMode	80	관절을 그랩 모드가 아닌 위치 제어로 고정
int	graspMode	4	그랩 모드 설정
int	graspOption	4	그랩 옵션 설정 (추후 추가 예정)
int	smoothGrasping	4	부드러운 그랩 모드 (추후 추가 예정)
int	number	4	레시피 그랩 고유 번호

참조 자료

자료형 x MAX_JOINT_COUNT = 80 byte. positionMode
9.2.9 enum DG_GRASP_MODE

9.3.8 struct ReceivedFingertipSensorData

그리퍼 핑거팁 센서 데이터를 확인하기 위한 구조체

자료형	명칭	크기 (byte)	설명
int	sensorType	4	연결된 센서 유형
int	attachedFinger	20	센서가 부착된 손가락 위치 배열
float	forceTorque	120	그리퍼 핑거팁 센서 Force / Torque 정보
uint16_t	tactile	180	택타일 센서 정보

참조 자료

자료형 x MAX_FINGER_COUNT = 20 byte. attachedFinger
자료형 x MAX_FINGER_COUNT x 6(센서 데이터 개수) = 120 byte. forceTorque
자료형 x MAX_FINGER_COUNT x 18(센서 데이터 개수) = 180 byte. tactile

9.3.9 struct ReceivedGPIOData

그리퍼 GPIO 데이터를 확인하기 위한 구조체

자료형	명칭	크기 (byte)	설명
int	GPIO	16	그리퍼 GPIO 데이터 (Output 1,2,3 / Input 1)

참조 자료

자료형 x MAX_GRIPPER_GPIO_SIZE = 16 byte. GPIO

제 10 장 시스템 함수

10.1 시스템 설정 함수

10.1.1 SetGripperSystem

기능 설명

그리퍼와 통신을 연결하기 위한 설정값을 입력하는 함수입니다. 해당 함수는 가장 우선적으로 수행되어야 하며, 해당 함수가 수행되지 않을 경우 다른 모든 함수를 수행할 수 없습니다.

파라미터

자료형	변수	설명
struct	GripperSystemSetting	9.3.3 struct GripperSystemSetting 참조

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
// 사용자 모드 Modbus RTU 연결 시
GripperSystemSetting setting;
setting.communicationMode = COMMUNICATION_MODE_RS485;
setting.controlMode = CONTROL_MODE_OPERATOR;
char port[MAX_COMPORT_NAME_SIZE] = "COM1"; // 연결 가능한 콤포트 입력
memcpy(setting.comport, port, MAX_COMPORT_NAME_SIZE);
setting.baudrate = 115200; // 그리퍼에 설정된 Baudrate 입력
setting.slaveID = 1; // 그리퍼에 설정된 Slave ID 입력
DG_RESULT result = SetGripperSystem(setting);

// 사용자 모드 Modbus TCP 연결 시
GripperSystemSetting setting;
setting.communicationMode = COMMUNICATION_MODE_ETHERNET;
setting.controlMode = CONTROL_MODE_OPERATOR;
char ip[MAX_GRIPPER_IP_ADDRESS_SIZE] = "169.254.186.72"; // 그리퍼 IP 주소 입력
memcpy(setting.ip, ip, MAX_GRIPPER_IP_ADDRESS_SIZE);
setting.port = 502; // 그리퍼에 설정된 Port 입력
setting.slaveID = 1; // 그리퍼에 설정된 Slave ID 입력
```

```

setting.readTimeout= 60000;    // 그리퍼의 통신 타임 아웃 시간 입력
DG_RESULT result = SetGripperSystem(setting);

// 개발자 모드 TCP/IP 연결 시
GripperSystemSetting setting;
setting.communicationMode = COMMUNICATION_MODE_ETHERNET;
setting.controlMode = CONTROL_MODE_DEVELOPER;
char ip[MAX_GRIPPER_IP_ADDRESS_SIZE] = "169.254.186.72";    // // 그리퍼 IP 주소 입력
memcpy(setting.ip, ip, MAX_GRIPPER_IP_ADDRESS_SIZE);
setting.port = 502;    // 그리퍼에 설정된 Port 입력
setting.slaveID = 1;    // 그리퍼에 설정된 Slave ID 입력
DG_RESULT result = SetGripperSystem(setting);

```

10.1.2 SetGripperOption

기능 설명

그리퍼 동작을 위한 기본값을 설정하는 옵션입니다. 해당 옵션을 수행하지 않을 경우 그리퍼를 제어할 수 없습니다.

파라미터

자료형	변수	설명
GripperSetting	setting	9.3.4 struct GripperSetting 참조

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```

// DG-3F-M 설정 시
float offset[MAX_JOINT_COUNT] = { 0 };
float inpose[MAX_JOINT_COUNT] = { 0 };
GripperSetting gs;
offset[0] = 2; // 1번 관절 오프셋 2도 설정
inpose[2] = 3; // 3번 관절 인포즈 3도 설정
gs.model = DG_MODEL_DG_3F_M; // 그리퍼 모델 설정
memcpy(gs.jointOffset, offset, sizeof(offset));
memcpy(gs.jointInpose, inpose, sizeof(inpose));
gs.movingInpose = 0.3f;

```

```
DG_RESULT result = SetGripperOption(gs);
```

10.1.3 ConnectToGripper

기능 설명

그리퍼에 통신 접속을 시도합니다. InitializedCallback 함수에 포함된 콜백 함수들은 '10.2 콜백'을 통해 세부 명령을 확인하실 수 있습니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

10.1.4 DisconnectToGripper

기능 설명

그리퍼에 통신 해제를 시도합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
DG_RESULT result = DisconnectToGripper();
```

10.1.5 SystemStart

기능 설명

시스템을 시작합니다. 해당 기능을 수행해야만 그리퍼의 데이터를 수신 받을 수 있으며, 그리퍼를 제어할 수 있습니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.1.6 SystemStop

기능 설명

시스템을 정지합니다. 해당 기능이 수행되면 그리퍼의 데이터가 수신되지 않으며, 제어도 수행할 수 없습니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.1.7 SetIp

기능 설명

그리퍼 IP와 Port를 설정합니다.

파라미터

자료형	변수	설명
int*	ip	변경할 ip 입력
int	port	변경할 port 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.1.8 SetJointEncoderZero

기능 설명

그리퍼 현재 위치를 영점으로 설정합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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


10.2 콜백

10.2.1 CallbackForOnConnected

기능 설명

그리퍼와의 통신 연결에 대한 이벤트를 수신합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.2.2 CallbackForOnDisconnected

기능 설명

그리퍼와의 통신이 끊어짐에 대한 이벤트를 수신합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.2.3 CallbackForOnReceivedGripperData

기능 설명

그리퍼로부터 데이터를 수신합니다.

파라미터

자료형	변수	설명
struct	ReceivedGripperData	9.3.1 struct ReceivedGripperData 참조

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.2.4 CallbackForOnCommunicationPeriod

기능 설명

그리퍼와 통신 주기를 수신합니다. 해당 값은 1초 동안 그리퍼와 PC간 통신을 주고받은 횟수를 나타냅니다. 그리퍼와 연결된 컨트롤러 사양에 따라 해당 값은 낮아질 수 있으며, 해당 값이 낮아질 경우 원활한 제어가 수행되지 않을 수 있습니다.

파라미터

자료형	변수	설명
int	period	통신 주기

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.2.5 CallbackForOnReceivedFingertipSensorData

기능 설명

그리퍼 핑거팁 센서 데이터를 수신합니다.

파라미터

자료형	변수	설명
struct	ReceivedFingertipSensorData	핑거팁 센서 데이터

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.2.6 CallbackForOnReceivedGPIOData

기능 설명

그리퍼 GPIO 데이터를 수신합니다.

파라미터

자료형	변수	설명
struct	ReceivedGPIOData	GPIO 데이터

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

10.3 기타 시스템 함수

10.3.1 SetLowPassFilterAlpha

기능 설명

저역 통과 필터 게인을 설정합니다. 설정 게인값이 낮을수록 부드러운 움직임이 나타납니다.

파라미터

자료형	변수	설명
int	isUsed	필터 사용 설정
float	alpha	필터 게인값 설정

리턴

값	설명
DG_RESULT_NONE	정상
DG_RESULT_SYSTEM_SETTING_NOT_PERFORMED	시스템 세팅 미수행
DG_RESULT_DATA_IS_NOT_BOOLEAN	0 과 1 만 입력 가능
DG_RESULT_DATA_IS_NEGATIVE	음수 값 입력 불가

예제

```
C/C++
float alpha = 0.1; // 게인값 설정
int isUsed = 1; // 필터 On/Off
DG_RESULT result = SetLowPassFilterAlpha(isUsed, alpha);
```

10.3.2 SetBootMode

기능 설명

펌웨어 다운로드를 위한 부트 모드 진입 기능입니다. 부트 모드로 진입하기 위해서는 패스워드가 필요합니다. 패스워드는 support@tesollo.com 으로 문의해주세요.

파라미터

자료형	변수	설명
char*	password	부트 모드 진입을 위한 패스워드 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.3.3 SetGPIOOutput

기능 설명

그리퍼 GPIO 출력을 설정합니다.

파라미터

자료형	변수	설명
int	gpio	출력값 설정
int	outputNumber	출력 번호 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
    int gpio = 1; // output true 출력
    int outputNumber = 1; // 출력 번호 설정
    DG_RESULT result = SetGPIOOutput(gpio, outputNumber);
```

10.3.4 SetGPIOOutputAll

기능 설명

그리퍼 전체 GPIO 출력을 설정합니다.

파라미터

자료형	변수	설명
int*	gpio	출력값 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

10.3.5 SetTorqueLimitMode

기능 설명

그리퍼의 토크 제한 모드를 설정합니다.

파라미터

자료형	변수	설명
int	isOn	토크 제한 모드 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int isOn = 0; // 토크 제한 모드 해제
DG_RESULT result = SetTorqueLimitMode(isOn);
```

참고

주의!! 사용자 모드에서 'Torque limit mode' off 사용 또는 개발자 모드에서 stall torque 로 사용 시, 급격한 제품 온도 상승 및 제품 손상 가능성이 있으며, 고장 시 A/S 가 어려운 점 주의바랍니다.

10.3.6 SetBootRecipe

기능 설명

그리퍼 부팅 시 시작할 레시피를 설정합니다.

파라미터

자료형	변수	설명
int	recipePoseNumber	포즈 레시피 번호 설정
int	recipeGainNumber	포즈 게인 번호 설정
int	recipeGraspNumber	포즈 그랩 번호 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
DG_RESULT result = SetBootRecipe(1,1,1); // 1번 레시피 포즈, 1번 레시피 게인, 1번 레시피 그랩
번호를 부팅 시 시작할 레시피로 설정
```


10.3.7 EEPROMWrite

기능 설명

현재 설정된 값을 EEPROM에 저장합니다. 사용자 모드에서는 그리퍼의 EEPROM에 저장되며, 개발자 모드에서는 PC 저장소에 저장됩니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
DG_RESULT result = EEPROMWrite(); // 현재 설정된 값을 EEPROM에 저장
```

10.3.8 BackupRecipeData

기능 설명

그리퍼에 적용된 데이터를 백업하여 PC에 파일로 저장할 수 있는 기능입니다.

파라미터

자료형	변수	설명
char*	path	백업 저장 경로

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

10.3.9 RestoreRecipeData

기능 설명

이 기능은 사용자 모드(Operator Mode)에서만 지원됩니다. 지정된 경로의 백업 파일에서 데이터를 읽어들이어 연결된 그리퍼로 데이터를 전송합니다.

파라미터

자료형	변수	설명
char*	path	복원 파일 경로

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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

10.4 데이터 조회

10.4.1 GetReceivedGripperData

기능 설명

그리퍼에서 수신되는 데이터를 가져옵니다.

파라미터

자료형	변수	설명
struct*	receivedGripperData	수신 데이터 구조체 변수

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

관련 함수

[10.2.3 CallbackForOnReceivedGripperData](#)

10.4.2 GetCommunicationPeriod

기능 설명

그리퍼와의 통신 주기를 가져옵니다.

파라미터

자료형	변수	설명
int*	period	통신 주기 수신용 변수

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int period;
DG_RESULT result = GetCommunicationPeriod(&period);
```

관련 함수

[10.2.4 CallbackForOnCommunicationPeriod](#)

10.4.3 GetReceivedFingertipSensorData

기능 설명

그리퍼 핑거 팁에 부착된 힘 토크 센서 데이터를 가져옵니다.

파라미터

자료형	변수	설명
struct*	receivedFingertipData	핑거팁 데이터 수신 구조체 변수

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

관련 함수

[10.2.5 CallbackForOnReceivedFingertipSensorData](#)

10.4.4 GetReceivedGPIOData

기능 설명

그리퍼 GPIO 데이터를 가져옵니다.

파라미터

자료형	변수	설명
struct*	receivedGPIOData	GPIO 수신 구조체 변수

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
ReceivedGPIOData receivedGPIOData;
DG_RESULT result = GetReceivedGPIOData(&receivedGPIOData);
```

관련 함수

[10.2.6 CallbackForOnReceivedGPIOData](#)

10.4.5 GetLibraryVersion

기능 설명

현재 사용중인 라이브러리 버전을 가져옵니다.

파라미터

자료형	변수	설명
int*	version	버전 정보 복사할 배열

예제

C/C++

```
int version[3] = {0,0,0}; // 라이브러리 버전 : major, minor, patch 순으로 저장
GetLibraryVersion(version);
std::cout << "Received Library Version : major : " << version[0]
          << ", minor : " << version[1]
          << ", patch : " << version[2]
          << std::endl;
```

제 11 장 동작 설정 함수

11.1 그랩 설정 함수

힘 제어 기반의 그랩 알고리즘을 설정합니다. 5 지, 3 지, 2 지, 평행, 감싸쥐기 등 그랩 모드를 제공하며, 그랩 옵션, 위치 제어 모드 등을 활용하여 다양한 방식으로 그랩 동작을 수행할 수 있습니다.

11.1.1 SetGraspData

기능 설명

그랩을 수행하기 위해서 그랩 데이터를 설정합니다.

파라미터

자료형	변수	설명
int	graspMode	그랩 모드 설정
float	graspForce	그랩 세기 설정
int	graspOption	그랩 옵션 설정
int	smoothGrasping	부드러운 그랩 사용

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
#define IS_ON 1
DG_GRASP_MODE graspMode = DG_GRASP_MODE_3F_3FINGER; // 세손가락 중점 그랩모드
float graspForce = 3.0;
int graspOption = DG_GRASP_OPTION_NONE; // 그랩 옵션 미사용
int smoothGrasping = IS_ON; // 부드러운 그랩 사용
DG_RESULT result = SetGraspData(graspMode, graspForce, graspOption, smoothGrasping);
```

관련 함수

[13.2 StartGraspMotion](#)

11.1.2 SetGraspForce

기능 설명

그랩 세기를 변경합니다. 해당 함수는 그랩 중에서도 그랩 세기 변경이 필요할 때 별도 세팅 없이 세기만을 변경할 수 있는 함수입니다.

파라미터

자료형	변수	설명
float	graspForce	그랩 세기 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float graspForce = 5.0;
DG_RESULT result = SetGraspForce(graspForce);
```

관련 함수

[13.2 StartGraspMotion](#)

11.2 게인 P 설정

그리퍼 P 게인을 설정합니다. P 게인 설정 값이 높으면 목표 지점에 빠르게 도달할 수 있으나, 그리퍼가 떨리는 현상이 나타날 수 있습니다.

11.2.1 SetJointGainP

기능 설명

한 개 관절의 P 게인값을 설정합니다.

파라미터

자료형	변수	설명
float	gainP	P 게인 입력
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.2.2 SetJointGainPFinger

기능 설명

한 개 손가락 관절들의 P 게인값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainP	P 게인 입력
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.2.3 SetJointGainPAI

기능 설명

모든 관절의 P 게인값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainP	P 게인 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float gainP[MAX_JOINT_COUNT] = { 0 };
gainP[0] = gainP[1] = 1; // 1번째, 2번째 관절 게인 설정
gainP[4] = gainP[6] = 2; // 5번째, 7번째 관절 게인 설정
DG_RESULT result = SetJointGainPAI(gainP);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.2.4 SetJointGainPBase

기능 설명

베이스 관절의 P 게인값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	gainP	P 게인 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float gainP[MAX_BASE_JOINT_COUNT] = { 0 };  
gainP[0] = 1; // 17번 관절, 1번 손가락에 해당하는 베이스 조인트 게인 설정  
gainP[1] = 2; // 18번 관절, 4번 손가락에 해당하는 베이스 조인트 게인 설정  
DG_RESULT result = SetJointGainPBase(gainP);
```

관련 함수

12.2 레시피 게인 함수

13.3 위치 제어

11.3 게인 D 설정

그리퍼 D 게인을 설정합니다. D 게인은 P 게인으로 인한 오버 슈트를 방지할 수 있으며 시스템의 안정성을 높일 수 있습니다. 하지만 설정 값이 높을 경우 그리퍼가 떨리는 현상이 나타날 수 있습니다.

11.3.1 SetJointGainD

기능 설명

한 개 관절의 D 게인값을 설정합니다.

파라미터

자료형	변수	설명
float	gainD	D 게인 입력
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.3.2 SetJointGainDFinger

기능 설명

한 개 손가락의 D 게인값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainD	D 게인 입력
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
```

관련 함수

[12.2 레시피 게인 함수](#)
[13.3 위치 제어](#)

11.3.3 SetJointGainDAII

기능 설명

모든 관절의 D 게인값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainD	D 게인 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상

0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조
-----------	-------	--------------------------------

예제

C/C++

```
float gainD[MAX_JOINT_COUNT] = { 0 };
gainD[0] = gainD[1] = 1; // 1번째, 2번째 관절 게인 설정
gainD[4] = gainD[6] = 2; // 5번째, 7번째 관절 게인 설정
DG_RESULT result = SetJointGainDAll(gainD);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.3.4 SetJointGainDBase

기능 설명

베이스 관절의 D 게인값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	gainD	D 게인 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float gainD [MAX_BASE_JOINT_COUNT] = { 0 };
gainD[0] = 1; // 17번 관절, 1번 손가락에 해당하는 베이스 조인트 게인 설정
gainD[1] = 2; // 18번 관절, 4번 손가락에 해당하는 베이스 조인트 게인 설정
DG_RESULT result = SetJointGainDBase(gainD);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.4 게인 I 설정

그리퍼 I 게인과 오차 적분 최대값을 설정합니다. I 게인은 오차를 줄이고 작은 변화에도 민감하게 반응하여 목표 값에 정밀하게 도달하기 위해 설정하는 게인입니다. I 게인값이 높으면 오차에 빠르게 반응하지만 그리퍼가 떨리는 현상이 발생할 수 있으며, 반대로 값이 낮으면 오차 보정이 느려지고 오차 적분 값의 회복이 지연되어 진동이 발생할 수 있습니다. 오차 적분 최대값은 오차 적분 값이 급격히 누적되는 것을 방지하는 역할을 하며, 값이 너무 낮으면 오차가 충분히 보정되지 않고, 너무 높으면 오차 적분 값의 회복 속도가 느려질 수 있습니다.

I 게인은 오차가 0 이 될 때까지 제어 신호가 계속 누적됩니다. 따라서, 그리퍼가 오차를 자체적으로 회복할 수 없는 상황에서는 관절 과부하로 인한 고장이 발생할 수 있습니다. 따라서, 안정적인 제어를 위해 적절한 값으로 정밀한 튜닝이 필요하며 사용 시 주의가 요구됩니다. I 게인값과 오차 적분 최대값의 0.01 단위 튜닝을 권장드립니다.

11.4.1 SetJointGainI

기능 설명

한 개 관절의 I 게인값과 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float	gainI	I 게인 입력
float	iLimit	오차 적분 최대값 설정
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
```

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

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.4.2 SetJointGainIFinger

기능 설명

한 개 손가락의 I게인값과 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainI	I 게인 입력
float*	iLimit	오차 적분 최대값 설정
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.4.3 SetJointGainAll

기능 설명

모든 관절의 I 게인값과 오차 적분 최댓값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainI	I 게인 입력
float*	iLimit	오차 적분 최댓값 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float gainI[MAX_JOINT_COUNT] = { 0 };
float iLimit[MAX_JOINT_COUNT] = { 0 };
gainI[0] = gainI[1] = 0.015; // 1번째, 2번째 관절 게인 설정
iLimit[0] = iLimit[1] = 0.06; // 1번째, 2번째 관절 오차 적분 최댓값 설정
DG_RESULT result = SetJointGainAll(gainI, iLimit);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.4.4 SetJointGainBase

기능 설명

베이스 관절의 I 게인값과 오차 적분 최댓값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	gainI	I 게인 입력
float*	iLimit	오차 적분 최댓값 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float gainI[MAX_BASE_JOINT_COUNT] = { 0 };
float iLimit[MAX_BASE_JOINT_COUNT] = { 0 };
gainI[0] = gainI [1] = 0.015; // 17번, 18번 관절에 해당하는 베이스 조인트 게인 설정
iLimit [0] = iLimit[1] = 0.06; // 17번, 18번 관절에 해당하는 베이스 조인트 오차 적분 최대값 설정
DG_RESULT result = SetJointGainIBase(gainI, iLimit);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.5 게인 PID 설정

11.5.1 SetControlPIDMode

기능 설명

그리퍼 제어기를 설정합니다. 해당 함수는 그리퍼 제어기를 PD 모드와 PID 모드로 설정할 수 있습니다.

파라미터

자료형	변수	설명
int	mode	게인 모드 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int mode = GAIN_MODE_PD; // PD 게인 모드로 설정
DG_RESULT result = SetControlPIDMode(mode);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.5.2 SetJointGainPID

기능 설명

한 개 관절의 P, I, D 게인 및 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float	gainP	P 게인 입력
float	gainD	D 게인 입력
float	gainI	I 게인 입력
float	iLimit	오차 적분 최대값 설정
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.5.3 SetJointGainPIDFinger

기능 설명

한 개 손가락의 P, I, D 게인 및 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainP	P 게인 입력
float*	gainD	D 게인 입력
float*	gainI	I 게인 입력
float*	iLimit	오차 적분 최대값 설정
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.5.4 SetJointGainPIDAll

기능 설명

모든 관절의 P, I, D 게인 및 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float*	gainP	P 게인 입력
float*	gainD	D 게인 입력
float*	gainI	I 게인 입력
float*	iLimit	오차 적분 최대값 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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; // 1번째, 2번째 관절 게인 설정
gainD[0] = gainD[1] = 5; // 1번째, 2번째 관절 게인 설정
gainI[0] = gainI[1] = 0.015; // 1번째, 2번째 관절 게인 설정
iLimit[0] = iLimit[1] = 0.06; // 1번째, 2번째 관절 오차 적분 최대값 설정
DG_RESULT result = SetJointGainPIDAll(gainP, gainD, gainI, iLimit);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.5.5 SetJointGainPIDAllEqual

기능 설명

모든 관절의 P, I, D 게인 및 오차 적분 최대값을 일괄 설정합니다.

파라미터

자료형	변수	설명
float	gainP	P 게인 입력
float	gainD	D 게인 입력
float	gainI	I 게인 입력
float	iLimit	오차 적분 최대값 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.5.6 SetJointGainPIDBase

기능 설명

베이스 관절의 P, I, D 게인 및 오차 적분 최대값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	gainP	P 게인 입력
float*	gainD	D 게인 입력
float*	gainI	I 게인 입력
float*	iLimit	오차 적분 최대값 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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; // 17번, 18번 관절에 해당하는 베이스 조인트 게인 설정
gainD[0] = gainD[1] = 5; // 17번, 18번 관절에 해당하는 베이스 조인트 게인 설정
gainI[0] = gainI[1] = 0.015; // 17번, 18번 관절에 해당하는 베이스 조인트 게인 설정
iLimit[0] = iLimit[1] = 0.06; // 17번, 18번 관절에 해당하는 베이스 오차 적분 최대값 설정
DG_RESULT result = SetJointGainPIDBase (gainP, gainD, gainI, iLimit)
```

관련 함수

[12.2 레시피 게인 함수](#)

[13.3 위치 제어](#)

11.6 동작 시간 설정

그리퍼가 목표 위치까지 도달하는 시간을 설정합니다. 동작 시간을 낮게 설정할수록 동작 속도가 빨라지며, 높게 설정할수록 동작 속도가 느려집니다. 동작 시간을 매우 낮게 설정할 경우 그리퍼가 최대 속도로 동작합니다. 이는 위치 제어에서만 반영되는 기능입니다. 동작시간은 컨트롤러 사양에 따라 다르게 동작할 수 있습니다. SDK 는 내부 타이머를 통해 동작 시간을 관리하고 있으나, SDK 를 사용하는 컨트롤러 환경이 낮을 경우 내부 타이머 시간을 보장할 수 없습니다. 또한 동작 시간에 움직임은 그리퍼 설정 값에 따라서 낮은 동작 시간을 보장할 수 없습니다.

11.6.1 SetMotionTimeJoint

기능 설명

한 개 관절의 위치 이동 시간을 설정합니다.

파라미터

자료형	변수	설명
int	motionTime	위치 이동 시간 설정(ms)
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

관련 함수

[12.1 레시피 포즈 함수](#)

[13.3 위치 제어](#)

11.6.2 SetMotionTimeFinger

기능 설명

한 개 손가락의 위치 이동 시간을 설정합니다.

파라미터

자료형	변수	설명
int*	motionTime	위치 이동 시간 설정
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int motionTime[MAX_FINGER_JOINT_COUNT] = { 0 };
int fingerNumber = 1;
motionTime[0] = 1000; // 1번 손가락을 위치 이동시간 1초로 설정
motionTime[2] = 1500; // 3번 손가락을 위치 이동시간 1.5초로 설정
DG_RESULT result = SetMotionTimeFinger(motionTime, fingerNumber);
```

관련 함수

[12.1 레시피 포즈 함수](#)

[13.3 위치 제어](#)

11.6.3 SetMotionTimeAll

기능 설명

모든 관절의 위치 이동 시간을 각각 설정합니다.

파라미터

자료형	변수	설명
int*	motionTime	위치 이동 시간 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int motionTime[MAX_JOINT_COUNT] = { 0 };
motionTime[0] = 1000; // 1번 관절을 위치 이동시간 1초로 설정
motionTime[2] = 1500; // 3번 관절을 위치 이동시간 1.5초로 설정
DG_RESULT result = SetMotionTimeAll(motionTime);
```

관련 함수

[12.1 레시피 포즈 함수](#)

[13.3 위치 제어](#)

11.6.4 SetMotionTimeAllEqual

기능 설명

모든 관절의 위치 이동 시간을 일괄 설정합니다.

파라미터

자료형	변수	설명
int	motionTime	위치 이동 시간 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int motionTime = 1500; // 위치 이동시간 1.5초 설정
DG_RESULT result = SetMotionTimeAllEqual(motionTime);
```

관련 함수

[12.1 레시피 포즈 함수](#)

[13.3 위치 제어](#)

11.6.5 SetMotionTimeBase

기능설명

베이스 관절의 위치 이동 시간을 각각 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
int*	motionTime	위치 이동 시간 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int motionTime[MAX_BASE_JOINT_COUNT] = { 0 };
motionTime[0] = 1000; // 17번 관절을 위치 이동시간 1초로 설정
motionTime[1] = 1500; // 18번 관절을 위치 이동시간 1.5초로 설정
DG_RESULT result = SetMotionTimeBase(motionTime);
```

관련 함수

[12.1 레시피 포즈 함수](#)

[13.3 위치 제어](#)

11.7 위치 제어 모드 설정

위치 제어 모드는 그랩 동작 시에 적용됩니다. 위치 제어 모드로 설정된 관절은 그랩 동작 시 그랩을 하지 않고 위치 제어를 수행합니다. 이는 그랩 수행 중 특정 관절을 움직이지 않고 싶을 때 사용하며, 다양한 그랩 방식을 구현할 수 있습니다.

11.7.1 SetPositionModeJoint

기능 설명

한 개 관절의 위치 제어 모드를 설정합니다.

파라미터

자료형	변수	설명
int	positionMode	위치 제어 모드 사용 설정
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
#define IS_ON 1
int positionMode = IS_ON;
int jointNumber = 1;
// 1번 관절은 위치 제어만 가능하도록 설정
DG_RESULT result = SetPositionModeJoint(positionMode, jointNumber);
```

관련 함수

[12.3 레시피 그랩 함수](#)

[13.2 StartGraspMotion](#)

11.7.2 SetPositionModeFinger

기능 설명

한 개 손가락 관절들의 위치 제어 모드를 설정합니다

파라미터

자료형	변수	설명
int*	positionMode	위치 제어 모드 사용 설정
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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;
// 1번 손가락 1번, 2번 관절은 위치 제어만 가능하도록 설정
// 1번 손가락 3번, 4번 관절은 위치 제어 모드 해제
DG_RESULT result = SetPositionModeFinger(positionMode, fingerNumber);
```

관련 함수

[12.3 레시피 그랩 함수](#)

[13.2 StartGraspMotion](#)

11.7.3 SetPositionModeAll

기능 설명

모든 관절들의 위치 제어 모드를 설정합니다.

파라미터

자료형	변수	설명
int*	positionMode	위치 제어 모드 사용 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
#define IS_OFF 0
#define IS_ON 1
int positionMode[MAX_JOINT_COUNT] = { IS_OFF };
positionMode[0] = IS_ON; // 1번 관절은 위치 제어만 가능하도록 설정
positionMode[11] = IS_OFF; // 12번 관절은 위치 제어 모드 해제
DG_RESULT result = SetPositionModeAll(positionMode);
```

관련 함수

[12.3 레시피 그래프 함수](#)

[13.2 StartGraspMotion](#)

11.7.4 SetPositionModeBase

기능 설명

베이스 관절들의 위치 제어 모드를 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
int*	positionMode	위치 제어 모드 사용 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
#define IS_OFF 0
#define IS_ON 1
int positionMode[MAX_BASE_JOINT_COUNT] = { IS_OFF };
positionMode[0] = IS_ON; // 17번 관절, 1번 손가락에 해당하는 베이스 조인트 위치 제어 모드 ON
positionMode[1] = IS_OFF; // 18번 관절, 4번 손가락에 해당하는 베이스 조인트 위치 제어 모드 OFF
DG_RESULT result = SetPositionModeBase(positionMode);
```

관련 함수

[12.3 레시피 그래프 함수](#)

[13.2 StartGraspMotion](#)

11.8 핑거팁 센서 설정

그리퍼 손 끝에 부착된 센서 데이터와 관련된 설정입니다. 해당 기능은 옵션 상품 핑거 팁 센서와 함께 사용할 수 있습니다.

11.8.1 SetFingerTipDataZero

기능 설명

현재 수신받은 핑거팁 데이터를 0으로 초기화합니다. 재호출 시 초기화하기 전의 원시 데이터로 돌아갑니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
DG_RESULT result = SetFingerTipDataZero();
```

관련 함수

[10.1.2 SetGripperOption](#)

11.9 전류 I 게인 설정

그리퍼의 전류 제어에서 사용되는 전류 I 게인과 관련된 설정입니다.

11.9.1 SetJointCurrentGainI

기능 설명

하나의 관절에 대해 전류 I 게인을 설정합니다.

파라미터

자료형	변수	설명
float	currentGainI	전류 I 게인 입력
float	currentILimit	오차 적분 최대값 입력
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainI = 0.005;
float currentILimit = 1000.0;
int jointNumber = 1;
DG_RESULT result = SetJointCurrentGainI(currentGainI, currentILimit, jointNumber);
```

관련 함수

[13.4 전류 제어](#)

11.9.2 SetJointCurrentGainIFinger

기능 설명

한 손가락의 관절들에 대해 전류 I 게인과 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float*	currentGainI	전류 I 게인 입력
float*	currentILimit	오차 적분 최대값 입력
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainI[MAX_FINGER_JOINT_COUNT] = { 0.005, 0.005, 0.005, 0.005 };
float currentILimit[MAX_FINGER_JOINT_COUNT] = { 1000.0, 1000.0, 1000.0, 1000.0 };
int fingerNumber = 1;
DG_RESULT result = SetJointCurrentGainIFinger(currentGainI, currentILimit, fingerNumber);
```

관련 함수

[13.4 전류 제어](#)

11.9.3 SetJointCurrentGainAll

기능 설명

모든 관절들에 대한 전류 I 게인과 오차 적분 최대값을 설정합니다.

파라미터

자료형	변수	설명
float*	currentGainI	전류 I 게인 입력
float*	currentILimit	오차 적분 최대값 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainI[MAX_JOINT_COUNT] = { 0 };
float currentILimit[MAX_JOINT_COUNT] = { 0 };
currentGainI[0] = currentGainI[1] = 0.005; // 1, 2번째 관절 전류 I 게인 설정
currentILimit[0] = currentILimit[1] = 1000.0; // 1, 2번째 관절 오차 적분 최대값 설정
DG_RESULT result = SetJointCurrentGainAll(currentGainI, currentILimit);
```

관련 함수

[13.4 전류 제어](#)

11.9.4 SetJointCurrentGainBase

기능 설명

베이스 관절의 전류 I 게인 및 오차적분 최대값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	currentGainI	전류 I 게인 입력
float*	currentILimit	오차 적분 최대값 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainI[MAX_BASE_JOINT_COUNT] = { 0.005, 0.005 };
float currentILimit[MAX_BASE_JOINT_COUNT] = { 1000.0, 1000.0 };
DG_RESULT result = SetJointCurrentGainBase(currentGainI, currentILimit);
```

관련 함수

[13.4 전류 제어](#)

11.10 전류 P 게인 설정

그리퍼의 전류 제어에서 사용되는 전류 P 게인과 관련된 설정입니다.

11.10.1 SetJointCurrentGainP

기능 설명

하나의 관절에 대해 전류 P 게인을 설정합니다.

파라미터

자료형	변수	설명
float	currentGainP	전류 P 게인 입력
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainP = 0.005;
int jointNumber = 1;
DG_RESULT result = SetJointCurrentGainP(currentGainP, jointNumber);
```

관련 함수

[13.4 전류 제어](#)

11.10.2 SetJointCurrentGainPFinger

기능 설명

한 손가락의 관절들에 대해 전류 P 게인을 설정합니다.

파라미터

자료형	변수	설명
float*	currentGainP	전류 P 게인 입력
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainP[MAX_FINGER_JOINT_COUNT] = { 0.005, 0.005, 0.005, 0.005 };
int fingerNumber = 1;
DG_RESULT result = SetJointCurrentGainPFinger(currentGainP, fingerNumber);
```

관련 함수

[13.4 전류 제어](#)

11.10.3 SetJointCurrentGainPAI

기능 설명

모든 관절들에 대한 전류 P 게인을 설정합니다.

파라미터

자료형	변수	설명
float*	currentGainP	전류 I 게인 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainP[MAX_JOINT_COUNT] = { 0 };
currentGainP[0] = currentGainP[1] = 0.005; // 1, 2번째 관절 전류 P 게인 설정
DG_RESULT result = SetJointCurrentGainPAI(currentGainP);
```

관련 함수

[13.4 전류 제어](#)

11.10.4 SetJointCurrentGainPBase

기능 설명

베이스 관절의 전류 P 게인 및 오차적분 최대값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	currentGainP	전류 P 게인 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float currentGainP[MAX_BASE_JOINT_COUNT] = { 0.005, 0.005 };
DG_RESULT result = SetJointCurrentGainPBase(currentGainP);
```

관련 함수

[13.4 전류 제어](#)

제 12 장 레시피 함수

12.1 레시피 포즈 함수

12.1.1 UpdateRecipeCurrentPoseData

기능 설명

현재 그리퍼 관절 위치 정보, 현재 설정된 동작 시간을 입력된 번호의 포즈 레시피에 저장합니다.

파라미터

자료형	변수	설명
int	recipeNumber	레시피 포즈 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float joint[MAX_JOINT_COUNT] = { 0 }; // 그리퍼 관절 개수만큼 입력
int motionTime = 500; // 그리퍼 목표 위치 도착 시간 설정
int recipeNumber = 1; // 레시피 번호 설정
DG_RESULT result = MoveJointAll(joint);
if (result == DG_RESULT_NONE) result = SetMotionTimeAllEqual(motionTime);
if (result == DG_RESULT_NONE) result = UpdateRecipeCurrentPoseData(recipeNumber);
```

관련 함수

[11.2.1 SetJointGainP](#) - [11.5.1 SetControlPIDMode](#)

[13.3 위치 제어](#)

12.1.2 UpdateRecipeTargetPoseData

기능 설명

현재 설정된 그리퍼 관절의 목표 위치 정보, 동작 시간을 입력된 번호의 포즈 레시피에 저장합니다.

파라미터

자료형	변수	설명
int	recipeNumber	레시피 포즈 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float joint[MAX_JOINT_COUNT] = { 0 }; // 그리퍼 관절 개수만큼 입력
int motionTime = 500; // 그리퍼 목표 위치 도착 시간 설정
int recipeNumber = 1; // 레시피 번호 설정
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;
```

관련 함수

[11.2.1 SetJointGainP](#) - [11.5.1 SetControlPIDMode](#)

[13.3 위치 제어](#)

12.1.3 LoadRecipePoseData

기능 설명

입력된 번호의 포즈 레시피 정보를 읽어 동작합니다.

파라미터

자료형	변수	설명
int	recipeNumber	레시피 포즈 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int recipeNumber = 1; // 레시피 번호 설정
DG_RESULT result = LoadRecipePoseData(recipeNumber);
```

관련 함수

[12.1.1 UpdateRecipeCurrentPoseData](#)

[12.1.2 UpdateRecipeTargetPoseData](#)

12.2 레시피 게인 함수

12.2.1 UpdateRecipeGainData

기능 설명

현재 설정된 P, I, D 게인, 오차 적분 최대값 및 PID 제어 모드를 입력된 번호의 레시피에 저장합니다.

파라미터

자료형	변수	설명
int	recipeNumber	레시피 게인 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float gainP = 1; // P 게인 설정
float gainD = 5; // D 게인 설정
float gainI = 0.015; // I 게인 설정
float iLimit = 0.02; // 오차 적분 최대값 설정
int controlPIDMode = GAIN_MODE_PID; // PID 제어모드 설정
int recipeNumber = 1; // 레시피 번호 설정
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;
```

관련 함수

[12.2.2 LoadRecipeGainData](#)

12.2.2 LoadRecipeGainData

기능 설명

입력된 번호의 게인 레시피 데이터를 읽어 설정합니다.

파라미터

자료형	변수	설명
int	recipeNumber	레시피 게인 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int recipeNumber = 1; // 레시피 번호 설정
DG_RESULT result = LoadRecipeGainData(recipeNumber);
```

관련 함수

[12.2.1 UpdateRecipeGainData](#)

12.3 레시피 그랩 함수

12.3.1 UpdateRecipeGraspData

기능 설명

현재 설정된 그랩 옵션, 그랩 모드, 부드러운 그랩 모드, 위치 제어 모드, 그랩 세기 설정 값을 입력된 그랩 레시피에 저장합니다.

파라미터

자료형	변수	설명
int	graspNumber	레시피 그랩 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float graspForce = 5; // 그랩 세기 설정
int graspOption = 1; // 그랩 옵션 설정
int smoothGrasping = 1; // 부드러운 그랩 모드 설정
int positionMode[MAX_JOINT_COUNT] = { 0 }; // 그리퍼 관절 개수만큼 입력
int recipeNumber = 1; // 레시피 번호 설정
DG_GRASP_MODE graspMode = DG_GRASP_MODE_3F_3FINGER; // 그랩 모드 설정
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);
```

관련 함수

[11.1 그랩 설정 함수](#)

[11.7.1 SetPositionMode](#)

[11.7.2 SetPositionModeFinger](#)

[11.7.3 SetPositionModeAll](#)

12.3.2 LoadRecipeGraspData

기능 설명

입력된 번호의 그랩 레시피 데이터를 읽어 설정합니다.

파라미터

자료형	변수	설명
int	graspNumber	레시피 그랩 번호

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int recipeNumber = 1; // 레시피 번호 설정
DG_RESULT result = LoadRecipeGraspData(recipeNumber);
if (result != DG_RESULT_NONE) return result;

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

관련 함수

[12.3.1 UpdateRecipeGraspData](#)

12.4 레시피 블렌드 함수

12.4.1 UpdateBlendJoint

기능 설명

입력된 블렌드 동작 데이터의 포즈 레시피 번호, 게인 레시피 번호, 동작 완료 후 대기 시간을 블렌드 동작 레시피에 추가합니다.

파라미터

자료형	변수	설명
RecipeBlendData	blendData	'9.3.2 struct RecipeBlendData' 참조

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
RecipeBlendData blendData;
blendData.blendWaitTime = 500; // 동작 완료 후 대기 시간 설정
blendData.recipeGainNumber = 1; // 게인 레시피 번호 설정
blendData.recipePoseNumber = 1; // 포즈 레시피 번호 설정
DG_RESULT result = UpdateBlendJoint(blendData);
```

관련 함수

[12.1.1 UpdateRecipeCurrentPoseData](#)

[12.1.2 UpdateRecipeTargetPoseData](#)

[12.2.1 UpdateRecipeGainData](#)

12.4.2 ClearMoveBlendJoint

기능 설명

현재 추가된 블렌드 데이터 리스트를 삭제합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
ClearMoveBlendJoint();
```

관련 함수

[12.4.1 UpdateBlendJoint](#)
[12.4.3 AddMoveBlendJoint](#)
[12.4.4 SetMoveBlendJoint](#)
[12.4.5 StartMoveBlendJoint](#)
[12.4.6 StopMoveBlendJoint](#)

12.4.3 AddMoveBlendJoint

기능 설명

업데이트 된 블렌드 데이터를 블렌드 리스트에 추가합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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();
}

```

관련 함수

[12.4.1 UpdateBlendJoint](#)
[12.4.2 ClearMoveBlendJoint](#)
[12.4.4 SetMoveBlendJoint](#)
[12.4.5 StartMoveBlendJoint](#)
[12.4.6 StopMoveBlendJoint](#)

12.4.4 SetMoveBlendJoint

기능 설명

현재 추가된 블렌드 리스트를 블렌드 모션 리스트에 저장합니다.

파라미터

자료형	변수	설명
int	blendNumber	블렌드 모션 넘버 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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);
}

```

관련 함수

[12.4.1 UpdateBlendJoint](#)
[12.4.2 ClearMoveBlendJoint](#)
[12.4.3 AddMoveBlendJoint](#)
[12.4.5 StartMoveBlendJoint](#)
[12.4.6 StopMoveBlendJoint](#)

12.4.5 StartMoveBlendJoint

기능 설명

입력된 블렌드 번호로 블렌드 모션을 수행합니다.

파라미터

자료형	변수	설명
int	blendNumber	블렌드 모션 넘버 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```

int blendNumber = 1;
DG_RESULT result = StartMoveBlendJoint(blendNumber);

```

관련 함수

[12.4.1 UpdateBlendJoint](#)
[12.4.2 ClearMoveBlendJoint](#)
[12.4.3 AddMoveBlendJoint](#)
[12.4.4 SetMoveBlendJoint](#)
[12.4.6 StopMoveBlendJoint](#)

12.4.6 StopMoveBlendJoint

기능 설명

현재 수행중인 블렌드 모션을 정지합니다.

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

관련 함수

[12.4.1 UpdateBlendJoint](#)

[12.4.2 ClearMoveBlendJoint](#)

[12.4.3 AddMoveBlendJoint](#)

[12.4.4 SetMoveBlendJoint](#)

[12.4.5 StartMoveBlendJoint](#)

제 13 장 동작 함수

13.1 ManualTeachMode

기능 설명

그리퍼 수동 티칭 모드를 설정합니다. 수동 티칭 모드는 사람이 직접 그리퍼 포즈를 설정할 수 있는 모드로 모든 관절을 직접 교시하여 티칭할 수 있습니다.

파라미터

자료형	변수	설명
int	isOn	수동 조작 모드 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

```
C/C++
int isOn = 1; // 수동 조작 모드 설정
ManualTeachMode(isOn);
```


13.2 StartGraspMotion

기능 설명

그랩 모션을 수행합니다.

파라미터

자료형	변수	설명
int	isGrasp	그랩 모션 수행 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
#define IS_ON 1
DG_GRASP_MODE graspMode = DG_GRASP_MODE_3F_3FINGER; // 세손가락 중점 그랩모드
float graspForce = 3.0;
int graspOption = DG_GRASP_OPTION_NONE; // 그랩 옵션 미사용
int smoothGrasping = IS_ON; // 부드러운 그랩 사용
DG_RESULT result = SetGraspData(graspMode, graspForce, graspOption, smoothGrasping);
if (result == DG_RESULT_NONE){
    int isGrasp = IS_ON;
    result = StartGraspMotion(isGrasp);
}
```

관련 함수

[11.1.1 SetGraspData](#)

[11.7 위치 제어 모드 설정](#)

13.3 위치 제어

13.3.1 MoveJoint

기능 설명

한 개 관절의 위치 이동을 수행합니다.

파라미터

자료형	변수	설명
float	targetJoint	목표 위치 값(degree) 입력
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float targetJoint = 20.9; // 목표 위치 20.9도 설정
int jointNumber = 1;
DG_RESULT result = MoveJoint(targetJoint, jointNumber);
```

13.3.2 MoveJointFinger

기능 설명

한 개 손가락 관절들의 위치 이동을 수행합니다.

파라미터

자료형	변수	설명
float*	targetJoint	목표 위치 값(degree) 입력
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
float targetJoint[MAX_FINGER_JOINT_COUNT] = { 0 };
targetJoint[0] = 0;
targetJoint[1] = 0;
targetJoint[2] = 60; // 목표 위치 60도 설정
targetJoint[3] = 60; // 목표 위치 60도 설정
int fingerNumber = 1;
DG_RESULT result = MoveJointFinger(targetJoint, fingerNumber);
```

관련 함수

[11.2 게인 P 설정](#)

[11.6 동작 시간 설정](#)

13.3.3 MoveJointAll

기능 설명

모든 관절들의 위치 이동을 수행 합니다.

파라미터

자료형	변수	설명
float*	targetJoint	목표 위치 값(degree) 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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; // 목표 위치 60도 설정
targetJoint[3] = targetJoint[7] = targetJoint[11] = 60; // 목표 위치 60도 설정
DG_RESULT result = MoveJointAll(targetJoint);
```

13.3.4 MoveServoJoint

기능 설명

모든 관절들의 위치 이동을 실시간으로 수행합니다. 해당 함수는 동작 시간 설정에 영향을 받지 않고 즉시 움직입니다. 해당 기능은 개발자 모드에서 사용가능합니다.

파라미터

자료형	변수	설명
float*	targetJoint	목표 위치 값(degree) 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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; // 목표 위치 60도 설정
targetJoint[3] = targetJoint[7] = targetJoint[11] = 60; // 목표 위치 60도 설정
DG_RESULT result = MoveServoJoint(targetJoint);
```

13.3.5 MoveJointBase

기능 설명

베이스 관절들의 위치 이동을 수행합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
float*	targetJoint	목표 위치 값(degree) 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

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

13.4 전류 제어

각 관절을 전류 제어로 구동 시킬 수 있는 방법을 안내합니다. 전류 제어 모드에서는 PID Gain 값 설정이 불가능 합니다.

13.4.1 SetCurrentControlMode

기능 설명

그리퍼 관절을 전류 제어 모드로 설정합니다.

파라미터

자료형	변수	설명
int	isOn	전류 제어 모드 설정

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int isOn = 1; // 전류 제어 모드 설정
DG_RESULT result = SetCurrentControlMode(isOn);
```

13.4.2 SetTargetCurrentJoint

기능 설명

한 개 관절의 목표 전류 값을 설정합니다.

파라미터

자료형	변수	설명
int	targetCurrent	목표 전류 값(mA) 입력
int	jointNumber	관절 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int targetCurrent = 50; // 목표 전류 값 50mA
int jointNumber = 1; // 1번 관절
DG_RESULT result = SetTargetCurrentJoint(targetCurrent, jointNumber);
```


13.4.3 SetTargetCurrentFinger

기능 설명

한 개 손가락 관절들의 목표 전류 값을 설정합니다.

파라미터

자료형	변수	설명
int*	targetCurrent	목표 전류 값(mA) 입력
int	fingerNumber	손가락 번호 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```

int targetCurrent[4] = { 50,50,50,50 }; // 목표 전류 값 50mA
int fingerNumber = 1; // 1번 손가락
DG_RESULT result = SetTargetCurrentFinger(targetCurrent, fingerNumber);

```

13.4.4 SetTargetCurrentAll

기능 설명

전체 관절들의 목표 전류 값을 설정합니다.

파라미터

자료형	변수	설명
int*	targetCurrent	목표 전류 값(mA) 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

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 SetTargetCurrentBase

기능 설명

베이스 관절들의 목표 전류 값을 설정합니다. 해당 함수는 DG-4F-M 전용 함수입니다.

파라미터

자료형	변수	설명
int*	targetCurrent	목표 전류 값(mA) 입력

리턴

번호	값	설명
0	DG_RESULT_NONE	정상
0 이 아닌 경우	ERROR	부록 2 에러 리스트 참조

예제

C/C++

```
int targetCurrent[MAX_BASE_JOINT_COUNT];
targetCurrent[0] = 50; // 17 번 관절, 1 번 손가락에 해당하는 베이스 조인트 목표 전류값 설정
targetCurrent[1] = 50; // 18 번 관절, 4 번 손가락에 해당하는 베이스 조인트 목표 전류값 설정
DG_RESULT result = SetTargetCurrentBase(targetCurrent);
```

부록 1 용어집 및 기능설명

B

Baudrate (보드레이트)

데이터 전송의 단위가 baud 인 데이터 전송 속도.

Blend list (블렌드 리스트)

레시피 포즈와 레시피 게인이 포함된 연속 위치 동작을 위한 리스트 모음.

Blend motion (블렌드 모션)

블렌드 리스트가 저장된 모션.

Blend move (블렌드 동작)

레시피 포즈와 레시피 게인으로 이뤄진 블렌드 리스트를 등록된 순서대로 연속적으로 동작.

C

Cartesian coordinate (카테시안 좌표계)

3D 공간 상에서 점의 위치를 수학적으로 표현하는 좌표계로, X축, Y축, Z축으로 이루어진 공간 좌표계 및 직교 좌표계.

Communication Period (통신 주기)

장치 간에 데이터 송수신 시간.

COMPORT(컴포트)

직렬 통신에 사용되는 컴퓨터의 입출력 포트.

D

Developer Mode (개발자 모드)

사용자가 직접 개발한 제어기를 이용하는 모드. 그리퍼의 각 관절의 정보를 수신하여 각 관절을 직접 제어할 수 있는 모드.

E

Encoder (엔코더)

회전 운동이나 직선 운동을 하는 물체의 위치와 속도 정보를 전기적인 신호로 출력하는 센서.

F

Fix tilt option (기울기 고정 그랩 옵션)

그리퍼 현재 포즈 기준, 해당 옵션으로 그랩 동작 실행 시 손가락의 4 번째 관절의 기울기가 유지된 채로 그랩 동작을 수행하는 기능.

I

Inpose (인포즈)

인포즈는 위치제어에서 목표 관절 위치 값으로 이동할 때, 목표 위치 값에 도착했다는 것을 확인하기 위한 범위. 이는 목표 위치에 도달하기 전에 다른 모션을 수행하기 위해 주로 사용.

J

Joint (관절)

1 자유도 회전 축을 가지는 구동부. Delto Gripper 시리즈 경우 DG-3F-M, DG-3F-B 는 12 개 관절, DG-4F-M 은 18 개 관절, DG-5F-R, DG-5F-L 은 20 개 관절로 구성.

L

Low Pass Filter (저역 통과 필터)

특정 주파수 이하를 통과시키는 필터로, 수신된 디지털 신호의 잡음을 줄이고, 급격한 변화를 완화시켜 부드러운 결과를 출력.

M

Modbus protocol (모드버스 프로토콜)

자동화 장치 간 통신을 위해 개발된 산업용 표준 통신 프로토콜로 마스터-슬레이브 구조를 사용하며, RTU, ASCII, TCP/IP 등의 다양한 통신 방식을 지원. Slave ID 를 통해 연결된 장치들을 식별 가능.

Modbus RTU (모드버스 RTU)

RS-232, RS-485 등 직렬 통신을 기반으로 하는 프로토콜이며, 바이너리 형식의 전송 방식으로 효율적이고 빠른 통신이 가능.

Modbus TCP (모드버스 TCP)

TCP/IP 네트워크를 기반으로 동작하는 Modbus Protocol. 데이터를 패킷의 형태로 전송하며, 네트워크를 통한 빠르고 유연한 통신이 가능.

Module (모듈)

독립적인 부분으로 분리한 최소 구동 단위. Delto Gripper 시리즈는 각 관절이 하나의 독립적인 모듈로 이루어져 있어, 독립적인 구동이 가능하며, 확장성 및 유지 보수가 용이. 각 모듈은 모터, 엔코더, 모터드라이브, 감속기 등으로 구성.

O

Operator Mode(사용자 모드)

제품 내부의 제어기를 사용하여 제어하는 모드. Modbus Protocol 을 지원하며, 단순히 레지스터 맵을 변경하거나, 제공되는 UI 조작으로 그리퍼를 쉽게 제어 가능한 모드.

Overshoot (오버 슈트)

제어 시스템에서 목표 값으로 수렴하는 과정에서 일시적으로 목표 값을 초과하는 현상

P

Parallel option (평행 그랩 옵션)

기울기 고정 그랩 옵션과 달리, 그리퍼의 현재 포즈와 상관없이 항상 손 끝의 평면끼리 맞붙도록 그랩하는 옵션으로 그랩 시 물체와의 접촉면을 최대화할 수 있음.

PD control (PD 제어)

비례(Proportional), 미분(Derivative) 제어를 결합한 피드백 제어 시스템. 오차에 비례한 반응과 오차의 변화율에 따른 반응을 결합하여 속도가 빠르고 안정적인 제어가 가능하나 완벽한 오차 보정은 어려움.

PID control (PID 제어)

비례(Proportional), 적분(Integral), 미분(Derivative) 제어를 결합한 피드백 제어 시스템. 자동화 및 로봇 시스템에서 원하는 목표 값을 달성하기 위해 사용되며, 안정적이고 정밀한 제어가 가능.

Position control (위치 제어)

모터나 기계 부품의 위치를 조절하는 제어 방식. 현재 위치를 바탕으로 목표 위치까지 이동시키는 것을 위치 제어라고 하며, PID 제어를 통한 정밀한 제어가 가능.

R

Recipe (레시피)

로봇(그리퍼)을 간편하고, 효율적으로 제어하기 위해 주요 설정 값들을 미리 저장해둔 데이터의 집합. 사전에 레시피를 저장해두면 저장된 레시피를 통해 제어의 편의성 향상.

S

Smooth grasping (부드러운 그랩 모드)

물체를 그랩할 때 물체가 상하지 않도록 약하게 그랩하는 모드이며, 물체에 접촉한 이후에는 설정된 그랩 세기로 변경.

Socket communication (소켓 통신)

네트워크를 통해 두 개의 장치가 데이터를 송수신하는 방식. IP 주소, PORT 번호를 통해 송수신할 장치를 식별.

Software Development Kit (SDK)

소프트웨어 애플리케이션을 개발하기 위한 도구 및 라이브러리 모음. 사용자는 SDK 를 활용하여 그리퍼 제어를 위한 기능들을 손쉽게 구현 가능.

T

TCP/IP

인터넷 및 네트워크 통신을 위한 핵심 프로토콜로, 데이터를 신뢰성 있게 전달할 수 있으며. LAN 포트를 통해 유선으로 네트워크를 연결할 수 있고, 여러 장치를 효율적으로 연결 가능.

W

WSA (Window Socket API)

윈도우 운영 체제에서 소켓 프로그래밍을 지원하는 API 로 클라이언트와 서버 간의 데이터 통신을 가능하게 하는 API.

부록 2 에러 리스트

값	명칭	설명
0	DG_RESULT_NONE	정상
1	DG_RESULT_SYSTEM_SETTING_NOT_PERFORMED	시스템 세팅이 완료되지 않았습니다. 시스템 세팅 함수를 수행한 이후 수행하세요
100	DG_RESULT_VALUE_IS_NEGATIVE	입력 데이터가 음수입니다.
101	DG_RESULT_OVERFLOW_JOINT_COUNT	입력 가능한 최대 모터 번호를 초과했습니다. MAX_JOINT_COUNT = 20
102	DG_RESULT_OVERFLOW_FINGER_COUNT	입력 가능한 최대 손가락 번호를 초과했습니다. MAX_FINGER_COUNT = 5
103	DG_RESULT_OVERFLOW_TCP_COUNT	입력 가능한 최대 좌표계 개수를 초과했습니다. CARTESIAN_COORDINATE_POSE_COUNT = 6
105	DG_RESULT_OVERFLOW_ADD_BLEND_SIZE	포즈 리스트가 최대치입니다. 더 이상 블렌드 포즈를 추가할 수 없습니다. MAX_BLEND_ADD_POSE_COUNT = 50
106	DG_RESULT_OVERFLOW_RECIPE_BLEND_SIZE	입력 가능한 최대 블렌드 모션 번호를 초과했습니다. MAX_BLEND_COUNT = 10
107	DG_RESULT_BLEND_SIZE_ZERO	저장된 블렌드 모션이 없습니다.
108	DG_RESULT_NOT_SUPPORTED_MODEL	해당 기능을 지원하는 모델이 아닙니다.
109	DG_RESULT_DATA_IS_ZERO	입력 데이터가 0 입니다.
110	DG_RESULT_DATA_IS_NOT_BOOLEAN	0 과 1 만 입력이 가능합니다.
111	DG_RESULT_NOT_FOUND_MODEL	모델 번호를 찾을 수 없습니다.
112	DG_RESULT_OVERFLOW_RECIPE_POSE_COUNT	입력 가능한 최대 레시피 포즈 번호를 초과했습니다. MAX_RECIPE_POSE_COUNT = 100
113	DG_RESULT_OVERFLOW_RECIPE_GAIN_COUNT	입력 가능한 최대 레시피 게인 번호를 초과했습니다. MAX_RECIPE_GAIN_COUNT = 20
114	DG_RESULT_OVERFLOW_RECIPE_	입력 가능한 최대 레시피 그래프 번호를

	GRASP_COUNT	초과했습니다. MAX_RECIPES_GRASP_COUNT = 20
115	DG_RESULT_INVALID_CONTROL_MODE	유효하지 않은 시스템 컨트롤 모드입니다.
116	DG_RESULT_INVALID_GRASP_MODE_DATA	입력 가능한 최대 그랩 모드 번호를 초과했습니다. 9.2.6 enum GAIN_MODE
118	DG_RESULT_OVERFLOW_MAX_BYTE_DATA	입력 가능한 byte 최댓값(255)을 초과했습니다.
119	DG_RESULT_OVERFLOW_CURRENT_LIMIT	입력 가능한 최대 전류 값을 초과했습니다. M 모델: MAX_INPUT_CURRENT = 150 S 모델: MAX_INPUT_CURRENT = 360
120	DG_RESULT_IS_NOT_TORQUE_CONTROL_MODE	토크 제어 모드가 아닙니다.
121	DG_RESULT_NOT_SUPPORTED_DATA_TYPE	지원하지 않는 데이터 수신 명령어입니다.
122	DG_RESULT_BACKUP_START	데이터 백업 중입니다.
123	DG_RESULT_INVALID_PASSWORD	잘못된 비밀번호 입력입니다.
124	DG_RESULT_OVERFLOW_GPIO_COUNT	입력 가능한 GPIO 최대 번호를 초과했습니다. MAX_GRIPPER_GPIO_SIZE = 6
125	DG_RESULT_OVERFLOW_GRASP_FORCE	그랩 세기 입력값을 초과했습니다. MAX_GRASP_FORCE = 20
126	DG_RESULT_OVERFLOW_BLEND_WAIT_TIME	블렌드 대기 시간 입력값을 초과했습니다. MAX_BLEND_WAIT_TIME = 30000
127	DG_RESULT_NOT_FOUND_RESTORE_DATA	경로상에 복원할 파일이 없습니다.
128	DG_RESULT_RESTORE_START	복원이 수행 중입니다.
200	DG_RESULT_NOT_ARRIVED	목표 위치에 도달하지 못했습니다.
201	DG_RESULT_NOT_START_BLEND_MOVE	블렌드 동작 모드가 아닙니다.
202	DG_RESULT_ALREADY_BLEND_MOVE_STATE	블렌드 동작 중입니다.
203	DG_RESULT_ALREADY_TCP_MOVE	TCP 위치 이동 중입니다.
204	DG_RESULT_RECIPES_IS_NOT_JOINT_MODE	포즈 레시피가 관절 위치 제어 모드가 아닙니다.

205	DG_RESULT_ACTIVATE_CURRENT_ CONTROL_MODE	전류 제어 모드가 활성화된 상태입니다.
206	DG_RESULT_ACTIVATE_ GRASP_MOTION	그랩을 수행하고 있는 상태입니다.
207	DG_RESULT_ACTIVATE_MANUAL_ CONTROL_MODE	수동 제어 모드가 활성화된 상태입니다.
401	DG_RESULT_MODULE_FAULT_1	1 번 모듈에 이상 발견
402	DG_RESULT_MODULE_FAULT_2	2 번 모듈에 이상 발견
403	DG_RESULT_MODULE_FAULT_3	3 번 모듈에 이상 발견
404	DG_RESULT_MODULE_FAULT_4	4 번 모듈에 이상 발견
405	DG_RESULT_MODULE_FAULT_5	5 번 모듈에 이상 발견
406	DG_RESULT_MODULE_FAULT_6	6 번 모듈에 이상 발견
407	DG_RESULT_MODULE_FAULT_7	7 번 모듈에 이상 발견
408	DG_RESULT_MODULE_FAULT_8	8 번 모듈에 이상 발견
409	DG_RESULT_MODULE_FAULT_9	9 번 모듈에 이상 발견
410	DG_RESULT_MODULE_FAULT_10	10 번 모듈에 이상 발견
411	DG_RESULT_MODULE_FAULT_11	11 번 모듈에 이상 발견
412	DG_RESULT_MODULE_FAULT_12	12 번 모듈에 이상 발견
413	DG_RESULT_MODULE_FAULT_13	13 번 모듈에 이상 발견
414	DG_RESULT_MODULE_FAULT_14	14 번 모듈에 이상 발견
415	DG_RESULT_MODULE_FAULT_15	15 번 모듈에 이상 발견
416	DG_RESULT_MODULE_FAULT_16	16 번 모듈에 이상 발견
417	DG_RESULT_MODULE_FAULT_17	17 번 모듈에 이상 발견
418	DG_RESULT_MODULE_FAULT_18	18 번 모듈에 이상 발견
419	DG_RESULT_MODULE_FAULT_19	19 번 모듈에 이상 발견
420	DG_RESULT_MODULE_FAULT_20	20 번 모듈에 이상 발견
421	DG_RESULT_MODULE_FAULT_21	첫 번째 핑거팁 모듈에서 이상 발견
422	DG_RESULT_MODULE_FAULT_22	두 번째 핑거팁 모듈에서 이상 발견
423	DG_RESULT_MODULE_FAULT_23	세 번째 핑거팁 모듈에서 이상 발견
424	DG_RESULT_MODULE_FAULT_24	네 번째 핑거팁 모듈에서 이상 발견
425	DG_RESULT_MODULE_FAULT_25	다섯 번째 핑거팁 모듈에서 이상 발견
500	DG_RESULT_SOCK_EXCEPTION	소켓 통신에 예외가 발생했습니다.
501	DG_RESULT_SOCK_FAILED_	소켓 통신 연결을 실패했습니다.

	WSA_START_UP	
1002	DG_RESULT_NO_GRASP_OBJECT	물체를 잡지 않았습니다.
1003	DG_RESULT_ONLY_SUPPORTED_3FINGER	DG-3F-B, DG-3F-M 모델 전용
1004	DG_RESULT_NOT_SUPPORTED_CONTROL_MODE_OPERATOR	사용자 모드는 지원하지 않는 기능입니다.
1005	DG_RESULT_NOT_SUPPORTED_CONTROL_MODE_DEVELOPER	개발자 모드는 지원하지 않는 기능입니다.
1006	DG_RESULT_NOT_SUPPORTED_PORT_NUM	지원하지 않는 포트 번호입니다.
2000	DG_RESULT_PORT_EXCEPTION	시리얼 통신에서 예외가 발생했습니다.
2001	DG_RESULT_PORT_FAILED_START_UP	시리얼 통신 연결을 실패했습니다.
2002	DG_RESULT_PORT_SET_CONFIG_ERROR	시리얼 포트 설정을 실패했습니다.
2003	DG_RESULT_PORT_SET_TIMEOUT_ERROR	시리얼 포트 타임 아웃 설정을 실패했습니다.
2004	DG_RESULT_PORT_SET_COMMASK_ERROR	시리얼 통신 마스크 설정을 실패했습니다.
2009	DG_RESULT_DIAGNOSING_SYSTEM	진단 기능이 수행중입니다.

