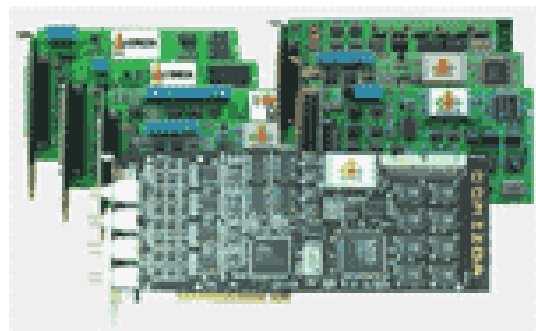
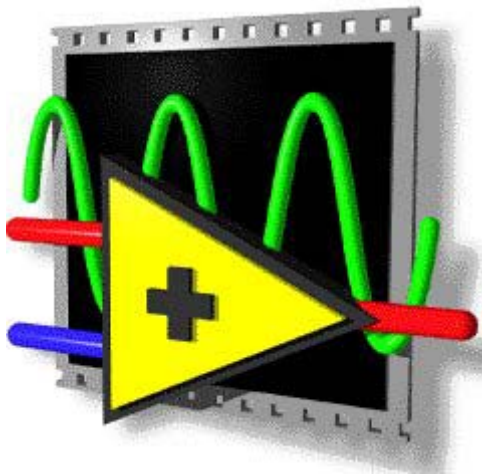


COMIDAS (LX Series)

LABVIEW 라이브러리 매뉴얼



*COMputer Innovation
is Zoomed by Our Affection!*



저작권자 : ㈜커미조아

Copyright (c) by COMIZOA CO.,LTD. All right reserved.

이 사용자 설명서는 저작권법에 의해 보호되고 있습니다.

㈜커미조아의 사전 서면 동의 없이 사용자설명서의 일부 또는 전체를 어떤 형태로든 복사, 전재할 수 없습니다.

Hardware Support : Hardware@comizoa.co.kr

Software Support : Software@comizoa.co.kr



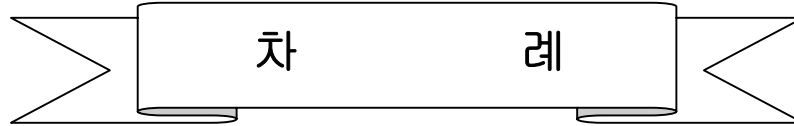
㈜커미조아

www.comizoa.co.kr

www.comizoa.com

Tel) 042 - 861 - 3301~3

Fax) 042 - 861 - 3304



PART I INTRODUCTION.....	7
LABVIEW V5.5에서 설치 및 사용.....	8
<i>VI Search Path</i> 등록.....	10
LABVIEW V6.01에서 설치 및 사용.....	14
<i>VI Search Path</i> 등록.....	17
PART II FUNCTIONS	27
LOAD/UNLOAD	28
<i>COMILX_LoadDevice</i>	28
<i>COMILX_UnloadDevice</i>	29
AI GENERAL	30
<i>COMILX_AD_SetInputType</i>	30
<i>COMILX_AD_SetRange</i>	31
<i>COMILX_AD_GetDigit</i>	32
<i>COMILX_AD_GetVolt</i>	33
AI-SCAN FOR LX10X SERIES	34
<i>COMILX_US1_Start</i>	35
<i>COMILX_US1_Stop</i>	36
<i>COMILX_US1_CurCount</i>	37
<i>COMILX_US1_RetrOne</i>	38
<i>COMILX_US1_RetrChannel</i>	39
<i>COMILX_US1_RetrBlock</i>	40
<i>COMILX_US1_ReleaseBuf</i>	41
AI-SCAN FOR LX20X SERIES	42
<i>COMILX_US2_SetTriggerEvent</i>	43
<i>COMILX_US2_Start</i>	46
<i>COMILX_US2_Stop</i>	48

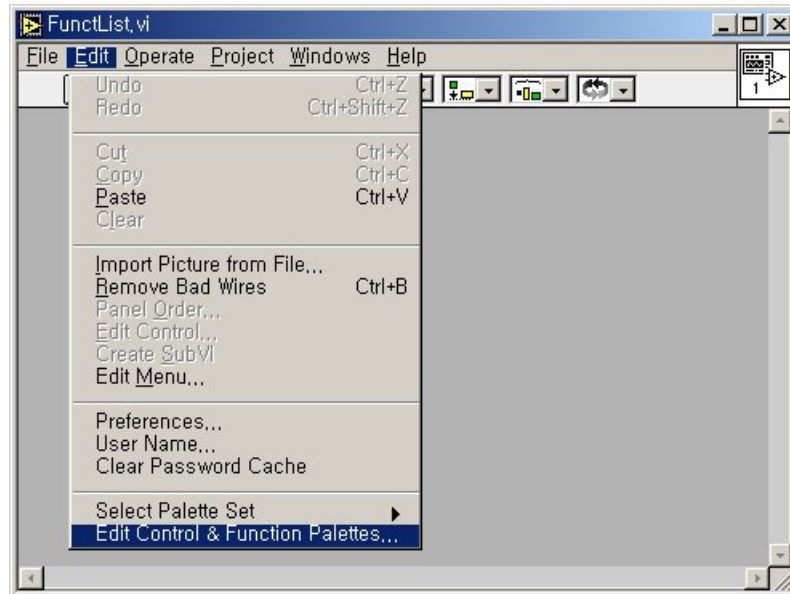
<i>COMILX_US2_Acquire</i>	49
<i>COMILX_US2_Resume</i>	49
<i>COMILX_US2_IsBufFull</i>	50
<i>COMILX_US2_RetrChannel</i>	51
<i>COMILX_US2_ChangeScanFreq</i>	52
<i>COMILX_US2_DmaCount</i>	53
<i>COMILX_US2_ReleaseBuf</i>	54
ANALOG OUT.....	55
<i>COMILX_DA_Out</i>	55
<i>COMILX_WFM_Start</i>	56
<i>COMILX_WFM_Stop</i>	57
<i>COMILX_WFM_RateChange</i>	58
<i>COMILX_WFM_GetCurPos</i>	59
<i>COMILX_WFM_GetCurLoops</i>	60
COUNTER 32BIT.....	61
<i>COMILX_ReadCounter32</i>	61
<i>COMILX_ClearCounter32</i>	62
DIGITAL I/O.....	63
<i>COMILX_DIO_SetUsage</i>	63
<i>COMILX_DI_GetOne</i>	64
<i>COMILX_DI_GetAll</i>	65
<i>COMILX_DO_PutOne</i>	66
<i>COMILX_DO_PutAll</i>	67
SERIAL DIGITAL I/O.....	68
<i>COMILX_SDIO_InitComm</i>	68
<i>COMILX_SDIO_CheckModule</i>	69
<i>COMILX_SDIO_SetDioUsage</i>	70
<i>COMILX_SDIO_ReadLowByte</i>	71
<i>COMILX_SDIO_ReadHighByte</i>	72
<i>COMILX_SDIO_WriteLowByte</i>	73
<i>COMILX_SDIO_WriteHighByte</i>	74



PART | Introduction

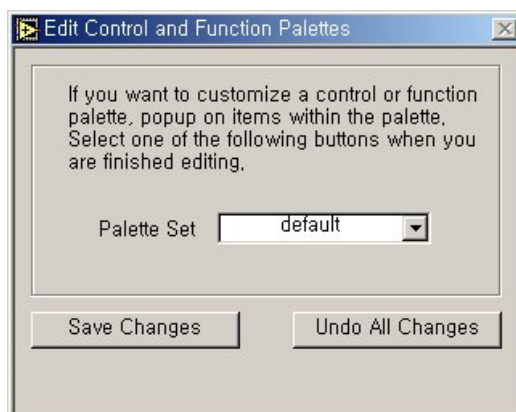
Labview V5.5에서 설치 및 사용

LabView V5.5 에서 COMIDAS 라이브러리를 사용하기 위해서는 먼저 LabView 의 Function Palette 에 COMIDAS Function 블록을 등록하여야 합니다. Function Palette 에 등록하기 위해서는 [그림 1-1]에서와 같이 “Edit=>Edit Control & Function Palettes” 메뉴를 선택합니다.

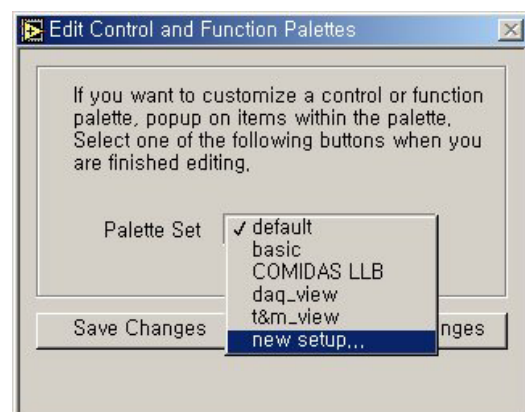


[그림 1-1] Edit Control & Function Palettes 메뉴 선택

[그림 1-2]와 같은 화면이 나타나면 [그림 1-3]과 같이 Palette Set 항목에서 “new setup...” 항목을 선택하십시오.

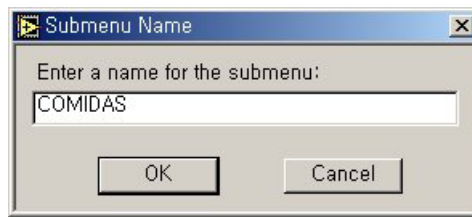


[그림 1-2] Palette 선택 대화상자 1



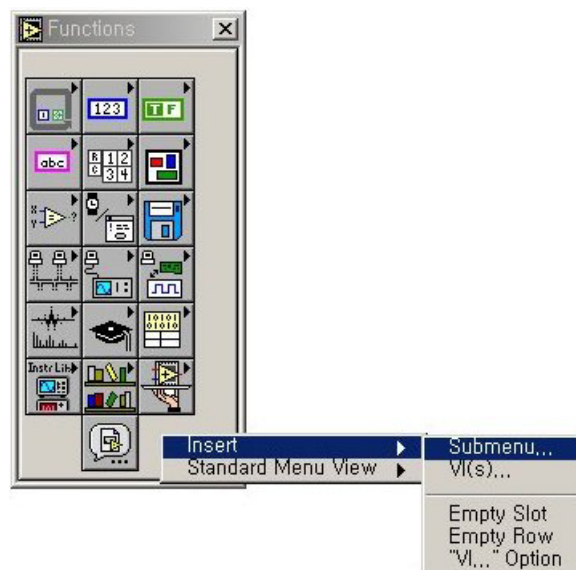
[그림 1-3] Palette 선택 대화상자 2

[그림 1-4]와 같은 화면이 나타나면 사용자가 사용할 Palette 의 이름을 지정합니다. 이 이름은 사용자 마음대로 지정할 수 있습니다.



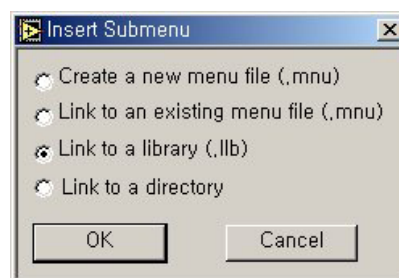
[그림 1-4] 새 Palette 이름 지정 대화상자

새 Palette 의 이름을 지정하였으면 Functions Palette 에서 빈 공간에 오른쪽 마우스 클릭을 하여 [그림 1-5]와 같이 “Insert => Submenu...” 메뉴를 선택합니다.



[그림 1-5] “Insert => Submenu...” 메뉴 선택

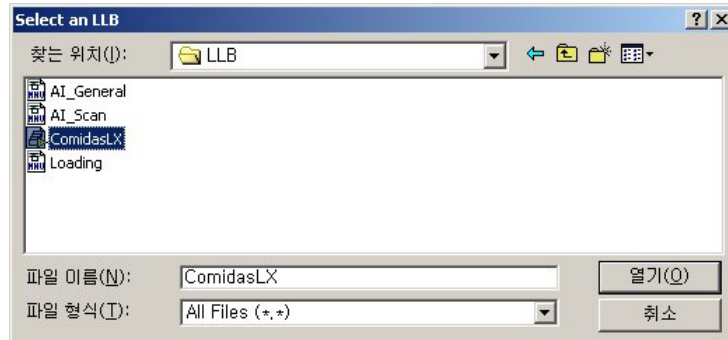
[그림 1-6]과 같은 화면이 나타나면 “Link to a library”를 선택한 후 OK 버튼을 클릭합니다.



[그림 1-6] “Insert Submenu” 대화상자

[그림 1-7]과 같이 파일을 선택하는 화면이 나타나면 C:\Program Files\WComiza\WComidas-LX\Labview\WLib 폴더에서 ComidasLX.llb 파일을 선택한 후 열

기 버튼을 클릭합니다. 단, 폴더명은 COMIDAS Programs 를 설치할 때 설치옵션에 따라 다를 수 있습니다.



[그림 1-7] “Select an LLB” 대화상자

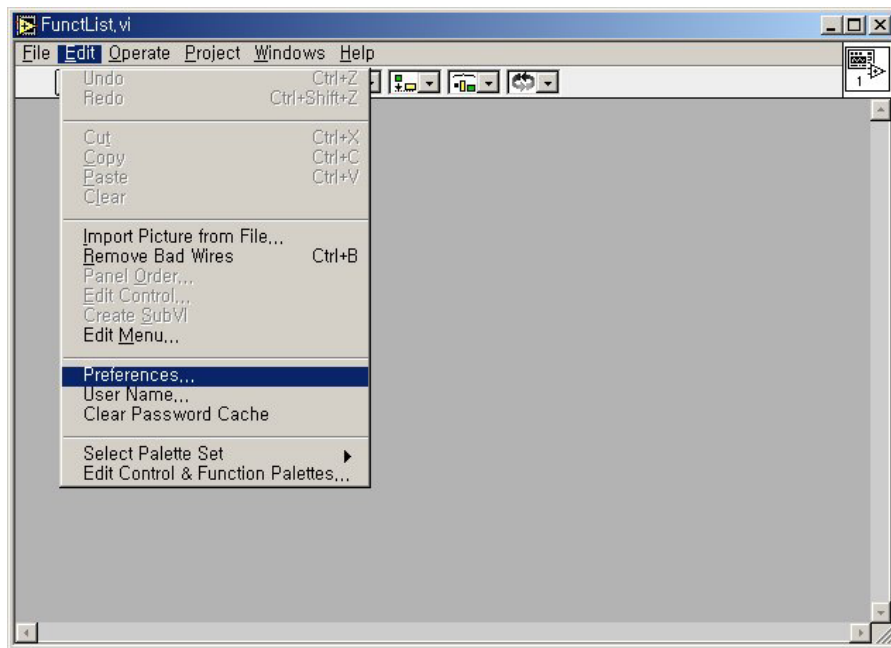
위의 과정을 성공적으로 마쳤으면 [그림 1-8]과 같이 Functions Palette 에 COMIDAS-LX Functions 블록이 나타납니다. 이 후에 사용자는 Functions Palette 에서 필요한 COMIDAS-LX Function 을 선택하여 사용할 수 있습니다.



[그림 1-8] Functions Palette 에 COMIDAS-LX Functions 블록이 등록된 화면

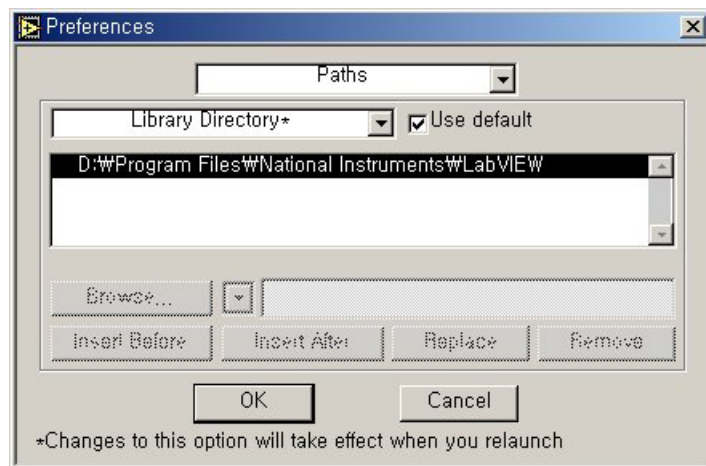
VI Search Path 등록

COMIDAS Function Library 를 VI Search Path 에 등록하는 것이 좋습니다. 그렇지 않은 경우 Function VI 를 이용하여 프로그램을 작성 후 다시 Open 할 때 사용한 Function VI 들의 위치를 재차 확인하는 경우가 발생할 수 있습니다. VI Search Path 에 등록하려면 [그림 1-9]와 같이 “Edit=>Preferences” 메뉴를 선택합니다.

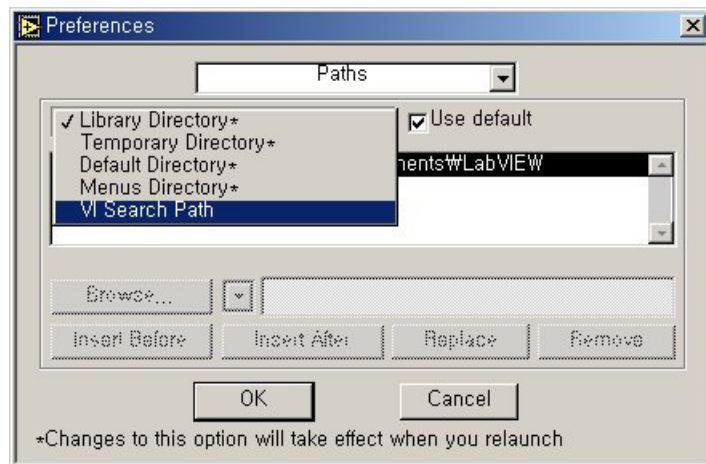


[그림 1-9] Preferences 메뉴 선택

[그림 1-10]과 같은 화면이 나타나면 [그림 1-11]과 같이 VI Search Path 를 선택한 후 우측에 있는 "Use default" 항목을 언체크(Uncheck)합니다.

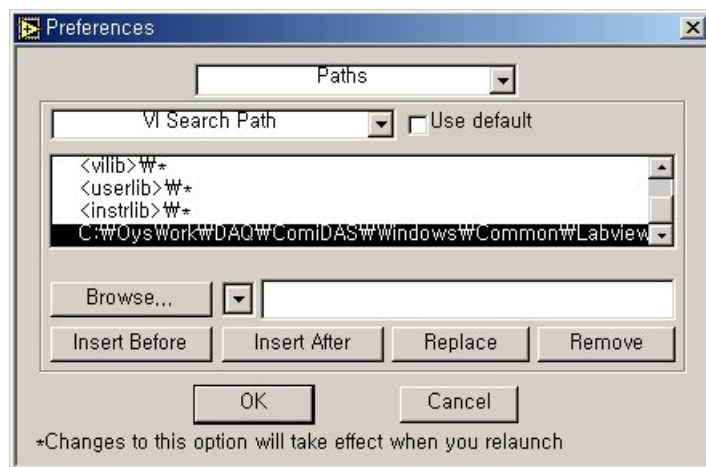


[그림 1-10] Preference 대화 상자 1



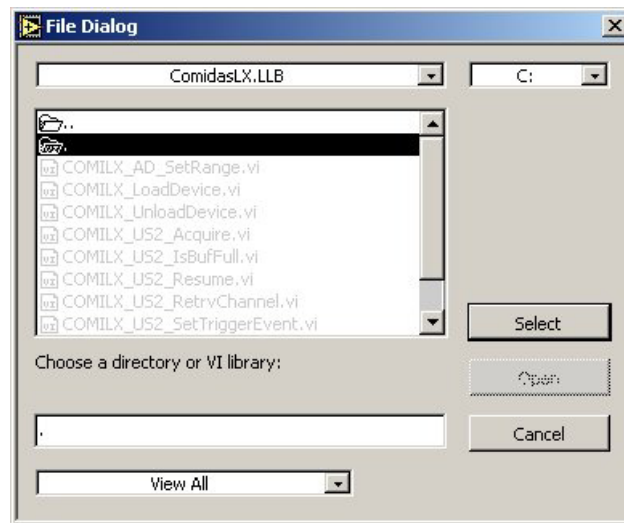
[그림 1-11] Preference 대화 상자 2

그러면 [그림 1-12]와 같이 하단에 있는 버튼들의 Dimmed 가 풀리게 됩니다. "Browse"버튼을 클릭하여 C:\WProgram Files\WComiza\WComidas-LX\Labview\WLib 폴더에서 ComidasLX.lib 를 선택합니다. 이 때 C:\WProgram Files\WComiza\WComidas-LX 는 COMIDAS 프로그램이 설치된 폴더를 의미합니다.



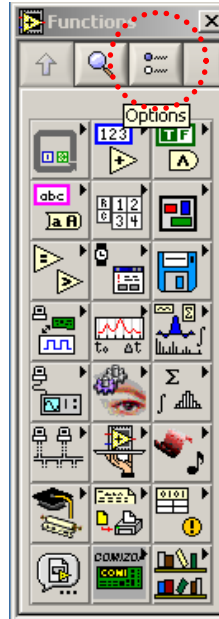
[그림 1-12] Preference 대화 상자 3

아래와 같은 화면이 나타나면 그림과 같이 두 번째 폴더그림을 선택한 후 "Select" 버튼을 클릭합니다. 그러면 다시 [그림 1-12]의 화면으로 복귀하는데 여기서 "Insert Before" 또는 "Insert After"버튼을 클릭한 후 OK 버튼을 클릭하면 VI Search Path 등록이 완료됩니다.



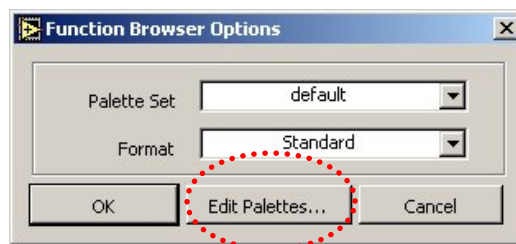
Labview V6.01에서 설치 및 사용

LabView V6 에서 COMIDAS 라이브러리를 사용하기 위해서는 LabView V5.5 와 마찬가지로 먼저 LabView 의 Function Palette 에 COMIDAS Function 블록을 등록 하여야 합니다. Function Palette 에 등록하기 위해서는 [그림 1-13]에서와 같이 Function Palette 에서 “Options” 버튼을 클릭합니다.



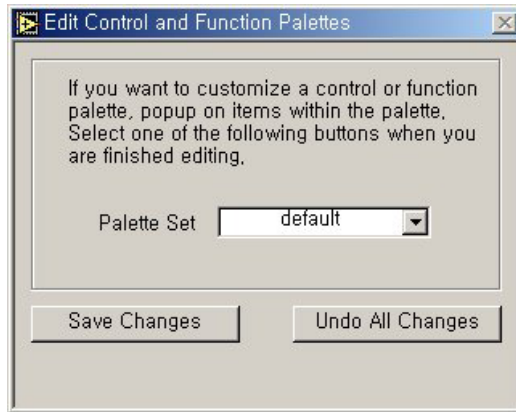
[그림 1-13] Function Palette 에서 Option 버튼 클릭

[그림 1-14]와 같이 Function Browser Options 대화상자 화면이 나타나면 “Edit Palettes...” 버튼을 클릭 하십시오.

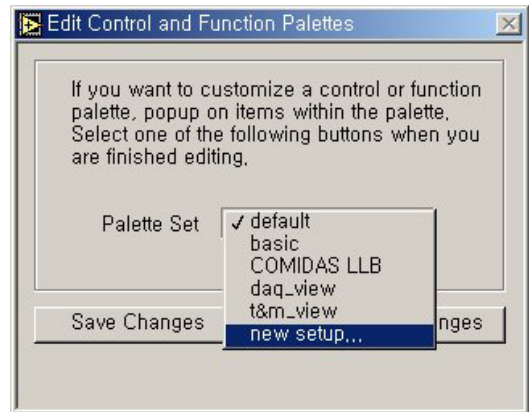


[그림 1-14] Function Browser Options 대화상자

[그림 1-15]와 같은 화면이 나타나면 [그림 1-16]과 같이 Palette Set 항목에서 “new setup...” 항목을 선택하십시오.

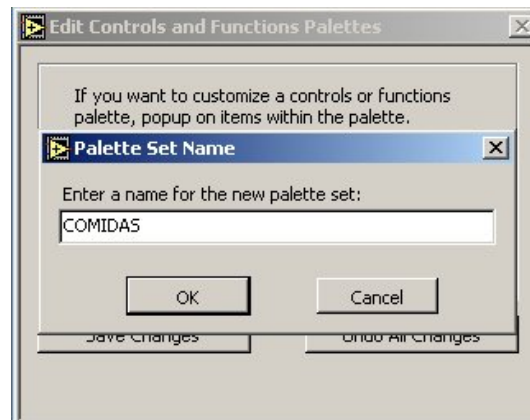


[그림 1-15] Palette 선택 대화상자 1



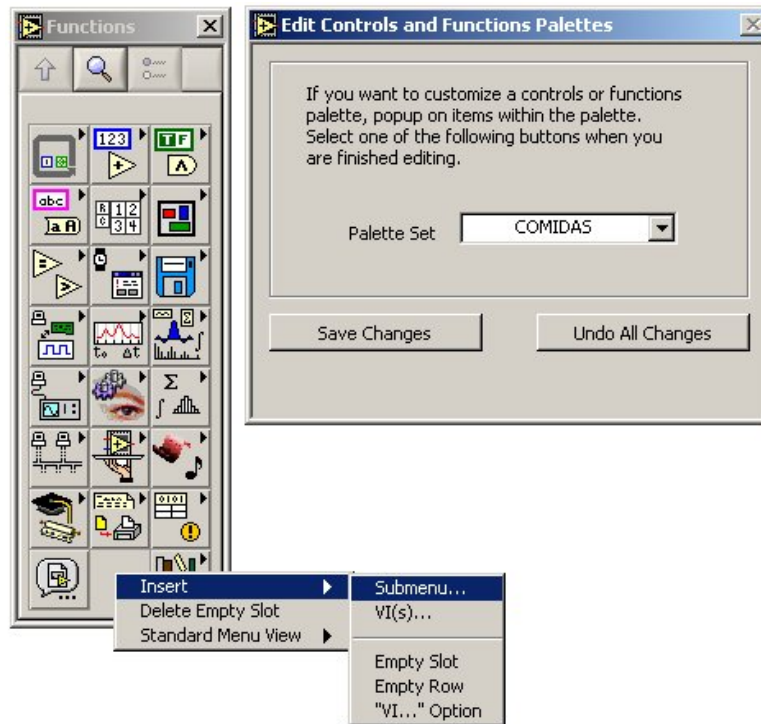
[그림 1-16] Palette 선택 대화상자 2

[그림 1-17]과 같은 화면이 나타나면 사용자가 사용할 Palette 의 이름을 지정합니다. 이 이름은 사용자 마음대로 지정할 수 있습니다.



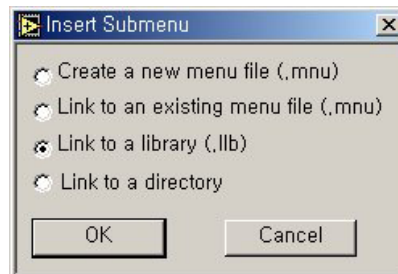
[그림 1-17] Palette Set Name 설정 대화상자

새 Palette 의 이름을 지정 하였으면 Functions Palette 에서 빈 공간에 오른쪽 마우스 클릭을 하여 [그림 1-18]과 같이 "Insert => Submenu..." 메뉴를 선택합니다.



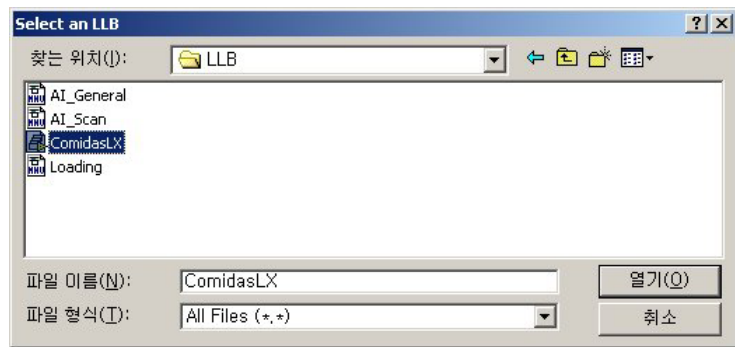
[그림 1-18] Insert Submenu 메뉴 선택 화면

[그림 1-19]과 같은 화면이 나타나면 “Link to a library”를 선택한 후 OK 버튼을 클릭합니다.



[그림 1-19] “Insert Submenu” 대화상자

[그림 1-20]과 같이 파일을 선택하는 화면이 나타나면 C:\Program Files\COMIDAS-LX\Window\Labview\WLib 폴더에서 ComidasLX.llb 파일을 선택한 후 열기 버튼을 클릭합니다.



[그림 1-20] “Select an LLB” 대화상자

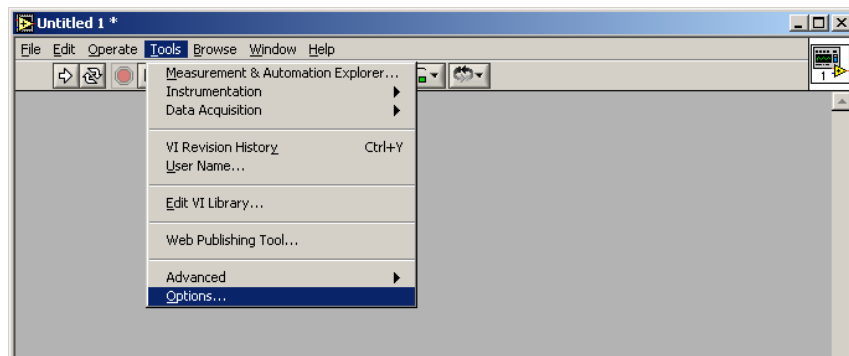
위의 과정을 성공적으로 마쳤으면 [그림 1-21]과 같이 Functions Palette 에 COMIDAS Functions 블록이 나타납니다. 이 후에 사용자는 Functions Palette 에서 필요한 COMIDAS Function 을 선택하여 사용할 수 있습니다.



[그림 1-21] Functions Palette 에 COMIDAS-LX Functions 블록이 등록된 화면

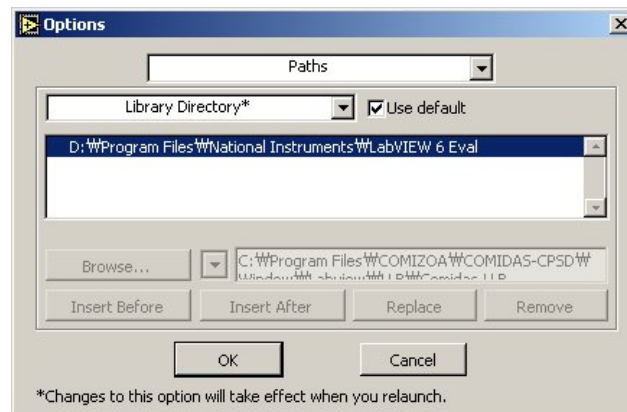
VI Search Path 등록

COMIDAS Function Library 를 VI Search Path 에 등록하는 것이 좋습니다. 그렇지 않은 경우 Function VI 를 이용하여 프로그램을 작성 후 다시 Open 할 때 사용한 Function VI 들의 위치를 재차 확인하는 경우가 발생할 수 있습니다. VI Search Path 에 등록하려면 [그림 1-22]와 같이 “Tools=>Options...” 메뉴를 선택합니다.

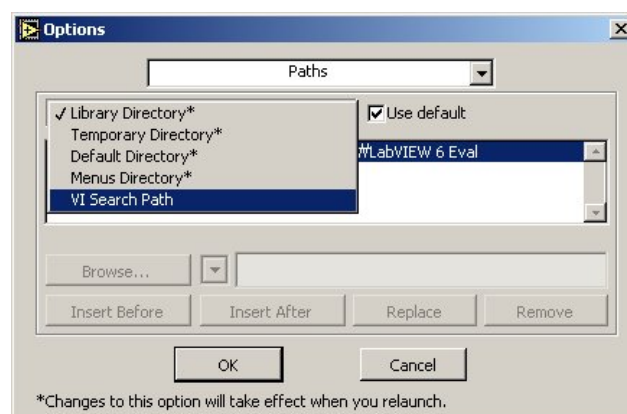


[그림 1-22] Tools=>Options 메뉴 선택

[그림 1-23]과 같은 화면이 나타나면 [그림 1-24]과 같이 VI Search Path 를 선택한 후 우측에 있는 “Use default” 항목을 언체크(Uncheck)합니다.



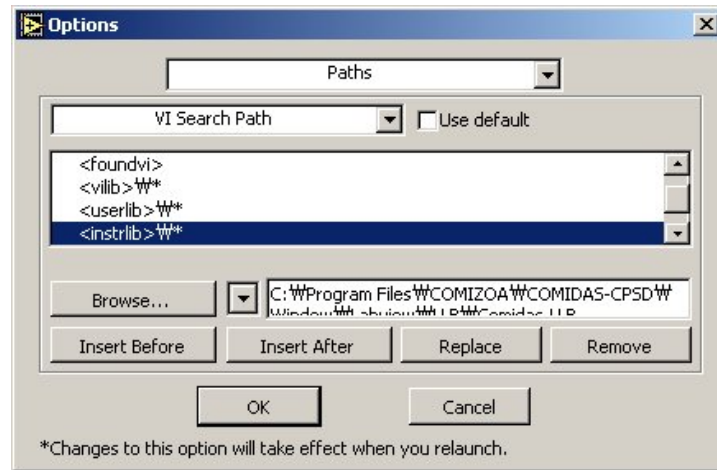
[그림 1-23] Options 대화상자



[그림 1-24] Options 대화상자에서 “VI Search Path” 선택화면

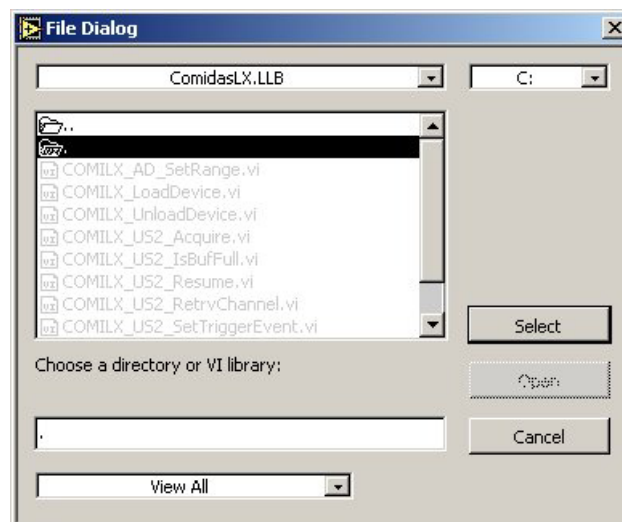
그러면 [그림 1-25]와 같이 하단에 있는 버튼들의 Dimmed 가 풀리게 됩니다. “Browse”버튼을 클릭하여 C:\Program Files\Comiza\Comidas-LX\Labview\Lib 폴더에

서 ComidasLX.llb 를 선택합니다. 이 때 C:\Program Files\Comizo\Comidas-LX 는 COMIDAS 프로그램이 설치된 폴더를 의미합니다.



[그림 1-25] Preference 대화 상자 3

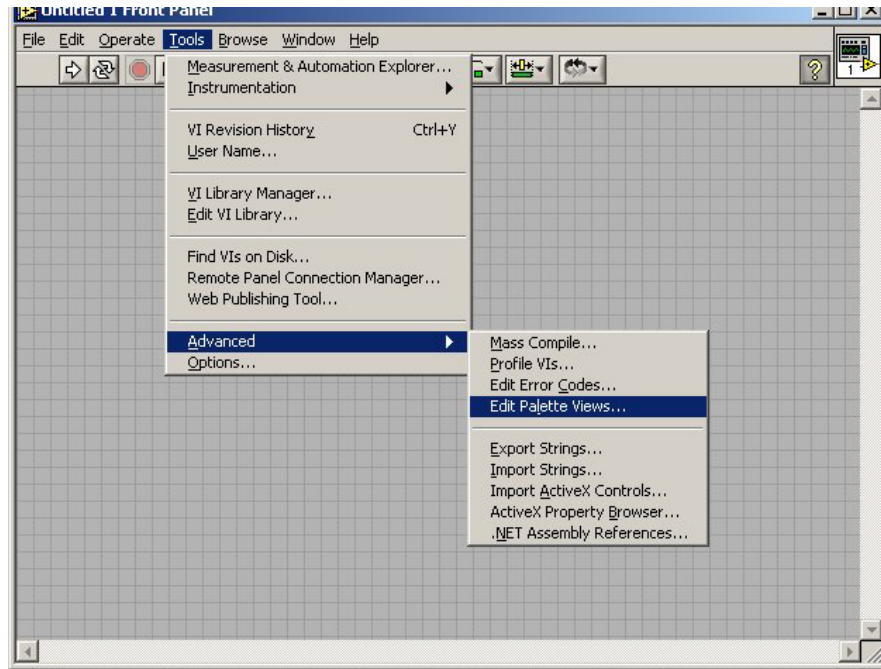
아래와 같은 화면이 나타나면 그림과 같이 두 번째 폴더그림을 선택한 후 "Select" 버튼을 클릭합니다. 그러면 다시 [그림 1-25]의 화면으로 복귀하는데 여기서 "Insert Before" 또는 "Insert After" 버튼을 클릭한 후 OK 버튼을 클릭하면 VI Search Path 등록이 완료됩니다.



[그림 1-26] File Dialog

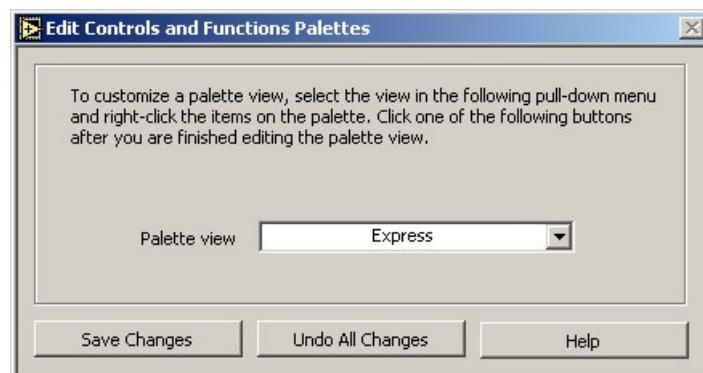
Labview V7 에서 설치 및 사용

LabView V7 에서 COMIDAS 라이브러리를 사용하기 위해서는 먼저 LabView 의 Function Palette 에 COMIDAS Function 블록을 등록 하여야 합니다. Function Palette 에 등록 하기 위해서는 [그림 1-26]에서와 같이 Tools=>Advanced=>Edit Palette Views 메뉴를 선택합니다.



[그림 1-27] “Edit Pallete Views” 메뉴 선택

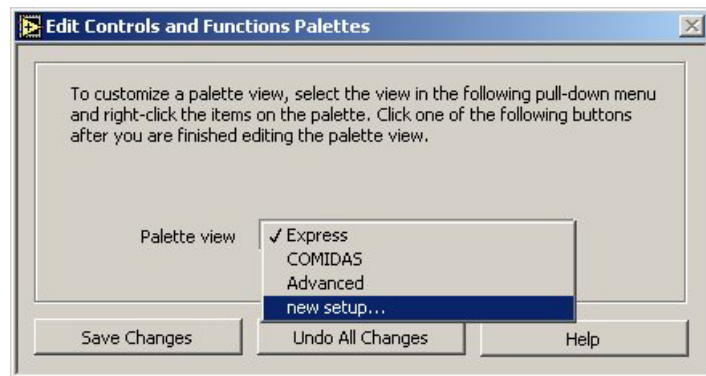
그러면 [그림 1-28]과 같이 “Edit Controls and Functions Palettes” 대화상자 화면이 나타납니다.



[그림 1-28] Function Browser Options 대화상자

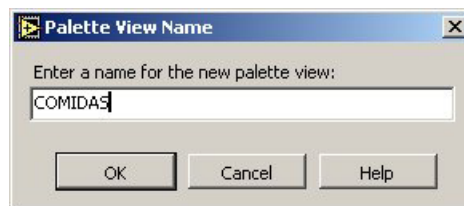
[그림 1-28]대화상자에서 “Palette view” 항목을 클릭하여 [그림 1-29]와 같이 “new

setup...” 메뉴를 선택합니다.



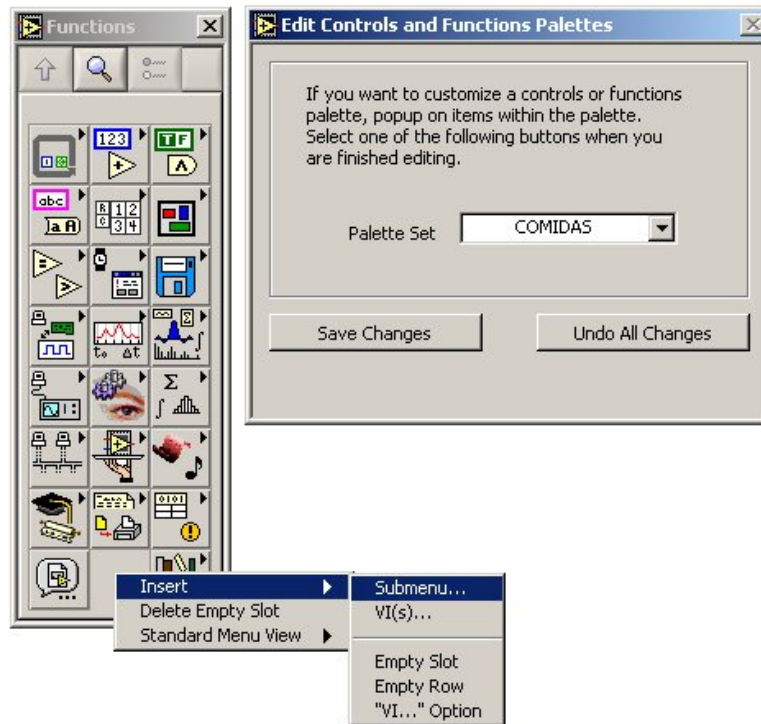
[그림 1-29] new setup 메뉴 선택 화면

[그림 1-30] 화면이 나타나면 Palette view 의 이름을 설정합니다. 여기서의 이름은 임의로 설정할 수 있습니다.



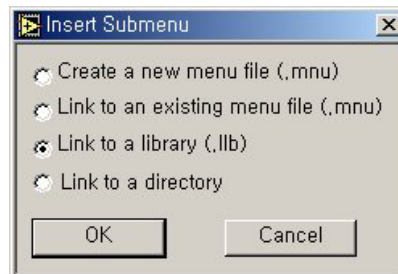
[그림 1-30] Palette view 이름 설정

[그림 1-28] 대화상자 화면으로 복귀되면 이번에는 [그림 1-31]과 같이 Function Palette 창에서 오른쪽 마우스클릭을 이용하여 Insert=>Submenu... 메뉴를 선택합니다.



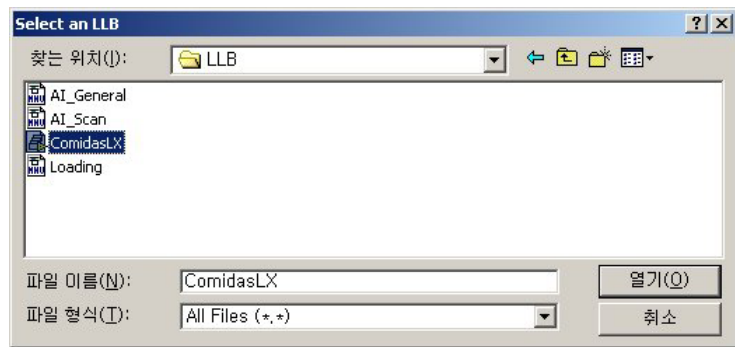
[그림 1-31] Insert=>Submenu... 메뉴 선택

[그림 1-32]와 같은 화면이 나타나면 “Link to a library”를 선택한 후 OK 버튼을 클릭합니다.



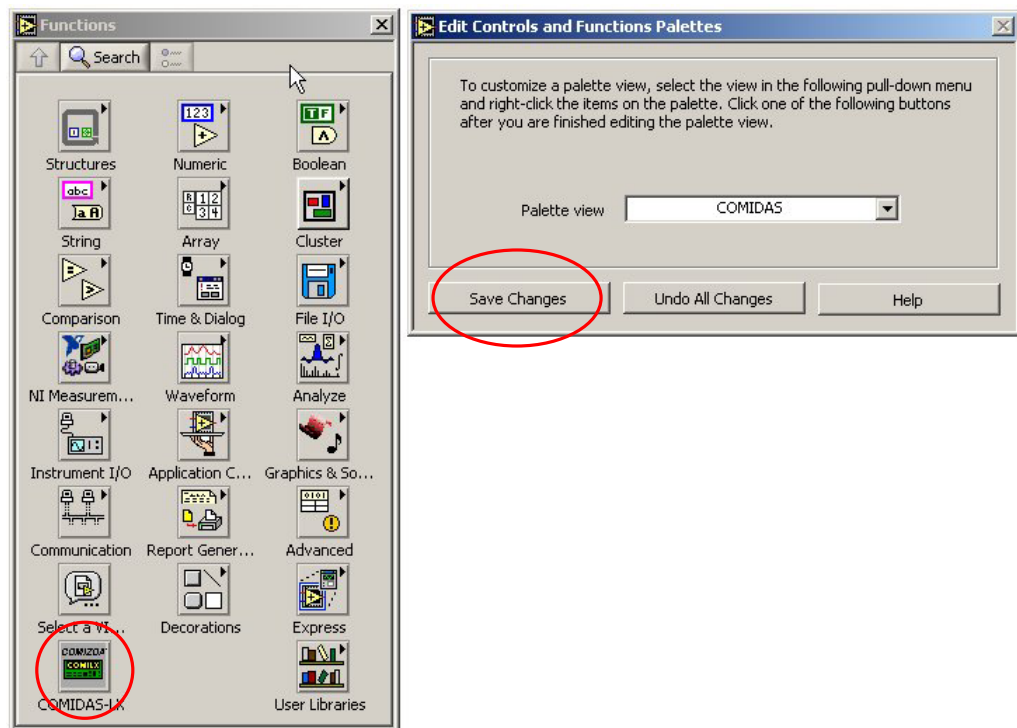
[그림 1-32] “Insert Submenu” 대화상자

[그림 1-33]과 같이 파일을 선택하는 화면이 나타나면 C:\Program Files\WCOMIDAS-LX\Window\Labview\WLib 폴더에서 ComidasLX.llb 파일을 선택한 후 열기 버튼을 클릭합니다.



[그림 1-33] “Select an LLB” 대화상자

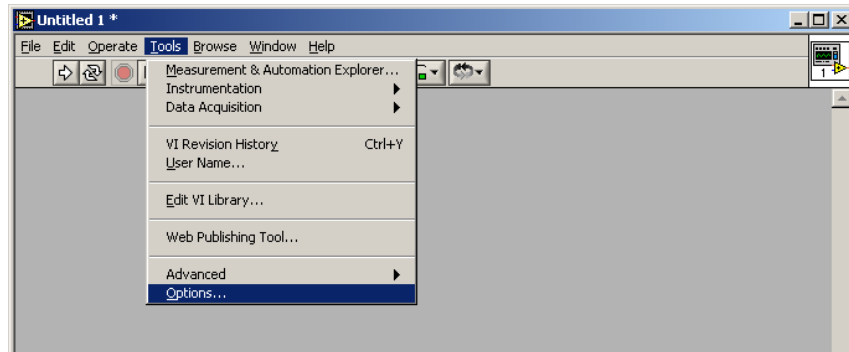
위의 과정을 성공적으로 마쳤으면 [그림 1-34]과 같이 Functions Palette 에 COMIDAS-LX Functions 블록이 나타납니다. 성공적으로 등록된 것이 확인되면 “Save Changes” 버튼을 클릭하여 설정을 저장합니다. 이 후에 사용자는 Functions Palette 에서 필요한 COMIDAS Function 을 선택하여 사용할 수 있습니다.



[그림 1-34] Functions Palette 에 COMIDAS-LX Functions 블록이 등록된 화면

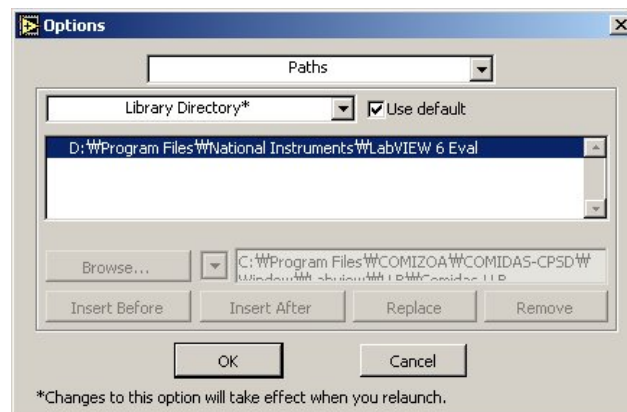
VI Search Path 등록

COMIDAS Function Library 를 VI Search Path 에 등록하는 것이 좋습니다. 그렇지 않은 경우 Function VI 를 이용하여 프로그램을 작성 후 다시 Open 할 때 사용한 Function VI 들의 위치를 재차 확인하는 경우가 발생할 수 있습니다. VI Search Path 에 등록하려면 [그림 1-35]와 같이 “Tools=>Options...” 메뉴를 선택합니다.

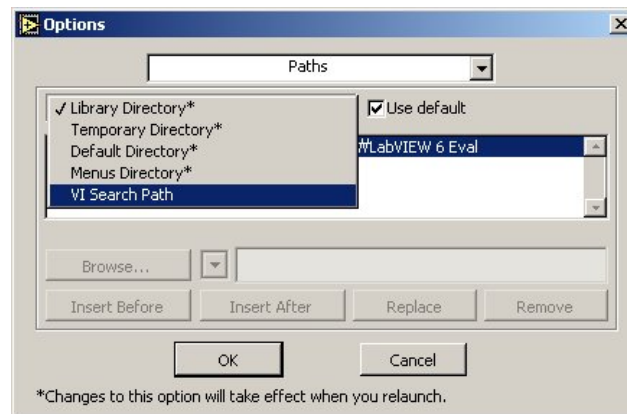


[그림 1-35] Tools=>Options 메뉴 선택

[그림 1-36]과 같은 화면이 나타나면 [그림 1-37]과 같이 VI Search Path 를 선택한 후 우측에 있는 “Use default” 항목을 언체크(Uncheck)합니다.

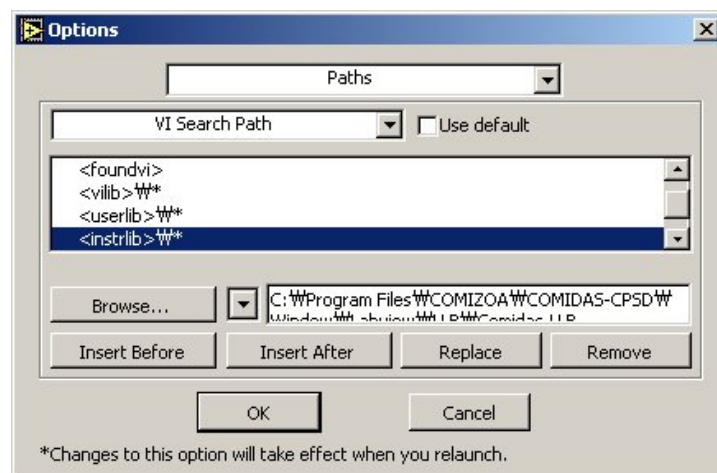


[그림 1-36] Options 대화상자



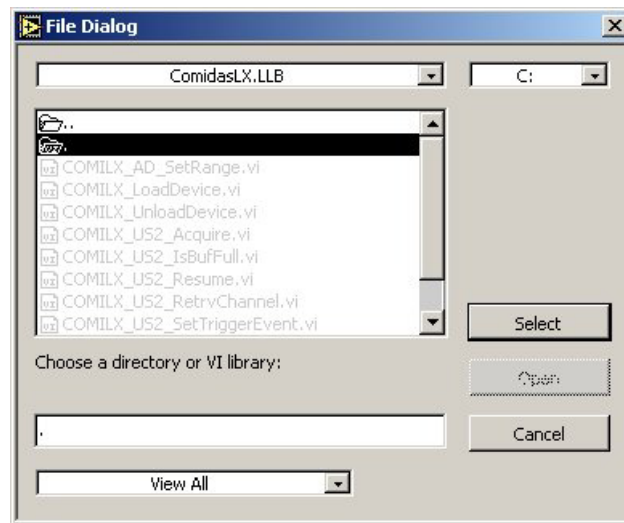
[그림 1-37] Options 대화상자에서 “VI Search Path” 선택 화면

그러면 [그림 1-38]와 같이 하단에 있는 버튼들의 Dimmed 가 풀리게 됩니다. “Browse”버튼을 클릭하여 C:\WProgram Files\WComizoa\WComidas-LX\Labview\WLib 폴더에서 ComidasLX.lib 를 선택합니다. 이 때 C:\WProgram Files\WComizoa\WComidas-LX 는 COMIDAS 프로그램이 설치된 폴더를 의미합니다.



[그림 1-38] “Use default” 언체크된 화면

[그림 1-39]와 같은 화면이 나타나면 그림과 같이 두 번째 폴더그림을 선택한 후 “Select” 버튼을 클릭합니다. 그러면 다시 [그림 1-38]의 화면으로 복귀하는데 여기서 “Insert Before” 또는 “Insert After”버튼을 클릭한 후 OK 버튼을 클릭하면 VI Search Path 등록이 완료됩니다.



[그림 1-39] File Dialog



PART II Functions

Load/Unload



Load/Unload Function 블록은 각 디바이스를 로드(Load)/언로드(Unload)하는 함수들을 모아놓은 것입니다. 이 함수들은 COMIDAS 라이브러리를 사용하기 위해 필수적으로 적용되어져야 할 함수들입니다. COMIDAS 라이브러리의 거의 모든 함수들은 디바이스 핸들을 parameter 로 입력 하여야 합니다. 디바이스 핸들을 얻기 위해서는 디바이스를 로드해야 합니다. 또한 프로그램 종료 시에는 디바이스를 언로드 하여야 합니다.

함수명	각 보드 별 지원 여부									
	LX101	LX102	LX103	LX201	LX202	LX203	LX301	LX401	LX501	L502
COMILX_LoadDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMILX_UnloadDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

디바이스 시작/종료 함수 리스트 및 각 보드별 지원 여부

COMILX_LoadDevice

■ 아이콘



■ Function 개요

하나의 COMIDAS 디바이스를 로드(load)합니다. 각 디바이스를 제어하기 위해서는 먼저 이 함수를 이용하여 해당 디바이스에 대한 핸들을 얻어와야 합니다.

■ Parameters

Parameters	Type	Description
Device ID	U16	각 디바이스의 고유한 아이디 값입니다. 이 값은 4 자리의 16 진수 값으로 구성됩니다. 첫 자리는 A로 시작하고 나머지 세 자리는 각 디바이스의 모델 번호와 같은 값을 적어주시면 됩니다. Ex) COMI-LX101 => A101 (Hex) COMI-LX201 => A201 (Hex)

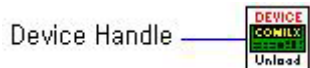
Instance	U32	동일 Device ID를 가진 여러 개의 디바이스를 구분하기 위한 값입니다. 같은 종류의 디바이스가 동일 컴퓨터에 여러 개 장착된다면 장착된 순서대로 instance 번호가 부여됩니다. Instance 번호는 0번부터 차례로 부여됩니다. 예를 들어 2개의 COMI-LX101 보드가 장착되어 있다면 처음 장착된 보드의 instance값은 0이 되며, 두 번째 장착된 보드의 instance값은 1이 됩니다.
----------	-----	---

■ Return 값

Device Handle	U32	이 함수는 디바이스 핸들을 반환합니다. 이 값은 디바이스를 제어하는 각 함수의 첫 번째 parameter로 사용됩니다. 만일 이 값이 FFFFFFFF(Hex)이면 디바이스 로딩이 실패한 것입니다.
---------------	-----	---

COMILX_UnloadDevice

■ 아이콘



■ Function 개요

하나의 COMIDAS 디바이스를 언로드(Unload)합니다. 프로그램 종료 부분에서 이 함수를 이용하여 디바이스를 언로드 하여야 합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ 참고

이 함수는 보통 프로그램 종료 시에 수행하면 됩니다.

AI General



AI General Function 블록은 Analog Input (A/D)을 위하여 일반적으로 사용되는 함수들을 모아놓은 것입니다.

함수명	각 보드 별 지원 여부					
	LX10x	LX20x	LX301	LX401	LX402	LX501
COMILX_AD_SetInputType	✓					
COMILX_AD_SetRange	✓	✓				
COMILX_AD_GetDigit	✓					
COMILX_AD_GetVolt	✓					

아날로그 입력 공통 함수 리스트 및 각 보드별 지원 여부

COMILX_AD_SetInputType

■ 아이콘



■ Function 개요

이 함수는 아날로그 입력 신호의 연결 형식을 소프트웨어적으로 설정합니다. 아날로그 입력 신호의 연결 형식에는 Single ended 방식과 Differential 방식의 두 가지가 있습니다. (하드웨어 매뉴얼 참조)

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Input Mode	U8	아날로그 입력 신호의 연결 형식을 설정합니다. 이 값은 다음 중 하나의 값이어야 합니다. 기본값은 1 (AI_SINGLE) 로 설정 되어 있습니다.

		▷ 0 => 아날로그 입력 신호의 연결형식을 Differential 로 설정합니다. ▷ 1 => 아날로그 입력 신호의 연결형식을 Single Ended 로 설정합니다.
--	--	--

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_AD_SetRange

■ 아이콘



■ Function 개요

각 Analog Input 채널의 입력 범위를 설정합니다. 입력범위는 기본적으로 -10 ~ +10 Volt 로 설정됩니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch No	I32	Analog Input 채널 번호 (0 – Based)
Min. Volt	Single	A/D범위의 최소값을 지정합니다. 유효한 값은 보드 종류에 따라 다음과 같습니다. •COMI-LX10x 디바이스 ~ 0, -1, -2, -5, -10 •COMI-LX20x 디바이스 ~ -1, -2, -5, -10
Max. Volt	Single	A/D범위의 최대값을 지정합니다. 유효한 값은 1, 2,

		5, 10 입니다.
--	--	------------

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_AD_GetDigit

아이콘



Function 개요

주어진 채널에 대하여 A/D 변환을 수행하고 그 값을 정수 값으로 반환합니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch No	I32	A/D 를 수행할 채널 번호 (0 – Based)

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Return	I16	정수형의 A/D 결과값. 이 값의 범위는 각 디바이스의 A/D 분해능에 따라 다르며 그 값은 다음과 같습니다.

		<div>디바이스 - 분해능 - return 값의 범위</div> <hr/> <div>COMI-LX101 - 12 bit - 0 ~ 4095</div> <div>COMI-LX102 - 14 bit - 0 ~ 16383</div> <div>COMI-LX103 - 16 bit - -32768 ~ 32767</div>
--	--	---

COMILX_AD_GetVolt

■ 아이콘



■ Function 개요

주어진 채널에 대하여 A/D 변환을 수행하고 그 값을 Voltage 값으로 반환합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch No	I32	A/D 를 수행할 채널 번호 (0 - Based)

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Return	Single	A/D 결과값

AI-Scan for LX10x Series



AI-Scan for LX10x Series Function 블록은 COMI-LX10x Series DAQ 보드 전용 A/D 스캔 기능을 지원하는 함수들의 그룹입니다. 이 함수들은 사용자가 직접 A/D trigger 를 하지 않고, 디바이스에 내장된 타이머가 일정 주기로 A/D trigger 를 해주고 변환된 A/D 데이터를 특정 버퍼에 저장하는 방식입니다. 이 때 Scan 데이터를 저장하는 버퍼는 환형 버퍼 형식으로 운용되며 사용자는 필요 시에 이 버퍼로부터 데이터를 취하게 됩니다. 이 방식은 **Unlimited Scan** 방식이라고도 불리우며, Single point A/D 방식에 비해 속도가 빠르고 정확한 샘플링 주기를 보장할 수 있습니다. 따라서 이 방식은 고속 A/D 를 할 때 아주 유용하게 사용될 수 있습니다.

AI-Scan 에 관한 자세한 내용은 소프트웨어 매뉴얼의 4.4.2 절을 참조 하십시오.

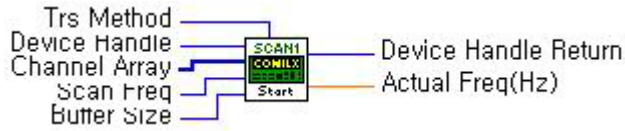
AI-Scan for LX10x Series Function 블록에 속하는 함수(VI)들은 다음의 표와 같습니다.

함수명	각 보드 별 지원 여부								
	LX101	LX102	LX103	LX201	LX202	LX203	LX301	LX401	LX501
COMILX_US1_Start	✓	✓	✓						
COMILX_US1_Stop	✓	✓	✓						
COMILX_US1_CurCount	✓	✓	✓						
COMILX_US1_RetrVOne	✓	✓	✓						
COMILX_US1_RetrVChannel	✓	✓	✓						
COMILX_US1_RetrVBlock	✓	✓	✓						
COMILX_US1_ReleaseBuf	✓	✓	✓						

Unlimited A/D Scan 1 함수 리스트 및 각 보드 별 지원 여부

COMILX_US1_Start

■ 아이콘



■ Function 개요

이 함수는 Unlimited scan 기능을 시작합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel Array	I32 Array	A/D scan을 수행할 채널 리스트를 담고 있는 배열
Scan Freq	U32	SCAN 주파수(Hz). 이 값은 SCAN과 SCAN사이의 시간차를 결정합니다.
Buffer Size	U32	스캔 데이터를 저장할 환형버퍼의 크기를 지정합니다. 이 값의 단위는 바이트가 아니고 SCAN 횟수입니다. 예를 들어, 채널이 2 채널일때 Buffer Size값을 10240으로 지정하면 실제로는 $2 * 10240 * \text{sizeof}(\text{short}) = 40960$ 바이트 크기의 버퍼가 할당됩니다. 이 값은 1024의 배수로 지정하는 것이 좋습니다.
Trs Method	I32	A/D 디바이스에서 스캔버퍼로 데이터를 전송하는 방식을 지정합니다. 이 값은 다음의 두 값 중의 하나이어야 합니다. ▷ 1 => 1회의 SCAN이 완료될 때마다 인터럽트를 발생시켜 데이터를 전송합니다. 인터럽트의 한계에 따라 Scan frequency가 30 KHz 이상이 되면 이 방식이 적절히 작동하지 않을 수 있습니다. ▷ 2 => 이 방식을 선택하면, A/D 디바이스는 A/D Conversion 데이터를 디바이스에 내장되어 있는 FIFO 메모리에 일단 저장하고, 1024개의 데이터가 쌓이면 인터럽트를 발생시켜 데이터를 1024

		개 단위로 사용자 버퍼에 전송합니다. 이 방식은 고속 A/D scan 시에 적합합니다.
--	--	--

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Actual Freq(Hz)	I32	실제로 설정되는 스캔 주파수를 Hz단위로 반환합니다. 사용자가 지정한 스캔 주파수와 실제로 설정되는 스캔 주파수는 약간의 차이가 있을 수 있습니다. 이 값이 0 보다 작으면 수행도중 에러가 발생 하였음을 의미합니다.

COMILX_US1_Stop

아이콘



Function 개요

Unlimited Scan 을 종료합니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
ReleaseBuf	U8	COMILX_US1_Start 가 수행될 때 할당되었던 스캔 버퍼를 메모리 해제 시킬 것인지를 결정합니다. 만일 이 값을 0 으로 지정하면 후에 반드시 COMILX_US1_ReleaseBuf 를 사용하여 버퍼를 해제하여야 합니다. 이 값을 1로 지정하면 COMILX_US1_ReleaseBuf 함수를

		수행할 필요가 없습니다.
--	--	---------------

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

COMILX_US1_CurCount

■ 아이콘



■ Function 개요

이 함수는 현재까지 수행된 SCAN 횟수를 반환합니다. 사용자는 버퍼에서 데이터를 취할 때에 이 함수를 참조하면 최근 스캔된 데이터의 위치를 알아낼 수 있습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Scan Count	U32	현재까지 수행된 총 SCAN 횟수

COMILX_US1_RetrVOne

■ 아이콘



■ Function 개요

이 함수는 스캔 버퍼에서 하나의 데이터를 취할 때 사용됩니다.

■ Parameters

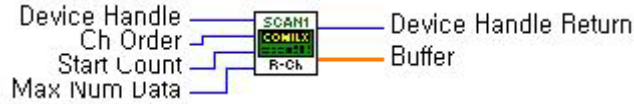
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch Order	U8	데이터를 취하기 원하는 채널의 채널 리스트상의 순서 (0 based)입니다. 이 값은 채널 번호가 아님을 주의 하여야 합니다. • 채널 리스트 - COMILX_US1_Start의 parameter 참조
Scan Count	U32	원하는 데이터의 Scan count.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Voltage	Single	Ch Order와 Scan Count에 의해 지정된 데이터를 Voltage 형식으로 반환합니다.

COMILX_US1_RetrChannel

■ 아이콘



■ Function 개요

이 함수는 A/D Scan 채널 중에서 하나의 채널에 대한 데이터 블록을 Voltage 값으로 환산하여 전달합니다. 데이터 블록은 사용자가 지정한 Start Count 에서부터 Max Num Data 에서 지정한 수만큼 의 크기를 가집니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch Order	U8	데이터를 취하기 원하는 채널의 채널 리스트상의 순서 (0 based)입니다. 이 값은 채널 번호가 아님을 주의 하여야 합니다. • 채널 리스트 - COMILX_US1_Start의 parameter 참조
Start Count	U32	전달 받고자 하는 데이터 블록의 시작 Scan count.
Max Num Data	I32	전달 받고자 하는 데이터 블록의 크기(데이터 수)를 지정 합니다. 이 값이 양수이면 Start Count부터 이후에 스캔된 데이터 중 Max Num Data에서 지정한 수만큼 데이터를 전달합니다. 이 값이 음수이면 Start Count부터 이전에 스캔된 데이터 중 Max Num Data에서 지정한 수만큼 데이터를 전달합니다.

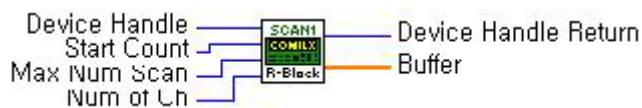
■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Buffer	Single	스캔버퍼로부터 사용자가 지정한 조건에 의하여 취해

	Array	진 A/D 데이터 블록. A/D 데이터는 Voltage값으로 취해집니다.
--	-------	--

COMILX_US1_RetrVBlock

■ 아이콘



■ Function 개요

이 함수는 A/D Scan 전 채널에 대한 데이터를 사용자가 지정하는 버퍼에 전달합니다. 전달되는 데이터 블록의 시작 위치는 Start Count 에 의해 결정되며, 그 크기는 Max Num Scan 에 의해 결정됩니다. (데이터 블록의 실제 크기는 Max Num Scan * 채널수가 됩니다.)

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Start Count	U32	전달 받고자 하는 데이터 블록의 시작 Scan count.
Max Num Scan	I32	전달 받고자 하는 데이터 블록의 크기(스캔 횟수)를 지정 합니다. 이 값이 <u>양수</u> 이면 Start Count부터 이후에 스캔된 데이터 중 Max Num Data에서 지정한 수만큼 데이터를 전달합니다. 이 값이 <u>음수</u> 이면 Start Count부터 이전에 스캔된 데이터 중 Max Num Data에서 지정한 수만큼 데이터를 전달합니다.
Num of Ch	U8	스캔 하고자 하는 채널의 개수입니다. 이 값과 Max Num Scan 값에 의해 데이터 블록의 실제 크기가 결정 됩니다.

■ Return 값

Parameters	Type	Description
------------	------	-------------

Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Buffer	Single Array	스캔버퍼로부터 사용자가 지정한 조건에 의하여 취해진 A/D 데이터 블록. A/D 데이터는 Voltage값으로 취해집니다. 저장된 데이터 블록의 구성은 소프트웨어 매뉴얼 4.4.2절의 COMILX_US1_RetrVBlock 단원을 참조하십시오.

COMILX_US1_ReleaseBuf

■ 아이콘



■ Function 개요

COMILX_US1_Start 가 수행될 때 할당되었던 스캔 버퍼를 메모리 해제 시킵니다. 이 함수는 COMILX_US1_Stop 함수가 호출 되기 전에 수행되어서는 안되며, COMILX_US1_Stop 함수를 호출할 때 두 번째 parameter 를 1로 지정 하였을 때는 사용하지 않아도 됩니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

AI-Scan for LX20x Series



AI-Scan for LX20x Series Function 블록은 COM1-LX20x Series DAQ 보드 전용 A/D 스캔 기능을 지원하는 함수들의 그룹입니다. 이 함수들은 사용자가 직접 A/D trigger 를 하지 않고, 디바이스에 내장된 타이머가 일정 주기로 A/D trigger 를 해주고 변환된 A/D 데이터를 DMA 를 통하여 전송함으로써 고속으로 A/D 를 수행할 수 있도록 하는 기능입니다.

AI-Scan for LX20x Series Function 블록에 속하는 함수(VI)들은 다음의 표와 같습니다.

함수명	각 보드 별 지원 여부								
	LX101	LX102	LX103	LX201	LX202	LX203	LX301	LX401	LX501
COM1LX_US2_SetTriggerEvent				✓	✓	✓			
COM1LX_US2_Start				✓	✓	✓			
COM1LX_US2_Stop				✓	✓	✓			
COM1LX_US2_Acquire				✓	✓	✓			
COM1LX_US2_Resume				✓	✓	✓			
COM1LX_US2_IsBufFull				✓	✓	✓			
COM1LX_US2_RetrVChannel				✓	✓	✓			
COM1LX_US2_ChangeScanFreq				✓	✓	✓			
COM1LX_US2_DmaCount				✓	✓	✓			
COM1LX_US2_ReleaseBuf				✓	✓	✓			

Unlimited A/D Scan 2 함수 리스트 및 각 보드 별 지원 여부

LX20x 전용 AI-Scan 기능이 운용되는 기본 메커니즘을 간단히 설명드리면 다음과 같습니다.

단계 1. 사용자는 **COM1LX_US2_Start** 함수를 수행하여 스캔 채널, 스캔 주파수, 스캔 버퍼의 크기 등을 설정하여 스캔을 시작합니다. Frame Scan (소프트웨어 매뉴얼 4.4.3 절 참조) 의 경우 실제 데이터 획득은 **COM1LX_US2_Acquire**, 혹은 **COM1LX_US2_Resume** 함수에 의해 이루어집니다.

(**COM1LX_US2_Acquire** 함수는 Scan 의 편의를 위해 만들어진 함수 입니다. 이 함수는 **COM1LX_US2_Resume** 함수와 **COM1LX_US2_IsBufFull** 함수를 포함합니다. 이 함수의 사용법은 Labview Examples 중 **UnlimitedScan2_3.vi** 을 참고 하십시오.)

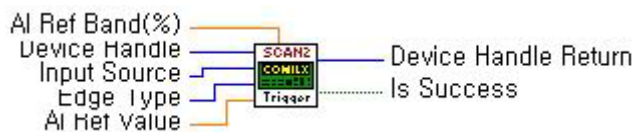
단계 2. **COMILX_US2_Acquire** 함수 등을 수행하여 A/D 스캔을 수행합니다. 즉, 앞에서 설정한 주파수에 따라서 각 채널에 대하여 A/D 샘플링을 수행하고 설정된 스캔 버퍼 크기만큼 데이터를 스캔 버퍼에 저장합니다.

단계 3. **COMILX_US2_RetrvChannel** 함수를 수행하여 스캔 버퍼에 저장된 각 채널의 데이터를 꺼내어 필요한 처리(Plotting, 저장 등)를 합니다.

단계 4. 필요한 횟수만큼 단계 1 과 단계 2 의 작업을 수행합니다. 필요한 작업이 모두 완료되면 **COMILX_US2_Stop** 함수를 수행하여 AI-SCAN 을 종료합니다.

COMILX_US2_SetTriggerEvent

■ 아이콘



■ Function 개요

이 함수는 Trigger Event를 사용할 것인지를 결정하고, Trigger Event신호의 종류와 운용 방법을 설정합니다. Trigger Event는 A/D 스캔을 시작하는 신호를 의미합니다. Trigger Event를 사용하지 않는 경우에는 A/D 스캔 시작 함수를 호출함과 동시에 A/D 스캔이 시작됩니다. 그러나 Trigger Event를 사용하면 A/D 스캔 시작 함수를 호출하여도 외부에서 Trigger Event신호가 입력되기 전까지 스캔을 시작하지 않고 Trigger Event 신호가 발생할 때까지 기다립니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Input Source	U8	Trigger Event 신호로 사용되는 신호원(Signal Source)을 설정합니다. 이 값은 다음 중 하나의 값이어야 합니다. ▷ 0 => (TS_NONE) 이 값을 지정하면 Trigger Event를 사용하지 않음을 의미합니다. 즉,

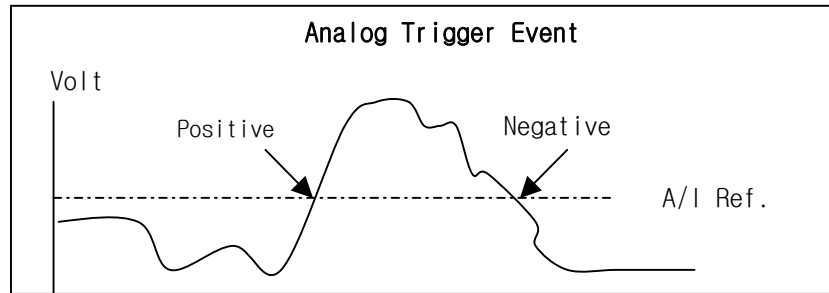
		<p>COMILX_US2_Acquire함수가 호출됨과 동시에 스캔을 시작합니다.</p> <p>▷ 1 => (TS_ANALOG) 이 값을 지정하면 A/D CH0의 신호를 Trigger Event의 신호원으로 사용합니다.</p> <p>▷ 2 => (TS_DIGITAL) 이 값을 지정하면 Digital 신호(ON/OFF)가 Trigger Event의 신호원으로 사용됩니다. COMI-LX20x 디바이스는 Trigger Event 전용 Digital Input단자를 제공합니다. 자세한 사항은 H/W매뉴얼을 참조하십시오.</p>
Edge Type	U8	<p>Trigger Event 신호원에서 발생하는 신호가 Trigger Event로서 동작하기 위한 신호의 상태를 설정합니다. 이 값은 다음 중 하나의 값이어야 합니다. 자세한 내용은 아래의 참고 사항을 참조하십시오.</p> <p>▷ 0 => Positive Edge</p> <p>▷ 1 => Negative Edge</p>
AI Ref Value	Single	<p>Trigger 신호의 기준 값입니다. 이 값은 Input Source가 1, 즉 Analog Trigger 로 지정되었을 때만 의미를 갖습니다. 자세한 내용은 아래의 참고 사항을 참조하십시오.</p>
AI Ref Band(%)	Single	<p>유효한 Trigger신호를 검출하기 위하여 A/I Reference의 Band를 설정하기 위한 값입니다. 이 값의 단위는 A/I Range 백분율(%)입니다. 예를 들어, 만일 Input Range가 -10 ~ +10 이고, 이 값이 1인 경우에는 Range 크기가 20이므로 Band 크기는 $20 * 0.01 = 0.2 \text{ Volt}$가 됩니다. 이 값은 Input Source가 1, 즉 Analog Trigger 로 지정되었을 때만 의미를 갖습니다. 자세한 내용은 아래의 참고 사항을 참조 하십시오.</p>

■ Return 값

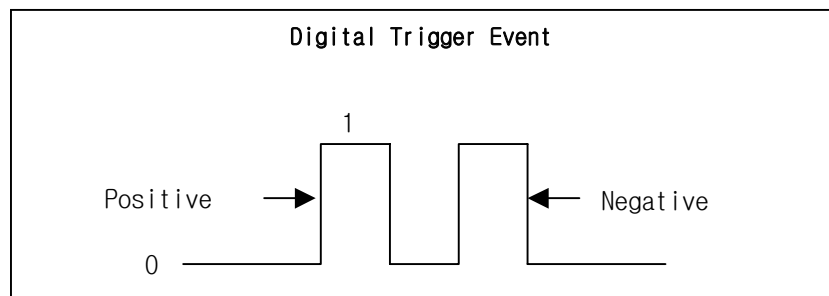
Parameters	Type	Description
Device Handle Return	U32	<p>디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.</p>
Is Success	BOOL	<p>함수의 실행 성공 여부. 0이면 실패 1이면 성공</p>

■ 참고 사항

1. 컴퓨터가 처음 부팅될 때에 COMI-LX20x 디바이스는 기본적으로 Trigger Event를 사용하지 않는 것으로 설정됩니다.
2. **Edge Type**을 그림과 함께 설명하면 다음과 같습니다.



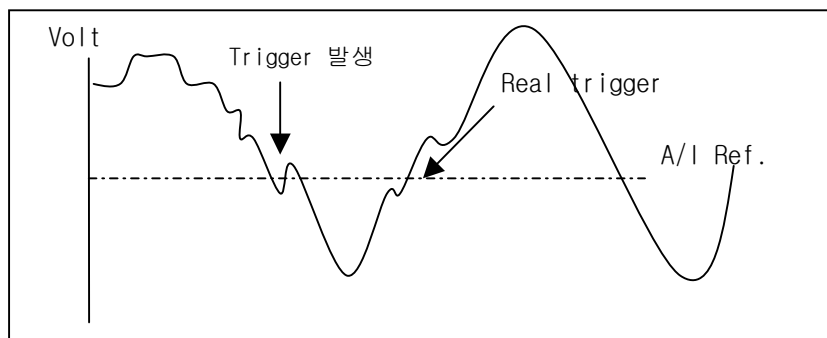
[그림 1] Analog Trigger Event시에 Edge Type의 종류



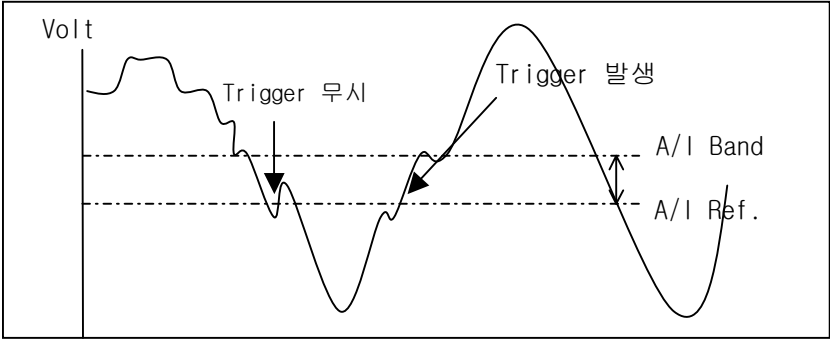
[그림 2] Digital Trigger Event시에 Edge Type의 종류

3. Analog Input Reference Band의 역할

Analog Input Reference Band는 Noise등에 의하여 발생할 수 있는 잘못된 Trigger를 방지하기 위한 것입니다. 예를 들어 [그림 3]과 같이 Positive Edge Trigger 모드에서 Negative Edge순간임에도 불구하고 Noise의 영향에 의하여 순간적으로 Positive Edge가 발생할 수 있습니다. 이 때에는 원하지 않는 Trigger가 발생하게 됩니다. 그러나 [그림 4]와 같이 Band를 지정하면 A/I Reference에서뿐만 아니라 Band limit값에서도 Positive Edge가 발생 해야만 진정한 Trigger Event로 간주되므로 잘못된 Trigger를 방지할 수 있습니다.



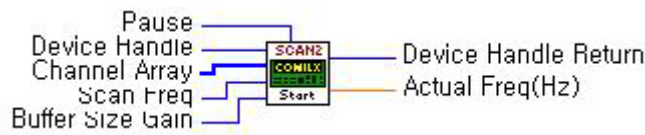
[그림 3] Positive Edge 방식에서 Analog Input Reference Band 를 사용하지 않아 잘못된 Trigger 가 발생되는 예



[그림 4] Positive Edge 방식에서 Analog Input Reference Band 를 사용함으로써 잘못된 Trigger 의 발생을 방지하는 예

COMILX_US2_Start

■ 아이콘



■ Function 개요

이 함수는 LX20x Series 전용 AI-Scan 기능을 시작합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel Array	I32 Array	A/D scan을 수행할 채널 리스트를 담고 있는 배열
Scan Freq	U32	SCAN 주파수(Hz)를 설정합니다.

Buffer Size Gain	U16	이 값은 스캔 버퍼의 크기를 결정합니다. 그러나 이 값이 스캔 버퍼의 직접적인 크기를 지정하지는 않습니다. 실제 버퍼의 크기는 $Buffer\ Size\ Gain * 4096$ 개의 데이터를 저장할 수 있는 크기로 할당됩니다(여기서 4096값은 한번의 DMA가 수행될 때 전송되는 데이터 개수입니다). 예를 들어 채널 당 4096개의 데이터를 저장할 수 있는 크기로 버퍼를 할당하려면 이 값을 1로 하여야 하며, 8192개의 데이터를 저장할 수 있도록하려면 이 값을 2로 하여야 합니다.
Pause	U8	이 값은 스캔 버퍼에 데이터가 꽉 찬 경우에 스캔을 일시 중지할 것인지를 결정합니다. 이 값을 1로 하면 Frame Scan, 0으로 하면 Continuous Scan 방식으로 운용됩니다. 자세한 사항은 소프트웨어 매뉴얼 4.4.3 절을 참조하십시오.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Actual Freq	Double	실제로 설정되는 스캔 주파수를 Hz단위로 반환합니다. 사용자가 지정한 스캔 주파수와 실제로 설정되는 스캔 주파수는 약간의 차이가 있을 수 있습니다. 이 값이 0보다 작으면 수행도중 에러가 발생하였음을 의미합니다.

■ 참고

Buffer Size Gain 및 Pause 값을 설정할 때 주의 사항

① COMI-LX20x 디바이스는 각 채널당 8192 개의 데이터를 저장할 수 있는 FIFO 메모리를 가지고 있습니다. 따라서 Buffer Size Gain 값을 2로 하고 Pause 을 TRUE 로 하면 속도와 상관없이 8 K의 연속 데이터를 얻을 수 있습니다.

② (채널수 * 스캔 주파수) 가 5 MHz 보다 큰 경우에는 Buffer Size Gain 을 2로하고 Pause 을 TRUE 로 하는 것이 좋습니다. 이는 DMA 속도가 스캔속도보다 느리게 되어 연속적으로 데이터를 전송할 수 없기 때문입니다.

③ CH0 와 CH3 만을 사용하는 경우에 Buffer Size Gain 을 4 로 하고 Pause 을 TRUE 로 하면 자동적으로 2 개의 FIFO 를 각 채널에 할당하는 CASCADE 방식으로 운용하여 스캔 속도와 관계없이 16384 (16K)개의 연속 데이터를 얻을 수 있습니다.

COMILX_US2_Stop

■ 아이콘



■ Function 개요

LX20x Series 전용 AI-Scan 기능을 종료합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
ReleaseBuf	U8	COMILX_US2_Start 가 수행될 때 할당되었던 스캔 버퍼를 메모리 해제 시킬 것인지를 결정합니다. 만일 이 값을 0으로 지정하면 후에 반드시 COMILX_US2_ReleaseBuf 를 사용하여 버퍼를 해제하여야 합니다. 이 값을 1로 지정하면 COMILX_US2_ReleaseBuf 함수를 수행할 필요가 없습니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

COMILX_US2_Acquire

■ 아이콘



■ Function 개요

이 함수는 Frame Scan 의 편의를 위해 만들어진 함수 입니다. 이 함수는 COMILX_US2_Resume 함수와 COMILX_US2_IsBufFull 함수를 포함하며 COMILX_US2_Start 함수에서 설정한 환경을 바탕으로 실제 A/D 변환을 시작하여 스캔 버퍼에 저장합니다. 스캔채널로 지정된 모든 채널에 대하여 지정한 수의 데이터가 획득되었으면 A/D 변환을 멈추고 반환됩니다. 여기서 “지정한 수”의 의미는 스캔 버퍼의 크기를 의미하며 COMILX_US2_Start 함수의 ‘Buffer Size Gain’ parameter 의 설명을 참조하십시오.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_US2_Resume

■ 아이콘



■ Function 개요

이 함수는 일시 중지된 A/D 스캔을 재개하여 줍니다. COMILX_US2_Start 함수에서 Pause parameter 값을 1로 하였을 경우, 즉 Frame Scan의 경우에는 스캔 버퍼에 데이터가 다 차게 되면 스캔이 일시 중지됩니다. 이 때 사용자는 필요에 따라 스캔 데이터를 처리하고 이 함수를 이용하여 스캔을 재개할 수 있습니다.

■ Parameters

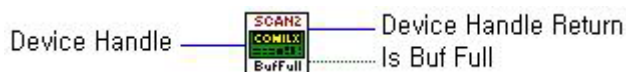
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_US2_IsBufFull

■ 아이콘



■ Function 개요

이 함수는 지정한 크기(개수)의 스캔 버퍼에 데이터가 다 찼는지를 알려 주는 함수입니다. 이것은 COMILX_US2_Start 함수에서 Pause parameter 값을 1로 하였을 경우, 즉 Frame Scan을 할 경우에만 사용하는 함수입니다.

■ Parameters

Parameters	Type	Description
------------	------	-------------

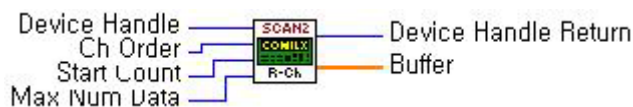
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
---------------	-----	---

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

COMILX_US2_RetrVChannel

■ 아이콘



■ Function 개요

이 함수는 AI-Scan 채널 중에서 하나의 채널에 대한 데이터 블록을 Voltage 값으로 환산하여 전달합니다. 데이터 블록은 사용자가 지정한 Start Count 에서부터 Max Num Data 에서 지정한 수 만큼이 됩니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch Order	U8	데이터를 취하기 원하는 채널의 채널 리스트상의 순서 (0 based)입니다. 이 값은 채널 번호가 아님을 주의 하여야 합니다. • 채널 리스트 - COMILX_US2_Start의 parameter 참조
Start Count	U32	스캔버퍼에서 취하고자 하는 데이터의 시작위치를 지정합니다. 이 값은 1 based로 스캔 버퍼에 있는 맨 처

		음 데이터부터 취하고자 한다면 이 값을 1로 지정 하여야 합니다.
Max Num Data	U32	전달 받고자 하는 데이터 블록의 크기(데이터 수)를 지정 합니다. 만일 Start Count이후에 현재까지 스캔 된 데이터 수가 이 값보다 작으면, 현재 스캔된 데이터까지만 전달합니다. 이 값이 음수이면 Start Count 부터 이전에 스캔된 데이터 중 Max Num Data 에서 지정한 수만큼 데이터를 전달합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행 되어야 할 순서를 명확히 해줄 수 있습니다.
Buffer	Single Array	스캔버퍼로부터 사용자가 지정한 조건에 의하여 취해진 A/D 데이터 블록. A/D 데이터는 Voltage값으로 취해집니다.

COMILX_US2_ChangeScanFreq

■ 아이콘



■ Function 개요

이 함수는 스캔이 진행되는 중에 스캔 주파수를 변경합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Scan Frequency	U32	변경할 SCAN 주파수(Hz)를 설정합니다.

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Scan Frequency	Double	실제로 설정되는 스캔 주파수를 Hz단위로 반환합니다. 사용자가 지정한 스캔 주파수와 실제로 설정되는 스캔 주파수는 약간의 차이가 있을 수 있습니다. 이 값이 0 보다 작으면 수행도중 에러가 발생하였음을 의미합니다.

COMILX_US2_DmaCount

아이콘



Function 개요

이 함수는 스캔이 시작된 후에 현재까지 몇 회의 DMA 데이터 블록이 전송됐는지를 알려주는 함수입니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되

		어야 할 순서를 명확히 해줄 수 있습니다.
Dma Count	U16	현재까지 진행된 DMA 전송 횟수를 반환합니다. 1회의 DMA 전송에 의해 각 채널당 4096 개의 데이터가 전송 되므로 실제로 획득된 데이터수는 $4096 * \text{Dma Count}$ 가 됩니다.

COMILX_US2_ReleaseBuf

■ 아이콘



■ Function 개요

COMILX_US2_Start 가 수행될 때 할당되었던 스캔 버퍼를 메모리 해제 시킵니다. 이 함수는 COMILX_US2_Stop 함수가 호출 되기 전에 수행되어서는 안되며, COMILX_US2_Stop 함수를 호출할 때 두 번째 parameter 를 1로 지정 하였을 때는 사용하지 않아도 됩니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

Analog Out



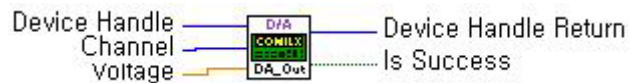
Analog Out Function 블록은 Analog Output 에 관련된 함수들을 모아 놓은 것입니다. COMIDAS 에서는 두 가지 형태의 Analog Output 기능이 있습니다. 첫 번째는 일반적인 Analog Output 기능으로써 사용자가 지정한 전압을 출력하는 기능입니다. 두 번째는 Waveform Generation 기능입니다. Waveform Generation 기능은 Sine Wave 또는 Square Wave 등과 같이 사용자가 지정하는 주기성을 가지는 신호를 자동으로 생성해주는 기능입니다.

함수명	각 보드 별 지원 여부								
	LX101	LX102	LX103	LX201	LX202	LX203	LX301	LX401	LX501
COMILX_DA_Out	✓	✓	✓				✓		
COMILX_WFM_Start	✓	✓	✓				✓		
COMILX_WFM_Stop	✓	✓	✓				✓		
COMILX_WFM_RateChange	✓	✓	✓				✓		
COMILX_WFM_GetCurPos	✓	✓	✓				✓		
COMILX_WFM_GetCurLoops	✓	✓	✓				✓		

아날로그 출력 공통 함수 리스트 및 각 보드 별 지원 여부

COMILX_DA_Out

■ 아이콘



■ Function 개요

사용자가 지정한 Analog Output 채널에 지정한 Voltage 를 출력합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

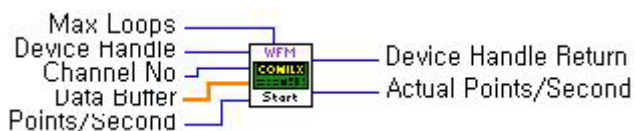
Channel	U8	Analog Output 채널 번호 (0 – Based)
Voltage	Single	원하는 출력 Voltage

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 성공 여부. 0 – 실패, 1 – 성공

COMILX_WFM_Start

아이콘



Function 개요

Waveform Generation 을 시작합니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	U8	Analog Output 채널 번호 (0 – Based)
Data Buffer	Single Array	Waveform 데이터를 담은 버퍼. 예를 들어 -5V, 5V 의 Square wave 를 출력 하려면 Array 에 -5 와 5 를 넣어 주면 됩니다.
Points/Second	U32	Waveform Generation의 주기를 결정합니다. 이 값은 Points/Second입니다. 예를 들어 100개의 데이터로 한 주기를 구성하였다면 10Hz의 신호를 만들기 위해서는

		이 값이 1000이 되어야 합니다.
Max Loops	132	이 값이 0보다 크면 생성되는 Wave 신호의 수를 제한합니다. 이 값이 0이면 COMILX_WFM_Stop()함수가 수행되기 전까지 계속하여 Wave 신호를 생성합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Actual Points/ Second	132	실제로 설정되는 Points/Second를 반환합니다. 사용자가 지정한 PPS와 실제로 설정되는 PPS는 약간의 차이가 있을 수 있습니다. 이 값이 0보다 작으면 수행도중 에러가 발생하였음을 의미합니다.

COMILX_WFM_Stop

■ 아이콘



■ Function 개요

Waveform Generation 을 종료합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	U8	Analog Output 채널 번호 (0 – Based)

■ Return 값

Parameters	Type	Description
------------	------	-------------

Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
-------------------------	-----	---

COMILX_WFM_RateChange

■ 아이콘



■ Function 개요

Waveform Generation 이 진행되고 있는 중에 주파수(PPS)를 변경합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	U8	Analog Output 채널 번호 (0 – Based)
Points/Sec	U32	Waveform Generation의 주기를 결정합니다. 이 값은 Points/Second입니다. 예를 들어 100개의 데이터로 한 주기를 구성하였다면 10Hz의 신호를 만들기 위해서는 이 값이 1000이 되어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Actual Points/ Second	I32	실제로 설정되는 Points/Second를 반환합니다. 사용자가 지정한 PPS와 실제로 설정되는 PPS는 약간의 차이가 있을 수 있습니다. 이 값이 0보다 작으면 수행도

		중 에러가 발생하였음을 의미합니다.
--	--	---------------------

COMILX_WFM_GetCurPos

■ 아이콘



■ Function 개요

현재 출력되고 있는 주기 데이터의 위치를 반환합니다. 즉, 현재 출력되고 있는 데이터 포인트가 주기 데이터의 몇 번째 데이터인지를 알려줍니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	U8	Analog Output 채널 번호 (0 – Based)

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Current Data Index	I32	현재 출력되고 있는 주기 데이터의 인덱스(Index).

COMILX_WFM_GetCurLoops

■ 아이콘



■ Function 개요

이 함수는 COMILX_WFM_Start 함수의 'Max Loops' parameter 를 양의 값으로 설정하였을 때만 사용 가능한 함수입니다. 이 함수는 현재 남아있는 Wave 신호의 주기 수를 반환합니다. 예를 들어 Max Loops 를 1000 으로 하였을 때 이 함수가 100 을 반환한다면 현재까지 900 회의 Wave 신호가 발생하였으며, 100 회의 Wave 신호가 남았음을 의미합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	U8	Analog Output 채널 번호 (0 – Based)

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Remained loops	I32	현재 남아있는 Wave 신호의 출력 횟수. 이 값이 0이면 지정한 횟수의 Wave 신호가 모두 출력되었음을 의미합니다.

Counter 32bit



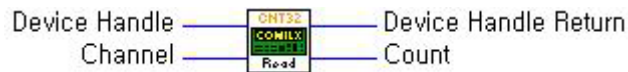
Counter 32bit 블록은 32bit 카운터와 관련된 함수를 모아 놓은 것입니다. 카운터는 펄스 신호를 카운트 하는 데 사용됩니다.

함수명	각 보드 별 지원 여부								
	LX101	LX102	LX103	LX201	LX202	LX203	LX301	LX401	LX501
COMILX_ReadCounter32	✓	✓	✓				✓	✓	
COMILX_ClearCounter32	✓	✓	✓				✓	✓	

Counter 32bit 에 관련된 함수 리스트 및 각 보드 별 지원 여부

COMILX_ReadCounter32

■ 아이콘



■ Function 개요

이 함수는 지정한 카운터 채널의 카운트 값을 읽어옵니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호, 채널 번호는 0 부터 시작합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

Count	U32	지정한 채널의 Count 값
-------	-----	-----------------

COMILX_ClearCounter32

■ 아이콘



■ Function 개요

이 함수는 지정한 카운터 채널의 카운트 값을 0 으로 리셋(reset) 하여 줍니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호, 채널 번호는 0 부터 시작합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

Digital I/O



Digital I/O Function 블록은 Digital Input / Output 에 관련된 함수들을 모아 놓은 것입니다. 일반적으로 Digital Input 은 스위치(Switch)의 상태를 읽어 들이는 데 사용되고, Digital Output 은 스위치의 상태를 제어 하는 데 사용됩니다.

함수명	각 보드 별 지원 여부					
	LX10x	LX20x	LX301	LX401	LX402	LX501
COMILX_DIO_SetUsage	✓	✓	✓	✓		
COMILX_DI_GetOne	✓	✓	✓	✓		✓
COMILX_DI_GetAll	✓	✓	✓	✓		✓
COMILX_DO_PutOne	✓	✓	✓	✓		✓
COMILX_DO_PutAll	✓	✓	✓	✓		✓

Digital Input/Output에 관련된 함수 리스트 및 각 보드 별 지원 여부

COMILX_DIO_SetUsage

■ 아이콘



■ Function 개요

DIGITAL In/Out 채널의 용도를 지정합니다. COMILX Series 디바이스는 Digital Input/ Output 단자가 따로 구분되어 있지 않고 필요에 따라 용도를 설정할 수 있도록 되어 있습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Usage	U8	디바이스의 용도를 선택합니다. 이 값은 다음의 네 값 중 하나이어야 합니다.

		<p>▷ 0 => 전 채널을 Digital Input 채널로 사용합니다.</p> <p>▷ 1 => 전 채널의 반(앞 채널)은 Digital Input 으로, 나머지 반은 Digital Output 으로 사용합니다.</p> <p>▷ 2 => 전 채널의 반(앞 채널)은 Digital Output 으로, 나머지 반은 Digital Input 으로 사용합니다.</p> <p>▷ 3 => 전 채널을 Digital Output 채널로 사용합니다.</p>
--	--	--

■ Return 값

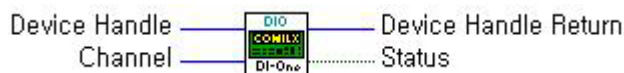
Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고

DIO 단자의 용도 설정에 따라 디지털 입력과 디지털 출력 채널 번호가 어떻게 되는지에 대해 혼동될 수 있습니다. 디지털 입력과 출력의 채널번호는 언제나 각각 0 번부터 시작합니다. 예를 들어 COMI-LX101 보드는 16 채널의 DIO 단자를 제공하는데 이를 DI_D0 로 설정하였다면 DI00 ~ DI07 의 단자는 디지털 입력 CH0 ~ CH7 로 사용되고 DI08 ~ DI015 의 단자는 디지털 출력 CH0 ~ CH7 로 사용됩니다. 반대로 DO_DI 모드로 설정하였다면 DI00 ~ DI07 의 단자는 디지털 출력 CH0 ~ CH7 로 사용되고 DI08 ~ DI015 의 단자는 디지털 입력 CH0 ~ CH7 로 사용됩니다.

COMILX_DI_GetOne

■ 아이콘



■ Function 개요

이 함수는 지정한 Digital Input 한 채널의 Status 를 반환합니다.

■ Parameters

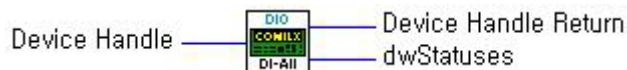
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Digital Input 채널번호. 채널번호는 0부터 시작합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Status	BOOL	지정한 Digital Input 채널의 Status

COMILX_DI_GetAll

■ 아이콘



■ Function 개요

이 함수는 해당 디바이스의 모든 Digital Input 채널의 Status 를 반환합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

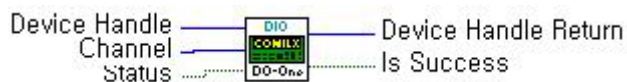
■ Return 값

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device

Return		Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
dwStatuses	U32	32개 채널에 대한 Input Status 를 32 비트 값으로 반환합니다. 각 비트는 각 채널의 On/Off 상태를 나타냅니다. 단, 디바이스에 따라 32 채널 미만의 Digital 채널을 지원하는 경우에는 0 Bit 부터 해당 채널 수 만큼의 비트만 사용하시면 됩니다.

COMILX_DO_PutOne

■ 아이콘



■ Function 개요

이 함수는 지정한 Digital Output 채널에 지정한 Status 로 출력을 내보냅니다.

■ Parameters

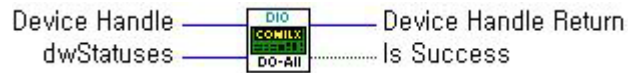
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Digital Output 채널번호. 채널 번호는 0부터 시작합니다.
Status	BOOL	출력 Status. 0 - OFF, 1 - ON.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 성공 여부. 0 - 실패, 1 - 성공

COMILX_DO_PutAll

■ 아이콘



■ Function 개요

이 함수는 해당 디바이스의 모든 Digital Output 채널에 출력을 내보냅니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
dwStatuses	U32	모든 Digital Output 채널의 출력 Status 를 나타내는 32 bit 값. 이 값의 각 비트의 값이 각 채널의 Status 를 나타냅니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter 로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 성공 여부. 0 – 실패, 1 – 성공

Serial Digital I/O



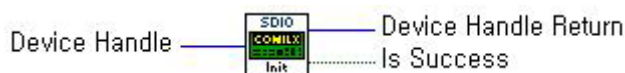
Serial Digital I/O Function 블록은 Serial 통신을 이용한 Digital Input / Output 에 관련된 함수들을 모아 놓은 것입니다. 이 방식은 PC 에 장착되어 시리얼 통신을 관장하는 마스터 보드(COMI-LX402 보드) 와 외부에 설치되어 실제 디지털 입출력을 제어하는 터미널 모듈(COMI-STM4A) 이 RS-422 시리얼 통신으로 연결되어 제어되는 방식입니다. 이 방식은 사용자가 프로그램 상에서 디지털 입출력 명령을 수행하면 마스터 보드가 해당 터미널 모듈에 시리얼 통신을 이용하여 명령을 전달하고 해당 터미널 모듈이 디지털 입출력 명령을 수행하는 메커니즘을 사용합니다. 이 방식은 하나의 마스터 보드에 최대 16 개 까지 터미널 모듈을 확장하여 사용할 수 있다는 큰 장점이 있습니다.

함수명	각 보드 별 지원 여부					
	LX10x	LX20x	LX301	LX401	LX402	LX501
COMILX_SDIO_InitComm					✓	
COMILX_SDIO_CheckModule					✓	
COMILX_SDIO_CheckModule					✓	
COMILX_SDIO_SetDioUsage					✓	
COMILX_SDIO_ReadLowByte					✓	
COMILX_SDIO_ReadHighByte					✓	
COMILX_SDIO_WriteLowByte					✓	
COMILX_SDIO_WriteHighByte					✓	

Serial Digital Input/Output에 관련된 함수 리스트 및 각 보드 별 지원 여부

COMILX_SDIO_InitComm

■ 아이콘



■ Function 개요

COMI-LX402 마스터 보드의 통신 포트를 초기화 합니다. 일반적으로 컴퓨터가 부팅되

면서 통신 초기화는 자동으로 이루어집니다. 따라서 사용자는 통신 초기화를 별도로 하지 않아도 상관은 없으나 프로그램 시작부분에서 통신 초기화를 해주는 것이 좋습니다.

■ Parameters

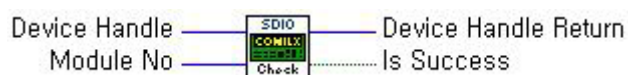
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

COMILX_SDIO_CheckModule

■ 아이콘



■ Function 개요

이 함수는 지정한 주소 값(모듈 번호)을 가지는 COMI-STM4A 디지털 입출력 터미널 모듈이 COMI-LS402 마스터 보드에 현재 연결되어 있는지를 체크하는 함수입니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Module No	18	검색하고자 하는 COMI-STM4A 디지털 입출력 터미널 모듈 번호를 지정합니다. 모듈 번호는 0 ~ 15 까지 지정

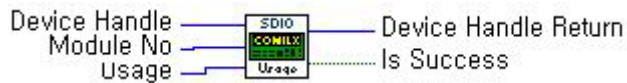
		할 수 있으며 해당 COMI-STM4A 터미널 모듈에서 점퍼로 설정되는 모듈번호와 일치하여야 합니다.
--	--	--

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	지정한 COMI-STM4A 터미널 모듈이 연결되어 있는지 확인

COMILX_SDIO_SetDioUsage

아이콘



Function 개요

이 함수는 지정한 COMI-STM4A 터미널 모듈의 Digital Input/Output 단자의 용도를 설정합니다. 각 COMI-STM4A 터미널 모듈은 16 개의 디지털 입출력 채널을 제공하며 16 채널의 입출력 모드를 4 가지 방법으로 설정할 수 있습니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Module No	18	설정하고자 하는 COMI-STM4A 디지털 입출력 터미널 모듈 번호를 지정합니다. 모듈 번호는 0 ~ 15 까지 지정할 수 있으며 해당 COMI-STM4A 터미널 모듈에서 점퍼로 설정되는 모듈번호와 일치하여야 합니다.
Usage	18	DIO 단자의 용도를 설정합니다. 이 값은 다음의 값 중 하나이어야 합니다.

		▷ 0 => 전 채널을 디지털 입력 채널로 사용합니다. ▷ 1 => Ch0 ~ Ch7 : 디지털 입력, Ch8 ~ Ch 15 : 디지털 출력 ▷ 2 => Ch0 ~ Ch7 : 디지털 출력, Ch8 ~ Ch 15 : 디지털 입력 ▷ 3 => 전 채널을 디지털 출력 채널로 사용합니다.
--	--	--

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

COMILX_SDIO_ReadLowByte

■ 아이콘



■ Function 개요

이 함수는 지정한 COMI-STM4A 터미널 모듈의 0 번 채널부터 7 번 채널 까지의 현재 입력 또는 출력 상태를 읽어 들입니다. 이 함수는 Ch 0 ~ Ch 7 의 채널 그룹이 디지털 입력용으로 설정되었을 때 뿐 아니라 디지털 출력용으로 설정된 경우에도 사용할 수 있습니다. 만일 디지털 출력용으로 설정된 경우에 이 함수를 사용하시면 현재 Ch 0 ~ Ch 7 의 출력 상태를 반환 받을 수 있습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Module No	18	설정하고자 하는 COMI-STM4A 디지털 입출력 터미널 모

		둘 번호를 지정합니다. 모듈 번호는 0 ~ 15 까지 지정할 수 있으며 해당 COMI-STM4A 터미널 모듈에서 점퍼로 설정되는 모듈번호와 일치하여야 합니다.
--	--	--

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Output	BOOL Array	지정한 COMI-STM4A 터미널 모듈의 0 번 채널부터 7 번 채널까지의 현재 입력 또는 출력 상태.

COMILX_SDIO_ReadHighByte

■ 아이콘



■ Function 개요

이 함수는 지정한 COMI-STM4A 터미널 모듈의 8 번 채널부터 15 번 채널까지의 현재 입력 또는 출력 상태를 읽어 들입니다. 이 함수는 Ch 8 ~ Ch 15 의 채널 그룹이 디지털 입력용으로 설정되었을 때 뿐 아니라 디지털 출력용으로 설정된 경우에도 사용할 수 있습니다. 만일 디지털 출력용으로 설정된 경우에 이 함수를 사용하시면 현재 Ch 8 ~ Ch 15 의 출력 상태를 반환 받을 수 있습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Module No	18	설정하고자 하는 COMI-STM4A 디지털 입출력 터미널 모듈 번호를 지정합니다. 모듈 번호는 0 ~ 15 까지 지정할 수 있으며 해당 COMI-STM4A 터미널 모듈에서 점퍼

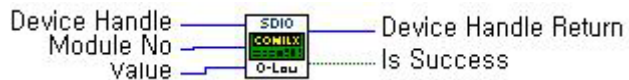
		로 설정되는 모듈번호와 일치하여야 합니다.
--	--	-------------------------

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Output	BOOL Array	지정한 COMI-STM4A 터미널 모듈의 8 번 채널부터 15 번 채널까지의 현재 입력 또는 출력 상태.

COMILX_SDIO_WriteLowByte

아이콘



Function 개요

이 함수는 지정한 COMI-STM4A 터미널 모듈의 Ch 0 ~ Ch 7 에 디지털 출력을 내보냅니다. 이 함수를 사용하기 전에 COMILX_SDIO_SetDioUsage 함수를 사용하여 Ch 0 ~ Ch 7 을 디지털 출력 채널로 설정하여야 합니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Module No	18	설정하고자 하는 COMI-STM4A 디지털 입출력 터미널 모듈 번호를 지정합니다. 모듈 번호는 0 ~ 15 까지 지정할 수 있으며 해당 COMI-STM4A 터미널 모듈에서 점퍼로 설정되는 모듈번호와 일치하여야 합니다.
Value	U8	Ch 0 ~ Ch 7 의 디지털 출력 상태를 지정합니다. 이 값은 8비트 값으로서 각 비트의 상태가 각 채널의 On/Off 상태를 의미합니다.

		예를 들어 Ch 1 만 0n 상태로 하고자 하면 00000010을 입력합니다.
--	--	---

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

■ 참고

특정 채널의 상태만 변경하고자 하는 경우는 COMILX_SDIO_ReadLowByte 함수를 통하여 현재의 출력상태를 읽어 들인 후 원하는 채널에 해당하는 비트만 변경하여 출력시키면 됩니다.

COMILX_SDIO_WriteHighByte

■ 아이콘



■ Function 개요

이 함수는 지정한 COMI-STM4A 터미널 모듈의 Ch 8 ~ Ch 15 에 디지털 출력을 내보냅니다. 이 함수를 사용하기 전에 COMILX_SDIO_SetDioUsage 함수를 사용하여 Ch 8 ~ Ch 15 를 디지털 출력 채널로 설정하여야 합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Module No	18	설정하고자 하는 COMI-STM4A 디지털 입출력 터미널 모듈 번호를 지정합니다. 모듈 번호는 0 ~ 15 까지 지정

		할 수 있으며 해당 COMI-STM4A 터미널 모듈에서 점퍼로 설정되는 모듈번호와 일치하여야 합니다.
Value	U8	Ch 8 ~ Ch 15 의 디지털 출력 상태를 지정합니다. 이 값은 8비트 값으로서 각 비트의 상태가 각 채널의 On/Off 상태를 의미합니다. 예를 들어 Ch 8 만 On 상태로 하고자 하면 00000010 을 입력합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Success	BOOL	함수의 실행 성공 여부. 0이면 실패 1이면 성공

■ 참고

특정 채널의 상태만 변경하고자 하는 경우는 COMILX_SDIO_ReadHighByte 함수를 통하여 현재의 출력상태를 읽어 들인 후 원하는 채널에 해당하는 비트만 변경하여 출력시키면 됩니다.

Motion – General



“Motion General Functions” 함수블록은 Motion Controller 의 초기화 및 Motion Controller 를 사용하는데 있어서 공통적으로 사용되는 함수들을 모아 놓은 것입니다. “Motion General Functions” 블록에 포함된 함수들은 아래의 표와 같습니다.

함수명	각 보드 별 지원 여부					
	LX10x	LX20x	LX301	LX401	LX402	LX50x
COMILX_MC_InitFromFile						✓
COMILX_MC_Reset						✓
COMILX_MC_SetBlockingMode						✓
COMILX_MC_ServoOn						✓
COMILX_MC_GetPosition_C						✓
COMILX_MC_SetPosition_C						✓
COMILX_MC_GetPosition_A						✓
COMILX_MC_SetPosition_A						✓

“Motion General Functions” 블록에 포함된 함수 리스트

COMILX_MC_InitFromFile

■ 아이콘



■ Function 개요

이 함수는 장치 초기화 파일로부터 정보를 전달받아서 모션제어보드를 초기화하는 함수입니다. 모션제어보드는 펄스 입출력 모드 및 각종 I/O 신호들의 환경 등, 초기화 해주어야 할 사항들이 많습니다. 이전 버전의 라이브러리에서는 이러한 환경들을 설정 해주기 위해서 각각의 관련함수들을 수행해주어야 했습니다. 그러나 라이브러리(DLL) V3.1.0.1 이후 버전에서는 이 함수를 이용하여 쉽게 장치초기화를 할 수 있습니다.

장치 초기화 파일은 (주)커미조아의 모션제어 전용 프로그램인 “모션빌더”에서 생성됨

니다. 사용자는 모션빌더의 “Device Init & Config” 메뉴를 통하여 장치의 환경을 설정할 수 있으며, 각 설정값을 파일로 저장할 수 있습니다. 단, 이 기능은 라이브러리(DLL) V3.1.0.1 이후 버전과 모션빌더 V1.1 이후 버전부터 지원됩니다. 라이브러리(DLL) V3.1.0.1 보다 낮은 버전의 라이브러리를 사용하거나, 모션빌더 V1.0 을 사용하시는 사용자께서 이 기능을 사용하시려면 프로그램을 업데이트하셔야 합니다.

■ Parameters

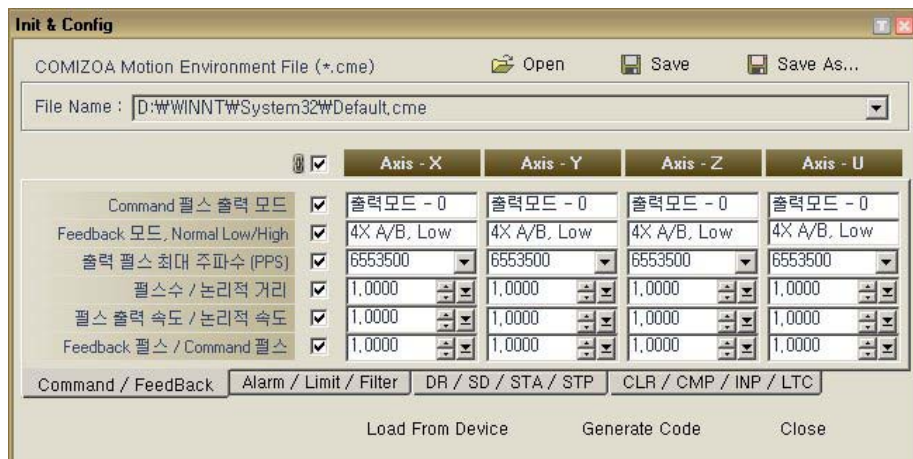
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
CME File Path	String	장치 초기화 파일(.cme)의 경로를 포함한 파일명을 스트링으로 지정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고사항

□ 모션빌더에서 “Device Init & Config”메뉴를 선택하면 아래와 같은 화면이 나타납니다.



[그림 #] 모션빌더의 “Device Init & Config” 화면

사용자는 [그림 #] 화면에서 사용자의 시스템에 맞게 입출력 모드 및 각 신호의 환경

을 설정한후 “Save”버튼을 이용하여 저장하면 지정한 파일에 저장이 됩니다. 이렇게 저장된 파일을 COMILX_MC_InitFromFile() 함수를 이용하여 참조할 수 있습니다.

모션빌더는 윈도우의 시스템폴더에 “Default.cme”의 기본 장치 초기화 파일을 제공합니다. 사용자는 이 기본 파일을 사용하거나 “Save As” 버튼을 이용하여 다른이름으로 초기화 파일을 저장할 수 있습니다.

COMILX_MC_Reset

■ 아이콘



■ Function 개요

이 함수는 모션 제어 보드의 하드웨어와 소프트웨어적인 모든 상태를 리셋합니다. 프로그램 초기와 종료 부분에서는 이 함수를 사용하여 모션을 리셋시켜주는 것이 좋습니다.

■ Parameters

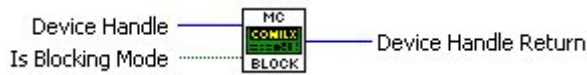
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_SetBlockingMode

■ 아이콘



■ Function 개요

이 함수는 Blocking 모드를 결정합니다. COMILX_MC_Move()와 같은 함수들은 Motion 을 시작해준 후에 내부적으로 while loop 를 돌면서 모션이 완료되는 것을 체크한 후 모션이 완료되면 함수에서 리턴(Return)됩니다. Blocking 이란 while loop 를 수행하게 되면 키보드, 마우스, 타이머 이벤트 등의 윈도우 이벤트가 처리되지 않는 현상을 말합니다.

Blocking 모드를 FALSE 로 하면 COMILX_MC_Move () 등의 함수 내부에서 while loop 를 돌때에 PeekMessage 와 TranslateMessage 등의 API 함수를 수행하여 루프 수행중에도 키보드, 마우스, 타이머 이벤트 등과 같은 윈도우 이벤트나 메시지를 처리할 수 있도록 합니다.

■ Parameters

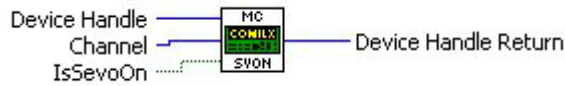
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Is Blocking Mode	BOOL	Blocking 모드를 결정합니다. 이 값을 False로 하면 COMILX_MC_Move 함수등을 사용할때에도 마우스, 키보드등의 이벤트를 처리할 수 있습니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_ServoOn

■ 아이콘



■ Function 개요

지정한 채널(축)의 SERVO-ON 신호를 제어합니다. 서보 드라이버를 사용하실 때는 외부에서 스위치를 이용하여 서보드라이버의 ON/OFF를 제어할 수 있도록 하는데, 이를 SERVO-ON 신호라 합니다. 따라서 서보드라이버를 사용하여 모터를 구동하는 경우에는 반드시 이 함수를 이용하여 SERVO-ON 신호를 Enable 시켜야 합니다.

단, 이 함수는 COMI-LX501 보드에서는 지원하지 않습니다. COMI-LX501에서는 디지털 출력 채널을 이용하여 SERVO-ON/OFF 제어를 하셔야합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
IsServoOn	BOOL	SERVO-ON 제어값. False=> SERVO OFF, True=>SERVO ON

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_GetPosition_C

■ 아이콘



■ Function 개요

이 함수는 현재의 명령 위치(Command Position)를 논리 단위로 반환합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Command Position	Double	현재의 명령 위치(Command Position)를 논리 단위로 반환합니다.

COMILX_MC_SetPosition_C

■ 아이콘



■ Function 개요

이 함수는 명령 위치(Command Position)값을 사용자가 변경할 수 있도록하는 함수로

써, 설정단위는 논리단위입니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Command Position	Double	새로이 설정할 명령 위치(Command Positon)값.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_GetPosition_A

■ 아이콘



■ Function 개요

이 함수는 현재의 실제 위치(Actual Position)를 논리 단위로 반환합니다. 실제 위치는 엔코더 피드백값을 의미합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Actual Position	Double	현재의 실제 위치(Actual Position)를 논리 단위로 반환합니다.

COMILX_MC_SetPosition_A

■ 아이콘



■ Function 개요

이 함수는 명령 위치(Command Position)값을 사용자가 변경할 수 있도록하는 함수로써, 설정단위는 논리단위입니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Command Position	Double	새로이 설정할 명령 위치(Command Positon)값.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되

		어야 할 순서를 명확히 해줄 수 있습니다.
--	--	-------------------------

Motion – Home Return



“Home Return Operations” 함수블록은 Motion Controller의 원점복귀 작업과 관련된 함수들을 모아 놓은 것입니다. 원점 복귀는 모션제어의 대상이 되는 구조물이 원점 위치로 자동 복귀하도록 하는 작업입니다. 원점 복귀 작업이 완료되면 Command Counter, Feedback Counter, Deviation Counter는 자동으로 0으로 초기화됩니다.

원점 복귀 작업을 수행하기 위해서는 ORG(HOME), EZ 및 EL 신호가 참조되는데 이 신호들의 의미 및 작용은 다음과 같습니다.

□ ORG (HOME) 신호

ORG 신호는 구조물이 원점에 복귀했는지를 센서로부터 입력받는 신호입니다. 일반적으로는 근접 센서와 같은 센서들을 이용하여 원점 복귀 여부를 감지하게 됩니다.

□ EZ 신호

엔코더의 제로 펄스 신호를 의미합니다. 이 신호는 원점 복귀 모드에 따라 ORG 신호 또는 EL 신호와 함께 사용되어 보다 정밀한 원점복귀 작업을 수행할 수 있도록 해줍니다.

□ EL 신호

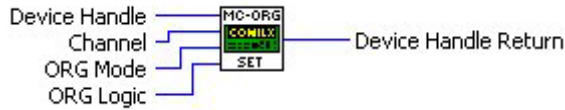
기계적인 리미트(Limit) 신호를 의미합니다. 이 신호는 일반적으로 구조물이 움직일 수 있는 한계점을 감지하기 위해 사용되나 원점 복귀 모드에 따라 ORG 신호의 대용으로도 사용될 수 있습니다. EL 신호는 (+)방향 리미트 신호와 (-)방향 리미트 신호의 두 가지 신호가 있습니다. (+)방향 리미트 신호는 터미널 보드의 ‘+EL’ 단자, 그리고 (-)방향 리미트 신호는 ‘-EL’ 단자를 통하여 COM1-LX50x 보드에 입력되어야 합니다.

함수명	각 보드 별 지원 여부					
	LX10x	LX20x	LX301	LX401	LX402	LX50x
COM1LX_MC_SetHomeConfig						✓
COM1LX_MC_HomeMoveAuto						✓
COM1LX_MC_HomeMoveAutoAll						✓

“Home Return Operations” 블록에 포함된 함수 리스트

COMILX_MC_SetHomeConfig

■ 아이콘



■ Function 개요

이 함수는 원점 복귀 작업을 수행하기 위한 여러가지 환경설정을 수행합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
ORG Mode	I32	원점 복귀 모드 번호를 설정합니다. COMI-LX50x 모션 제어보드는 13가지(0 ~ 12)의 다양한 복귀 모드를 제공합니다. 각 복귀 모드에 대한 자세한 사항은 아래의 참고사항편을 참조하십시오.
ORG Logic	I32	ORG 신호의 Action Level을 설정합니다. 즉 ORG 신호의 레벨(Level)이 High상태일때 ON인지, Low상태일때 ON인지를 설정합니다. 이 값을 0으로 하면 B점점(Normal close) 센서와 로직이 일치하며 이 값을 양수로 하면 A점점(Normal open) 센서와 로직이 일치합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

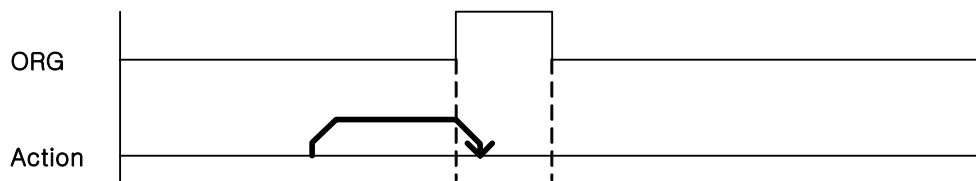
■ 참고사항

원점복귀모드

COMI-LX50x 모션제어보드는 다음과 같이 13 가지의 다양한 원점 복귀 모드를 제공합니다. 원점 복귀 모드는 COMILX_MC_SetHomeConfig()함수를 통하여 설정됩니다. 아래의 그림은 모두 속도모드를 Trapezoidal 모드로 설정한 상태임을 가정하여 그려진 것이며, 만일 Constant 속도 모드로 설정된 경우에는 감속이 없이 즉시 정지하게 됩니다.

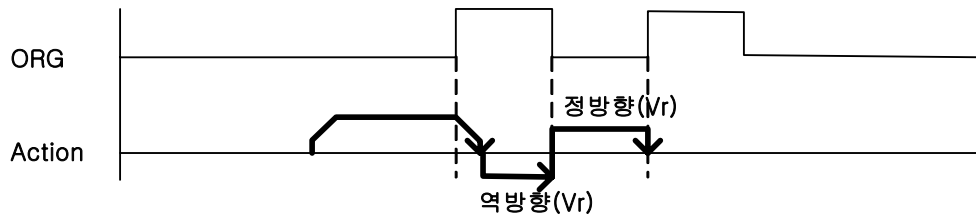
□ MODE 0 : ORG -> Slow down -> Stop

MODE 0에서는 ORG 신호가 OFF 에서 ON 으로 바뀌는 순간에 모션을 감속 후 정지하고 복귀 작업을 종료합니다.



□ MODE 1 : ORG -> Slow down -> Go back -> Go forward -> Stop

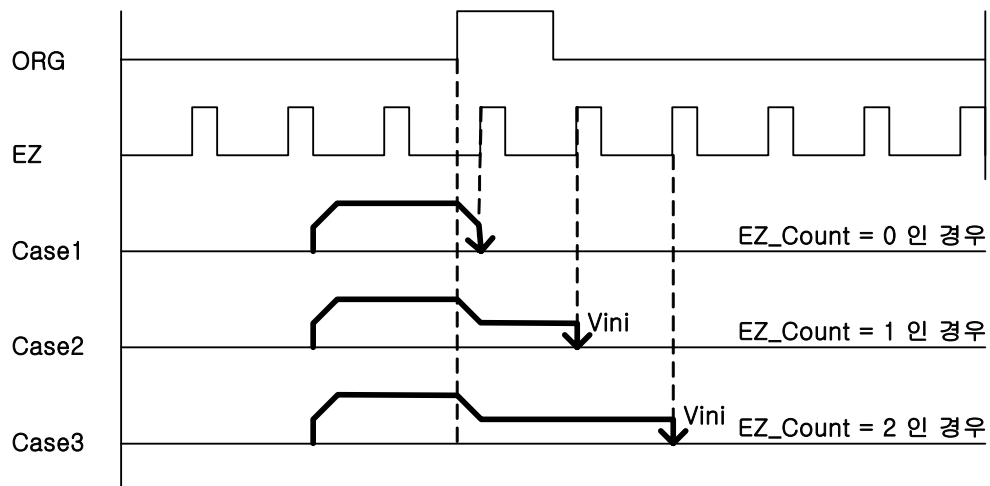
MODE 1에서는 ORG 신호가 OFF 에서 ON 으로 바뀌는 순간에 모션을 감속 후 정지한 후 ORG 신호가 OFF 가 될때까지 Vr (Reverse Speed)의 속도로 역방향 회전을 수행합니다. ORG 신호가 OFF 되는 순간에 다시 Vr 의 속도로 정방향 회전을 수행하다가 ORG 신호가 다시 ON 되는 순간에 복귀작업을 종료합니다.



Vr : Reverse Speed를 의미하며 COMILX_MC_HomeMove 함수 참조

□ MODE 2 : ORG -> Slow down -> Stop on EZ signal

MODE 2에서는 ORG 신호가 OFF 에서 ON 으로 바뀌는 순간에 모션을 감속한 후 초기속도값으로 모션을 진행하다가 EZ 신호에 따라 복귀작업을 종료합니다. COMILX_MC_SetHomeConfig 함수를 통하여 미리 설정된 EzCount 값에 따라 종료하는 시점은 아래와같이 달라집니다.



Case1 : EzCount = 0 인 경우, COMILX_MC_SetHomeConfig 함수 참조

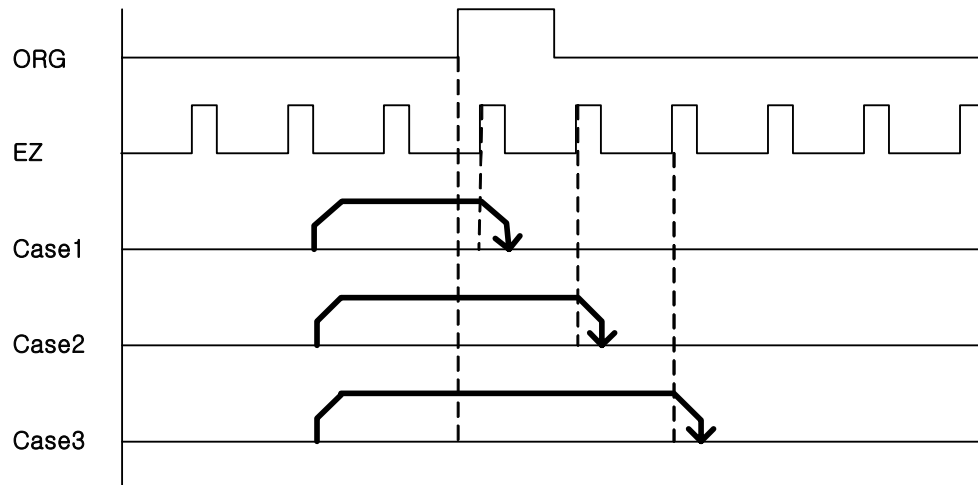
Case2 : EzCount = 1 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Case3 : EzCount = 2 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Vini : 초기속도를 의미하며, COMILX_MC_SetSpeed 함수 참조

□ MODE 3 : ORG -> EZ -> Slow down -> Stop

MODE 3에서는 ORG 신호가 OFF에서 ON으로 바뀌고 난 후 발생하는 EZ 신호에 따라 감속 후 정지합니다. COMILX_MC_SetHomeConfig 함수를 통하여 미리 설정된 EzCount 값에 따라 감속을 시작하는 시점은 아래와 같이 달라집니다.



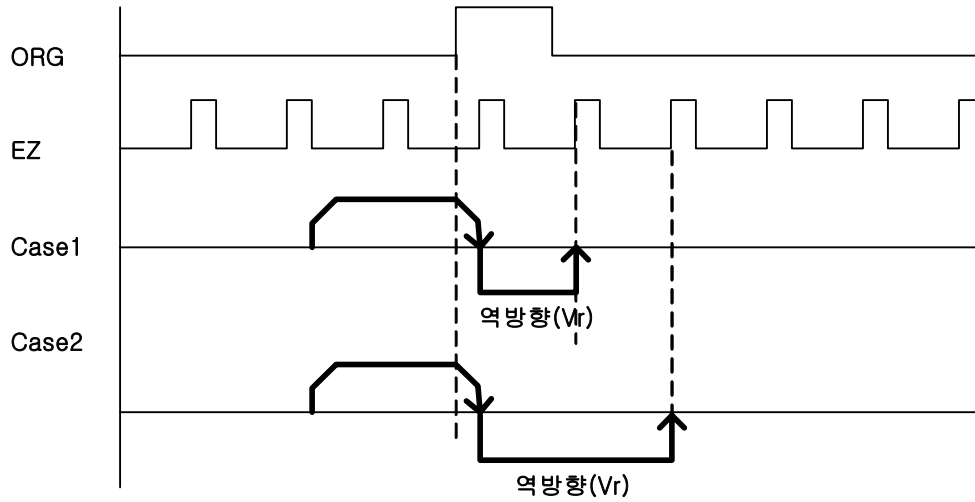
Case1 : EzCount = 0 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Case2 : EzCount = 1 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Case3 : EzCount = 2 인 경우, COMILX_MC_SetHomeConfig 함수 참조

□ MODE 4 : ORG -> Slow down -> Go back at Vr -> Stop on EZ signal

MODE 4에서는 ORG 신호가 OFF에서 ON으로 바뀌는 순간 감속 후 정지합니다. 그리고 Vr의 속도로 역방향 회전한 후 EZ 신호에 따라 복귀 작업을 종료합니다.



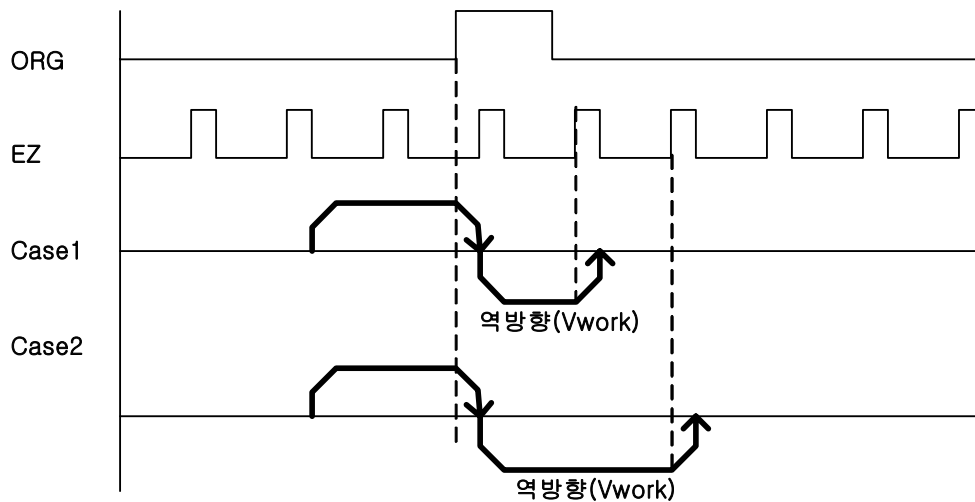
Case1 : EzCount = 1 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Case2 : EzCount = 2 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Vr : Reverse Speed를 의미하며 COMILX_MC_HomeMove 함수 참조

□ MODE 5 : ORG → Slow down → Go back → Accelerate to Vwork → EZ → Slow down → Stop

MODE 5에서는 ORG 신호가 OFF에서 ON으로 바뀌는 순간 감속 후 정지합니다. 그리고 작업 속도까지 가속하여 역방향 회전한 후 EZ 신호에 따라 감속 후 복귀 작업을 종료합니다.

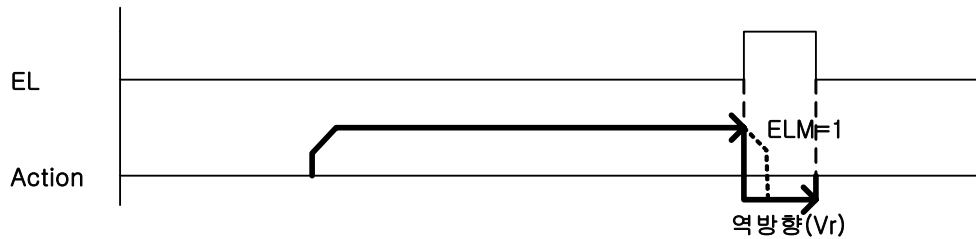


Case1 : EzCount = 1 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Case2 : EzCount = 2 인 경우, COMILX_MC_SetHomeConfig 함수 참조

□ **MODE 6 : EL ON → Stop → Go back at Vr → EL OFF → Stop**

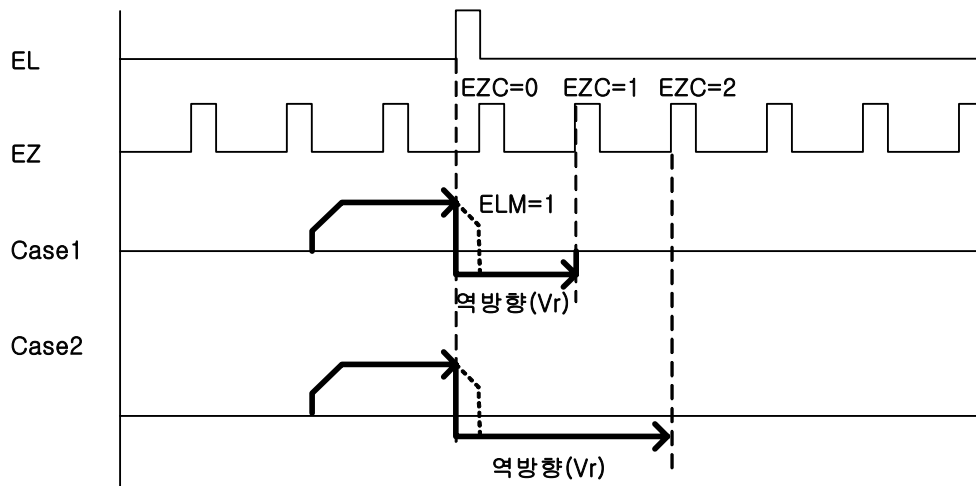
MODE 6에서는 EL 신호가 ON으로 바뀌는 순간 즉시 정지(또는 ELM=1인 경우에 감속후 정지)합니다. 그리고 반대 방향으로 Vr 속도로 회전하다가 EL 신호가 OFF 되는 순간에 복귀 작업을 종료합니다. 여기서 ELM=1은 COMILX_MC_SetMioCfgEL()함수에서 nElMode를 1로 설정했음을 의미합니다.



Vr : Reverse Speed를 의미하며 COMILX_MC_HomeMove 함수 참조

□ **MODE 7 : EL ON → Go back at Vr → Stop on EZ signal**

MODE 7에서는 EL 신호가 ON으로 바뀌는 순간 즉시 정지(또는 ELM=1인 경우에 감속후 정지)합니다. 그리고 반대 방향으로 Vr 속도로 회전하다가 EZ_Count에 따라 복귀작업을 종료합니다. 여기서 ELM=1은 COMILX_MC_SetMioCfgEL()함수에서 nElMode를 1로 설정했음을 의미합니다.



Case1 : EzCount = 1인 경우, COMILX_MC_SetHomeConfig 함수 참조

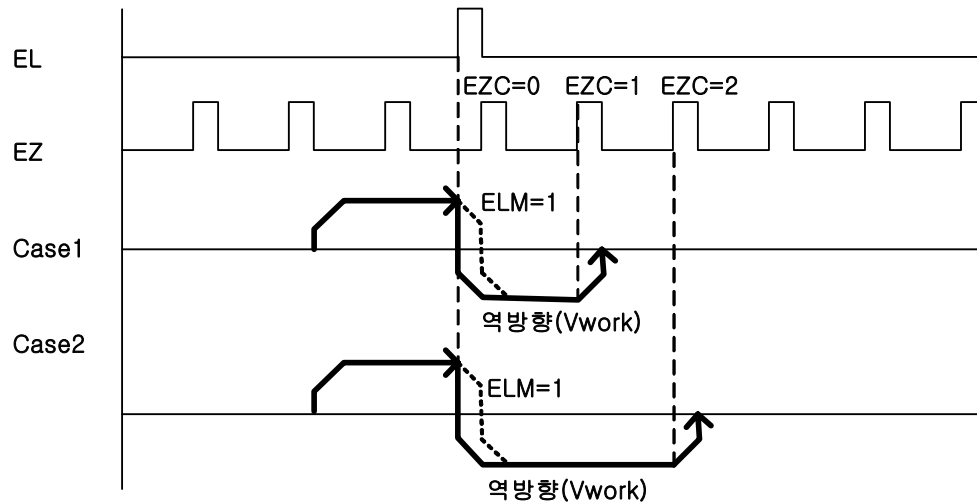
Case2 : EzCount = 2인 경우, COMILX_MC_SetHomeConfig 함수 참조

Vr : Reverse Speed를 의미하며 COMILX_MC_HomeMove 함수 참조

□ **MODE 8 : EL ON → Accelerate to Vwork → EZ → Slow down → Stop**

MODE 8에서는 EL 신호가 ON으로 바뀌는 순간 즉시 정지(또는 ELM=1인 경우에 감속후 정

지)합니다. 그리고 작업속도까지 가속하여 반대 방향으로 회전한 후 EZ 신호에 따라 감속 후 복귀 작업을 종료합니다. 여기서 ELM=1 은 COMILX_MC_SetMioCfgEL()함수에서 nElMode 를 1 로 설정했음을 의미합니다.



Case1 : EzCount = 1 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Case2 : EzCount = 2 인 경우, COMILX_MC_SetHomeConfig 함수 참조

Vwork : 작업속도를 의미하며 COMILX_MC_SetSpeed 함수 참조

□ MODE 9 : EL ON → Accelerate to Vwork → EZ → Slow down → Stop

MODE 9에서는 MODE 0 과 동일한 복귀 작업을 수행한다. 그리고, Feedback Counter 가 0 이 되도록 모션을 다시 취한 후에 복귀 작업을 종료한다.

□ MODE 10 : EL ON → Accelerate to Vwork → EZ → Slow down → Stop

MODE 10에서는 MODE 3 과 동일한 복귀 작업을 수행한다. 그리고, Feedback Counter 가 0 이 되도록 모션을 다시 취한 후에 복귀 작업을 종료한다.

□ MODE 11 : EL ON → Accelerate to Vwork → EZ → Slow down → Stop

MODE 11에서는 MODE 5 과 동일한 복귀 작업을 수행한다. 그리고, Feedback Counter 가 0 이 되도록 모션을 다시 취한 후에 복귀 작업을 종료한다.

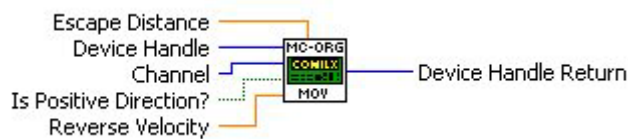
□ MODE 12 : EL ON → Accelerate to Vwork → EZ → Slow down → Stop

MODE 12에서는 MODE 8 과 동일한 복귀 작업을 수행한다. 그리고 난후 Feedback Counter

가 0 이 되도록 모션을 다시 취한 후에 복귀 작업을 종료한다.

COMILX_MC_HomeMoveAuto

■ 아이콘



■ Function 개요

이 함수는 원점 복귀 작업을 시작합니다. 이 함수는 원점 복귀 작업을 시작시킨 후에 바로 리턴(Return)합니다. 따라서 사용자는 COMILX_MC_Done() 함수를 사용하여 복귀 작업이 완료되었는지를 체크하여야 합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Is Positive Direction	BOOL	원점 복귀 모션을 수행할 방향을 지정합니다. 원점복귀 작업을 시작시키면 지정한 방향으로 이동하면서 원점을 검색합니다.
Reverse Velocity	Double	Reverse Speed를 설정합니다. 원점복귀 모드에 따라 Reverse Speed를 필요로 하는 모드가 있습니다. 앞의 원점복귀 모드 설명에서 Reverse Speed는 Vr로 표기되었습니다.
Escape Distance	Double	원점탈출거리를 지정합니다. 원점탈출거리는 원점복귀를 시작하는 시점에 이미 원점센서가 ON되어 있거나, 원점과 -EL 사이에 위치한 경우에 자동으로 원점센서로부터 지정한 거리만큼 탈출한 후 원점복귀를 재개합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고사항

COMILX_MC_HomeMoveAuto() 함수는 기구물이 어느 위치에 있든 자동으로 원점을 찾아가도록 합니다. 기구물의 위치를 크게 3 가지의 경우로 나눌 수 있는데 각각의 경우에 대하여 COMILX_MC_HomeMoveAuto() 함수를 사용한 경우 동작되는 방식은 다음과 같습니다(단, 원점복귀 작업의 방향을 음의 방향(nDirection = -1)로 한경우에 대한 것입니다).

☐ CASE 1. 원점센서를 기준으로 양의 방향 위치에서 원점복귀를 시작한 경우

정상적인 원점복귀 작업을 수행합니다.

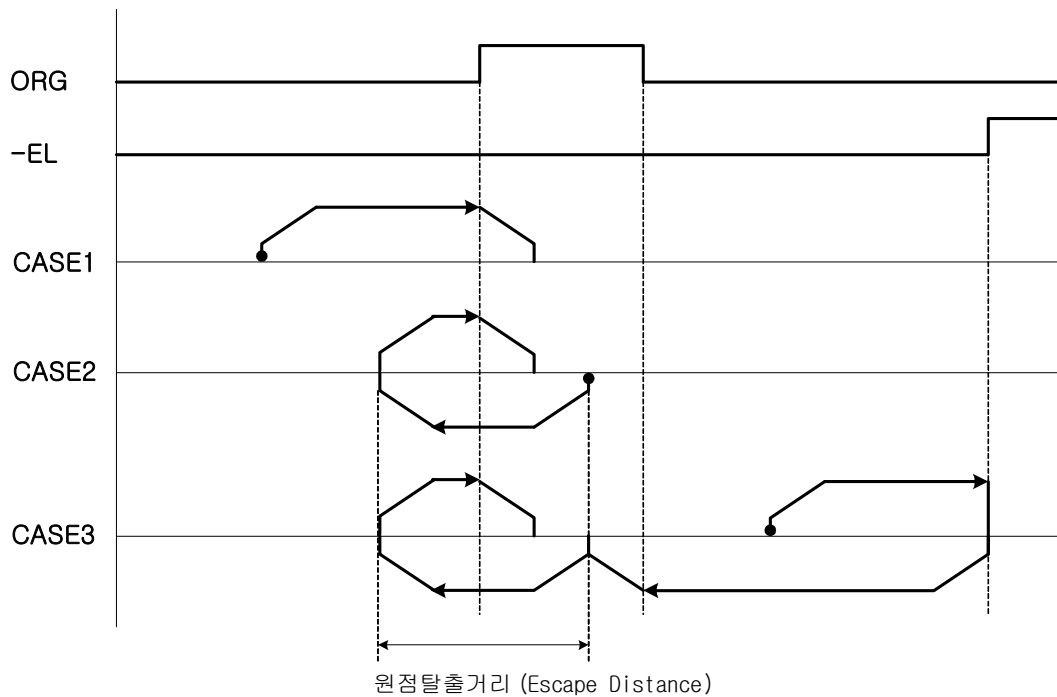
☐ CASE 2. 원점센서가 ON 인 상태에서 원점복귀를 시작한 경우

이러한 경우에는 먼저 원점탈출거리(Escape Distance)만큼 양의 방향으로 이동한 후 다시 정상적인 원점복귀 작업을 수행합니다.

☐ CASE 3. 원점센서를 기준으로 양의 방향 위치에서 원점복귀를 시작한 경우

이러한 경우에는 먼저 음의 방향으로 먼저 이동을 시작합니다. 그리고 -EL(Negative Limit) 센서가 ON 되면 정지 후 양의방향으로 이동합니다. 원점센서가 ON 되면 다시 정지 후 CASE 2 에서와 같이 원점탈출거리(Escape Distance)만큼 양의 방향으로 이동한 후 다시 정상적인 원점복귀 작업을 수행합니다.

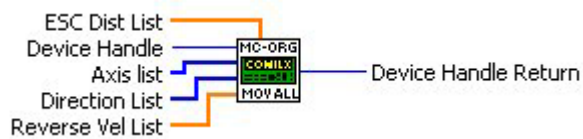
[그림 #]은 원점복귀모드를 0, 속도 패턴을 사다리꼴 방식으로 하고 COMILX_MC_HomeMoveAuto() 함수를 이용하여 원점복귀 작업을 수행할 때 위의 각 경우에 대하여 동작하는 방식을 도식적으로 나타낸 것입니다.



[그림 #] COMILX_MC_HomeMoveAuto 함수를 이용한 원점복귀 작업

COMILX_MC_HomeMoveAutoAll

■ 아이콘



■ Function 개요

이 함수는 여러축에 대하여 COMILX_MC_HomeMoveAuto 작업을 동시에 시작하는 함수입니다. 동작 방식은 COMILX_MC_HomeMoveAuto 함수와 동일합니다. 원점복귀의 자세한 내용은 COMILX_MC_HomeMoveAuto 함수 설명편을 참조하시기 바랍니다.

■ Parameters

Parameters	Type	Description
------------	------	-------------

Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Axis List	I32 Array	채널(축) 번호(0부터 시작합니다).
Direction List	I32 Array	각 축의 원점복귀 진행 방향을 배열로 지정합니다. 진행방향값의 의미는 다음과 같습니다. 0 또는 -1 : (-) 방향 양수 : (+) 방향
Reverse Vel List	Double Array	Reverse Speed 배열.
ESC Dist List	Double Array	원점탈출거리 배열.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

Motion – Single Axis Motion



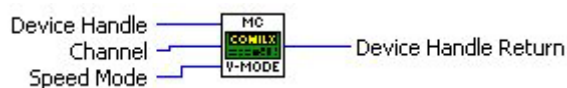
“Single Axis Motion” 함수블록은 Single Axis 모션 제어에 관련된 함수들을 모아놓은 것입니다. Single Axis 모션은 한 축만을 독립적으로 제어하는 작업을 의미합니다. Single Axis 모션은 먼저 속도설정 함수들을 이용하여 속도를 설정하고 이동 함수를 사용하여 이동 작업을 수행합니다. 그리고 필요에 따라 정지 함수를 사용하여 모션을 정지합니다. Single Axis Motion에 관련된 함수들은 다음과 같습니다.

함수명	각 보드 별 지원 여부					
	LX10x	LX20x	LX301	LX401	LX402	LX50x
COMILX_MC_SetSpeedMode						✓
COMILX_MC_SetSpeed						✓
COMILX_MC_SetAccel						✓
COMILX_MC_StartVMove						✓
COMILX_MC_StartMove						✓
COMILX_MC_Move						✓
COMILX_MC_StartMoveTo						✓
COMILX_MC_MoveTo						✓
COMILX_MC_Stop						✓
COMILX_MC_Done						✓

“Single Axis Motion” 블록에 포함된 함수 리스트

COMILX_MC_SetSpeedMode

■ 아이콘



■ Function 개요

Motion의 속도 모드를 설정합니다. 단, 이 함수는 Motion에 바로 영향을 주는 것이 아니고 Move, MoveTo 등의 이송 함수가 수행될 때 설정된 내용이 적용됩니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Speed Mode	I32	속도모드를 지정합니다. 0 : Constant speed mode 1 : Trapezoidal speed mode 2 : S-curve speed mode

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

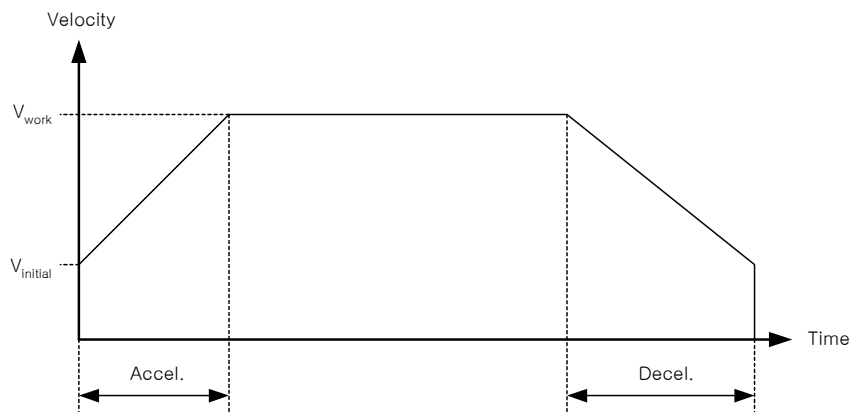
참고사항

□ Constant speed mode

Constant speed mode에서는 Motion을 수행할 때 가속/감속을 적용하지 않고 일정속도로 Motion을 수행합니다. 여기서 적용되는 일정 속도는 COMILX_MC_SetSpeed 함수에서 주어지는 fEndSpeed에서 주어진 값이 적용됩니다.

□ Trapezoidal speed mode

Trapezoidal speed mode에서는 Motion을 수행하는데 있어서 속도의 패턴을 [그림 5-4]와 같이 Linear acceleration → Working speed(constant) → Linear deceleration의 형태로 운용하는 모드입니다.



[그림 5-4] Trapezoidal speed pattern

여기서 Acceleration time 은

$$T_{acc} = (V_{work} - V_{initial})/a$$

Where,

T_{acc} : Acceleration time

$V_{initial}$: Initial speed

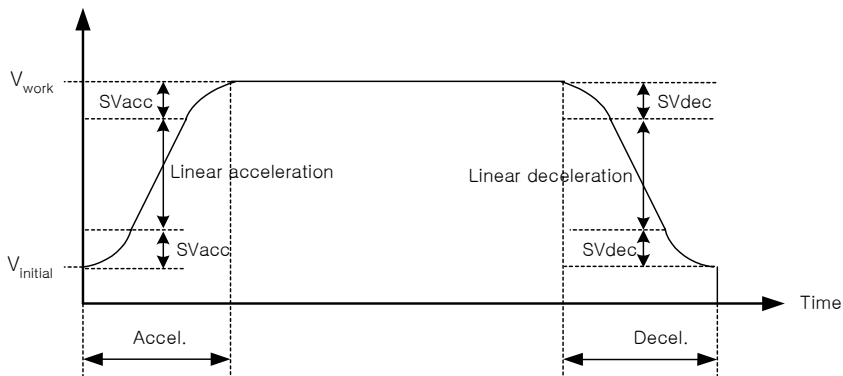
V_{work} : Working speed

a : Acceleration setting value

과 같으며 Deceleration time 또한 위와 같은 계산식이 적용됩니다.

□ S-curve speed mode

S-curve speed mode에서는 Motion을 수행할 때 S자형 형태로 가속과 감속을 수행합니다. S-curve speed mode에서 가(감)속 구간은 [그림 5-5]과 같이 S-curve section과 Linear acceleration section으로 구성됩니다.



[그림 5-5] S-curve speed pattern

※ **S-curve section** : S-curve 형식의 가/감속이 이루어지는 구간. 이 구간은 COMILX_MC_SetSCurve 함수의 fSVacc와 fSVdec 파라미터에 의해 설정됩니다. fSVacc 값이 0 이거나 속도 범위(Working speed - Initial speed)의 50%로 설정되면 가속구간은 Linear acceleration section이 없이 모두 S-curve section으로 구성됩니다. 마찬가지로, fSVdec 값이 0 이거나 속도 범위(Working speed - Initial speed)의 50%로 설정되면 감속구간은 Linear deceleration section이 없이 모두 S-curve section으로 구성됩니다.

※ S-curve speed mode에서 COMILX_MC_SetAccel 함수를 통하여 설정한 가(감)속 값은 S-curve section을 포함한 전체 가(감)속 시간을 결정하는 파라미터로 사용되며 실제 가(감)속도 또는 Jerk는 자동으로 계산됩니다. 전체 가속 시간 Tacc는

$$T_{acc} = (V_{work} - V_{initial})/a$$

여기서,

T_{acc} : Acceleration time

$V_{initial}$: Initial speed

V_{work} : Working speed

a : Acceleration setting value

과 같으며 Deceleration time 또한 위와 같은 계산식이 적용됩니다.

COMILX_MC_SetSpeed

■ 아이콘



■ Function 개요

Motion 의 속도를 설정합니다. 단, 이 함수는 Motion 에 바로 영향을 주는 것이 아니고 Move, MoveTo 등의 이송 함수가 수행될 때 설정된 내용이 적용됩니다.

■ Parameters

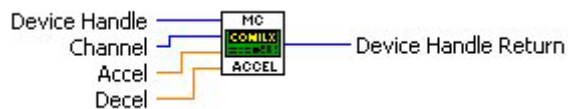
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Initial Speed	Double	초기속도를 설정합니다. 단, Constant 속도 모드에서는 이 값이 무시됩니다.
Work Speed	Double	작업속도를 설정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_SetAccel

■ 아이콘



■ Function 개요

Motion 의 가/감속도를 설정합니다. 단, 이 함수는 Motion 에 바로 영향을 주는 것이 아니고 Move, MoveTo 등의 이송 함수가 수행될 때 설정된 내용이 적용됩니다.

■ Parameters

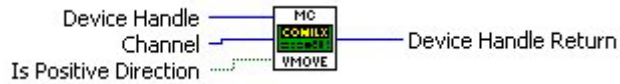
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Accel	Double	가속도를 설정합니다.
Decel	Double	감속도를 설정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_StartVMove

■ 아이콘



■ Function 개요

작업속도까지 가속한 후에 작업속도를 유지하며 정지함수가 호출될 때까지 지정한 방향으로의 모션을 계속 수행합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Is Positive Direction	BOOL	True => (+) 방향으로 이동합니다. False => (-) 방향으로 이동합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고사항

□ Velocity Move 를 정지할 때 COMILX_MC_Stop 또는 COMILX_MC_EngStop 을 사용합니다.

COMILX_MC_StartMove

■ 아이콘



■ Function 개요

하나의 축에 대하여 현재의 위치에서 지정한 거리만큼 이동을 수행합니다. 이 함수는 모션(Motion)을 시작 시킨 후에 바로 Return 합니다. 모션이 완료됨을 체크하려면 COMILX_MC_Done 함수를 사용합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Distance	Double	이동할 거리를 지정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고사항

□ COMILX_MC_Move 함수가 모션이 완료될 때까지 Return 되지 않는데 반하여, 이 함수는 지정한 모션을 시작시킨 후에 바로 Return 하게 됩니다.

□ COMILX_MC_StartMoveTo 함수가 절대좌표로의 이동을 수행하는데 반하여, 이 함수는 현재 위치에서 상대적인 거리를 파라미터로하여 이동을 수행합니다.

COMILX_MC_Move

■ 아이콘



■ Function 개요

하나의 축에 대하여 현재의 위치에서 지정한 거리만큼 이동을 수행합니다. 이 함수는 지정한 위치로의 이동이 완료되기 전까지 Return 되지 않습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Distance	Double	이동할 거리를 지정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고사항

□ COMILX_MC_StartMove 함수가 모션이 완료되는 것을 기다리지 않고 바로 Return 하는데 반하여, 이 함수는 지정한 상대좌표로의 이동이 완료되기 전까지 Return 되지 않고 루프를 돌게 됩니다. 루프를 도는 동안 윈도우 이벤트나 메시지가 처리될 수 있도록 하려면 이 함수를 수행하기 이전에 COMILX_MC_SetBlockingMode 함수를 사용하여 Blocking 이 일어나지 않도록 설정하여야 합니다.

□ COMILX_MC_MoveTo 함수가 절대좌표로의 이동을 수행하는데 반하여, 이 함수는 현재 위치에서 상대적인 거리를 파라미터로하여 이동을 수행합니다.

COMILX_MC_StartMoveTo

■ 아이콘



■ Function 개요

하나의 축에 대하여 지정한 절대좌표로의 이동을 수행합니다. 이 함수는 모션(Motion)을 시작 시킨 후에 바로 Return 합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Target Position	Double	이동할 목표좌표를 절대좌표값으로 지정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

■ 참고사항

□ COMILX_MC_MoveTo 함수가 모션이 완료될 때까지 Return 되지 않는데 반하여, 이 함수는 지정한 모션을 시작시킨 후에 바로 Return 하게 됩니다.

□ COMILX_MC_StartMove 함수가 현재위치에 대한 상대좌표로의 이동을 수행하는데 반하여, 이 함수는 절대좌표로의 이동을 수행합니다.

COMILX_MC_MoveTo

■ 아이콘



■ Function 개요

하나의 축에 대하여 지정한 절대좌표로의 이동을 수행합니다. 이 함수는 모션 (Motion)을 완료되기 전까지 Return 되지 않습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).
Target Position	Double	이동할 목표좌표를 절대좌표값으로 지정합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_Stop

■ 아이콘



■ Function 개요

지정한 축에 대한 모션을 감속 후 정지합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_EmgStop

■ 아이콘



■ Function 개요

지정한 축에 대한 모션을 감속없이 즉시 정지합니다.

■ Parameters

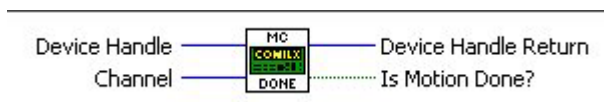
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_Done

■ 아이콘



■ Function 개요

하나의 축에 대하여 모션이 완료됐는지를 체크합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	채널(축) 번호(0부터 시작합니다).

■ Return 값

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device

Return		Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Motion Done	BOOL	지정한 축의 모션이 완료됐는지를 알려줍니다. False => 모션이 완료되지 않았음 True => 모션이 완료됨

Motion – Multiple Axes Motion



“Multiple Axes Motion” 블록은 다축 동시 제어에 관련된 함수들을 모아놓은 것입니다. 다축 동시 제어는 여러 개의 축을 완전한 동기를 맞추어 동시에 제어하는 기능을 말합니다. 만일 속도 패턴을 동일하게 설정하였다면 여러 개의 제어 대상 축이 시작 및 종료 시점은 물론이고 가속/감속 구간까지 완전히 동기를 맞추어 제어될 수 있습니다.

다축 동시제어 기능은 Velocity Move 와 In-position Move 모두에 적용 가능합니다.

COMILX_MC_StartVMoveAll

■ 아이콘



■ Function 개요

여러 개의 축에 대하여 Velocity Move 작업을 동시에 시작합니다. Velocity Move 는 작업속도까지 가속한 후에 작업속도를 유지하며 정지함수가 호출될 때까지 지정한 방향으로의 모션을 계속 수행합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다. 따라서 이 함수는 여러축이 동기를 맞추어 작업을 시작해야하는 경우에 유용하게 사용될 수 있습니다. 속도 패턴은 각 축에 대하여 COMILX_MC_SetSpeedMode, COMILX_MC_SetSpeed, COMILX_MC_SetAccel 함수등에 의해 설정된 대로 이루어집니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Axis List	I32 Array	동시에 작업을 수행할 대상 축의 배열

Dir List	I32 Array	방향을 지시하는 값의 배열 주소값. 모션의 방향을 지시하는 값은 다음과 같습니다. 0 또는 음수: (-) 방향 양수 : (+)방향
----------	--------------	--

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_StartMoveAll

아이콘



Function 개요

여러 개의 축에 대하여 현재의 위치에서 지정한 거리만큼 이동을 동시에 시작합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다. 따라서 이 함수는 여러축이 동기를 맞추어 작업을 시작해야하는 경우에 유용하게 사용될 수 있습니다.

이 함수는 모션(Motion)을 시작 시킨 후에 바로 Return 합니다. 속도 패턴은 각 축에 대하여 COMILX_MC_SetSpeedMode, COMILX_MC_SetSpeed, COMILX_MC_SetAccel, COMILX_MC_SetScurve 함수등에 의해 설정된 대로 이루어집니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

Axis List	I32 Array	동시에 작업을 수행할 대상 축의 배열
Distance List	Double Array	이동할 거리값의 배열. 이동 거리값은 현재의 위치에 대한 상대 좌표이며 거리입니다.

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_MoveAll

아이콘



Function 개요

여러 개의 축에 대하여 현재의 위치에서 지정한 거리만큼 이동을 동시에 시작합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다. 따라서 이 함수는 여러축이 동기를 맞추어 작업을 시작해야하는 경우에 유용하게 사용될 수 있습니다.

이 함수는 모션(Motion)이 완료될 때까지 Return 되지 않습니다. 속도 패턴은 각 축에 대하여 COMILX_MC_SetSpeedMode, COMILX_MC_SetSpeed, COMILX_MC_SetAccel, COMILX_MC_SetScurve 함수등에 의해 설정된 대로 이루어집니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

Axis List	I32 Array	동시에 작업을 수행할 대상 축의 배열
Distance List	Double Array	이동할 거리값의 배열. 이동 거리값은 현재의 위치에 대한 상대 좌표이며 거리입니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_StartMoveToAll

■ 아이콘



■ Function 개요

여러 개의 축에 대하여 지정한 절대좌표로의 이동을 동시에 시작합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다. 따라서 이 함수는 여러축이 동기를 맞추어 작업을 시작해야하는 경우에 유용하게 사용될 수 있습니다.

이 함수는 모션(Motion)을 시작 시킨 후에 바로 Return 합니다. 속도 패턴은 각 축에 대하여 COMILX_MC_SetSpeedMode, COMILX_MC_SetSpeed, COMILX_MC_SetAccel 함수등에 의해 설정된 대로 이루어집니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

Axis List	I32 Array	동시에 작업을 수행할 대상 축의 배열
Position List	Double Array	이동할 목표위치의 절대좌표값을 담은 배열.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_MoveToAll

■ 아이콘



■ Function 개요

여러 개의 축에 대하여 지정한 절대좌표로의 이동을 동시에 시작합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다. 따라서 이 함수는 여러축이 동기를 맞추어 작업을 시작해야하는 경우에 유용하게 사용될 수 있습니다.

이 함수는 모션(Motion)이 완료될 때까지 Return 되지 않습니다. 속도 패턴은 각 축에 대하여 COMILX_MC_SetSpeedMode, COMILX_MC_SetSpeed, COMILX_MC_SetAccel 함수등에 의해 설정된 대로 이루어집니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

Axis List	I32 Array	동시에 작업을 수행할 대상 축의 배열
Position List	Double Array	이동할 목표위치의 절대좌표값을 담은 배열.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_StopAll

■ 아이콘



■ Function 개요

여러 개의 축에 대한 모션을 동시에 감속 후 정지합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Axis List	I32 Array	동시에 정지할 대상 축의 배열

■ Return 값

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device

Return		Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
--------	--	---

COMILX_MC_EmgStopAll

■ 아이콘



■ Function 개요

여러 개의 축에 대한 모션을 동시에 감속없이 즉시 정지합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Axis List	132 Array	동시에 정지할 대상 축의 배열

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_AllDone

■ 아이콘



■ Function 개요

여러 개의 축에 대하여 지정한 모든 축의 모션이 완료됐는지를 체크합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Axis List	I32 Array	동시에 정지할 대상 축의 배열

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
Is Motion Done	BOOL	지정한 모든 축의 모션이 완료되었는지를 알려줍니다. False => 일부 또는 전체 축의 모션이 완료되지 않음 True => 모든 축의 모션이 완료됨

Motion – Interpolation

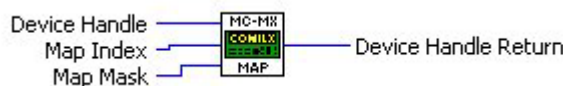


“Coordinated Motion” 함수블록은 Interpolation 작업에 관련된 함수들을 모아놓은 것입니다. Coordinated Motion 이란 두 축 이상의 축이 연동되어 직선 보간 (Linear Interpolation), 원호 보간(Circular Interpolation) 등의 모션을 수행하는 것을 의미합니다.

다축 동시 제어 기능도 여러 개의 축을 동시에 제어하지만, 이것은 각 축이 서로 연동되어서 모션을 수행하는 것이 아니고 각 축이 개별적으로 모션을 수행하되 동시에 시작하는 것임에 반하여, Coordinated Motion 은 여러 개의 축이 서로 연동되어서 보간 이동을 수행한다는 것이 다축 동시 제어와 차이가 있습니다.

COMILX_MC_MapAxes

■ 아이콘



■ Function 개요

Coordinated Motion 을 수행할 축들을 그룹화합니다. Coordinated Motion 에 관련된 거의 모든 함수는 ‘Map Index’를 통하여 제어대상 축을 지정받게 됩니다. 따라서 기타 Coordinated Motion 과 관련된 함수들을 수행하기전에 먼저 이 함수를 이용하여 축맵을 구성해야합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Map Mask	I32	그룹에 포함할 축들을 지정할 마스크 값. 이 값의 BIT0~BIT3을 이용하여 그룹에 포함할 축들을 지정합니

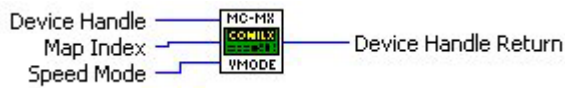
		다. 각 비트의 값이 0이면 해당 축(비트의 순서와 일치하는 축)은 배제되는 것이며 1이면 해당 축이 포함되는 것입니다.
--	--	---

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_SetSpeedModeMx

■ 아이콘



■ Function 개요

Coordinated Motion 의 속도 모드를 설정합니다. 단, 이 함수는 Motion 에 바로 영향을 주는 것이 아니고 Line, Arc 등의 Coordinated Motion 이송 함수가 수행될 때 설정된 내용이 적용됩니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Speed Mode	I32	속도 모드를 지정합니다. 속도 모드는 아래의 표와 같이 3가지로 설정할 수 있습니다. Coordinated Motion 에서는 S-curve 속도 모드로 설정하면 가/감속 구간에서 Linear Section이 없는 완전한 S-curve 가/감속 모드로 구성됩니다. 각각의 속도 모드에 대한 자세한 사

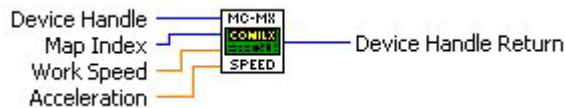
		<p>함은 COMILX_MC_SetSpeedMode 함수를 참조하십시오.</p> <p>0 => Constant speed mode</p> <p>1 => Trapezoidal speed mode</p> <p>2 => S-curve speed mode</p>
--	--	---

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_SetSpeedMx

아이콘



Function 개요

Coordinated Motion 의 속도 및 가속도를 설정합니다. 단, 이 함수는 Motion 에 바로 영향을 주는 것이 아니고 Line, Arc 등의 Coordinated Motion 이송 함수가 수행될 때 설정된 내용이 적용됩니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Work Speed	Double	작업속도를 벡터 속도값으로 지정합니다.
Acceleration	Double	가속도와 감속도를 지정합니다. Coordinated Motion에 서는 가속도와 감속도가 같은값으로 설정됩니다.

Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_StartLine

아이콘



Function 개요

이 함수는 현재 위치로부터의 상대 좌표로의 직선 보간 이동을 수행합니다. 이 함수는 모션을 시작시킨후에 바로 Return 합니다. 해당 작업이 완료되었는지는 사용자가 COMILX_MC_MxdDone 함수를 통하여 직접 체크하여야 합니다.

Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Distance List	Double Array	현재 위치로부터의 상대적인 이동 좌표값(각 축의 이동 거리값)의 배열. 이 배열의 크기는 COMILX_MC_MapAxes 함수를 통하여 맵핑된 축의 수와 일치하여야 합니다.

Return 값

Parameters	Type	Description
------------	------	-------------

Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.
-------------------------	-----	---

COMILX_MC_Line

■ 아이콘



■ Function 개요

이 함수는 현재 위치로부터의 상대 좌표로의 직선 보간 이동을 수행합니다. 이 함수는 해당 작업이 완료될 때까지 Return 되지 않습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Distance List	Double Array	현재 위치로부터의 상대적인 이동 좌표값(각 축의 이동 거리값)의 배열. 이 배열의 크기는 COMILX_MC_MapAxes 함수를 통하여 맵핑된 축의 수와 일치하여야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_StartLineTo

■ 아이콘



■ Function 개요

이 함수는 지정한 절대좌표로의 직선 보간 이동을 수행합니다. 이 함수는 모션을 시작시킨후에 바로 Return 합니다. 해당 작업이 완료되었는지는 사용자가 COMILX_MC_MxdDone 함수를 통하여 직접 체크하여야 합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Position List	Double Array	이동할 목표 절대좌표값(각 축의 절대좌표값)의 배열. 이 배열의 크기는 COMILX_MC_MapAxes 함수를 통하여 맵핑된 축의 수와 일치하여야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.

COMILX_MC_LineTo

■ 아이콘



■ Function 개요

이 함수는 지정한 절대좌표로의 직선 보간 이동을 수행합니다. 이 함수는 해당 작업이 완료되기전까지 Return 되지 않습니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMILX_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Map Index	I32	축 그룹 인덱스.
Position List	Double Array	이동할 목표 절대좌표값(각 축의 절대좌표값)의 배열. 이 배열의 크기는 COMILX_MC_MapAxes 함수를 통하여 맵핑된 축의 수와 일치하여야 합니다.

■ Return 값

Parameters	Type	Description
Device Handle Return	U32	디바이스 핸들 값입니다. 이 값은 입력 받은 Device Handle 값을 그대로 반환하는 것으로써 다음에 사용할 함수의 입력 parameter로 사용함으로써 함수가 수행되어야 할 순서를 명확히 해줄 수 있습니다.