

Robot Controller

PRC SERIES



User Guide

System Configuration
Program Installation
Teach Pendant Operation
Robot Programming

PRESTO

프레스토솔루션

Preface

본 문서는 프레스토 솔루션의 PRC 시리즈 로봇 제어기의 사용 설명서입니다.

PRC 시리즈 로봇 제어기를 최고의 성능으로 유지하기 위해서 본 매뉴얼을 사전에 충분히 숙지한 후 설치, 운전, 점검 및 유지 보수 작업을 정확한 방법으로 실행하기 바랍니다.

어떠한 경우에도 제품을 분해하거나 제품 또는 일부 부품을 임의로 변경하는 행위는 허용되지 않습니다. 제조사의 승인 없이 인증되지 않은 인력이 제품 또는 일부 부품을 변경할 경우, 이로 인해 발생하는 오동작이나 제품 수명 저하에 대해서는 제조사의 책임이 없습니다.

(주)프레스토솔루션은 고객의 의견을 수렴해 제품 개선에 반영하고자 항상 노력하고 있습니다. 정기/비정기적 제품 개선, 사양 변경 또는 사용자의 편의를 위해 별도의 고지 없이 본 매뉴얼의 내용을 변경할 수 있습니다.

본 매뉴얼에서는 사용 주의 사항, 시스템 구성 방법, PC 프로그램 설치 방법, 티치 팬던트 소개 및 로봇 프로그래밍에 대한 설명을 다룹니다.

본 매뉴얼 내용 중 의문 사항이 있거나 개선에 관한 의견이 있으면, 언제든지 (주)프레스토솔루션 본사로 문의해 주시기 바랍니다.

(주)프레스토솔루션의 서면 승인 없이, 본 매뉴얼의 일부 또는 전체를 복사, 인쇄 및 재배포 하는 것을 금지합니다.

안전 기호

본 문서에서 안전에 관계된 내용은 아래 기호를 사용합니다. 해당 기호가 있는 설명은 중요한 내용이므로 반드시 숙지 후 실시하여 주시기 바랍니다.



취급을 잘못할 경우 사용자가 사망 또는 중상을 입는 위험이 예상되는 경우 (각 언어 적용)



취급을 잘못할 경우 사용자가 경상을 입거나 또는 물적 손해가 발생하는 위험이 예상되는 경우 (각 언어 적용)

제품 사용에 유용한 추가 정보는 다음과 같은 기호로 안내합니다.



제품 사용에 필요한 정보 및 기타 주의 사항 (각 언어 적용)

안전 주의 사항

본 매뉴얼에 명시된 정보를 능숙하고 올바르게 적용하는 것은 시스템의 안전한 설치, 시운전 및 유지 관리를 위한 전제 조건입니다. 반드시 매뉴얼에 제시된 지침을 숙지하고 이행할 수 있는 전문 인력이 제품을 다루어야 합니다.

보관 및 설치 시 주의

직사광선을 피해 주시고, 보관 온도 및 습도 범위에서 제품을 보관하시기 바랍니다.

과열로 인한 제품의 손상 및 오동작 방지를 위해 동작 주위 온도 및 습도를 고려하시기 바랍니다.

**경고**

다음 환경에서는 보관 및 사용을 금합니다.

- 폭발이나 화재 위험이 있는 장소
- 야외
- 광업 분야
- 비 또는 수분 (액체), 물기가 있는 곳

전문 인력 필요

**경고**

특정 자격을 갖춘 직원만이 제어 시스템 관련 작업을 수행할 수 있습니다.

제품에 대한 전문성이 부족한 인력이 작업할 경우, 제어 장치의 부적절한 개입으로 인해 기계/장치의 비정상적인 작동, 인체 상해, 장비 손상이 야기될 수 있습니다.

프로그래밍 안전 지침

경고

- 매뉴얼에 명시된 지침은 모든 상황에서 정확하게 준수해야 합니다. 위반 시 잠재적 위험이 발생할 수 있습니다.
 - 매뉴얼에 제공된 안전 지침 이외에도 문제의 상황에 적합한 안전 예방책 및 사고 예방 조치를 준수해야 합니다.
 - 전력 저하 또는 정전 발생 시 중단된 프로그램을 올바르게 다시 시작할 수 있도록 조치해야 합니다. 이러한 상황에서는 위험한 작동 조건이 일시적으로 발생하지 않아야 합니다.
 - 자동화 시스템에서 발생하는 오류로 인하여 부상이나 심각한 장비 손상을 초래할 수 있는 모든 환경에서 시스템이 전체적으로 안전한 작동 상태를 유지할 수 있도록 추가적으로 외부 안전 조치를 취해야 합니다.
-

유지 보수에 대한 안전 지침

경고

- 제품을 임의로 분해 또는 개조하지 마십시오. 위반 시 이로 인해 발생한 모든 인명 피해 또는 물적 피해에 대해 제조사는 책임지지 않습니다.
 - 문제가 발생하고 조치가 불가할 경우, 당사 기술 지원 및 A/S팀에 문의하여 주십시오.
-

Table of contents

Preface.....	i
안전 기호	ii
안전 주의 사항.....	iii
보관 및 설치 시 주의.....	iii
전문 인력 필요	iii
프로그래밍 안전 지침.....	iv
유지 보수에 대한 안전 지침.....	iv
Chapter 1. Overview	9
1.1 제품 번호 체계	10
1.2 제품 사양.....	11
1.3 제품 치수.....	12
1.4 각 부 명칭	13
1.5 System Configuration	14
1.6 제품 설치.....	15
1.7 Create a ENI File (EC-Engineer).....	16
Chapter 2. PRC Manager.....	19
2.1 프로그램 설치 및 제거	19
2.2 로봇 컨트롤러 접속.....	23
2.3 메인 화면.....	26
2.4 System Configuration	30
2.5 Firmware Upgrade.....	33
2.6 Reboot Controller.....	33
2.7 System Viewer	34
2.8 Variable.....	35
2.9 IO.....	37
2.10 Log Message.....	38
2.11 Position	39
2.12 Program Manager	39
2.13 Program Editor.....	40
2.14 Sub Program Manager.....	41
2.15 Sub Program Editor	42
2.16 Command Terminal.....	42
2.17 Scope.....	43

	2.18 Jog & Teaching	48
	2.19 Homing.....	50
	2.20 Motion Wizard	51
Chapter 3.	로봇 구동	55
	3.1 Manual Motion	55
	3.2 Programming Motion.....	57
Chapter 4.	로봇제어기 소프트웨어	59
	4.1 소프트웨어 아키텍처	59
	4.2 프로그램 아키텍처	60
Chapter 5.	Robot Motion.....	61
	5.1 Motion Type & Mode.....	61
	5.2 Coordinate System	62
	5.3 Motion Profile.....	63
	5.4 Motion 궤적.....	64
Chapter 6.	Pre-defined Commands for System & Robot.....	65
	6.1 System Commands	65
	6.2 Robot Commands.....	66
부록 A. ERROR Code 및 조치.....		67

List of tables

Table 1 PRC 제품 사양	11
Table 2 PRC 명령어	11
Table 3 Dimension	12
Table 4 각 부 명칭 설명	13
Table 5 PC 최소 사양	19
Table 6 User Level 별 접근 권한	24
Table 7 Manual, Program (Auto), Homing 모드	61
Table 8 Joint, Work, Ready Motion	61

List of figures

Figure 1 PRC 전면부	9
Figure 2 제품 번호 체계	10
Figure 3 각 부 명칭	13
Figure 4 시스템 구성	14
Figure 5 설치 이격 거리	15
Figure 6 메인 화면 구성	26
Figure 7 스카라 로봇	61

Chapter 1. Overview

프레스토솔루션의 PRC 시리즈는 최대 10대의 로봇을 구동할 수 있는 고성능 로봇 제어기 제품군입니다.

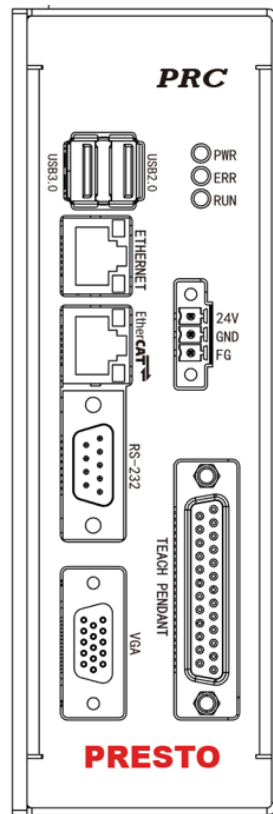


Figure 1 PRC 전면부

- PRC 시리즈 제어기 한 대로 최대 10대의 로봇 구동
- 최대 64축 모터 제어
- 10개의 프로그램을 독립적으로 구동
- EtherCAT 오픈 프로토콜 사용: 슬레이브 확장이 편리하고 호환성 우수
- 자체 로봇 전용 프로그래밍 언어("RobotGul"): 초보자도 쉽게 프로그램 문법을 익힐 수 있는 환경 제공
- GUI를 통해 로봇 추가 후 바로 사용 (파라미터 설정 작업 불필요)
- 프로그램 안에서 명령어를 쉽게 검색 가능한 Intellisense 기능 제공

1.1 제품 번호 체계

제품 번호는 총 8 자리로 구성됩니다.

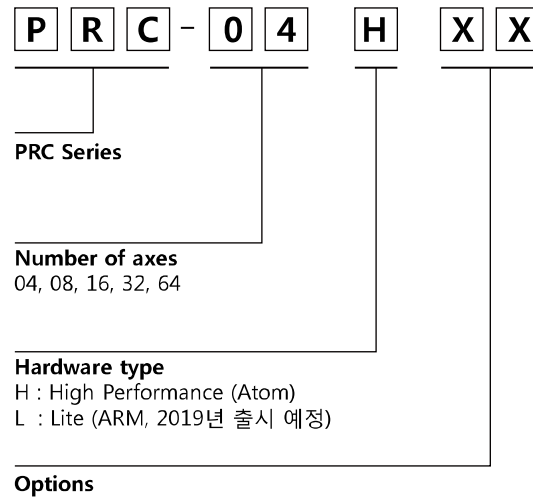


Figure 2 제품 번호 체계

1.2 제품 사양

기본 사양

Table 1 PRC 제품 사양

Item	Descriptions
사이즈	55 x 200 x 123 (mm)
외함 재질	알루미늄
입력 전원	12 ~ 24V _{DC}
소비 전류	500 mA (조건: 24V _{DC} , TP 사용)
지원 로봇 및 축수	로봇 10EA, 64 Axes
통신 방식 및 제어 주기	<ul style="list-style-type: none"> EtherCAT® CSP Mode Cycle Time : 0.5 msec, 1 msec, 2 msec ※ 로봇 종류, 개수, 축 수에 따라 속도를 조정
지원 로봇 형태	직교, 스카라, 다관절, 델타 로봇
Motion	Jog, PTP, LI, CI, Blending
프로그램 지원 언어	RobotGul (10개 프로그램 태스크, 100개 서브 프로그램 지원)
사용자 인터페이스	PC GUI, 7인치 TP, 3.5인치 TP
Processor	인텔 아톰 프로세서 E 시리즈
Memory	RAM 4 GB, ROM (16 ~ 32 GB)
동작 주위 온/습도	0 ~ 45°C, 20 ~ 80% RH (결로가 없을 것)
보관 주위 온/습도	-15 ~ 70°C, 10 ~ 90% RH (결로가 없을 것)

명령어

Table 2 PRC 명령어

Item	Descriptions
Motion	17개의 Motion 명령어 제공 (Move, MoveJ, MoveL, MoveC, StartBlend...)
Parameter	JtVel, JtPos, WkVel, WkPos, VelP ...
IO	DIn, DOut, AIn, AOut
Wait	Wait, WaitM
Time	SysTime, SysData ...
Etc	Print, PrgStart, PrgStop ...

1.3 제품 치수

제품 치수

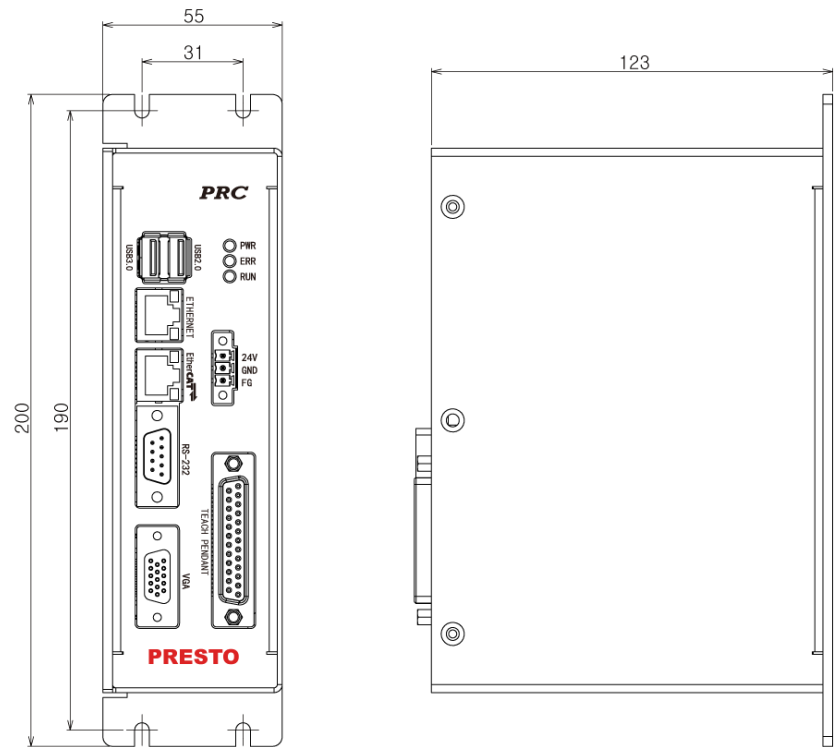


Table 3 Dimension

Item	Descriptions
H x W x D	200 x 55 x 123 (mm)

1.4 각 부 명칭

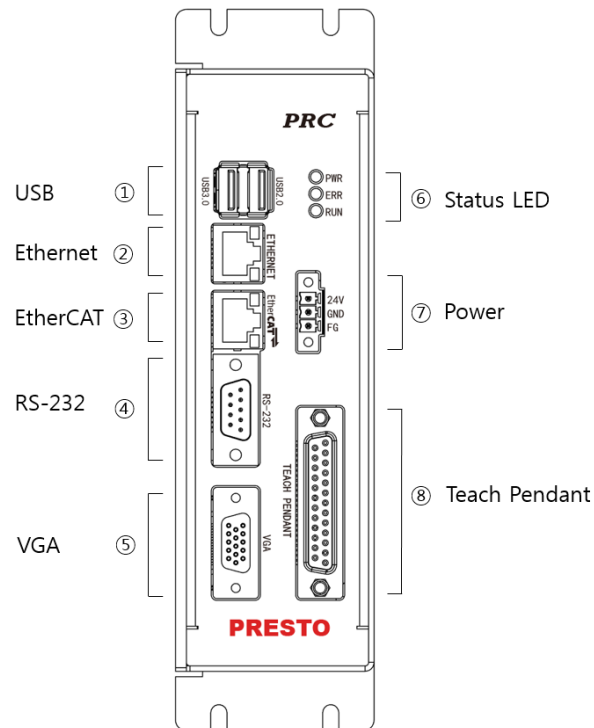


Figure 3 각 부 명칭

Table 4 각 부 명칭 설명

Item	Descriptions
① USB	USB 마우스, 키보드 연결 단자: 디버깅용으로 사용
② Ethernet	PC GUI 프로그램, 7인치 티치 팬던트와 연결하는 TCP/IP 통신 단자
③ EtherCAT	EtherCAT 슬레이브 모듈과 연결하는 통신 단자
④ RS-232	시리얼 통신용 단자
⑤ VGA	모니터 연결 단자: 디버깅용으로 사용
⑥ Status LED	<ul style="list-style-type: none"> PWR (Green): Power On (MCU, 메모리 전원) 상태 표시 ERR (Red): Error 상태 표시 (서보 알람, 로봇 알람, EtherCAT 통신 알람) RUN (Yellow): 로봇이 구동 중인 상태 표시
⑦ Power	12~24V DC 전원 연결 단자
⑧ Teach Pendant	3.5인치 티치 팬던트 연결 단자

1.5 System Configuration

시스템은 아래 그림과 같이 로봇 제어기와 조작기인 티치 팬던트로 구성됩니다.

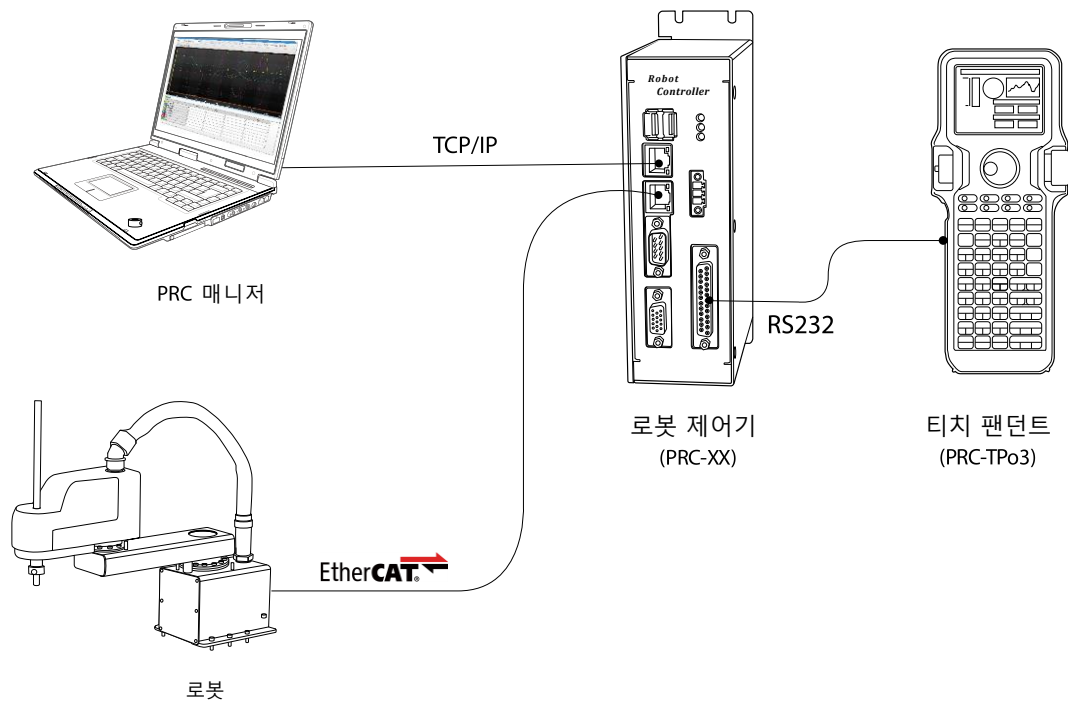


Figure 4 시스템 구성

1.6 제품 설치

설치 이격 거리

방열을 위해 아래 그림과 같이 타제품과 최소한의 이격 거리를 유지해야 합니다.

Item	Descriptions
상/하/좌우	50 x 50 x 20 (mm)

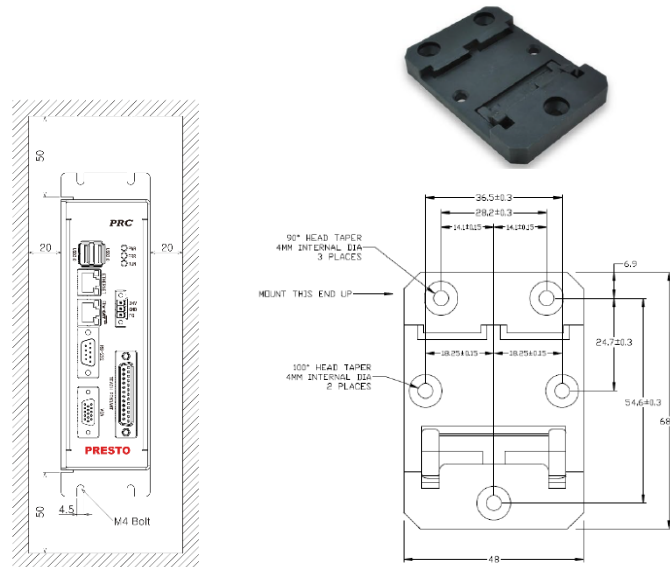


Figure 5 설치 이격 거리

⚠ 주의

- 직사광선을 피해 주십시오.
- 실외 설치는 금지합니다.

Power Connector

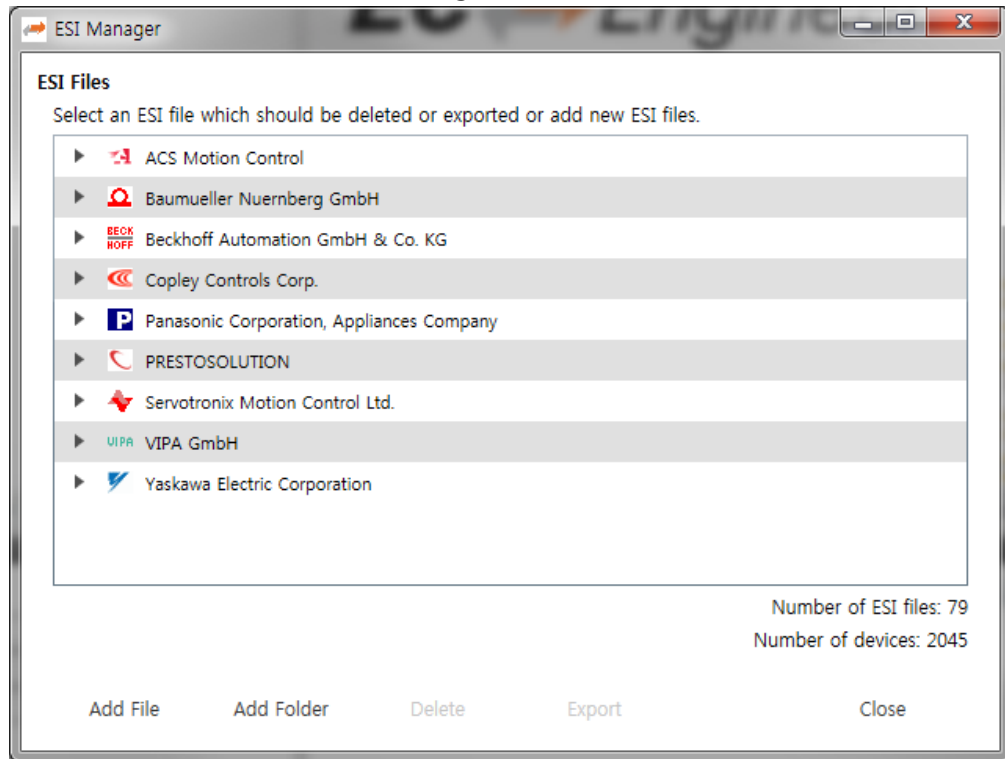
Pin No	Descriptions
1	+24V (DC 12~24V)
2	GND (+24V DC Ground)
3	FG (Frame Ground)

① 참고

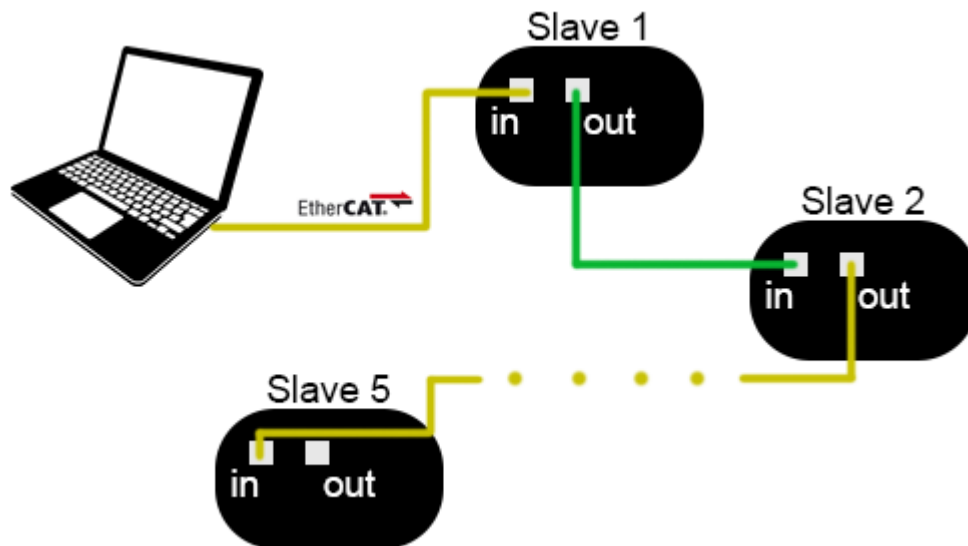
- DIN-Rail 사용시 FG는 Chassis와 연결해 주시기 바랍니다.
- 케이블 사양에 따라 EtherCAT 통신 에러가 발생할 수 있습니다. ETG (EtherCAT Technology Group)에서 권장하는 케이블 사양을 사용하시기 바랍니다.

1.7 Create a ENI File (*EC-Engineer*)

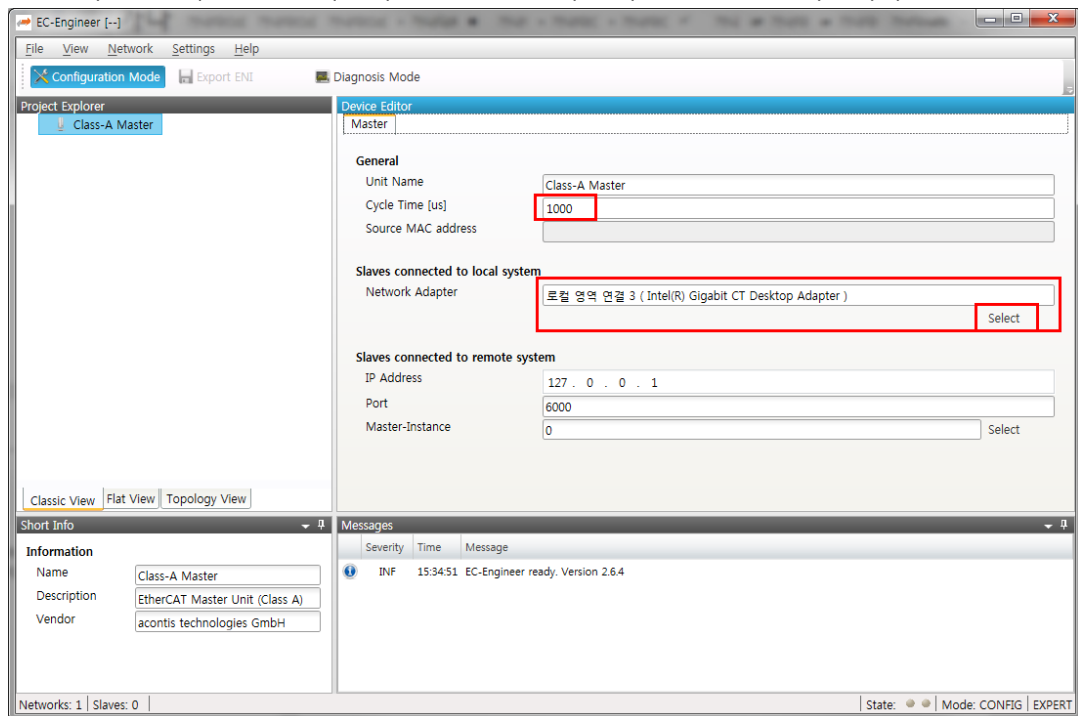
1. 각 Slave의 맞는 ESI 파일을 EC의 ESI Manager를 이용하여 추가합니다.



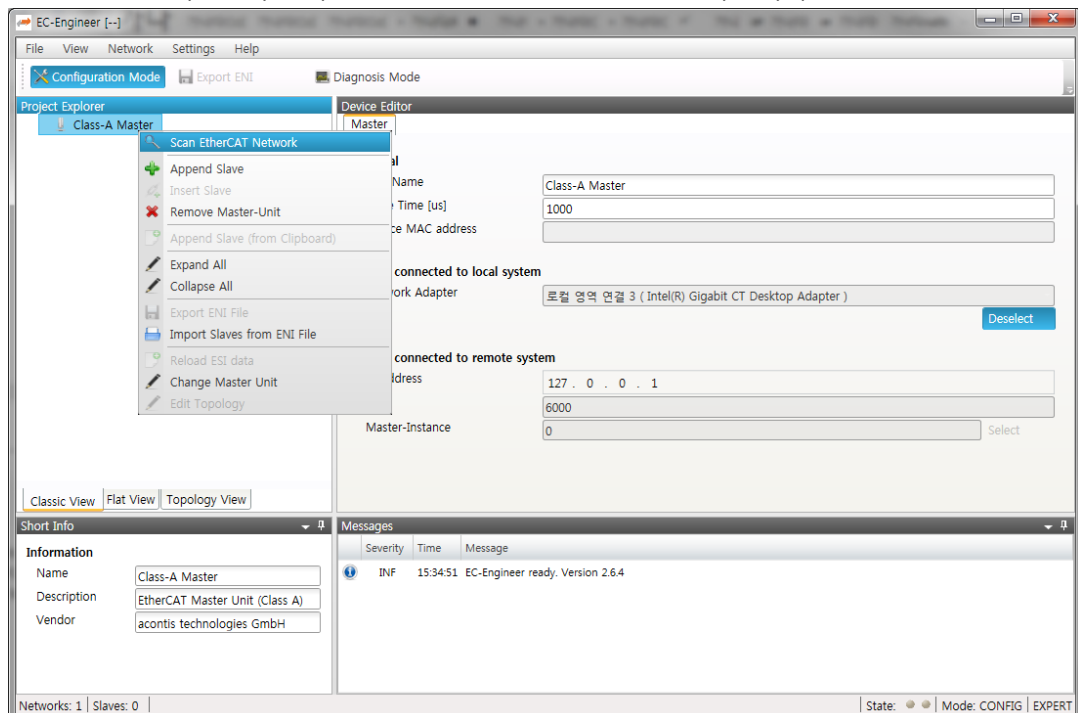
2. 모든 Slave의 환경을 구축하고, 첫번째 Slave 입력 단과 PC를 EtherCAT 케이블로 연결합니다.



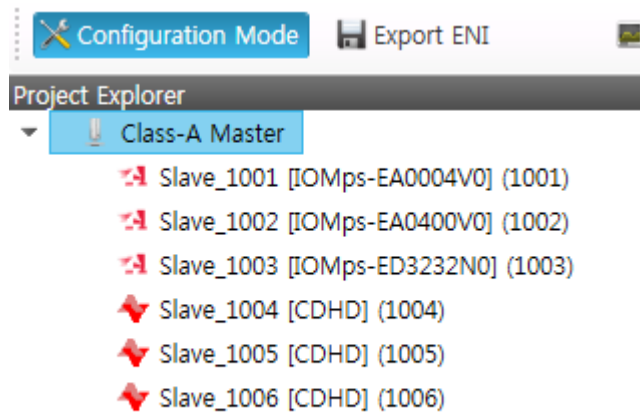
3. CycleTime과 Slave와 연결된 네트워크 포트를 선택한 후 Select를 클릭합니다.



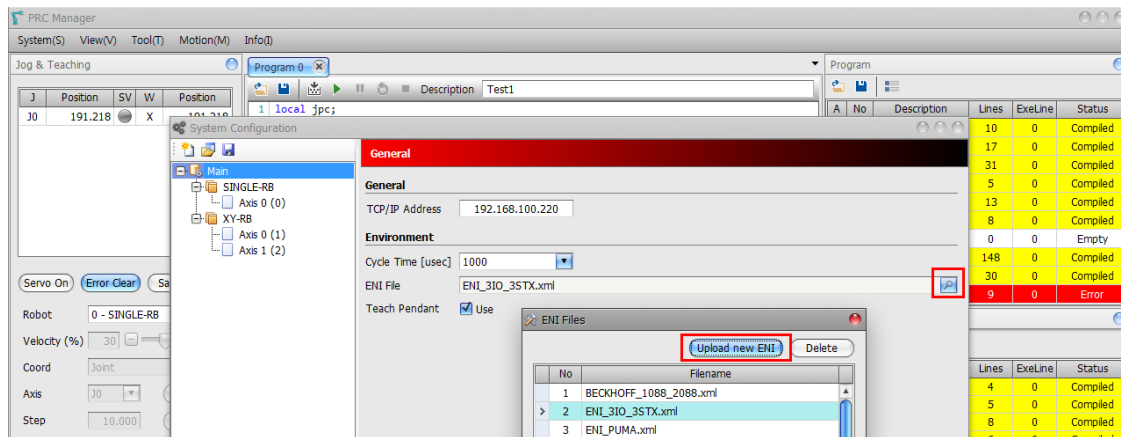
4. Class-A Master를 우 클릭한 후 Scan EtherCAT Network를 클릭합니다.



5. 모든 Slave가 정상적으로 스캔 되었다면, Export ENI를 클릭하여 ENI File을 원하는 경로에 저장합니다.



6. PC – PRC – Slave 의 환경을 다시 구축하여 PRC Manager에서 ENI File을 Upload하여 사용합니다.



Chapter 2. PRC Manager

PRC Manager는 PRC 시리즈 로봇 컨트롤러의 GUI 프로그램입니다. 파라미터 설정과 변경, 로봇의 조그와 프로그램 구동을 지원합니다. 로봇의 상태 확인과 진단을 위해 Scope, Log Message도 제공하고 있습니다. 추후에 3D View도 제공할 예정입니다.

PRC Manager는 필요에 따라 유저 레벨을 Operator, Programmer 및 Administrator의 3단계로 구분해, 로봇 기능에 대한 접근을 제한하고 있습니다.

2.1 프로그램 설치 및 제거

설치 PC 최소 사양

PRC Manager를 설치하고 구동하기 위해서 필요한 PC의 최소 사양은 다음과 같습니다.

Table 5 PC 최소 사양

Item	Descriptions
OS	Windows 7 이상 (32/64bit 지원)
HDD	200 Mbytes 이상의 여유 공간
RAM	1 Gbytes 이상의 여유 공간
사전 설치 프로그램	NET Framework v4.5.2 이상, DirectX 9 이상

설치 파일 구성

설치 파일은 아래와 같이 2개의 파일, 1개의 폴더로 구성됩니다.

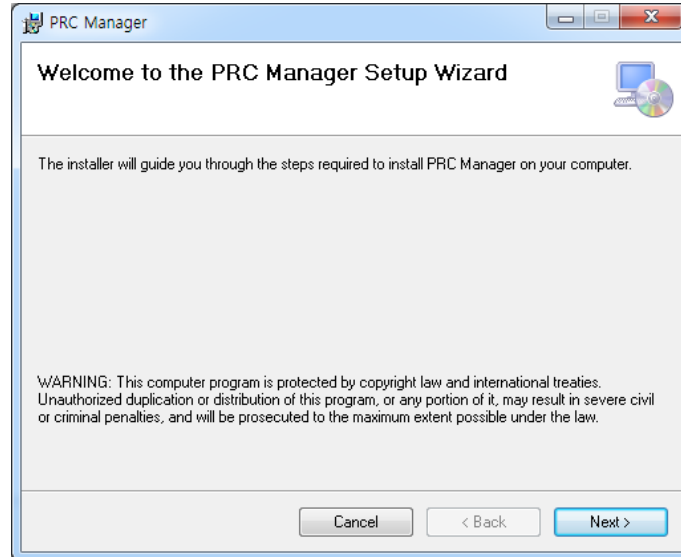
이름	수정한 날짜	유형	크기
DotNetFX452	2018-12-26 오전...	파일 폴더	
PRC_Manager	2018-12-24 오전...	Windows Installer...	68,967KB
setup	2018-12-24 오전...	응용 프로그램	782KB

Item	Descriptions
setup.exe	설치 실행 파일
RC_GUI.msi	설치 패키지 파일 (해당 파일로 설치가 가능하나, 설치 정보가 맞지 않아 정상적인 Uninstall을 할 수 없음)
DotNetFX452	프로그램 구동을 위한 필수 환경 파일 (.NET Framework 4.5.2 – Setup.exe 실행 시 자동으로 설치 진행됨)

프로그램 설치

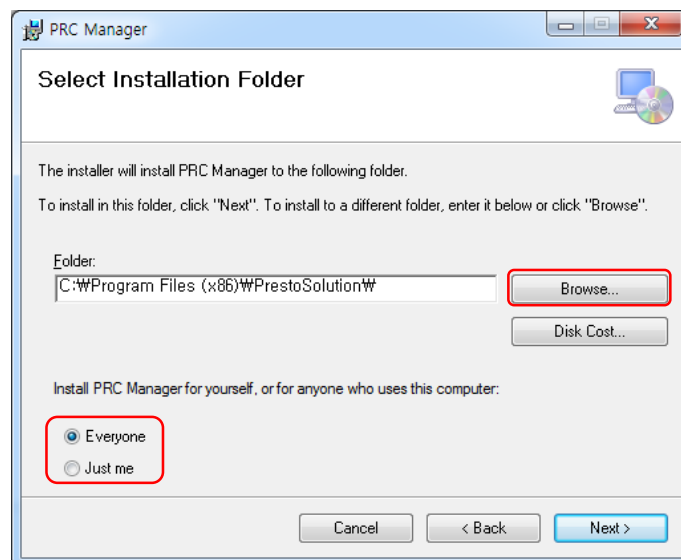
설치 파일을 **setup.exe**를 실행하면 아래와 그림과 같은 설치 마법사가 시작됩니다

i. 설치 마법사 시작



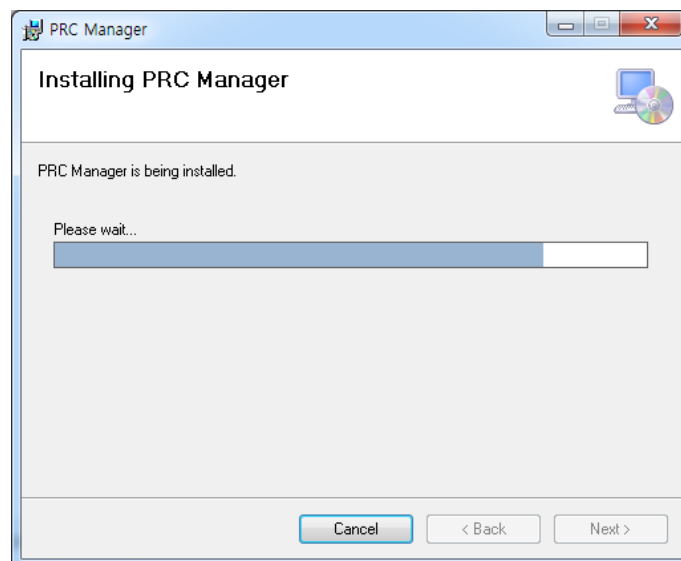
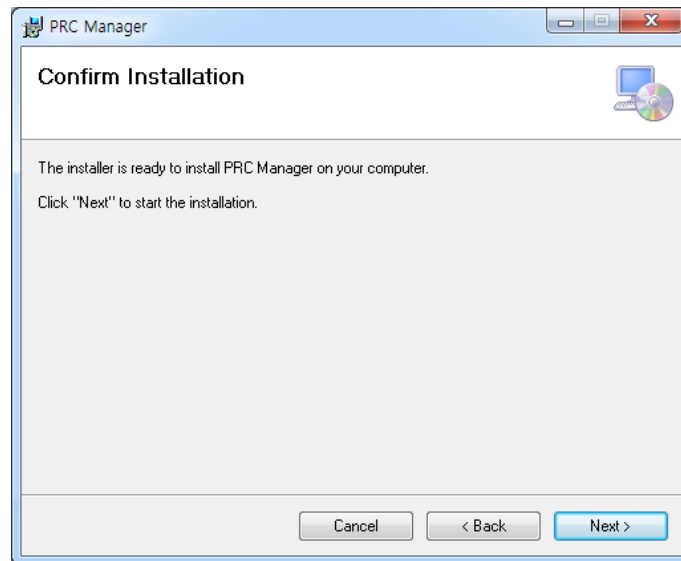
설치를 진행하기 위해 Next 버튼을 클릭합니다.

ii. Select Installation Folder

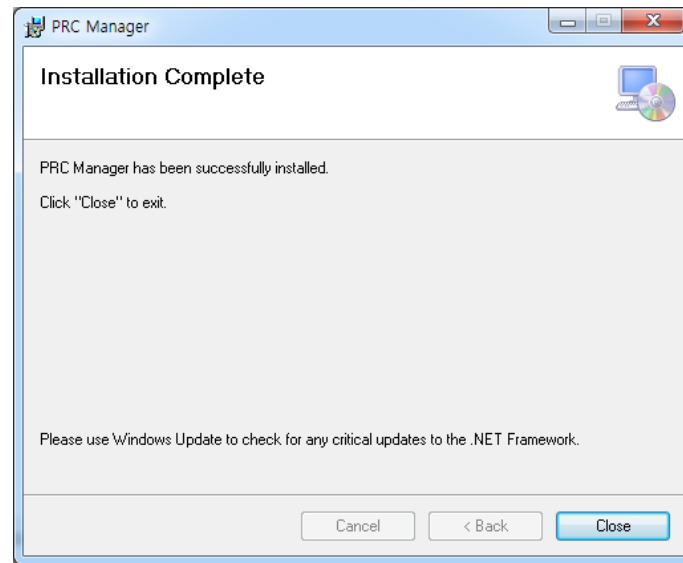


- Browse: 버튼을 눌러 설치할 경로를 선택합니다.
- Everyone/Just me: 프로그램을 사용할 수 있는 사용자를 선택합니다.

확인 후 Next 버튼을 클릭하면, 다음과 같이 설치 확인 화면과 진행 화면이 차례로 나타나며 설치가 진행됩니다.



iii. 설치 완료

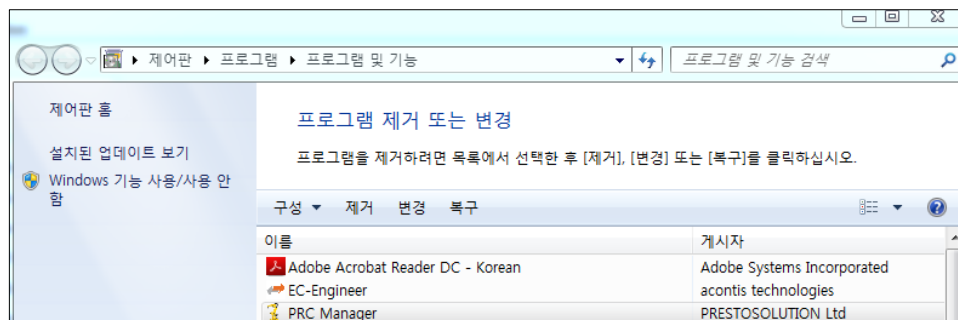


설치가 완료되어 Installation Complete 화면이 나타나면, [Close] 버튼을 눌러 설치를 마칩니다.

-
- ① **참 고** 정상적으로 설치된 경우 바탕 화면과 시작 프로그램에 PRC Manager 단축 아이콘이 생성됩니다.
-

프로그램의 제거

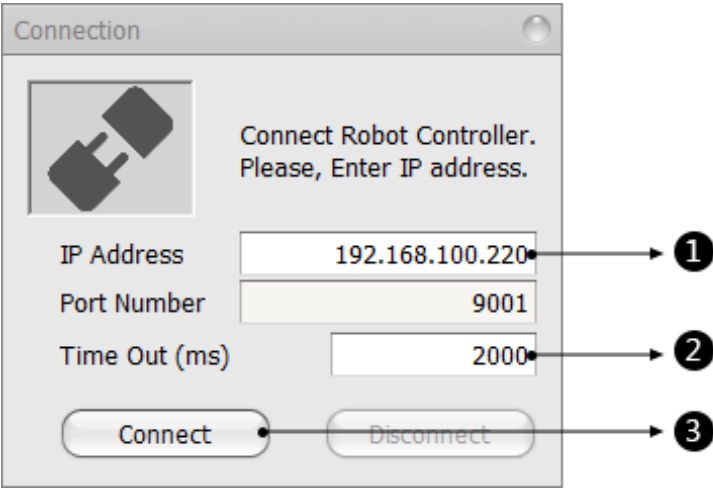
Windows 제어판의 '프로그램 추가/제거'에서 삭제가 가능합니다.



2.2 로봇 컨트롤러 접속

통신 연결

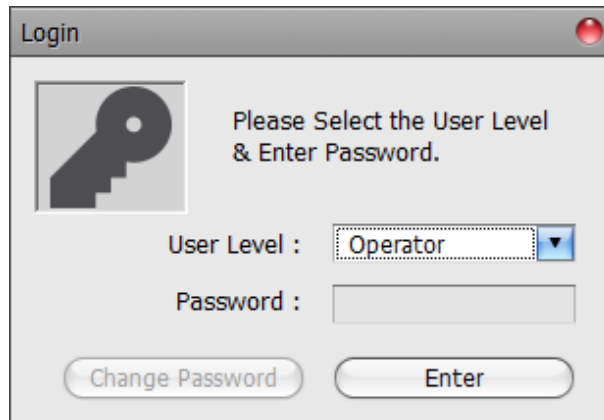
프로그램을 실행하면 아래와 같은 Connection 화면이 나타납니다.



User Level	Default Password
① IP address	<ul style="list-style-type: none">• Default IP Address: 192.168.100.100• PC IP Address도 192.168.100.XXX 설정
② Time Out (ms)	<ul style="list-style-type: none">• 로봇 컨트롤러와 PC GUI 간의 통신 Time Out 입력• 1,000 [msec] ~ 10,000 [msec] 지원
③ Connect 버튼	연결 시도 버튼

Login

로봇 컨트롤러와 연결이 되면 아래 Login 화면이 나타납니다.



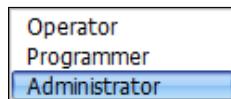
The Login dialog box has a title bar with the word 'Login' and a red close button. Inside, there is a key icon on the left. To the right of the icon, the text reads 'Please Select the User Level & Enter Password.' Below this text, there are two input fields: 'User Level' with a dropdown menu currently showing 'Operator', and 'Password' with an empty text box. At the bottom of the dialog, there are two buttons: 'Change Password' and 'Enter'.

User Level에 따른 접근 권한은 다음과 같습니다.

Table 6 User Level 별 접근 권한

Access Level	System Configuration	Programming
Operator	Read Only	Read Only
Programmer	Read Only	Read & Write
Administrator	Read & Write	Read & Write

- i. User Level을 선택합니다.



- ii. 패스워드를 입력합니다.

User Level	Default Password
Operator	패스워드 없이 로그인
Programmer	prc
Administrator	prc123

- iii. [Enter] 버튼을 클릭해 로그인을 시도합니다.

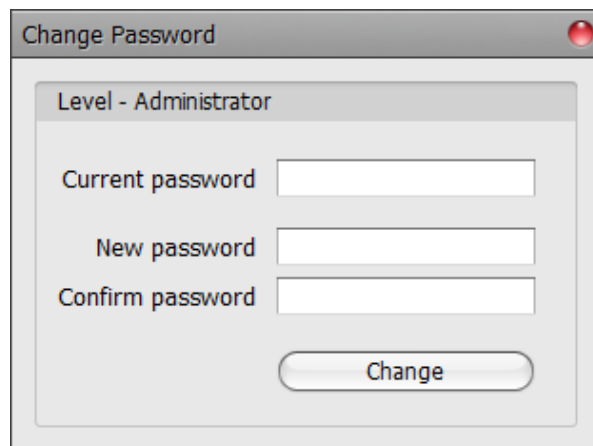
Change Password

로그인 후 패스워드를 변경할 수 있습니다.

- iv. Login 후 System Menu에 Login Menu를 선택하여 Login 창의 좌측 하단 Change Password 버튼을 클릭합니다.

A screenshot of the 'Login' dialog box. It has a title bar with 'Login' and a red close button. Inside, there is a key icon on the left. To the right of the icon, the text reads 'Please Select the User Level & Enter Password.' Below this, there is a 'User Level' label followed by a dropdown menu showing 'Administrator'. Below that is a 'Password' label followed by a text input field with three asterisks '***' on the right. At the bottom left, there is a button labeled 'Change Password' which is highlighted with a red dashed border. To its right is an 'Enter' button.

- v. 변경하고자 하는 User Level을 선택하고, 현재(Current) Password와 변경하고자 하는 New Password를 입력한 후 Change 버튼을 클릭합니다.

A screenshot of the 'Change Password' dialog box. It has a title bar with 'Change Password' and a red close button. Inside, there is a section titled 'Level - Administrator'. Below this title, there are three text input fields: 'Current password', 'New password', and 'Confirm password'. At the bottom right of the input fields, there is a button labeled 'Change'.

2.3 메인 화면

메인 화면은 다음과 같습니다. 메뉴 트리, 좌측 수동 조작 메뉴, 중앙 메인 화면, 하단 및 상태 표시줄, 그리고 우측 프로그램 메뉴로 구성되어 있습니다.

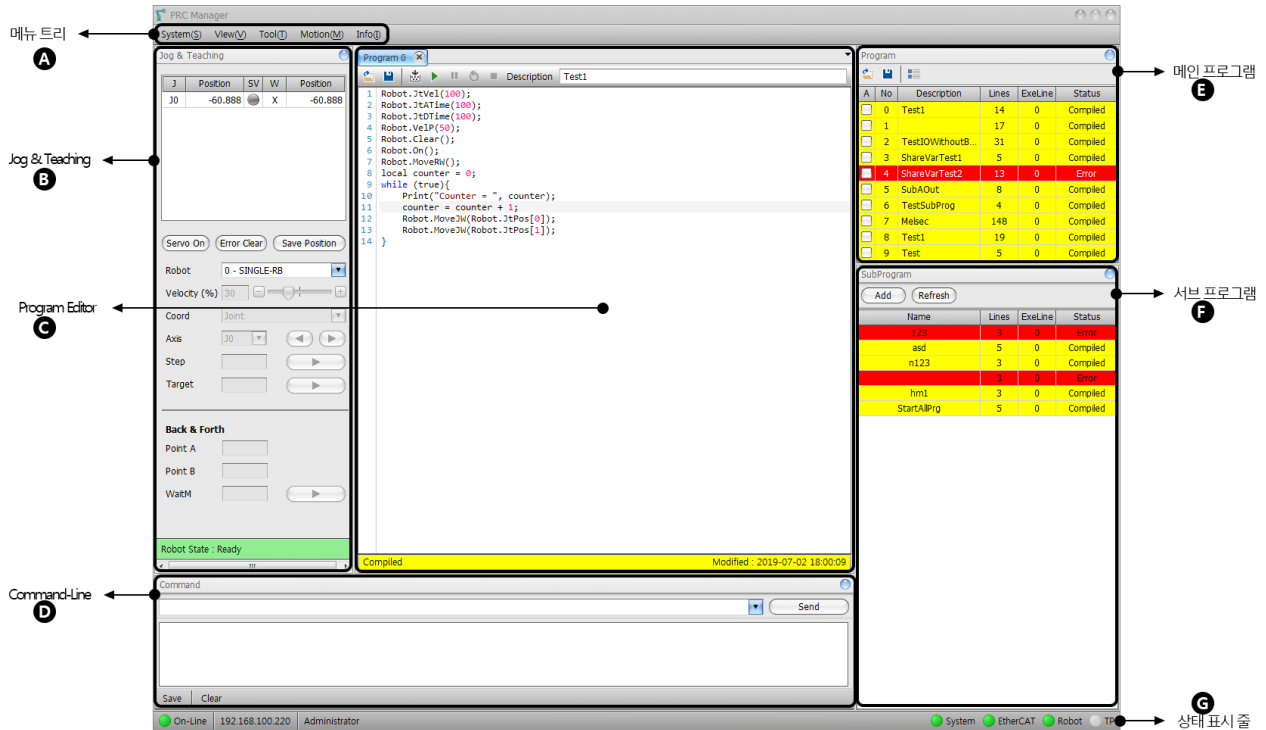
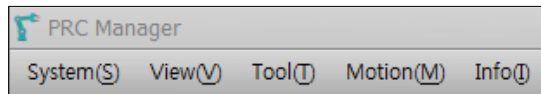


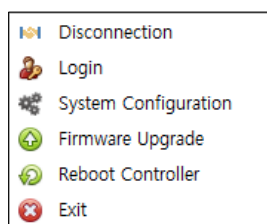
Figure 6 메인 화면 구성

A 메뉴 트리

메뉴 트리는 다음과 같이 System, View, Tool, Motion 및 Info와 같은 5가지 메뉴로 구성되어 있습니다.

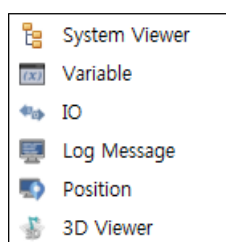


메뉴 트리 - System (S)



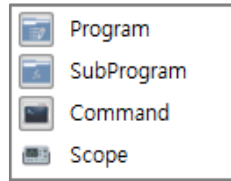
Item	Descriptions
(Dis)Connection	로봇 컨트롤러 연결/해제
Login	<ul style="list-style-type: none"> • Login Level을 변경 • 패스워드 변경
System Configuration	로봇 컨트롤러의 System, Robot, Axis 등의 초기 설정 및 수정
Firmware Upgrade	로봇 컨트롤러의 펌웨어 업그레이드
Reboot Controller	로봇 컨트롤러 재시작
Exit	PC GUI 종료

메뉴 트리 - View (V)



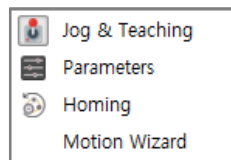
Item	Descriptions
System Viewer	로봇 컨트롤러에 설정된 System 설정 확인
Variable	각 로봇의 Variable의 추가/수정/삭제
IO	<ul style="list-style-type: none"> • 로봇 컨트롤러와 연결되어 있는 IO 값을 실시간으로 표시 • DO, AO 출력값 변경
Log Message	로봇 컨트롤러에서 발생한 각종 메시지를 실시간으로 표시
Position	모든 로봇의 Position을 실시간으로 표시
3D Viewer	추가 예정

메뉴 트리 - Tool (T)



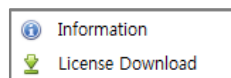
Item	Descriptions
Program	<ul style="list-style-type: none"> 최대 10개 Program Task를 구현 프로그램의 작성, 수정, 구동 등의 조작
Sub Program	<ul style="list-style-type: none"> Sub Program 파일은 함수 호출처럼 사용할 수 있는 프로그램 Program Task에서 호출해서 사용 가능 (최대 100개 파일 지원)
Command	로봇 컨트롤러에 Single Line Command를 전달한 후 응답 값 확인
Scope	각 Axis Motion 상태와 IO 등의 실제 구동 값을 파형 확인

메뉴 트리 - Motion (M)



Item	Descriptions
Jog & Teaching	수동 조작으로 로봇 조작
Parameters	Motion과 관련된 Parameter 값 변경
Homing	Homing 동작 및 Homing Parameter 설정
Motion Wizard	로봇의 모션을 쉽게 만들 수 있게 도와주는 기능

메뉴 트리 - Info (I)



Item	Descriptions
Information	PRC와 PC-GUI의 기본 정보 표시
License Download	PRC의 License 정보 등록

B 좌측 (Default : Jog & Teaching)

- 2.18(p.48)에서 자세히 설명

C 중앙 (Default : Program Editor)

- 2.13(p.40)에서 자세히 설명

D 하단 (Default: Command)

- 2.16(p.42)에서 자세히 설명

E 우측 상단(Default: Program)

- 2.12(p.39)에서 자세히 설명

F 우측 하단(Default: Sub-Program)

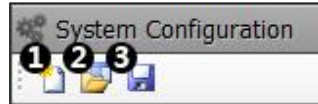
- 2.14(p.41)에서 자세히 설명

G 상태 표시 줄

- 로봇 컨트롤러와의 연결 상태 표시
- 연결 시 IP address, User Level 표시
- System, EtherCAT 통신, Robot 상태, TP 통신 상태 표시

2.4 System Configuration

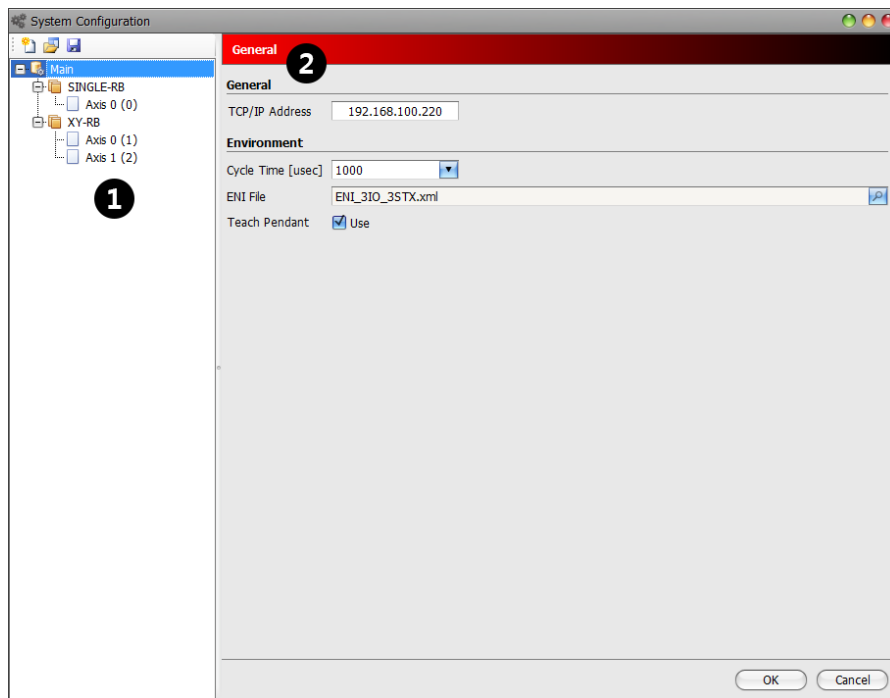
Configuration 저장/불러오기



- ① 새로운 Configuration을 생성할 때 사용합니다. 이전 Configuration은 사라집니다.
- ② 파일로 저장된 Configuration을 불러옵니다.
- ③ 현재 Configuration을 파일로 저장합니다.

로봇 컨트롤러 설정

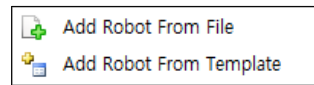
최상위 계층인 Main을 클릭하면 아래 화면이 나타납니다.



- ① System Parameter 항목, Robot 및 Axis List를 표시합니다.
- ② 설정 값을 표현하며, Administrator 권한으로 수정이 가능합니다.

로봇 추가

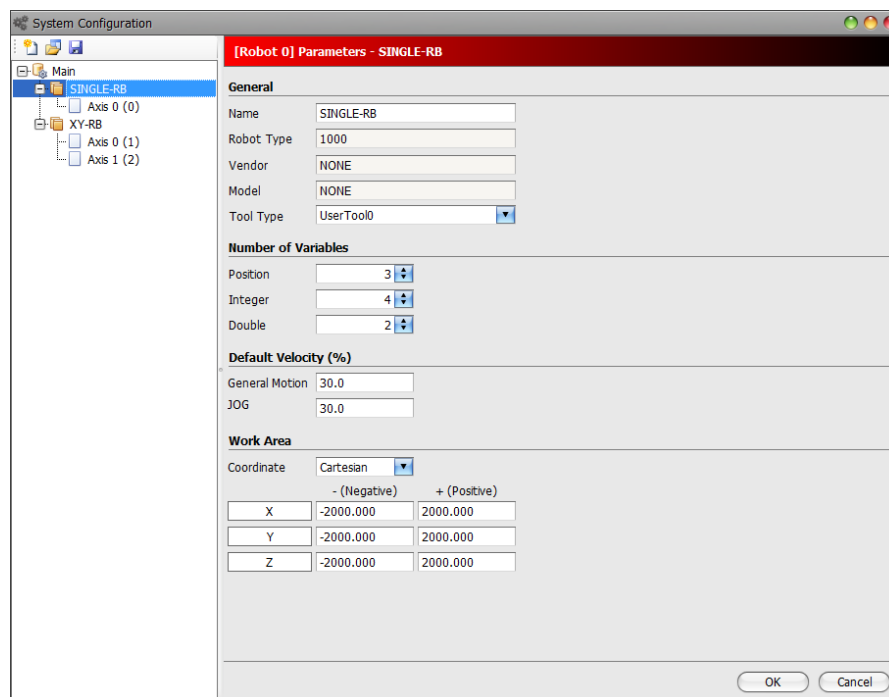
Administrator 권한으로 Main에 마우스 우 클릭하면 다음과 같은 Robot을 추가할 수 있는 창이 표시됩니다.



- Add Robot From Template: 제공되는 Template에서 Robot을 선택해서 추가할 경우
- Add Robot From File: 저장되어 있는 로봇 Parameter 파일을 불러와서 사용할 경우

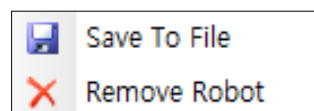
로봇 파라미터 설정

Robot의 이름을 누르면 일반 설정, 변수 설정, 속도, 동작 반경 등의 값을 설정할 수 있습니다.



Administrator 권한으로 Robot 이름에 마우스 우 클릭하면 다음과 같은 항목이 표시됩니다.

해당 로봇을 삭제하거나 설정 값을 파일로 저장할 수 있습니다.



로봇의 Axis Parameter 설정

아래 창을 통해 Axis에 관련된 Parameter들을 수정할 수 있습니다.

System Configuration

[Axis 0 (1)] Parameters

Axis Setting

Mapped Axis number: 1 * Assigned number on EtherCAT network

Motor Feedback

Pulse per revolution: 131072 Distance per revolution: 10.000

Gear Ratio: 1.0

Motor Direction: ☐ Inverse

Motion Completion

In-Position Range: 2 Setting Time [ms]: 5

Safety And Protection

Velocity Limit: 500.000 Brake Release Time [ms]: 10

Current Limit: 10.000 Negative Software Limit: -10000.000

Max. Position Error: 0.100 Positive Software Limit: 10000.000

Homing Parameters

Home Method: 1 * Homing method is depend on a drive

Velocity: 20.000 Return Velocity: 1.000

Acceleration: 100.000 Offset: 0.000

Ready Motion Parameters

Velocity: 5.000

Acceleration Time [ms]: 100.000 Acceleration Jerk [%]: 100.000

Deceleration Time [ms]: 100.000 Deceleration Jerk [%]: 100.000

Kill Time [ms]: 100.000 Ready Position: 0.000

Manual Motion Parameters

Joint

Velocity: 100.000

Acceleration Time [ms]: 100.000

Deceleration Time [ms]: 100.000

Acceleration Jerk [%]: 100.000

Deceleration Jerk [%]: 100.000

Kill Time [ms]: 100.000

Work

Velocity: 100.000

Acceleration Time [ms]: 100.000

Deceleration Time [ms]: 100.000

Acceleration Jerk [%]: 100.000

Deceleration Jerk [%]: 100.000

Kill Time [ms]: 100.000

Program Motion Parameters

Joint

Velocity: 500.000

Acceleration Time [ms]: 1000.000

Work

Velocity: 500.000

Acceleration Time [ms]: 1000.000

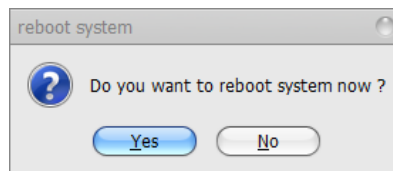
Copy Values from OK Cancel

2.5 Firmware Upgrade

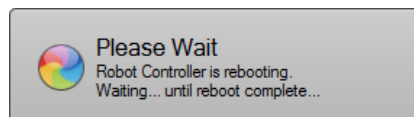
- I. Firmware Upgrade Menu 클릭
- II. Firmware Upgrade를 위한 Firmware 파일 선택
- III. 로봇 컨트롤러에 Firmware를 Upload 시도
- IV. 전송 성공 후 Reboot 진행 안내 메시지 표시
- V. 로봇 컨트롤러를 재 시작 시 Upload한 Firmware 적용 완료

2.6 Reboot Controller

- I. System -> Reboot Controller를 클릭



- II. 'Yes'를 클릭할 시 로봇 컨트롤러의 Reboot을 시도하며, Controller Reboot 완료 시 자동 재 접속 (최대 Reboot 시간은 60초로 설정)



- III. 재 접속 성공 후 Login 창이 나타나며, 다시 Login 필요

2.7 System Viewer

현재 System의 설정 상태를 보여주는 창입니다. 이 화면에서는 Parameter 설정값을 확인할 수 있지만 변경은 할 수 없습니다.

파라미터 항목 ←

Name	Value
RSystem	
Cycle Time	1000
ENI File	ENI_3IO_3STX.xml
Total Number O...	3
SINGLE-RB	
Model	NONE
Vendor	NONE
Default Velo...	30
Default JOG ...	30
Axis 0	
EtherCA...	0
Encoder ...	131072
Distance ...	10
Gear Ratio	1
Max. Vel...	500
Home Me...	1
Ready Po...	0
XY-RB	
Model	NONE
Vendor	NONE
Default Velo...	30
Default JOG ...	30
Axis 0	
EtherCA...	1
Encoder ...	131072
Distance ...	10
Gear Ratio	1
Max. Vel...	500
Home Me...	1
Ready Po...	0
Axis 1	
EtherCA...	2
Encoder ...	131072
Distance ...	10
Gear Ratio	1
Max. Vel...	500
Home Me...	1
Ready Po...	0

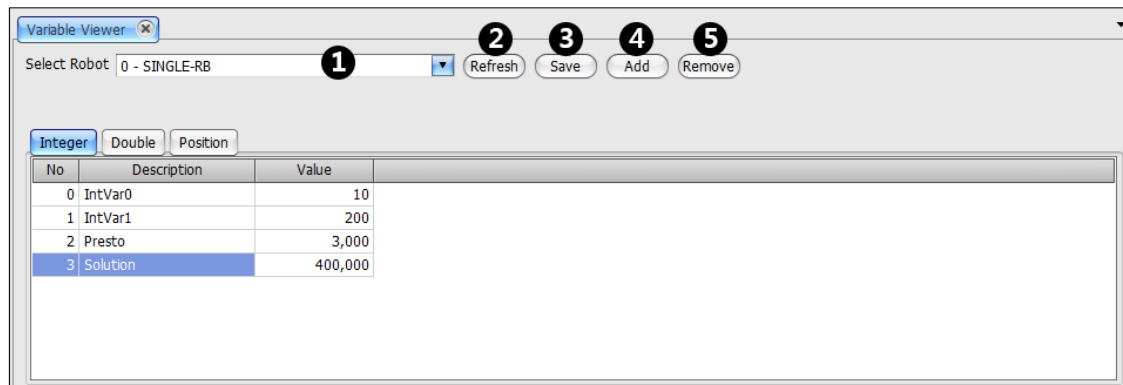
값 →

로봇의 개수, 로봇 타입, 로봇을 구성하는 항목 등의 파라미터를 한눈에 확인할 수 있습니다.

① 참고

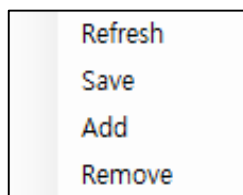
시스템 파라미터는 Administrator 권한으로 System Configuration에서 변경 가능합니다.

2.8 Variable



	Item	Description
①	Select Robot	로봇 선택
②	Refresh	선택된 로봇의 모든 Variable을 로봇 컨트롤러로부터 갱신
③	Save	선택된 로봇의 Variable 저장 (Save 해야 Reboot시 수정된 Variable 값이 적용)
④	Add	선택된 로봇의 선택된 Variable Type을 마지막 인덱스로 한 개 추가
⑤	Remove	선택된 로봇의 선택된 Variable Type의 마지막 인덱스를 삭제

- 마우스 우 클릭으로 Refresh, Save, Add, Remove 등의 기능을 수행



Variable Type

Integer

Integer		
No	Description	Value
0	IntVar0	10
1	IntVar1	200
2	Presto	3,000
3	Solution	400,000

- 위 창과 같이 Integer type 변수의 Description과 값을 표시
- Value를 클릭하여 선택해서 변경이 가능하며, Program에서 사용 가능
(Ex: Robot[0].IVar[0])

Double

Integer Double Position		
No	Description	Value
0	PI	3.142
1	DoubleVar1	1.234
2	GoodPosition	44.123

- 위 창과 같이 Double type 변수의 Description과 값을 표시
- Value를 클릭하여 선택해서 변경이 가능하며, Program에서 사용 가능
(Ex: Robot[0].DVar[0])

Position

Integer	Double	Position			
No	Description	Coord	X / J0	Y / J1	
0	Test	J	123.450	345.120	
	Test	W	123.450	345.120	
1	PosVar1	J	0.000	0.000	
	PosVar1	W	0.000	0.000	
2	Pos2	J	555.122	0.009	
	Pos2	W	555.122	0.009	
3	Pos	J	0.077	0.009	
	Pos	W	0.077	0.009	

- 위 창과 같이 Double Type Position 변수의 Description과 각 Axis Position 값을 표시
- W는 Work(World), J는 Joint 좌표를 의미
- 선택하고 싶은 변수의 좌표계와 축을 더블 클릭하여 변경이 가능, Program에서 사용 가능
(Ex: Robot[0].WkPos[0], Robot[0].JtPos[0])

No:	Description:	Coord:
0	Test	J
X / J0:	123.450	Y / J1:
	345.120	
<div>Update Cancel</div>		

- J, W 좌표계 중 하나를 변경하면 나머지 한 좌표계의 값은 해당 로봇의 키네마틱에 의해 계산되어 자동으로 변경됩니다.
- Get Current Position을 누르면 선택된 변수 인덱스의 값이 현재 위치 값으로 변경됩니다.

Refresh

Save

Add

Remove

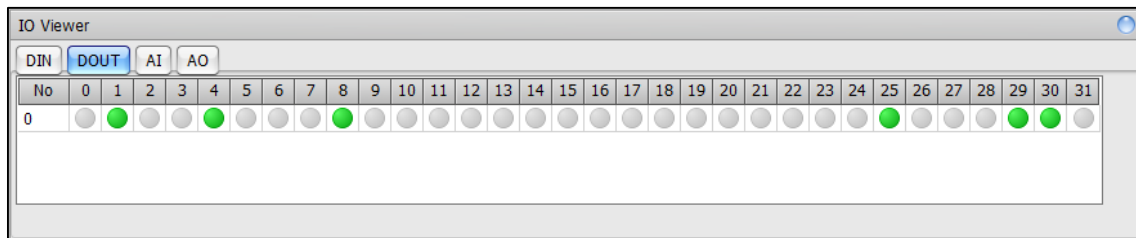
Get Current Position

2.9 IO

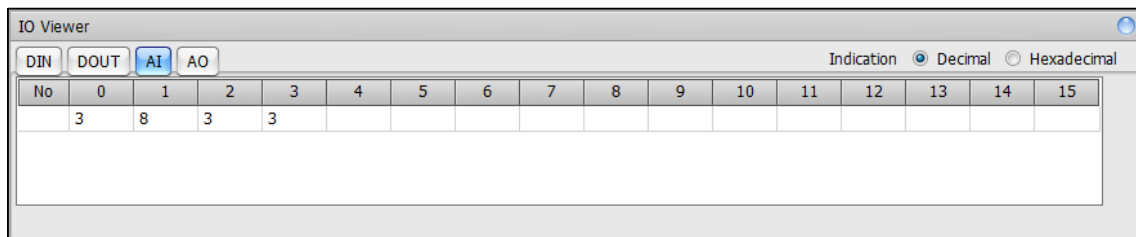
IO Mapping은 로봇 컨트롤러에서 Slave 순서대로 자동 설정됩니다.

- 참고** 자동으로 Mapping이 지원되는 IO List를 참조하시기 바랍니다.

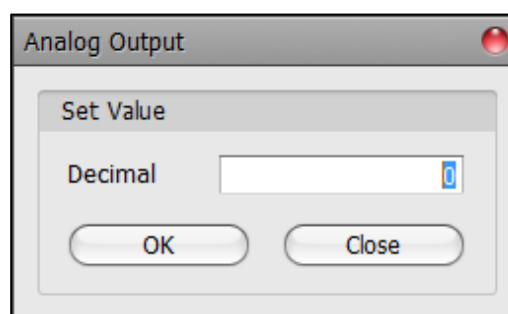
IO 값을 실시간으로 확인할 수 있습니다.



- DIN:** Digital Input의 현재 값을 읽어옵니다.
- DOUT:** Digital Output의 현재 값 표시, 마우스 더블 클릭 시 상태가 변경됩니다.



- AI:** Analog Input의 현재 값 표시. 우측상단 Dec, Hex Mode를 통해 10진, 16진으로 출력할 수 있습니다.



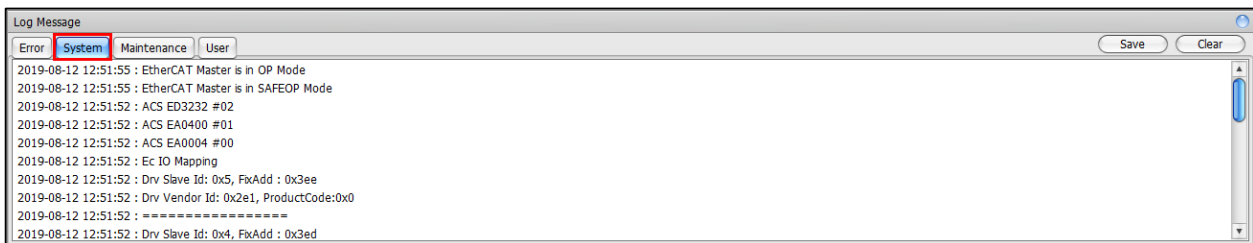
- AO:** Analog Output의 현재 값 표시. 변경하고자하는 값을 더블클릭하여 값을 변경합니다.

2.10 Log Message

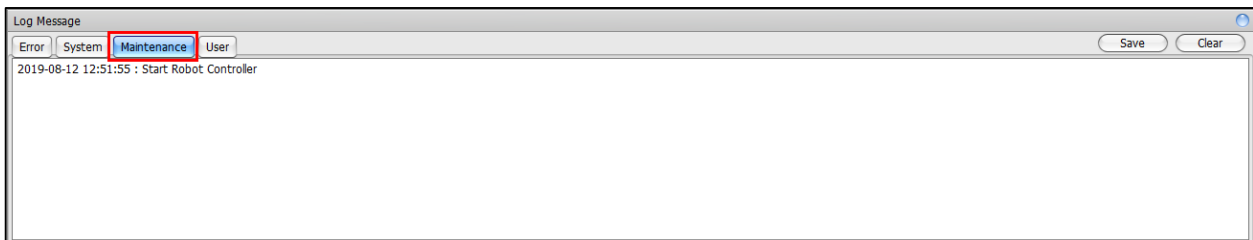
- **Save:** 선택되어 있는 Log Message를 파일로 저장합니다.
- **Clear:** 로봇 컨트롤러와 Message 창을 초기화합니다.
- **Error:** 로봇 컨트롤러에서 발생한 Error Message를 실시간으로 표시합니다.



- **System:** 로봇 컨트롤러에서 발생한 System Message를 실시간으로 표시합니다.



- **Maintenance:** 로봇 컨트롤러에서 발생한 Maintenance Message를 실시간으로 표시합니다.



- **User:** 로봇 컨트롤러에서 발생한 User Message를 실시간으로 표시합니다.



① 참고

User Log를 남기시려면 Print() 로봇 명령어를 참고하세요.

2.11 Position

모든 로봇을 설정 순서대로 각 Axis의 위치 값을 표시합니다.

No	Robot Name	Coord	J0	J1	J2	J3	J4	J5	J6	J7
0	SINGLE-RB	W	0.000							
		J	0.000							
1	XY-RB	W	0.094	0.092						
		J	0.094	0.092						

각 로봇의 Work , Joint 좌표 위치 값을 실시간으로 출력합니다.

2.12 Program Manager

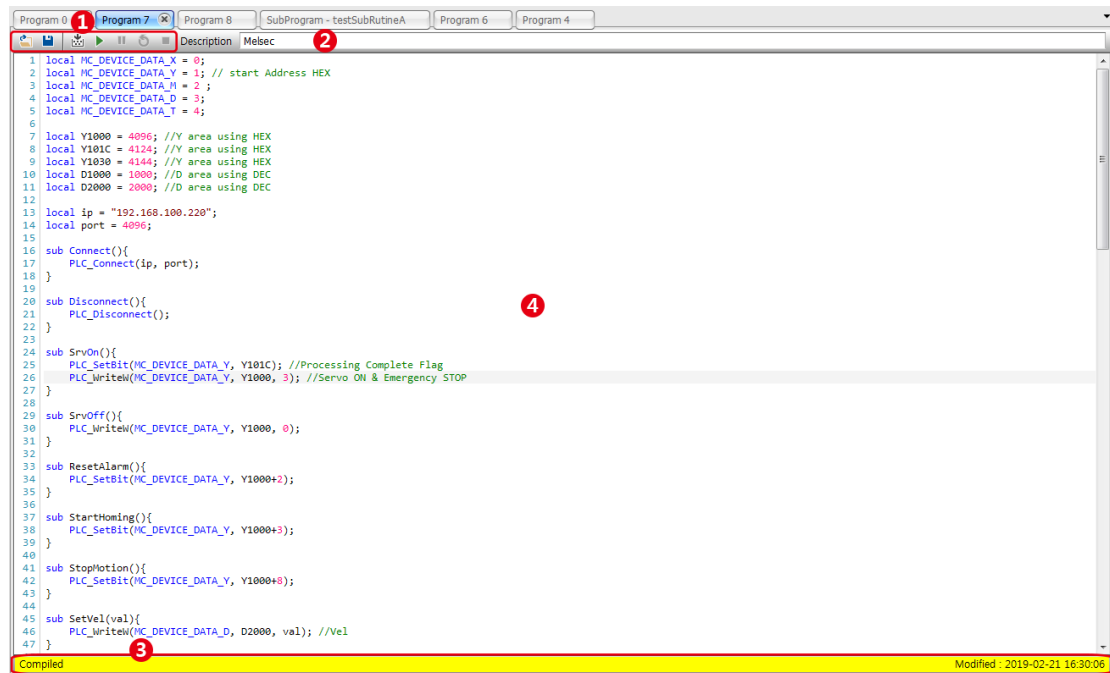
Program List를 실시간으로 실행 라인, 상태 등을 표시하여 줍니다.

Program					
A	No	Description	Lines	ExeLine	Status
<input type="checkbox"/>	0	Test1	14	0	Compiled
<input type="checkbox"/>	1		17	0	Compiled
<input type="checkbox"/>	2	TestIOWithoutB...	31	0	Compiled
<input type="checkbox"/>	3	ShareVarTest1	5	0	Compiled
<input type="checkbox"/>	4	ShareVarTest2	13	0	Compiled
<input type="checkbox"/>	5	SubAOut	8	0	Compiled
<input type="checkbox"/>	6	blank	0	0	Empty
<input type="checkbox"/>	7	Melsec	148	0	Compiled
<input type="checkbox"/>	8	Test1	25	0	Compiled
<input type="checkbox"/>	9	Test	9	0	Compiled

사용하고자 하는 프로그램을 더블 클릭 또는 마우스 우 클릭을 통해서도 해당 프로그램을 열 수 있습니다.

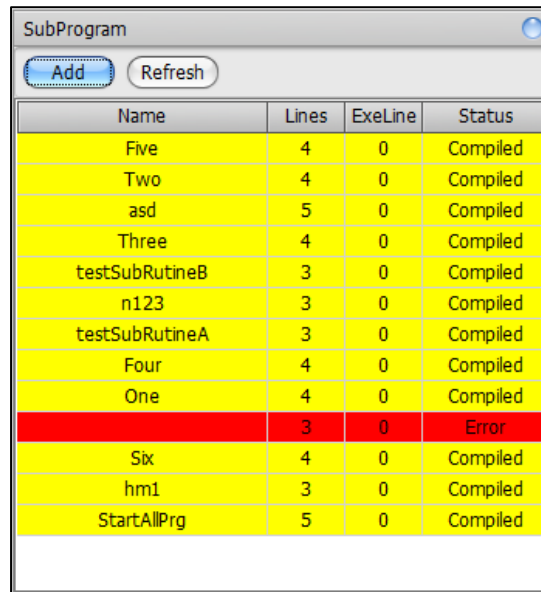
Open
Compile
Start
Stop
Pause
Resume
Close

2.13 Program Editor



Item	Description
① Menu	순서대로 Open, Save, Compile, Run, Pause, Resume, Stop 기능입니다.
② Description	해당 프로그램에 대한 Description 표시 및 수정 가능합니다.
③ 상태 표시 줄	현재 프로그램의 상태를 표시합니다.
④ Program Editor	<ul style="list-style-type: none"> 프로그램을 작성 및 수정할 수 있습니다. 좌측에 Line 번호가 표시되고, Program 실행 시 현재 실행되고 있는 Line Tracing 됩니다.

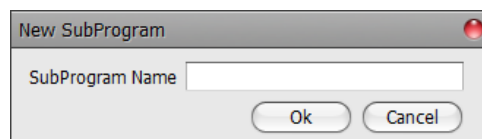
2.14 Sub Program Manager



Name	Lines	ExeLine	Status
Five	4	0	Compiled
Two	4	0	Compiled
asd	5	0	Compiled
Three	4	0	Compiled
testSubRoutineB	3	0	Compiled
n123	3	0	Compiled
testSubRoutineA	3	0	Compiled
Four	4	0	Compiled
One	4	0	Compiled
	3	0	Error
Six	4	0	Compiled
hm1	3	0	Compiled
StartAllPrg	5	0	Compiled

Add

- 새로운 Sub Program을 추가할 경우 사용하며, 클릭 시 이름을 입력하는 창이 나타납니다.
- 이름은 변수의 이름과 마찬가지로 숫자나 특수문자로 시작할 수 없으며, 영어로 시작해야 합니다.



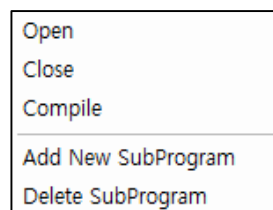
New SubProgram

SubProgram Name

Ok Cancel

Refresh

- Sub Program의 상태를 로봇 컨트롤러로부터 갱신합니다.
 - Compile 동작 시 상태가 자동으로 갱신됩니다.
- 현재 Sub Program 이름, 상태 등을 표시해 줍니다.
 - 수정 하고자 하는 Sub Program을 더블 클릭하면 파일을 열 수 있습니다.
 - 아래의 그림과 같이 Sub Program에 마우스 우 클릭 시 Open, Close, Compile, Add New SubProgram, Delete SubProgram 등의 기능을 사용할 수 있습니다.



Open

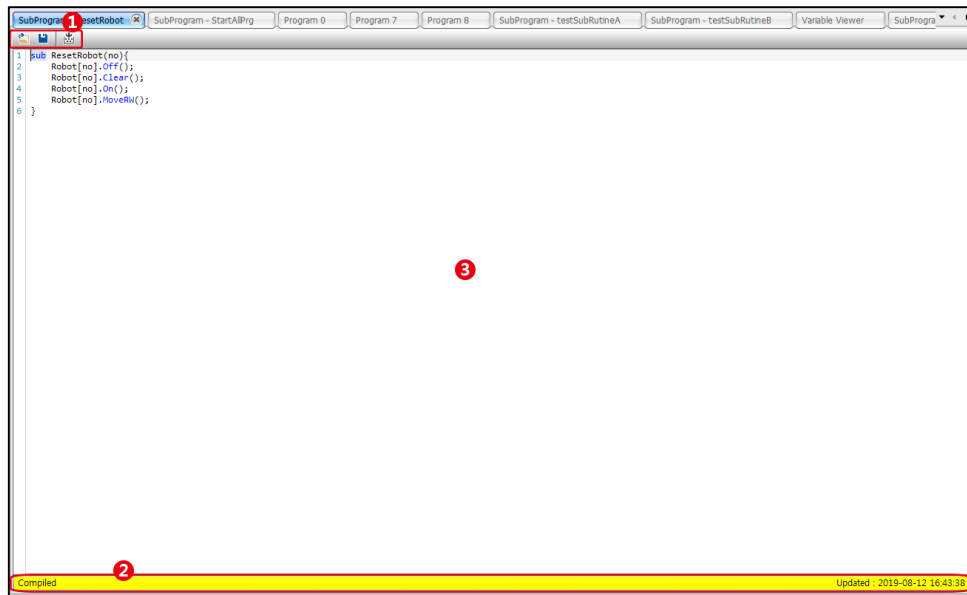
Close

Compile

Add New SubProgram

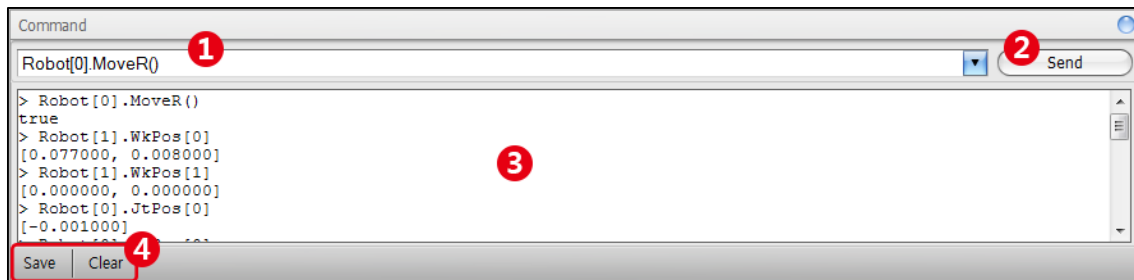
Delete SubProgram

2.15 Sub Program Editor



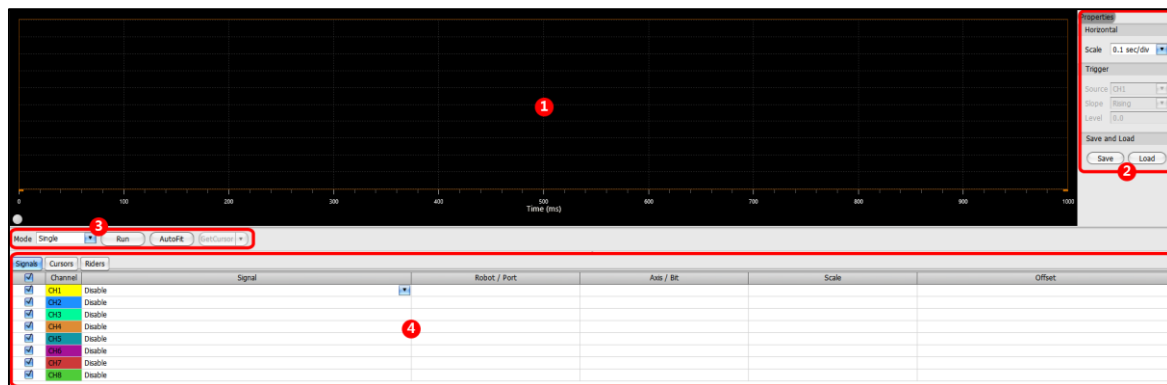
Item	Description
① Menu	순서대로 Open, Save, Compile 기능입니다.
② 상태 표시 줄	현재 서브 프로그램의 상태를 표시합니다.
③ Sub Program Editor	서브 프로그램을 작성 및 수정할 수 있는 창으로 좌측에는 Line 번호를 알 수 있습니다.

2.16 Command Terminal



Item	Description
①	로봇 컨트롤러에 Single Line Command를 입력하여 전달, 입력한 Command List는 자동으로 저장됩니다.
②	Send 클릭 시 '>' Prompt와 함께 입력된 Command가 표시됩니다.
③	Command 전달 후 응답 Message가 표시됩니다.
④	Save, Clear를 클릭하여, 입력 및 응답 로그를 파일로 저장하거나 삭제할 수 있습니다.

2.17 Scope



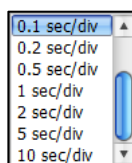
① Scope Display

- 선택된 채널의 해당하는 값들을 real-time으로 Display에 출력해줍니다.
(Disable 상태인 채널은 Display에 출력 안됨.)
- X축으로 마우스 이동 시 마우스 아이콘 변경되며, 이때 마우스 좌측 버튼을 이용하여 좌, 우로 드래그 시 X축의 좌우 이동이 가능합니다.
- Zooming
 - ✓ 마우스 휠을 위로 스크롤 시 Zoom In 됩니다.
 - ✓ 마우스 휠을 아래로 스크롤 시 Zoom Out 됩니다.
 - ✓ Ctrl키를 누른 상태로 마우스 휠 동작 시 모든 채널의 YScale의 조절이 가능합니다.
 - ✓ Shift키를 누른 상태로 마우스 휠 동작 시 모든 채널의 XScale의 조절이 가능합니다.
 - ✓ 마우스 좌측 버튼을 이용해 드래그 하여 사각형 영역을 만들면 해당 영역으로 Zoom In 됩니다.
- Panning : 그래프 위에 마우스 우측 버튼을 누르고 드래그 하여 이동 가능합니다.

② Properties

■ Horizontal

- ✓ Scale: 그래프의 X축 한 칸의 시간 표현 범위를 설정할 수 있습니다.



■ Trigger

- Source
 - Trigger를 적용할 Channel를 선택합니다.
 - 선택된 Channel의Signal이 Disable이면 Scope Run 안됩니다.
- Slope
 - Rising : 신호가 Rising할 때 Trigger됩니다.
 - Falling : 신호가 Falling할 때 Trigger됩니다.
- Level : Trigger 조건의 Level을 설정합니다.

■ Save and Load

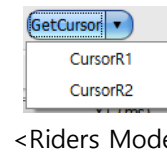
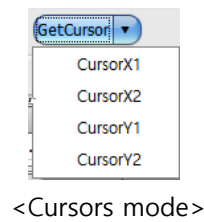
- ✓ Save: 현재 Display에 출력 되어있는 채널들의 정보를 저장합니다.
(Sampling Time, 신호의 종류, Index, SubIndex, YScale , Offset ,X의 최대 최소 , Y의 최대 최소)
- ✓ Load: 저장된 Scope 파일을 로드합니다. 해당 파일의 저장된 정보를 바탕으로 Display에 채널들을 출력합니다.

③ Scope Mode

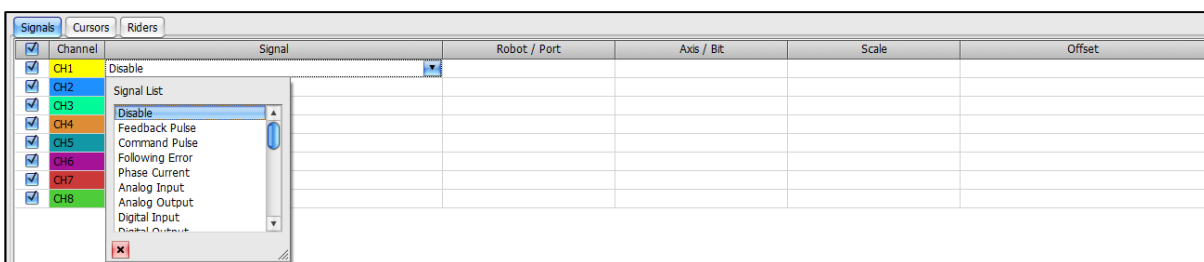


- Auto : 설정한 Scale(시간)마다 데이터를 갱신하여 Display에 출력하는 모드입니다.
- Single : 설정한 Scale(시간)만큼의 데이터를 수집한 후 Display에 출력하는 모드입니다.
- Trigger : Trigger 조건을 만족하는 시점부터 한Scale(시간)만큼의 데이터를 수집한 후 Display에 해당 데이터들 출력하는 모드입니다.(다음 Trigger 조건을 만족하는 한Scale(시간)만큼의 데이터가 수집되면 데이터 갱신됨)
- Continue : 데이터를 Display에 real-time으로 출력하는 모드입니다.
- **Run** : 선택한 신호, 로봇, 축의 데이터를 받기 시작하고 선택한 신호에 맞는 파형을 그립니다.
- **AutoFit** : 모든 채널이 한 화면에 보일 수 있도록 채널 별 스케일이 자동으로 조정됩니다.

- GetCursor** : ▼ 클릭 시 체크박스 출력됩니다. 해당 체크박스에서 Cursor를 선택한 후 GetCursor 버튼을 누르면 선택된 해당 Cursor가 Display안으로 들어옵니다.
 해당 버튼은 Cursors, Riders 모드에서만 활성화되고 Run 상태일 때 사용할 수 없습니다.



④ Scope Settings



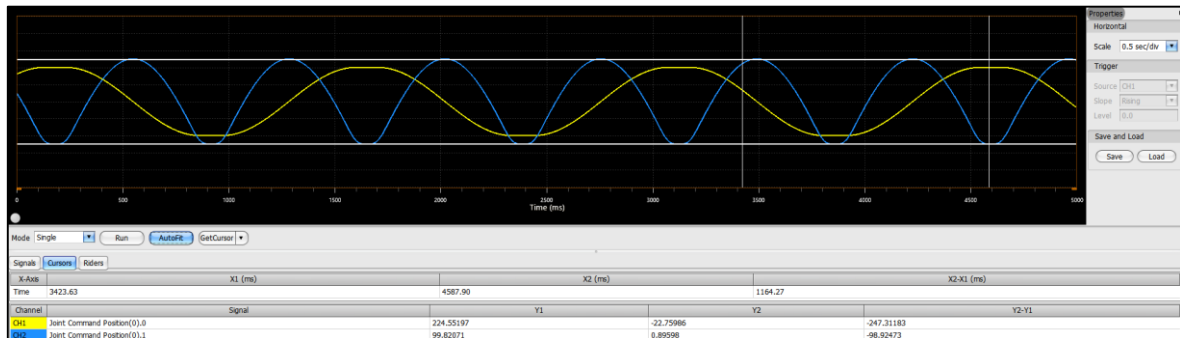
- 최대 8 Channel까지 지원하며, 한 채널 당 하나의 신호를 선택할 수 있습니다.
- Signals** : Scope에 나타낼 신호, 로봇, 축, 스케일, 오프셋 등을 설정하는 창입니다.

총 24개의 신호를 받아볼 수 있습니다.

- ✓ Disable – 아무 신호도 받지 않습니다. (Default)
- ✓ Feedback Pulse – 선택한 로봇, 축에 대하여 모터가 실제 수행한 펄스 값을 신호로 받습니다.
- ✓ Command Pulse – 선택한 로봇, 축에 대하여 사용자가 명령한 펄스 값을 신호로 받습니다.
- ✓ Following Error – 선택한 로봇, 축에 대하여 사용자가 명령한 위치 값과 모터가 실제 수행하는 위치 값의 차이를 신호로 받습니다. (신호의 진폭이 클수록 성능이 떨어지는 것)
- ✓ Phase Current – 선택한 로봇, 축에 대하여 상 전류 값을 신호로 받습니다.
- ✓ Analog Input – 선택한 인덱스의 아날로그 입력 값을 신호로 받습니다.
- ✓ Analog Output – 선택한 인덱스의 아날로그 출력 값을 신호로 받습니다.
- ✓ Digital Input – 선택한 포트, 비트의 디지털 입력 값을 신호로 받습니다.
- ✓ Digital Output – 선택한 포트, 비트의 디지털 출력 값을 신호로 받습니다.
- ✓ Joint Command Position – 선택한 로봇, 축에 대하여 사용자가 명령한 각도 값을 신호로 받습니다.
- ✓ Joint Command Velocity – 선택한 로봇, 축에 대하여 사용자가 명령한 속도 값을 신호로 받습니다.
- ✓ Joint Command Acceleration – 선택한 로봇, 축에 대하여 사용자가 명령한 가속도 값을 신호로 받습니다.
- ✓ Joint Command Jerk – 선택한 로봇, 축에 대하여 사용자가 명령한 가가속도 값을 신호로 받습니다.
- ✓ Joint Feedback Position – 선택한 로봇, 축에 대하여 모터가 실제 수행한 각도 값을 신호로 받습니다.

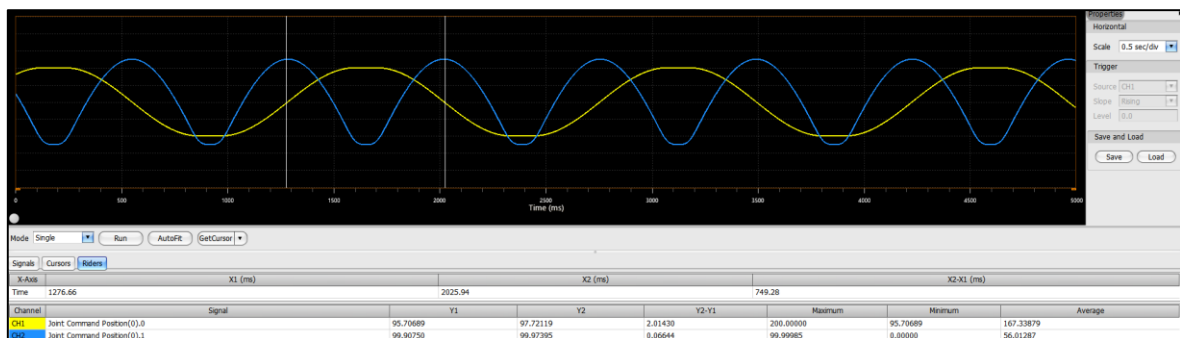
- ✓ Joint Feedback Velocity – 선택한 로봇, 축에 대하여 모터가 실제 수행한 속도 값을 신호로 받습니다.
- ✓ Joint Feedback Acceleration – 선택한 로봇, 축에 대하여 모터가 실제 수행한 가속도 값을 신호로 받습니다.
- ✓ Joint Feedback Jerk – 선택한 로봇, 축에 대하여 모터가 실제 수행한 가가속도 값을 신호로 받습니다.
- ✓ Work Command Position – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 사용자가 명령한 좌표 값을 신호로 받습니다.
- ✓ Work Command Velocity – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 사용자가 명령한 속도 값을 신호로 받습니다.
- ✓ Work Command Acceleration – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 사용자가 명령한 가속도 값을 신호로 받습니다.
- ✓ Work Command Jerk – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 사용자가 명령한 가가속도 값을 신호로 받습니다.
- ✓ Work Feedback Position – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 모터가 실제 수행한 좌표 값을 신호로 받습니다.
- ✓ Work Feedback Velocity – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 모터가 실제 수행한 속도 값을 신호로 받습니다.
- ✓ Work Feedback Acceleration – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 모터가 실제 수행한 가속도 값을 신호로 받습니다.
- ✓ Work Feedback Jerk – 선택한 로봇, 축에 대하여 World 좌표계를 기준으로 모터가 실제 수행한 가가속도 값을 신호로 받습니다.

■ Cursors



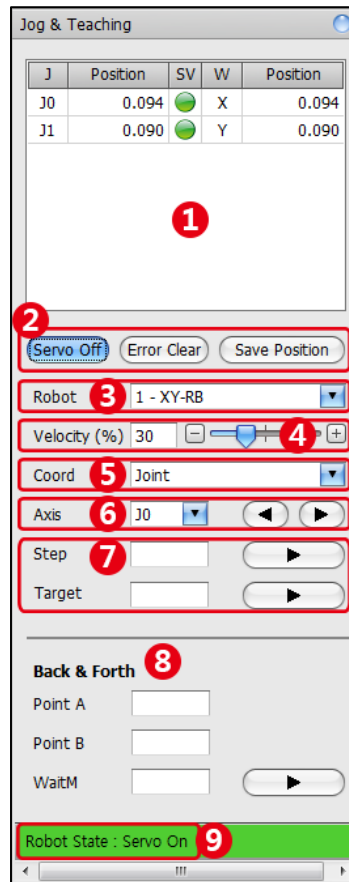
- ✓ X1, X2(흰색 세로 얇은 선)을 이용하여 X축(시간)의 값을 알 수 있습니다.
- ✓ Y1, Y2(흰색 가로 굵은 선)을 이용하여 각 신호의 값을 알 수 있습니다.
(파형의 진폭, Peak to Peak 등)
- ✓ 미세 조정하고 싶은 Cursor를 마우스로 클릭 후 NumPad 4, 6을 사용하여 좌우로 선택된 Cursor를 이동시킬 수 있습니다,
- ✓ Continue 모드로 Scope Running중일 때 해당 X1, X2, Y1, Y2 Cursor 사용 불가입니다.

■ Riders : 수직선과 신호가 교차되는 지점(Y)의 데이터 값을 표시합니다.



- **Y1** : 1번 수직선과 신호가 만나는 지점의 데이터를 채널 별로 표시합니다.
- **Y2** : 2번 수직선과 신호가 만나는 지점의 데이터를 채널 별로 표시합니다.
- **Y2-Y1** : Y1과 Y2의 차이 값을 채널 별로 표시합니다.
- **Maximum** : 1번 수직선과 2번 수직선 사이에서 가장 큰 데이터 값을 채널 별로 표시합니다.
- **Minimum** : 1번 수직선과 2번 수직선 사이에서 가장 작은 데이터 값을 채널 별로 표시합니다.
- **Average** : 1번 수직선과 2번 수직선 사이의 평균 데이터 값을 채널 별로 표시합니다.
- 미세 조정하고 싶은 Cursor를 마우스로 클릭 후 NumPad 4, 6을 사용하여 좌우로 선택된 Cursor를 이동시킬 수 있습니다,
- Continue 모드로 Scope Running중일 때 Y1, Y2 커서 사용 불가합니다.

2.18 Jog & Teaching



Item	Description
① Robot Monitor	선택한 로봇의 모든 축 상태와 위치 값을 알 수 있습니다.
② Robot Control	선택한 로봇의 Servo On/Off, Error Clear, Position 변수를 관리할 수 있습니다.
③ Robot Select	제어할 로봇을 선택합니다. 인덱스는 0번부터 시작합니다.
④ Velocity	선택한 로봇의 조그 모션 수행 속도를 조절합니다. (1~100%)
⑤ Coord	Work 혹은 Joint 좌표계 중 사용할 좌표계를 선택합니다.
⑥ Axis	조그 모션을 수행할 축을 선택합니다. 인덱스는 0번부터 시작합니다.
⑦ Step & Target	지정한 Step(상대적), Target(절대적)의 위치로 이동합니다.
⑧ Back & Forth	Point A, Point B 사이를 WaitM만큼의 대기시간(ms)을 두고 반복하여 움직입니다.
⑨ State	로봇의 현재 상태를 표시합니다.

Parameter

The screenshot shows the 'Motion Parameter' dialog box. It has a title bar with standard window controls. Below the title bar, there are several controls: a 'Robot' dropdown menu (callout 1) showing '1 - XY-RB', a 'Profile' dropdown menu (callout 2) showing 'Time Base', a 'Motion Type' dropdown menu (callout 3) showing 'Jog', a 'Refresh' button (callout 4), and a 'Save to Flash' button (callout 5). Below these controls, there are two sections. The first section is labeled 'Joint' (callout 6) and contains a table with 7 columns: Axis, Vel, Acc Time, Dec Time, Acc Jerk, Dec Jerk, and Kill Time. The table has two rows: J0 and J1. The second section is labeled 'Work' (callout 6) and contains a similar table with 7 columns: Axis, Vel, Acc Time, Dec Time, Acc Jerk, Dec Jerk, and Kill Time. The table has two rows: X and Y. Each table has a 'Get' and a 'Set' button next to it.

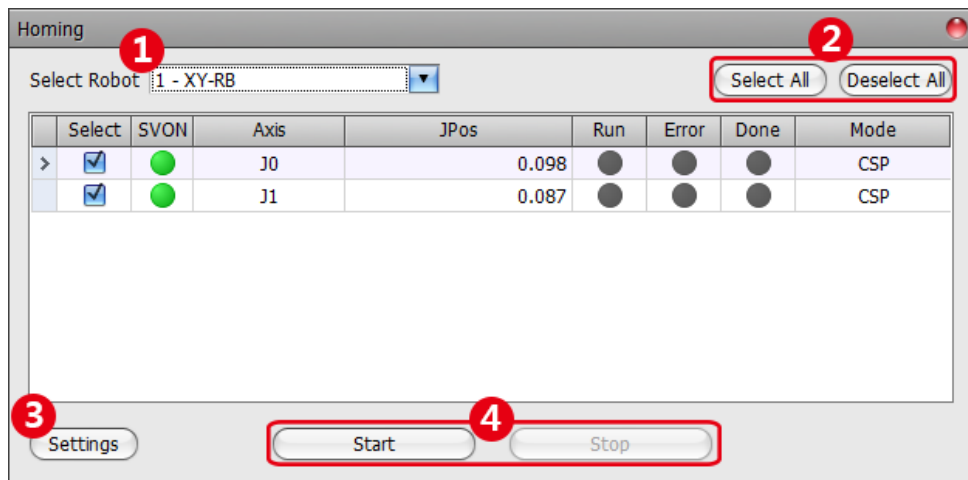
Axis	Vel	Acc Time	Dec Time	Acc Jerk	Dec Jerk	Kill Time
J0	100	100.000	100.000	100.000	100.000	100.000
J1	100	100.000	100.000	100.000	100.000	100.000

Axis	Vel	Acc Time	Dec Time	Acc Jerk	Dec Jerk	Kill Time
X	100	100.000	100.000	100.000	100.000	100.000
Y	100	100.000	100.000	100.000	100.000	100.000

Item	Description
① Robot	확인하고 수정할 로봇을 선택합니다.
② Motion Type	Motion Type마다 파라미터 값을 설정할 수 있습니다. Motion Type은 Jog, Program, Ready 총 세 가지로 구분됩니다.
③ Refresh	수정한 값을 무시하고 기존에 저장된 모든 파라미터 값들로 새로고침합니다.
④ Save to Flash	Set된 파라미터 값을 Flash로 저장합니다. (Reboot 후에도 유지)
⑤ Joint Get & Set	<ul style="list-style-type: none"> Set 클릭 시 Joint Parameter가 수정한 값으로 저장됩니다. Get 클릭 시 Joint Parameter를 불러옵니다.
⑥ Work Get & Set	<ul style="list-style-type: none"> Set 클릭 시 Work Parameter가 수정한 값으로 저장됩니다. Get 클릭 시 Work Parameter를 불러옵니다.

2.19 Homing

- Robot을 선택하고, 해당 로봇에서 Homing을 수행하고자 하는 축을 선택하여 동작시킵니다.

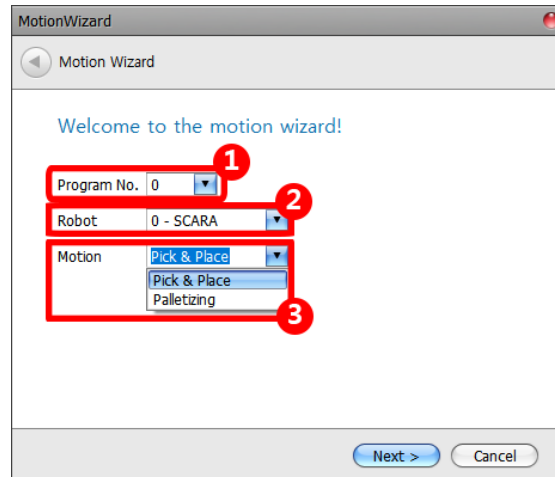


Item	Description
① Select Robot	Homing할 로봇을 선택합니다.
② (De)Select All	선택된 로봇의 모든 축을 선택(해제)합니다.
③ Settings	<div data-bbox="738 1014 1112 1393" data-label="Image"> </div> <p>축을 선택하고 필요한 Parameter를 수정하여 Apply, OK 버튼을 눌러서 적용합니다.</p> <ul style="list-style-type: none"> Method: 원점 복귀 방식을 지정하며, -128 ~ 127번까지 지정할 수 있고, 이 번호는 드라이브 제조사에 따라 다릅니다. 지원 및 테스트된 모터 리스트를 확인하시기 바랍니다. Velocity: 최초 Limit, Home 등의 위치로 이동할 때의 속도 Return Vel: 최초 위치를 만난 뒤, Index (Z상) 위치로 이동할 때의 속도 Acceleration: 원점 복귀 시의 가속도 Offset: 기본 원점 복귀가 완료되고, 원하는 원점 위치로 이동을 위한 위치 값
④ Start, Stop	<p>원점 복귀를 수행하거나 멈춥니다. (이때, Robot의 모든 축은 반드시 ON 상태)</p> <ul style="list-style-type: none"> 수행할 축을 선택하고, Start 버튼을 눌러서 동작을 시작합니다.

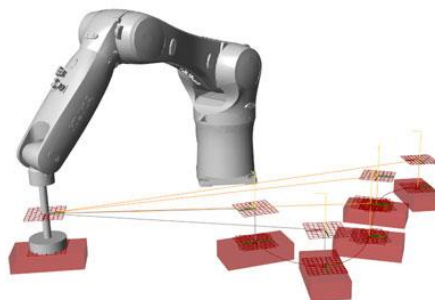
2.20 Motion Wizard

여러가지 모션을 좌표 입력만으로 손쉽게 구현할 수 있는 모션 위자드입니다.

- I. 먼저, 프로그램 번호와 로봇, 수행할 모션을 선택한 후 Next를 누릅니다.



- ① **Program No** : 자동 생성된 모션 프로그램을 작성할 프로그램 번호를 지정합니다.
- ② **Robot** : 해당 모션을 수행할 로봇을 지정합니다.
- ③ **Motion** : 수행하고자하는 모션을 지정합니다. 현재는 Pick and Place, Palletizing 두 가지 모션만 지원합니다.
 - *Pick and Place* : 한 장소에서 물건을 집어 들어 다른 장소에 놓는 모션



(출처: Premier Tech)

- *Palletizing* : Pallet에 특정한 배열 방식을 가지고 물건을 쌓아올리는 모션



(출처: Robots in Architecture)

II. 수행할 모션에 대한 인자를 올바르게 모두 입력한 후 Next를 누릅니다.

A. Pick & Place

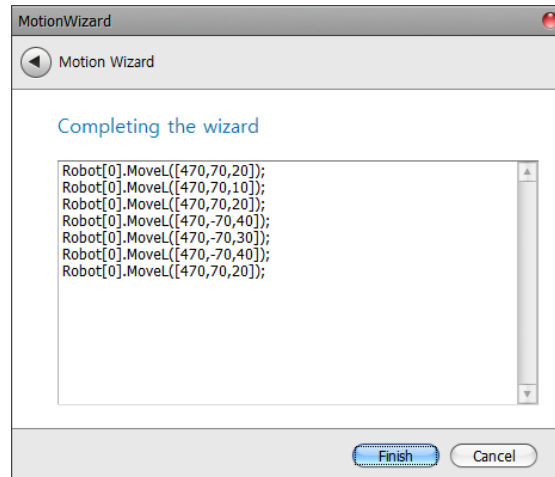
- Point A : 집어 들 물건의 위치 좌표
- Point B : 물건을 내려놓을 위치 좌표
 - ✓ X : 집어 들거나 내려놓을 물건의 X 좌표값
 - ✓ Y : 집어 들거나 내려놓을 물건의 Y 좌표값
 - ✓ Z : 물건을 집어 들거나 내려놓을 위치의 Z축 좌표 값
- Height : 물건을 집어 들거나 내려 놓을 때 움직이는 높이 값 (Z축)
- MoveC : 체크 상태면 아치 모션, 체크 해제 상태면 직선 보간을 합니다.
아치 모션을 하기 위해서는 Arch Height를 입력해야합니다.
 - ✓ Arch Height : 물건을 집고 Height만큼 올라간 높이에서 다음 지점으로 원형 보간을 하기 위한 추가 높이 값 (Z축)

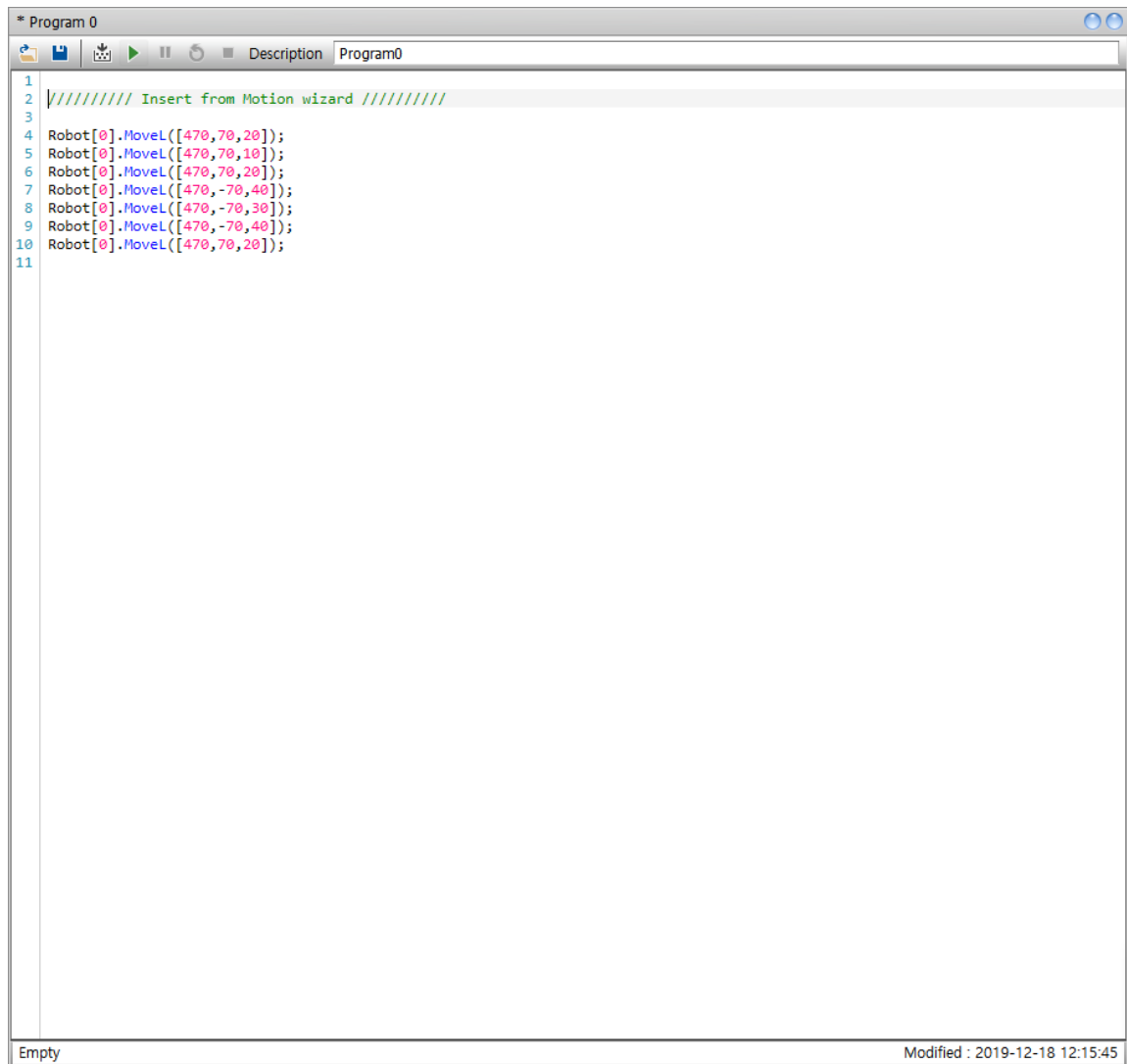
MoveC ☒
Arch Height:

B. Palletizing

(추가 예정)

III. 자동으로 생성된 모션 프로그램을 확인합니다.



IV. Finish를 눌러 선택한 프로그램에 해당 모션을 작성합니다.

The screenshot displays the PRC Manager software interface. At the top, there is a menu bar with options like 'File', 'Edit', 'View', 'Tools', 'Help', and a 'Description' tab. Below the menu bar is a toolbar with icons for file operations and execution. The main area is a text editor window titled 'Program 0' containing the following code:

```
1  
2 |////////// Insert from Motion wizard //////////  
3  
4 Robot[0].MoveL([470,70,20]);  
5 Robot[0].MoveL([470,70,10]);  
6 Robot[0].MoveL([470,70,20]);  
7 Robot[0].MoveL([470,-70,40]);  
8 Robot[0].MoveL([470,-70,30]);  
9 Robot[0].MoveL([470,-70,40]);  
10 Robot[0].MoveL([470,70,20]);  
11
```

At the bottom of the window, there is a status bar showing 'Empty' on the left and 'Modified : 2019-12-18 12:15:45' on the right.

Chapter 3. 로봇 구동

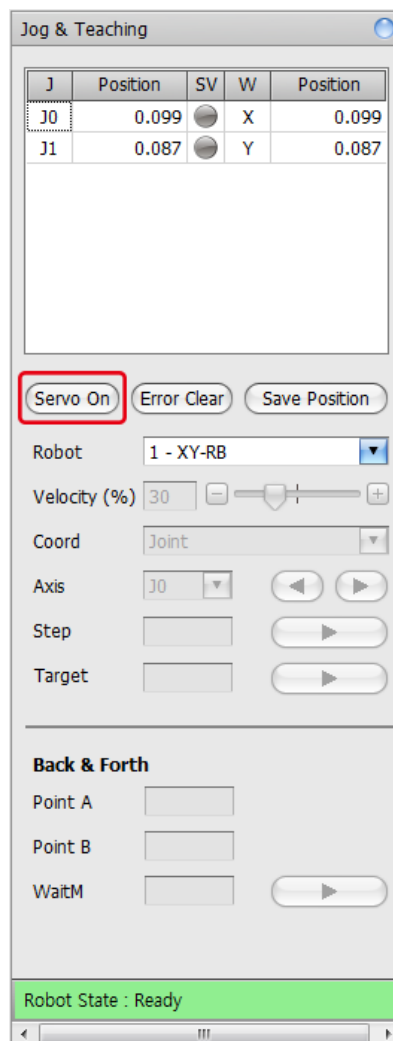
메뉴 및 구동 모드에 따라 Manual, Program (Auto) 및 Homing 모드로 구분되고, 이동 축 방향에 따라 Joint, Work 및 Ready Motion으로 구분됩니다.

- Joint (단축): 축(Axis) 별로 제어
- Work: World 좌표(직교좌표의 경우 X, Y, Z) 기준으로 로봇 제어
- Ready: System Configuration or Motion Parameter에서 설정되어 있는 Ready 위치로 이동

3.1 Manual Motion

프로그램을 실행하면 Jog&Teaching창이 기본으로 좌측에 위치합니다.

닫았거나 창이 보이지 않을 경우 상단 메뉴 트리 Motion -> Jog&Teaching으로 다시 열 수 있습니다.

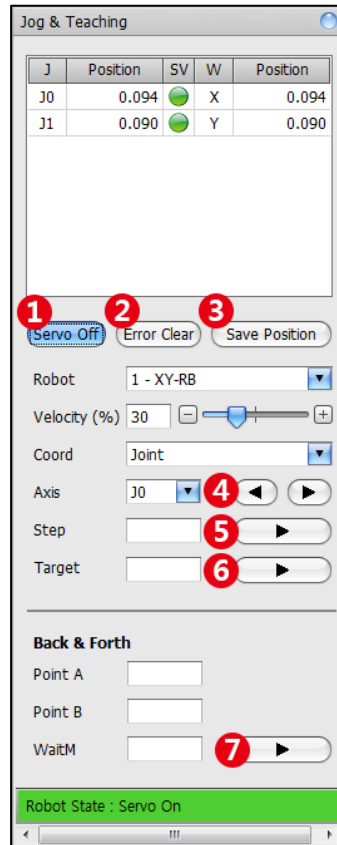


- Servo On 버튼을 클릭하여 선택된 로봇의 모든 축을 On(구동 가능 상태) 할 수 있습니다.

- 아래 그림과 같이 SV 부분이 빨간색이면 Drive Alarm이 발생한 상태입니다. 이 경우, Error Clear 버튼을 클릭해서 Alarm을 Clear 한 후 Servo On 버튼을 클릭합니다.

J	Position	SV	W	Position
J0	40.603		X	40.603
J1	35.082		Y	35.082

Jog & Teaching 기능 설명



Item	Description
① Servo On/Off	선택된 로봇의 모든 축을 On/Off 합니다.
② Error Clear	선택된 로봇의 Drive Alarm을 Clear 합니다.
③ Save Position	Position 변수 관리 창이 열립니다. (View-Variable-Position과 동일)
④ Axis ◀ ▶	좌표계와 축을 선택한 후 해당 버튼을 누르면 + 혹은 - 방향으로 움직입니다.
⑤ Step ▶	좌표계와 축을 선택한 후 해당 버튼을 누르면 현재 위치에서 상대적 위치로 이동합니다. 역방향으로 이동하려면 음수를 입력합니다.
⑥ Target ▶	좌표계와 축을 선택한 후 해당 버튼을 누르면 현재 위치에서 절대적 위치로 이동합니다. 역방향으로 이동하려면 음수를 입력합니다.
⑦ Back & Forth ▶	입력된 값을 이용하여 Back & Forth 동작을 수행합니다.

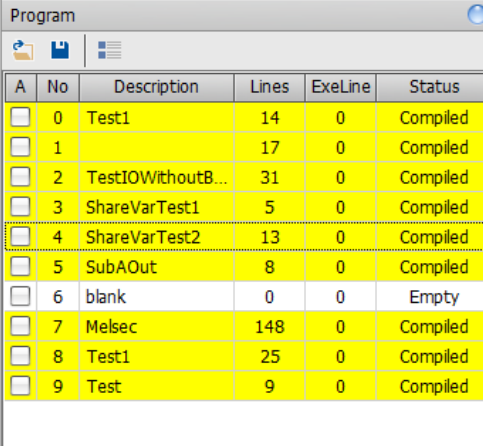
① 참고

Motion Parameter (속도, 가감속 시간 등)의 수정은 상단 메뉴 Motion - Parameters 창에서 가능합니다.

3.2 Programming Motion

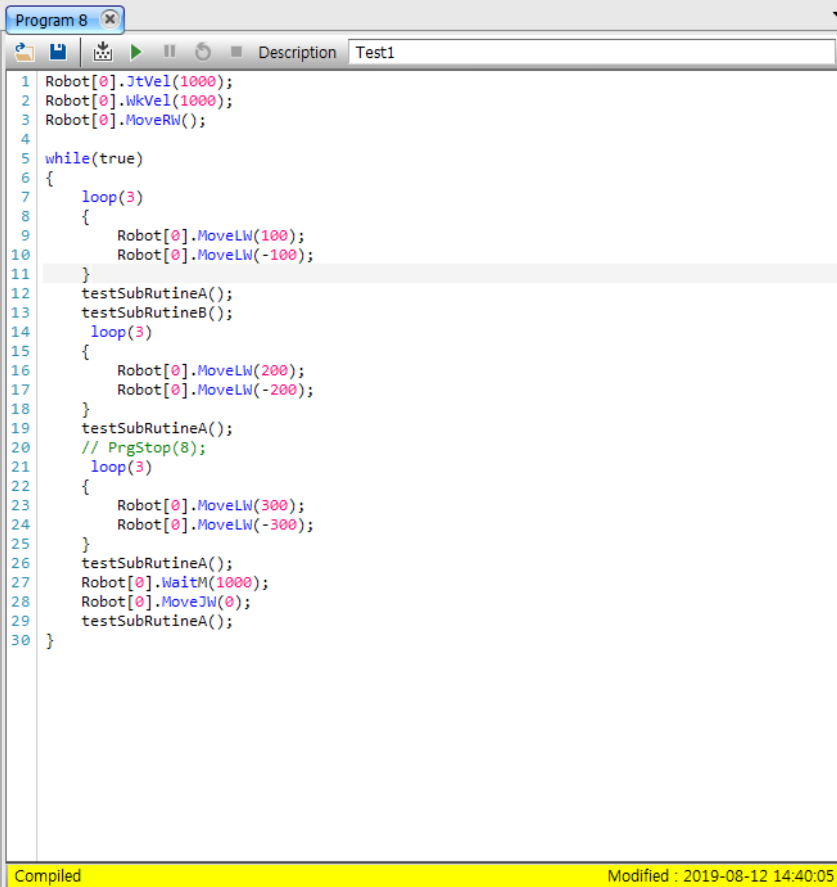
프로그램을 실행하면 우측에 기본으로 위치합니다.

닫았거나 보이지 않을 경우 상단 메뉴 트리 Tool -> Program으로 다시 열 수 있습니다.



A	No	Description	Lines	ExeLine	Status
<input type="checkbox"/>	0	Test1	14	0	Compiled
<input type="checkbox"/>	1		17	0	Compiled
<input type="checkbox"/>	2	TestIOWithoutB...	31	0	Compiled
<input type="checkbox"/>	3	ShareVarTest1	5	0	Compiled
<input type="checkbox"/>	4	ShareVarTest2	13	0	Compiled
<input type="checkbox"/>	5	SubAOut	8	0	Compiled
<input type="checkbox"/>	6	blank	0	0	Empty
<input type="checkbox"/>	7	Melsec	148	0	Compiled
<input type="checkbox"/>	8	Test1	25	0	Compiled
<input type="checkbox"/>	9	Test	9	0	Compiled

Program Editor



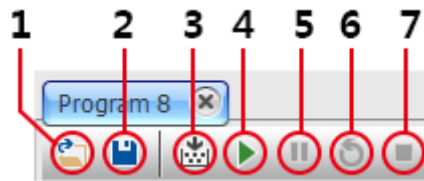
```

1 Robot[0].JtVel(1000);
2 Robot[0].WkVel(1000);
3 Robot[0].MoveRw();
4
5 while(true)
6 {
7     loop(3)
8     {
9         Robot[0].MoveLW(100);
10        Robot[0].MoveLW(-100);
11    }
12    testSubRoutineA();
13    testSubRoutineB();
14    loop(3)
15    {
16        Robot[0].MoveLW(200);
17        Robot[0].MoveLW(-200);
18    }
19    testSubRoutineA();
20    // PrgStop(8);
21    loop(3)
22    {
23        Robot[0].MoveLW(300);
24        Robot[0].MoveLW(-300);
25    }
26    testSubRoutineA();
27    Robot[0].WaitM(1000);
28    Robot[0].MoveJW(0);
29    testSubRoutineA();
30 }
  
```

Compiled Modified : 2019-08-12 14:40:05

- 10 개의 Program Task 중에 하나를 선택하여 엽니다. (더블클릭 혹은 마우스 우 클릭 후 Open)
- 프로그램을 작성하고 컴파일 후 프로그램을 실행합니다.

Program Editor 상단 버튼



Item	Description
① Open	저장되어 있는 prg 확장자를 가진 프로그램을 불러옵니다.
② Save	프로그램을 prg 확장자를 가진 파일로 원하는 경로에 저장합니다.
③ Compile	작성한 프로그램으로 컴파일을 수행합니다.
④ Start	컴파일 후 프로그램을 실행합니다.
⑤ Pause	실행 중인 프로그램을 잠시 정지합니다.
⑥ Resume	잠시 정지한 프로그램 라인부터 다시 실행합니다.
⑦ Stop	프로그램을 완전히 정지합니다.

① 참고

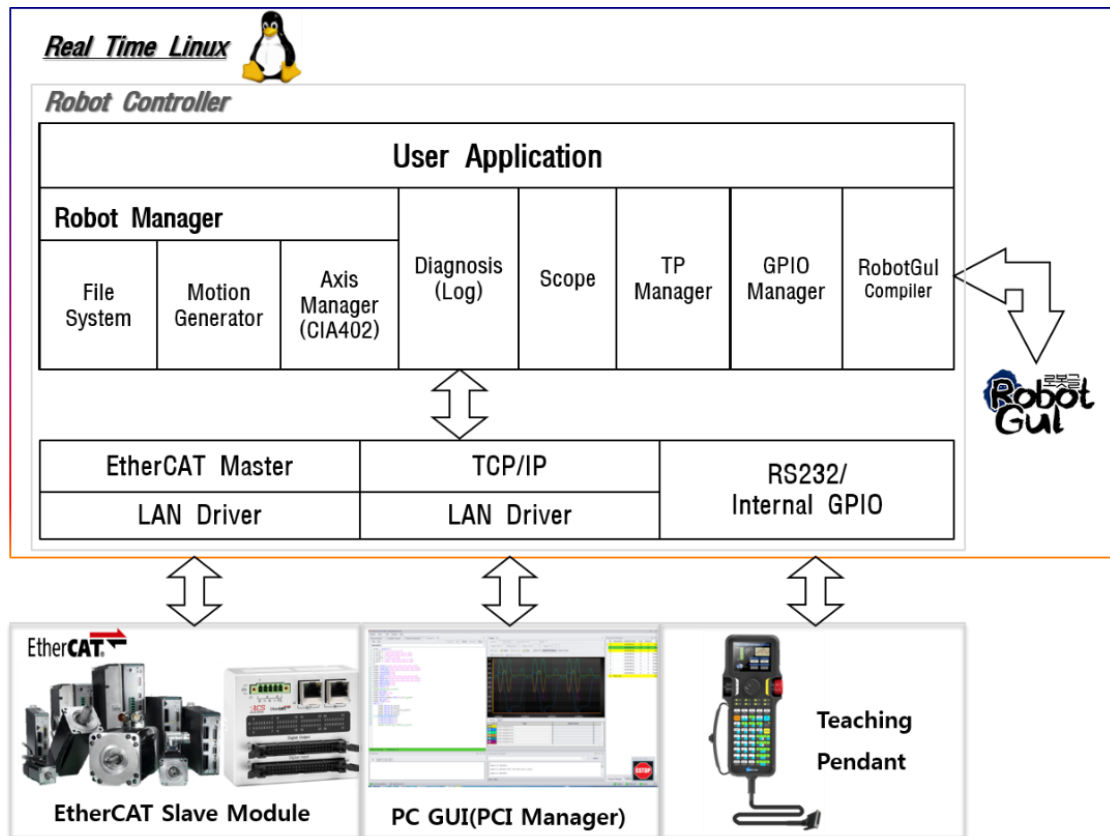
프로그램 문법 및 명령어에 대해서는 프로그램 가이드 자료를 참조하시기 바랍니다.

Chapter 4. 로봇제어기 소프트웨어

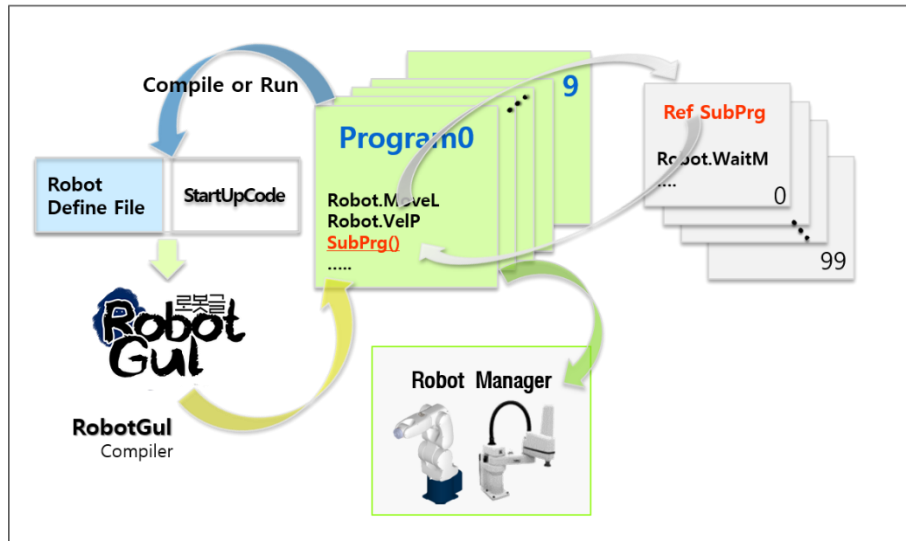
4.1 소프트웨어 아키텍처

아래 그림과 같이 프레스토 로봇 제어기(PRC)는 자체 제작한 RobotGul(로봇글)을 프로그래밍 언어로 사용하고 있습니다.

로봇글은 C언어와 유사하여 이를 학습한 경험이 있는 사용자에게 접근성이 좋습니다.



4.2 프로그램 아키텍처



- 10개의 독립적인 Program Task를 지원합니다.
- 프로그램에서 함수 호출과 같이 사용할 수 있는 Sub Program을 최대 100개까지 지원합니다.

Chapter 5. Robot Motion

5.1 Motion Type & Mode

메뉴 및 구동 모드에 따라 Jog (Manual), Program (Auto), Homing Mode으로 구분되고, 이동 축 방향에 따라 Joint, Work, Ready Motion으로 구분됩니다.

Manual, Program (Auto), Homing 모드

Table 7 Manual, Program (Auto), Homing 모드

Item	Descriptions
Jog (Manual)	PRC Manager(GUI)의 Jog & Teaching Menu를 통해 구동합니다.
Program (Auto)	PRC Manager의 Program에서 모션 명령어를 수행해서 구동합니다.
Homing	PRC Manager의 Homing Menu를 통해 원점 복귀 모션을 수행합니다.

Joint, Work, Ready Motion

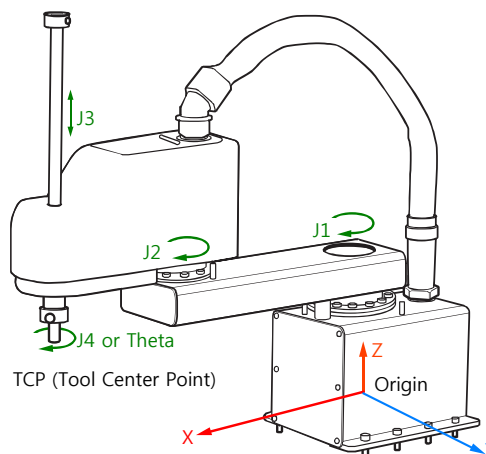


Figure 7 스카라 로봇

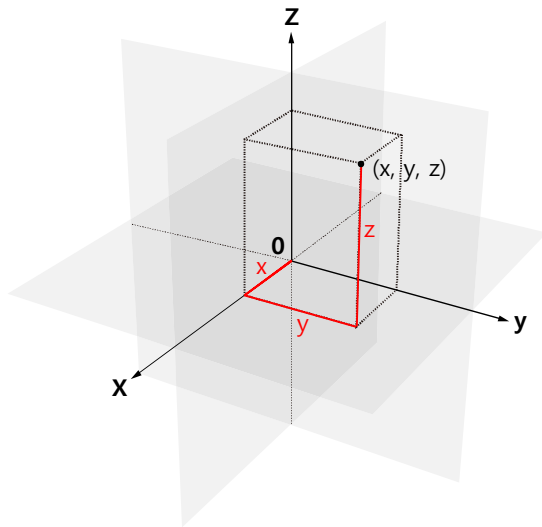
3차원 직교 좌표계를 World 좌표계로 사용

Table 8 Joint, Work, Ready Motion

Item	Descriptions
Joint (단축)	<ul style="list-style-type: none"> 축(Axis) 별로 제어합니다. 위 그림의 J1~J4로 표시된 축 방향으로 이동하는 모션입니다. 해당 축의 연결된 링크의 이동 각도 또는 이동 거리로 이동하는 모션입니다.
Work	<ul style="list-style-type: none"> World 좌표 (직교 좌표의 경우 X, Y, Z) 기준으로 로봇을 제어합니다. World 좌표로 TCP 위치를 이동하는 모션입니다.
Ready	System Configuration 또는 Motion Parameter에서 설정되어 있는 Ready 위치로 이동합니다.

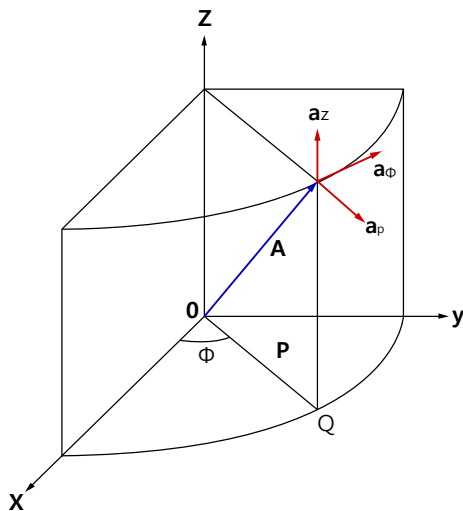
5.2 Coordinate System

직교 좌표 (XYZ)



Coordinate Cartesian		
	- (Negative)	+ (Positive)
X	-500.000	500.000
Y	-500.000	500.000
Z	-0.100	200.100

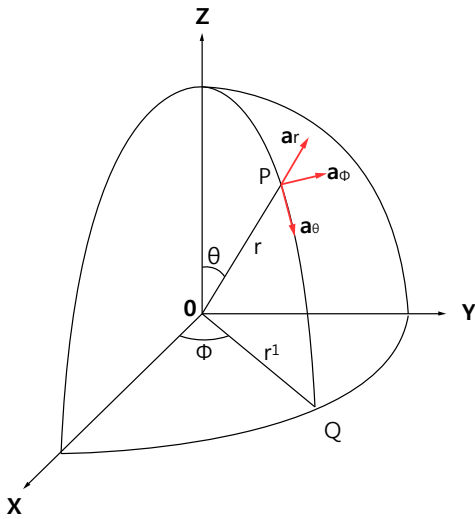
원통 좌표



Coordinate Cylindrical		
	- (Negative)	+ (Positive)
ρ	-500.000	500.000
ϕ	-500.000	500.000
z	-0.100	200.100

- ρ : x-y평면에서 원점으로부터 떨어진 거리
- ϕ : x축 양의 방향으로부터 OQ가 이루는 각도
- z: x-y평면으로부터 높이

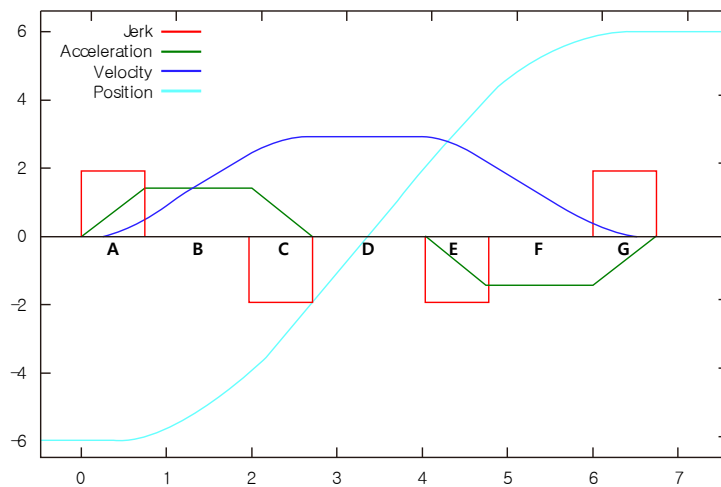
구 좌표



Coordinate	Spherical	
	- (Negative)	+ (Positive)
r	-500.000	500.000
θ	-500.000	500.000
φ	-0.100	200.100

- r: 원점에서 점P까지의 거리
- θ: z축 양의 방향으로부터 OP가 이루는 각도
- φ: x축 양의 방향으로부터 OQ가 이루는 각도

5.3 Motion Profile

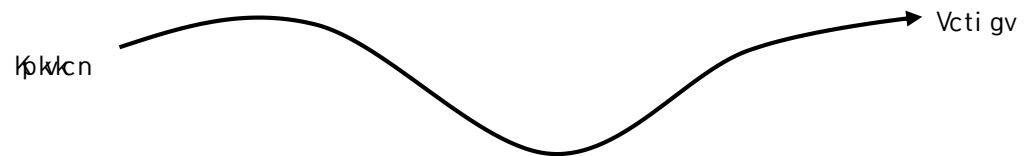


- **Velocity:** D 구간의 속도
- **Acceleration Time:** A+B+C 구간의 총 시간
- **Deceleration Time:** E+F+G 구간의 총 시간
- **Acceleration Jerk:** Acceleration Time중 A+C 구간 시간의 백분율
- **Deceleration Jerk:** Deceleration Time중 E+G 구간 시간의 백분율
- **Kill Time:** Motion을 급속 정지할 때 정지까지 걸리는 시간

5.4 Motion 궤적

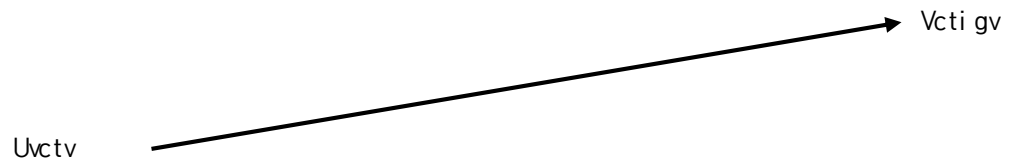
Move, MoveJ

- 현재 위치에서 Target Point까지 Joint Motion으로 움직입니다.
- Move는 각 축의 Motion Parameter 기준으로 구동되기 때문에 종료 시점이 다르고, Move J는 가장 오랜 모션 시간을 가지는 축을 기준으로 보간 모션으로 구동됩니다.



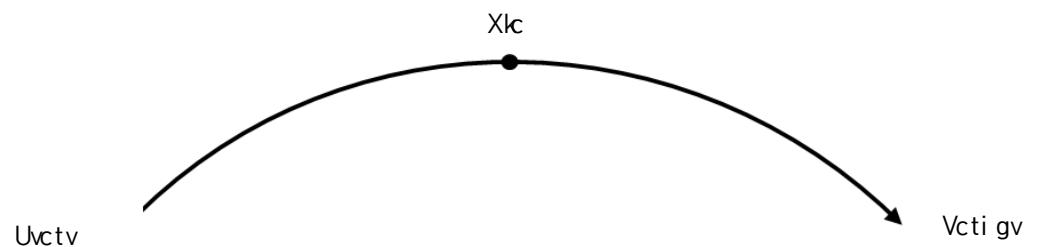
MoveL

현재 위치에서 Target Point로 직선 보간 이동



MoveC (원호보간)

Starting Point, Via Point, Target Point는 원 위의 세 점이 되어 Motion 궤적을 만듭니다.



Chapter 6. Pre-defined Commands for System & Robot

6.1 System Commands

Item	Command	Description
System Control	SysDate	시스템 현재 날짜 읽기
	SysTime	시스템 현재 시간 읽기
	TimerRead	타이머 시간 읽기
	TimerReset	타이머 기능 초기화
	TimerStart	타이머 기능 시작
	TimerStop	타이머 기능 정지
	Print	입력한 내용을 User Log Message 창에 출력
	Wait	지정한 시간(msec)만큼 대기
Program	PrgPause	프로그램 일시 정지
	PrgResume	프로그램 재 시작
	PrgStart	프로그램 시작
	PrgStop	프로그램 정지
System IO	DIn	Digital Input
	DOut	Digital Output
	AIn	Analog Input
	AOut	Analog Output

① 참고

각 명령어에 대한 상세 내용은 프로그램 가이드 문서를 참고하십시오.

6.2 Robot Commands

Robot[0].On(); // 0번 Robot Servo On

Item	Command	Description
Robot Class	Robot	Robot 객체 선언
Power(Servo On/Off)	On	Drive(AMP) On
	Off	Drive Off
Motion	Move	목표 위치로 직선 이동
	MoveJ	목표 위치로 직선 보간(Joint)
	MoveL	목표 위치로 직선 보간(World)
	MoveC	목표 위치로 원호보간
	MoveR	Ready 위치로 직선 이동
	StartBlend EndBlend	여러 Motion명령을 중첩(Blending) 이동
	MoveStop	모션 정지
	MoveEStop	비상 정지
	WaitM	모션 후 지정 시간만큼 대기
Motion Parameter	AJerkP	가속구간 Jerk 범위(Unit: %)
	ATime	가속시간
	Vel	속도
	DTime	감속시간
	DJerkP	감속구간 Jerk 범위(Unit: %)
	KTime	급속정지 감속시간
	CPos	지령 위치 값
	FPos	피드백 위치 값
Non-Volatile Variable (Store in Flash Memory)	JtPos	Joint Position Variables
	WkPos	Work Position Variables
	IVar	Integer(정수) Variables
	DVar	Double(실수) Variables

① 참고

Motion Parameter의 경우 Joint, Work, Ready Motion으로 구분

Ex) Joint Velocity → JtVel

Work Velocity → WkVel

Ready Velocity → RdVel

부록 A. ERROR Code 및 조치

분류	Code	원인	조치
Motion	0x10000001	RobotModel, Invalid.	로봇 모델번호를 확인
	0x10000002	Number of axes, Invalid. ($1 \leq N \leq 64$)	축 개수를 64 이하로 설정
	0x10000003	Jog motion, not move.	조그방향 변경
	0x10000004	WPos, Overreach.	로봇(PUMA) 목표위치 확인
	0x10000005	WPos, Overreach.	로봇(SCARA) 목표위치 확인
	0x10000006	WPos, Overreach.	로봇(LTR) 목표위치 확인
	0x10000007	WPos, Overreach.	로봇(DELTА) 목표위치 확인
	0x10000008	JPos, Overreach.	로봇(DELTА) 목표위치 확인
	0x10000009	WPos, Overreach.	로봇(HEXAPOD) 목표위치 확인
	0x1000000A	JPos, Overreach.	로봇(HEXAPOD) 목표위치 확인
	0x1000000B	MoveType, Invalid.	로봇(PUMA) 모션 변경
	0x1000000C	MoveType, Invalid.	로봇(HEXAPOD) 모션 변경
	0x1000000D	MoveType, Invalid.	로봇(SCARA) 모션 변경
	0x1000000E	MoveType, Invalid.	로봇(DELTА) 모션 변경
	0x1000000F	MoveType, Invalid.	로봇(SINGLE) 모션 변경
	0x10000010	RobotType, Invalid.	로봇 모션 변경
	0x10000011	RobotType, Invalid.	로봇 모션 변경
	0x10000012	RobotType, Invalid.	로봇 모션 변경
	0x10000013	Axis index, overflow.	축 번호 확인
	0x10000014	Configuration, Invalid.	로봇 자세 확인

	0x10000015	Configuration, Invalid.	로봇 자세 확인
	0x10000016	Configuration, Invalid.	로봇 자세 확인
	0x10000017	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x10000018	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x10000019	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x1000001A	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x1000001B	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x1000001C	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x1000001D	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x1000001E	Start JPos, out of limitJPos.	로봇 현재 위치 확인
	0x1000001F	Target JPos, out of limitJPos.	로봇 목표 위치 확인
	0x10000020	Target JPos, out of limitJPos.	로봇 목표 위치 확인
	0x10000021	Target JPos, out of limitJPos.	로봇 목표 위치 확인
	0x10000022	Target JPos, out of limitJPos.	로봇 목표 위치 확인
	0x10000023	Target JPos, out of limitJPos.	로봇 목표 위치 확인
	0x10000024	JPos, out of limitJPos.	로봇 위치 경로 확인
	0x10000025	JPos, out of limitJPos.	로봇 위치 경로 확인
	0x10000026	JVel, out of limitJVel.	로봇 제한 속도 확인
	0x10000027	JVel, out of limitJVel.	로봇 제한 속도 확인
	0x10000028	WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000029	WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x1000002A	Start WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x1000002B	Start WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x1000002C	Start WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x1000002D	Start WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x1000002E	Start WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x1000002F	Start WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000030	Target WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000031	Target WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000032	Target WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000033	Target WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000034	Target WPos, out of limitWorkArea.	로봇 제한 영역 확인
	0x10000035	Shoulder singularity.	로봇 위치 경로 확인
	0x10000036	Elbow singularity.	로봇 위치 경로 확인
	0x10000037	Wrist singularity.	로봇 위치 경로 확인
	0x10000038	Wrist singularity.	로봇 위치 경로 확인
	0x10000039	Wrist singularity.	로봇 위치 경로 확인
	0x1000003A	Start position, is near to wrist singularity.	로봇 현재 위치 확인
	0x1000003B	Target position, is near to wrist	로봇 목표 위치 확인

		singularity.	
	0x1000003C	StartBlend, does not exist.	Blend 문법 확인
	0x1000003D	StartBlend, already exist.	Blend 문법 확인
	0x1000003E	Vel, is zero.	모션 해당 속도 확인
	0x1000003F	Acc, is zero.	모션 해당 가속도 확인
	0x10000040	Dec, is zero.	모션 해당 감속도 확인
	0x10000041	KDec, is zero.	모션 해당 K감속도 확인
	0x10000042	AJerk, is zero.	모션 해당 가속구간 저크 확인
	0x10000043	DJerk, is zero.	모션 해당 감속구간 저크 확인
	0x10000044	ATime, is zero.	모션 해당 가속시간 확인
	0x10000045	DTime, is zero.	모션 해당 감속시간 확인
	0x10000046	KTime, is zero.	모션 해당 K감속시간 확인
	0x10000047	AJeckP, out of range.	모션 해당 가속구간 저크% 확인 (0~100%)
	0x10000048	DJeckP, out of range.	모션 해당 감속구간 저크% 확인 (0~100%)
	0x10000049	CI motion, need to more axis.	축 개수 확인
	0x1000004A	CI motion, arc is zero.	목표 위치, 경유 위치 확인
	0x1000004B	BI motion input data, does not exist.	blend 문법 확인

PRESTO 프레스토솔루션

www.prestosolution.co.kr

101-1404, Digital Empire Building 2nd, 88 Sinwon RD, Yeongtong-gu, Suwon-si, Gyeonggi-do, KOREA

T. +82-70-7167-8608 | F. +82-70-7159-2628