

COMIZOA EtherCAT Software Development Kit Basic

TEST & MEASUREMENT & AUTOMATION COMIZOA LX551 / LX552 / LX554 API REFERENCE MANUAL

JULY 2015
P/N 0728-2015-07
© 2015 COMIZOA Inc. All rights reserved

ComiECAT MASTER Manual

Copyright © 2015 by COMIZOA, Inc. All rights reserved.

COMIZOA owns all right, title and interest in the property and products described herein, unless otherwise indicated. No part of this document may be translated to another language or produced or transmitted in any form or by any information storage and retrieval system without written permission from COMIZOA. COMIZOA reserves the right to change products and specifications without written notice. Customers are advised to obtain the latest versions of any product specifications.

COMIZOA MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OTHER THAN COMPLIANCE WITH THE APPLICABLE COMIZOA SPECIFICATION SHEET FOR THE PRODUCT AT THE TIME OF DELIVERY. IN NO EVENT SHALL COMIZOA BE LIABLE FOR ANY INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES AS A RESULT OF THE PRODUCT'S PERFORMANCE OR FAILURE TO MEET ANY ASPECT OF SUCH SPECIFICATION. COMIZOA PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN LIFE SUPPORT APPLIANCES, DEVICES OR SYSTEMS WHERE A MALFUNCTION OF A COMIZOA DEVICE COULD RESULT IN A PERSONAL INJURY OR LOSS OF LIFE. CUSTOMERS USING OR SELLING COMIZOA DEVICES FOR USE IN SUCH APPLICATIONS DO SO AT THEIR OWN RISK AND AGREE TO FULLY INDEMNIFY COMIZOA FOR ANY DAMAGES RESULTING FROM SUCH IMPROPER USE OR SALE.

Information contained herein is presented only as a guide for the applications of our products. COMIZOA does not warrant this product to be free of claims of patent infringement by any third party and disclaims any warranty or indemnification against patent infringement. No responsibility is assumed by COMIZOA for any patent infringement resulting from use of its products by themselves or in combination with any other products. No license is hereby granted by implication or otherwise under any patent or patent rights of COMIZOA or others. COMIZOA software and its documentation are available only under the terms of a Master Software Use and Support Agreement.

Trademarks

The COMIZOA logo is a registered trademark. All other brand names, product names, trademarks, and registered trademarks are the property of their respective owners.

Visit our web page at <http://www.comizoa.com>

For support requests, contact us at csteam@comizoa.com

For documentation suggestions, corrections, or requests, contact tech@comizoa.com

고객(顧客) 기술 지원

전자 우편 : csteam@comizoa.com

파일 서버 : <ftp.comizoa.com>

웹 사이트 : <http://www.comizoa.com>

본사 안내

대전광역시 유성구 테크노 2로 314번지 (탑립동 914번지)

전화번호 : 042-936-6500

모사전송 : 042-936-6507

COMIZOA ECAT MASTER System Integrated Control Library Reference

© 2015 COMIZOA

All Rights Reserved. No Part of this publication may be reproduced, stored in retrieval system or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission, in writing, from the publisher.

Table of Contents

Trademarks	1-2
Tble of Contents	3
1 ComiECAT API 소개	6
1.1 함수의 명명 규칙	6
1.1.1 함수 이름의 접두어	6
1.1.2 함수 이름의 접미어	7
1.2 데이터형 표기	8
2 DLL Functions	11
2.1 함수 요약	11
2.2 함수 설명	12
3 General Functions	16
3.1 함수 요약	16
3.2 함수 설명	17
4 Net Interface Functions	22
4.1 함수 요약	22
4.2 함수 설명	23
5 Slave I/F Functions	30
5.1 함수 요약	30
5.2 함수 설명	31
6 Digital I/O Functions	41
6.1 함수 요약	41
6.2 함수 설명	43
6.2.1 전역채널(Global Channel) 번호를 이용하는 함수군	45
6.2.2 지역채널(Local Channel) 번호를 이용하는 함수군	54
7 Analog I/O Functions	60
7.1 함수 요약	60
7.2 함수 설명	61
8 Motion General Functions	69
8.1 함수 요약	69
8.2 함수 설명	70
9 Motion Single Axis Functions	73

9.1	함수 요약	73
9.2	함수 설명	77
9.2.1	기본 환경 설정	77
9.2.2	상태 제어 명령	94
9.2.3	이송 명령	97
9.2.4	모션 상태 감시	113
10	<i>Motion InterPolation Functions</i>	124
10.1	함수 요약	124
10.2	함수 설명	126
11	<i>Homing Operation Functions</i>	145
11.1	함수 요약	145
11.2	함수 설명	146

ComiECAT Introduction

다년간의 보다 강력하고 편리한 라이브러리 기술개발을 통해 자신있게 제공하여 드리는 ComiECAT 은 편리한 함수명의 규칙(Rule)을 통해 사용자 편의성을 극대화 하였습니다. (주)커미조아 ComiECAT 의 최신 기능과속련된 모션 제어 기술은 결코 흉내낼 수 없는 커미조아의 기술입니다. 지금 확인(確認)하십시오.

본 장에서는 ComiECAT 이 제공하는 라이브러리 인터페이스에 대한 자세한 설명(說明)을 안내합니다. ComiECAT 은 라이브러리 기능을 보다 강력하고 효율적으로 지원할 수 있는 다양한 런타임(Run-time) 인터페이스와 라이브러리의 다양한 기능을 직관적(直觀的)으로 제공합니다. 본 매뉴얼에서는 ComiECAT 에서 제공하는 라이브러리 함수에 대한 설명(說明)을 기능에 따라 그룹별로 수록 하였습니다.



1 ComiECAT API 소개

1.1 함수의 명명 규칙

ComiECAT에서 제공하는 모든 함수는 다른 API 함수와 이름이 중복되는 것을 피하기 위하여 아래의 예와 같이 “ec”이라는 접두어가 붙습니다.

ecDll_Load(), ecDll_Unload(), ecGn_LoadDevices(), ecGn_UnloadDevices (),...

그리고 “ec” 접두어 바로 다음에는 해당 함수가 속하는 기능의 그룹을 대표하는 접두어가 이어집니다. 이러한 이유는 동일한 기능 그룹에 속한 함수들을 쉽게 구분할 수 있고, 함수가 리스트될 때에 일반적으로 알파벳순으로 정렬되므로 동일 기능 그룹에 속한 함수들이 함께 리스트될 수 있도록 하기 위함입니다.

1.1.1 함수 이름의 접두어

ComiECAT SDK에서 제공하는 함수들의 대표적인 접두어를 살펴보면 아래의 표와 같은 것들이 있습니다.

구분	접두어	설명	적용 함수 예
1. General Functions	ecGn_	특정 기능군에 속하지 않고, 라이브러리 전체에서 일반적으로 사용될 수 있는 기능을 제공하는 함수군	ecGn_LoadDevices, ecGn_UnloadDevices, ecGn_GetNumDevices
2. 네트워크 인터페이스	ecNet_	이더넷 네트워크의 제어와 상태 감시 및 데이터의 읽기 쓰기 관련한 기능을 제공하는 함수군	ecNet_IsBootCompt, ecNet_SetAIState, ecNet_GetAIState
3. 슬레이브 인터페이스	ecSlv_	슬레이브와의 인터페이스 기능을 제공하는 함수군	ecSlv_GetAIState, ecSlv_ReadReg, ecSlv_WriteReg
4. 디지털입력	ecdi	디지털입력과 관련된 기능을 제공하는 함수군	ecdiGetOne, ecdiGetMulti, ecdiGetNumChannels
5. 디지털출력	ecdo	디지털출력과 관련된 기능을 제공하는 함수군	ecdiGetOne, ecdiGetMulti, ecdiGetNumChannels
6. 아날로그 입력	ecai	아날로그 입력(A/D)과 관련된 기능을 제공하는 함수군	ecaiGetNumChannels, ecaiGetChanVal_I, ecaiGetChanVal_F
7. 아날로그 출력	ecao	아날로그 출력과 관련된 기능을 제공하는 함수군	ecaoGetNumChannels, ecaoSetChanVal_I, ecaoSetChanVal_F
8. 모션 제어	ecm	모션제어와 관련한 함수군. 모션제어 함수군들은 그 수가 많아서 다시 세부 기능별로 접두어가 추가됨	ecmGn_GetAxisList, ecmSxCfg_SetUnitDist, ecmMxMot_VMoveStart,
8.1. 단축 환경설정	ecmSxCfg_	단축의 환경 설정과 관련한 기능을 제공하는 함수군	ecmSxCfg_SetUnitDist, ecmSxCfg_SetUnitSpeed, ecmSxCfg_SetSpeedPatt
8.2. 단축 상태 제어	ecmSxCtl_	서보온/오프와 같이 단축의 상태를 제어하는 함수군(이송 함수는 제외)	ecmSxCtl_SetSvon, ecmSxCtl_SetAlmRst, ecmSxCtl_ResetAlm
8.3. 단축 이송 명령	ecmSxMot_	단축의 이송과 관련된 명령의 함수군	ecmSxMot_VMoveStart, ecmSxMot_MoveStart, ecmSxMot_Stop

8.4. 단축 상태 감시	ecmSxSt_	단축의 모션 상태를 감시하거나 설정하는 기능을 제공하는 함수군	ecmSxSt_IsBusy, ecmSxSt_GetMotState ecmSxSt_SetPosition
8.5. 다축 이송 명령	ecmMxMot_	다축(Multiple Axes)의 이송과 관련된 명령의 함수군	ecmMxMot_VMoveStart, ecmMxMot_MoveStart, ecmMxMot_Stop
8.6. 다축 상태 감시	ecmMxSt_	다축의 모션 상태를 감시하거나 설정하는 기능을 제공하는 함수군	ecmMxSt_IsBusy, ecmMxSt_WaitCompt ecmMxSt_WaitCompt_NB
8.7. 다축 보간 환경 설정	ecmIxCfg_	다축 보간 이송의 환경을 설정하는 기능을 제공하는 함수군	ecmIxCfg_MapAxes, ecmIxMot_ArcAng_R, ecmIxMot_Stop
8.8. 다축 보간 이송 명령	ecmIxMot_	다축 보간 이송과 관련된 명령의 함수군	ecmIxMot_LineStart, ecdiGetMulti, ecdiGetNumChannels
8.9. 다축 보간 상태 감시	ecmIxSt_	다축 보간의 모션 상태를 감시하거나 설정하는 기능을 제공하는 함수군	ecmIxSt_IsBusy, ecmIxSt_WaitCompt ecmIxSt_WaitCompt_NB
8.10. 원점복귀 환경 설정	ecmHomeCfg_	원점복귀 구동의 환경을 설정하는 기능을 제공하는 함수군	ecmHomeCfg_MapAxes, ecmIxMot_ArcAng_R, ecmIxMot_Stop
8.11. 원점복귀 이송 명령	ecmHomeMot_	원점복귀 이송 명령과 관련된 함수군	ecmIxMot_LineStart, ecdiGetMulti, ecdiGetNumChannels
8.12. 원점복귀 상태 감시	ecmHomeSt_	원점복귀 구동의 상태를 감시하거나 설정하는 기능을 제공하는 함수군	ecmIxSt_IsBusy, ecmIxSt_WaitCompt ecmIxSt_WaitCompt_NB

1.1.2 함수 이름의 접미어

ComiECAT SDK 에서 제공하는 함수들의 일부에서는 기능을 구분하기 위해서 특정 접미어를 붙여서 명명하였습니다. ComiECAT SDK 에서 사용하는 주요 함수 접미어는 다음 표와 같습니다.

구분	접미어	설명	적용 함수 예
1. FastFunctions	_FF	‘4.1.2.1 FAST FUNCTION 모드’ 설명 참고	ecSlv_WriteReg_FF, ecmSxCfg_SetSpeedPatt_FF, ecmSxMot_MoveStart_FF
2. Non-Blocking Functions	_NB	‘4.1.2.2 NON-BLOCKING 모드’ 설명 참고	ecmSxMot_Move_NB, ecmSxMot_MoveTo_NB, ecmSxSt_WaitCompt_NB
3. 연결 순서에 따른 슬레이브 지정 방식	_A	슬레이브와의 인터페이스 함수들 중에서 _A 접미어가 붙은 함수들은 슬레이브 주소를 연결 순서값으로 지정함. _A 가 붙지 않으면 슬레이브 물리주소 값을 사용함. 단, 슬레이브 인터페이스 함수들 이외의 다른 함수들에서 사용된 _A 접미어는 이와 다른 의미를 가지는 것임.	ecSlv_GetAIState_A, ecSlv_ReadReg_A, ecSlv_WriteReg_A


4. 지역 채널(Local Channel) 지정 방식	└	채널 번호를 지정할 때 슬레이브 주소와 슬레이브 내부에서의 채널 번호로 채널을 지정하는 방식의 함수임을 의미함	ecdiGetOne└, ecdiGetMulti└, ecdoPutOne└
-------------------------------	---	---	---

1.1.2.1 FAST FUNCTION 모드

많은 수의 ComiECAT SDK 함수들은 ComiECA 마스터 장치에 독립적으로 존재하는 프로세서에 전달되고, 해당 프로세서에서 실행되게 됩니다. 이러한 함수들은 사용자 프로그램에서 호출되면 마스터 장치의 프로세서로 전달되어서 해당 프로세서에서 해당 명령이 실행된 후에 그 결과값을 반환 받은 후에 사용자 프로그램으로 반환됩니다. 따라서 일반적으로 해당 함수들은 실행에 필요한 시간만큼 함수 내부에서 지연이 발생하게 됩니다.

Fast Function 모드는 위에서 말한 바와 같이 함수가 실행에 소요되는 시간동안 지연이 발생되게 되는 부분을 제거하기 위해서 사용되는 모드입니다. Fast Function 모드를 사용하면 사용자가 호출한 함수는 마스터 장치의 프로세서에 해당 명령을 전달하기만 하고 바로 리턴하게 됩니다. 해당 명령이 실제 실행되기까지 기다리지 않습니다. 따라서 함수의 지연 시간이 최소화 됩니다.

Fast Function 모드로 동작하는 함수들은 ‘_FF’ 접미어가 붙습니다.


 <p>주의</p>	Fast Function 모드의 함수들은 함수의 실행이 완료되기 전에 함수에서 반환될 수 있다는 점을 유의해야 합니다.
--	--

1.1.2.2 NON-BLOCKING 모드

ecmSxMot_Move() 함수는 이송을 시작한 후에 이송이 완료될 때까지 해당 함수안에서 기다리다가 이송이 완료되어야만 함수에서 반환됩니다. 따라서 함수 내부에서 비교적 오랫동안 함수가 블로킹(blocking)되게 됩니다. 이러한 함수가 별도의 쓰레드에서 실행되면 상관없지만 윈도우의 이벤트/메시지 핸들러 등에서 실행되면 해당 함수가 반환되기 전까지는 해당 윈도우의 메시지는 처리되지 않게 됩니다.

Non-blocking 모드는 이렇게 함수 내부에서 기다리는 동안에도 윈도우 메시지를 전달하여 처리할 수 있도록 하는 모드입니다.

Non-Blocking 모드로 동작하는 함수들은 ‘_NB’ 접미어가 붙습니다.

 <p>주의</p>	작업 쓰레드에서 ComiECAT SDK 함수를 호출할 때에는 Non-blocking 함수들을 사용하지 말아야 합니다.
---	---

1.2 데이터형 표기

당사의 ComiECAT 인터페이스는 매뉴얼에서 명시한 윈도우 표준 Dynamic Link Library 를 지원하는 어떠한 개발 환경에서도 사용 가능합니다. 하지만 데이터형에 대한 이름은 개발환경에 따라서 서로 다릅니다. 따라서 본 매뉴얼에서는 데이터의 형 표기를 표 1 과 같이 통일하여 표기합니다. 이에 대한 각 컴파일러의 대응되는 데이터형 표기는 표 1 을 참조하여 사용하시기 바랍니다.

그리고 본 매뉴얼에서는 “[in]”과 “[out]” 표기를 사용하여 매개변수가 함수에 전달되는 것인지 아니면 전달받는 것인지를 명시하였습니다. “[in]”은 함수에 값을 전달함을 의미하고, “[out]”은 함수로부터 값을

전달받는다를 것을 의미합니다. 단, 이 표기는 본 매뉴얼에서만 사용되는 것이며, 실제 헤더파일에는 표기되어 있지 않습니다.

Data type	Description	C/C++
t_char	1 바이트 ASCII 문자 코드	char
t_byte	부호가 없는 양의 정수	unsigned char
t_uchar	부호가 없는 양의 정수	unsigned char
t_bool	True 나 False 형태의 논리형	bool
t_i8	1 바이트 ASCII 문자 코드	char
t_ui8	부호가 없는 양의 정수	unsigned char
t_i16	부호가 있는 2 바이트 정수	short
t_word	부호가 없는 2 바이트 정수	unsigned short
t_ui16	부호가 없는 2 바이트 정수	unsigned short
t_i32	부호가 있는 4 바이트 정수	int
t_ui32	부호가 없는 4 바이트 정수	unsigned int
t_dword	부호가 없는 4 바이트 정수	unsigned int
t_f32	부호가 있는 4 바이트 실수	float
t_f64	부호가 있는 8 바이트 실수	double
t_success	True 나 False 형태의 논리형	bool
t_cmdidx	부호가 없는 4 바이트 정수	unsigned int

표 1 언어독립적 데이터 표기 및 각 언어별 대응 데이터 형

DLL functions

㈜커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 (주)커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.



2 DLL Functions

2.1 함수 요약

Summary of Functions
□ t_success ecDll_Load() 라이브러리를 사용자(使用者) 응용프로그램의 사용을 위해 로드(Load) 합니다.
□ t_success ecDll_Unload() 라이브러리를 사용자(使用者) 응용프로그램의 사용 종료(終了)를 위해 언로드(Unload) 합니다.
□ BOOL ecDll_IsLoaded () 라이브러리가 로드 되었는지 알 수 있는 역할을 합니다.

2.2 함수 설명

NAME	INFORMATION
ecDll_Load	 DLL Function
- 라이브러리(Library) 로드	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

□ t_success ecDll_Load()

DESCRIPTION

ComiECAT 을 고객(顧客)님의 응용프로그램의 메모리 공간으로 호출합니다. 이 의미는 이 함수가 호출되는 순간 라이브러리는 고객(顧客)님의 프로그램 내부 함수처럼 호출할 수 있게 됩니다. 이 함수는 고객(顧客)님의 전체 프로그램에서 ComiECAT 을 사용하기 위한 수순으로서는 가장 먼저 호출되어야 합니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	DLL 을 로드하는데 실패 하였음을 의미합니다.
1	DLL 을 <u>성공적으로</u> 로드 하였음을 의미합니다.

SEE ALSO





* ecDll_Unload

EXAMPLE

```
C/C++

#include "ComiEcatSdk_Api.h"

void StartProgram(void) //만드시 함수 처음 부분에 작성
{
    // 이 함수의 반환값은 DLL 의 로드의 성공여부를 반환합니다.
    t_success nIsLoaded = ecDll_Load ();
}
```

<h2>NAME</h2> <p>ecDll_Unload</p> <p>- 라이브러리(Library) 로드 해제</p>	INFORMATION
	 DLL Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```
int t_success ecDll_Unload ()
```

DESCRIPTION

ComiECAT 을 고객(顧客)님의 응용프로그램의 메모리 공간에서 해제합니다. 이 의미는 이 함수가 호출되는 순간 (주)커미조아의 ComiECAT 은 고객(顧客)님의 응용프로그램에서 제거됩니다. 이 함수는 고객(顧客)님의 전체 프로그램에서 ComiECAT 를 사용을 종료 하기 위한 수순으로서는 가장 나중에 호출되어야 합니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	DLL 을 언로드하는데 실패 하였음을 의미합니다.
1	DLL 을 성공적으로 언로드 하였음을 의미합니다.





SEE ALSO

* ecDll_Load

EXAMPLE

```
C/C++
#include "ComiEcatSdk_Api.h"

void EndProgram(void) //반드시 모든 함수 해제 후 DLL 해제
{
    // 이 함수의 반환값은 DLL 의 언로드의 성공여부를 반환합니다.
    ecDll_Unload ();
}
```

<h2>NAME</h2> <p>ecDll_IsLoaded</p> <p>- 라이브러리(Library) 로드 상태 확인</p>	<h3>INFORMATION</h3> <ul style="list-style-type: none">  DLL Function  VC++/BCB/.NET  Level 1  위험 요소 없음
<h2>SYNOPSIS</h2> <p>□ BOOL ecDll_IsLoaded ()</p>	

DESCRIPTION

ComiECAT 이 고객(顧客)님의 응용프로그램의 메모리 공간으로 호출 되었는지를 확인합니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	DLL 을 로드하는데 <u>실패</u> 하였음을 의미합니다.
1	DLL 을 <u>성공적으로</u> 로드하였음을 의미합니다.

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"

void Dll_IsLoaded(void)
{
    If(ecDll_IsLoaded()) //로드 성공
    Else // 로드 실패
}

```

General functions

㈜커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서 저희 (주)커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.







3 General Functions

3.1 함수 요약

Summary of Functions
□ t_success ecGn_LoadDevices ([out] t_i32 *ErrCode) 라이브러리가 로드된 상태에서 장치를 로드(Load) 하는 역할을 합니다.
□ t_success ecGn_UnloadDevices ([out] t_i32 *ErrCode) 라이브러리가 로드된 상태에서 장치를 언로드(Unload) 하는 역할을 합니다.
□ BOOL ecGn_IsDevLoaded ([in] t_i32 DevIdx, [out] t_i32 *ErrCode) 시스템에 장치가 로드 되었는지 알 수 있는 역할을 합니다.

3.2 함수 설명

<h1>NAME</h1> <p>ecGn_LoadDevices</p> <p>- 장치 로드</p>	<h3>INFORMATION</h3> <ul style="list-style-type: none">  General Function  VC++/BCB/.NET  Level 1  위험 요소 없음
<h1>SYNOPSIS</h1> <pre> □ t_success ecGn_LoadDevices([out] t_i32 *ErrCode) </pre>	

DESCRIPTION

시스템에 설치된 하드웨어 장치를 로드합니다. 이 함수는 ComiECAT의 다른 함수가 호출되기 전에 반드시 한번은 수행되어야 합니다. 일반적으로 프로그램의 시작부분에서 수행해주면 됩니다.

PARAMETER

▶ **ErrCode**: 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	장치를 로드하는데 <u>실패</u> 하였음을 의미합니다.
1	장치를 <u>성공적</u> 으로 로드하였음을 의미합니다.

SEE ALSO

* ecGn_UnloadDevices

EXAMPLE

C/C++

```

#include "ComiEcatSdk_Api.h"

t_i32 Error_Num = 0; //함수 별 에러 코드 저장 변수

void Device_Load(void)
{
    TEcDevInfo Device_Info; //디바이스 정보 저장 할 구조체
    t_i32 DevIdx = 0;
    t_i32 Net_Count = 0;
    t_i32 Device_Count = 0;

    if(!ecGn_LoadDevices(&Error_Num)){ //장치 로드
        //로드 실패시 예외 처리
    }
    if(!ecGn_IsDevLoaded(Device_Num, &Error_Num)){ //장치 로드 확인
        //로드 확인 실패시 예외 처리
    }





    t_i32 Device_Count = ecGn_GetNumDevices(&Error_Num); //장치 개수 확인
    t_i32 Net_Count = ecGn_GetNumNetworks(&Error_Num); //네트 워크 개수 확인

    if(!ecGn_GetDevInfo(Device_Num,&Device_Info, &Error_Num)){
        //디바이스 정보 로드 실패 시 예외 처리
    }
}

/* Device_Load (void)함수의 끝 */

Void Device_UnLoad(void)
{
    ecGn_UnloadDevices(&Error_Num); //디바이스 언로드
}

```

<h2>NAME</h2> <p>ecGn_UnloadDevices</p> <p>- 장치 언로드</p>	INFORMATION
	 General Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```
□ t_success ecGn_UnloadDevices([out] t_i32 *ErrCode)
```

DESCRIPTION

시스템에 설치된 하드웨어 장치를 언로드합니다. 이 함수는 ComiECAT 장치 사용을 종료하기 위해 명시적으로 호출되어야 합니다. 일반적으로 프로그램의 종료부에서 수행해주면 됩니다.

PARAMETER

▶ **ErrCode**: 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	장치를 로드하는데 <u>실패</u> 하였음을 의미합니다.
1	장치를 <u>성공적</u> 으로 로드하였음을 의미합니다.

SEE ALSO

* ecGn_LoadDevices


EXAMPLE


* ecGn_LoadDevices EXAMPLE 참고


NAME


ecGn_IsDevLoaded
- 장치 로드 상태 확인

INFORMATION

 General Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_success ecGn_IsDevLoaded([in] t_i32 DevIdx, [out] t_i32 *ErrCode)

DESCRIPTION

현재 장치가 로드 되었는지 장치로드 상태를 확인할 수 있습니다.

PARAMETER

- ▶ DevIdx : 로드 되었는지 확인 할 장치 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	장치가 로드되지 않았음을 의미합니다.
1	장치가 로드되었음을 의미합니다.

EXAMPLE

* ecGn_LoadDevices EXAMPLE 참고

Net Interface functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.







4 Net Interface Functions

4.1 함수 요약

Summary of Functions
<p>□ t_success ecNet_GetVerInfo ([in] t_i32 NetID, [out] TEcFileVerInfo_SDK *pVerInfo_SDK, [out] TEcFileVerInfo_WDM *pVerInfo_WDM, [out] TEcFileVerInfo_Fw *pVerInfo_FW, [out] t_i32 *ErrCode) SDK, 장치 드라이버, 펌웨어의 버전 정보 및 호환성 여부를 반환 합니다.</p>
<p>□ t_ui32 ecNet_ScanSlaves([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 슬레이브를 검색합니다.</p>
<p>□ t_ui32 ecNet_GetCfgSlaveCount([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 슬레이브의 개수를 반환합니다.</p>
<p>□ t_cmdidx ecNet_SetAlState([in] t_i32 NetID, [in] t_ui8 AlState, [out] t_i32 *ErrCode) 장치의 AlState 를 설정 합니다.</p>
<p>□ EEcAlState ecNet_GetAlState([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치의 AlState 를 반환 합니다.</p>
<p>□ t_success ecNet_SlvComErrSum_ClearAll ([in] t_i32 NetID, [out] t_i32 *ErrCode) 슬레이브의 에러 카운트 레지스터를 초기화 합니다.</p>

4.2 함수 설명

NAME	INFORMATION
<p>ecNet_GetVerInfo</p> <p>-</p>	<p> Net Interface Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>
SYNOPSIS	
<pre> ❑ t_success ecNet_GetVerInfo ([in] t_i32 NetID, [out] TEcFileVerInfo_SDK *pVerInfo_SDK, [out] TEcFileVerInfo_WDM *pVerInfo_WDM, [out] TEcFileVerInfo_Fw *pVerInfo_FW, [out] t_i32 *ErrCode) </pre>	

DESCRIPTION

ComiECAT 소프트웨어 패키지(라이브러리, 드라이버, 펌웨어)들의 버전을 읽는 함수입니다.

PARAMETER

- ▶ NetID : 사용자가 사용할 Network 번호
- ▶ pVerInfo_SDK: SDK 라이브러리의 버전 정보를 반환받을 버퍼 포인터
- ▶ pVerInfo_WDM: WDM 드라이버 프로그램의 버전 정보를 반환받을 버퍼 포인터
- ▶ pVerInfo_FW: 펌웨어 프로그램의 버전 정보를 반환받을 버퍼 포인터
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE


함수 수행의 성공 여부를 반환합니다.


Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.


NAME


ecNet_ScanSlave
- 슬레이브 검색

INFORMATION

 Net Interface Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_ui32 ecNet_ScanSlave ([in] t_i32 NetID, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 사용자가 사용 할 네트워크에 연결되어 있는 모든 슬레이브를 검색하여 그 개수를 반환 합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 현재 네트워크에 연결되어 있는 슬레이브의 개수를 반환합니다.


NAME

ecNet_SetCfgSlaveCount


ecNet_GetCfgSlaveCount

- 마스터가 관리하는 슬레이브 수 설정 / 반환

INFORMATION
 Net Interface Function

 VC++/BCB/.NET

 Level 3

 위험 요소 없음
SYNOPSIS

□ t_success ecNet_SetCfgSlaveCount([in] t_i32 NetID, [in] t_i32 SlaveCount, [out] t_i32 *ErrCode)

□ t_ui32 ecNet_GetCfgSlaveCount ([in] t_i32 NetID, [out] t_i32 *ErrCode)

DESCRIPTION

ecNet_SetCfgSlaveCount() 함수는 환경설정에 의해서 마스터에 등록된 슬레이브의 수를 결정합니다. 이 함수는 네트워크 환경설정 프로그램에서 사용하기 위한 함수이며 일반 사용자들이 사용하면 안되는 함수입니다.

ecNet_GetCfgSlaveCount() 함수는 현재 환경설정에 의해서 마스터에 등록된 슬레이브의 수를 반환합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlaveCount : 환경설정에 의해서 마스터에 등록된 슬레이브의 수
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecNet_SetCfgSlaveCount 함수의 반환 값: 함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

* ecNet_GetCfgSlaveCount : 환경설정에 의해서 마스터에 등록된 슬레이브의 수를 반환합니다.

REFERENCE

□ ecNet_ScanSlave 함수는 환경설정에서 정의된 슬레이브들 정보와는 상관없이 현재 네트워크에 연결되어진 슬레이브의 총 수를 반환하는데 반해서, ecNet_GetCfgSlaveCount 함수는 현재 연결되어진 슬레이브의 수와 상관없이 네트워크환경설정 프로그램에 의해서 등록된 슬레이브의 수를 반환합니다.

<h2 style="margin: 0;">NAME</h2> <p style="margin: 5px 0 0 20px;">ecNet_SetAlState</p> <p style="margin: 5px 0 0 20px;">ecNet_GetAlState</p> <p style="margin: 5px 0 0 20px;">- 마스터 AL-STATE 설정 / 반환</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: left; padding: 2px;">INFORMATION</th> </tr> <tr> <td style="padding: 2px;"> Net Interface Function</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> VC++/BCB/.NET</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> Level 1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> 위험 요소 없음</td> <td style="padding: 2px;"></td> </tr> </table>	INFORMATION		Net Interface Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Net Interface Function											
VC++/BCB/.NET											
Level 1											
위험 요소 없음											

SYNOPSIS

- t_cmdidx ecNet_SetAlState([in] t_i32 NetID, [in] t_ui8 AlState, [out] t_i32 *ErrCode)
- EEcAlState ecNet_GetAlState([in] t_i32 NetID, [out] t_i32 *ErrCode)

DESCRIPTION

마스터의 AL-STATE 를 설정합니다. AL-STATE 는 EtherCAT 통신에서 EtherCAT Machine State 를 의미합니다. 이 Machine State 에는 아래 매개변수 표와 같은 상태들이 있는데 각 상태는 의미하는 바가 다르고 통신 상태도 각각 다릅니다. (아래 표 참조)

마스터의 AL-STATE 가 바뀌면 마스터는 연결된 모든 슬레이브의 AL-STATE 도 마스터의 AL-STATE 와 같은 값으로 바꿔줍니다.

ecNet_SetAlState 함수는 마스터 보드의 AlState 를 설정 합니다.

ecNet_GetAlState() 함수는 마스터 보드의 AlState 를 반환 합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ AlState : ecNet_SetAlState(), ecNet_SetAlState_FF() 함수의 인자이며, AlState 값입니다.

Value	Meaning
0 (ecAL_STATE_DISCON)	모든 슬레이브와 연결 상태를 Disconnect 상태로 만든다.
1 (ecAL_STATE_INITIAL)	Initial State. <ul style="list-style-type: none"> - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 불가능 - PDO 읽기/쓰기 불가능
2 (ecAL_STATE_PREOP)	PreOP State. <ul style="list-style-type: none"> - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 가능 - PDO 읽기/쓰기 불가능

4 (ecAL_STATE_SAFEOP)	SafeOP State. <ul style="list-style-type: none"> - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 가능 - Input PDO 읽기 가능 - Output PDO 쓰기 불가능
8 (ecAL_STATE_OP)	OP State. 모든 종류의 데이터 읽기/쓰기 가능

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecNet_SetAlState() 함수의 반환 값: CmdIdx 를 반환 합니다. CmdIdx 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.

* ecNet_GetAlState() : 리턴값은 마스터 보드의 AlState 값을 반환 합니다.

EXAMPLE

* ecNet_SetFastFuncType 참조


NAME


ecNet_SlvComErrSum_ClearAll


- 통신 에러카운트 초기화

INFORMATION

 Net Interface Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_success ecNet_SlvComErrSum_ClearAll([in] t_i32 NetID, [out] t_i32 *ErrCode)
```

DESCRIPTION

모든 슬레이브의 통신 에러카운트의 누적값을 0으로 클리어합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

Slave I/f functions

㈜커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 (주)커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.







5 Slave I/F Functions

5.1 함수 요약

Summary of Functions	
<p>□ EECAlState ecSlv_GetAlState([in] t_i32 NetID, [in] t_i32 SlvPhysAddr, [out] t_i32 *ErrCode)</p> <p>물리적 주소에 맞는 Slave 의 AlState 를 반환</p>	
<p>□ EECAlState ecSlv_GetAlState_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] t_i32 *ErrCode)</p> <p>슬레이브 인덱스에 맞는 Slave 의 AlState 를 반환</p>	
<p>□ t_i32 ecSlv_ReadCoeSdo([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 Index, [in] t_i32 SubIdx, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [out] void* pBuf, [out] t_i32 *ErrCode)</p> <p>물리적 주소의 슬레이브 Sdo 값 확인</p>	
<p>□ t_i32 ecSlv_ReadCoeSdo_A([in] t_i32 NetID, [in] t_i32 SlaveIdx, [in] t_i32 Index, [in] t_i32 SubIdx, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [out] void* pBuf, [out] t_i32 *ErrCode)</p> <p>슬레이브 인덱스의 슬레이브 Sdo 값 확인</p>	
<p>□ t_cmdidx ecSlv_WriteCoeSdo([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 Index, [in] t_i32 SubIdx, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [out] void* pData, [out] t_i32 *ErrCode)</p> <p>물리적 주소의 슬레이브 Sdo 값 저장</p>	
<p>□ t_cmdidx ecSlv_WriteCoeSdo_A([in] t_i32 NetID, [in] t_i32 SlaveIdx, [in] t_i32 Index, [in] t_i32 SubIdx, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [in] void* pData, [out] t_i32 *ErrCode)</p> <p>슬레이브 인덱스의 슬레이브 Sdo 값 확인</p>	
<p>□ t_ui16 ecSlv_GetComErrSum([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [out] t_i32 *ErrCode)</p> <p>슬레이브의(주소값 지정) 누적 통신 에러 카운트 값을 반환</p>	
<p>□ t_i32 ecSlv_GetComErrSum_A([in] t_i32 NetID, [in] t_ui16 SlaveIndex, [out] t_i32 *ErrCode)</p> <p>슬레이브의(순서 지정 방식) 누적 통신 에러 카운트 값을 반환</p>	
<p>□ t_success ecSlv_ClearComErrSum([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [out] t_i32 *ErrCode)</p> <p>슬레이브의(주소값 지정 방식) 누적 통신 에러 카운트 값을 0 으로 리셋</p>	
<p>□ t_success ecSlv_ClearComErrSum_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] t_i32 *ErrCode)</p> <p>슬레이브의(순서 지정 방식) 누적 통신 에러 카운트 값을 0 으로 리셋</p>	

5.2 함수 설명

NAME	INFORMATION
ecSlv_GetAlState	 Slave I/F Function
ecSlv_GetAlState_A	 VC++/BCB/.NET
- 슬레이브 AlState 확인	 Level 1
(슬레이브 EtherCAT Machine State 확인)	 위험 요소 없음

SYNOPSIS

```

❑ t_ui8 ecSlv_GetAlState([in] t_i32 NetID, [in] t_i32 SlavePhysAddr, [out] t_i32 *ErrCode)

❑ t_ui8 ecSlv_GetAlState_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] t_i32 *ErrCode)

```

DESCRIPTION

이 두 함수는 지정한 슬레이브의 현재 AL-STATE 값을 반환합니다.

ecSlv_GetAlState 함수는 슬레이브를 지정할 때 주소값을 사용하여 슬레이브를 지정합니다.

ecSlv_GetAlState_A 함수는 슬레이브를 지정할 때 네트워크 상에서 슬레이브의 연결 순서를 사용하여 슬레이브를 지정합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ SlaveIndex : 슬레이브의 연결 순서
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 슬레이브의 현재 AL-STATE 값을 반환 합니다. 반환되는 AL-STATE 값의 의미는 아래 표와 같습니다.

Value	Meaning
0 (ecAL_STATE_DISCON)	모든 슬레이브와 연결 상태를 Disconnect 상태로 만든다.

<p style="text-align: center;">1 (ecAL_STATE_INITIAL)</p>	<p>Initial State.</p> <ul style="list-style-type: none"> - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 불가능 - PDO 읽기/쓰기 불가능
<p style="text-align: center;">2 (ecAL_STATE_PREOP)</p>	<p>PreOP State.</p> <ul style="list-style-type: none"> - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 가능 - PDO 읽기/쓰기 불가능
<p style="text-align: center;">4 (ecAL_STATE_SAFEOP)</p>	<p>SafeOP State.</p> <ul style="list-style-type: none"> - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 가능 - Input PDO 읽기 가능 - Output PDO 쓰기 불가능
<p style="text-align: center;">8 (ecAL_STATE_OP)</p>	<p>OP State. 모든 종류의 데이터 읽기/쓰기 가능</p>

<h1>NAME</h1> <p>ecSlv_ReadCoeSdo</p> <p>ecSlv_ReadCoeSdo_A</p> <p>- SDO 데이터 읽기</p>	INFORMATION
	Slave I/F Function
	VC++/BCB/.NET
	Level 1
	위험 요소 없음

SYNOPSIS

- t_i32 ecSlv_ReadCoeSdo ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 Index, [in] t_i32 SubIndex, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [out] void* pBuf, [out] t_i32 *ErrCode)
- t_i32 ecSlv_ReadCoeSdo_A ([in] t_i32 NetID, [in] t_ui16 SlvIdx, [in] t_i32 Index, [in] t_i32 SubIndex, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [out] void* pBuf, [out] t_i32 *ErrCode)

DESCRIPTION

지정한 슬레이브에서 지정된 SDO(Service Data Object) 오브젝트를 읽어서 반환하는 함수입니다. SDO(Service Data Object)는 비주기적으로 전달되는 데이터 오브젝트(Object)를 말합니다.

CoE(CANOpen Over EtherCAT)를 지원하는 이더넷 슬레이브 장치는 해당 슬레이브가 지원하는 기능(Function)이나 데이터(Data)를 오브젝트(Object)라는 것으로 정의하여 제공합니다. 오브젝트는 인덱스라고 하는 숫자로 서로를 구분하며, 해당 오브젝트의 데이터 형식이나 기타 속성에 대해서는 오브젝트 사전(Object Dictionary)이라고 하는 것을 통해서 정의해줍니다. 오브젝트 사전은 해당 슬레이브에서 제공하는 오브젝트들을 정리하여 표나 기타 형태로 만들어 놓은 것을 말합니다.

Index	Object Code (OC)	Name	Data Type	Category
6000h	Array	Read Input 8-bit	Unsigned8	C: DI
6001h	-	Reserved	-	-
6002h	Array	Polarity Input 8-bit	Unsigned8	O
6003h	Array	Filter Constant Input 8-bit	Unsigned8	O
6004h	-	Reserved	-	-
6005h	Var	Global Interrupt Enable Digital	Boolean	O
6006h	Array	Interrupt Mask Any Change 8-bit	Unsigned8	O
6007h	Array	Interrupt Mask Low-to-High 8-bit	Unsigned8	O
6008h	Array	Interrupt Mask High-to-Low 8-bit	Unsigned8	O
6009h	-	Reserved	-	-
to				
601Eh	-	Reserved	-	-

[표 #] Object Dictionary 의 예

하나의 오브젝트 인덱스는 실제로는 인덱스(Index)와 서브인덱스(Sub-Index)로 구성됩니다. 오브젝트가 하나의 값을 가지는 객체라면 서브인덱스값은 0을 사용합니다. 하지만 여러

개의 배열로 이루어졌거나 구조체와 같은 형식이라면 여러 개의 서브인덱스로 구성될 수 있습니다.

Object Description

INDEX	6000h
Name	Read Input 8 Bit
Object Code	Array
Data Type	Unsigned8
Category	Conditional: Device with digital inputs

Entry Description

Sub-Index	0h
Description	Number of Input 8 Bit
Access	ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	1h to FEh
Default Value	No

Sub-Index	1h
Description	Read Input 1h to 8h
Access	ro
Entry Category	Mandatory
PDO Mapping	Default
Value Range	Unsigned8
Default Value	No

Sub-Index	2h .. 8h
Description	Read Input 9h to 10h .. Read Input 39h to 40h
Access	ro
Entry Category	Optional
PDO Mapping	Default
Value Range	Unsigned8
Default Value	No

[표 #] 오브젝트의 서브인덱스 구성 예

사용자는 오브젝트를 통해서 CoE 를 지원하는 슬레이브의 데이터를 읽거나 쓸 수 있습니다. ecSlv_ReadCoeSdo 함수는 사용자가 CoE 오브젝트를 읽을 수 있도록 해주는 함수입니다.

참고로, SDO(Service Data Object)는 비주기적으로 전달되는 데이터 오브젝트를 말하며, 주기적으로 전달되는 데이터 오브젝트는 PDO(Periodic Data Object)라고 합니다.

ecSlv_ReadCoeSdo 함수는 슬레이브를 지정할 때 주소값을 사용하여 슬레이브를 지정합니다.

ecSlv_ReadCoeSdo_A 함수는 슬레이브를 지정할 때 네트워크 상에서 슬레이브의 연결 순서를 사용하여 슬레이브를 지정합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ SlaveIndex : 슬레이브의 연결 순서
- ▶ Index : 오브젝트 인덱스(Object Index)
- ▶ SubIndex : 오브젝트의 서브인덱스(Sub-index)
- ▶ IsComptAccess: 오브젝트가 배열이나 구조체로 이루어진 경우에 해당 오브젝트의 모든 서브인덱스에 해당하는 값을 읽기할 것인지 아니면 서브인덱스로 지정된 데이터만 읽기할 것인지를 결정하는 플래그입니다. 일반적으로 이 값은 0(false)으로 설정해야 합니다.


Value	Meaning
0	지정한 서브인덱스에 대해서만 값을 읽습니다.
1	모든 서브인덱스에 대한 데이터를 한꺼번에 읽습니다. 이 경우에는 SubIndex 값을 0으로 전달해야 하며, 버퍼는 오브젝트의 전체 데이터 구조와 일치하는 버퍼가 제공되어야 합니다

- ▶ DataSize : 읽을 DataSize. 이 값은 오브젝트 사전에서 정의된 해당 오브젝트의 데이터 크기와 일치해야 합니다. 그렇지 않으면 읽기에 실패할 수 있습니다.
- ▶ pBuffer: 읽은 데이터를 전달 받을 버퍼.
- ▶ ErrorCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 반환 값이 양수이면 읽은 데이터의 크기를 반환하는 것입니다. 반환 값이 음수이면 에러코드를 반환하는 것입니다.

REFERENCE

	<p>이 함수는 해당 슬레이브가 PreOP 이상의 AL-STATE 에 있을 때에만 사용 가능합니다. 슬레이브가 INITIAL 상태에 있을 때 이 함수를 사용하면 읽기 실패합니다</p>
---	--

EXAMPLE

< 예제 설명 >

슬레이브 주소가 0x1 인 슬레이브로부터 CiA DS-402 프로파일에서 정의한 ‘Max motor speed’ 오브젝트(아래 그림 참조)값의 읽기 예시.

Index	Sub-Index	Name / Description	Units	Range	Data Type	Access	PDO	Opmode	EEPROM
6080h	00h	Max motor speed • Set the maximum velocity of motor. • The maximum value is limited by the maximum speed read out from the motor in internal processing. • Tq and cst and restricts speed with the preset value of this object.	r/min	0 - 4294967295	U32	rw	RxPDO	ALL	Yes

[그림 #] CiA DS-402 프로파일의 ‘Max motor speed’ 오브젝트의 정의

< 예제 코드 >

```
#define NETID          0
#define SLAVE_ADDR    0x1
t_ui32 errCode;
t_ui32 maxMotorSpeed;
ecSly_ReadCoeSdo (NETID, SLAVE_ADDR, 0x6080, 0, false, 4, &maxMotorSpeed, &errCode);
printf("maxMotorSpeed 값은 %d 입니다!", maxMotorSpeed);
```

NAME	INFORMATION
ecSlv_WriteCoeSdo	Slave I/F Function
ecSlv_WriteCoeSdo_A	VC++/BCB/.NET
- SDO 데이터 쓰기	Level 1
	위험 요소 없음

SYNOPSIS

□ t_i32 ecSlv_WriteCoeSdo ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 Index, [in] t_i32 SubIndex, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [in] void* pBuf, [out] t_i32 *ErrCode)

□ t_i32 ecSlv_WriteCoeSdo_A ([in] t_i32 NetID, [in] t_ui16 SlvIdx, [in] t_i32 Index, [in] t_i32 SubIndex, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [in] void* pBuf, [out] t_i32 *ErrCode)

DESCRIPTION

지정한 슬레이브에서 지정된 SDO(Service Data Object) 오브젝트를 읽어서 반환하는 함수입니다. SDO(Service Data Object)는 비주기적으로 전달되는 데이터 오브젝트(Object)를 말합니다. 오브젝트에 대한 설명은 ecSlv_ReadCoeSdo 함수 설명편을 참고하시기 바랍니다.

ecSlv_WriteCoeSdo 함수는 슬레이브를 지정할 때 주소값을 사용하여 슬레이브를 지정합니다.

ecSlv_WriteCoeSdo_A 함수는 슬레이브를 지정할 때 네트워크 상에서 슬레이브의 연결 순서를 사용하여 슬레이브를 지정합니다.

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ SlaveIndex : 슬레이브의 연결 순서
- ▶ Index : 오브젝트 인덱스(Object Index)
- ▶ SubIndex : 오브젝트의 서브인덱스(Sub-index)
- ▶ IsComptAccess: 오브젝트가 배열이나 구조체로 이루어진 경우에 해당 오브젝트의 모든 서브인덱스에 해당하는 값을 쓰기할 것인지 아니면 서브인덱스로 지정된 데이터만 쓰기할 것인지를 결정하는 플래그입니다. 일반적으로 이 값은 0(false)으로 설정해야 합니다.

Value	Meaning
0	지정한 서브인덱스에 대해서만 값을 씁니다.
1	모든 서브인덱스에 대한 데이터를 한꺼번에 씁니다. 이 경우에는 SubIndex 값을 0으로 전달해야 하며, 버퍼는 오브젝트의 전체 데이터 구조와 일치하는 버퍼가 제공되어야 합니다

- ▶ DataSize : 쓰기할 DataSize. 이 값은 오브젝트 사전에서 정의된 해당 오브젝트의 데이터 크기와 일치해야 합니다. 그렇지 않으면 읽기에 실패할 수 있습니다.
- ▶ pBuf: 쓰기 하는 데이터를 전달하는 버퍼.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 반환 값이 양수이면 쓰기한 데이터의 크기를 반환하는 것입니다. 반환 값이 음수이면 에러코드를 반환하는 것입니다.

EXAMPLE

< 예제 설명 >

슬레이브 주소가 0x1 인 슬레이브로부터 CiA DS-402 프로파일에서 정의한 ‘Max motor speed’ 오브젝트(아래 그림 참조)값을 3000 으로 쓰기 하는 예.

Index	Sub-Index	Name / Description	Units	Range	Data Type	Access	PDO	Opmode	EEPROM
6080h	00h	Max motor speed <ul style="list-style-type: none"> • Set the maximum velocity of motor. • The maximum value is limited by the maximum speed read out from the motor in internal processing. • Tq and cst and restricts speed with the preset value of this object. 	r/min	0 - 4294967295	U32	rw	RxPDO	ALL	Yes

[그림 #] CiA DS-402 프로파일의 ‘Max motor speed’ 오브젝트의 정의

< 예제 코드 >

```
#define NETID          0
#define SLAVE_ADDR    0x1
t_i32 errCode;
t_ui32 maxMotorSpeed = 3000;
ecSlv_WriteCoeSdo (NETID, SLAVE_ADDR, 0x6080, 0, false, 4, &maxMotorSpeed, &errCode);
```


NAME


ecSlv_GetComErrSum


ecSlv_GetComErrSum_A


- 슬레이브의 통신 에러 카운트 값 반환

INFORMATION

 Slave I/F Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

❑ t_ui16 ecSlv_GetComErrSum ([in] t_i32 NetID, [in] t_ui16 SlaveIndex, [out] t_i32 *ErrCode)

❑ t_i32 ecSlv_GetComErrSum_A ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [out] t_i32 *ErrCode)

DESCRIPTION

ecSlv_GetComErrSum(), ecSlv_GetComErrSum_A() 함수는

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlaveIndex : 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ SlvPhysAddr : 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecSlv_GetComErrSum():

* ecSlv_GetComErrSum_A():

EXAMPLE

NAME


ecSlv_ClearComErrSum

ecSlv_ClearComErrSum_A


- 슬레이브의 통신 에러 카운트 초기화

INFORMATION

 Slave I/F Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_ui16 ecSlv_ClearComErrSum ([in] t_i32 NetID, [in] t_ui16 SlaveIndex, [out] t_i32 *ErrCode)

□ t_i32 ecSlv_GetComErrSum_A ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [out] t_i32 *ErrCode)

DESCRIPTION

ecSlv_ClearComErrSum(), ecSlv_ClearComErrSum_A() 함수는

PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlaveIndex : 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ SlvPhysAddr : 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecSlv_ClearComErrSum():

* ecSlv_ClearComErrSum_A():

EXAMPLE

Digital I/O functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 (쥬커미조아)를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.







6 Digital I/O Functions

6.1 함수 요약

Summary of Functions
<p>❑ <code>t_i32 eccdiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)</code> 장치에 연결되어 있는 Digital Input 채널 개수를 반환합니다.</p>
<p>❑ <code>t_i32 eccdoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)</code> 장치에 연결되어 있는 Digital Output 채널 개수를 반환합니다.</p>
<p>❑ <code>t_bool eccdiGetOne([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode)</code> 대상 디지털 입력(Digital Input) 채널에 대해 입력 상태를 확인합니다.</p>
<p>❑ <code>t_dword eccdiGetMulti([in] t_i32,NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)</code> 다중 디지털 입력(Digital Input) 채널에 대해 입력 상태를 확인합니다.</p>
<p>❑ <code>t_bool eccdoGetOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode)</code> 대상 디지털 출력(Digital Output)채널에 대해 출력 상태를 확인합니다.</p>
<p>❑ <code>t_dword eccdoGetMulti([in] t_i32,NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)</code> 다중 디지털 출력(Digital Output)채널에 대해 출력 상태를 확인합니다.</p>
<p>❑ <code>t_success eccdoPutOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [in] t_bool OutState, [out] t_i32 *ErrCode)</code> 대상 디지털 출력(Digital Output)채널에 대해 출력을 발생합니다.</p>
<p>❑ <code>t_success eccdoPutMulti([in] t_i32,NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [in] t_dword dwOutStates, [out] t_i32 *ErrCode)</code> 다중 디지털 출력(Digital Output)채널에 대해 출력을 발생합니다.</p>
<p>❑ <code>t_bool eccdiGetOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode)</code> 물리적 주소와 로컬채널로 대상 디지털 입력(Digital Input)채널에 대한 입력 상태를 확인합니다.</p>
<p>❑ <code>t_bool eccdoGetOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode)</code> 물리적 주소와 로컬채널로 대상 디지털 출력(Digital Output)채널에 대한 출력 상태를 확인합니다.</p>
<p>❑ <code>t_bool eccdiGetMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 IniLocalChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)</code> 물리적 주소와 로컬채널로 다중 디지털 입력(Digital Input)채널에 대한 입력 상태를 확인합니다.</p>
<p>❑ <code>t_bool eccdoGetMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 IniLocalChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)</code> 물리적 주소와 로컬채널로 다중 디지털 출력(Digital Output)채널에 대한 출력 상태를 확인합니다.</p>
<p>❑ <code>t_bool eccdoPutOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [in] t_bool OutState, [out] t_i32 *ErrCode)</code> 슬레이브 인덱스와 로컬채널로 대상 디지털 출력(Digital Output)채널에 대한 출력을 발생합니다.</p>

❑ t_bool ecdPutMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] IniLocalChannel, [in] t_ui8 NumChannels, [in] t_dword dwOutStates, [out] t_i32 *ErrCode)
슬레이브 인덱스와 로컬 채널로 다중 디지털 출력(Digital Output) 채널에 대한 출력을 발생합니다.

6.2 함수 설명

NAME	INFORMATION
ecdiGetNumChannels	 Digital I/O Function
ecdoGetNumChannels	 VC++/BCB/.NET
- 디지털 입·출력 채널 갯수 반환	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_i32 ecdiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)
- t_i32 ecdoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)

DESCRIPTION

ecdiGetNumChannels 함수는 지정한 네트워크에 연결되어 있는 전체 디지털 입력 채널의 개수를 반환합니다.

ecdoGetNumChannels 함수는 지정한 네트워크에 연결되어 있는 전체 디지털 출력 채널의 개수를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecdiGetNumChannels 함수의 반환 값: 지정한 네트워크에 연결되어 있는 디지털 입력 채널의 개수를 반환 합니다.

* ecdoGetNumChannels 함수의 반환 값: 지정한 네트워크에 연결되어 있는 디지털 출력 채널의 개수를 반환 합니다.

EXAMPLE

```
C/C++
```

```
#include "ComiEcatSdk_Api.h"
```

```
t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수
```

```
/**/
```

```
9장에서 각각의 슬레이브 당 채널을 얻어 오는 것과는 다른 개념이므로 조심
```





```
*****/
```

```
void DI_GetChCount(void)
{
    t_i32 All_DiCount = 0;           //마스터에 연결 된 모든 DI 채널 개수 저장 할 변수
    t_i32 All_DoCount = 0;         //마스터에 연결 된 모든 DO 채널 개수 저장 할 변수
    All_DiCount = ecdiGetNumChannels(Device_Info.NetIdx, &Error_Num);
    // 마스터에 연결 된 모든 DI 채널 개수를 로드
    All_DoCount = ecdoGetNumChannels(Device_Info.NetIdx, &Error_Num);
    // 마스터에 연결 된 모든 DO 채널 개수를 로드
}
```

6.2.1 전역채널(Global Channel) 번호를 이용하는 함수군

전역채널(Global Channel) 번호는 네트워크에 연결된 모든 디지털입력(또는 출력) 접점을 종합하여 채널 번호를 매겨서 사용하는 것을 말합니다. 예를 들어서 네트워크에 32 채널짜리 디지털입력 모듈이 2 개 연결되어 있다면 전역채널 번호는 0~63의 번호로 구성됩니다. 따라서 두 번째 모듈의 첫 번째 채널은 전역채널 번호로는 32번 채널이 되는 것입니다.

이때 전역채널 번호의 순서는 모듈의 연결순서에 의해서 결정됩니다. 또한 디지털입력 채널과 디지털출력 채널은 서로 독립적으로 전역채널 번호가 관리되므로 서로 영향을 주지 않습니다. 예를 들어서 32 채널 디지털입력 모듈이 2 개 연결되고, 32 채널 디지털출력 모듈이 3 개 연결되었다고 가정하면, 디지털입력의 전역채널은 0~63 이 되고, 디지털출력의 전역채널은 0~95 가 됩니다.

<h2>NAME</h2> <p>ecdiGetOne</p> <p>- 단일 채널에 대한 입력 상태 반환</p>	INFORMATION
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```
□ t_bool ecdiGetOne([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode)
```

DESCRIPTION

이 함수는 단일 채널에 대한 디지털 입력 상태를 반환 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ DiChannel : 디지털 입력 채널 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 채널의 디지털 입력 상태 값을 반환 합니다.

EXAMPLE

```
C/C++

#include "ComiEcatSdk_Api.h"

t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수
t_i32 nNetID = 0;
void DI_GetOne(void)
{
    t_i16 DI_NumCh =0;        // DI 채널 개수 저장 변수

    t_bool State = false;    // DI 결과 값 저장 변수





    DI_NumCh = ecdiGetNumChannels(nNetID, Error_Num);
    For(int i = 0 ; i< DI_NumCh ; i++)
    {
```

```
State = ecdiGetOne(nNetID, i, &Error_Num);

// DI_NumCh (슬레이브의 채널 개수) 만큼 DI 실행

if(State == 1)
{
    //DI 결과 값이 1 일때 처리
}
else
{
    //DI 결과 값이 0 일때 처리
}
}

} // DI_GetOne 함수 종료
```

NAME ecdiGetMulti - 다중 채널에 대한 입력 상태 반환	INFORMATION
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```

  □ t_dword ecdiGetMulti([in] t_i32 NetID, [in] t_ui32 IniChannel, [in] t_ui8
    NumChannels, [out] t_i32 *ErrCode)
  
```

DESCRIPTION

이 함수는 다중 채널에 대한 디지털 입력 상태를 반환 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ IniChannel : 시작 채널 번호
- ▶ NumChannels : 시작 채널로부터 몇 개의 채널의 상태를 확인할 것인지에 대한 값
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 다중 채널의 디지털 입·출력 상태 값을 반환 합니다.

EXAMPLE

C/C++

```

#include "ComiEcatSdk_Api.h"
#define NUMCHNNELS 2 //사용자가 사용 할 채널 수를 입력
t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수
t_i32 nNetID = 0;
void DI_GetMulit(void)
{
    t_i16 DI_InitCh = 0; //시작 채널 번호 저장 변수
    t_i16 DI_NumCh =0; // DI 채널 개수 저장 변수
  
```

```
t_dword State = 0;          // DI_Multi 결과 값 저장 변수

State = ecdiGetMulti (nNetID, DI_InitCh, NUMCHNNELS, &Error_Num);

//*****
// State 에 결과 값에는 DI 결과 값이 십진수 형태로 저장
//ex)0 번 채널부터 2 번까지 모든 채널이 ON 이라면 결과값은 111 즉 10 진수 7
//*****//

} // DI_GetMulit 함수 종료
```

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecdoPutOne / ecdoGetOne</p> <p style="margin: 0;">- 단일 채널에 대한 출력 발생 및 출력 상태 확인</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left; padding: 2px;">INFORMATION</th> </tr> <tr> <td style="padding: 2px;"> Digital I/O Function </td> </tr> <tr> <td style="padding: 2px;"> VC++/BCB/.NET </td> </tr> <tr> <td style="padding: 2px;"> Level 1 </td> </tr> <tr> <td style="padding: 2px;"> 위험 요소 없음 </td> </tr> </table>	INFORMATION	Digital I/O Function	VC++/BCB/.NET	Level 1	위험 요소 없음
INFORMATION						
Digital I/O Function						
VC++/BCB/.NET						
Level 1						
위험 요소 없음						

SYNOPSIS

- t_success ecdoPutOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [in] t_bool OutState, [out] t_i32 *ErrCode)
- t_bool ecdoGetOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode)

DESCRIPTION

ecdoPutOne 함수는 단일 채널에 대한 디지털 출력 상태를 발생시키며, ecdoGetOne 함수는 단일 채널에 대한 디지털 출력 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ DoChannel : 디지털 출력 채널 번호
- ▶ OutState : ecdoPutOne() 함수의 인자이며, 단일 채널에 대한 디지털 출력상태를 발생합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecdoPutOne 함수의 반환 값: 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	출력발생에 실패 했음을 의미합니다.
1	출력발생에 성공 했음을 의미합니다.

* ecdoGetOne 함수의 반환 값: 이 리턴값은 출력 상태 값을 반환 합니다.

EXAMPLE

C/C++

```

#include "ComiEcatSdk_Api.h"
#define DO_CHNUMBER 0 //사용자가 출력을 원하는 DO 채널 번호
    
```

```

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

void DO_PutOne(void)
{
    TEcSlvTypeInfo pTypeInfoBuf // 구조체 관련 내용 9 장 참고

    t_i16 DO_InitCh = 0;          //시작 채널 번호 저장 변수
    t_i16 DO_NumCh = 0;          // DO 채널 개수 저장 변수

    t_bool State = 0;           // 출력 값

    DO_InitCh = pTypeInfoBuf . DO . StaGlobChannel;
    //구조체 할당 후 DI 시작 채널 얻어 옴
    DO_NumCh = pTypeInfoBuf . DO . NumChannels;
    //구조체 할당 후 DI 채널 개수 얻어옴
    if(!ecdoPutOne (Device_Info.NetIdx, DI_InitCh, DO_CHNUMBER, State, &Error_Num))
    {
        //DO 실패 시 예외 처리
    }
} // DO_PutOne 함수 종료

void DO_GetOne(void)
{
    TEcSlvTypeInfo pTypeInfoBuf // 구조체 관련 내용 9 장 참고





    t_i16 DO_InitCh = 0;          //시작 채널 번호 저장 변수
    t_i16 DO_NumCh = 0;          // DO 채널 개수 저장 변수

    DO_InitCh = pTypeInfoBuf . DO . StaGlobChannel;
    //구조체 할당 후 DI 시작 채널 얻어 옴
    DO_NumCh = pTypeInfoBuf . DO . NumChannels;
    //구조체 할당 후 DI 채널 개수 얻어옴

    for(int i = DO_InitCh; i < DO_NumCh ; i++)
    {
        if(ecdoGetOne (Device_Info.NetIdx, i, &Error_Num))
        {
            //DO 결과가 1 일 경우 처리
        }
        else
        {
            //DO 결과가 0 일 경우 처리
        }
    }

    //*****
    채널의 DO 상태를 확인 하는 예제
    슬레이브에서 할당 받은 초기 채널과 채널 수를 이용하여
    슬레이브의 모든 채널 Do 상태를 확인
    *****//
} // DO_GetOne 함수 종료

```

<h2>NAME</h2> <p>ecdoPutMulti / ecdoGetMulti</p> <p>- 다중 채널에 대한 출력 발생 및 출력 상태 확인</p>	INFORMATION
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_success ecdoPutMulti([in] t_i32 NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [in] t_dword dwOutStates, [out] t_i32 *ErrCode)
- t_dword ecdoGetMulti([in] t_i32 NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)

DESCRIPTION

ecdoPutMulti 함수는 다중 채널에 대한 디지털 출력 상태를 발생시키며, ecdoGetMulti 함수는 다중 채널에 대한 디지털 출력 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ IniChannel : 시작 채널 번호
- ▶ NumChannels : 시작 채널로부터 몇 개의 채널의 상태를 확인할 것인지에 대한 값
- ▶ dwOutStates : ecdoPutMulti() 함수의 인자이며, 다중 채널에 대한 디지털 출력상태를 발생립니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecdoPutMulti : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	출력발생에 실패 했음을 의미합니다.
1	출력발생에 성공 했음을 의미합니다.

* ecdoGetMulti : 이 리턴값은 출력 상태 값을 반환 합니다.

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"

#define NUMCHNNELS 2 //사용자가 확인 할 채널 수를 입력

t_32 Error_Num = 0; //함수 별 에러 코드 저장 변수

void DO_PutMulti(void)
{
    TEcSlvTypeInfo pTypeInfoBuf // 구조체 관련 내용 9 장 참고

    t_i16 DO_InitCh = 0; //시작 채널 번호 저장 변수

    t_dword dwOutStates = 0;
    //여러 수의 출력 값을 2 진수로 표현
    // ex) 0 번 3 번 ON -> 1001
    DO_InitCh = pTypeInfoBuf . DO. StaGlobChannel;
    //구조체 할당 후 DI 시작 채널 얻어 옴
    if(!ecdoPutMulti (Device_Info.NetIdx, DO_InitCh, NUMCHNNELS, dwOutStates,
&Error_Num))
    {
        //DO 실패 시 예외 처리
    }
} // DO_ PutMulti 함수 종료

void DO_GetMulti(void)
{
    TEcSlvTypeInfo pTypeInfoBuf // 구조체 관련 내용 9 장 참고

    t_i16 DO_InitCh = 0; //시작 채널 번호 저장 변수

    t_dword dwOutStates = 0;
    DO_InitCh = pTypeInfoBuf . DO. StaGlobChannel;
    //구조체 할당 후 DI 시작 채널 얻어 옴

    dwOutStates = ecdoGetMulti (Device_Info.NetIdx DO_InitCh, NUMCHNNELS &Error_Num);


} // DO_ GetMulti 함수 종료

```

6.2.2 지역채널(Local Channel) 번호를 이용하는 함수군

지역 채널(Global Channel) 번호는 각각의 슬레이브에 독립적으로 채널 번호를 부여하는 방식을 말합니다. 따라서 모든 슬레이브의 첫 번째 채널의 지역 채널 번호는 0 번이 됩니다.

지역 채널을 번호를 사용하는 함수에는 지역 채널 번호와 함께 슬레이브 모듈의 주소값이 매개변수로 함께 전달됩니다.


 <p>안내</p>	<p>디지털입력의 지역채널 번호는 해당 슬레이브의 Tx PDO(Input PDO) 영역의 비트순서에 해당합니다. 만일 해당 슬레이브의 Tx PDO 영역에 디지털입력 데이터외에 다른 정보가 함께 전달된다면 물리적인 채널 번호와 ComiECAT SDK에서 관리하는 지역채널 번호가 다를 수 있습니다. 이러한 경우에는 디지털입력 채널 데이터가 Tx PDO 영역의 몇 번째 비트를 통해서 전달되는지 알아야 하며, ComiECAT SDK에서의 지역채널 번호는 이 비트번호를 지정해주시기 바랍니다.</p> <p>디지털출력의 경우에는 Rx PDO(Output PDO) 영역의 비트순서에 해당하며, 디지털입력과 동일한 논리가 적용됩니다.</p>
---	--


NAME


ecdiGetOne_L


- 대상 채널의 입·출력상태 확인

INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_bool ecdiGetOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui32
LocalChannel, [out] t_i32 *ErrCode)
```

DESCRIPTION





ecdiGetOne_L() 함수는 지역 채널 방식으로 지정된 디지털입력 채널의 입력 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr: 슬레이브 주소
- ▶ LocalChannel : 해당 디지털 입력 채널의 지역 채널 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 채널의 신호 입력 상태를 반환합니다.

<h2>NAME</h2> <p>ecdiGetMulti_L</p> <p>- 다중 채널의 입·출력상태 확인</p>	INFORMATION
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```

□ t_dword ecdiGetMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in]
IniLocalChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)

```

DESCRIPTION





ecdiGetMulti_L() 함수는 지역 채널 방식으로 지정된 다중 채널의 디지털입력 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr : 슬레이브 주소
- ▶ IniLocalChannel : 입력 상태를 확인할 시작 채널 번호
- ▶ NumChannels : 입력 상태를 확인할 채널의 수
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 다중 채널의 입·출력 상태를 반환합니다. 반환되는 값의 첫 번째 비트(bit 0) 값이 IniLocalChannel 에 해당하는 채널의 신호상태값을 나타나게 되며, 비트 순서대로 그 다음 채널의 상태를 나타냅니다.

NAME	INFORMATION
ecdoPutOne_L	 Digital I/O Function
ecdoGetOne_L	 VC++/BCB/.NET
- 단일 채널의 DO 출력 발생 및 출력상태 확인	 Level 1
	 위험 요소 없음

SYNOPSIS

❑ t_bool ecdoPutOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] LocalChannel, [out] t_i32 *ErrCode)

❑ t_bool ecdoGetOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode)

DESCRIPTION

ecdoPutOne_L 함수는 단일 채널에 대한 디지털 출력 상태를 발생시키며, ecdoGetOne_L 함수는 단일 채널에 대한 디지털 출력 상태를 반환합니다. 이때 지정되는 채널번호는 지역채널(Local Channel) 번호로 지정됩니다.





PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr : 슬레이브 주소
- ▶ LocalChannel : 디지털출력의 지역 채널(Local Channel) 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	함수 수행이 실패 했음을 의미 합니다.
1	함수 수행이 성공 했음을 의미 합니다.

NAME	I N F O R M A T I O N
ecdoPutMulti_L	 Digital I/O Function
ecdoGetMulti_L	 VC++/BCB/.NET
- 다중 채널의 DO 출력	 Level 1
	 위험 요소 없음

SYNOPSIS

- ❑ t_bool ecdoPutMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] IniLocalChannel, [in] t_ui8 NUmChannels, [out] t_i32 *ErrCode)
- ❑ t_dword ecdoGetMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] IniLocalChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode)

DESCRIPTION

ecdoPutMulti_L 함수는 다중 채널에 대한 디지털 출력 상태를 출력시키며, ecdoGetMulti_L 함수는 다중 채널에 대한 디지털 출력 상태를 읽어서 반환합니다. 이때 지정되는 채널번호는 지역채널(Local Channel) 번호 방식으로 지정됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr : 슬레이브 주소
- ▶ IniLocalChannel : 출력 시작 채널의 지역 채널(Local Channel) 번호
- ▶ NumChannels : 출력 채널의 수
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	함수 수행이 실패 했음을 의미 합니다.
1	함수 수행이 성공 했음을 의미 합니다.

Analog I/O functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.







7 Analog I/O Functions

7.1 함수 요약

Summary of Functions
<p>❑ t_i32 ecaiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 아날로그 입력 채널 개수를 반환합니다.</p>
<p>❑ t_i32 ecoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 아날로그 출력 채널 개수를 반환합니다.</p>
<p>❑ t_i32 ecaiGetChanVal_I([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode) 대상 아날로그 입력 채널에 대하여 AD 결과를 Digit 값으로 반환합니다.</p>
<p>❑ t_f64 ecaiGetChanVal_F([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode) 대상 아날로그 입력 채널에 대하여 AD 결과를 Volt 값으로 반환합니다.</p>
<p>❑ t_success ecaoSetChanVal_I([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_i32 OutData, [out] t_i32 *ErrCode) 대상 아날로그 출력 채널을 통해 Digit 값을 출력합니다.</p>
<p>❑ t_success ecaoSetChanVal_F([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_f64 OutData, [out] t_i32 *ErrCode) 대상 아날로그 출력 채널을 통해 Volt 값을 출력합니다.</p>
<p>❑ t_i32 ecaoGetOutValue_I([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode) 대상 아날로그 출력 채널에 대하여 AD 결과를 Digit 값으로 반환합니다.</p>
<p>❑ t_f64 ecaoGetOutValue_F([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode) 대상 아날로그 출력 채널에 대하여 AD 결과를 Volt 값으로 반환합니다.</p>

7.2 함수 설명

NAME	INFORMATION
ecaiGetNumChannels	 Analog I/O Function
ecaoGetNumChannels	 VC++/BCB/.NET
- 아날로그 입·출력 채널 갯수 반환	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_i32 ecaiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)
- t_i32 ecaoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode)

DESCRIPTION

ecaiGetNumChannels(), ecaoGetNumChannels() 함수는 장치에 연결되어 있는 아날로그 입·출력 채널 갯수를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 아날로그 입·출력 채널 개수를 반환합니다.





EXAMPLE

```
#include "ComiEcatSdk_Api.h"

t_i32 Error_Num = 0; // 함수 별 에러 코드 저장 변수
t_i32 nNetID = 0;

void AI_ ecaiGetNumChannels(void)
{
    t_i32 All_AICount = 0;           // 마스터에 연결된 모든 AI 채널 개수 저장할 변수
    t_i32 All_AOCount = 0;         // 마스터에 연결된 모든 AO 채널 개수 저장할 변수
    All_AICount = ecaiGetNumChannels(nNetID, &Error_Num);
    // 마스터에 연결된 모든 AI 채널 개수를 로드
    All_AOCount = ecaoGetNumChannels(nNetID, &Error_Num);
    // 마스터에 연결된 모든 AO 채널 개수를 로드
}

```

NAME	I N F O R M A T I O N
ecaiGetChanVal_I - 대상 아날로그 입력 채널의 입력 Digit 값 반환	 Analog I/O Function  VC++/BCB/.NET  Level 1  위험 요소 없음

SYNOPSIS

```
□ t_i32 ecaiGetChanVal_I([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode)
```

DESCRIPTION





이 함수는 대상 아날로그 입력 채널에 대하여 A/D 변환을 수행하고, 그 값을 Digit 값으로 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 채널
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 아날로그 입력 Digit 값을 반환 합니다.

NAME ecaiGetChanVal_F - 대상 아날로그 입력 채널의 입력 Volt 값 반환	INFORMATION
	 Analog I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

□ t_f64 ecaiGetChanVal_F([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 대상 아날로그 입력 채널에 대하여 A/D 변환을 수행하고, 그 값을 Volt 값으로 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 채널
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 아날로그 입력 Volt 값을 반환 합니다.

REFERENCE

□ 이 함수가 정상적으로 동작하려면 슬레이브 장치의 ESI(XML) 파일 또는 “ComiECatDevInfo.xml” 파일에서 아날로그 입력 값의 스케일링 정보가 잘 정의되어 있어야만 적용 가능합니다. 그렇지 않은 경우에는 ecaiGetChanVal_FS 함수를 사용하시기 바랍니다.

EXAMPLE

* ecaiGetChanVal_I EXAMPLE 참고


NAME


ecaoSetChanVal_I


- 대상 아날로그 출력 채널을 통해 Digit 값 출력

INFORMATION

 Analog I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_success ecaoSetChanVal_I([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_i32 OutData, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 대상 아날로그 출력 채널에 대하여 지정한 Digit 값을 출력합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 출력 채널 번호
- ▶ OutData : Digit 값으로 아날로그 출력 값.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	출력발생에 실패 했음을 의미합니다.
1	출력발생에 성공 했음을 의미합니다.


NAME

ecaoSetChanVal_F


- 대상 아날로그 출력 채널을 통해 Volt 값 출력

INFORMATION

 Analog I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_success ecaoSetChanVal_F([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_f64 OutData, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 대상 아날로그 출력 채널에 대하여 지정한 Volt 값을 출력합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 출력 채널 번호
- ▶ OutData : Volt 값으로 아날로그 출력 값.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

REFERENCE

□ 이 함수가 정상적으로 동작하려면 슬레이브 장치의 ESI(XML) 파일 또는 “ComiECatDevInfo.xml” 파일에서 아날로그 출력 값의 스케일링 정보가 잘 정의되어 있어야만 적용 가능합니다. 그렇지 않은 경우에는 ecaoSetChanVal_FS 함수를 사용하시기 바랍니다.


NAME


ecaoGetOutValue_I


- 대상 아날로그 출력 채널의 출력 Digit 값 반환

INFORMATION

 Analog I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_j32 ecaoGetOutValue_I([in] t_j32 NetID, [in] t_ui32 Channel, [out] t_j32 *ErrCode)

DESCRIPTION




이 함수는 대상 아날로그 출력 채널에 대하여 출력 상태를 Digit 값으로 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 채널
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 아날로그 출력 Digit 값을 반환 합니다.

NAME	INFORMATION
ecaoGetOutValue_F - 대상 아날로그 출력 채널의 출력 Volt 값 반환	<div data-bbox="1110 394 1394 427">  Analog I/O Function </div> <div data-bbox="1110 439 1394 472">  VC++/BCB/.NET </div> <div data-bbox="1110 483 1394 517">  Level 1 </div> <div data-bbox="1110 528 1394 562">  위험 요소 없음 </div>
SYNOPSIS	
<pre>□ t_f64 ecaoGetOutValue_F([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode)</pre>	

DESCRIPTION

이 함수는 대상 아날로그 출력 채널에 대하여 출력 상태를 Volt 값으로 반환합니다

PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 채널
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 아날로그 출력 Volt 값을 반환 합니다.

Motion General Functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.




8 Motion General Functions

8.1 함수 요약

Summary of Functions	
<p>□ t_i32 ecmGn_GetAxisList([in] t_i32 NetID, [out] t_ui8 AxisListBuf[], [in] t_ui8 AxisListBufSize, t_i32 *ErrCode)</p> <p>대상 네트워크에 연결되어 있는 Servo 축 List 를 반환합니다.</p>	
<p>□ t_success ecmGn_InitFromFile ([in] t_i32 NetID, [in] char *szMotCfgFile, [out] t_i32 *ErrCode)</p> <p>Motion Configuration File 을 통하여 설정값 초기화를 수행 합니다.</p>	

8.2 함수 설명

NAME	INFORMATION
<p>ecmGn_GetAxisList</p> <p>- 네트워크 축 리스트 반환</p>	<p> Motion General Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>
SYNOPSIS	
<p>□ t_i32 ecmGn_GetAxisList([in] t_i32 NetID, [out] t_ui8 AxisListBuf[], [in] t_ui8 AxisListBufSize, t_i32 *ErrCode)</p>	

DESCRIPTION





이 함수는 대상 네트워크에 연결되어 있는 서보 축 번호 리스트를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ AxisListBuf : 서보 축 번호 리스트가 반환될 리스트 버퍼.
- ▶ AxisListBufSize : 축 리스트 버퍼의 사이즈.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 대상 네트워크에 연결되어 있는 축 개수가 반환됩니다..

NAME ecmGn_InitFromFile -	I N F O R M A T I O N
	 Motion General Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

□ t_success ecmGn_InitFromFile ([in] t_i32 NetID, [in] char *szMotCfgFile, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 XML 형식의 Motion Configuration 파일을 읽어와 모션관련 각종 초기값(속도, Unit Distance/Speed, Speed Pattern, Home 복귀 파라미터, 링카운터, 소프트웨어 리밋, 마스터/슬레이브 및 Motion IO 속성)을 설정합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ szMotCfgFile : XML 형식의 Motion Configuration File 명(경로 포함)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	함수 수행이 실패했음을 의미합니다.
1	함수 수행이 성공 했음을 의미합니다.

Motion Single Axis Functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.



9 Motion Single Axis Functions

9.1 함수 요약

Summary of Functions
<p>□ <code>t_cmdidx ecmSxCfg_SetMioProp</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>EEcmMioPropId</code> PropId, [in] <code>t_i32</code> PropVal, [out] <code>t_i32 *ErrCode</code>) 모션 입출력 신호의 환경설정을 구성합니다</p>
<p>□ <code>t_i32 ecmSxCfg_GetMioProp</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>EEcmMioPropId</code> PropId, [out] <code>t_i32 *ErrCode</code>) 모션 입출력 신호의 환경설정 값을 반환합니다.</p>
<p>□ <code>t_cmdidx ecmSxCfg_SetSpeedPatt</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>t_i32</code> SpeedMode, [in] <code>t_f64</code> VIni, [in] <code>t_f64</code> VEnd, [in] <code>t_f64</code> VWork, [in] <code>t_f64</code> Acc, [in] <code>t_f64</code> Dec, [out] <code>t_i32 *ErrCode</code>) 모션 이송의 전역 기준속도를 설정합니다. 이 속도의 비율을 통해 모션 이송의 실제 속도를 설정할 수 있습니다.</p>
<p>□ <code>t_success ecmSxCfg_GetSpeedPatt</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [out] <code>t_i32 *SpeedMode</code>, [out] <code>t_f64 *VIni</code>, [out] <code>t_f64 *VEnd</code>, [out] <code>t_f64 *VWork</code>, [out] <code>t_f64 *Acc</code>, [out] <code>t_f64 *Dec</code>, [out] <code>t_i32 *ErrCode</code>) 모션 이송의 전역 기준 속도를 반환합니다. 반환된 이 속도의 비율을 통해 모션 이송의 실제 속도가 설정됩니다.</p>
<p>□ <code>t_cmdidx ecmSxCfg_SetSpeedPatt_Time</code> ([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>t_i32</code> SpeedMode, [in] <code>t_f64</code> VIni, [in] <code>t_f64</code> VEnd, [in] <code>t_f64</code> VWork, [in] <code>t_f64</code> AccTime, [in] <code>t_f64</code> DecTime, [out] <code>t_i32 *ErrCode</code>) 모션의 이송 전역 기준속도를 설정합니다. 가속도를 초단위로 설정합니다.</p>
<p>□ <code>t_success ecmSxCfg_GetSpeedPatt_Time</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [out] <code>t_i32 *SpeedMode</code>, [out] <code>t_f64 *VIni</code>, [out] <code>t_f64 *VEnd</code>, [out] <code>t_f64 *VWork</code>, [out] <code>t_f64 *AccTime</code>, [out] <code>t_f64 *DecTime</code>, [out] <code>t_i32 *ErrCode</code>) 모션의 이송 전역 기준속도를 반환합니다. 가속도를 초단위로 반환합니다.</p>
<p>□ <code>t_cmdidx ecmSxCfg_SetSpeedRatio</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>t_f64</code> SpeedRatio, [out] <code>t_i32 *ErrCode</code>) 모션의 이송 전역 기준속도를 비율로 설정합니다..</p>
<p>□ <code>t_f64 ecmSxCfg_GetSpeedRatio</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [out] <code>t_i32 *ErrCode</code>) 모션의 이송 전역 기준속도를 비율로 반환합니다..</p>
<p>□ <code>t_cmdidx emcSxCfg_SetJerkRatio</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>t_f64</code> JerkTimeRatio, [out] <code>t_i32 *ErrCode</code>) 지정된 축이 S-Curve 속도 프로파일을 사용하는 경우에 가/감속 구간의 시간에 대해서 Jerk 가 0 이 아닌 구간의 시간 비율을 설정합니다.</p>
<p>□ <code>t_f64 emcSxCfg_GetJerkRatio</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [out] <code>t_i32 *ErrCode</code>) 지정된 축의 현재 설정된 Jerk 시간 비율을 반환합니다.</p>
<p>□ <code>t_success ecmSxCfg_SetUnitDist</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [in] <code>t_f64</code> UnitDist, [out] <code>t_i32 *ErrCode</code>) 논리적 단위 거리에 대한 펄스 수를 설정합니다.</p>
<p>□ <code>t_f64 ecmSxCfg_GetUnitDist</code>([in] <code>t_i32</code> NetID, [in] <code>t_i32</code> Axis, [out] <code>t_i32 *ErrCode</code>) 논리적 단위 거리에 대한 펄스 수를 반환합니다</p>

<p>□ t_success ecmSxCfg_SetUnitSpeed([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 UnitSpeed, [out] t_i32 *ErrCode) 논리적 단위 속도에 대한 실제 펄스 속도(PPS)를 설정합니다</p>
<p>□ t_f64 ecmSxCfg_GetUnitSpeed([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 논리적 단위 속도에 대한 실제 펄스 속도(PPS)를 반환합니다.</p>
<p>□ t_success ecmSxCfg_SetMastInp ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_f64 InpRange, [out] t_i32 *ErrCode) 마스터 장치에서 관리하는 Inposition 기능을 설정합니다. (위치 값을 논리 거리 단위로 설정)</p>
<p>□ t_success ecmSxCfg_GetMastInp([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool* IsEnable, [out] t_f64* InpRange, [out] t_i32 *ErrCode) 마스터 장치에서 관리하는 Inposition 기능 설정 값을 반환합니다. (위치 값을 논리 거리 단위로 읽음)</p>
<p>□ t_success ecmSxCfg_SetMastInp_Cnt([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_i32 InpRangeCnt, [out]t_i32 *ErrCode) 마스터 장치에서 관리하는 Inposition 기능을 설정합니다.(위치 값을 펄스 단위로 설정)</p>
<p>□ t_success ecmSxCfg_GetMastInp_Cnt ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool* IsEnable, [out] t_i32* InpRangeCnt, [out] t_i32 *ErrCode) 마스터 장치에서 관리하는 Inposition 기능 설정 값을 반환합니다.(위치 값을 펄스 단위로 읽음)</p>
<p>□ t_success ecmSxCfg_SetSoftLimit ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_boolIsEnable, [in] t_f64 NegLimit, [in] t_f64 PosLimit, [in] t_bool IsDecelStop, [out] t_i32 *ErrCode) 대상 축의 Software Limit 을 설정합니다.</p>
<p>□ t_success ecmSxCfg_GetSoftLimit ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool * IsEnable, [out] t_f64 *NegLimit, [out] t_f64 *PosLimit, [out] t_bool IsDecelStop, [out] t_i32 *ErrCode) 대상 축의 설정된 Software Limit 값을 확인합니다.</p>
<p>□ t_cmdidx ecmSxCtl_SetSvon([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SvonVal, [out] t_i32 *ErrCode) 대상 축의 서보 드라이브에 SERVO ON 신호 출력을 인가 혹은 차단 합니다.</p>
<p>□ t_cmdidx ecmSxCtl_ResetAlm([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 대상 축에 대해서 알람을 리셋합니다.</p>
<p>□ t_cmdidx ecmSxMot_VMoveStart([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Dir, [out] t_i32 *ErrCode) 단축 연속속도이송을 시작합니다. 이 구동 함수는 구동 시작 후 바로 반환됩니다.</p>
<p>□ t_cmdidx ecmSxMot_MoveStart([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Distance, [out] t_i32 *ErrCode) 단축 상대좌표이송을 시작합니다. 이 구동 함수는 구동 시작 후 바로 반환됩니다.</p>
<p>□ t_success ecmSxMot_Move([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Distance, [out] t_i32 *ErrCode) 단축 상대좌표 이송을 시작합니다. 이 구동 함수는 구동 시작 후 이송이 끝날때까지 반환되지 않습니다.</p>
<p>□ t_success ecmSxMot_Move_NB([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Distance, [out] t_i32 *ErrCode) 단축 상대좌표 이송을 시작합니다. 이 구동 함수는 구동 시작 후 이송이 끝날때까지 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmSxMot_MoveToStart([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Position, [out] t_i32 *ErrCode) 단축 절대좌표 이송을 시작합니다. 이 구동 함수는 구동 시작 후 바로 반환됩니다.</p>


<p>□ t_success ecmSxMot_MoveTo([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Position, [out] t_i32 *ErrCode) 단축 절대좌표 이송을 시작합니다. 이 구동 함수는 구동 시작후 이송이 끝날때까지 반환되지 않습니다.</p>
<p>□ t_success ecmSxMot_MoveTo_NB([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Position, [out] t_i32 *ErrCode) 단축 절대좌표 이송을 시작합니다. 이 구동 함수는 구동 시작후 이송이 끝날때까지 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmSxMot_OverrideSpeed([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 단축 모션 작업이 진행되고 있는 중에 속도를 변경합니다.</p>
<p>□ t_cmdidx ecmSxMot_OverrideMove([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 NewDist, [out] t_bool *IsIgnored, [out] t_i32 *ErrCode) 단축 구동 함수를 통해서 구동되는 단축상대좌표이송 모션에 대하여, 상대 좌표상의 목표 논리 거리 값을 수정합니다.</p>
<p>□ t_cmdidx ecmSxMot_OverrideMoveTo([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 NewPos, [out] t_bool *IsIgnored, [out] t_i32 *ErrCode) 단축 구동 함수를 통해서 구동되는 단축절대좌표이송 모션에 대하여, 절대 좌표상의 목표 논리 거리 값을 수정합니다.</p>
<p>□ t_cmdidx ecmSxMot_Stop([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode) 대상 축의 이송을 정지 합니다.</p>
<p>□ t_success ecmSxSt_WaitCompt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 대상 축의 이송의 완료 시점까지 대기 합니다</p>
<p>□ t_success ecmSxSt_WaitCompt_NB([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 대상 축의 이송의 완료 시점까지 대기 합니다</p>
<p>□ t_cmdidx ecmSxSt_SetCount([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 NewPosCount, [out] t_i32 *ErrCode) 대상 축에 지정한 카운터의 값으로 위치카운터를 설정합니다. 단, 이때 지정하는 카운터값의 단위는 펄스 수 입니다.</p>
<p>□ t_i32 ecmSxSt_GetCount([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode) 대상 축에 지정한 위치 카운터의 값을 반환합니다. 단, 이때 반환되는 카운터값의 단위는 펄스 수 입니다.</p>
<p>□ t_cmdidx ecmSxSt_SetPosition([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 NewPosition, [out] t_i32 *ErrCode) 대상 축에 지정한 위치로 위치 카운터를 설정합니다. 단, 이때 지정하는 카운터값의 단위는 논리적인 거리 단위 입니다.</p>
<p>□ t_f64 ecmSxSt_GetPosition([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode) 대상 축에 지정한 위치 값을 반환합니다. 단, 이때 반환되는 카운터값의 단위는 논리적인 거리 단위 입니다.</p>
<p>□ t_f64 ecmSxSt_GetCurSpeed([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode) 대상 축의 Command 또는 Feedback 속도를 확인하여, 논리적 속도 단위로 반환합니다.</p>
<p>□ t_f64 ecmSxSt_GetCurTorque([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 대상 축의 토크값을 확인하여 반환합니다.</p>

□ t_i32 ecmSxSt_GetMotState([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *SubErrData)
대상 축의 모션 상태를 확인합니다.

□ t_word ecmSxSt_GetDI([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)
대상 축에서 제공하는 디지털입력 상태 값을 반환합니다.

9.2 함수 설명

9.2.1 기본 환경 설정

NAME	I N F O R M A T I O N
ecmSxCfg_SetMioProp	 Single Axis Function
ecmSxCfg_GetMioProp	 VC++/BCB/.NET
- 모션 입출력 신호 환경설정 및 반환	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmSxCfg_SetMioProp([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmMioPropId PropId, [in] t_i32 PropVal, [out] t_i32 *ErrCode)
- t_i32 ecmSxCfg_GetMioProp)([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmMioPropId PropId, [out] t_i32 *ErrCode)

DESCRIPTION

ecmSxCfg_SetMioProp() 는 각종 모션 입출력 신호에 대한 환경을 설정합니다. PropId 에 따라 어떠한 환경을 설정할 지를 결정하게 됩니다.

ecmSxCfg_GetMioProp() 는 각종 모션 입출력 신호에 대하여 현재 설정된 환경설정값을 반환합니다. PropId 에 따라 어떠한 환경설정값을 반환할 지 결정됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis: 대상 축 번호.
- ▶ PropId : 어떠한 환경에 대하여 설정할 것인지를 지정하는 매개 변수입니다. 이 값에 대해서는 아래 표를 참조하십시오.
- ▶ PropVal : PropId 로 지정된 환경에 대한 설정 및 반환값.

PropId	Meaning & PropVal
0 or ecmMPID_EL_STOP_MODE	Limit 센서 감지 시의 정지 모드 0: 즉시정지(Default) 1: 감속정지
10 or ecmMPID_SWL_STOP_MODE	SW Limit 감지 시의 정지 모드 0: 즉시정지(Default) 1: 감속정지
20 or ecmMPID_ALM_STOP_MODE	Servo Alarm 감지 시의 정지 모드 0: 즉시정지(Default) 1: 감속정지

30 or ecmMPID_INP_ENABLE	서보드라이버의 INP 신호를 모션의 완료 상태 체크에 반영할 것인지 설정 0 : Disable(Default) 1 : Enable
40 or ecmMPID_SVOFF_MOVE_MODE	서보 OFF 상태에서의 이송 허용 모드 0 : 허용안함(Default) 1 : 허용함
50 or ecmMPID_CLEAR_INITIAL_ALARM	EtherCAT AL STATE 가 OP 모드로 전환될 때 1 회에 한해서 서보앰프의 알람을 클리어할 것인지에 대한 설정 0 : Disable(Default) 1 : Enable
60 or ecmMPID_INPUTPDO_TYPE	각 축의 InputPDO 데이터를 마스터 장치에서 PC 측으로 전달할 때의 데이터 구조를 결정
70 or ecmMPID_IGNORE_ELN	(-)Limit 신호를 무시하라는 옵션에 대한 설정 0 : Disable(Default) 1 : Enable
71 or ecmMPID_IGNORE_ELP	(+)Limit 신호를 무시하라는 옵션에 대한 설정 0 : Disable(Default) 1 : Enable
72 or ecmMPID_ELN_INPUT_SEL (과나소닉 서보 드라이버)	(-)Limit 신호로 사용할 신호 입력 핀을 설정하는 옵션 0:NOT 신호(Default), 1:SIMON1/EXT1 신호, 2:SIMON2/EXT2 신호, 3:SIMON3, 4:SIMON4, 5:SIMON5
73 or ecmMPID_ELP_INPUT_SEL (과나소닉 서보 드라이버)	(+)Limit 신호로 사용할 신호 입력 핀을 설정하는 옵션 0:NOT 신호(Default), 1:SIMON1/EXT1 신호, 2:SIMON2/EXT2 신호, 3:SIMON3, 4:SIMON4, 5:SIMON5

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmSxCfg_SetMioProp 함수의 반환 값: CmdIdx 를 반환 합니다. CmdIdx 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.

* ecmSxCfg_GetMioProp 함수의 반환 값: 현재 설정된 환경 설정 값을 반환합니다. 이 때의 환경 설정 값의 종류는 PropId 인자에 따라서 결정됩니다.

EXAMPLE


* ecmSxMot_MoveStart EXAMPLE 참고

NAME


ecmSxCfg_SetUnitDist
 ecmSxCfg_GetUnitDist
 - 논리적 거리 단위 설정 및 반환

INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

❑ t_success ecmSxCfg_SetUnitDist([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 UnitDist, [out] t_i32 *ErrCode)

❑ t_f64 ecmSxCfg_GetUnitDist([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 논리적 거리 단위 거리에 대한 펄스 수를 설정합니다. 여기서 논리적 단위 거리라 함은 Move 함수에서 사용하는 단위 거리 또는 위치에 대한 펄스 수를 의미합니다. 이 함수를 사용하여 특별히 지정하지 않는 경우에는 논리적 단위 거리에 대한 펄스 수는 초기 값인 '1'로 사용됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ UnitDist : 논리적 거리 1 을 이동하기 위해서 출력되어야 하는 펄스 수를 지정합니다
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmSxCfg_SetUnitDist : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	설정에 실패하였음을 의미합니다.
1	설정에 성공하였음을 의미합니다.

* ecmSxCfg_GetUnitDist : 이 리턴값은 논리적 거리 단위가 반환됩니다.

REFERENCE

□. Unit distance & Unit speed

모션컨트롤러에서 이동거리는 기본적으로 Command 펄스 수에 의해 결정되고, 이동속도는 펄스의 주파수에 의해서 결정됩니다. 따라서 이동거리를 물리적인 거리 단위로 하기 위해서는 매번 원하는 물리적인 거리를 이동하기 위해서 필요한 펄스의 수를 계산해야 합니다. 하지만, (주)커미조아 모션컨트롤러는 모든 이동 함수와 속도 설정 함수에서 사용되는 논리적거리와 논리적속도를 사용자가 정의할 수 있도록 하고 있으며, 이것을 정의하는 것이 “Unit distance”와 “Unit speed”입니다. 일반적으로 “Unit distance” Du 는 다음과 같이 계산하면 됩니다.

$$D_u = \frac{P_r}{L_r}$$

Pr: 모터 1 회전에 필요한 펄스수 (모터의 Command 분해능)
 Lr: 모터 1 회전시에 이동되는 기구물의 거리

결과적으로 “Unit distance”에 설정해야하는 값은 논리적 거리 1 을 이동하기 위해서 필요한 출력 펄스의 수가 됩니다. 그리고 특별한 경우가 아니면 “Unit speed”는 “Unit distance”와 같은 값을 설정하면 됩니다. 하지만 “Unit speed”와 “Unit distance”는 필요에 따라 서로 다른 단위를 사용할 수 있습니다.

□. Unit distance & Unit speed 계산 예

다음과 같은 사양의 볼스크류를 사용하는 기구물에서의 경우에 “Unit distance”와 “Unit speed”는 다음과 같이 계산할 수 있습니다.

감속비 $n = NL/NM = 1/2$

볼스크류 Lead (Pb) = 10 mm
 감속비 (n) = 1/2
 모터1회전시 이동거리 (Lr) = $10 * 1/2 = 5$ mm
 모터 Command 분해능 (Pr) = 10000 pulses/rev

Servo motor (10000 pulses/rev)

Unit distance (Du) = $Pr/Lr = 10000/5 = 2000$
 Unit speed (Vu) = 2000

//따라서, Unit distance”와 “Unit speed”를 2000 으로 설정하면 다음과 같이 모든 //이동함수에서 거리의 단위를 mm 단위에 해당하는 값으로 입력할 수 있습니다. //





```
#define NET 0
#define AXIS 0

// 2000/mm
ecmSxCfg_SetUnitDist (NET, AXIS, 2000, NULL);
ecmSxCfg_SetUnitSpeed (NET, AXIS, 2000, NULL);

// 속도=100(mm/s), 가/감속도=1000(mm/s^2) 으로 설정한다. //
ecmSxCfg_SetSpeedPatt (NET, AXIS, ecmSMODE_TRAPE, 0, 0, 100, 1000, 1000);

// (+)방향으로 20mm 이송한다 (실제는 20*2000 = 40000 펄스가 출력됨) //
ecmSxMot_Move (NET, AXIS, 20, NULL);

// (-)방향으로 20mm 이송한다 (실제는 20*2000 = 40000 펄스가 출력됨) //
ecmSxMot_Move (NET, AXIS, -20, NULL);
```


NAME	I N F O R M A T I O N
ecmSxCfg_SetUnitSpeed	 Single Axis Function
ecmSxCfg_GetUnitSpeed	 VC++/BCB/.NET
- 논리적 속도 단위 설정 및 반환	 Level 1
	 위험 요소 없음

SYNOPSIS

- ❑ t_success ecmSxCfg_SetUnitSpeed([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 UnitSpeed, [out] t_i32 *ErrCode)
- ❑ t_f64 ecmSxCfg_GetUnitSpeed([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 논리적 단위 속도(Unit Speed)에 대한 실제 펄스 출력속도(PPS)를 설정합니다. 여기서 논리적 단위 속도라 함은 속도 지정함수에서 사용하는 속도 또는 가속도에 대한 단위량을 의미합니다. 이 함수를 사용하여 특별히 지정하지 않는 경우에는 단위 속도에 대한 펄스 출력속도는 1 PPS 로 사용됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ UnitSpeed : 단위 속도에 대한 펄스 출력 속도(PPS)를 설정합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmSxCfg_SetUnitSpeed 함수의 반환 값: 함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

* ecmSxCfg_GetUnitSpeed 함수의 반환 값: 현재 설정된 논리적 단위 속도(Unit Speed)를 반환합니다.

EXAMPLE

* ecmSxCfg_SetUnitSpeed 함수 참고

<h2 style="margin: 0;">NAME</h2> <p style="margin: 10px 0 0 20px;">ecmSxCfg_SetMastInp</p> <p style="margin: 10px 0 0 20px;">ecmSxCfg_GetMastInp</p> <p style="margin: 10px 0 0 20px;">-</p>	<h3 style="margin: 0;">INFORMATION</h3> <hr/> <p style="margin: 0 0 5px 20px;"> Single Axis Function</p> <hr/> <p style="margin: 0 0 5px 20px;"> VC++/BCB/.NET</p> <hr/> <p style="margin: 0 0 5px 20px;"> Level 1</p> <hr/> <p style="margin: 0 0 5px 20px;"> 위험 요소 없음</p> <hr/>
--	---

SYNOPSIS

- ❑ t_success ecmSxCfg_SetMastInp([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, t_f64 InpRange, [out] t_i32 *ErrCode)
- ❑ t_success ecmSxCfg_GetMastInp([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool *IsEnable, t_f64 *InpRange, [out] t_i32 *ErrCode)

DESCRIPTION

ecmSxCfg_SetMastInp 함수는 마스터 인포지션(Master Inposition) 기능에 대한 환경을 설정하는 함수입니다.

ecmSxCfg_GetMastInp 함수는 마스터 인포지션(Master Inposition) 기능과 관련하여 현재 설정되어 있는 환경 설정 값들을 읽는 함수입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable: 마스터 인포지션 기능의 활성화 여부를 설정합니다.

Value	Meaning	Default
0	마스터 인포지션 기능을 비활성화 합니다.	Y
1	마스터 인포지션 기능을 활성화 합니다.	

- ▶ InpRange: 인포지션 범위를 설정합니다. 이 값은 논리적 거리 단위로 설정합니다.
- ▶ ErrCode: 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 두 함수 모두 함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

REFERENCE

인포지션(Inpostion) 이란?

인포지션이란 모터의 지령위치(Command Position)와 궤환위치(Feedback Position) 간의 편차가 사용자가 설정한 특정 범위 안에 들어와 있는 상태를 말합니다. 그리고 인포지션 기능을 활성화 한다는 것은 인포지션 상태를 이송의 완료 조건에 포함시킨다는 것입니다. 다시 말해서 인포지션이 비활성화된 상태에서는 지령 위치 기준으로 이송의 완료를 판단합니다. 그러나 인포지션 기능을 활성화하면 지령 위치 기준으로 이송이 완료되었어도 Command - Feedback 위치 편차가 인포지션 범위 안에 들어올 때까지 이송이 완료되지 않은 상태로 간주됩니다(이때 ecmSxSt_GetMotState 함수의 반환 값은 5(ecmMST_IN_WAIT_INP)가 됩니다).

인포지션 기능이 활성화된 상태에서는 지령위치가 모두 출력되고, 인포지션 상태가 만족되면 이송은 완료된 것으로 처리되는데, 그 이후에는 다시 이송을 시작하기 전까지 인포지션 범위를 벗어났더라도 이송 상태에는 영향을 주지 않습니다.

마스터 인포지션(Master Inpostion) 이란?

인포지션 상태를 마스터 제어기에서 직접 판단하여 적용하는 것을 말합니다.

일반적으로 서보드라이버에서도 인포지션 관리 기능을 제공하는데 이와 구분하기 위해서 마스터 인포지션이라는 용어를 사용합니다.

서보드라이버에서 제공하는 인포지션 기능을 사용하려면 서보드라이버에서 제공하는 PDO 데이터에 INP 신호가 포함되어 있어야 하는데, 모든 서보드라이버가 INP 신호를 PDO 데이터로 제공하는 것은 아니라서 서보드라이버 인포지션 기능을 사용하려면 INP 신호의 PDO 포함 여부를 잘 살펴보고 사용하여야 합니다.

이에 반해서 마스터 인포지션 기능은 모든 서보드라이버에서 사용 가능하므로 마스터 인포지션 기능을 사용하는 것이 바람직합니다.

EXAMPLE





```
#define NET 0
#define AXIS 0

// 논리거리 단위를 mm 단위로 설정 (1mm 이송에 2000 펄스 필요한 것으로 가정)
ecmSxCfg_SetUnitDist (NET, AXIS, 2000, NULL);
ecmSxCfg_SetUnitSpeed (NET, AXIS, 2000, NULL);

// 마스터인포지션 범위 설정 및 활성화 //
ecmSxCfg_SetMastInp ((NET, AXIS, true, 0.5, NULL);

// 속도=100(mm/s), 가/감속도=1000(mm/s^2) 으로 설정한다. //
ecmSxCfg_SetSpeedPatt (NET, AXIS, ecmSMODE_TRAPE, 0, 0, 100, 1000, 1000);

// 이송 : 지령이 완료되어도 커맨드-피드백 편차가 0.5mm 이내가 될 때까지 함수 반환되지 않음 //
ecmSxMot_Move (NET, AXIS, 20, NULL);
```

NAME	I N F O R M A T I O N
ecmSxCfg_SetMastInp_Cnt	 Single Axis Function
ecmSxCfg_GetMastInp_Cnt	 VC++/BCB/.NET
-	 Level 1
	 위험 요소 없음

SYNOPSIS

- ▣ t_success ecmSxCfg_SetMastInp_Cnt ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, t_i32 InpRangeCnt, [out] t_i32 *ErrCode)
- ▣ t_success ecmSxCfg_GetMastInp_Cnt ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool *IsEnable, t_i32 *InpRangeCnt, [out] t_i32 *ErrCode)

DESCRIPTION

ecmSxCfg_SetMastInp_Cnt 함수는 마스터 인포지션(Master Inposition) 기능에 대한 환경을 설정하는 함수입니다. 이 함수가 앞서 설명한 ecmSxCfg_SetMastInp 함수와 다른 점은 인포지션 범위를 펄스 단위로 설정한다는 것입니다. 참고로 ecmSxCfg_SetMastInp 함수는 인포지션 범위를 논리 거리 단위로 설정합니다.

ecmSxCfg_GetMastInp_Cnt 함수는 마스터 인포지션(Master Inposition) 기능과 관련하여 현재 설정되어 있는 환경 설정 값들을 읽는 함수입니다. 이때 인포지션 범위의 값은 카운트 단위로 반환됩니다.

마스터 인포지션에 대한 자세한 설명은 ecmSxCfg_SetMastInp 함수 설명편을 참고하시기 바랍니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable: 마스터 인포지션 기능의 활성화 여부를 설정합니다.

Value	Meaning	Default
0	마스터 인포지션 기능을 비활성화 합니다.	Y
1	마스터 인포지션 기능을 활성화 합니다.	

- ▶ InpRangeCnt: 인포지션 범위를 설정합니다. 이 값은 **펄스 단위로 설정**합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	설정에 실패하였음을 의미합니다.
1	설정에 성공하였음을 의미합니다.

NAME	INFORMATION
ecmSxCfg_SetSoftLimit	Single Axis Function
ecmSxCfg_GetSoftLimit	VC++/BCB/.NET
- SoftLimit 설정 , 반환 함수	Level 1
	위험 요소 없음

SYNOPSIS

- t_success ecmSxCfg_SetSoftLimit([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, t_f64 NegLimit, t_f64 PosLimit, t_bool IsDecelStop, [out] t_i32 *ErrCode)
- t_success ecmSxCfg_GetSoftLimit([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool *IsEnable, t_f64 *NegLimit, t_f64 *PosLimit, t_bool *IsDecelStop, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수들은 Software Limit 을 설정, 반환하는 함수입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable: 활성화 여부 (0: 활성화 1: 비활성화)
- ▶ NegLimit: (-)방향 SoftLimit
- ▶ PosLimit: (+)방향 SoftLimit
- ▶ IsDecelStop: 정지 방법(0: 즉시정지 1: 감속정지)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	설정에 실패하였음을 의미합니다.
1	설정에 성공하였음을 의미합니다.

NAME	INFORMATION
ecmSxCfg_SetSpeedPatt(_FF) ecmSxCfg_GetSpeedPatt - 모션 이송 기준 속도 설정 및 반환	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmSxCfg_SetSpeedPatt(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SpeedMode, [in] t_f64 Vini, [in] t_f64 Vend, [in] t_f64 Vwork, [in] t_f64 Acc, [in] t_f64 Dec, [out] t_i32 *ErrCode)
- t_success ecmSxCfg_GetSpeedPatt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *SpeedMode, [out] t_f64 *Vini, [out] t_f64 *Vend, [out] t_f64 *Vwork, [out] t_f64 *Acc, [out] t_f64 *Dec, [out] t_i32 *ErrCode)

DESCRIPTION

ecmSxCfg_SetSpeedPatt_(FF) 함수는 지정한 축의 '정격 이송 속도' 패턴을 설정합니다. 여기서 정격이라는 용어를 사용하는 이유는 ecmSxCfg_SetSpeedRatio 함수를 통해서 '이송 속도비'를 1.0(100%)이 아닌 다른 값으로 설정할 수 있기 때문이며, '정격 이송 속도'의 의미는 '이송 속도비'가 1.0(100%)으로 설정되었을 때의 이송 속도를 의미합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis: 대상 축 번호.
- ▶ SpeedMode: 속도모드의 설정 값 및 반환 값입니다. 아래와 같은 값을 가집니다.

Value	Meaning
0 or ecmSMODE_CONST	CONSTANT : 가감속을 수행하지 않습니다.
1 or ecmSMODE_TRAPE	TRAPEZOIDAL : 사다리꼴 가감속을 수행합니다.
2 or ecmSMODE_SCURVE	S-CURVE : S-CURVE 가감속을 수행합니다.

- ▶ Vini: 초기 속도에 대한 설정 및 반환 값입니다.
- ▶ Vend: 이송 완료 시 속도에 대한 설정 및 반환 값입니다.
- ▶ Vwork: 작업 속도에 대한 설정 및 반환 값입니다.
- ▶ Acc: 이 인자는 가속도를 의미합니다. 참고로 가속도(acc)와 가속시간(Tacc)의 관계는 아래와 같습니다.

$$acc = (Vwork - Vini) / Tacc.$$

만일 Vini=0 이라면

$$acc = Vwork / Tacc.$$

또는

$$Tacc = Vwork / acc$$

▶ Dec : 이 인자는 감속도를 의미합니다. 참고로 감속도(dec)와 감속시간(Tacc)의 관계는 아래와 같습니다.

$$dec = (Vwork - Vend) / Tdec.$$

만일 Vend=0 이라면

$$dec = Vwork / Tdec.$$

또는

$$Tdec = Vwork / dec$$

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmSxCfg_SetSpeedPatt 함수의 반환 값: CmdIdx 를 반환 합니다. CmdIdx 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.

* ecmSxCfg_GetSpeedPatt 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.





Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

REFERENCE

□ 속도와 가/감속도의 단위는 ecmSxCfg_SetUnitSpeed 함수에 의해서 결정됩니다.

EXAMPLE

* ecmSxMot_MoveStart EXAMPLE 참고

NAME	I N F O R M A T I O N
ecmSxCfg_SetSpeedPatt_Time(_FF) ecmSxCfg_GetSpeedPatt_Time - 모션 이송 기준 속도 시간 설정 및 반환	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmSxCfg_SetSpeedPatt_Time(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SpeedMode, [in] t_f64 VIni, [in] t_f64 VEnd, [in] t_f64 VWork, [in] t_f64 AccTime, [in] t_f64 DecTime, [out] t_i32 *ErrCode)
- t_success ecmSxCfg_GetSpeedPatt_Time([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *SpeedMode, [out] t_f64 *VIni, [out] t_f64 *VEnd, [out] t_f64 *VWork, [out] t_f64 *AccTime, [out] t_f64 *DecTime, [out] t_i32 *ErrCode)

DESCRIPTION

지정한 축에 대해 속도모드, 작업속도 및 가속 및 감속도를 설정할 수 있으며, 설정된 값을 읽을 수 있습니다. 이 속도는 각 모션제어의 기준 속도로 설정되며 가속도를 시간(초)단위로 설정할 수 있습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis: 대상 축 번호.
- ▶ SpeedMode: 속도모드의 설정 값 및 반환 값입니다. 아래와 같은 값을 가집니다.

Value	Meaning
0 or ecmSMODE_CONST	가감속을 수행하지 않습니다.
1 or ecmSMODE_TRAPE	사다리꼴(TRAPEZOIDAL) 가감속을 수행합니다.
2 or ecmSMODE_SCURVE	S-CURVE 가감속을 수행합니다.

- ▶ VIni: 초기 속도에 대한 설정 및 반환 값입니다.
- ▶ VEnd: 이송 완료 시 속도에 대한 설정 및 반환 값입니다.
- ▶ VWork: 작업 속도에 대한 설정 및 반환 값입니다.
- ▶ AccTime: 가속 시간을 초(sec) 단위의 값으로 설정합니다.
- ▶ DecTime: 감속 시간을 초(sec) 단위의 값으로 설정합니다.

▶ **ErrCode** : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* `ecmSxCfg_SetSpeedPatt_Time()` : 함수의 cmdidx 를 반환합니다.

* `ecmSxCfg_GetSpeedPatt_Time()` : 함수의 성공여부부를 나타내는 불형(Boolean Type) 을 반환합니다.

Value	Meaning
0	설정값 반환에 실패 했음을 의미합니다.
1	설정값 반환에 성공 했음을 의미합니다.

NAME	I N F O R M A T I O N
ecmSxCfg_SetSpeedRatio(_FF) ecmSxCfg_GetSpeedRatio - 모션 기준 속도 비율 설정	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

□ t_cmdidx ecmSxCfg_SetSpeedRatio(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 SpeedRatio, [out] t_i32 *ErrCode)

□ t_64 ecmSxCfg_GetSpeedRatio([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 SpeedRatio, [out] t_i32 *ErrCode)

DESCRIPTION

ecmSxCfg_SetSpeedRatio(_FF)/ ecmSxCfg_GetSpeedRatio 함수는 지정한 축의 '이송 속도 비'를 설정 또는 반환합니다. 실제 이송 지령속도는 해당 축의 정격 속도와 이송 속도 비를 곱한 값으로 적용됩니다.

실제 지령속도 = 정격 속도 x 이송 속도 비

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis: 대상 축 번호.
- ▶ SpeedRatio: 속도의 비율을 설정합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmSxCfg_SetSpeedPatt(), ecmSxCfg_SetSpeedPatt ()_FF : 이 리턴값은 함수의 cmdidx 를 반환합니다.

* ecmSxCfg_GetSpeedPatt() : 이 리턴값은 속도 비율을 반환합니다.

NAME	INFORMATION
<p><code>ecmSxCfg_SetJerkRatio(_FF)</code></p> <p><code>ecmSxCfg_GetJerkRatio</code></p> <p>- Jerk 시간 비율 설정 및 반환</p>	<p> Single Axis Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

SYNOPSIS

- `t_cmdidx ecmSxCfg_SetJerkRatio(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in]t_f64 JerkTimeRatio, [out] t_i32 *ErrCode)`
- `t_f64 ecmSxCfg_GetJerkRatio([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)`

DESCRIPTION

이 함수들은 지정된 축이 S-CURVE 속도 프로파일을 사용하는 경우에 가속/감속 구간의 시간에 대해서 Jerk 가 0 이 아닌 구간의 시간 비율을 설정 및 반환합니다.

PARAMETER

- ▶ `NetID` : Network 번호
- ▶ `Axis` : 대상 축 번호.
- ▶ `JerkTimeRatio` : 가속구간 또는 감속구간에서 Jerk 가 0 이 아닌 구간의 시간 비율을 1.0 기준으로 설정합니다(자세한 내용은 “참고” 항목 참고). 여기에 설정할 수 있는 값의 범위는 0.0 ~ 1.0 입니다.
- ▶ `ErrCode` : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

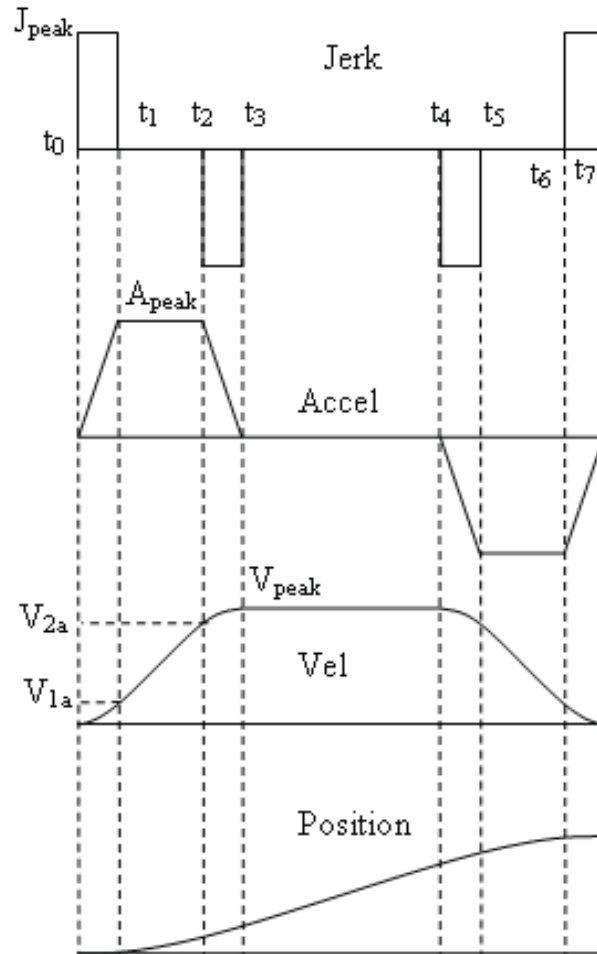
- * `ecmSxCfg_SetJerkRatio(_FF)` : 이 리턴값은 함수의 `cmdidx` 를 반환합니다.
- * `ecmSxCfg_GetJerkRatio()` : 이 리턴값은 `JerkRatio` 값을 반환합니다.

EXAMPLE

- * `ecmSxMot_MoveStart` EXAMPLE 참고

REFERENCE





- 물리적 관점으로 저크(Jerk)는 가속도의 미분 값입니다.
- 저크(Jerk)가 0 인 구간은 등가속도 구간입니다.
- 저크(Jerk), 가속도, 속도, 위치의 관계는 아래의 그래프와 같습니다. 여기서 시간이 $t_0 \sim t_1$ 인 구간과 $t_2 \sim t_3$ 인 구간은 Jerk 가 0 이 아닌 값을 가지는 구간입니다. 그리고 $t_1 \sim t_2$ 인 구간은 Jerk 가 0 인 구간입니다.



- JerkTimeRatio 값은 전체 가속 시간 중에서 $t_0 \sim t_1$ 과 $t_2 \sim t_3$ 구간이 차지하는 비율을 나타냅니다. 예를 들어서 등가속구간이($t_1 \sim t_2$)이 가속시간의 30%를 차지한다면 JerkTimeRatio 는 0.7 이 됩니다(왜냐하면 Jerk 가 0 이 아닌 구간은 1.0 에서 등가속 구간을 뺀 것이 되므로). 마찬가지로 등가속구간이 가속시간의 70%를 차지한다면 JerkTimeRatio 는 0.3 이 될 것입니다.

- JerkTimeRatio 값이 0 이면 사다리꼴 속도패턴과 같아지게 되며, JerkTimeRatio 가 1 이면 완전한 S-Curve 형태가 됩니다. JerkTimeRatio 의 기본값은 0.66 입니다.

9.2.2 상태 제어 명령

NAME	INFORMATION
<p><code>ecmSxCtl_SetSvon(_FF)</code></p> <p><code>ecmSxCtl_GetSvon</code></p> <p>- 서보 On/Off 제어 / 상태반환</p>	<p> Single Axis Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

SYNOPSIS

- `t_cmdidx ecmSxCtl_SetSvon(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SvonVal, [out] t_i32 *ErrCode)`
- `t_bool ecmSxCtl_GetSvon ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)`

DESCRIPTION

`ecmSxCtl_SetSvon(_FF)` 함수는 지정한 축의 서보온(Servo-on) 상태를 제어합니다.
`ecmSxCtl_GetSvon` 함수는 서보드라이버로부터 읽은 서보온(Servo-on) 상태를 반환합니다.

PARAMETER

- ▶ `NetID` : Network 번호
- ▶ `Axis` : 대상 축 번호.
- ▶ `SvonVal` : 지정한 축의 서보온(Servo-on) 상태
- ▶ `ErrCode` : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * `ecmSxCtl_SetSvon` 함수의 반환 값: `cmdidx` 를 반환합니다. `CmdIdx` 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.
- * `ecmSxCtl_GetSvon` 함수의 반환 값: 서보드라이버로부터 읽은 서보온(Servo-on) 상태를 반환합니다

REFERENCE

- `ecmSxCtl_SetSvon` 함수는 서보드라이버에게 서보온 명령을 하달하고 실제 서보온 상태가 변경되는 것을 확인한 후에 반환됩니다. 서보드라이버가 서보온 명령을 받고 실제로

서보온이 완료되기까지는 일정한 시간이 소요되므로 ecmSxCtl_SetSvon 함수의 수행 시간은 서보드라이버 제품의 특성에 따라서 수백 msec 또는 그 이상의 시간이 걸릴 수 있습니다.

만일 많은 수의 서보드라이버의 서보온 상태를 일괄적으로 제어하기 위해서 ecmSxCtl_SetSvon 함수를 순차적으로 실행한다면 시간이 많이 소요될 수 있습니다.

이러한 경우에 ecmSxCtl_SetSvon 함수의 Fast Function 모드인 ecmSxCtl_SetSvon_FF 함수를 사용하면 많은 수의 서보온을 일괄적으로 실행할 때에 빠르게 진행시킬 수 있습니다.

EXAMPLE

```

#define NETID          0
#define NUM_AXES      32
// Axis0 의 서보온 버튼 이벤트핸들러 //
void CMyDlg::OnClick_ServoOn_Axis0()
{
    ecmSxCtl_SetSvon(NETID, 0, true, NULL)
}





// Axis0 의 서보오프 버튼 이벤트핸들러 //
void CMyDlg::OnClick_ServoOff_Axis0()
{
    ecmSxCtl_SetSvon(NETID, 0, false, NULL)
}

// 모든축의 서보온 //
void CMyDlg::OnClick_ServoOn_All()
{
    for(int i=0; i<NUM_AXES; i++){
        if(ecmSxSt_IsConnected (NETID, i, NULL, NULL)){
            ecmSxCtl_SetSvon_FF (NETID, i, TRUE, NULL); // FF 함수 사용
        }
    }

    // 모든 축의 서보온 상태가 바뀔 때까지 대기 //
    BOOL IsAllON = FALSE;
    ULONG dwStartT = timeGetTime();
    while(!ecUtl_IsTimeOut (dwStartT, 2000) && !IsAllON){
        IsAllON = TRUE;
        for(i=0; i<ecMAX_NUM_AXES_PER_NET; i++){
            if(ecmSxSt_IsConnected(NETID, i, NULL, NULL)){
                if(!ecmSxCtl_GetSvon(NETID, i, NULL)){
                    IsAllON = FALSE;
                }
            }
        }
    }

    if(!IsAllON){
        ShowError("일부 축이 서보온 되지 않았습니다\n 다시 시도하십시오.");
    }
}

```

<h1>NAME</h1> <p>ecmSxCtl_ResetAlm(_FF)</p> <p>- 대상 서보 드라이버 알람 리셋</p>	INFORMATION
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```

❑ t_cmdidx ecmSxCtl_ResetAlm(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32
*ErrCode)
    
```

DESCRIPTION

이 함수는 지정한 채널(축)의 알람 리셋을 제어합니다.
 단, 이 함수를 통하여 모든 종류의 서보알람을 리셋할 수 없습니다. EtherCAT 서보 드라이버는 통신을 통하여 알람리셋이 가능한 경우와 서보드라이버의 전원을 차단 후 재투입해야 알람이 리셋되는 경우가 있습니다. 이는 서보 제조사별로 다르며 자세한 내용은 사용하시는 서보드라이버의 매뉴얼을 참조하시기 바랍니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 함수는 cmdidx가 반환됩니다. CmdIdx는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0이면 함수의 수행이 실패했음을 의미합니다.

EXAMPLE





```

C/C++
//본 예제는 ecmSxMot_Move 함수들을 사용하여 첫번째 축을 (+)5000 이동하는 예입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수

void Reset_Alm()
{
    ecmSxCtl_ResetAlm(Device_Info.NetIdx, AXIS0, &Error_Num);
    //0번 축에 알람을 리셋 합니다.
}
    
```


9.2.3 이송 명령

NAME	INFORMATION
ecmSxMot_VMoveStart(_FF) - 단축 연속 속도 이송	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

```

□ t_cmdidx ecmSxMot_VMoveStart(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Dir,
[out] t_i32 *ErrCode)

```

DESCRIPTION

작업속도까지 가속한 후에 작업속도를 유지하며 정지 함수가 호출될 때까지 지정한 방향으로의 모션을 계속 수행합니다. 이 함수는 모션을 시작시킨 후에 바로 반환됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ Dir : 모션 이송 방향을 설정합니다.

Value	Meaning
0 or ecmDIR_N	(-) 방향 (Negative Direction)
1 or ecmDIR_P	(+) 방향 (Positive Direction)

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 함수는 CmdIdx 를 반환 합니다. CmdIdx 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.

EXAMPLE

```

C/C++
// 다음의 예제는 "Jog 이동"을 하는 예입니다. 본 예제에서의
// "Jog 이동"은 버튼이 눌러진 상태에서는 Axis0 축의 이동을
// 수행하다가, 버튼이 풀리면 이동을 멈추는 예입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

```

```

t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* OnProgramInitial : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다. //5~9 장 사이 내용 참고
*****/
void OnProgramInitial()
{
    TEcDevInfo Device_Info; //디바이스 정보 저장 할 구조체
    t_i32 Device_Num = 0;

    t_success nIsLoaded = ecDll_Load ();

    if(lcGn_LoadDevices(&Error_Num)){ //장치 로드
        //로드 실패시 예외 처리
    }

    if(lcGn_GetDevInfo(Device_Num,&Device_Info, &Error_Num)){
        //디바이스 정보 로드 실패 시 예외 처리
    }

    ecNet_SetAlState(Device_Info.NetIdx, ecAL_STATE_OP, &Error_Num);
    //ALState OP 모드로 전환

    ecmSxCfg_SetSvon(Device_Info.NetIdx, AXIS0, &Error_Num);
} OnProgramInitial 함수 끝

/*****
* OnMoveSetting : 이 함수는 이동을 위한 준비 단계 함수
* ecmSxCfg_SetMioProp, ecmSxCfg_SetSpeedPatt, ecmSxCfg_SetJerkRatio
* ecmSxCfg_SetUnitDist, ecmSxCfg_SetUnitSpeed 를 이용하여 세팅
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/

void OnMoveSetting ()
{
    ecmSxCfg_SetMioProp(Device_Info.NetIdx, AXIS0, propID, propID, &Error_Num)
    //모션 입출력 신호 환경설정 (관련 함수 설명 참고)

    t_i64 UnitDist = //논리적 거리를 1 을 이동하기 위해 출력되어져하는 펄스 수 지정

    ecmSxCfg_SetUnitDist( Device_Info.NetIdx, AXIS0, UnitDist , &Error_Num);

    t_f64 UnitSpeed = //단위 속도에 대한 펄스 출력 소도 설정 값
    ecmSxCfg_SetUnitSpeed((Device_Info.NetIdx, AXIS0, UnitSpeed, &Error_Num);

    /* t_f64 VIni = 0//초기 속도 설정 및 반환 값
    t_f64 Vend = 0// 종료 속도 설정 및 반환 값
    t_f64 VWork = 40000//작업 속도에 대한 설정 및 반환 값
    t_f64 Acc = 10000//가속도에 대한 설정 및 반환 값
    t_f64 Dec = 10000//감속도에 대한 설정 및 반환값
    Speed 설정 관련 매개 변수 값*/

```

```

    ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS0, 0, 0, 40000, 10000, 10000, &Error_Num);

    t_f64 JerkTimeRatio = 1.0 //Jerk 값
    ecmSxCfg_SetJerkRatio (Device_Info.NetIdx, AXIS0, JerkTimeRatio, &Error_Num);

}




//*****
* OnPlusButtonDown() : (+)Move 버튼이 눌렸을 때 호출되는 가상의 함수
* 이 함수에서 (+)방향으로 V-Move 를 시작합니다.
*****//
void OnPlusButtonDown ()
{
    ecmSxMot_VMoveStart(Device_Info.NetIdx, AXIS0, ecmDIR_P, &Error_Num);
    //Positive dir V-MOVE
}

//*****
* OnPlusButtonUp() : (-)Move 버튼이 올라올 때 호출되는 가상의 함수
* 이 함수에서는 V-Move 를 종료합니다.
*****//
void OnPlusButtonUp ()
{
    ecmSxMot_Stop(Device_Info.NetIdx, AXIS0, 0, 0, &Error_Num);
}

//*****
* OnMinusButtonDown(): (-)Move 버튼이 눌렸을 때 호출되는 가상의 함수
* 이 함수에서 (+)방향으로 V-Move 를 시작합니다.
*****//
void OnMinusButtonDown()
{
    ecmSxMot_VMoveStart(Device_Info.NetIdx, AXIS0, ecmDIR_N, &Error_Num);
    //Negative dir V-MOVE
}

//*****
* OnMinusButtonUp() : (-)Move 버튼이 올라올 때 호출되는 가상의 함수
* 이 함수에서는 V-Move 를 종료합니다.
*****//
void OnMinusButtonUp()
{
    ecmSxMot_Stop(Device_Info.NetIdx, AXIS0, 0, 0, &Error_Num);
}

```

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;"><code>ecmSxMot_MoveStart(_FF)</code></p> <p style="margin: 0;"><code>ecmSxMot_Move(_NB)</code></p> <p style="margin: 0;">- 단축 상대 좌표 이송</p>	INFORMATION
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

□ `t_cmdidx` `ecmSxMot_MoveStart(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Distance, [out] t_i32 *ErrCode)`

□ `t_success` `ecmSxMot_Move(_NB)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Distance, [out] t_i32 *ErrCode)`

DESCRIPTION

하나의 축에 대하여 현재의 위치에서 지정한 거리(상대 위치)만큼 이동을 수행합니다. `ecmSxMot_MoveStart`, `ecmSxMot_MoveStart_FF` 는 모션을 시작시킨 후에 바로 반환되고, `ecmSxMot_Move`, `ecmSxMot_Move_NB` 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 `ecmSxMot_Move_NB` 는 함수가 반환되지 않지만 윈도우 메시지를 블록킹(Blocking) 하지 않습니다.

PARAMETER

- ▶ `NetID` : Network 번호
- ▶ `Axis` : 대상 축 번호
- ▶ `Distance` : 이동할 거리를 지정합니다.
- ▶ `ErrCode` : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 `NULL` 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* `ecmSxMot_MoveStart(_FF)` 함수의 반환 값: `CmdIdx` 를 반환 합니다. `CmdIdx` 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.

* `ecmSxMot_Move` 함수의 반환 값: 함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

EXAMPLE

```

C/C++
//본 예제는 ecmSxMot_MoveStart 사용하여 첫번째 축을 (+)5000 이동하는 예입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* OnProgramInitial : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다. //5~9 장 사이 내용 참고
*****/
void OnProgramInitial()
{
    TEcDevInfo Device_Info; //디바이스 정보 저장 할 구조체
    t_i32 Device_Num = 0;

    t_success nIsLoaded = ecDll_Load ();

    if(!ecGn_LoadDevices(&Error_Num)){ //장치 로드
        //로드 실패시 예외 처리
    }

    if(!ecGn_GetDevInfo(Device_Num,&Device_Info, &Error_Num)){
        //디바이스 정보 로드 실패시 예외 처리
    }

    ecNet_SetAlState(Device_Info.NetIdx, ecAL_STATE_OP, &Error_Num);
    //ALState OP 모드로 전환

    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXIS0, &Error_Num);
} OnProgramInitial 함수 끝

/*****
* OnMoveSetting : 이 함수는 이동을 위한 준비 단계 함수
* ecmSxCfg_SetMioProp, ecmSxCfg_SetSpeedPatt, ecmSxCfg_SetJerkRatio
* ecmSxCfg_SetUnitDist, ecmSxCfg_SetUnitSpeed 를 이용하여 세팅
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/

void OnMoveSetting ()
{
    ecmSxCfg_SetMioProp(Device_Info.NetIdx, AXIS0, propID, propID, &Error_Num)
    //모션 입출력 신호 환경설정 (관련 함수 설명 참고)

    t_i64 UnitDist = //논리적 거리를 1 을 이동하기 위해 출력되어져하는 펄스 수 지정

    ecmSxCfg_SetUnitDist( Device_Info.NetIdx, AXIS0, UnitDist , &Error_Num);

    t_f64 UnitSpeed = //단위 속도에 대한 펄스 출력 소도 설정 값
    ecmSxCfg_SetUnitSpeed(|Device_Info.NetIdx, AXIS0, UnitSpeed, &Error_Num);

    /* t_f64 VIni = 0//초기 속도 설정 및 반환 값
    t_f64 Vend = 0// 종료 속도 설정 및 반환 값

```

```

t_f64 VWork = 40000//작업 속도에 대한 설정 및 반환 값
t_f64 Acc = 10000//가속도에 대한 설정 및 반환 값
t_f64 Dec = 10000//감속도에 대한 설정 및 반환값
Speed 설정 관련 매개 변수 값*/

ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS0, 0, 0, 40000, 10000, 10000, &Error_Num );

t_f64 JerkTimeRatio = 1.0 //Jerk 값
ecmSxCfg_SetJerkRatio (Device_Info.NetIdx, AXIS0, JerkTimeRatio, &Error_Num);




}

/*****
* DO_MOVE : 작업 명령 시에 호출되는 가상의 함수 입니다.
*cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
*그대로 유지 하면서 상대 위치 만큼 +5000 이동
*****/
void DO_MOVE()
{
    //cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
    //그대로 유지 하면서 상대 위치 만큼 +5000 이동

    ecmSxMot_MoveStart(Device_Info.NetIdx, AXIS0, 5000m &Error_Num); //Move +5000

    if(!ecmSxSt_WaitCompt(Device_Info.NetIdx, AXIS0, &Error_Num)){
        //축이 +5000 만큼 이동 되는 동안 블로킹하는 함수
        // 모션 완료 대기 실패 예외 처리
        return;
    }
}

```

<h2>NAME</h2> <p>ecmSxMot_MoveToStart(_FF)</p> <p>ecmSxMot_MoveTo(_NB)</p> <p>- 단축 절대 좌표 이송</p>	INFORMATION
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

□ t_cmdidx ecmSxMot_MoveToStart(_FF)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Position, [out] t_i32 *ErrCode)

□ t_success ecmSxMot_MoveTo(_NB)([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Position, [out] t_i32 *ErrCode)

DESCRIPTION

대상 축에 대하여 지정한 절대좌표로의 이동을 수행합니다.
 ecmSxMot_MoveToStart, ecmSxMot_MoveToStart_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmSxMot_MoveTo, ecmSxMot_MoveTo_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmSxMot_MoveTo_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블로킹(Blocking) 하지 않습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ Position : 이동할 절대 좌표 값을 지정합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다

RETURN VALUE

* ecmSxMot_MoveToStart(_FF) () 함수의 반환 값: CmdIdx 를 반환 합니다. CmdIdx 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.

* ecmSxMot_MoveTo () 함수의 반환 값: 함수 수행의 성공 여부를 반환합니다.

Value	Meaning
0	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수





/*****
* 프로그램 앞 부분은 생략 -> ecmSxMot_MoveStart 예제 참고
*****/

/*****
* DO_MOVE : 작업 명령 시에 호출되는 가상의 함수 입니다.
* cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
* 그대로 유지 하면서 상대 위치 만큼 +5000 이동
*****/
void DO_TOMOVE()
{
    //cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
    //그대로 유지 하면서 절대 위치 5000 으로 이동

    ecmSxMot_MoveToStart(Device_Info.NetIdx, AXIS0, 5000m &Error_Num); //Move 5000

    if(!ecmSxSt_WaitCompt(Device_Info.NetIdx, AXIS0, &Error_Num)){
        //축이 5000 으로 이동 되는 동안 블로킹하는 함수
        // 모션 완료 대기 실패 예외 처리
        return;
    }
}

```

NAME ecmSxMot_OverrideMove - 단축 상대위치 오버라이드 이송	INFORMATION
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

```

□ t_cmdidx ecmSxMot_OverrideMove([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64
NewDist, [out] t_bool *IsIgnored, [out] t_i32 *ErrCode)

```

DESCRIPTION

이 함수는 ecmSxMot_MoveStart(_FF) () 이송 함수를 통하여 수행되는 상대좌표 이송 모션에 대하여 상대좌표값, 즉 목표 거리값을 변경(오버라이딩)하는 함수입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ NewDist : 새로운 목표 거리값을 지정합니다. 이 값의 기준 위치는 오버라이드하고자 하는 대상이 되는 ecmSxMot_MoveStart() 작업에서 사용한 기준점과 같습니다.
- ▶ IsIgnored : 오버라이드 적용 성공/실패 여부를 반환합니다.

Value	Meaning
0	모션에러가 발생하였거나 이미 이송이 완료되어 위치 오버라이드가 적용되지 않음
1	위치 오버라이드가 적용됨

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 이 리턴값은 함수의 cmdidx 가 반환됩니다.

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"

```

```
#define AXIS0 0

t_32 Error_Num = 0; //함수 별 에러 코드 저장 변수

//*****
* 프로그램 앞 부분은 생략 -> ecmSxMot_MoveStart 예제 참고
*****//


//*****
* DO_OverrideMove: 작업 명령 시에 호출되는 가상의 함수 입니다.
* 상대 좌표 값을 새롭게 지정 해주어 오버라이딩 하는 함수
*****//
void DO_OverrideMove ()
{
    t_bool IsIgnored = false; //성공 여부 확인
    ecmSxMot_OverrideMove(Device_Info.NetIdx, AXIS0, 8000, IsIgnored, &Error_Num);
    //이송 중인 축의 목표 상대 거리를 8000 으로 바꾼다.
}
```

NAME


ecmSxMot_OverrideMoveTo

- 단축 절대위치 오버라이드 이송

INFORMATION
 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
SYNOPSIS

```

❑ t_cmdidx ecmSxMot_OverrideMoveTo([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64
NewPos, [out] t_bool *IsIgnored, [out] t_i32 *ErrCode)

```

DESCRIPTION

이 함수는 ecmSxMot_MoveToStart(_FF)() 이송 함수를 통하여 수행되는 절대좌표 모션에 대하여 목표 절대좌표값, 즉 목표 위치값을 변경(오버라이딩) 하는 함수입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ NewPos : 새로운 목표 절대좌표값을 지정합니다.
- ▶ IsIgnored : 오버라이드 적용 성공/실패 여부를 반환합니다.

Value	Meaning
0	모션에러가 발생하였거나 이미 이송이 완료되어 위치 오버라이드가 적용되지 않음
1	위치 오버라이드가 적용됨

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 이 리턴값은 함수의 cmdidx 가 반환됩니다.

EXAMPLE

C/C++

```
#include "ComiEcatSdk_Api.h"
```

```
#define AXIS0 0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

//*****
* 프로그램 앞 부분은 생략 -> ecmSxMot_MoveStart 예제 참고
*****//

//*****
* DO_OverrideMoveTo: 작업 명령 시에 호출되는 가상의 함수 입니다.
** 절대 좌표 값을 새롭게 지정 해주어 오버라이딩 하는 함수
*****//
void DO_OverrideMoveTo()
{
    t_bool IsIgnored = false; //성공 여부 확인
    ecmSxMot_OverrideMove(Device_Info.NetIdx, AXIS0, 8000, IsIgnored, &Error_Num);
    //이송 중인 축의 목표 절대 좌표를 8000 으로 바꾼다.
}


```

NAME


ecmSxMot_OverrideSpeed
- 단축속도 오버라이딩 실행

INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_cmdidx ecmSxMot_OverrideSpeed([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)
```

DESCRIPTION

이 함수는 단축 모션이 진행되고 있는 중에 속도를 변경(오버라이딩)하고자 할 때 사용하는 함수입니다. 속도를 오버라이딩하기 위해서는 먼저 ecmSxCfg_SetSpeedPatt() 속도 패턴 설정 함수를 통하여 변경하고자 하는 속도 또는 가속도값을 설정하고 나서 이 함수를 수행해야 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 함수의 cmdidx가 반환됩니다.

EXAMPLE

```
C/C++
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

//*****
* 프로그램 앞 부분은 생략 -> ecmSxMot_MoveStart 예제 참고
*****//

//*****
```


```
* DO_OverrideSpeed: 작업 명령 시에 호출되는 가상의 함수입니다.  
*cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를  
* 함수에서 재 설정 한 속도로 바꾸어 이송을 진행하는 함수  
*****//  
void DO_OverrideSpeed ()  
{  
    ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS0, 0, 0, 40000, 10000, 10000, &Error_Num );  
    //SpeedPatt 재 설정  
    ecmSxMot_OverrideSpeed (Device_Info.NetIdx, AXIS0, &Error_Num);  
    //재 설정 된 SpeedPatt 으로 남은 이송 진행  
  
}
```


NAME


ecmSxMot_Stop(_FF)
- 단축 이송 정지

INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_cmdidx ecmSxMot_Stop([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 지정한 축에 대한 모션을 정지 합니다. 함수 인자에 따라 정지시에 감속 후 정지, 급정지를 수행합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsDecStop : 정지 시 감속 정지, 급정지를 결정하는 매개변수입니다..

Value	Meaning
0	급 정지
1	감속 정지

- ▶ IsWaitCompt : 정지가 끝날때까지 대기할지 여부를 결정하는 매개변수 입니다.

Value	Meaning
0	대기하지 않음
1	대기함

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 이 리턴값은 함수의 cmdidx 가 반환됩니다.

EXAMPLE





```
C/C++
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

void DO_Stop()
{
    ecmSxMot_Stop(Device_Info.NetIdx, AXIS0, 0, 0, &Error_Num);
    //급정지, 정지가 끝날때까지 대기 하지 않고 정지
}

```

9.2.4 모션 상태 감시

NAME	I N F O R M A T I O N
ecmSxSt_WaitCompt(_NB) - 단축 모션 완료 대기	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

t_success ecmSxSt_WaitCompt(_NB)([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 단일 축에 대하여 모션이 완료될 때까지 대기 합니다. ecmSxSt_WaitCompt_NB() 는 모션이 완료될때까지 함수가 반환되지 않지만 윈도우 메시지를 블로킹(Blocking)하지 않습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	모션 완료 대기에 실패하였음을 의미합니다
1	모션 완료 대기에 성공하였음을 의미합니다.

EXAMPLE

* ecmSxMot_MoveStart EXAMPLE 참고


NAME

`ecmSxSt_SetCount`


- 사용자정의 위치 카운트 값 설정

INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```
□ t_cmdidx ecmSxSt_SetCount([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 NewPosCount,
[out] t_i32 *ErrCode)
```

DESCRIPTION

이 함수는 대상 채널의 지정한 위치 카운터의 값을 새로이 설정합니다. 단, 이때 지정하는 카운터값의 단위는 펄스 수입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ NewPosCount : 지정한 값으로 대상 축의 카운터 값을 설정합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 함수의 cmdidx가 반환됩니다.

NAME

ecmSxSt_GetCount

- 사용자정의 위치 카운트 값 반환

INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

SYNOPSIS

□ t_i32 ecmSxSt_GetCount([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 대상 채널의 지정한 위치 카운터의 값을 읽어서 반환합니다., 단 이때 반환되는 값의 단위는 펄스 수 입니다.

PARAMETER





- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TargCntr : 값을 읽을 카운터 번호.

Value	Meaning
0 or ecmCNT_COMM	Command Counter
1 or ecmCNT_FEED	Feedback Counter

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 이 리턴값은 지정한 카운터 값이 반환됩니다.

NAME	INFORMATION
<p>ecmSxSt_SetPosition</p> <p>- 사용자정의 논리적 위치 카운트 값 설정</p>	<p> Single Axis Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

SYNOPSIS

```

□ t_cmdidx ecmSxSt_SetPosition([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 NewPosition,
[out] t_i32 *ErrCode)

```

DESCRIPTION

이 함수는 대상 채널의 지정한 위치 카운터의 값을 새로이 설정합니다. 단 이때 지정하는 카운터 값의 단위는 논리적 거리 단위입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ NewPosition : 지정한 값으로 대상 축의 카운터 값을 설정합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 함수의 cmdidx 가 반환됩니다.

NAME

ecmSxSt_GetPosition

- 사용자정의 논리적 위치 카운트 값 반환

INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

SYNOPSIS

```

□ t_f64 ecmSxSt_GetPosition([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out]
t_i32 *ErrCode)

```

DESCRIPTION

이 함수는 대상 채널의 지정한 위치 카운터의 값을 읽어서 반환합니다., 단 이때 반환되는 값의 단위는 논리적 거리 입니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TargCntr : 값을 읽을 카운터 번호.

Value	Meaning
0 or ecmCNT_COMM	Command Counter
1 or ecmCNT_FEED	Feedback Counter

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * 이 리턴값은 지정한 카운터 값이 반환됩니다.

NAME

ecmSxSt_GetCurSpeed

- 논리적 속도 반환

INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

SYNOPSIS

□ t_f64 ecmSxSt_GetCurSpeed([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 Command /Feedback 속도를 읽어서 논리적 속도 단위로 반환됩니다. TargCntr 매개변수에 따라서 Command /Feedback 속도에 대해서 반환됩니다.

PARAMETER





- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TargCntr : 값을 읽을 속도 번호.

Value	Meaning
0 or ecmCNT_COMM	Command Speed
1 or ecmCNT_FEED	Feedback Speed

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 지정된 속도가 논리적 단위로 반환됩니다.

NAME	INFORMATION
ecmSxSt_GetCurTorque - 서보모터 토크 값 반환	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

□ t_f64 ecmSxSt_GetCurTorque([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 대상 축 모터의 토크값이 반환됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 지정된 축의 토크값이 반환됩니다.

<h2 style="margin: 0;">NAME</h2> <p style="margin: 10px 0;">ecmSxSt_GetMotState</p> <p style="margin: 0 0 10px 20px;">- 단축 이송 모션 상태 반환</p>	<h3 style="margin: 0;">INFORMATION</h3> <ul style="list-style-type: none"> <li style="border-bottom: 1px solid black; padding: 2px 0;"> Single Axis Function <li style="border-bottom: 1px solid black; padding: 2px 0;"> VC++/BCB/.NET <li style="border-bottom: 1px solid black; padding: 2px 0;"> Level 1 <li style="padding: 2px 0;"> 위험 요소 없음
---	--

SYNOPSIS

□ t_i32 ecmSxSt_GetMotState([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 단축, 다축, 보간 이송시 해당 축의 현재 모션 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 아래 현재 속도 상태에 대한 값이 반환됩니다,

Value	Meaning
0 or ecmMST_STOP	정지 상태
1 or ecmMST_IN_ACCEL	가속 상태
2 or ecmMST_IN_WORKSPD	정속 상태
3 or ecmMST_IN_DECEL	감속 상태
4 or ecmMST_IN_INISPD	초기 속도 상태
5 or ecmMST_IN_WAIT_INP	인포지션 대기 상태
10 or ecmMST_IN_HOME_IDLE	HOME IDLE 상태
13 or ecmMST_IN_SLAVE_MODE	슬레이브 모드 상태
14 or ecmMST_IN_WAIT	대기 상태
15 or ecmMST_UNKNOWN	알수없는 상태

EXAMPLE





```
C/C++
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

void DO_GetMotState()
{
    t_i32 MotState = 0;

    MotState = ecmSxSt_GetMotState Device_Info.NetIdx, AXIS0, &Error_Num);

    // MotState 에서 나오는 값에 따라 현재 축에 상태를 측정 가능 - 함수 설명 참고
}
```

NAME	INFORMATION
<p>ecmSxSt_GetDI</p> <p>- 모터드라이버에서 제공하는 디지털입력의 상태를 반환</p>	<p> Home Operation Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>
SYNOPSIS	
<p>□ t_word ecmSxSt_GetDI ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)</p>	

DESCRIPTION

이 함수는 모터 드라이버에서 제공하는 디지털입력의 상태를 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 모터 드라이버에서 제공하는 디지털입력의 상태를 반환합니다.

Motion Interpolation Functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서 저희 쥬커미조아를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.



10 Motion Interpolation Functions

10.1 함수 요약

Summary of Functions
<p>□ <code>t_cmdidx ecmIxCfg_MapAxes([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_i32 IxMode, [out] t_i32 *ErrCode)</code> 보간 대상 축 그룹을 설정합니다</p>
<p>□ <code>t_cmdidx ecmIxCfg_UnmapAxes([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)</code> 보간 대상 축 그룹을 해제합니다.</p>
<p>□ <code>t_cmdidx ecmIxCfg_SetSpeedPatt([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsVectSpeed, [in] t_i32 SpeedMode, [in] t_f64 VIni, [in] t_f64 VEnd, [in] t_f64 VWork, [in] t_f64 Acc, [in] t_f64 Dec, [out] t_i32 *ErrCode)</code> 보간 이송 속도를 설정합니다. 보간 이송 속도는 마스터 속도모드와 벡터 속도모드를 설정 할 수 있습니다.</p>
<p>□ <code>t_success ecmIxCfg_GetSpeedPatt([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *IsVectSpeed, [out] t_i32 *SpeedMode, [out] t_f64 *VIni, [out] t_f64 *VEnd, [out] t_f64 *VWork, [out] t_f64 *Acc, [out] t_f64 *Dec, [out] t_i32 *ErrCode)</code> 설정된 보간 이송 속도를 반환합니다. 보간 이송 속도는 마스터 속도 혹은 벡터 모드에 해당하는 속도모드를 반환합니다.</p>
<p>□ <code>t_cmdidx ecmIxCfg_SetJerkRatio([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode)</code> 지정된 보간 맵에 S-CURVE 속도 프로파일을 사용하는 경우에 가속/감속 구간의 시간에 대해서 Jerk 가 0 이 아닌 구간의 시간 비율을 설정합니다</p>
<p>□ <code>t_f64 ecmIxCfg_GetJerkRatio([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)</code> 지정된 보간 맵의 현재 설정된 Jerk 시간 비율을 반환합니다.</p>
<p>□ <code>t_cmdidx ecmIxBot_LineStart([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode)</code> 보간 제어에 있어 직선 보간을 수행하며, 상대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ <code>t_success ecmIxBot_Line([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode)</code> 보간 제어에 있어 직선 보간을 수행하며, 상대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ <code>t_cmdidx ecmIxBot_LineToStart([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)</code> 보간 제어에 있어 직선 보간을 수행하며, 절대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환됩니다.</p>
<p>□ <code>t_success ecmIxBot_LineTo([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)</code> 보간 제어에 있어 직선 보간을 수행하며, 절대좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ <code>t_cmdidx ecmIxBot_Stop([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode)</code> 보간 제어 구동 이송을 정지 합니다.</p>

❑ t_bool ecmIxSt_IsBusy([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)
보간 제어 구동 이송이 이송중인지 확인합니다

❑ t_success ecmIxSt_WaitCompt([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)
보간 제어 구동 이송이 완료 될 때까지 대기합니다.

10.2 함수 설명

NAME	INFORMATION
ecmlxCfg_MapAxes	 Interpolation Function
ecmlxCfg_MapAxes_FF	 VC++/BCB/.NET
- 보간 대상 축 그룹 설정	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmlxCfg_MapAxes([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [out] t_i32 *ErrCode)
- t_cmdidx ecmlxCfg_MapAxes_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 보간 작업을 수행할 축들을 맵번호로 맵핑(Mapping)합니다. 맵번호는 다른 “기본보간제어”에 관련된 함수들의 두번째 매개변수로 전달되므로써 각 함수들이 제어해야 할 축들에 대한 정보가 간편하게 전달됩니다. 따라서 다른 보간제어에 관련된 함수들을 사용하기 전에 가장 먼저 이 함수를 사용하여 사용할 축들을 맵핑하여야 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호. 맵 번호의 범위는 0 ~ 15 입니다.
- ▶ MapMask1 : 축맵에 포함할 축들을 지정할 마스크 값(32BIT Bit0 ~ Bit31). 이값의 Bit0 ~ Bit31 을 이용하여 그룹에 포함할 축들을 지정합니다. 각 비트의 값이 0 이면 해당 축(비트의 순서와 일치하는 축)은 배제되는 것이며, 1 이면 해당 축이 포함되는 것이다. 이 매개변수의 각 비트별 정보는 아래 표와 같다.(아래 표는 8 축을 사용하는 경우를 기준으로 하였다)

Bit Number	Meaning
BIT0	0 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT1	1 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT2	2 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT3	3 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT4	4 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT5	5 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT6	6 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)
BIT7	7 번 축의 포함 여부 (0 : 포함안함, 1 : 포함)

▶ MapMask2: 이 매개변수는 보간 제어시 32축 이상을 사용하고자 할 때 사용하는 매개변수이다. MapMask1 에 포함되지 못한 축들은 이 매개변수를 사용하여 지정합니다.

▶ IxMode: 보간모드 값 입니다.

Value	Meaning
0 or ecmIX_MODE_LINEAR	직선 보간
1 or ecmIX_MODE_CIRCULAR	원호보간
2 or ecmIX_MODE_HELICARL	헬리컬 보간(미구현)
3 or ecmIX_MODE_SPLINE	스플라인 보간(미구현)

▶ ErrCode: 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 함수의 cmdidx 가 반환됩니다.

EXAMPLE

X 축과 Y 축의 직선보간

C/C++:

```
// 맵번호 설정
#define MAP0 0

#define AXISX 0
#define AXISY 1

ecmIxCfg_MapAxes (Device_Info.NetIdx, MAP0, 0x3, 0, ecmIX_MODE_LINEAR , &Error_Num);
// 또는 ecmIxCfg_MapAxes (Device_Info, 0, AXISX | AXISY, 0, ecmIX_MODE_LINEAR ,
&Error_Num); //
```

NAME	INFORMATION
ecmlxCfg_UnmapAxes ecmlxCfg_UnmapAxes_FF - 보간 대상 축 그룹 해제	 Interpolation Function <hr/>  VC++/BCB/.NET <hr/>  Level 1 <hr/>  위험 요소 없음

SYNOPSIS

- t_cmdidx ecmlxCfg_UnmapAxes([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)
- t_cmdidx ecmlxCfg_UnmapAxes_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 보간 작업을 수행하는 맵을 해제합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호. 맵 번호의 범위는 0 ~ 15 입니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 함수의 cmdidx 가 반환됩니다.

NAME

ecmlxCfg_SetSpeedPatt


ecmlxCfg_SetSpeedPatt_FF

ecmlxCfg_GetSpeedPatt


- 보간 이송 속도 설정 및 반환

INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

❑ t_cmdidx ecmlxCfg_SetSpeedPatt([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsVectSpeed, [in] t_i32 SpeedMode, [in] t_f64 VIni, [in] t_f64 VEnd, [in] t_f64 VWork, [in] t_f64 Acc, [in] t_f64 Dec, [out] t_i32 *ErrCode)

❑ t_cmdidx ecmlxCfg_SetSpeedPatt_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsVectSpeed, [in] t_i32 SpeedMode, [in] t_f64 VIni, [in] t_f64 VEnd, [in] t_f64 VWork, [in] t_f64 Acc, [in] t_f64 Dec, [out] t_i32 *ErrCode)

❑ t_success ecmlxCfg_GetSpeedPatt([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *IsVectSpeed, [out] t_i32 *SpeedMode, [out] t_f64 *VIni, [out] t_f64 *VEnd, [out] t_f64 *VWork, [out] t_f64 *Acc, [out] t_f64 *Dec, [out] t_i32 *ErrCode)

DESCRIPTION

ecmlxCfg_SetSpeedPatt, ecmlxCfg_SetSpeedPatt_FF 는 “기본보간제어”의 이송 속도에 대한 환경설정을 정의 합니다. 사용자가 지정한 작업 속도는 “IsVectSpeed”의 설정값이 “TRUE”이면 벡터 스피드, “FALSE”이면 마스터 스피드가 적용됩니다. 보간 제어는 보간 작업속도를 벡터속도로 설정해야만 하는 특별한 경우를 제외하고는 보간 작업 속도를 마스터속도로 설정하는 것이 모터의 최대속도를 활용하는데 있어서 편리합니다. ecmlxCfg_GetSpeedPatt 은 “기본보간제어”의 이송속도에 대한 설정된 값을 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ IsVectSpeed : 보간 작업 속도의 설정 값 및 반환 값입니다. 아래와 같은 값을 가집니다.

Value	Meaning
0	Master Speed Mode
1	Vector Speed Mode

▶ SpeedMode : 속도모드의 설정 값 및 반환 값입니다. 아래와 같은 값을 가집니다.

Value	Meaning
0 or ecmSMODE_CONST	CONSTANT :가감속을 수행하지 않습니다.
1 or ecmSMODE_TRAPE	TRAPEZOIDAL : 사다리꼴 가감속을 수행합니다.
2 or ecmSMODE_SCURVE	S-CURVE : S-CURVE 가감속을 수행합니다.

▶ VIni : 초기 속도에 대한 설정 및 반환 값입니다.

▶ VEnd : 이송 완료 시 속도에 대한 설정 및 반환 값입니다.

▶ VWork : 작업 속도에 대한 설정 및 반환 값입니다.

▶ Acc : 가속도에 대한 설정 및 반환 값입니다.

▶ Dec : 감속도에 대한 설정 및 반환 값입니다.

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE





* ecmIxCfg_SetSpeedPatt(), ecmIxCfg_SetSpeedPatt()_FF : 이 리턴값은 함수의 cmdidx 를 반환합니다.

* ecmIxCfg_GetSpeedPatt() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	설정값 반환에 실패 했음을 의미합니다.
1	설정값 반환에 성공 했음을 의미합니다.

EXAMPLE

* ecmIxCfg_LineStart EXAMPLE 참고

NAME	I N F O R M A T I O N
ecmlxCfg_SetJerkRatio	 Interpolation Function
ecmlxCfg_SetJerkRatio_FF	 VC++/BCB/.NET
ecmlxCfg_GetJerkRatio	 Level 1
- 보간 Jerk 시간 비율 설정 및 반환	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmlxCfg_SetJerkRatio([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode)
- t_cmdidx ecmlxCfg_SetJerkRatio_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode)
- t_f64 ecmlxCfg_GetJerkRatio([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)

DESCRIPTION

지정된 보간 맵이 S-CURVE 속도 프로파일을 사용하는 경우에 가속/감속 구간의 시간에 대해서 Jerk 가 0 이 아닌 구간의 시간 비율을 설정 및 반환합니다.

PARAMETER

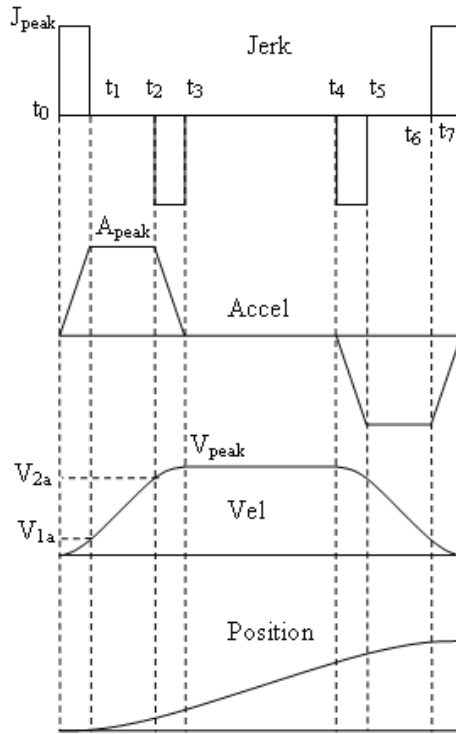
- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ JerkTimeRatio : 가속구간 또는 감속구간에서 Jerk 가 0 이 아닌 구간의 시간 비율을 1.0 기준으로 설정 및 반환합니다(자세한 내용은 “REFERENCE” 항목 참고). 여기에 설정할 수 있는 값의 범위는 0.0 ~ 1.0 입니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * emcIxCfg_SetJerkRatio(), ecmIxCfg_SetJerkRatio() : 이 리턴값은 함수의 cmdidx 를 반환합니다.
- * emcIxCfg_GetJerkRatio() : 이 리턴값은 JerkRatio 값을 반환합니다.





REFERENCE

□ 저크(Jerk), 가속도, 속도, 위치의 관계는 아래의 그래프와 같다. 여기서 시간이 $t_0 \sim t_1$ 인 구간과 $t_2 \sim t_3$ 인 구간은 Jerk 가 0 이 아닌 값을 가지는 구간이다. 그리고 $t_1 \sim t_2$ 인 구간은 Jerk 가 0 인 구간이다(Jerk 가 0 인 구간을 등가속구간이라 한다).



□ JerkTimeRatio 값은 전체 가속 시간 중에서 $t_0 \sim t_1$ 과 $t_2 \sim t_3$ 구간이 차지하는 비율을 나타낸다. 예를 들어서 등가속구간이($t_1 \sim t_2$)이 가속시간의 30%를 차지한다면 JerkTimeRatio 는 0.7 이 된다(왜냐하면 Jerk 가 0 이 아닌 구간은 1.0 에서 등가속 구간을 뺀 것이 되므로). 마찬가지로 등가속구간이 가속시간의 70%를 차지한다면 JerkTimeRatio 는 0.3 이 될 것이다.

□ JerkTimeRatio 값이 0 이면 사다리꼴 속도패턴과 같아지게 되며, JerkTimeRatio 가 1 이면 완전한 S-Curve 형태가 된다. JerkTimeRatio 의 기본값은 0.66 이다.

NAME	INFORMATION
ecmlxMot_LineStart	 Interpolation Function
ecmlxMot_LineStart_FF	 VC++/BCB/.NET
ecmlxMot_Line	 Level 1
ecmlxMot_Line_NB - 직선 보간 상대 좌표 이송	 위험 요소 없음

SYNOPSIS

- ▣ t_cmdidx ecmlxMot_LineStart([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode)
- ▣ t_cmdidx ecmlxMot_LineStart_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode)
- ▣ t_success ecmlxMot_Line([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode)
- ▣ t_success ecmlxMot_Line_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode)

DESCRIPTION

현재 위치로부터의 상대 좌표로의 직선 보간 이동을 수행합니다.
 ecmlxMot_LineStart, ecmlxMot_LineStart_FF 는 모션을 시작시킨 후에 바로 반환되고,
 ecmlxMot_Line, ecmlxMot_Line_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다.
 그리고 ecmlxMot_Line_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블로킹(Blocking)
 하지 않습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ DistList : 현재 위치로부터의 상대적인 이동 좌표값(각 축의 이동 거리값)의 배열 주소. 이 배열의 크기는 맵핑된 축의 수와 일치하여야 합니다. 거리의 단위는 논리적 거리를 적용합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmIxMot_LineStart(), ecmIxMot_LineStart_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

* ecmIxMot_Line(), ecmIxMot_Line_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	직선보간 이송이 실패 했음을 의미합니다.
1	직선보간 이송이 성공 했음을 의미합니다.

EXAMPLE

```

C/C++

#include "ComiEcatSdk_Api.h"

#define AXISX      0
#define AXISY      1

#define MAP0      0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* OnProgramInitial : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
*****/
void OnProgramInitial()
{
    TEcDevInfo Device_Info; //디바이스 정보 저장 할 구조체
    t_i32 Device_Num = 0;

    t_success nIsLoaded = ecDll_Load ();

    if(!ecGn_LoadDevices(&Error_Num)){ //장치 로드
        //로드 실패시 예외 처리
    }

    if(!ecGn_GetDevInfo(Device_Num,&Device_Info, &Error_Num)){
        //디바이스 정보 로드 실패 시 예외 처리
    }

    ecNet_SetAlState(Device_Info.NetIdx, ecAL_STATE_OP, &Error_Num);
    //ALState OP 모드로 전환

    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXISX, &Error_Num);
    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXISY, &Error_Num);
}

/*****
* OnSetSpeed : 이 함수는 속도설정의 변경이 필요할 때
* 호출되는 가상의 함수 입니다.
*****/

void OnSetSpeed()

```

```

{
    ecmIxCfg_MapAxes (Device_Info.NetIdx , MAP0, AXISX | AXISY, 0, ecmIX_MODE_LINEAR ,
    &Error_Num);
    //보간 그룹에 축을 설정하고 직선 보간 모드로 설정
    ecmIxCfg_SetSpeedPatt( Device_Info.NetIdx , MAP0, 0, 0, 0, 40000, 10000, 10000,
    &Error_Num);
    //보간 그룹을 Master Speed Mode 로 설정하고 초기속도, 종료속도, 작업 속도, 가속도,
    감속도 순으로 설정
}
/*****
* DoMotion : 작업명령시에 호출되는 가상의 함수 입니다.
* 이 함수는 X, Y 축에 대하여 상대 좌표 (1000, 2000)만큼 직선 보간 이동합니다.
* *****/
void DoMotion()
{
    double fDistList[2] = {1000, 2000}; //각축의 이동할 거리입니다.

    ecmIxCfg_LineStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);
}

```

NAME	INFORMATION
ecmlxMot_LineToStart	 Interpolation Function
ecmlxMot_LineToStart_FF	 VC++/BCB/.NET
ecmlxMot_LineTo	 Level 1
ecmlxMot_LineTo_NB	 위험 요소 없음
- 직선 보간 절대 좌표 이송	

SYNOPSIS

- ❑ t_cmdidx ecmlxMot_LineToStart([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)
- ❑ t_cmdidx ecmlxMot_LineToStart_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)
- ❑ t_success ecmlxMot_LineTo ([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)
- ❑ t_success ecmlxMot_LineTo_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)

DESCRIPTION

절대 좌표로의 직선 보간 이동을 수행합니다.
 ecmlxMot_LineToStart, ecmlxMot_LineToStart_FF 는 모션을 시작시킨 후에 바로 반환되고,
 ecmlxMot_LineTo, ecmlxMot_LineTo_NB 는 모션이 완료되기 전까지 함수가 반환되지
 않습니다. 그리고 ecmlxMot_LineTo_NB 는 함수가 반환되지 않지만 윈도우 메시지를
 블록킹(Blocking) 하지 않습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ PosList : 이동할 목표 절대 좌표값(각 축의 절대좌표값). 이 배열의 크기는 맵핑된 축의 수와 일치하여야 합니다.거리의 단위는 논리적 거리를 적용합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

EXAMPLE

```

C/C++

#include "ComiEcatSdk_Api.h"

#define AXISX      0
#define AXISY 1

#define MAP0      0

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수
/*****
* 세팅 및 프로그램 초기화 부분은 생략 (ecmIxMot_LineStart 예제참고)
* *****/

/*****
* DoMotion T 작업명령시에 호출되는 가상의 함수 입니다.
* 이 함수는 X, Y 축에 대하여 절대 좌표(1000, 2000)만큼 직선 보간 이동합니다.
* *****/
void DoMotionT()
{
    double fDistList[2] = {1000, 2000}; //각축의 이동할 거리입니다.

    ecmIxMot_LineToStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);
}

```

RETURN VALUE

* ecmIxMot_LineToStart(), ecmIxMot_LineToStart_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

* ecmIxMot_LineTo(), ecmIxMot_LineTo_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	직선보간 이송이 실패 했음을 의미합니다.
1	직선보간 이송이 성공 했음을 의미합니다.

NAME	I N F O R M A T I O N
ecmlxMot_Stop	 Interpolation Function
ecmlxMot_Stop_FF	 VC++/BCB/.NET
- 보간 이송 정지	 Level 1
	 위험 요소 없음

SYNOPSIS

□ t_cmdidx ecmlxMot_Stop([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode)

□ t_cmdidx ecmlxMot_Stop_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 보간맵에 대한 보간 작업을 정지 합니다. IsDecStop 매개변수를 통해 감속정지 할것인지, 급정지 할것인지 설정합니다.

ecmlxMot_Stop 함수는 IsWaitCompt 매개변수를 통해 보간 작업이 완료될 때까지 대기할 것인지 대기 하지 않을 것인지를 설정합니다.

PARAMETER

▶ NetID : Network 번호

▶ MapIndex : 맵 번호

▶ IsDecStop : 정지 시 감속 정지, 급정지를 결정하는 매개변수입니다..

Value	Meaning
0	급 정지
1	감속 정지

▶ IsWaitCompt : ecmlxMot_Stop 의 매개변수로서, 보간 작업이 끝날때까지 대기할지 여부를 결정하는 매개변수 입니다.

Value	Meaning
0	대기하지 않음
1	대기함

▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 함수의 cmdidx 가 반환됩니다.

EXAMPLE





```

C/C++
//본 예제는 ecmMxMot_MoveStart 사용하여 두 축을 절대 좌표 5000 으로 이동하는 예 입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0
#define AXIS1 1
#define MAP0 0
#define MAP1 1
t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* 프로그램 앞 부분은 생략 (설정 부분 다 생략)
*****/

/*****
* DO_Stop: 작업 명령 시에 호출되는 가상의 함수 입니다.
*****/
void DO_Stop()
{
    ecmIxMot_Stop (Device_Info.NetIdx, MAP1, 0, 0, &Error_Num);
    //보간 그룹 MAP1 은 급정지, 작업이 끝날때까지 대기하지 않는 정지를 실행
}

```

<h2>NAME</h2> <p>ecmlxSt_IsBusy</p> <p>- 보간 모션 이송 확인</p>	INFORMATION
	 Interpolation Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

SYNOPSIS

t_bool ecmlxSt_IsBusy([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 지정한 보간맵에 해당하는 보간작업이 진행중인지를 확인합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	보간 이송 중이 아님을 의미합니다,
1	보간 이송 증임을 의미합니다.

EXAMPLE

```

C/C++
//본 예제는 ecmMxMot_MoveStart 사용하여 두 축을 절대 좌표 5000 으로 이동하는 예 입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0
#define AXIS1 1
#define Map0 0
t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* 프로그램 앞 부분은 생략
(속도 설정, 보간 그룹 설정 등)
    
```

```
***** /
/*****
* DO_Busy: 작업 명령 시에 호출되는 가상의 함수 입니다.
***** /
void DO_Busy()
{
    double fDistList[2];


    fDistList[0]=1000; fDistList[1]=0;
    ecmIxMot_LineStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);


    While(1)
    {
        if(lcemIxSt_IsBusy (Device_Info.NetIdx, Map0, &Error_Num)
        {
            //이송 중이 아니면 break;
            break;
        }
    }
}
```


NAME


ecmlxSt_WaitCompt
 ecmlxSt_WaitCompt_NB
 - 보간 모션 완료 대기

INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

```

  □ t_success ecmlxSt_WaitCompt([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32
  *ErrCode)
  □ t_success ecmlxSt_WaitCompt_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32
  *ErrCode)
  
```

DESCRIPTION

이 함수는 지정한 보간맵에 해당하는 보간작업이 완료될 때까지 기다립니다.
 ecmlxSt_WaitCompt_NB 는 완료 대기시 윈도우 메시지를 블로킹(Blocking) 하지 않습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	보간 이송 완료대기에 실패하였음을 의미합니다.
1	보간 이송 완료대기에 성공하였음을 의미합니다.

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"

#define Map0 0
t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수
  
```

```
* DO_Wait : 작업 명령 시에 호출되는 가상의 함수 입니다.  
* 보간 이송 완료를 대기 하는 함수 입니다.  
*****//  
void DO_Wait()  
{  
  
    if(!lcmSxSt_WaitCompt(Device_Info.NetIdx, Map0, &Error_Num)){  
        // Map0 의 보간 이송 완료를 대기  
        // 보간 이송 대기 실패 예외 처리  
        return;  
    }  
}
```

Homing Operation Functions

쥬커미조아는 ComiECAT 를 통해 다양한 최신 개발환경을 지원하기 위해 노력하고 있습니다. 본 장에서 다루지 않는 개발 환경을 이용하시는 고객(顧客)님께서서는 저희 (쥬커미조아)를 통해 문의해주시면 신속히 대처해 드리도록 하겠으며, 제공되는 라이브러리 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리를 사용할 수 있도록 지원하여 드립니다. 뛰어난 성능을 기반으로 한 ComiECAT 라이브러리의 즐거움을 이제 함께 하십시오.







11 Homing Operation Functions

11.1 함수 요약

Summary of Functions
<p>❑ t_cmdidx ecmHomeCfg_SetMode([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 HomeOpMode, [out] t_i32 *ErrCode) 지정된 축의 원점복귀 모드를 설정합니다.</p>
<p>❑ t_i32 ecmHomeCfg_GetMode([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 지정된 축의 현재 설정된 원점복귀 모드를 반환합니다.</p>
<p>❑ t_cmdidx ecmHomeCfg_SetOffset([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Offset, [out] t_i32 *ErrCode) 지정된 축의 원점 오프셋(Offset)을 설정합니다.</p>
<p>❑ t_f64 ecmHomeCfg_GetOffset([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 지정된 축의 현재 설정된 원점 오프셋(Offset)을 반환합니다.</p>
<p>❑ t_cmdidx ecmHomeCfg_SetSpeedPatt([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SpeedMode, [in] t_f64 Vel, [in] t_f64 Acc, [in] t_f64 Dec, [in] t_f64 HomeSpecVel, [out] t_i32 *ErrCode) 지정된 축의 원점복귀 속도 패턴을 설정합니다.</p>
<p>❑ t_success ecmHomeCfg_GetSpeedPatt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32* SpeedMode, [out] t_f64* Vel, [out] t_f64* Acc, [out] t_f64* Dec, [out] t_f64* HomeSpecVel, [out] t_i32 *ErrCode) 지정된 축의 현재 설정된 원점복귀 속도 패턴을 반환합니다.</p>
<p>❑ t_cmdidx ecmHomeMot_MoveStart([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Direction, [out] t_i32 *ErrCode) 지정된 축의 원점복귀 이송을 수행합니다.</p>
<p>❑ t_success ecmHomeMot_Move([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Direction, [out] t_i32 *ErrCode) 지정된 축의 원점복귀 이송을 수행합니다.</p>
<p>❑ t_bool ecmHomeSt_IsBusy([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 지정된 축에 대해 원점복귀가 수행중인지 확인합니다.</p>
<p>❑ t_success ecmHomeSt_WaitCompt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 지정된 축의 원점복귀 완료시까지 대기합니다.</p>
<p>❑ t_word ecmHomeSt_GetFlags([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 원점복귀에 관련된 상태 Flag 값을 반환합니다.</p>

11.2 함수 설명

NAME	INFORMATION
ecmHomeCfg_SetMode ecmHomeCfg_GetMode - 원점복귀 모드 설정 및 반환	 Home Operation Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmHomeCfg_SetMode([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 HomeOpMode, [out] t_i32 *ErrCode)
- t_i32 ecmHomeCfg_GetMode([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 지정된 축의 원점복귀 모드를 설정합니다. 원점복귀 모드는 크게 두가지 종류로 나뉠수 있는데 하나는 서보드라이버에서 직접 제어하는 원점복귀 모드이며, 다른 하나는 마스터 장치에서 직접 제어하는 원점복귀 모드 입니다.

서보드라이버에서 직접 제어하는 원점 복귀 모드는 1 번부터 37 번까지 있습니다. 이 종류의 원점복귀 모드를 수행할 때에는 마스터 장치는 서보드라이버에게 원점복귀 시작 명령만을 내리며 원점복귀를 수행하는 과정에서는 마스터 장치가 제어에 관여하지 않습니다.

마스터장치에서 직접 제어하는 원점복귀 모드는 101 번부터 110 번까지 있습니다. 이 종류의 원점복귀 모드를 수행할 때에는 서보드라이버는 Cyclic Position 제어모드로 동작하며 매 사이클마다 모터의 위치를 마스터가 직접 제어하여 이송을 수행합니다. 각 모드의 조건에 따라서 원점복귀가 완료됨을 마스터가 직접 판단하고 원점복귀 작업을 완료합니다. 이때 원점의 위치를 정확히 래치하기 위해서 “Touch Probe Function”을 이용하므로 원점센서(모드에 따라서 리미트 센서가 원점센서 대치)의 신호가 Touch Probe 신호 입력으로 연결되어 있어야 합니다. 마스터장치 직접 제어 모드에서도 자동 원점 탈출 기능을 제공합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ HomeMode : ecmHomeCfg_SetMode 의 매개변수이며, 원점복귀 모드에 대한 내용은 “REFERENCE”를 참고 하시기 바랍니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmHomeCfg_SetMode() : 이 리턴값은 함수의 cmdidx 를 반환합니다.

* ecmHomeCfg_GetMode() : 이 리턴값은 원점복귀 모드 값을 반환합니다.

REFERENCE

- 서보드라이버에서 제어하는 원점복귀모드





Value	Meaning
1	-EL & Index Pulse
2	+EL & Index Pulse
3	(+)ORG & Index Pulse direction reversal
4	(+)ORG & Index Pulse no direction change
5	(-)ORG & Index Pulse direction reversal
6	(-)ORG & Index Pulse no direction change
7	on (+)ORG -Index Pulse
8	on (+)ORG +Index Pulse
9	After (+)ORG reverse +Index Pulse
10	After (+)ORG +Index Pulse
11	on (-)ORG -Index Pulse
12	on (-)ORG +Index Pulse
13	After (-)ORG reverse +Index Pulse
14	After (-)ORG +Index Pulse
17	Same as 1 without Index Pulse
18	Same as 2 without Index Pulse
19	Same as 3 without Index Pulse
20	Same as 4 without Index Pulse
21	Same as 5 without Index Pulse
22	Same as 6 without Index Pulse
23	Same as 7 without Index Pulse
24	Same as 8 without Index Pulse
25	Same as 9 without Index Pulse
26	Same as 10 without Index Pulse
27	Same as 11 without Index Pulse
28	Same as 12 without Index Pulse
29	Same as 13 without Index Pulse
30	Same as 14 without Index Pulse
33	On Index Pulse (+) Direction
34	On Index Pulse (-) Direction
35	Current Position = Home
37	Current Position = Home

□ 마스터 장치에서 제어하는 원점복귀모드

Value	Meaning
101	ORG ON > STOP
102	ORG ON > STOP > FWD > Z-Pulse ON > STOP
103	ORG ON > STOP > BWD > Z-Pulse ON > STOP
104	ORG ON > STOP > BWD > ORG OFF > STOP > FWD(HomeSpec) > ORG ON > STOP
105	ORG ON > STOP > BWD > ORG OFF > STOP > FWD(HomeSpec) > ORG ON > STOP > FWD(HS) > Z-Pulse ON > STOP
106	ORG ON > STOP > BWD > ORG OFF > STOP > FWD(HomeSpec) > ORG ON > STOP > BWD(HS) > Z-Pulse ON > STOP
107	EL ON > STOP > BWD(HS) > EL OFF > STOP
108	EL ON > STOP > BWD(HS) > EL OFF > STOP > BWD(HS) > Z-Pulse ON > STOP
109	EL ON > STOP > BWD(HS) > EL OFF > STOP > FWD(HS) > EL ON > STOP
110	EL ON > STOP > BWD(HS) > EL OFF > STOP > FWD(HS) > EL ON > STOP > BWD > Z-Pulse ON > STOP

EXAMPLE

* ecmHomeMot_MoveStart EXAMPLE 참고

NAME	INFORMATION
ecmHomeCfg_SetOffset	 Home Operation Function
ecmHomeCfg_GetOffset	 VC++/BCB/.NET
- 원점복귀 Offset 설정 및 반환	 Level 1
	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmHomeCfg_SetOffset([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 Offset, [out] t_i32 *ErrCode)
- t_f64 ecmHomeCfg_GetOffset([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 원점 복귀 위치에서 일정거리 이상을 상대 이송할 필요가 있을 경우 그에 대한 Offset 을 설정 및 반환합니다. 원점복귀 종료 위치를 기준으로 추가 모션 이송을 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ Offset : 원점 복귀 위치에서 이송할 거리 값. 이 값의 단위는 논리적 거리 입니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- * ecmHomeCfg_SetOffset : 이 리턴값은 함수의 cmdidx 가 반환됩니다.
- * ecmHomeCfg_GetOffset : 이 리턴값은 설정된 Offset 값이 반환됩니다.

EXAMPLE

- * ecmHomeMot_MoveStart EXAMPLE 참고

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmHomeCfg_SetSpeedPatt</p> <p style="margin: 0;">ecmHomeCfg_GetSpeedPatt</p> <p style="margin: 0;">- 원점복귀 속도 설정 및 반환</p>	<h3 style="margin: 0;">I N F O R M A T I O N</h3>
	<ul style="list-style-type: none"> Home Operation Function VC++/BCB/.NET Level 1 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmHomeCfg_SetSpeedPatt([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SpeedMode, [in] t_f64 Vel, [in] t_f64 Acc, [in] t_f64 Dec, [in] t_f64 HomeSpecVel, [out] t_i32 *ErrCode)
- t_success ecmHomeCfg_GetSpeedPatt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32* SpeedMode, [out] t_f64* Vel, [out] t_f64* Acc, [out] t_f64* Dec, [out] t_f64* HomeSpecVel, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 지정된 축의 원점복귀 속도 모드와 가감속도 및 작업속도 HomeSpec 속도등을 설정 및 반환합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ SpeedMode : 속도모드의 설정 값 및 반환 값입니다. 아래와 같은 값을 가집니다.

Value	Meaning
0 or ecmSMODE_CONST	CONSTANT : 가감속을 수행하지 않습니다.
1 or ecmSMODE_TRAPE	TRAPEZOIDAL : 사다리꼴 가감속을 수행합니다.
2 or ecmSMODE_SCURVE	S-CURVE : S-CURVE 가감속을 수행합니다.

- ▶ Vel : 작업 속도에 대한 설정 및 반환 값입니다.
- ▶ Acc : 가속도에 대한 설정 및 반환 값입니다.
- ▶ Dec : 감속도에 대한 설정 및 반환 값입니다.
- ▶ HomeSpecVel : HomeSpec 속도의 설정 및 반환 값입니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* ecmHomeCfg_SetSpeedPatt() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

* ecmHomeCfg_GetSpeedPatt() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	설정 값 반환에 실패하였음을 의미합니다.
1	설정 값 반환에 성공하였음을 의미합니다.

EXAMPLE

* ecmHomeMot_MoveStart EXAMPLE 참고

NAME	I N F O R M A T I O N
ecmHomeMot_MoveStart	 Home Operation Function
ecmHomeMot_MoveStart_FF	 VC++/BCB/.NET
ecmHomeMot_Move	 Level 1
- 단축 원점복귀 이송	 위험 요소 없음

SYNOPSIS

- t_cmdidx ecmHomeMot_MoveStart([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Direction, [out] t_i32 *ErrCode)
- t_cmdidx ecmHomeMot_MoveStart_FF([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Direction, [out] t_i32 *ErrCode)
- t_cmdidx ecmHomeMot_Move([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 Direction, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 원점복귀 작업을 수행합니다. ecmHomeMot_MoveStart, ecmHomeMot_MoveStart_FF 함수는 모션을 시작 시킨 후 바로 반환되며, ecmHomeMot_Move 함수는 모션이 완료되지 전까지 반환되지 않습니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ Direction : 원점 복귀 모션을 수행할 방향을 지정합니다.

Value	Meaning
0 or ecmDIR_N	(-) 방향
1 or ecmDIR_P	(+) 방향

- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

*: 이 리턴값은 함수의 cmdidx 가 반환됩니다.

EXAMPLE

본 예제는 ecmHomeMot_MoveStart () 함수를 이용하여 x,y 축 의 원점복귀를 수행하는 함수입니다. 단 원점복귀에 대한 환경설정은 이미 이루어진 것으로 가정합니다.

```

C/C++
#include "ComiEcatSdk_Api.h"
#define AXISX 0
#define AXISY 1

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* OnProgramInitial : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다. //5~9 장 사이 내용 참고
*****/
void OnProgramInitial()
{
    TEcDevInfo Device_Info; //디바이스 정보 저장 할 구조체
    t_i32 Device_Num = 0;

    t_success nIsLoaded = ecDll_Load ();

    if(!ecGn_LoadDevices(&Error_Num)){ //장치 로드
        //로드 실패시 예외 처리
    }

    if(!ecGn_GetDevInfo(Device_Num,&Device_Info, &Error_Num)){
        //디바이스 정보 로드 실패 시 예외 처리
    }

    ecNet_SetAlState(Device_Info.NetIdx, ecAL_STATE_OP, &Error_Num);
    //ALState OP 모드로 전환

    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXISX, &Error_Num);
    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXISY, &Error_Num);

} OnProgramInitial 함수 끝

/*****
* OnHomeSetSpeed : 이 함수는 속도설정의 변경이 필요할 때
* 호출되는 가상의 함수 입니다. 이때 m_fVwork, m_fAcc, m_fDec, m_sPecVel 변수를
* 통하여 속도, 가속도, 감속도, 홈리턴 속도 값이 적절하게 전달된다고 가정합니다.
*****/
void OnHomeSetSpeed()
{
    ecmHomeCfg_SetMode(Device_Info.NetIdx, AXISX, 7,&Error_Num);
    //X 축의 홈복귀 모드를 설정합니다. (함수 설명에 Rererence 참고)
    ecmHomeCfg_SetMode(Device_Info.NetIdx, AXISY, 7,&Error_Num);
    //Y 축의 홈복귀 모드를 설정합니다. (함수 설명에 Rererence 참고)
    ecmHomeCfg_SetOffset(Device_Info.NetIdx, AXISX, 1000, &Error_Num);
    //X 축의 홈복귀 추가 이동 거리를 설정합니다.
    ecmHomeCfg_SetOffset(Device_Info.NetIdx, AXISY, 1000, &Error_Num);
    //Y 축의 홈복귀 추가 이동 거리를 설정합니다.
    ecmHomeCfg_SetSpeedPatt(Device_Info.NetIdx, AXISX, ecmSMODE_SCURVE, m_fVworkx,
m_fAccx, m_fDecx, m_sPecVel, &Error_Num);

```

```

    //X 축 가감속을 수행 모드로 설정 하고 속도, 가속도, 감속도 설정
    ecmHomeCfg_SetSpeedPatt(Device_Info.NetIdx, AXISY, ecmSMODE_SCURVE, m_fVworky
    m_fAccy, m_fDecy, m_sPecVel, &Error_Num);
    //Y 축 가감속을 수행 모드로 설정 하고 속도, 가속도, 감속도 설정

}

/*****
* OnHomeReturn : 이 함수는 가상의 함수로서 원점복귀를 실행합니다.
*****/
void OnHomeRetrun()
{
    t_i32 Direction = ecmDIR_P; // +방향으로 원점 복귀 실행

    ecmHomeMot_MoveStart(Device_Info.NetIdx, AXISX, Direction, &Error_Num) ;
    // X 축 원점복귀 시작
    ecmHomeMot_MoveStart(Device_Info.NetIdx, AXISY, Direction, &Error_Num) ;
    // Y 축 원점복귀 시작

    if(!ecmHomeSt_WaitCompt (Device_Info.NetIdx, AXISX, &Error_Num)){
        //X 축이 원점 복귀 하는 동안 블로킹하는 함수
        // 복귀 완료 대기 실패 예외 처리
        return;
    }

    if(!ecmHomeSt_WaitCompt (Device_Info.NetIdx, AXISY, &Error_Num)){
        //Y 축이 원점 복귀 하는동안 블로킹하는 함수
        // 복귀 완료 대기 실패 예외 처리
        return;
    }

}

```

<h2 style="margin: 0;">NAME</h2> <p style="margin: 10px 0;">ecmHomeSt_IsBusy</p> <p style="margin: 0 0 10px 20px;">- 원점복귀 모션 이송 확인</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: left; padding: 2px;">INFORMATION</th> </tr> <tr> <td style="padding: 2px;"> Home Operation Function</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> VC++/BCB/.NET</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> Level 1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> 위험 요소 없음</td> <td style="padding: 2px;"></td> </tr> </table>	INFORMATION		Home Operation Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Home Operation Function											
VC++/BCB/.NET											
Level 1											
위험 요소 없음											

SYNOPSIS

□ t_bool ecmHomeSt_IsBusy([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 단일 축에 대하여 원점복귀 모션 이송중인지를 확인합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드(음수) 또는 현재의 모션 상태(양수)를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	이송 중이 아님을 의미합니다,
1	이송 중임을 의미합니다.

EXAMPLE

```

C/C++
//본 예제는 ecmMxMot_MoveStart 사용하여 두 축을 절대 좌표 5000 으로 이동하는 예 입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0

t_i32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* 프로그램 앞 부분은 생략
(설정 부분 다 생략)
*****/
/*****
* DO_Busy: 작업 명령 시에 호출되는 가상의 함수 입니다.
*****/
    
```


```
*****/  
void DO_Busy()  
{  
    ecmHomeMot_MoveStart(Device_Info.NetIdx, AXISX, ecmDIR_P, &Error_Num) ;  
    While(1)  
    {  
        if(!ecmHomeSt_IsBusy (Device_Info.NetIdx, AXIS0, &Error_Num)  
        {  
            //이송 중이 아니면 break;  
            break;  
        }  
    }  
}
```


NAME


ecmHomeSt_WaitCompt
- 원점복귀 모션 완료 대기

INFORMATION

 Home Operation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

SYNOPSIS

□ t_success ecmHomeSt_WaitCompt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 단일 축에 대하여 원점복귀 모션이 완료될 때까지 대기 합니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	원점복귀 모션 완료 대기에 실패하였음을 의미합니다
1	원점복귀 모션 완료 대기에 성공하였음을 의미합니다.

EXAMPLE

* ecmHomeMot_MoveStart EXAMPLE 참고

<h2 style="margin: 0;">NAME</h2> <p style="margin: 10px 0;">ecmHomeSt_GetFlags</p> <p style="margin: 0 0 10px 20px;">- 원점복귀 상태 Flag 값 반환</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: left; padding: 2px;">INFORMATION</th> </tr> <tr> <td style="padding: 2px;"> Home Operation Function</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> VC++/BCB/.NET</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> Level 1</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;"> 위험 요소 없음</td> <td style="padding: 2px;"></td> </tr> </table>	INFORMATION		Home Operation Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Home Operation Function											
VC++/BCB/.NET											
Level 1											
위험 요소 없음											

SYNOPSIS

□ t_success ecmHomeSt_GetFlags([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)

DESCRIPTION

이 함수는 단일 축에 대해서 원점복귀 상태에 관련된 상태 Flag 값들이 반환됩니다.

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

* 이 리턴값은 SDK 에 정의된 TEcmHomeSt_Flags 구조체를 이용하여 받아올 수 있습니다. 구조체 안 매개 변수는 아래와 같습니다.

Parameter	Value & Meaning
HomeBusy	0: Homing 동작이 진행하지 않음을 나타냄 1: Homing 동작이 진행중임을 나타냄
Moving	0: 모터가 구동하지 않음을 나타냄 1: 모터가 구동중임을 나타냄
HomeAttained	0: Homing 동작이 실패했음을 나타냄 1: Homing 동작이 성공적으로 완료되었음을 나타냄
HomingError	0: 에러가 발생하지 않음을 나타냄 1: Homing 동작 중 에러가 발생했음을 나타냄

EXAMPLE

본 예제는 Set_HomeFlag () 함수를 이용하여 원점 복귀 관련 Flag 를 얻어오고 그 에 관련된 설명을 하는 예제입니다..

```
C/C++
#include "ComiEcatSdk_Api.h"
#define AXISX 0
#define AXISY 1

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

void Set_HomeFlag()
{
    TEcmHomeSt_Flags MotHomeFlg;    //원점 복귀 관련 구조체

    MotHomeFlg.word = ecmHomeSt_GetFlags(Device_Info.NetIdx , AXISX, &Error_Num);

    if(MotHomeFlg.bit.HomeBusy)
        //Homing 동작이 진행 중 일 때 처리
    else
        //Homing 동작이 진행 하지 않을 때 처리

    if(MotHomeFlg.bit.Moving)
        //모터가 구동 중 일 때 처리
    else
        //모터가 구동 중이 아닐 때 처리

    if(MotHomeFlg.bit.HomeAttained)
        //Homing 동작이 성공적으로 완료되었을 때 처리
    else
        //Homing 동작이 실패했을 때 처리

    if(MotHomeFlg.bit.HomingError)
        //Homing 동작 중 에러가 발생 했을 때 처리
    else
        //에러가 발생 하지 않았을 때 처리

}
```

TEST & MEASUREMENT & AUTOMATION / COMIZOA

ComiECAT Manual

저작권자 : **㈜커미조아**

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

2013년 07월 22일 1판 인쇄

매뉴얼 자료 번호 : 5.0.1



㈜커미조아

<http://www.comizoa.com>

Tel) 042 - 936 - 6500~6

Fax) 042 - 936 - 6507

이 사용자 설명서 상의 삽입된 삽화 및 예제 프로그램을 포함한 전체 내용은 대한민국 저작권법에 의해 보호되고 있습니다.
㈜커미조아의 사전 서면 동의 없이 사용자 설명서의 일부 또는 전체를 어떤 형태로든 복사, 전재할 수 없습니다.