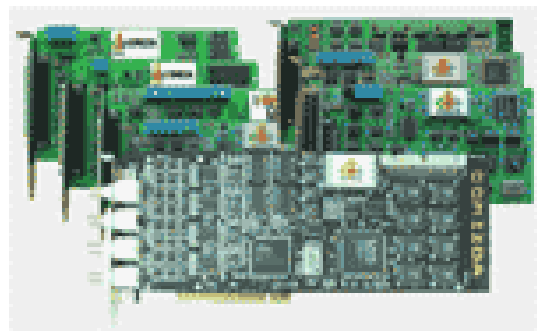
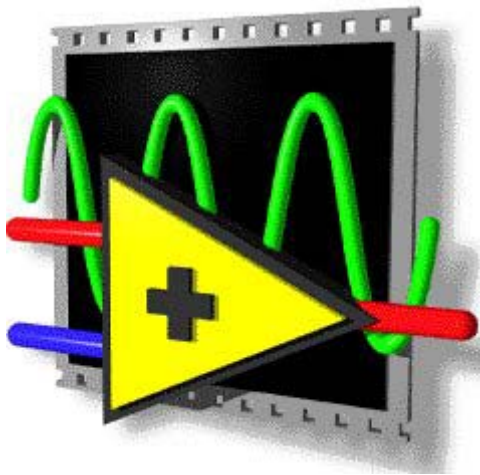


# COMIDAS (CP/SD)

## 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](mailto:Hardware@comizoa.co.kr)

Software Support : [Software@comizoa.co.kr](mailto:Software@comizoa.co.kr)



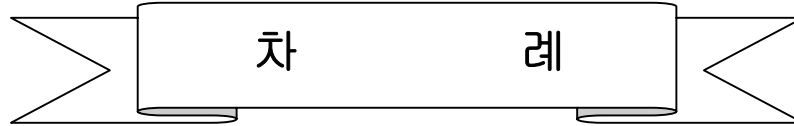
㈜커미조아

[www.comizoa.co.kr](http://www.comizoa.co.kr)

[www.comizoa.com](http://www.comizoa.com)

Tel) 042 - 861 - 3301~3

Fax) 042 - 861 - 3304



<b>PART I INTRODUCTION.....</b>	<b>7</b>
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
<b>PART II FUNCTIONS .....</b>	<b>20</b>
LOAD/UNLOAD .....	21
<i>COMI_LoadDevice</i> .....	21
<i>COMI_UnloadDevice</i> .....	22
AI GENERAL .....	22
<i>COMI_AD_SetRange</i> .....	23
<i>COMI_AD_GetDigit</i> .....	23
<i>COMI_AD_GetVolt</i> .....	24
AI-SCAN .....	25
<i>COMI_US_Start</i> .....	25
<i>COMI_US_Stop</i> .....	27
<i>COMI_US_CurCount</i> .....	27
<i>COMI_US_RetrOne</i> .....	28
<i>COMI_US_RetrChannel</i> .....	28
ANALOG OUT .....	30
<i>COMI_DA_Out</i> .....	30
<i>COMI_WFM_Start</i> .....	31
<i>COMI_WFM_Stop</i> .....	32
<i>COMI_WFM_RateChange</i> .....	32
<i>COMI_WFM_GetCurPos</i> .....	33
<i>COMI_WFM_GetCurLoops</i> .....	33

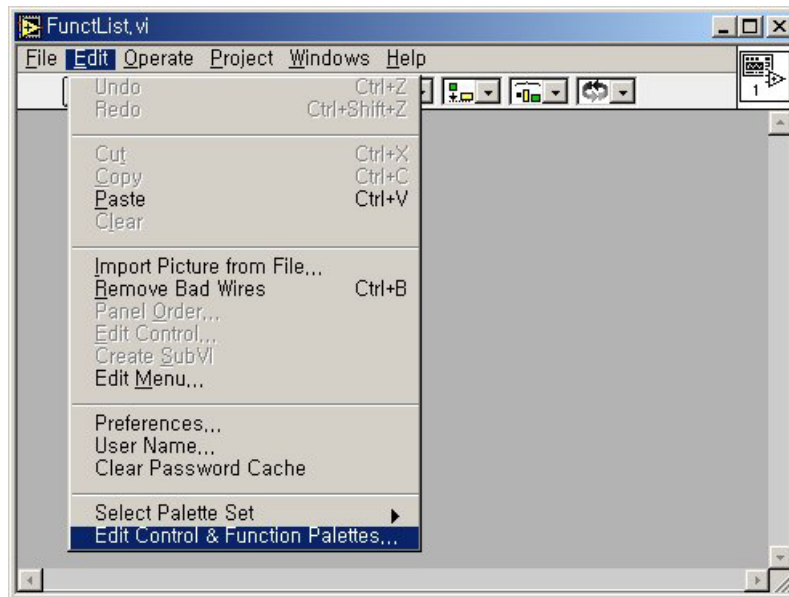
DIGITAL IN/OUT .....	35
<i>COMI_DIO_SetUsage</i> .....	35
<i>COMI_DI_GetOne</i> .....	36
<i>COMI_DI_GetAll</i> .....	36
<i>COMI_DI_GetAllEx</i> .....	37
<i>COMI_DO_PutOne</i> .....	38
<i>COMI_DO_PutAll</i> .....	38
<i>COMI_DO_PutAllEx</i> .....	39
8254 COUNTER .....	40
<i>COMI_SetCounter</i> .....	40
<i>COMI_LoadCount</i> .....	41
<i>COMI_ReadCount</i> .....	42
ENCODER COUNTER .....	43
<i>COMI_ENC_Config</i> .....	44
<i>COMI_ENC_Load</i> .....	45
<i>COMI_ENC_LoadZ</i> .....	45
<i>COMI_ENC_Read</i> .....	46
<i>COMI_ENC_ReadZ</i> .....	47
<i>COMI_ENC_Reset</i> .....	47
<i>COMI_ENC_ResetZ</i> .....	48
PULSE GENERATOR .....	49
<i>COMI_PG_Start</i> .....	49
<i>COMI_PG_ChangeFreq</i> .....	51
<i>COMI_PG_Stop</i> .....	51
<i>COMI_PG_IsActive</i> .....	52



## **PART | Introduction**

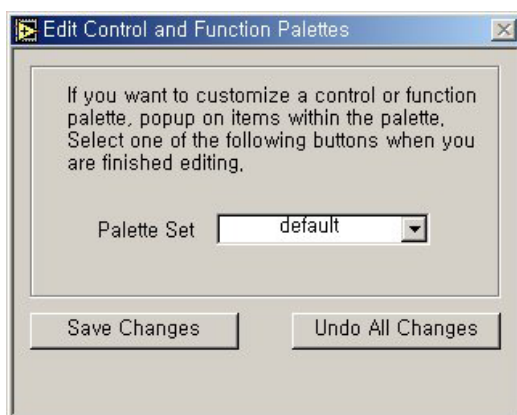
# Labview V5.5에서 설치 및 사용

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.

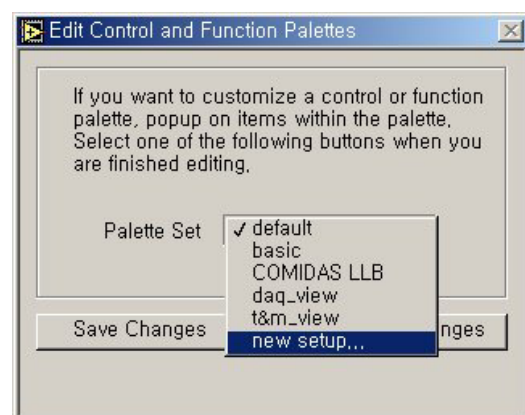


[그림 1-1] Edit Control & Function Palettes 메뉴 선택

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



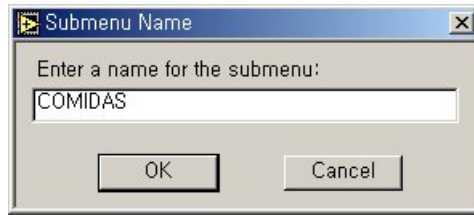
[그림 1-2] Palette 선택 대화상자 1



[그림 1-3] Palette 선택 대화상자 2

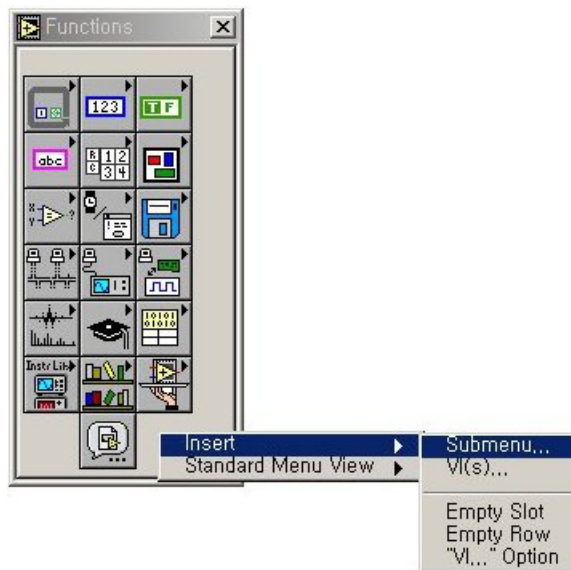
[그림 1-4]와 같은 화면이 나타나면 사용자가 사용할 Palette 의 이름을 지정합니다. 이 이름은 사용자 마음대로 지정할 수 있습니다.





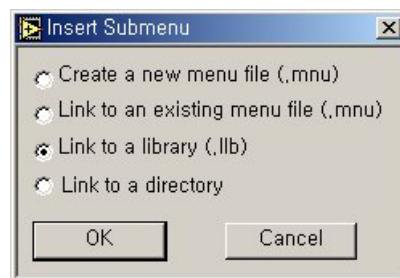
[그림 1-4] 새 Palette 이름 지정 대화상자

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



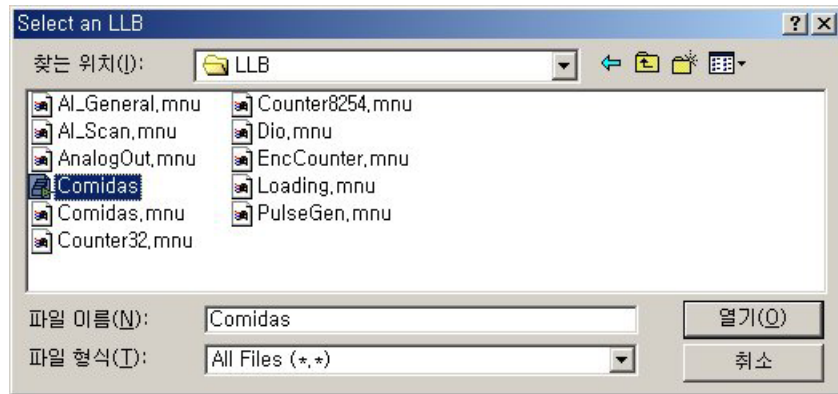
[그림 1-5] "Insert => Submenu..." 메뉴 선택

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



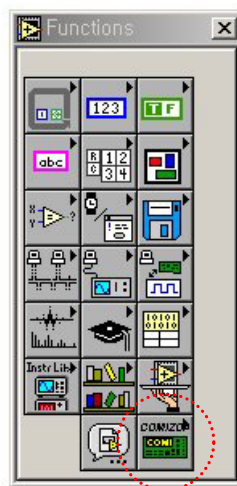
[그림 1-6] "Insert Submenu" 대화상자

[그림 1-7]과 같이 파일을 선택하는 화면이 나타나면 C:\WProgram Files\WCOMIDAS-CPSD\Window\Labview\WLib 폴더에서 Comidas.llb 파일을 선택한 후 열기 버튼을 클릭합니다.



[그림 1-7] “Select an LLB” 대화상자

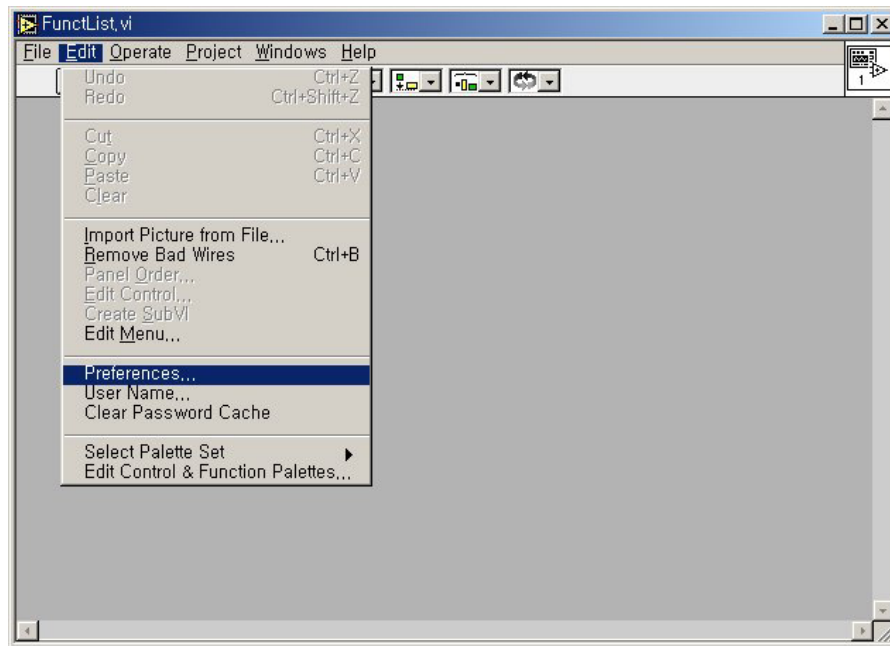
위의 과정을 성공적으로 마쳤으면 [그림 1-8]과 같이 Functions Palette 에 COMIDAS Functions 블록이 나타납니다. 이 후에 사용자는 Functions Palette 에서 필요한 COMIDAS Function 을 선택하여 사용할 수 있습니다.



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

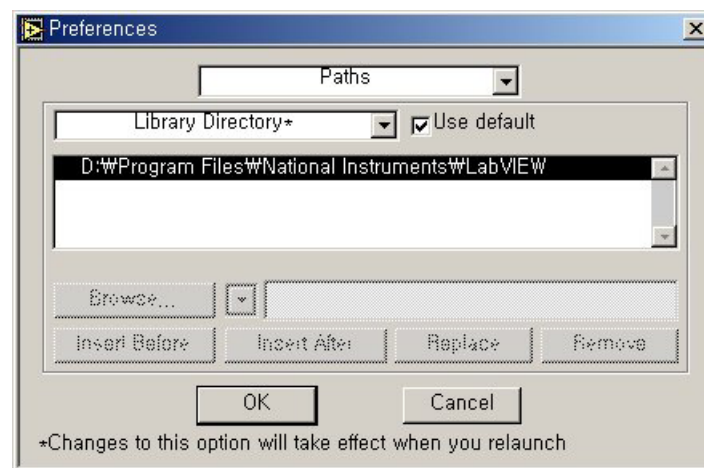
## VI Search Path 등록

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.

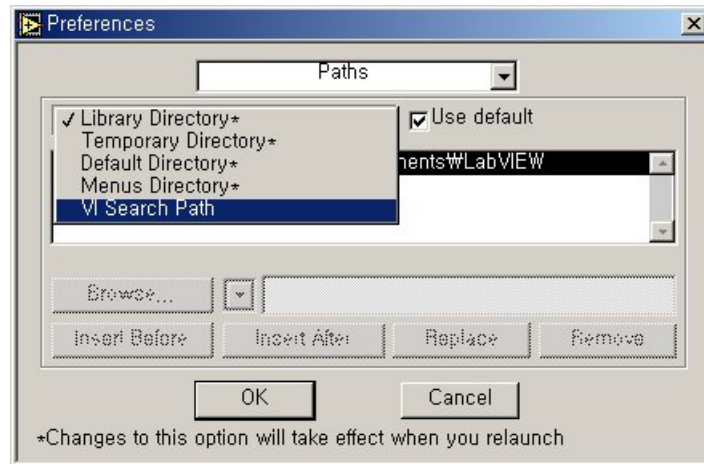


[그림 1-9] Preferences 메뉴 선택

[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.

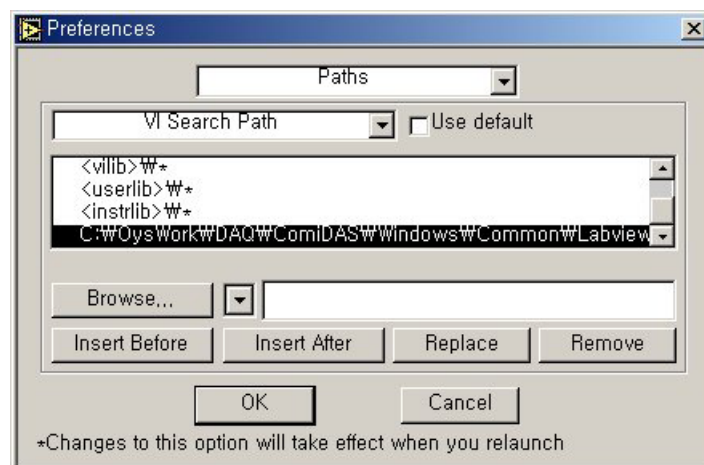


[그림 1-10] Preference 대화 상자 1



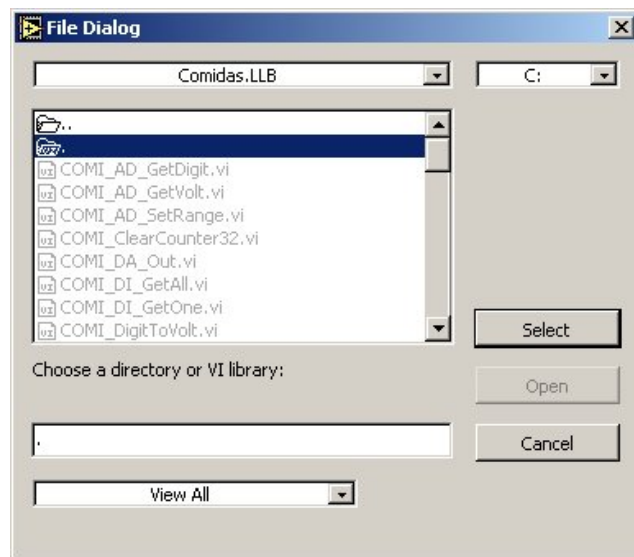
[그림 1-11] Preference 대화 상자 2

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



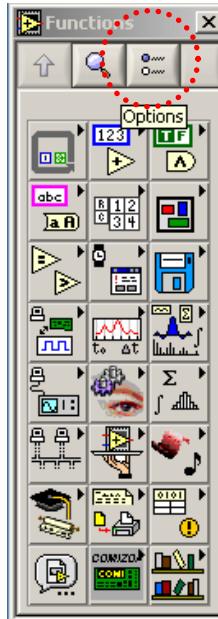
[그림 1-12] Preference 대화 상자 3

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.



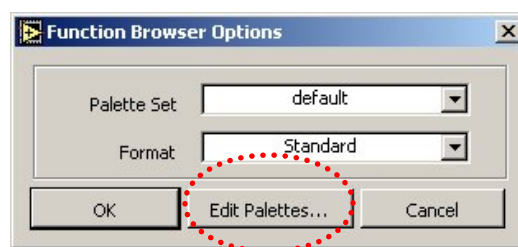
## Labview V6.01에서 설치 및 사용

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



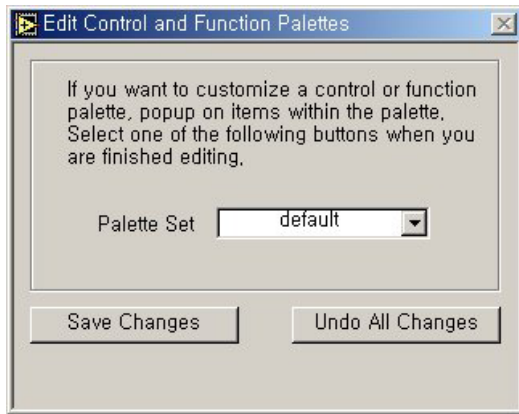
[그림 1-13] Function Palette 에서 Option 버튼 클릭

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

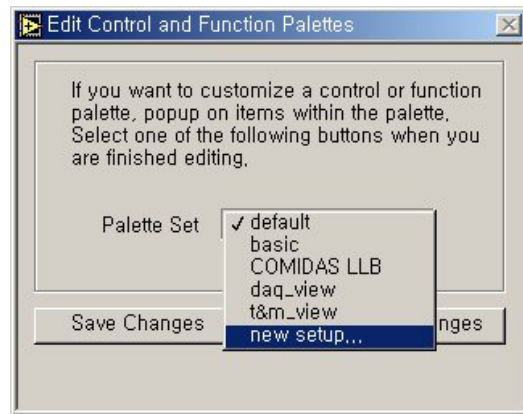


[그림 1-14] Function Browser Options 대화상자

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

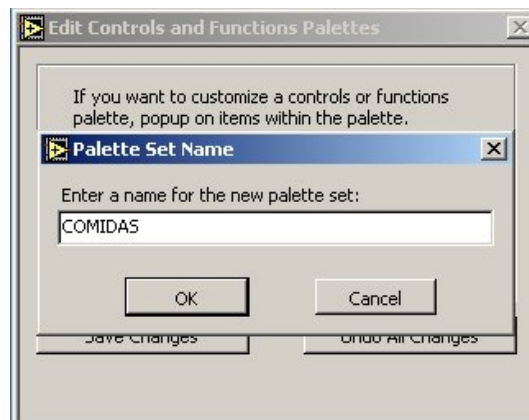


[그림 1-15] Palette 선택 대화상자 1



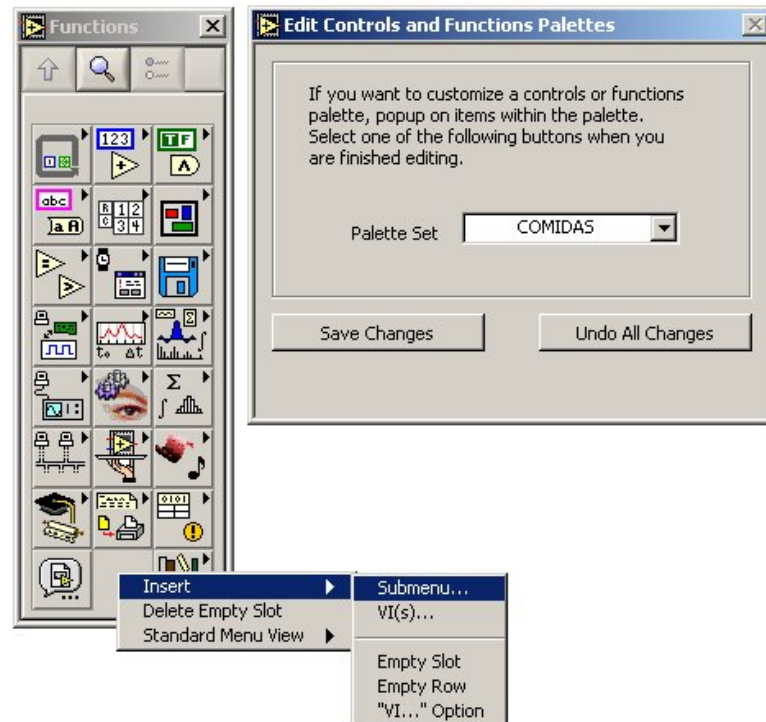
[그림 1-16] Palette 선택 대화상자 2

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



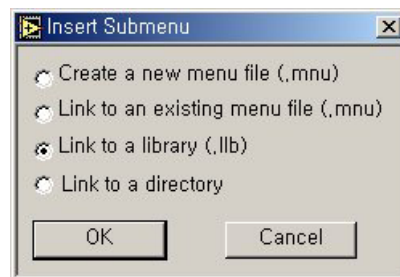
[그림 1-17] Palette Set Name 설정 대화상자

새 Palette 의 이름을 지정하였으면 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

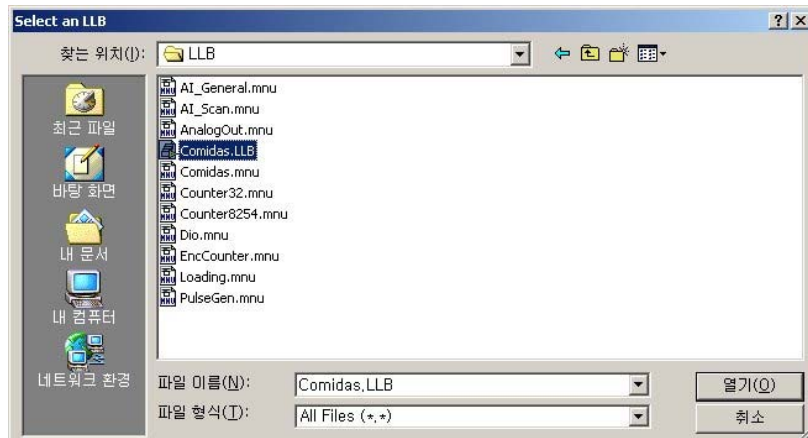
[그림 1-19]과 같은 화면이 나타나면 “Link to a library”를 선택한 후 OK 버튼을 클릭합니다.



[그림 1-19] “Insert Submenu” 대화상자

[그림 1-20]과 같이 파일을 선택하는 화면이 나타나면 C:\Program Files\WCOMIDAS-CPSD\Window\Labview\WLib 폴더에서 Comidas.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.

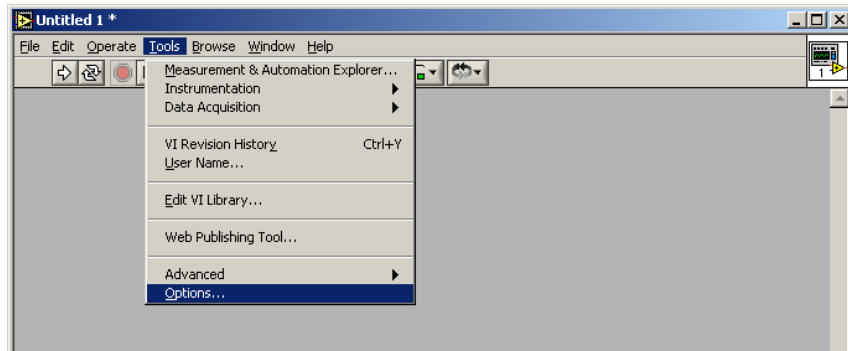


[그림 1-21] Functions Palette 에 COMIDAS Functions 블록이 등록된 화면

## VI Search Path 등록

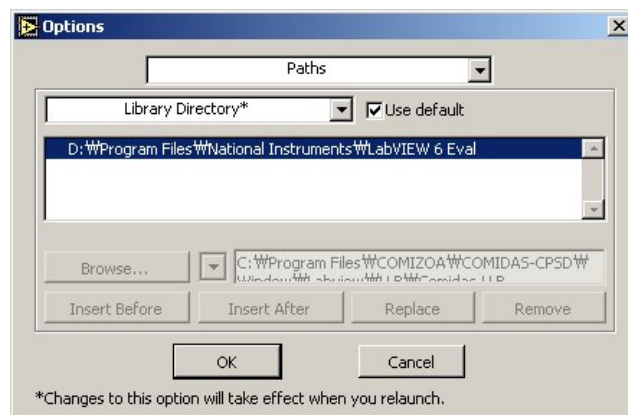
COMIDAS Function Library 를 VI Search Path 에 등록하는 것이 좋습니다. 그렇지 않은 경우 Function VI 를 이용하여 프로그램을 작성 후 다시 Open 할 때 사용한 Function VI 들의 위치를 재차 확인하는 경우가 발생할 수 있습니다. VI Search Path

에 등록하려면 [그림 1-22]와 같이 “Tools=>Options...” 메뉴를 선택합니다.

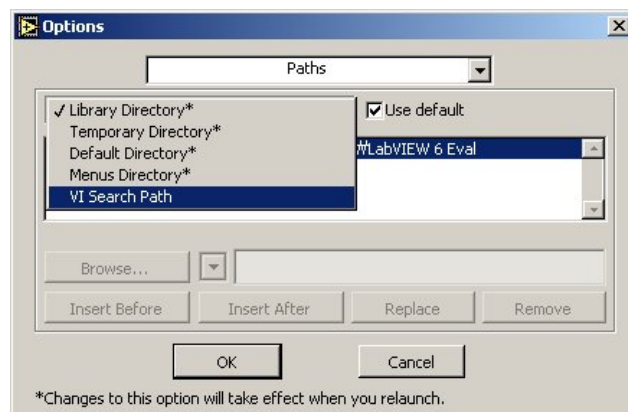


[그림 1-22] Tools=>Options 메뉴 선택

[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.



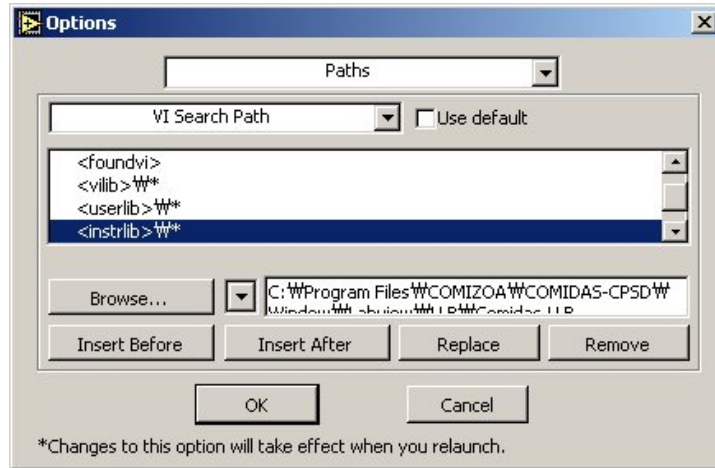
[그림 1-23] Options 대화상자



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

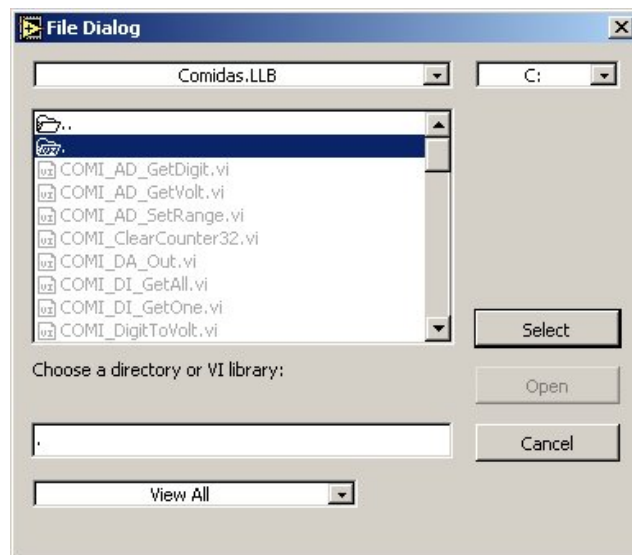
그러면 [그림 1-25]와 같이 하단에 있는 버튼들의 Dimmed 가 풀리게 됩니다.

“Browse”버튼을 클릭하여 C:\WComidas\Window\Labview\WLib\WComidas.Llb 를 선택합니다.  
이 때 C:\WComidas 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 1-12]의 화면으로 복귀하는데 여기서 “Insert Before” 또는 “Insert After”버튼을 클릭한 후 OK 버튼을 클릭하면 VI Search Path 등록이 완료됩니다.





## **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 적용되어야 할 함수들입니다. COMIDAS 라이브러리의 거의 모든 함수들은 디바이스 핸들을 파라미터로 입력하여야 합니다. 디바이스 핸들을 얻기 위해서는 디바이스를 로드해야 합니다. 또한 프로그램 종료 시에는 디바이스를 언로드하여야 합니다.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_LoadDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMI_UnloadDevice	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

디바이스 시작/종료 함수 리스트 및 각 보드별 지원 여부

## COMI\_LoadDevice

### ■ 아이콘



### ■ Function 개요

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

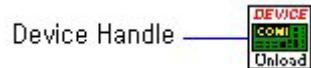
### ■ Parameters

Parameters	Type	Description
Device ID	U16	<p>각 디바이스의 고유한 아이디 값입니다. 이 값은 4 자리의 16 진수 값으로 구성됩니다. CP 시리즈는 C 로 시작하고, SD 시리즈는 B 로 시작하며, 나머지 세 자리는 각 디바이스의 모델 번호와 같은 값을 적어주시면 됩니다.</p> <p>Ex) COMI-CP101 =&gt; <b>C101</b> (Hex)  COMI-SD101 =&gt; <b>B101</b> (Hex)</p>

Instance	U32	동일 deviceID를 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-CP101 보드가 장착되어 있다면 처음 장착된 보드의 instance값은 0이 되며, 두 번째 장착된 보드의 instance값은 1이 됩니다.
Device Handle	U32	이 함수는 디바이스 핸들을 반환합니다. 이 값은 디바이스를 제어하는 각 함수의 첫번째 파라미터로 사용됩니다. 만일 이 값이 FFFFFFFF(Hex)이면 디바이스 로딩이 실패한 것입니다.

## COMI\_UnloadDevice

### ■ 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 Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.

## AI General



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

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_AD_SetRange	✓	✓				✓	✓			
COMI_AD_GetDigit	✓	✓				✓	✓			
COMI_AD_GetVolt	✓	✓				✓	✓			

아날로그 입력 공통 함수 리스트 및 각 보드별 지원 여부

## COMI\_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 COMI_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. CP 시리즈 보드 ~ -1, -2, -5, -10 SD/LX 시리즈 보드 ~ 0, -1, -2, -5, -10
Max. Volt	Single	A/D범위의 최대값을 지정합니다. 유효한 값은 1, 2, 5, 10 입니다.

## COMI\_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 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch No	I32	Analog Input 채널 번호 (0 – Based)
Return	I16	정수형의 A/D 결과값. 12 Bit resolution 디바이스 (COMI-CP101, COMI-CP201, COMI-SD101)는 0 ~ 4095 사이의 값을 가지며, 16 Bit resolution 디바이스 (COMI-SD201)는 -32768 ~ 32767 사이의 값을 갖는다.

# COMI\_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 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch No	I32	Analog Input 채널 번호 (0 – Based)
Voltage	Single	A/D 결과값 (voltage)



# AI-Scan



AI-Scan 은 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.

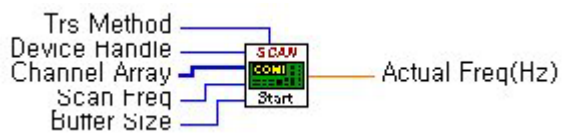
AI-Scan 에 관한 자세한 내용은 소프트웨어 매뉴얼에서 3-2-3 절의 Unlimited Scan 단원을 참조하십시오.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_US_Start	✓	✓				✓	✓			
COMI_US_Stop	✓	✓				✓	✓			
COMI_US_CurCount	✓	✓				✓	✓			
COMI_US_RetrVOne	✓	✓				✓	✓			
COMI_US_RetrVChannel	✓	✓				✓	✓			

Unlimited A/D Scan 함수 리스트 및 각 보드별 지원 여부

## COMI\_US\_Start

### ■ 아이콘



### ■ Function 개요

이 함수는 Unlimited scan 기능을 시작합니다.

## ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_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. 자세한 내용은 소프트웨어 매뉴얼에서 3-2-3절의 Unlimited Scan 단원을 참조하십시오.
Buffer Size	U32	스캔 데이터를 저장할 환형버퍼의 크기를 지정합니다. 이 값의 단위는 바이트가 아니고 SCAN 횟수입니다. 예를 들어, 채널이 2채널일때 bufSize값을 10240으로 지정하면 실제로는 $2 * 10240 * \text{sizeof}(\text{short}) = 40960$ 바이트 크기의 버퍼가 할당됩니다. 이 값은 1024의 배수로 지정하는 것이 좋습니다. 환형버퍼에 대한 자세한 내용은 소프트웨어 매뉴얼에서 3-2-3절의 Unlimited Scan 단원을 참조하십시오.
Trs Method	I32	A/D 디바이스에서 스캔버퍼로 데이터를 전송하는 방식을 지정합니다. 이 값은 다음의 두 값 중의 하나이어야 합니다. 단, CP 시리즈 보드에서는 1만 사용할 수 있습니다. ▷ 1 => CP시리즈 보드의 경우에는 매 채널 A/D 변환이 완료될 때마다 그리고 SD 시리즈 보드의 경우에는 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 1024개 단위로 사용자 버퍼에 전송합니다. 이 방식은 고속 A/D scan 시에 적합합니다. 단, 이 방식은 CP 시리즈 보드에서는 사용할 수 없다.

Actual Freq	132	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.
-------------	-----	--

## COMI\_US\_Stop

### ■ 아이콘



### ■ Function 개요

Unlimited Scan 을 종료합니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
IsSuccess	132	함수의 실행 성공 여부. 0이면 실패 1이면 성공

## COMI\_US\_CurCount

### ■ 아이콘



### ■ Function 개요

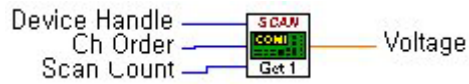
이 함수는 현재까지 수행된 SCAN 횟수를 반환합니다. 사용자는 버퍼에서 데이터를 취할 때에 이 함수를 참조하여 가장 최근 스캔된 데이터의 위치를 알아낼 수 있습니다.

#### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Scan Count	U32	현재까지 수행된 총 SCAN 횟수

## COMI\_US\_RetrVOne

#### ■ 아이콘



#### ■ Function 개요

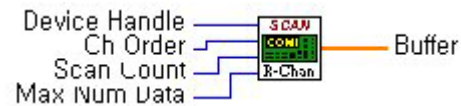
이 함수는 스캔 버퍼에서 하나의 데이터를 취할 때 사용됩니다.

#### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch Order	I32	데이터를 취하기 원하는 채널의 채널 리스트 상의 순서(0 based)입니다. 채널 리스트는 COMI_US_Start 함수를 수행할 때 전달되는 파라미터입니다. 이 값은 채널 번호가 아님을 주의하여야 합니다.
Scan Count	U32	원하는 데이터의 Scan count.
Scan Count	U32	Ch Order와 Scan Count에 의해 지정된 데이터를 Voltage 형식으로 반환합니다.

## COMI\_US\_RetrVChannel

#### ■ 아이콘



## ■ Function 개요

이 함수는 A/D Scan 채널 중에서 하나의 채널에 대한 데이터 블록을 Voltage 값으로 환산하여 전달합니다. 데이터 블록은 사용자가 지정한 startCount 에서부터 maxNumData 에서 지정한 수만큼이 됩니다.

## ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Ch Order	I32	데이터를 취하기 원하는 채널의 채널 리스트 상의 순서(0 based)입니다. 채널 리스트는 COMI_US_Start함수를 수행할 때 전달되는 파라미터입니다. 이 값은 채널 번호가 아님을 주의하여야 합니다.
Scan Count	U32	전달 받고자 하는 데이터 블록의 시작 Scan count.
Max Num Data	U32	전달 받고자 하는 데이터 블록의 크기(데이터 수)를 지정 합니다. 이 값이 <u>양수</u> 이면 startCount부터 이후에 스캔된 데이터 중 maxNumData에서 지정한 수만큼 데이터를 전달합니다. 이 값이 <u>음수</u> 이면 startCount부터 이전에 스캔된 데이터 중 maxNumData에서 지정한 수만큼 데이터를 전달합니다.
Buffer	Single Array	스캔버퍼로부터 사용자가 지정한 조건에 의하여 취해진 A/D 데이터 블록. A/D 데이터는 Voltage값으로 취해집니다.

# Analog Out



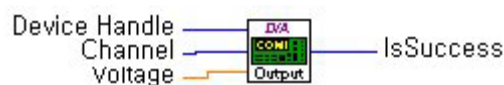
Analog Out Function 블록은 Analog Output 에 관련된 함수들을 모아 놓은 것입니다. COMIDAS 에서는 두 가지 형태의 Analog Output 기능이 있습니다. 첫 번째는 일반적인 Analog Output 기능으로써 사용자가 지정한 전압을 출력하는 기능입니다. 두 번째는 Waveform Generation 기능입니다. Waveform Generation 기능은 Sine Wave 또는 Square Wave 등과 같이 사용자가 지정하는 주기성을 가지는 신호를 자동으로 생성해주는 기능입니다. 이 기능은 COMI-SD301 보드의 0 번 채널과 1 번채널에서만 지원하는 기능입니다.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_DA_Out	✓		✓			✓		✓		
COMI_WFM_Start								✓		
COMI_WFM_RateChange								✓		
COMI_WFM_GetCurPos								✓		
COMI_WFM_GetCurLoops								✓		
COMI_WFM_Stop								✓		

아날로그 출력 함수 리스트 및 각 보드별 지원 여부

## COMI\_DA\_Out

### ■ 아이콘



### ■ Function 개요

사용자가 지정한 Analog Output 채널에 지정한 Voltage 를 출력합니다.

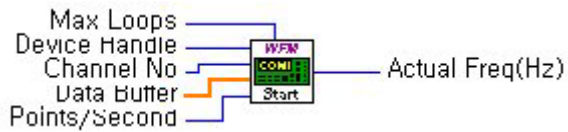
### ■ Parameters

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

Channel	I32	Analog Output 채널 번호 (0 – Based)
Voltage	Single	출력 Voltage
IsSuccess	I32	함수의 성공 여부. 0 – 실패, 1 – 성공

## COMI\_WFM\_Start

### ■ 아이콘



### ■ Function 개요

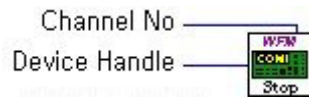
Waveform Generation 을 시작합니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	I32	Analog Output 채널 번호 (0 – Based)
Data Buffer	Single Array	Waveform 데이터를 담은 버퍼. Waveform Generation 기능을 지원하는 보드는 이 버퍼에 담겨진 데이터를 보드에 내장된 FIFO 메모리에 로드하므로 이 버퍼는 지역변수이어도 상관 없습니다.
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	I32	이 값이 0보다 크면 생성되는 Wave 신호의 수를 제한합니다. 이 값이 0이면 COMILX_WFM_Stop()함수가 수행되기 전까지 계속하여 Wave 신호를 생성합니다.
Actual Freq	Single	실제로 설정되는 Points/Second를 반환합니다. 사용자가 지정한 PPS와 실제로 설정되는 PPS는 약간의 차이가 있을 수 있습니다. 이 값이 0보다 작으면 수행도중 에러가 발생하였음을 의미합니다.

## COMI\_WFM\_Stop

### ■ 아이콘



### ■ Function 개요

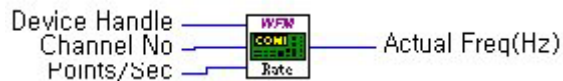
Waveform Generation 을 종료합니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	I32	Analog Output 채널 번호 (0 – Based)

## COMI\_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 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	I32	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.
Actual Freq	Single	실제로 설정되는 Points/Second를 반환합니다. 사용자가 지정한 PPS와 실제로 설정되는 PPS는 약간의 차이가 있을 수 있습니다. 이 값이 0보다 작으면 수행도중 에러가 발생하였음을 의미합니다.

## COMI\_WFM\_GetCurPos

### ■ 아이콘



### ■ Function 개요

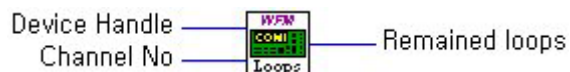
현재 출력되고 있는 주기 데이터의 위치를 반환합니다. 즉, 현재 출력되고 있는 데이터 포인트가 주기 데이터의 몇 번째 데이터인지를 알려줍니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	I32	Analog Output 채널 번호 (0 - Based)
Current Data Index	I32	현재 출력되고 있는 주기 데이터의 인덱스(Index).

## COMI\_WFM\_GetCurLoops

### ■ 아이콘



### ■ Function 개요

이 함수는 COMI\_WFM\_Start 함수의 nMaxLoops 파라미터를 양의 값으로 설정하였을 때만 사용 가능한 함수입니다. 이 함수는 현재 남아있는 Wave 신호의 주기 수를 반환함

니다. 예를 들어 nMaxLoops 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 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel No	I32	Analog Output 채널 번호 (0 – Based)
Remained loops	I32	Wave신호의 출력 횟수를 제한한 경우 현재 남아있는 Wave신호의 출력 횟수. 이 값이 0이면 지정한 횟수의 Wave신호가 모두 출력되었음을 의미합니다.

# Digital In/Out



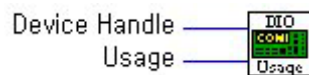
Digital In/Out Function 블록은 Digital In/Out 에 관련된 함수들을 모아 놓은 것입니다. 일반적으로 Digital Input 은 스위치(Switch)의 상태를 읽어들이는데 사용되고, Digital Output 은 스위치의 상태를 제어하는데 사용됩니다.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_DIO_SetUsage				✓				✓		✓
COMI_DI_GetOne	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMI_DI_GetAll	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMI_DO_PutOne	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMI_DO_PutAll	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

디지털 입출력 함수 리스트 및 각 보드별 지원 여부

## COMI\_DIO\_SetUsage

### ■ 아이콘



### ■ Function 개요

DIGITAL In/Out 채널의 용도를 지정합니다. 디바이스에 따라서 Digital In/Out 채널의 용도를 변경할 수 있습니다. DIO 채널의 수는 디바이스에 따라 다르므로 하드웨어 매뉴얼을 참조하십시오.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Usage	I32	디바이스의 용도를 선택합니다. 이 값은 다음의 세 값 중 하나이어야 합니다. ▷ 0 => DIO 채널 모두를 Digital Input 채널로 사

		<p>용.</p> <p>▷ 1 =&gt; DIO 채널을 Input과 Output으로 절반씩 나누어 사용.</p> <p>▷ 3 =&gt; DIO 채널 모두를 Digital Output 채널로 사용.</p>
--	--	---

## COMI\_DI\_GetOne

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 Digital Input 채널의 Status를 반환합니다.

### ■ Parameters

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

## COMI\_DI\_GetAll

### ■ 아이콘



### ■ Function 개요

이 함수는 해당 디바이스의 모든 Digital Input 채널의 Status를 반환합니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
dwStatuses	U32	CH0 ~ CH31의 32채널의 Digital Input 채널의 Status를 나타내는 32 bit 값. 이 값의 각 비트의 값이 각 채널의 Status를 나타냅니다.

## COMI\_DL\_GetAllEx

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 디지털 입력 디바이스로부터 32 개 채널의 Status 를 32 비트값으로 반환합니다. 이 함수는 32 채널보다 많은 디지털 입력 채널을 제공하는 장치에서 32 채널씩 데이터를 읽어들이 수 있도록 하기 위한 함수로써, 이때의 채널은 GroupIndex 파라미터에 따라 달라집니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Group Index	I32	32채널 단위의 채널 그룹인덱스를 지정합니다. 예를 들어 이 값이 0이면 CH0~CH31의 Status를, 1이면 CH32~CH63의 Status를 반환하도록 합니다.
dwStatuses	U32	32개의 채널에 대한 Input Status를 32비트 값으로 반환합니다. 각비트는 비트 순서와 일치하여 각 채널의 ON/OFF 상태를 나타냅니다.

# COMI\_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 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Digital Output 채널번호. 채널 번호는 0부터 시작한다.
Status	Boolean	출력 Status. 0 - OFF, 1 - ON.
IsSuccess	I32	함수 수행 성공 여부.

# COMI\_DO\_PutAll

■ 아이콘



■ Function 개요

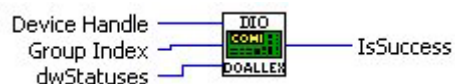
이 함수는 해당 디바이스의 모든 Digital Output 채널에 출력을 내보낸다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
dwStatuses	U32	CH0~CH31의 32채널의 Digital Output 출력 Status를 나타내는 32 bit 값. 이 값의 각 비트의 값이 각 채널의 Status를 나타냅니다.
IsSuccess	I32	함수 수행 성공 여부.

## COMI\_DO\_PutAllEx

### ■ 아이콘



### ■ Function 개요

이 함수는 32 개 Digital Output 채널에 출력을 내보냅니다. 이 함수는 32 채널보다 많은 디지털 출력 채널을 제공하는 장치에서 32 채널씩 출력을 제어할 수 있도록 하기 위한 함수로써, 이때의 채널은 GroupIndex 파라미터에 따라 달라집니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Group Index	I32	32채널 단위의 채널 그룹인덱스를 지정합니다. 예를 들어 이 값이 0이면 CH0~CH31에 출력을 내보내고, 1이면 CH32~CH63에 출력을 내보내도록 합니다.
dwStatuses	U32	Digital Input 채널번호. 채널번호는 0부터 시작합니다.
IsSuccess	I32	함수 수행 성공 여부.

# 8254 Counter



8254 Counter Function 블록은 8254 Counter 에 관련된 함수들을 모아 놓은 것입니다. COMI-CP101, COMI-CP201, COMI-CP501 보드에는 사용자가 사용할 수 있는 Intel 8254 카운터가 장착되어 있습니다. 8254 카운터는 펄스를 카운트하거나, 일정 주기의 펄스를 출력하는데 사용됩니다. Intel 8253/4 카운터에 대한 자세한 내용은 하드웨어 매뉴얼의 부록을 참조하기 바랍니다.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_SetCounter	✓	✓			✓					
COMI_LoadCount	✓	✓			✓					
COMI_ReadCount	✓	✓			✓					

8254 Counter 함수 리스트 및 각 보드별 지원 여부

## COMI\_SetCounter

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 카운터 채널의 동작 방식을 결정하고, 원하는 카운트 값을 로드 (Load)합니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Read_Load	I32	Read/Write mode를 선택합니다. 이 값은 다음 중 하나

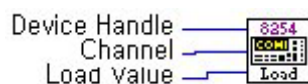


		의 값이어야 합니다. ▷ 1 => 상위 1 바이트값을 로드합니다. ▷ 2 => 하위 1 바이트값을 로드합니다. ▷ 3 => 2 바이트값을 로드합니다.
Operation Mode	I32	카운터의 동작 모드(Operation mode)를 결정합니다. 이 값은 CMODE0 , CMODE1, CMODE2, CMODE3, CMODE4, CMODE5 중의 하나이어야 합니다. Operation mode에 관한 자세한 내용은 하드웨어 매뉴얼의 부록을 참조하십시오.
Load Value	U16	카운터에 로드할 값을 지정합니다. 이 값은 0 ~ 65535 사이의 값이어야 합니다.
BIN_BCD	I32	카운트를 BCD값으로 할 것인지, BINARY 값으로 할 것인지를 결정합니다. ▷ 0 => BINARY ▷ 1 => BCD

🔗 지원 디바이스 : COMI-CP101, COMI-CP201, COMI-CP501

## COMI\_LoadCount

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 카운터에 카운트 값을 로드합니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Load Value	U16	카운터에 로드할 값을 지정합니다. 이 값은 0 ~

		65535 사이의 값이어야 합니다.
--	--	---------------------

☞ 지원 디바이스 : COMI-CP101, COMI-CP201, COMI-CP501

# COMI\_ReadCount

## ■ 아이콘



## ■ Function 개요

이 함수는 지정한 카운터에 카운트 값을 로드합니다.

## ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Count	U16	카운트 값

☞ 지원 디바이스 : COMI-CP101, COMI-CP201, COMI-CP501

# Encoder Counter



Encoder Counter Function 블록은 Encoder Counter 에 관련된 함수들을 모아 놓은 것입니다. Encoder Counter 기능은 COMI-SD501 보드에서만 적용되는 카운터 기능으로써 주로 엔코더 센서를 계측하는데 사용되는 카운터 기능입니다. 일반적으로 엔코더 센서는 회전체의 위치나 속도를 계측하기 위해 사용되는 센서입니다.

엔코더에서 발생하는 신호는 A 상, B 상 그리고 Z 상으로 구분되는 3 개의 신호이며 이들은 모두 펄스 형태로 출력되게 됩니다. A 상과 B 상 신호는 회전체의 미소 위치 변화(엔코더의 분해능에 따라 그 값은 다름)가 발생할 때마다 발생하는 펄스 신호로써, 이 두 신호는 일정한 위상차를 가지게 되어 회전 방향까지도 함께 알수 있습니다. COMI-SD501 은 회전 방향에 따라 자동으로 UP/DOWN 카운트가 됩니다. Z 상 신호는 회전체가 1 회전할 때마다 발생하는 펄스 신호입니다. COMI-SD501 은 Z 상 신호가 A/B 상 카운터의 값을 자동으로 리셋되도록 설정될 수 있습니다. 이 방식을 이용하면 회전체의 절대 위치를 파악하는데 용이합니다.

COMI-SD501 보드는 엔코더의 3 가지 신호를 모두 입력받을 수 있으며, A/B 상과 Z 상을 동시에 계측할 수 있습니다. 이 단원에서 소개되는 엔코더 카운터 함수중에서 'Z'자가 함수명 끝에 붙은 함수는 Z 상 신호를 계측하는데 사용되는 함수이며, 나머지 함수들은 모두 A/B 상 신호를 계측하는데 사용되는 함수입니다.

## ※ COMI-SD501 디바이스에서 일반 펄스를 카운트하는 방법

COMI-SD501 에서는 엔코더 신호뿐만 아니라 일반 펄스 신호도 카운트할 수 있습니다. 일반 펄스 신호는 A 상과 B 상이 따로 존재하지 않으므로 신호선을 어떻게 연결해야 할지 혼동될 수 있습니다. 일반 펄스 신호를 카운트하기 위해서는 펄스 신호를 터미널 보드의 A 상 입력단에 연결하면 됩니다. 사용되는 라이브러리 함수는 엔코더 신호를 카운트하는 것과 동일합니다.

한편, 일반 펄스 신호를 카운트할 때에도 B 상 입력단에 0 Volt 또는 5 Volt 를 인가하여 Up/down 카운트를 할 수도 있습니다. B 상 입력단에 0 Volt 가 인가되면 업카운트(Up-count)가 되고, 5 Volt 가 인가되면 다운카운트(Down-count)가 됩니다. 만일 B 상 단자에 아무 신호도 연결되어 있지 않으면 업카운트를 하게 되며 펄스가 입력될 때마다 카운트는 증가하게 됩니다.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_ENC_Config									✓	
COMI_ENC_Reset									✓	
COMI_ENC_Load									✓	
COMI_ENC_Read									✓	
COMI_ENC_ResetZ									✓	
COMI_ENC_LoadZ									✓	
COMI_ENC_ReadZ									✓	

Encoder Counter 함수 리스트 및 각 보드별 지원 여부

## COMI\_ENC\_Config

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 카운터에 카운트 값을 로드합니다.

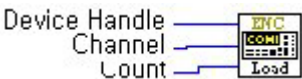
### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
ResetByZ	Boolean	A/B 상 카운트값을 Z상 입력단자에 펄스가 입력될 때마다 리셋(Reset)할 것인지를 지정합니다. ▷ <b>FALSE</b> => 이 값으로 지정하면 A/B상 카운트값은 Z상에 의해 영향을 받지 않습니다. ▷ <b>TRUE</b> => 이 값으로 지정하면 A/B상 카운트값은 Z-펄스가 발생할 때마다 리셋(Reset)됩니다.

🔗 지원 디바이스 : COMI-SD501

# COMI\_ENC\_Load

■ 아이콘



■ Function 개요

이 함수는 지정한 A/B 상 카운터 채널의 카운트값을 지정한 값으로 로드(Load)합니다. 일반적으로 이 함수는 회전체가 원점에서 시작되지 않는 경우 회전체의 초기 위치를 지정하기 위하여 사용됩니다. 카운트값을 0 으로 초기화하기 위해서는 이 함수를 사용하는 것보다 COMI\_ENC\_Reset 함수를 사용하는 것이 속도면에서 유리합니다.

■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Count	I32	카운터에 로드할 값. 이 값의 범위는 -2147483648 ~ 2147483647 입니다.

🔗 지원 디바이스 : COMI-SD501

# COMI\_ENC\_LoadZ

■ 아이콘



■ Function 개요

이 함수는 지정한 Z 상 카운터 채널의 카운트값을 지정한 값으로 로드(Load)합니다. 일반적으로 이 함수는 회전체가 한방향으로만 회전하는 경우에 Z 상 카운터의 16 비트값의 전체 영역을 모두 사용하기 위한 것입니다. 기본적으로 Z 상 카운터의 초기값

은 0 이므로 정방향으로 회전시 최대 계측 가능한 펄스 수는 32767 회가 됩니다. 그러나 초기값을 -32767 값으로 로드한 후 계측한다면 정방향으로 회전시 65535 회의 펄스를 계측할 수 있습니다.

#### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Count	I32	카운터에 로드할 값. 이 값의 범위는 -32767 ~ 32768 입니다.

🔗 지원 디바이스 : COMI-SD501

## COMI\_ENC\_Read

#### ■ 아이콘



#### ■ Function 개요

이 함수는 지정한 A/B 상 카운터 채널의 카운트값을 읽어서 그 값을 반환합니다.

#### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Count	U16	A/B상 카운트 값. 카운트 값의 범위는 32 비트 정수 값으로써 -2147483648 ~ 2147483647 입니다. 반환된 카운트 값이 음의 값이면 회전체가 역 방향으로 회전했음을 의미하며, 양의 값이면 정 방향으로 회전했음을 의미합니다.

☞ 지원 디바이스 : COMI-SD501

## COMI\_ENC\_ReadZ

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 Z 상 카운터 채널의 카운트 값을 읽어서 그 값을 반환합니다.

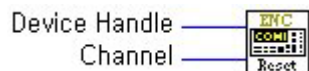
### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Count	U16	Z상 카운트 값. 카운트 값의 범위는 16 비트 정수 값으로써 -32768 ~ 32767입니다. 반환된 카운트 값이 음의 값이면 회전체가 역 방향으로 회전했음을 의미하며, 양의 값이면 정 방향으로 회전했음을 의미합니다.

☞ 지원 디바이스 : COMI-SD501

## COMI\_ENC\_Reset

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 A/B 상 카운터 채널의 카운트 값을 0 으로 리셋합니다.

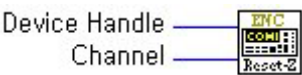
■ Parameters

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

☞ 지원 디바이스 : COMI-SD501

# COMI\_ENC\_ResetZ

■ 아이콘



■ Function 개요

이 함수는 지정한 Z 상 카운터 채널의 카운트 값을 0으로 리셋합니다.

■ Parameters

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

☞ 지원 디바이스 : COMI-SD501



# Pulse Generator



Pulse Generator(이후 PG) 기능은 COMI-SD501 보드에서만 제공하는 기능으로써 사용자가 원하는 주파수로 펄스를 생성시켜주는 기능입니다. 이 기능은 생성되는 펄스의 주파수를 제어할 수 있을 뿐 아니라 펄스의 수까지 제어할 수 있습니다. 이 기능은 주로 서보모터 및 스텝모터의 제어에 사용되는 기능입니다.

함수명	보드별 지원 여부									
	CP101	CP201	CP301	CP401	CP501	SD10x	SD201	SD301	SD501	SD502
COMI_PG_Start									✓	
COMI_PG_ChangeFreq									✓	
COMI_PG_IsActive									✓	
COMI_PG_Stop									✓	

Pulse Generator 함수 리스트 및 각 보드별 지원 여부

## COMI\_PG\_Start

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 Z 상 카운터 채널의 카운트 값을 0으로 리셋합니다.

### ■ Parameters

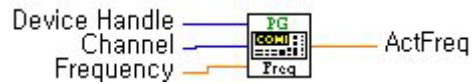
Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Frequency	Double	출력할 펄스의 주파수를 Hz단위로 지정합니다.
NumPulses	U32	출력할 펄스의 수를 지정합니다. 만일 이 값을 0으로 지정하면 COMI_PG_Stop()함수가 수행되기 전까지 계속

		해서 펄스를 출력하게 됩니다.
ActFreq	Double	출력되는 펄스의 실제 주파수. 펄스의 주파수는 장치 내부에 장착된 타이머의 Base clock을 정수로 분주하여 결정하므로 사용자가 지정한 주파수와 약간의 차이가 있을 수 있습니다.

👉 **지원 디바이스** : COMI-SD501

## COMI\_PG\_ChangeFreq

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 PG 채널을 통하여 현재 출력되고 있는 펄스의 주파수를 Runtime 상에서 변경하여 줍니다.

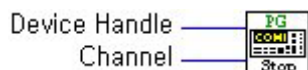
### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Frequency	Double	출력할 펄스의 주파수를 Hz단위로 지정합니다.
ActFreq	Double	출력되는 펄스의 실제 주파수. 펄스의 주파수는 장치 내부에 장착된 타이머의 Base clock을 정수로 분주하여 결정하므로 사용자가 지정한 주파수와 약간의 차이가 있을 수 있습니다.

🔍 지원 디바이스 : COMI-SD501

## COMI\_PG\_Stop

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 PG 채널의 펄스 출력을 중지합니다.

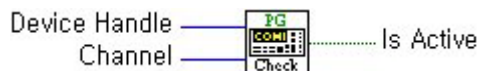
### ■ Parameters

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

🔗 지원 디바이스 : COMI-SD501

## COMI\_PG\_IsActive

### ■ 아이콘



### ■ Function 개요

이 함수는 지정한 PG 채널에 현재 펄스가 출력되고 있는지를 알려줍니다. 이 함수를 이용하면 출력 펄스 수를 제한한 경우에 원하는 펄스의 출력이 완료되었는지를 알아볼 수 있습니다.

### ■ Parameters

Parameters	Type	Description
Device Handle	U32	디바이스 핸들 값입니다. 이 값은 COMI_LoadDevice 함수에 의해 얻어진 값이어야 합니다.
Channel	I32	Counter 채널번호. 채널 번호는 0부터 시작합니다.
Is Active	Boolean	현재 펄스가 출력되고 있는지를 알려줍니다. <b>FALSE</b> => 현재 펄스가 출력되고 있지 않음 <b>TRUE</b> => 현재 펄스가 출력되고 있음

🔗 지원 디바이스 : COMI-SD501

