

# **COMIZOA EtherCAT Software Development Kit**

## **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](mailto:csteam@comizoa.com)

For documentation suggestions, corrections, or requests, contact [tech@comizoa.com](mailto:tech@comizoa.com)

---

### 고객(顧客) 기술 지원

전자 우편 : [csteam@comizoa.com](mailto:csteam@comizoa.com)

파일 서버 : [ftp.comizoa.com](ftp://ftp.comizoa.com)

웹 사이트 : <http://www.comizoa.com>

본사 안내

대전광역시 유성구 테크노 2 로 314 번지 ( 탑립동 914 번지 )

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

<b>Trademarks</b>	2-2
<b>Tble of Contents</b>	2-3
<b>Introduction</b>	7
<b>1 ComiECAT 사전 안내 사항</b>	8
<b>1.1 Overview</b>	8
1.1.1 제품 보증 안내	8
1.1.2 제품 보증 규정	8
1.1.3 저작권	8
1.1.4 상표안내	8
1.1.5 주의사항	8
1.1.6 매뉴얼 용어 안내	9
1.1.7 매뉴얼 아이콘의 설명	9
<b>1.2 Features</b>	11
1.2.1 독립성	11
1.2.2 호환성	11
1.2.3 편의성	11
1.2.4 신뢰성	11
1.2.5 풍부한 예제와 신속한 기술 지원	11
<b>Before working with ComiECAT</b>	12
<b>2 EtherCAT &amp; ComiECAT</b>	13
<b>2.1 EtherCAT 소개</b>	13
<b>2.2 ComiECAT 라이브러리 주요 내용</b>	13
2.2.1 ComiECAT 개요 사항	13
2.2.2 함수 이름 규칙 가이드	13
<b>Development Environment for ComiECAT</b>	15
<b>3 개발 환경 별 ComiECAT 사용 안내</b>	16
<b>3.1 개발 환경 지원 안내</b>	16
<b>3.2 ComiECAT 구성</b>	17
3.2.1 HARDWARE Layer	17
3.2.2 HAL(Hardware Abstract Layer)	17
3.2.3 ComiECAT Layer (API Layer)	18
3.2.4 ComiECAT 인터페이스 구성 파일	18
<b>3.3 각 개발 환경 별 안내</b>	18
3.3.1 Visual C++ 6.x 개발자를 위한 안내	20
3.3.2 Visual C++ 10.x 개발자를 위한 안내	26
3.3.3 Borland C++ Builder 개발자를 위한 안내	32
3.3.4 Borland Delphi 개발자를 위한 안내	37
3.3.5 Visual Basic 개발자를 위한 안내	42

<i>ComiECAT Introduction</i>	47
<b>4 ComiECAT 소개</b>	48
<b>4.1 함수의 명명 규칙</b>	48
4.1.1 함수 이름의 접두어	48
4.1.2 함수 이름의 접미어	49
<b>4.2 데이터형 표기</b>	50
<i>DLL Functions</i>	52
<b>5 DLL Functions</b>	53
<b>5.1 함수 요약</b>	53
<b>5.2 함수 설명</b>	54
<i>General Functions</i>	58
<b>6 General Functions</b>	59
<b>6.1 함수 요약</b>	59
<b>6.2 함수 설명</b>	60
<i>General Utility Functions</i>	72
<b>7 General Utility Functions</b>	73
<b>7.1 함수 요약</b>	73
<b>7.2 함수 설명</b>	74
<i>Debug Logging Functions</i>	79
<b>8 Debug Logging Functions</b>	80
<b>8.1 함수 요약</b>	80
<b>8.2 함수 설명</b>	81
<i>Net Interface Functions</i>	86
<b>9 Net Interface Functions</b>	87
<b>9.1 함수 요약</b>	87
<b>9.2 함수 설명</b>	89
<i>Slave I/F Functions</i>	109
<b>10 Slave I/F Functions</b>	110
<b>10.1 함수 요약</b>	110
<b>10.2 함수 설명</b>	113
<i>Digital I/O Functions</i>	139
<b>11 Digital I/O Functions</b>	140
<b>11.1 함수 요약</b>	140

<b>11.2 함수 설명</b>	<b>143</b>
11.2.1 전역채널(Global Channel) 번호를 이용하는 함수군	147
11.2.2 지역채널(Local Channel) 번호를 이용하는 함수군	169
11.2.3 논리비트주소(Logic Bit Address) 관리 함수군	185
<b>Analog I/O Functions</b>	<b>192</b>
<b>12 Analog I/O Functions</b>	<b>193</b>
<b>12.1 함수 요약</b>	<b>193</b>
<b>12.2 함수 설명</b>	<b>194</b>
<b>Motion General Functions</b>	<b>209</b>
<b>13 Motion General Functions</b>	<b>210</b>
<b>13.1 함수 요약</b>	<b>210</b>
<b>13.2 함수 설명</b>	<b>211</b>
<b>Motion Single Axis Functions</b>	<b>219</b>
<b>14 Motion Single Axis Functions</b>	<b>220</b>
<b>14.1 함수 요약</b>	<b>220</b>
<b>14.2 함수 설명</b>	<b>231</b>
14.2.1 기본 환경 설정	231
14.2.2 상태 제어 명령	251
14.2.3 이송 명령	255
14.2.4 모션 상태 감시	267
14.2.5 1 차원 위치 보정 (1D Position Correction) 기능 환경 설정	289
14.2.6 2 차원 위치 보정 (2D Position Correction) 기능 환경 설정	298
14.2.7 링카운터(Ring Counter) 기능 환경 설정	312
14.2.8 외부정지(External Stop) 기능 환경 설정	318
14.2.9 SD(Slow Down) 기능	329
14.2.10 오토토크(Auto Torque) 기능 환경 설정	340
14.2.11 기타 환경 설정	350
<b>Motion Mutiple Axes Functions</b>	<b>386</b>
<b>15 Motion Multiple Axes Functions</b>	<b>387</b>
<b>15.1 함수 요약</b>	<b>387</b>
<b>15.2 함수 설명</b>	<b>388</b>
<b>Motion InterPolation Functions</b>	<b>402</b>
<b>16 Motion InterPolation Functions</b>	<b>403</b>
<b>16.1 함수 요약</b>	<b>403</b>
<b>16.2 함수 설명</b>	<b>409</b>
<b>Homing Operation Functions</b>	<b>474</b>
<b>17 Homing Operation Functions</b>	<b>475</b>
<b>17.1 함수 요약</b>	<b>475</b>

TABLE OF CONTENTS

<b>17.2</b> 함수 설명	477
<i>Motion Master/Slave Functions</i>	494
<b>18</b> <i>Motion Master/Slave Functions</i>	495
<b>18.1</b> 함수 요약	495
<b>18.2</b> 함수 설명	496
<i>Motion LIST Motion Functions</i>	503
<b>19</b> <i>Motion LIST Motion Functions</i>	504
<b>19.1</b> 함수 요약	504
<b>19.2</b> 함수 설명	506
<i>Motion PT Motion Functions</i>	527
<b>20</b> <i>Motion PT Motion Functions</i>	528
<b>20.1</b> 함수 요약	528
<b>20.2</b> 함수 설명	529
<i>Motion Motor Performance monitor Functions</i>	544
<b>21</b> <i>Motion Motor Performance Monitor Functions</i>	545
<b>21.1</b> 함수 요약	545
<b>21.2</b> 함수 설명	546
<i>ComiECAT Error Code</i>	560
<b>I</b> <i>ComiECAT Error Code 안내</i>	561
<i>Frequently Asked Questions</i>	563
<b>II</b> <i>Frequently Asked Questions (FAQ)</i>	564
<b>III</b> <i>Visual Studio 2005</i>	564
<b>IV</b> <i>Visual Basic</i>	566
<b>V</b> <i>Borland C++ Builder</i>	566
<i>Index of ComiECAT Functions</i>	572
<b>VI</b> <i>Index of ComiECAT Functions</i>	573
<b>VII</b> <i>Quick Reference to ComiECAT Functions</i>	573

# Introduction

본 장에서는 제품 보증안내를 비롯한 제품 보증 규정, 저작권, 상표안내, 주의사항을 설명하고 있습니다. 다양한 모션 제어 환경에서 보다 강력하고 빠른 모션 제어를 위해, 그리고 안정적인 모션을 위해 고객(顧客)님께서 선택하신 저희 (주)커미조아 제품은 이제 고객(顧客)님에게 큰 감사와 보답으로 이바지 하도록 하겠습니다.

**가** 장 안정적이고, 빠르고 정확한 모션, 그리고 고객(顧客) 중심에서의 제품의 완성을 추구하는 (주)커미조아는 고객(顧客)님들을 위해 언제나 최선과 정성을 다하는 자세로 지금 이 시간에도 보다 나은 제품 개발을 위해 성실과 열정을 바탕으로 제품에 꿈을 담고 있습니다.

먼저 고객(顧客)님들께, 국내 최고의 모션 기술력을 자랑하는 저희 (주)커미조아 제품을 선택하여 주신 여러분들께 다시 한번 감사의 말씀을 드리며, 본 장에서는 제품 보증안내, 규정, 저작권, 상표 안내에 대한 내용을 설명하고 있습니다. 본 장에서 설명드리는 내용은 ECAT MASTER 운용과는 별개의 내용이나, 제품의 보증과 규정 그리고 저작권에 대한 내용을 설명 드리고 있으므로, 반드시 습득하여 주시기 바랍니다.



# 1 ComiECAT 사전 안내 사항

## 1.1 Overview

### 1.1.1 제품 보증 안내

저희 (주)커미조아는 고객(顧客) 여러분들께 가장 안정된 소프트웨어와 하드웨어를 공급함으로써, 고객(顧客) 여러분들을 만족시켜드리는 것을 최우선의 목표로 하고 있습니다. 만약 구입하신 제품에 외관상의 하자, 동작이상 또는 불량이 발견되는 경우에는 언제든지 저희 (주)커미조아를 통해 문의(問議)해주시기를 바라며, 가까운 대리점 혹은 총판점을 통해 구입하신 경우에는 해당 구입처(購買處)를 통해 문의하시면, 더욱 빠른 기술 지원을 받으실 수 있습니다.

### 1.1.2 제품 보증 규정

구입하신 당사의 제품은 소비자의 과실 이외의 자체 결함 및 동작이상에 대해 2 년간 그 전체 혹은 일부에 대해서 보증하고 있습니다. 당사의 제품에 대한 자세한 제품 보증 규정은 별도로 관리되는 각 제품의 ‘제품 보증 규정’ 에 의거하며, 자세한 보증 규정을 알기 원하시는 경우 (주)커미조아 혹은 총판점(總販店) 및 대리점(代理店) 등 해당 구입처를 통해 문의해 주시기 바랍니다.

### 1.1.3 저작권

이 매뉴얼의 일부 혹은 전체를 무단복사, 복제, 전제하는 것은 대한민국 저작권법에 저촉됩니다.

### 1.1.4 상표안내

Windows 는 Microsoft Corp. 의 등록상표입니다.



Visual C++ 는 Microsoft Corp. 의 등록상표입니다.

Visual Basic 은 Microsoft Corp. 의 등록상표입니다.



C++ Builder 는 Borland Software Corp. 의 등록상표입니다.

Delphi 는 Borland Software Corp. 의 등록상표입니다.

이외의 상표는 각 회사의 등록상표입니다.

### 1.1.5 주의사항

(주)커미조아의 제품군에는 제품의 특성(特性)에 따라서 하드웨어 및 소프트웨어 기술 지원이 필요한 경우가 있습니다. 필요하신 경우 본사 혹은 총판 및 대리점을 통해 제품 구입 이전에 점검 또는 요청해주시기 바랍니다.

(주)커미조아의 소프트웨어 및 하드웨어 제품군은 제품 성능 향상을 위해 예고 없이 변경될 수 있습니다. 고객(顧客)님께서서는 본 매뉴얼을 읽기전에 하드웨어 및 소프트웨어의 최신 변경사항에 대한 정보를 (주)커미조아를 통해 요청하실 수 있습니다.

본 매뉴얼은 (주)커미조아 ComiECAT 에 대한 정보를 포함한, 실제 라이브러리를 통한 모션 사항에 대한 기반 설명을 포함하고 있습니다.

최신 내용 및 기술되지 않은 사항 혹은 누락된 사항은 (주)커미조아 제품군 구입시에 고객(顧客) 등록을 해주신 고객(顧客)님의 정보를 통해 안내해 드릴 예정이며, 기술 지원 요청시에 보다 자세하고 정확한 방법과 내용을 통해 도움 드릴 것을 약속드립니다.



본 매뉴얼에 시작하기 앞서, 본 장(Chapter)에서는 본 매뉴얼을 보다 정확하게 빠르게 이해(理解)하실 수 있도록 ComiECAT 의 전체 구성과 형식에 대해서 안내해 드리겠습니다.

### 1.1.6 매뉴얼 용어 안내

본 매뉴얼에서는 필요에 따라 매뉴얼에 기술된 내용을 설명하기 위해 함축된 의미의 용어나 현장 용어를 사용할 수도 있습니다. 최대한 보충 설명이 필요한 내용에 대해서는 해당 단원(Chapter)에서 설명하도록 하겠습니다.




매뉴얼 전체 범위에서 사용되는 범용적인 용어의 의미는 다음 표 1 매뉴얼 용어 정리 같이 미리 안내하여 드립니다.

명칭	의미
모터 (Motor)	서보모터를 비롯한 스텝 모터
서보팩 (Servo Pack)	서보 앰프
서보 드라이브 (Servo Drive)	서보모터와 서보앰프의 편성
스텝 드라이브 (Step Drive)	스텝모터와 스텝 모터 제어회로의 편성
서보 시스템 (Servo System)	서보 드라이브와 (주)커미조아 상위 제어를 조합한, 일련의 완성된 시스템
지령 위치 출력 신호 (Command Pulse)	(주)커미조아 모션 제어 제품에서 서보 팩 혹은 스텝 드라이브 측으로 출력하는 전기적 펄스(Pulse) 열 신호
실제 위치 입력 신호 (Feedback Pulse)	(주)커미조아 모션 제어 제품에서 ETHERCAT 네트워크를 통해 위치 검출기(Encoder) 에서 입력받는 전기적 펄스(Pulse) 열 신호
ETHERCAT NC (ETHERCAT Network Controller)	ETHERCAT 네트워크 제어를 위한 본 제품을 일컫는 명칭

표 1 매뉴얼 용어 정리

### 1.1.7 매뉴얼 아이콘의 설명

매뉴얼에서 안내되는 내용을 보다 신속하고 정확하게 알 수 있도록 하며, 그 의미가 바르게 전달 되고 이해(理解)를 돕기 위해 원하는 의미에서 아래와 같은 매뉴얼 아이콘을 사용하고 있습니다.

 <p>보충</p>	이해(理解)하기 어려운 용어나 보충(補充)이 필요한 용어, 해설 및 사전에 설명 없이 새로 나온 용어를 설명합니다.
 <p>안내</p>	고객님의 편의와 기능의 자세한 내용에 대해 부가적으로 안내되는 사항입니다.
 <p>주의</p>	이 동작이나 실행 함수(Function)에 있어서 그 동작의 주의를 요망하는 내용을 나타냅니다.


	<p>이 동작이나 실행 함수(Function)에 있어서 그 동작의 이상이나 문제가 발생할 경우 이 사항에 대해서 경고의 의미를 나타냅니다.</p>
---	---

표 2 매뉴얼 아이콘의 설명

## 1.2 Features

(주)커미조아의 ETHERCAT 라이브러리인 ComiECAT의 장점은 다음과 같습니다.

### 1.2.1 독립성

ComiECAT은 Microsoft社의 표준라이브러리인 DLL(Dynamic Link Library) 형태의 독립된 라이브러리 인터페이스를 제공하며, 라이브러리 자체의 독립된 장치 관리의 편의성을 제공하고 있으며, DLL 형태의 라이브러리 장점을 통해 유지 보수와 귀사의 제품 구현에 보다 간편하게 하고 신뢰성 있는 독립형 동적 연결 라이브러리를 제공합니다.

### 1.2.2 호환성

우수한 소프트웨어 개발 도구를 이용하여 전통적인 개발 방법보다 더 적은 시간과 비용으로 더 좋은 품질의 소프트웨어를 개발하는 방법을 이야기하는 최신 RAD(Rapid Application Development)를 지향하고 있으며, 이에 맞는 최신 소프트웨어 개발 환경을 지원하고 있습니다.

고객(顧客)님께서서는 언제나 ComiECAT를 통하여 귀사의 제품에 보다 신속하고 정확한 시스템을 구현하실 수 있습니다.

### 1.2.3 편의성

인터페이스 함수 명명 규칙의 통일화와 의사 코드 주제를 매뉴얼과 라이브러리 인자(Parameter)에 부각시켜, 보다 빠른 시간내에 숙련된 라이브러리 사용자로 만들어드립니다. 특히, ComiECAT의 모든 함수 명에는 의미적 명명 규칙을 내포하였으며, 이것은 분명 실무 개발 환경에서 많은 부분 이점으로 작용할 것입니다.

### 1.2.4 신뢰성

(주)커미조아의 라이브러리 제품군은 오랜 시간 산업현장에서의 현장 경험을 바탕으로 형성된 신뢰성 있는 제품군입니다. ‘부드럽고, 정확하고, 빠르고, 쉬움’ 모션제어 기능을 통해 모션의 기본에 충실하였으며, 각 동작에 대한 입력과 출력을 ComiECAT 인터페이스 전역에 걸쳐 사용하기 쉽도록 안내해 놓았습니다.

또한, 응용프로그램에 기반하지 않는 자체 디버그 기능을 바탕으로 응용프로그램에 의한 오류를 최단시간내에 해결할 수 있도록 합니다. 디버그 모드의 지원과 향상된 디버그 정보를 바탕으로 오류발생에 대한 원인을 신속히 분석하실 수 있도록 도움을 드리며, 저희 (주)커미조아는 고객(顧客)님께 항상 정직과 신뢰를 드릴 수 있도록 최선을 다할 것입니다.

### 1.2.5 풍부한 예제와 신속한 기술 지원

지금 이시간에도 (주)커미조아는 타사에 비해 보다 많은 개발 선상의 활용가능한 라이브러리 예제를 지원해드리려고 노력하고 있으며, 예제간 중복 코드와 라이브러리 구성 이외의 부분을 최소화한 예제로 빠른 시간내에 예제의 코드를 바로 적용시켜드릴 수 있도록 노력하고 있습니다.

저희 (주)커미조아는 최신 .NET Framework 상의 C # (Sharp) 부터, Visual Basic 및 현존하는 RAD(Rapid Application Development) 환경을 지원하는 ComiECAT으로 보다 향상된 고객(顧客) 지원과 기술 지원을 통해 고객(顧客)여러분들의 성원에 이바지하겠습니다.

# Before working with ComiECAT

정밀한 모션제어와 고속의 모션제어는 최상의 모션 제어가 반드시 갖춰야할 필수요건입니다. 안정적인 하드웨어와 더불어 고급 기능의 소프트웨어는 고객(顧客)여러분들의 최상의 응용 프로그램 구현을 돕고 있습니다. 이제 더 이상 고민하지 마십시오 ComiECAT 은 보다 견고하고 우수한 기능을 바탕으로 여러분들이 원하시는 기능을 보다 빠르고 정확하게 안내하여 드릴 것입니다.

**본** 장에서는 EtherCAT 에 대한 정의와 저희 (주)커미조아의 ComiECAT 에 대한 안내를 위한 내용으로 구성되어 있습니다. 보다 신속한 기술 지원과 빠른 라이브러리 기능 습득(習得)을 위해 구성(構成)되었으며, 사전 안내사항부터 함수 이름 규칙 가이드까지 모든 내용을 쉽고 빠르게 이해(理解)하실 수 있도록 최선을 다하였습니다.



## 2 EtherCAT & ComiECAT

본 장에서는 EtherCAT 에 대한 사항과 네트워크 제어를 자세히 안내해 드리고 있습니다.

### 2.1 EtherCAT 소개

EtherCAT (Ethernet for Control Automation Technology) 은 토폴로지에 제한을 받지않는 유연한 토폴로지를 가진 고성능, 저비용의, 사용하기 쉬운 산업용 이더넷 기술입니다.

EtherCAT 의 상위 제어기 (주) 커미조아의 LX551 / LX552 / LX554 제품은 기존 이더넷 케이블을 통해 배선을 자유롭게 구성할 수 있습니다.

### 2.2 ComiECAT 라이브러리 주요 내용

(주) 커미조아의 LX551 / LX552 / LX554 는 전 세계적으로 가장 많은 사용자 층을 확보하고 있는 마이크로소프트 윈도우용 라이브러리를 제공하고 있습니다. 제공되는 라이브러리인 ComiECAT 의 다양한 기능을 통해 EtherCAT 네트워크의 상위 네트워크 제어기(Network Controller)인 LX551 / LX552 / LX554 의 모든 기능을 경험하실 수 있습니다.

#### 2.2.1 ComiECAT 개요 사항

ComiECAT 은 소프트웨어는 윈도우 표준 동적 라이브러리(Dynamic Link Library) 형태로 고객 여러분들께 제공되며, 마이크로소프트의 Visual C++ , Visual Basic .NET Framework 환경의 Visual C#, Visual J# 개발환경, 볼랜드 C++ Builder 전제품, Delphi 전제품 인텔 컴파일러, 기타 Visual Fortan 등의 다양한 개발환경을 지원하고 있습니다.

#### 2.2.2 함수 이름 규칙 가이드

구분 명	ComiECAT
장치 초기화 (Device Initialize)	ecGn_LoadDevices
	ecGn_UnLoadDevices
장치 인터페이스 (Net Interface)	ecNet_SetAlState
	ecNet_GetAlState
장치 설정 (ECAT Config)	ecCfg_SetNetConfig
	ecCfg_GetNetConfig
슬레이브 설정 (Slave I/F)	ecSlv_SlvIdx2PhysAddr
	ecSlv_PhysAddr2SlvIdx
모션 단축 (Motion Single Axis)	ecmSxMot_MoveStart
	ecmSxMot_MoveToStart
모션 보간 (Motion Interpolate)	ecmIxMot_LineStart
	ecmIxMot_LineToStart

표 3 함수이름 규칙

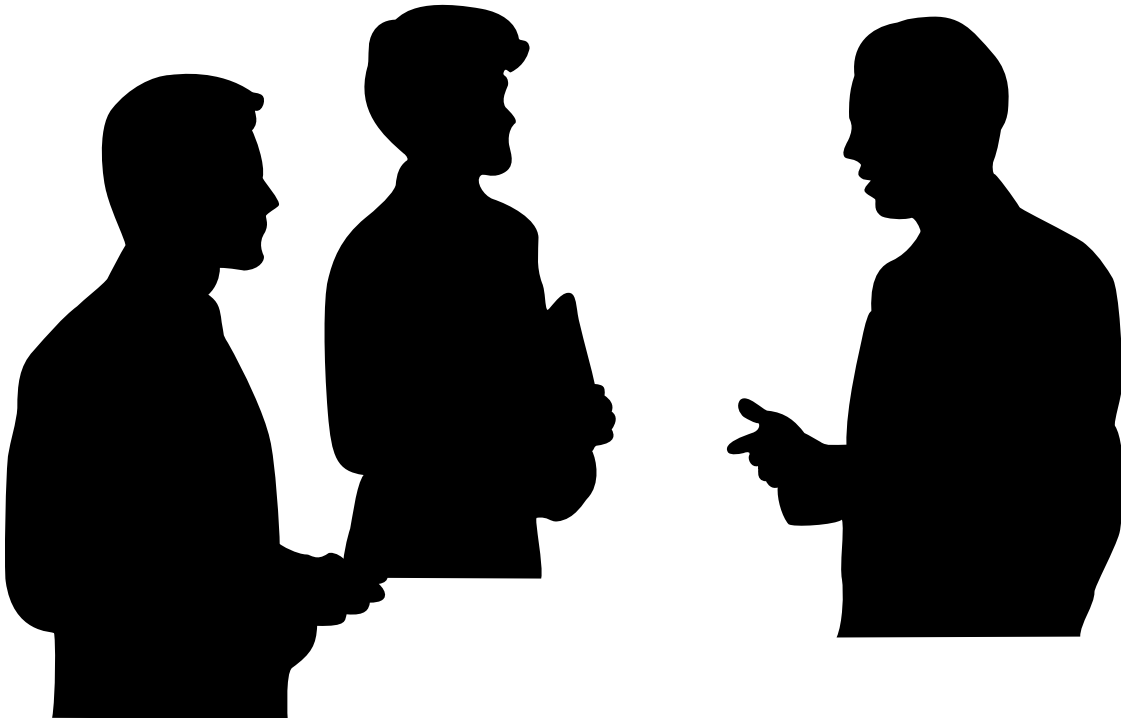
- 본 지면 상 전체 함수에 대해서 다루지 못한 부분은 각 제품 매뉴얼을 참조해주시기 바랍니다.

# Development Environment for ComiECAT

---

㈜커미조아는 통합 라이브러리 ComiECAT 을 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 ComiECAT 인터페이스를 통해 보다 편리하고 빠르게 저희 라이브러리에서 사용할 수 있도록 지원하여 드립니다.

**커**미조아 모션 및 범용 디지털 ComiECAT 은 다양한 고객(顧客) 여러분들의 요구에 발맞추어 개발되었습니다., 가까운 대리점 혹은 총판점을 통해 구입(購入)하신 경우에는 해당 구입처(購入處)를 통해 문의하시면, 더욱 빠른 기술 지원을 받으실 수 있습니다.



### 3 개발 환경 별 ComiECAT 사용 안내

#### 3.1 개발 환경 지원 안내

Supported Development Environment		개발 환경 별	언어 별	버전 별	제품 별
Microsoft	C#	Microsoft 社 Visual Studio	Visual C++	6.0 및 이하 버전을 포함한 VS2003, VS2005 환경	Enterprise Edition 을 포함한 전 제품
	Power Builder		Visual Basic		
Borland	Visual C++	Borland International 社 Borland Development Studio	C++ Builder	Builder 5, Builder 6 및 하위 버전 포함한 2006 버전	Architect, Professional, Enterprise
	Visual Basic		Delphi		
Sybase	Visual C++ .NET	Borland International 社 Borland Turbo Series	Visual Basic .NET	BDS 2006 버전	전 제품
	C++ Builder		Delphi		
Microsoft	C++ Builder	Borland International 社 Borland Turbo Series	Delphi	Turbo C++ Turbo Delphi / Turbo Delphi for .NET	전 제품
	Delphi		C# (Sharp)		
	PowerBuilder	Sybase PowerBuilder	PowerBuilder	PowerBuilder 8, PowerBuilder 9, PowerBuilder 10.5	전 제품

표 4 지원 개발 환경

ComiECAT 은 위와 같은 개발 환경을 지원하며, 별도로 명시되지 않은 개발 환경에서도 윈도우의 Dynamic Link Library 형태를 사용가능한 경우 ComiECAT 을 사용하실 수 있습니다. 이 사항에 대해서 더욱 자세한 내용을 알기 원하실 경우 Appendix(부록) 편을 참고해 주시기 바랍니다.



## 3.2 ComiECAT 구성

ComiECAT 은 아래 그림과 같이 실제 COMIZOA 제품군의 LX551 / LX552 / LX554 하드웨어를 추상화하여, 공통(共通) API 구조를 제공하고 있습니다.

ComiECAT 은 “Integration EtherCAT System Control Application Programming Interface” 라 명명하며, 그 의미는 저희 (주)커미조아 제품군의 기능과 그 기능을 사용하는 방법을 정의한 함수(Function)의 집합이라고 말할 수 있습니다. 고객(顧客)님께서서는 이제 저희 (주)커미조아 API 를 통해 제품군이 가지고 있는 다양한 기능을 사용할 수 있습니다.

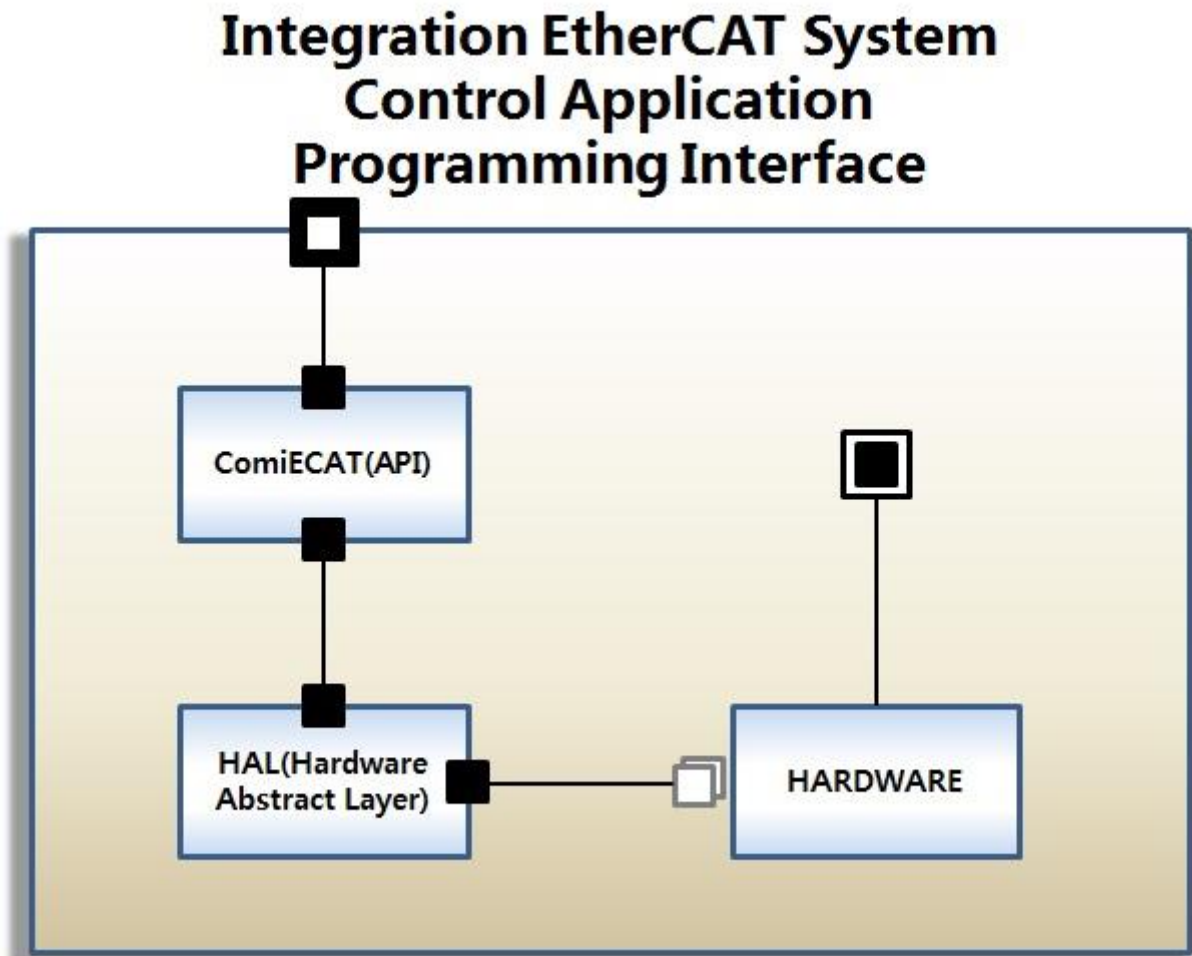


그림 3-1 ComiECAT 의 구조

### 3.2.1 HARDWARE Layer

다양한 커미조아의 제품군은 개별적인 인터페이스를 필요로 합니다. 그러나 이제 하드웨어 구조를 추상화하여, 일관적인 API 인터페이스를 제공하게 해주는 ComiECAT 을 통해 편리하게 모든 모션 장치를 일관적으로 제어하실 수 있습니다.

### 3.2.2 HAL(Hardware Abstract Layer)

Hardware Device 는 ComiECAT 이 지원하는 자사의 EtherCAT 장치를 총체적으로 이르고 있습니다. 특히 모션 장치는 고객(顧客)님께서 개발하는 응용 프로그램 계층과의 인터페이스인 ComiECAT 에서 총괄하고 있으며, 최상의 속도와 안정적인 동작에 대한 보장을 위한 다양한 기법으로 구조화되어

있습니다. 고객(顧客)님께서서는 COMIZOA 의 어떠한 모션에 대해서 단 하나의 API 를 통해 제어하실 수 있습니다. 구조의 바탕은 바로 ComiECAT 이 고급 HAL(Hardware Abstract Layer) 기능을 바탕화 했기 때문입니다.

### 3.2.3 ComiECAT Layer (API Layer)

마이크로소프트社의 Windows 98/ME/2000/XP/Vist/7 등의 제품군에서 동적 연결 라이브러리 방식을 지원하는 라이브러리를 의미합니다. 사용자는 Integration EtherCAT System Control API, 즉 ComiECAT 을 통해 제품 개별적인 하드웨어 인터페이스에 대한 사항들을 사용자는 직접 인지하고 있지 않아도 ComiECAT 은 고객(顧客)여러분들의 요구에 맞는 인터페이스를 제공하며, 신속하고 안정적인 응용프로그램 개발에 도움을 드립니다.

### 3.2.4 ComiECAT 인터페이스 구성 파일

개발 환경	MS VC++	Borland C++ Builder	Borland Delphi	MS Visual Basic	MS C# (C Sharp)
ComiECAT 인터페이스 파일 명	ComiECATSDK _Api.h	ComiECATSDK _Api.h			
	ComiECATSDK _Loader.cpp	ComiECATSDK _Loader.cpp			
ComiECAT 상수(常數)정의 파일	ComiECATSDK _Def.h	ComiECATSDK _Def.h	ComiECATSDK .PAS	ComiECATSDK .BAS	ComiECATSDK .CS
	ComiECATSDK _Err.h	ComiECATSDK _Err.h			
	ComiECATSDK _SysDef.h	ComiECATSDK _SysDef.h			

표 5 ComiSSCNET3 인터페이스 구성 파일 안내

Microsoft 社의 윈도우 운영체제에서 동작하는 DLL(Dynamic Link Library) 형태의 ComiECAT 은 상기 표에서 나타낸 것 처럼, 라이브러리 인터페이스를 위해 “ComiECAT 인터페이스” 파일과 라이브러리의 반환값 및 전달인자, 각 데이터 표기등을 위한 “ComiECAT 상수(常數) 정의 파일”을 제공하고 있습니다.

(쥬커미조아 고객(顧客)님께서서는 “ComiECAT 인터페이스 파일” 을 통해 ComiECAT 의 함수를 명시적으로 응용프로그램에 포함시킬 수 있으며, 이에 따라 상기 표기한 개발 환경을 지원해드리고 있습니다. 만약 이외의 환경에서 저희 ComiECAT 을 사용하시기 원하신다면, 저희 (쥬커미조아 고객(顧客) 지원 팀으로 연락 주시기 바랍니다.


### 3.3 각 개발 환경 별 안내

저희 (쥬커미조아에서는 고객(顧客)님들의 개발환경에 대한 고민을 덜어드리기 위하여, 범용적인 객체지향 언어인 C++ 부터 Borland International 社의 Object Pascal 의 Delphi 제품군, 그리고 최신 .NET 환경까지 고려한 라이브러리 인터페이스를 제공하고 있습니다. 특히 이번 ComiECAT 에서는 개발 환경의 새 패러다임인 C Sharp(C#) 언어를 본격 지원하고 있으며, 보다 새로운 개발환경에서 저희 (쥬커미조아 제품군을 경험하실 수 있도록 만전을 기하고 있습니다.

필요한 인터페이스(Interface) 파일은 실제 개발언어 및 환경에서 Header 파일또는 프로젝트 구성파일로 반드시 사용이 됩니다. 저희 (쥬커미조아 ComiECAT 에서 제공하는 인터페이스파일의 경로는 다음과 같습니다.

Visual C++ / Borland C++ Builder(이 부분은 아직 수정해야 하므로 참고하지 마시기 바랍니다.)	
경로 명	C:\Program Files\COMIZOA\AUTOMATION3\ECAT\Libraries\x86\Motion(DLL)\SSCNET3_LX540\VC++
파일 명	ComiSSCNET3_SDK.cpp, ComiSSCNET3_SDK.h, ComiSSCNET3_SDK_Def.h
Delphi	
경로 명	C:\Program Files\COMIZOA\AUTOMATION3\NEMO\Libraries\x86\Motion(DLL)\SSCNET3_LX540\Del
파일 명	ComiSSCNET3.PAS
Visual Basic	
경로 명	C:\Program Files\COMIZOA\AUTOMATION3\NEMO\Libraries\x86\Motion(DLL)\SSCNET3_LX540\VB
파일 명	ComiSSCNET3.BAS
C# (CSharp)	
경로 명	C:\Program Files\COMIZOA\AUTOMATION3\NEMO\Libraries\x86\Motion(DLL)\SSCNET3_LX540\CSharp
파일 명	ComiSSCNET3_SDK.cs

표 6 이 경로는 ㈜커미조아에서 제공하는 COMI-AUTOMATION 설치 프로그램이 기본 경로인 'C:' 드라이브에 설치된 것을 기준으로 합니다.

 <p>안내</p>	<p>인터페이스 파일에 대해서 자세히 설명해 주십시오.</p> <p>먼저, 저희 ㈜커미조아에서 제공하는 ComiECAT 은 DLL 형태의 독립 라이브러리 인터페이스를 제공하는 마이크로소프트웨어의 윈도우 표준 라이브러리를 말합니다. 이것은 당사의 제품의 기능을 담당하는 가장 중요한 함수의 집합 구성입니다.</p> <p>고객(顧客)님들께서는 이러한 함수, 즉 기능을 통해서 저희 ㈜커미조아의 제품군을 이용하시게 됩니다.</p> <p>그러나 마이크로소프트의 DLL 라이브러리 형태는 구조상 제공하는 함수들을 어떻게 써야 하는 지의 정보를 정확하게 알 수 없습니다. 다시 말씀 드려서, 함수의 매개변수의 개수나 매개변수의 형태 즉, 함수의 사용법을 알 수가 없습니다.</p> <p>ComiECAT 에 대해서 그 사용방법을 각 개발환경에 맞게 제공해드리기 위해 [인터페이스] 파일을 제공하고 있습니다. 각 개발환경(VC++, Delphi 등)에서 개발을 하시는 고객(顧客) 여러분들께서는 반드시 이 [인터페이스] 파일을 사용하셔야만, ComiECAT 을 사용할 수 있습니다.</p>
---	---

### 3.3.1 Visual C++ 6.x 개발자를 위한 안내

Microsoft Visual C++ 6.x 에서 ComiECAT 을 사용하시려면 다음의 절차에 따라 사용하시면 됩니다.

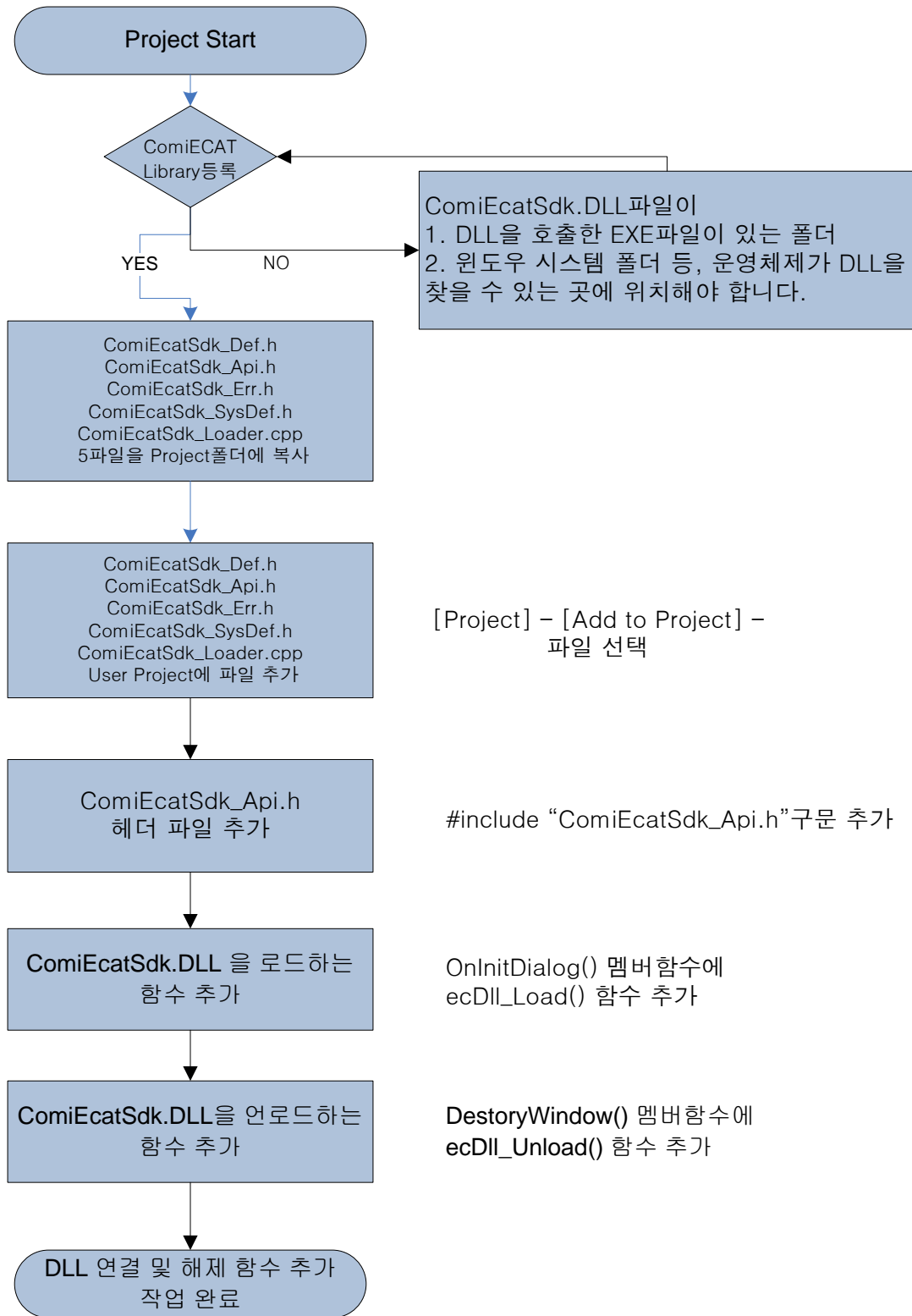


그림 3-2 Visual Studio 6.x 에서의 ComiECAT 사용 순서도

Visual C++ 6.x 을 실행합니다. 메뉴에서 'File'->'New' 를 선택하여 새로운 프로젝트 생성을 시작합니다.

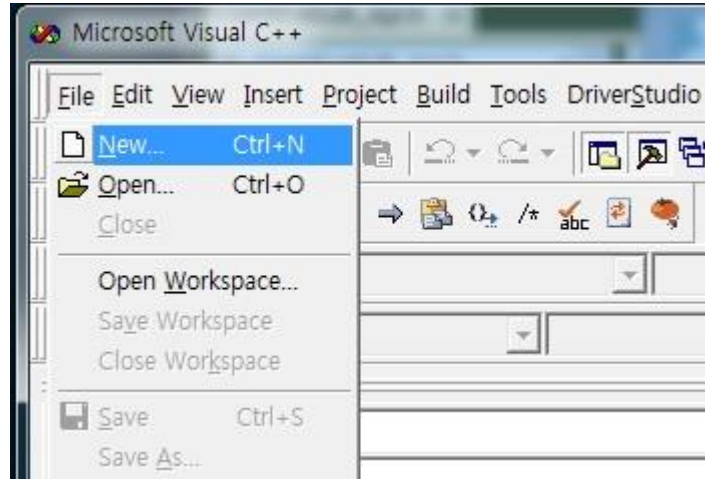


그림 3-3 Visual C++ 6.x 의 새로운 프로젝트 생성 화면

MFC AppWizard(exe)를 선택하고, 프로젝트를 생성할 위치와 프로젝트 이름을 입력한 후 [OK]버튼을 클릭 합니다.

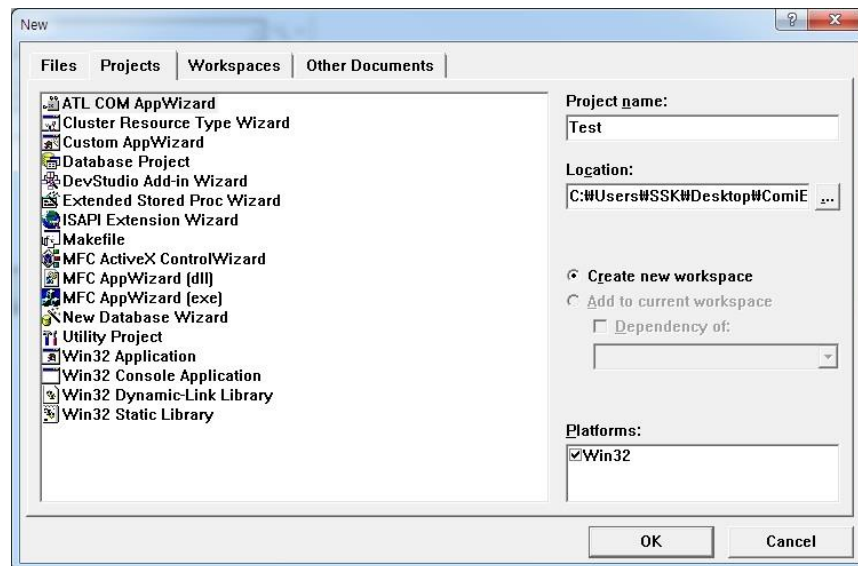


그림 3-4 새로운 프로젝트 생성 화면

MFC AppWizard 창이 나타나면 [Dialog based]를 선택하고 [Finish]버튼을 클릭합니다.

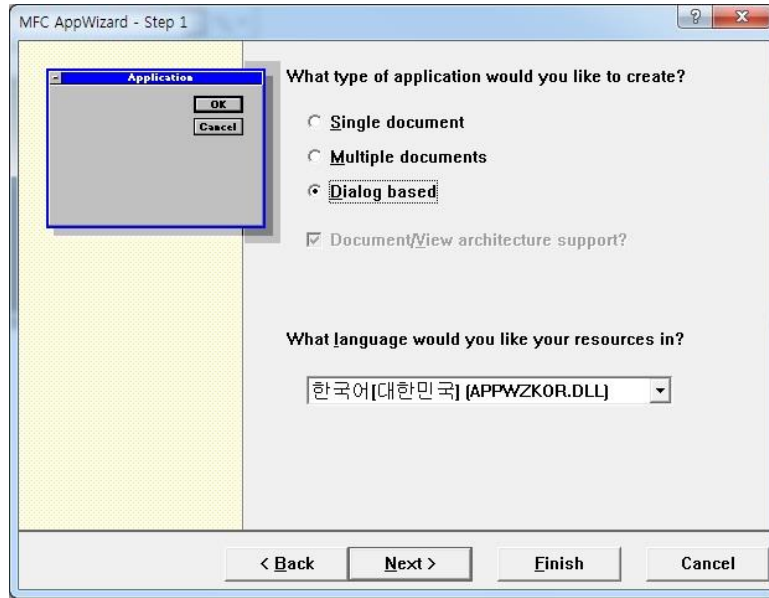


그림 3-5 MFC AppWizard 의 Application Type 선택 화면

VC++ 용 인터페이스 정의 파일인 ComiEcatSdk\_Def.h, ComiEcatSdk\_Api.h, ComiEcatSdk\_Err.h, ComiEcatSdk\_SysDef.h, ComiEcatSdk\_Loader.cpp 파일을 신규로 생성한 프로젝트 폴더로 복사 합니다. 메뉴에서 [Project]->[Add To Project]->[Files]를 선택합니다.

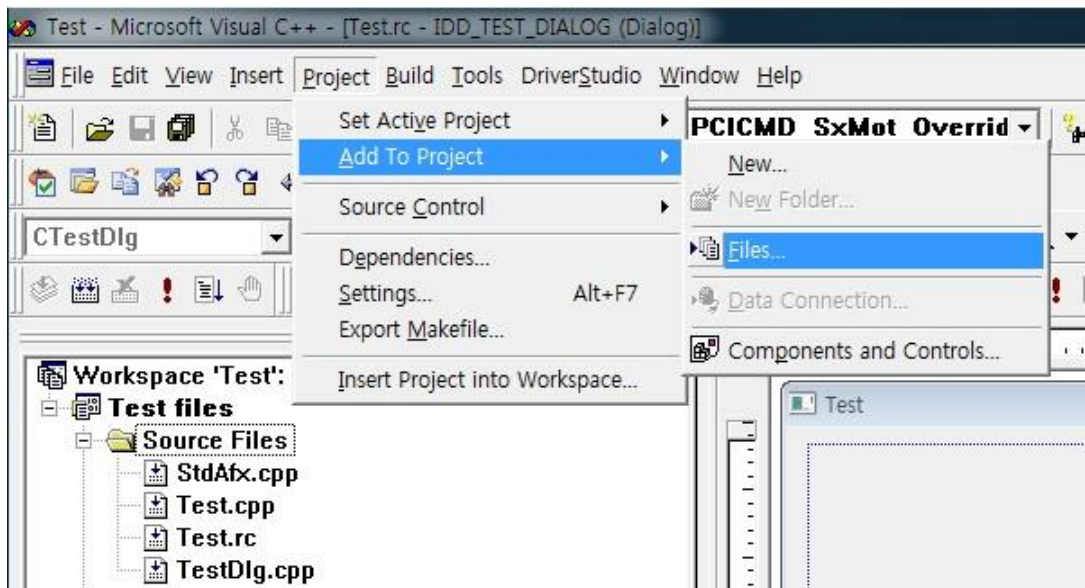


그림 3-6 프로젝트에 새로운 파일 추가 선택화면



추가될 파일을 선택한 후 [OK]버튼을 클릭하여 통합 모션 라이브러리 인터페이스 파일인 다섯 프로젝트에 추가 합니다.

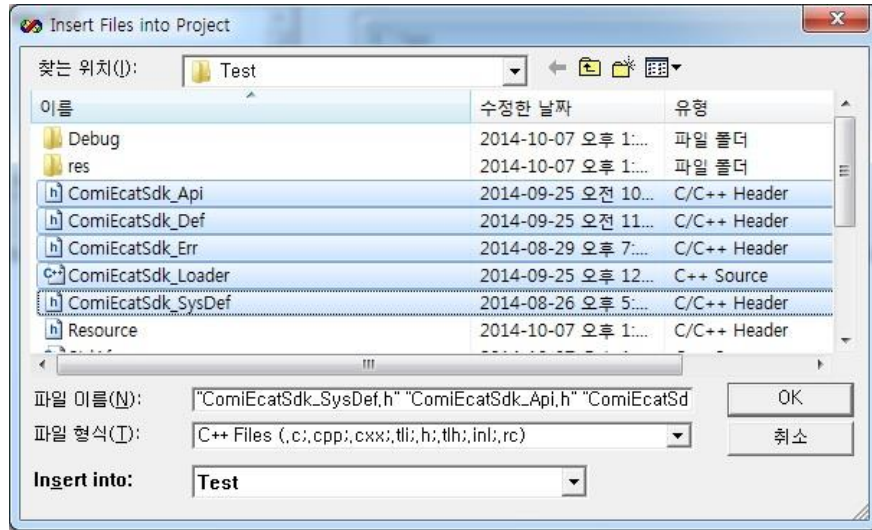


그림 3-7 프로젝트에 추가할 파일 선택 화면

Workspace 창의 FileView 탭에서 [생성한 프로젝트 이름]+Dlg.cpp 파일을 선택합니다.

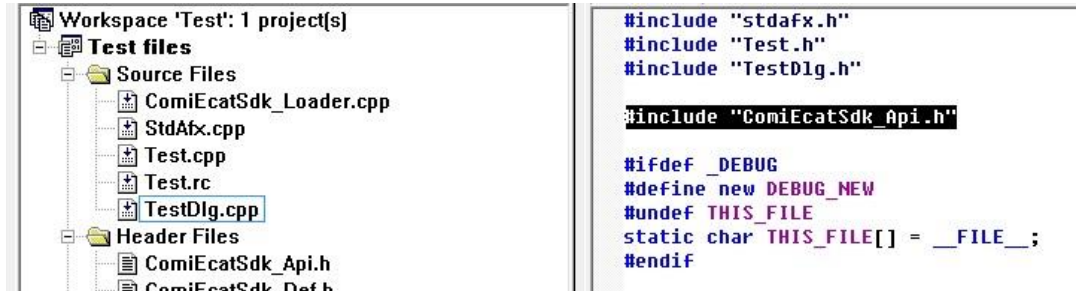


그림 3-8 사용자가 MFC AppWizard 를 통해서 생성한 소스코드에 ComiSSCNET3 파일을 추가 함

([생성한 프로젝트 이름]+Dlg.cpp) 파일의 OnInitDialog()함수 내부의 “TODO”아래에 “ecDll\_Load();”을 추가 합니다.

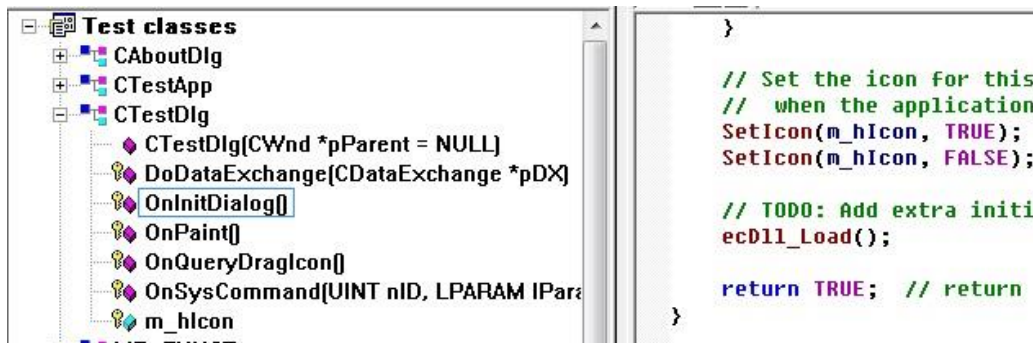


그림 3-9 DLL 로드 함수 호출 화면

사용자 작성 프로그램이 종료되면 DLL 을 Unload 시켜야 합니다. DLL 의 Unload 는 사용자 작성 프로그램의 종료시 이루어져야 하며 ecDll\_Unload()이라는 함수를 통해서 이루어 집니다. ecDll\_Unload()를 추가 하는 방법은 다음과 같습니다.

Class View 창에서 [(생성한 프로젝트 이름)+Dlg] 클래스를 마우스 오른쪽 버튼으로 클릭 합니다. 팝업 메뉴에서 [Add Virtual Function]을 선택합니다.

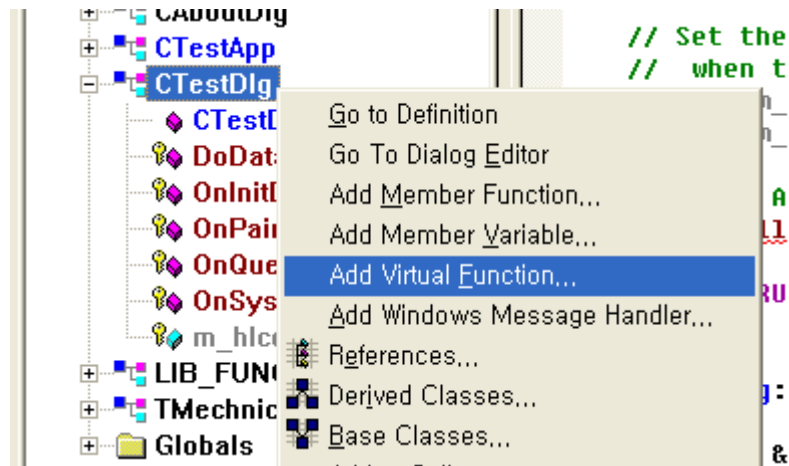


그림 3-10 가상 함수 추가

'New Virtual Functions'항목에서 'DestroyWindow'를 선택한 다음 [Add and Edit]버튼을 클릭합니다.

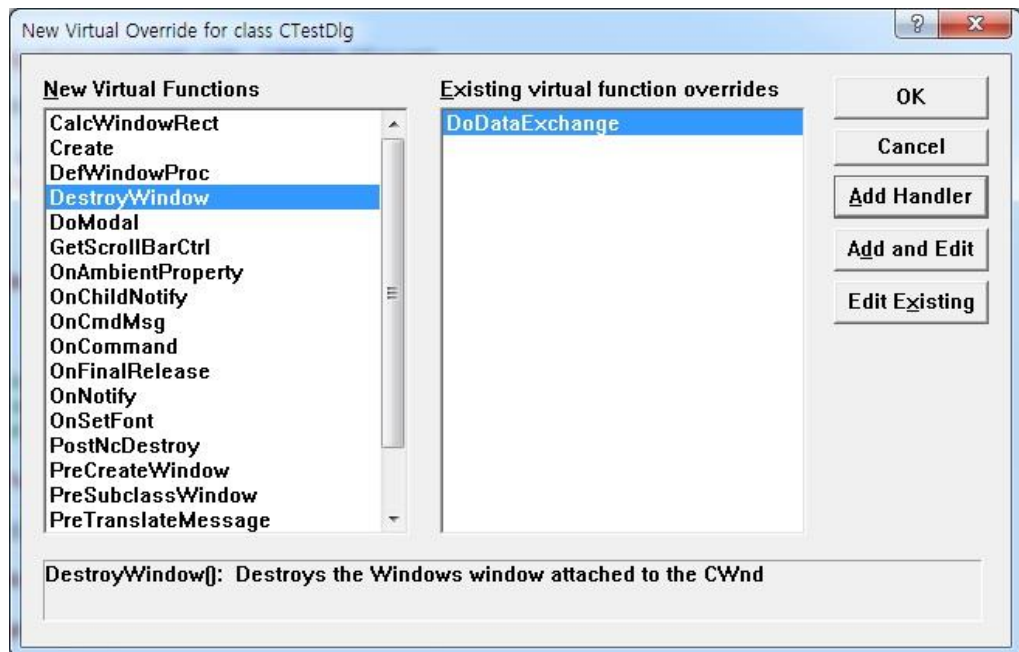


그림 3-11 DestroyWindow 함수 추가



(생성한 프로젝트 이름)+Dlg 클래스의 멤버함수인 'DestroyWindow()'에 'ecDll\_Unload();'을 추가합니다. 'ecDll\_Unload()'함수를 추가 하면 윈도우가 종료 될 때 자동으로 DLL도 해제 됩니다.

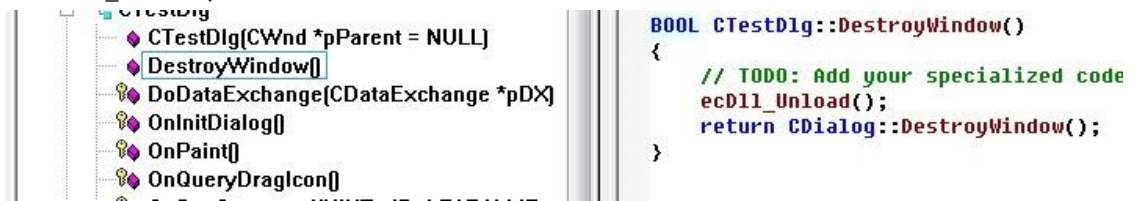


그림 3-12 DLL 로드 및 언로드 코드 추가

### 3.3.2 Visual C++ 10.x 개발자를 위한 안내

Microsoft 社의 Visual C++ 10.x (Visual Studio 2010)에서 ComiECAT 을 사용하시려면 다음의 절차에 따라 사용하시면 됩니다.

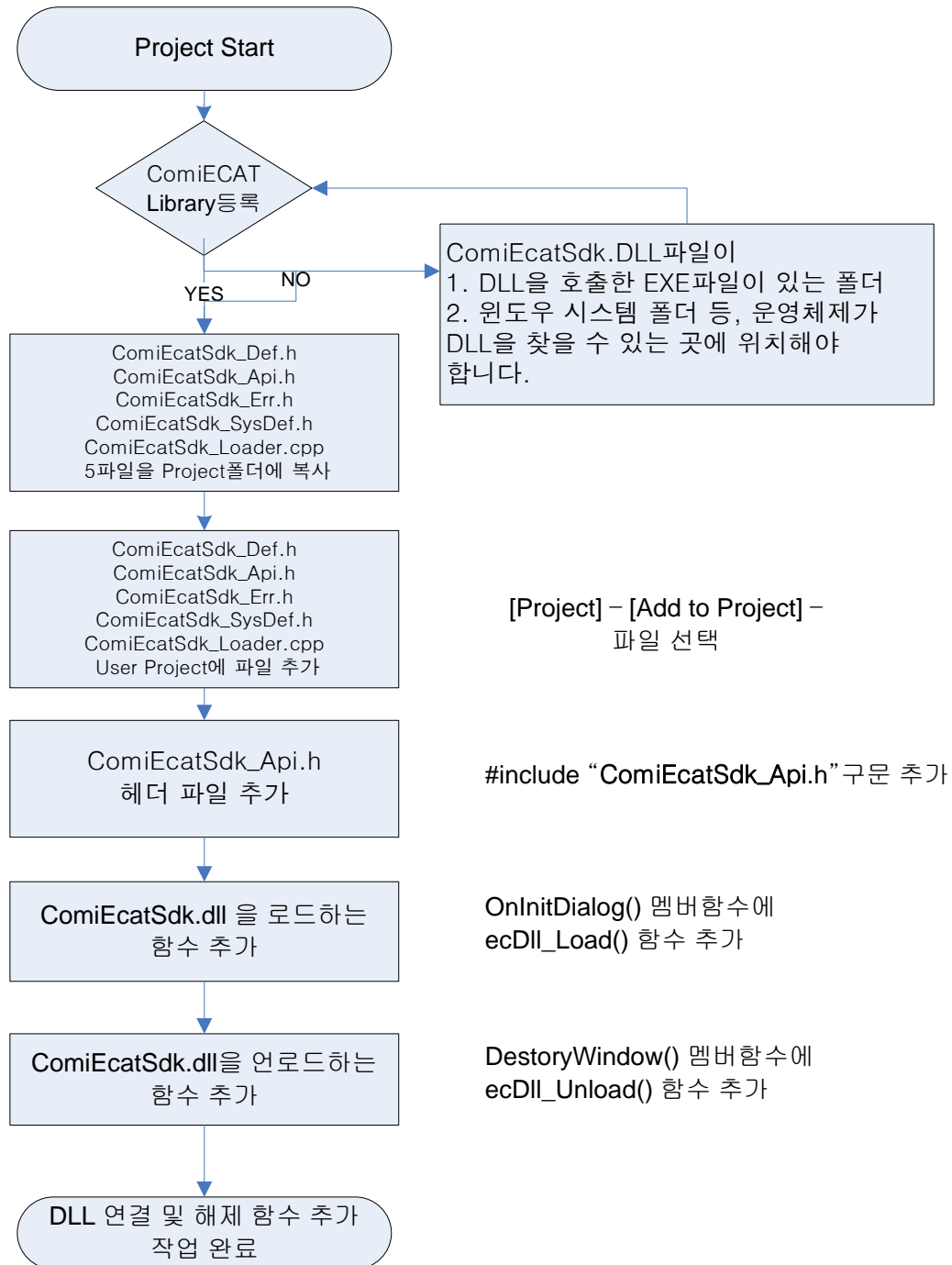


그림 3-13 Visual Studio 10.x 에서의 ComiECAT 사용 순서도

Microsoft 社의 Visual Studio 2010(이하 VS2010)를 실행 합니다.

메뉴에서 [파일]>[새로 만들기]>[프로젝트]를 선택하여 새로운 프로젝트를 시작 합니다.

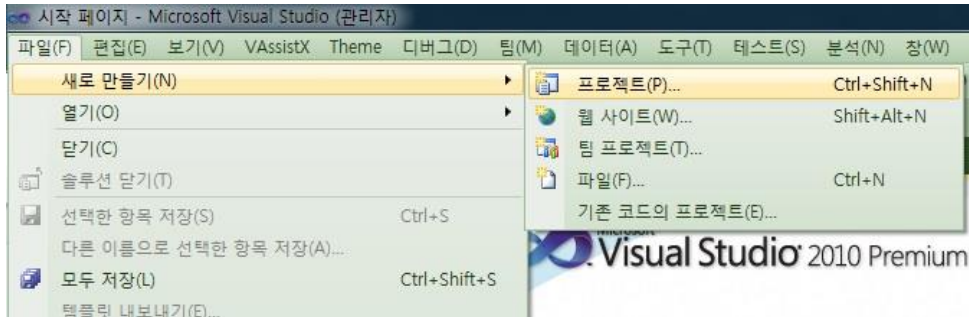


그림 3-14 새로운 프로젝트 생성 시작

[새 프로젝트]창이 화면에 나타나면, [설치된 템플릿]에서는 [Visual C++]을 선택하고, [Templates]에서 [MFC 응용 프로그램]을 선택합니다. 그리고 프로젝트를 생성할 위치와 프로젝트 이름을 입력한 후 [OK]버튼을 클릭 합니다.

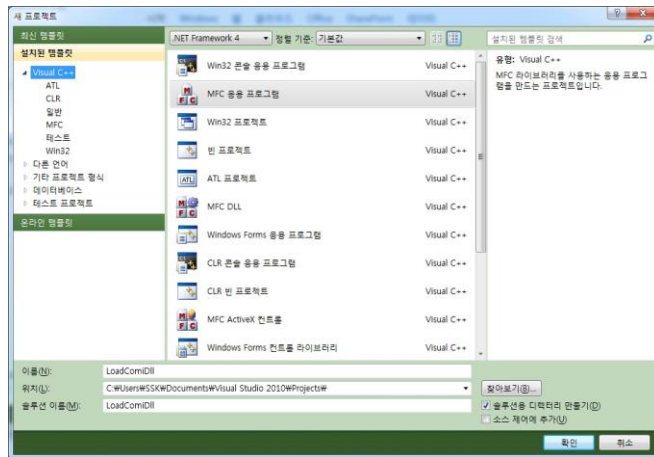


그림 3-15 새 프로젝트 옵션 선택화면

[MFC 응용 프로그램 마법사] 창이 화면에 나타나면, [다음]을 클릭합니다.

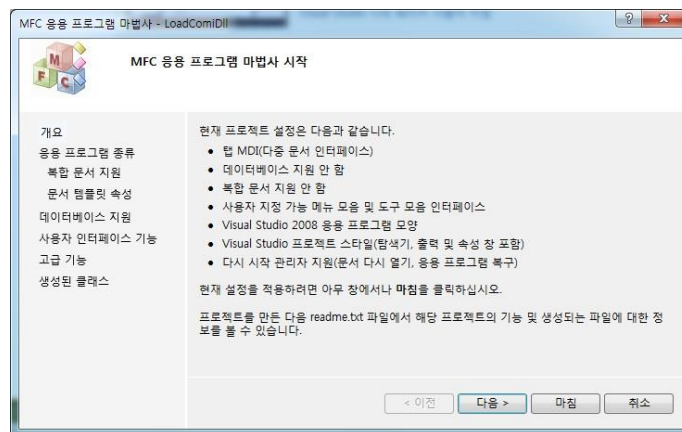


그림 3-16 MFC Application Wizard 의 Overview 화면

[응용 프로그램 종류]에서 [대화 상자 기반]을 선택하고,[유니코드 라이브러리 사용]를 해제(Uncheck)한 다음 [마침]를 클릭합니다.

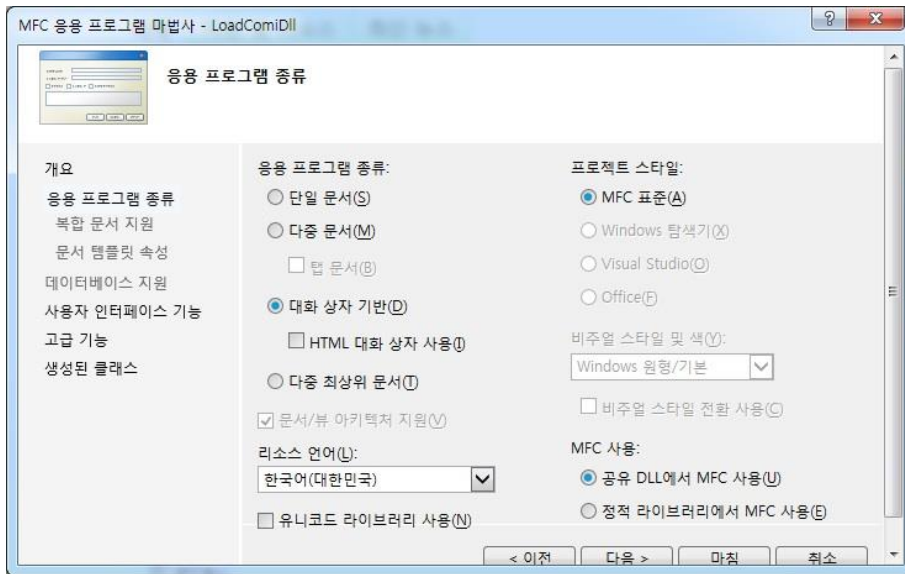


그림 3-17 MFC Application Wizard 의 Application Type 화면

VC++ 용 인터페이스 정의 파일인 ComiEcatSdk\_Api.h, ComiEcatSdk\_Def.h, ComiEcatSdk\_Err.h, ComiEcatSdk\_SysDef.h, ComiEcatSdk\_Loader.cpp 파일을 신규로 생성한 프로젝트 폴더로 복사 합니다.

메뉴에서 [프로젝트]->[기존 항목 추가]을 선택합니다.

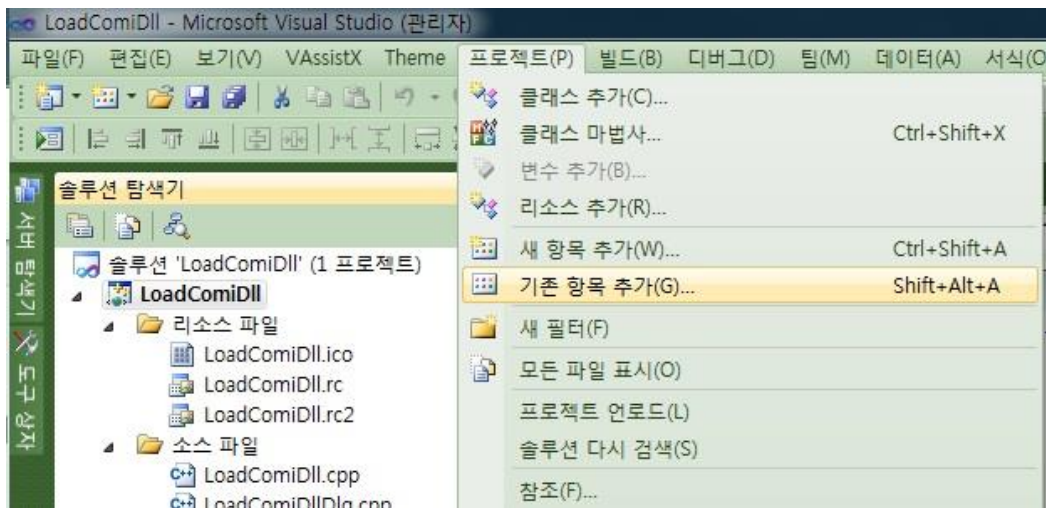


그림 3-18 메뉴에서 기존 항목 추가 선택 화면

추가될 파일을 선택한 후 [OK]버튼을 클릭하여 다섯 개의 파일을 프로젝트에 추가 합니다.

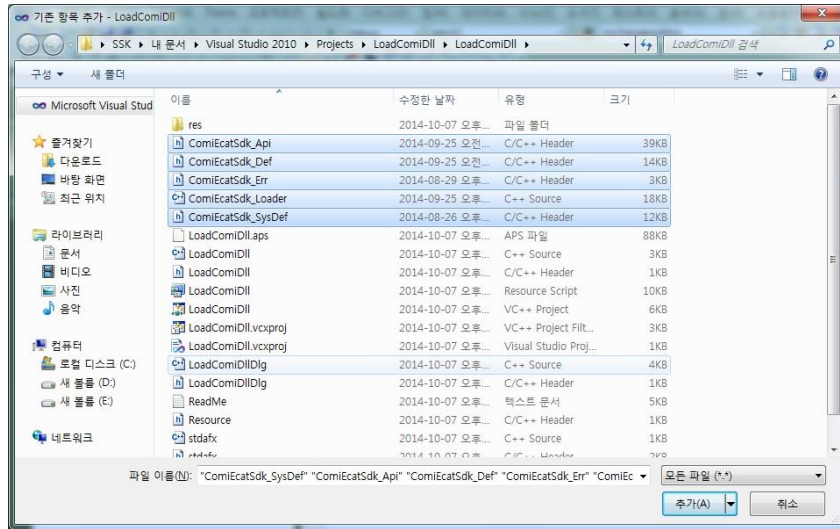


그림 3-19 새로 추가할 파일들 선택 화면

WorkSpace 창의 FileView 탭에서 ([생성한 프로젝트 이름]+Dlg.cpp) 파일을 선택합니다. 파일의 가장 위쪽에 인터페이스 파일을 추가합니다.

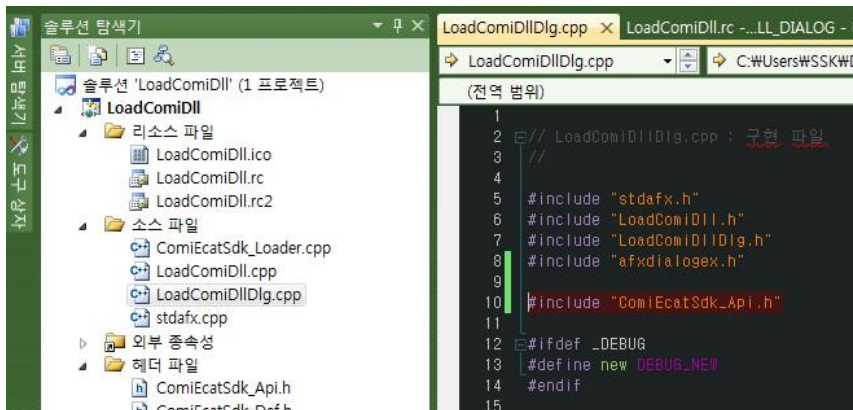


그림 3-20 사용자 생성 CPP 파일에 헤더 추가 화면

(생성한 프로젝트 이름)+Dlg.cpp 파일의 OnInitDialog() 함수 내부의 “TODO”아래에 “ecDll\_Load();”를 추가 합니다.



그림 3-21 LoadDll 추가

사용자 작성 프로그램이 종료되면 DLL 을 Unload 시켜야 합니다. DLL 의 Unload 는 사용자 작성 프로그램의 종료시 이루어져야 하며 ecDll\_Unload()이라는 함수를 통해서 이루어 집니다. ecDll\_Unload()을 추가 하는 방법은 다음과 같습니다.

Class View 창에서 ([생성한 프로젝트 이름]+Dlg) 클래스를 선택합니다.

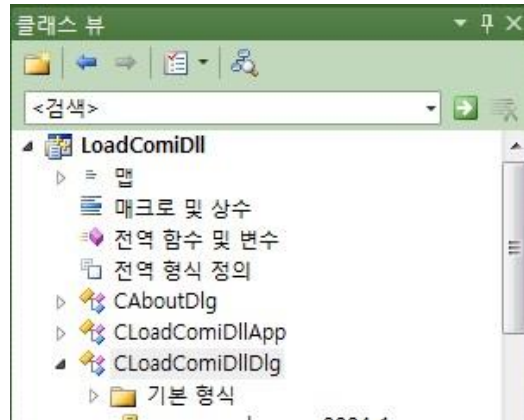


그림 3-22 사용자 생성 Dialog Class 선택

(생성한 프로젝트 이름)+Dlg 클래스가 선택된 상태에서 Properties 창의 'Overrides'를 선택합니다.

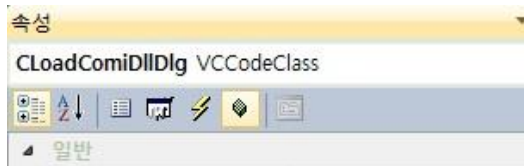


그림 3-23 Overrides 선택

'DestroyWindow' 항목 옆의 콤보 박스를 클릭하여 '<Add> DestroyWindow'를 선택합니다. (생성한 프로젝트 이름)+Dlg 클래스에 'DestroyWindow'라는 이름의 오버라이드된 함수가 추가 됩니다.

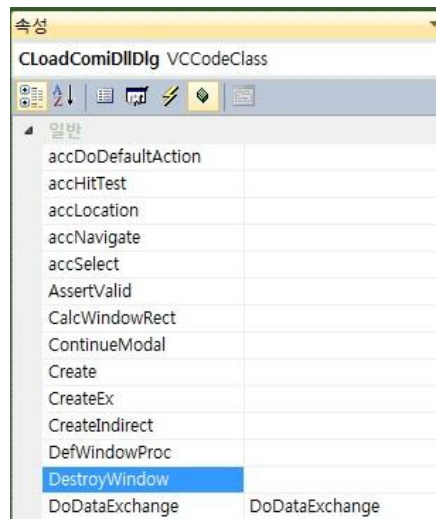


그림 3-24 Destroy Window 함수 추가

([생성한 프로젝트 이름]+Dlg) 클래스의 멤버함수인 DestroyWindow()에 'ecDll\_Unload();'을 추가합니다. 'ecDll\_Unload()'함수를 추가 하면 윈도우가 종료 될 때 자동으로 DLL도 해제 됩니다.



```
157
158
159 800L CLoadComiDllDlg::DestroyWindow()
160 {
161     // TODO: 여기에 특수화된 코드를 추가 및
162
163     ecDll_Unload();
164
165     return CDialogEx::DestroyWindow();
166 }
167
```

그림 3-25 UnloadDll 함수 추가



### 3.3.3 Borland C++ Builder 개발자를 위한 안내

Borland C++ Builder 는 해당 개발 환경 버전인 BCB 5, BCB 6 및 BDS 2006 버전에서 ComiECAT의 인터페이스 연결 방법이 매우 유사하기 때문에 공통적인 부분으로서 안내를 해드립니다.

전체 버전(Version)의 Borland C++ Builder 에서 ComiECAT 을 사용하시려면 다음의 절차를 통해 안내 받으시기 바랍니다.

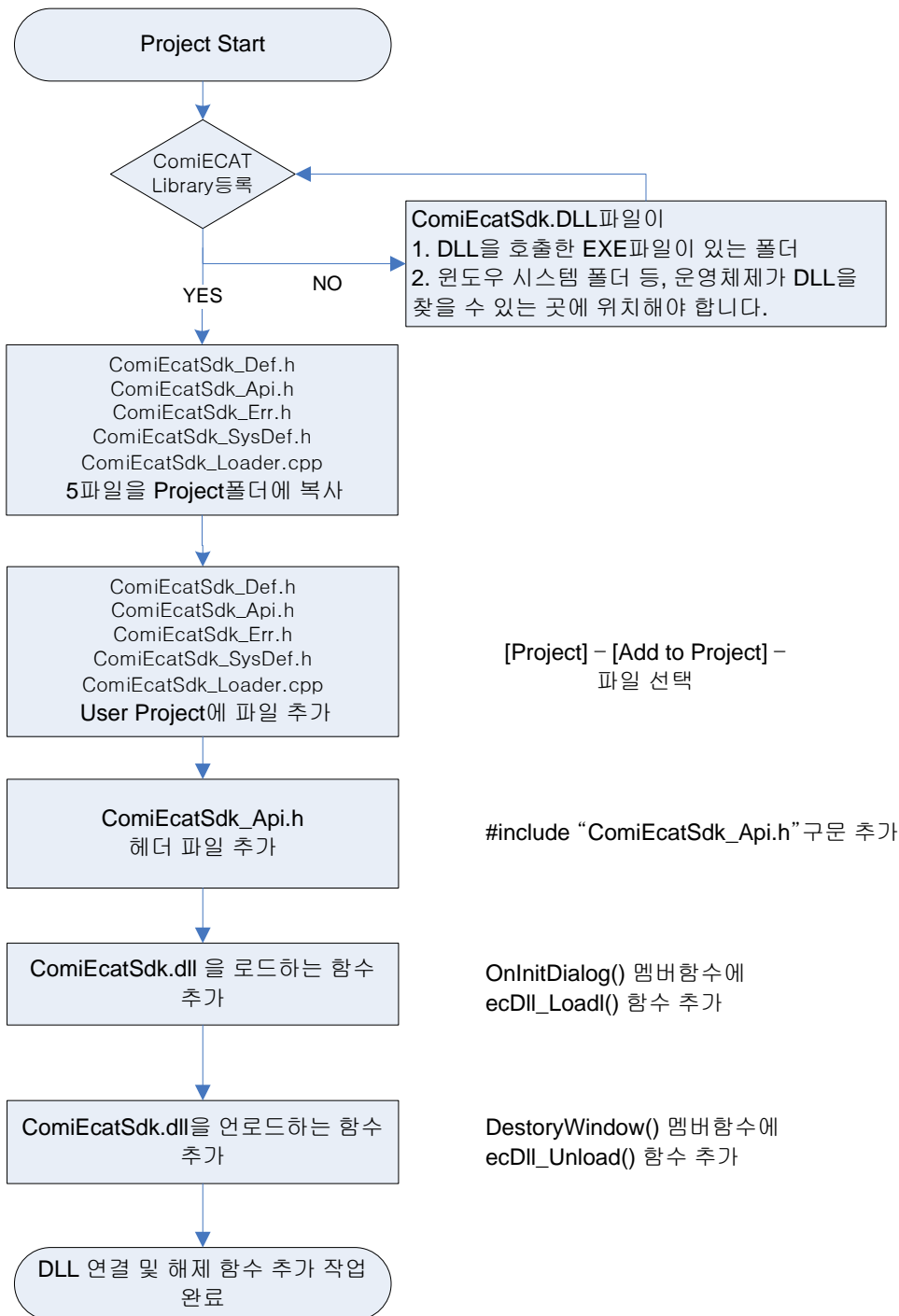


그림 3-26 Borland C++ Builder 에서 ComiECAT 사용 순서도



본 개발자를 위한 실제 안내에서는 다양한 버전의 Borland C++ Builder 의 화면을 통해 안내 해드리도록 하겠습니다.

Borland C++ Builder 를 실행합니다. 메뉴에서 [File]->[New]->[Application]을 선택하여 새로운 프로젝트를 시작합니다.



그림 3-27 BCB 5 에서 새로운 프로젝트 생성

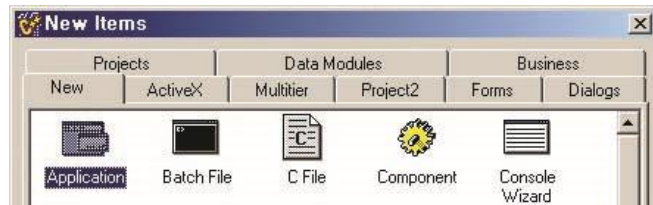


그림 3-28 BCB 5 에서 새로운 프로젝트 생성

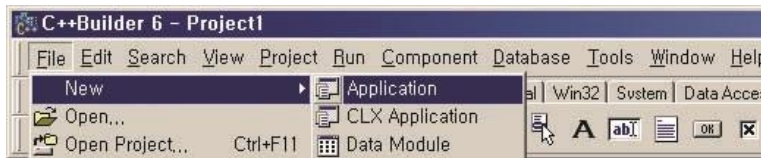


그림 3-29 BCB 6 에서 새로운 프로젝트 생성




그림 3-30 BDS 2006 에서 새로운 프로젝트 생성

Borland C++ 및 VC++ 용 공용 인터페이스 정의 파일인 ComiEcatSdk\_Api.h, ComiEcatSdk\_Def.h, ComiEcatSdk\_Err.h, ComiEcatSdk\_SysDef.h, ComiEcatSdk\_Loader.cpp 파일을 신규로 생성한 프로젝트 폴더로 복사 합니다.

이름	수정한 날짜	유형
ComiEcatSdk_Api	2014-09-25 오전...	C/C++ Header
ComiEcatSdk_Def	2014-09-25 오전...	C/C++ Header
ComiEcatSdk_Err	2014-08-29 오후...	C/C++ Header
ComiEcatSdk_Loader	2014-09-25 오후...	C++ Source
ComiEcatSdk_SysDef	2014-08-26 오후...	C/C++ Header

그림 3-31 ComiECAT 사용시 공통으로 사용되는 파일

 <p>안내</p>	<p><b>참고 사항</b></p> <p>Borland 社の C++ Builder 에서는 [File]-[Save] or [Save All]을 해주어야 프로젝트 폴더 및 프로젝트 관련 파일들을 생성합니다.</p>
---	---

그림과 같이 C++ Builder 에서 추가할 인터페이스 파일을 실제 사용자 프로젝트에 추가합니다. Project 의 메뉴의 Add to Project 를 사용하시면 됩니다.

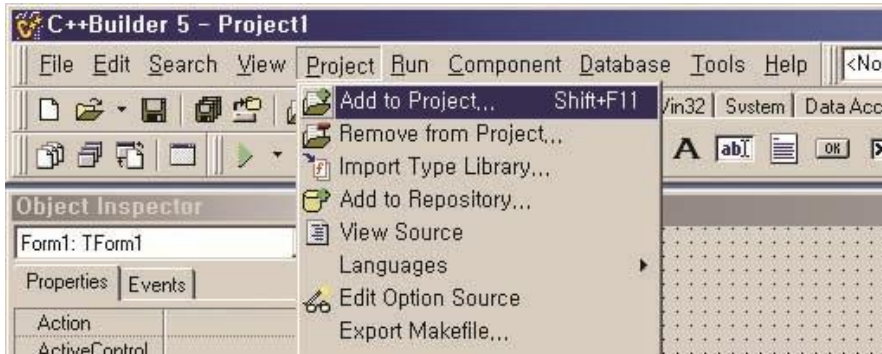


그림 3-32 C++ Builder 에서 프로젝트에 파일 추가 단계 1

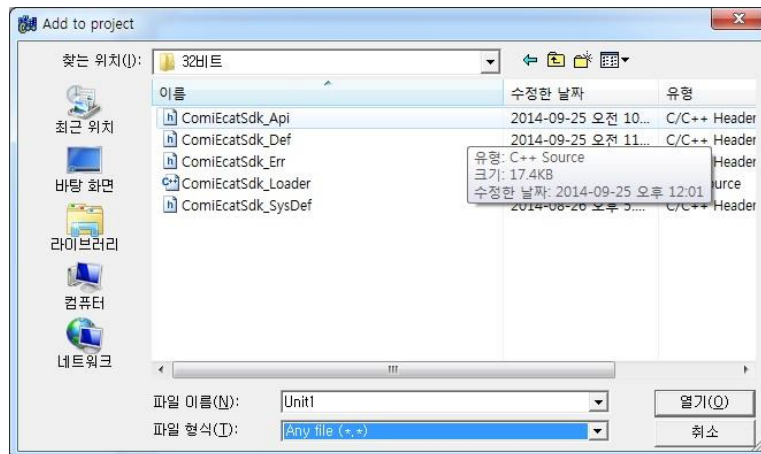


그림 3-33 C++ Builder 에서 프로젝트에 파일 추가

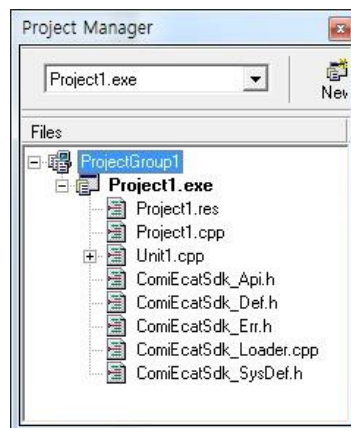


그림 3-34 Project Manager 에서 추가된 파일 확인(確認)

라이브러리 함수를 사용하고자 하는 대상 구현부 응용프로그램 파일에 인터페이스 파일을 선언합니다.

```

Unit1.cpp | Project1.cpp |
//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "ComiEcatSdk_Api.h"
//-----
    
```

그림 3-35 라이브러리 사용시 필요한 헤더 파일 선언

ComiEcatSdk.dll 파일을 ecDll\_Load() 함수를 이용하여 로드할 수 있도록 FormCreate 함수 또는 응용프로그램 시작 부분에 추가합니다.

ComiEcatSdk.dll 파일을 ecDll\_Load() 함수를 이용하여 로드합니다. ecDll\_Load()을 추가 하는 방법을 FormCreate 를 통해 할 수 있으며, 그 예를 소개해 드립니다.

[Object Inspector] - [Events]탭의 OnCreate 에서 더블클릭합니다.



그림 3-36 OnCreate Event 추가하여 FormCreate 함수와 연결

추가된 FormCreate() 또는 응용프로그램 종료 함수 안에 ecDll\_Load() 함수를 추가합니다.

```

Unit1.cpp | Project1.cpp |
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    ecDll_Load();
}
//-----
    
```

그림 3-37 ecDll\_Load 함수 추가

고객(顧客)님이 작성하신 프로그램이 종료되면 DLL 을 명시적으로 Unload 시켜야 합니다. DLL 의 Unload 시점은 고객(顧客)님께서 작성하신 응용프로그램이 종료되는 시점에 반드시

이루어져야 하며 ecDll\_Unload()라는 함수를 통해서 이루어 집니다. ecDll\_Unload()을 추가 하는 방법은 다음과 같습니다.

[Object Inspector] - [Events]탭의 OnDestroy 에서 더블클릭합니다.

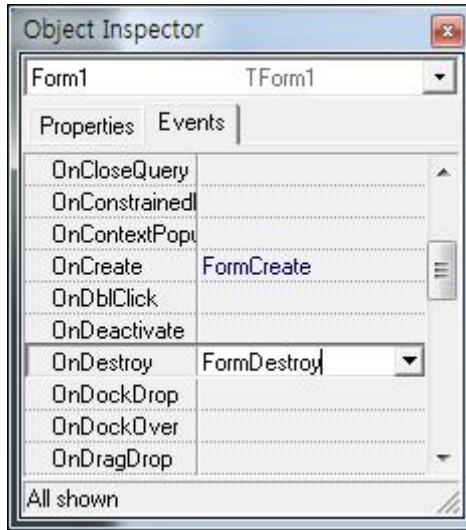


그림 3-38 응용프로그램의 종료시 DLL 이 명시적으로 UnLoad 될 수 있도록 OnDestroy Event 와 함수의 연결

추가된 FormDestroy()또는 응용프로그램 종료 함수에 ecDll\_Unload() 함수를 추가합니다.

```

Unit1.cpp | Project1.cpp |
-----
void __fastcall TForm1::FormDestroy(TObject *Sender)
{
    ecDll_Unload();
}
-----
    
```

그림 3-39 FormDestroy 함수를 통하여 명시적인 ecDll\_Unload 함수 구현

### 3.3.4 Borland Delphi 개발자를 위한 안내

Borland Delphi 는 해당 개발 환경 버전인 Delphi 5, Delphi 6 및 Delphi 7, BDS 2006 버전에서 ComiECAT 의 인터페이스 연결 방법이 매우 유사하기 때문에 공통적인 부분으로서 안내를 해드립니다.

전체 버전(Version)의 Delphi 에서 ComiECAT 을 사용하시려면 다음의 절차를 통해 안내 받으시기 바랍니다.

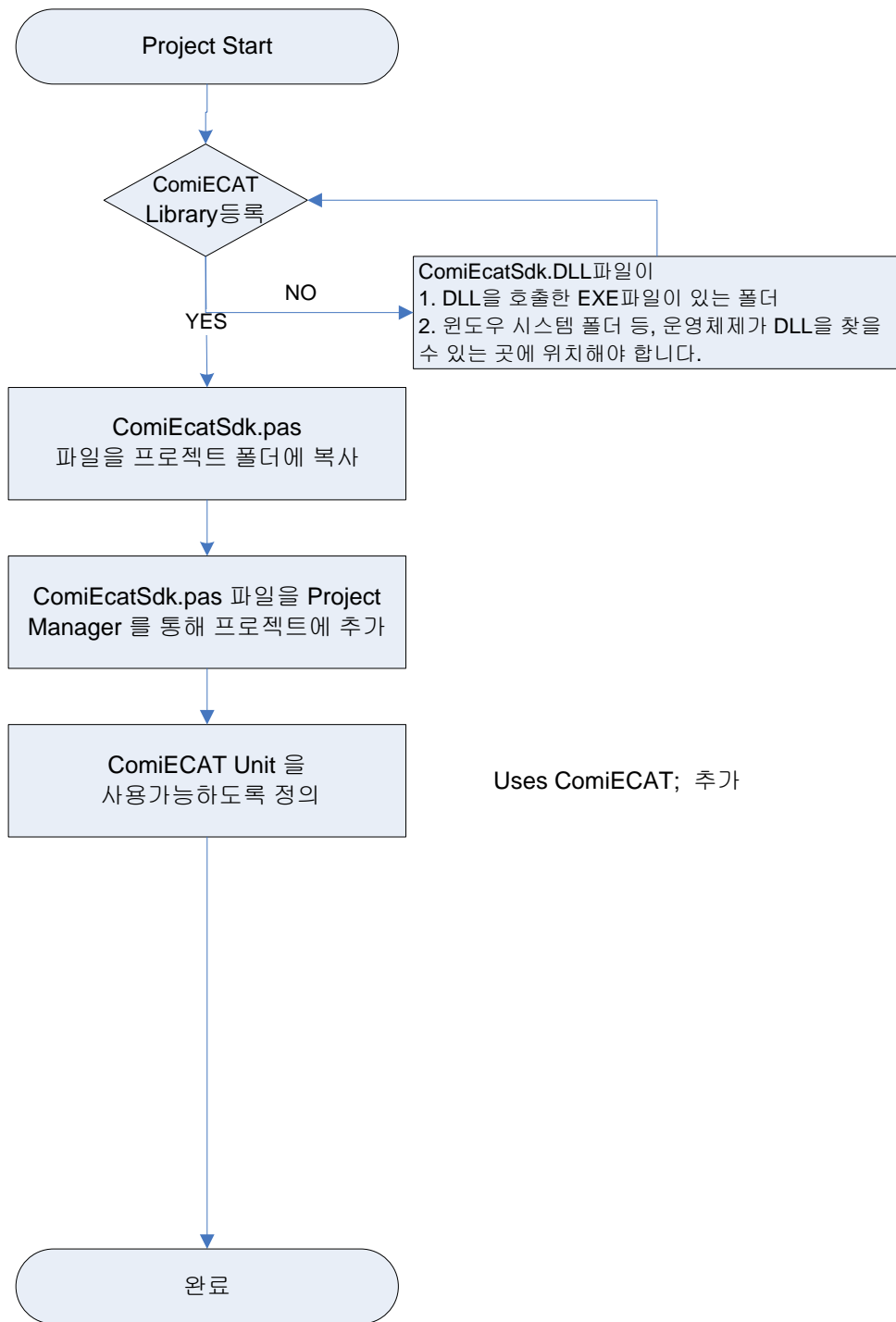


그림 3-40 Borland Delphi 에서 ComieCAT 사용 순서도

Delphi 7 을 기준으로 설명드리겠습니다. 만약 안내해드리는 도중에 다른 개발 환경과 구분이 되어야 할 내용은 별도로 설명 드리겠습니다. Borland Delphi 를 실행합니다.

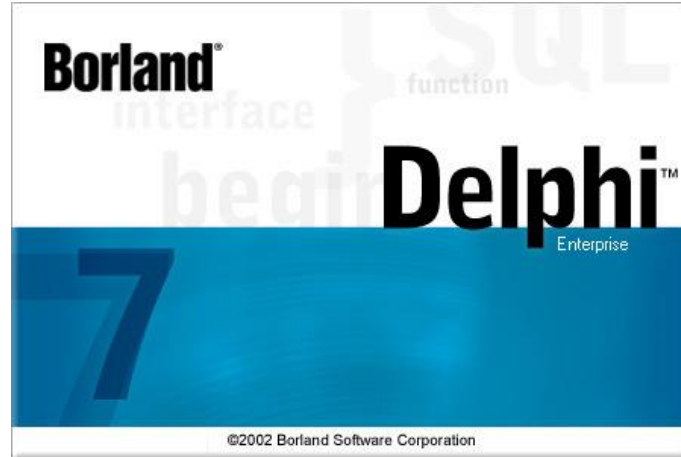


그림 3-41 Borland Delphi 7 화면

프로젝트 시작 전에 (주)커미조아 ComiECAT 의 Delphi 용 공용 인터페이스 파일을 프로젝트 폴더에 복사합니다. 이 파일은 (주)커미조아 ComiECAT 의 DLL(Dynamic Link Library) 와 고객님의 응용 프로그램과의 인터페이스를 정의하여 놓은 파일입니다.

델파이(Delphi)는 명시적으로 프로젝트파일 간의 상호 변환이 필요 없습니다. 따라서 Delphi 5, Delphi 6, Delphi 7 각각의 버전에서 몇 가지 기본적인 컴포넌트에 기반한 내용을 제외하고는 그대로 사용할 수 있습니다. (주)커미조아 ComiECAT 에서는 Delphi 에 대한 풍부한 예제를 제공하고 있습니다.

새로운 프로젝트를 시작하기 위해서, 메뉴의 [File]-[New] 명령을 통해, 새로운 응용 프로그램 개발을 시작합니다.

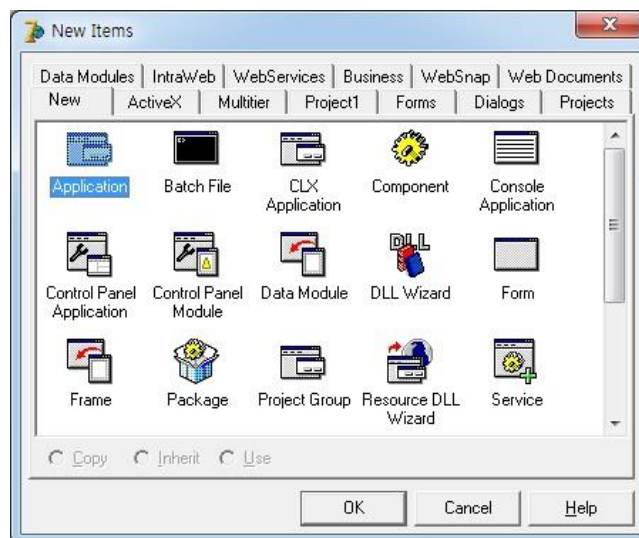


그림 3-42 Delphi 7 의 프로젝트 시작



프로젝트가 시작되면 화면상에 ‘Form1’ 혹은 Delphi IDE 의 Project1 이 나타납니다.

인터페이스 파일을 추가하기 위한 작업으로서 [Project] 메뉴의 [Add to Project] 를 선택합니다.

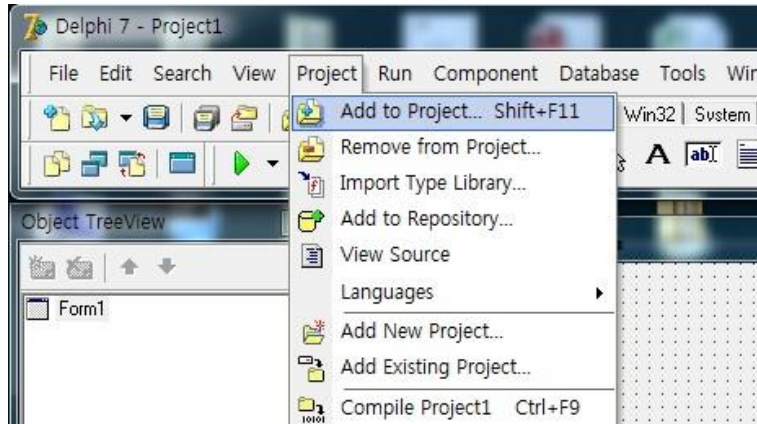


그림 3-43 Delphi 용 인터페이스 파일 추가

(주) 커미조아 ComiECAT 의 공용 인터페이스 정의 파일인 ComiEcatSdk.PAS 파일을 프로젝트에 추가합니다.

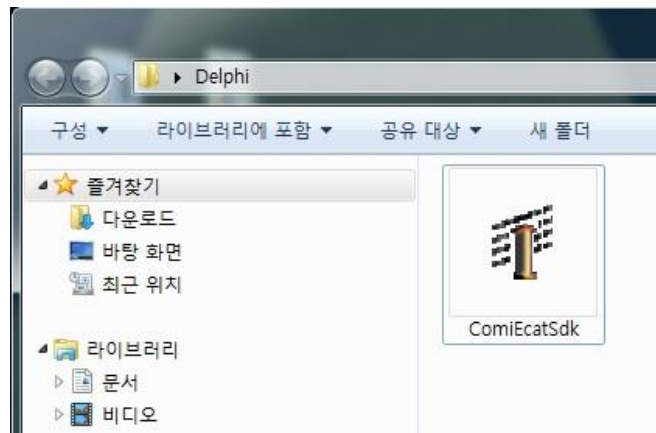


그림 3-44 Delphi 용 인터페이스 파일

Project Manager 를 통해 확인해 보면 ComiEcatSdk.PAS 파일이 프로젝트에 등록된 것을 확인할 수 있습니다.

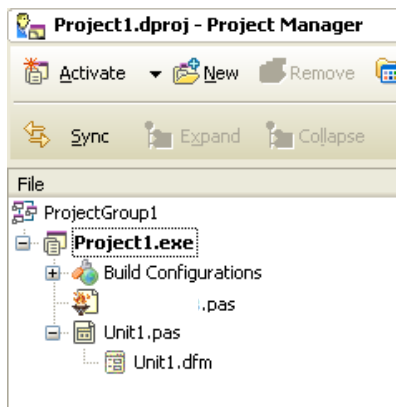



그림 3-45 Delphi 용 인터페이스 파일을 프로젝트 매니저에 추가



	<p>Delphi 의 ComiECAT 인터페이스 파일의 사용에 대한 부가 안내</p>
	<p>저희 (주)커미조아의 ComiECAT 의 ComiEcatSdk.pas 파일은 다른 개발환경(VC++, C++ Builder)와 달리 DLL 의 묵시적인 로드와 언로드가 자동으로 이루어집니다. 이것은 델파이가 가지고 있는 Initialization 과 Finalization 을 이용한 것으로 프로젝트에 별다른 LoadDll 함수 호출 없이 자동으로 DLL 이 로드 됩니다.</p>
	<p>또한 응용프로그램 종료시에 UnloadDll 이 명시적으로 호출되지 않아도, 자동적으로 응용 프로그램이 종료시에 UnloadDll 이 실행됩니다.</p>

실제 구현 부의 코드를 에디터를 통해 확인합니다.

```

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
  //////////////////////////////////////
  //// COMIZOA SDK LIBRARY 들 위한 인터페이스 파일을 사용합니다.
  uses ComiEcatSdk;
  //////////////////////////////////////

  {$R *.dfm}

end.
    
```

그림 3-46 uses 구문을 통해 ComiEcatSdk Unit 사용

위와 같이 implementation 부에 **uses** 를 통해 라이브러리 Unit 을 사용할 수 있도록 반드시 지정해 주십시오. (상단의 uses 에 선언하여도 무방합니다) 이후, 델파이 에서는 다른 개발과 동일하게 DLL 라이브러리를 사용하실 수 있습니다.

### 3.3.5 Visual Basic 개발자를 위한 안내

Visual Basic 6.0 은 마이크로소프트의 컴포넌트 기반 응용프로그램 개발을 위해 태어난 뛰어난 개발 환경입니다. ComiECAT 는 Visual Basic 6.0 를 완벽히 지원하며, 인터페이스 파일을 제공하고 있습니다. Visual Basic 고객(顧客)님들께서도 응용프로그램 개발에 편의성을 드리기위해 저희 (주)커미조아는 언제나 노력하고 있습니다.

실제 Visual Basic 6.0 의 프로젝트 시작 전에, 프로젝트 디렉토리에 (주)커미조아 ComiECAT 인터페이스 파일인 ComiEcatSdk.BAS 파일과 ComiECAT 파일 'ComiEcatSdk.dll' 파일을 반드시 프로젝트 디렉토리에 복사해주시기 바랍니다.

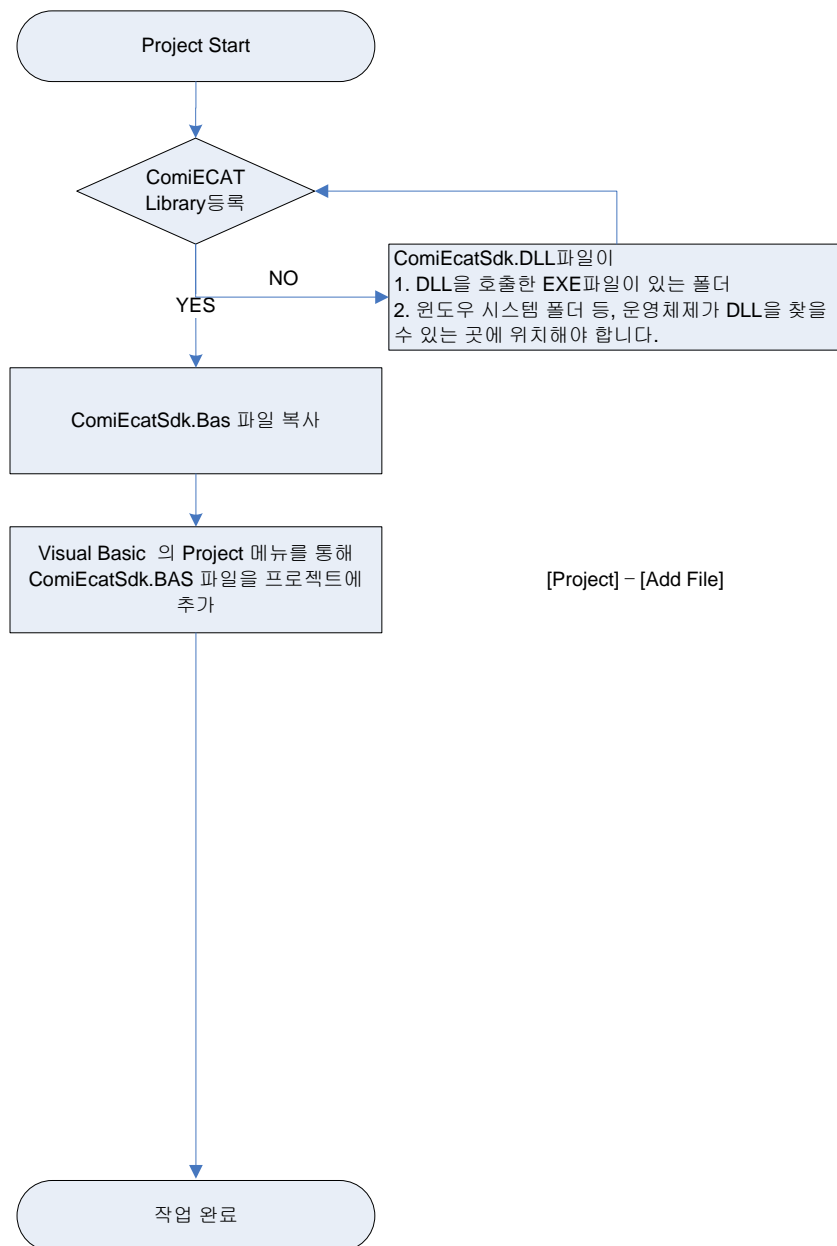


그림 3-47 Visual Basic 에서 ComiECAT 사용 순서도

Visual Basic 을 실행합니다. Visual Basic 이 시작되면 다음과 같은 신규 프로젝트 화면이 표시됩니다.



그림 3-48 새로운 프로젝트 생성

	<p>Visual Basic 에 대한 ComiECAT 지원 사항은 어떠합니까? ComiECAT 은 .NET 환경의 Visual Basic 까지 지원을 하고 있습니다.</p>
	<p>특히 미리 구성된 ComiECAT 인터페이스 파일은 간단히 프로젝트에 추가만 하시면, 바로 ComiECAT 를 사용하실 수 있도록 도움을 드리고 있습니다.</p> <p>만약, Visual Basic 사용 고객(顧客)님께서 ComiECAT 사용에 어려움이 있으시다면, 언제든지 저희 (주)커미조아 고객(顧客) 지원팀으로 연락주시기 바랍니다. 최선을 다해 지원해 드릴 것을 약속드립니다.</p>

신규프로젝트 창에서 ‘Standard EXE’ 를 통해 표준 응용프로그램 개발을 시작합니다. ‘열기’ 버튼을 누릅니다. 만약 이 화면이 나타나지 않으면, 아래의 화면과 같이 ‘File’ 메뉴의 ‘New Project’ 항목을 통해 신규 프로젝트를 시작합니다.

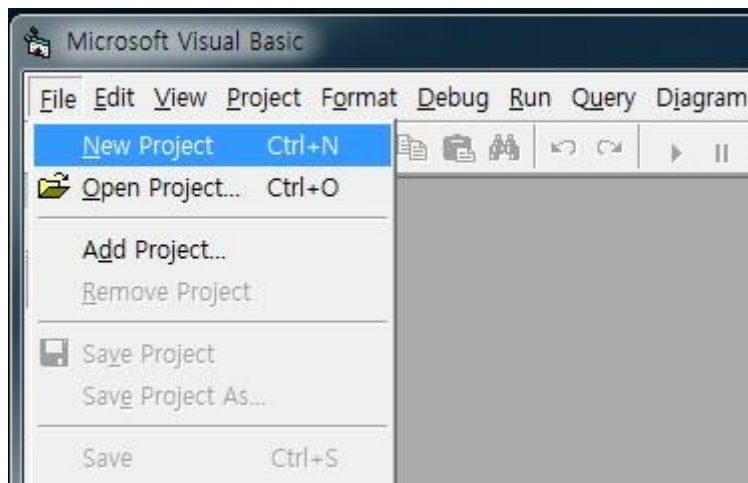


그림 3-49 신규 프로젝트의 시작

시작된 표준 EXE 응용프로그램 개발 메뉴에서 아래 그림과 같이 Project 메뉴를 통해 ‘Add File...’ 을 선택합니다.

그림 3-50 프로젝트에 인터페이스 파일을 추가하기 위한 과정

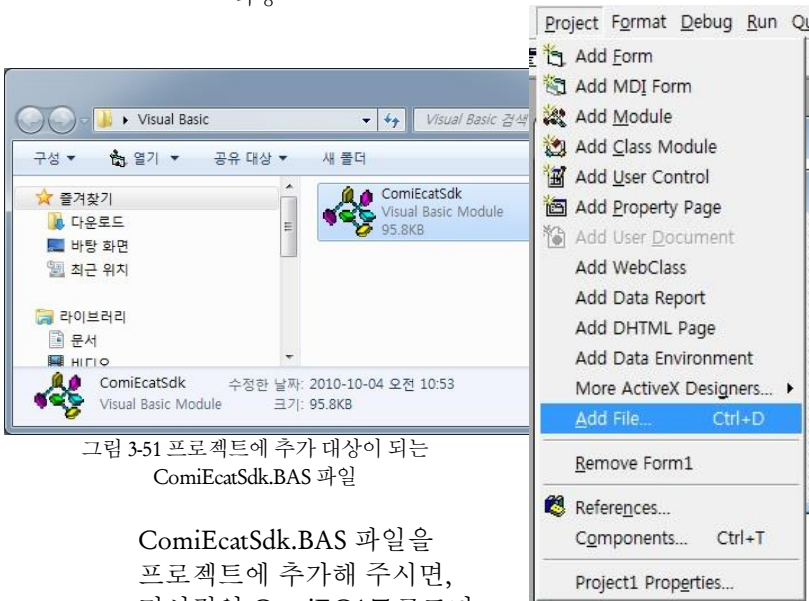


그림 3-51 프로젝트에 추가 대상이 되는 ComiEcatSdk.BAS 파일

ComiEcatSdk.BAS 파일을 프로젝트에 추가해 주시면, 명시적인 ComiECAT 로드가 이루어지게 되며, Visual Basic 의 프로젝트에서 함께 사용하실 수 있습니다.

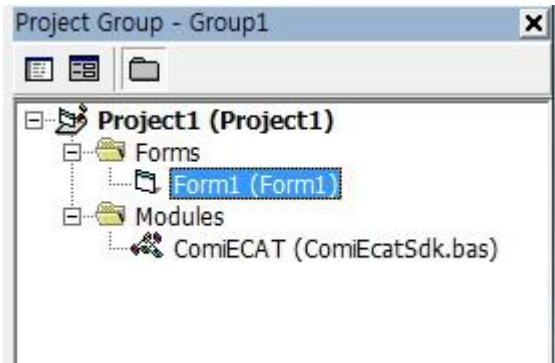


그림 3-52 프로젝트에 모듈로 추가된 ComiEcatSdk.BAS 파일

추가된 인터페이스 파일을 통해 다음과 같이 실제 응용프로그램 구현이 가능합니다.

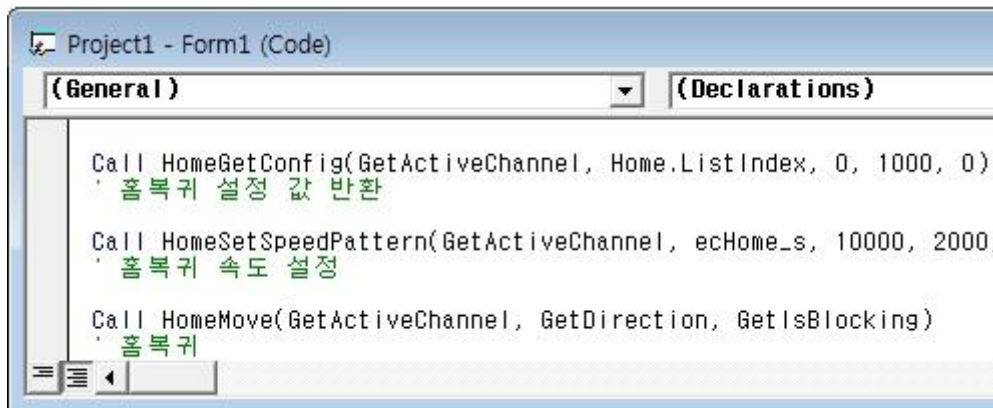


그림 3-53 ComiECAT 을 통한 응용프로그램 구현

**모충**

Visual Basic 의 명시적인 ComiECAT 인터페이스에 대한 부가 안내

ComiECAT에 포함된 Visual Basic 용 인터페이스 파일은 별도의 DLL 라이브러리(ComiECAT)의 로드(Load) 및 언로드(Unload)가 필요 없습니다. 따라서, 고객(顧客)님의 프로젝트의 Form1에 추가된 인터페이스 파일을 통하여, 응용프로그램 구현을 바로 시작하실 수 있습니다.



# ComiECAT Introduction

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

**본** 장에서는 ComiECAT이 제공하는 라이브러리 인터페이스에 대한 자세한 설명(說明)을 안내합니다. ComiECAT은 라이브러리 기능을 보다 강력하고 효율적으로 지원할 수 있는 다양한 런타임(Run-time) 인터페이스와 라이브러리의 다양한 기능을 직관적(直觀的)으로 제공합니다. 본 매뉴얼에서는 ComiECAT에서 제공하는 라이브러리 함수에 대한 설명(說明)을 기능에 따라 그룹별로 수록하였습니다.



## 4 ComiECAT 소개

### 4.1 함수의 명명 규칙

ComiECAT 에서 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.

#### 4.1.1 함수 이름의 접두어

ComiECAT SDK 에서 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

#### 4.1.2 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└
-------------------------------	---	---	---

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

4.1.2.2 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 블록킹(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.

Fast Function 모드로 동작하는 함수들은 ‘\_NB’ 접미어가 붙습니다.

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

4.2 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 7 과 같이 통일하여 표기합니다. 이에 대한 각 컴파일러의 대응되는 데이터형 표기는 표 7 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_succes s	True 나 False 형태의 논리형	bool
t_cmdidx	부호가 없는 4 바이트 정수	unsigned int

표 7 언어독립적 데이터 표기 및 각 언어별 대응 데이터 형

## DLL 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.



## 5 DLL Functions

### 5.1 함수 요약

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

**5.2** 함수 설명

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecDll_Load</p> <p style="margin: 0;">- 라이브러리(Library) 로드</p>	<h3 style="margin: 0;">INFORMATION</h3> <p style="margin: 0;"> DLL Function</p> <p style="margin: 0;"> VC++/BCB/.NET</p> <p style="margin: 0;"> Level 1</p> <p style="margin: 0;"> 위험 요소 없음</p>
<h2 style="margin: 0;">SYNOPSIS</h2> <p style="margin: 0;">□ t_success ecDll_Load()</p>	

## 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 <u>실패</u> 하였음을 의미합니다.
1	DLL 을 <u>성공적</u> 으로 로드하였음을 의미합니다.

## SEE ALSO

\* ecDll\_Unload

## EXAMPLE

```
C/C++

#include "ComiEcatSdk_Api.h"

void StartProgram(void) //반드시 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>	<h3>INFORMATION</h3> <ul style="list-style-type: none"> <li> DLL Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>
<h2>SYNOPSIS</h2> <pre>int t_success ecDll_Unload ()</pre>	

### DESCRIPTION

ComiECAT 을 고객(顧客)님의 응용프로그램의 메모리 공간에서 해제합니다. 이 의미는 이 함수가 호출되는 순간 쥘커미조아의 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 <u>실패</u> 하였음을 의미합니다.
1	DLL 을 <u>성공적</u> 으로 언로드하였음을 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 없습니다. 따라서 다른 예러처리는 필요하지 않습니다.
    ecDll_Unload ();
}
```



<b>NAME</b>  ecDll_IsLoaded - 라이브러리(Library) 로드 상태 확인	<b>INFORMATION</b>  DLL Function
	 VC++/BCB/.NET  Level 1  위험 요소 없음
<b>SYNOPSIS</b> □ BOOL ecDll_IsLoaded ()	

#### 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>성공</u> 적으로 로드하였음을 의미합니다.

#### EXAMPLE

```

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

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

```

## 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.







## 6 General Functions

### 6.1 함수 요약

Summary of Functions
<p>□ t_success ecGn_LoadDevices ([out] t_i32 *ErrCode) 라이브러리가 로드된 상태에서 장치를 로드(Load) 하는 역할을 합니다.</p>
<p>□ t_success ecGn_UnloadDevices ([out] t_i32 *ErrCode) 라이브러리가 로드된 상태에서 장치를 언로드(Unload) 하는 역할을 합니다.</p>
<p>□ BOOL ecGn_IsDevLoaded ([in] t_i32 DevIdx, [out] t_i32 *ErrCode) 시스템에 장치가 로드 되었는지 알 수 있는 역할을 합니다.</p>
<p>□ t_i32 ecGn_GetNumDevices ([out] t_i32 *ErrCode) 현재 시스템에 설치된 장치 개수를 확인하는 역할을 합니다.</p>
<p>□ t_i32 ecGn_GetNumNetworks ([out] t_i32 *ErrCode) 현재 시스템에 설치된 장치들의 네트워크 개수를 확인하는 역할을 합니다.</p>
<p>□ t_success ecGn_GetDevInfo ([in] t_i32 DevIdx, [out] TEcDevInfo *pDevInfo, [out] t_i32 *ErrCode) 장치의 상세 정보를 확인할 수 있는 역할을 합니다.</p>
<p>□ t_i32 ecGn_SetBootWaitMode([in] t_bool IsWaitBootCompt, [out] t_i32 *ErrCode) ecGn_LoadDevices() 함수를 호출했을 때에 ComiECAT 마스터장치의 부팅이 완료되지 않은 경우에 부팅이 완료될 때까지 기다릴지 말지를 설정합니다.</p>
<p>□ BOOL ecGn_GetBootWaitMode([out] t_i32 *ErrCode) ecGn_LoadDevices() 함수를 호출했을 때에 ComiECAT 마스터장치의 부팅이 완료되지 않은 경우에 부팅이 완료될 때까지 기다릴지 말지에 대한 설정을 반환합니다.</p>
<p>□ t_bool ecGn_SetEnableTimerResolSet([in] isEnabled, [out] t_i32 *ErrCode) 라이브러리 내부에서 Sleep()을 적용할 때 타이머 분해능을 1msec 로 조절할지를 설정합니다</p>
<p>□ t_bool ecGn_GetEnableTimerResolSet([out] t_i32 *ErrCode) 라이브러리 내부에서 Sleep()을 적용할 때 타이머 분해능을 1msec 로 조절할지에 대한 설정값을 반환합니다</p>

**6.2** 함수 설명

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecGn_LoadDevices</p> <p style="margin: 0;">- 장치 로드</p>	<h3 style="margin: 0;">INFORMATION</h3> <ul style="list-style-type: none"> <li> General Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>
<h2 style="margin: 0;">SYNOPSIS</h2> <p style="margin: 0;">□ t_success ecGn_LoadDevices([out] t_i32 *ErrCode)</p>	

## 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>성공적으로</u> 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>	<b>INFORMATION</b>
	 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 종료부분에서 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>성공</u> 적으로 로드하였음을 의미합니다.

### SEE ALSO

\* ecGn\_LoadDevices

### EXAMPLE


\* ecGn\_LoadDevices EXMPLE 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 EXMPLE 참고

NAME	I N F O R M A T I O N
ecGn_GetNumDevices - 장치 개수 확인	 General Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

```
□ t_i32 ecGn_GetNumDevices([out] t_i32 *ErrCode)
```

### DESCRIPTION

사용자의 시스템에 현재 연결된 장치의 개수를 확인할 수 있습니다..

### 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 현재 사용자 시스템에 연결된 장치의 개수를 반환합니다.

### EXAMPLE

\* ecGn\_LoadDevices EXMPLE 참고





## NAME


ecGn\_GetNumNetworks  
- 네트워크 개수 확인

### INFORMATION

 General Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_i32 ecGn\_GetNumDevices([out] t\_i32 \*ErrCode)

### DESCRIPTION

사용자의 시스템에 현재 사용할 수 있는 네트워크의 개수를 확인할 수 있습니다..

### 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 현재 사용자 시스템에서 사용할 수 있는 네트워크의 개수를 반환합니다.

### EXAMPLE

\* ecGn\_LoadDevices EXMPLE 참고

<h1>NAME</h1> <p>ecGn_GetDevInfo</p> <p>- 장치 정보 확인</p>	<b>INFORMATION</b>
	<ul style="list-style-type: none"> <li> General Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>

## SYNOPSIS

□ t\_success ecGn\_GetDevInfo([in] t\_i32 DevIdx, TEcDevInfo \*pDevInfo, [out] t\_i32 \*ErrCode)

### DESCRIPTION

장치의 상세 정보를 확인 할 수 있습니다. 장치의 정보는 선언된 구조체를 이용하여 확인하실 수 있습니다..

### PARAMETER

- ▶ DevIdx : 정보를 가져 올 장치 번호.
- ▶ pDevInfo : 구조체 변수, 아래와 같은 정보가 구조체에 포함되어 있다..

Parameter	Meaning
ProdId	Product ID, (0xA550, 0xA551, 0xA552)
PhysId	Board 에 장착되어 있는 스위치에 의해서 결정되는 ID.
NumNet	해당 디바이스 하나에서 제공하는 네트워크 개수
NetIdx	해당 디바이스의 첫번째 네트워크가 전체 네트워크 리스트상에서 차지하는 순서
FwVerMinor	Firmware Version(minor)
FwVerMajor	Firmware Version(major)

- ▶ 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	장치 정보를 가져오는데 실패했음을 의미합니다..
1	장치 정보를 성공적으로 가져왔음을 의미합니다.

### EXAMPLE

\* ecGn\_LoadDevices EXMPLE 참고

<h2>NAME</h2> <p>ecGn_SetBootWaitMode</p> <p>ecGn_GetBootWaitMode</p> <p>- ComiECAT 마스터장치의 부팅완료를 기다리는 속성 설정/반환</p>	<b>INFORMATION</b>
	<p>📁 General Function</p> <hr/> <p>✍ VC++/BCB/.NET</p> <hr/> <p>📖 Level 1</p> <hr/> <p>😊 위험 요소 없음</p>

## SYNOPSIS

- ❑ t\_success ecGn\_SetBootWaitMode([in] t\_bool IsWaitBootCompt, t\_i32 \*ErrCode)
- ❑ t\_bool ecGn\_GetBootWaitMode([out] t\_i32 \*ErrCode)

## DESCRIPTION

ComiECAT 마스터장치는 처음 전원인 인가된 후에 부팅이 완료될 때까지 약 20 초 정도 시간이 걸립니다. ComiECAT 마스터장치의 부팅이 완료되지 않은 상태에서 ComiECAT 라이브러리 함수들이 오동작하게 됩니다.

ecGn\_SetBootWaitMode 함수는 ecGn\_LoadDevices 함수가 실행될 때에 ComiECAT 마스터장치의 부팅이 완료될 때까지 내부에서 대기할지 말지를 설정하는 함수입니다.

## PARAMETER

▶ IsWaitBootCompt : ecGn\_LoadDevices 함수가 실행될 때에 ComiECAT 마스터장치의 부팅이 완료될 때까지 내부에서 대기할지 말지에 대한 속성을 설정. 이 설정은 기본적으로 1로 설정되어 있습니다.

Value	Meaning	Default
0	ecGn_LoadDevices 함수 실행 시에 부팅이 완료되지 않았으면 바로 리턴	
1	ecGn_LoadDevices 함수 실행 시에 부팅이 완료되지 않았으면 부팅이 완료될 때까지 대기	Y

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

## RETURN VALUE

\* ecGn\_SetBootWaitMode 함수의 반환 값: 함수 수행의 성공 여부를 반환합니다.

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





\* ecGn\_SetBootWaitMode 함수의 반환값 : ecGn\_LoadDevices 함수가 실행될 때에 ComiECAT 마스터장치의 부팅 완료를 기다릴 지에 대한 설정값

## REFERENCE

- 이 함수는 `ecGn_LoadDevices` 함수를 호출하기 전에 실행해야만 효과가 있습니다.
- `ecNet_IsBootCompt` 함수를 통해서 마스터 장치의 부팅이 완료되었는지를 체크해볼 수 있습니다.

## SEE ALSO

- \* `ecNet_IsBootCompt`

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecGn_SetEnableTimerResolSet</p> <p style="margin: 0;">ecGn_GetEnableTimerResolSet</p> <p style="margin: 0;">- Sleep 타이머 분해능 설정</p>	<b>INFORMATION</b>
	 General Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecGn\_SetEnableTimerResolSet([in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)
- t\_bool ecGn\_GetEnableTimerResolSet([out] t\_i32 \*ErrCode)

## DESCRIPTION

ecGn\_SetEnableTimerResolSet 라이브러리 내부에서 Sleep()을 적용할 때 타이머 분해능을 1msec 로 조절할지를 설정합니다.

ecGn\_SetBootWaitMode 함수는 라이브러리 내부에서 Sleep()을 적용할 때 타이머 분해능을 1msec 로 조절할지에 대한 설정값을 반환합니다

ecmSxMot\_Move 함수와 같이 함수 내부에서 특정 조건이 만족될 때까지 기다리는 함수에서는 기다리는 동안에 다른 쓰레드나 프로세스에 부하를 주지 않기 위해서 Sleep() 함수를 사용합니다. 그런데 Sleep(1)과 같이 1msec 의 Sleep 을 실행하였더라도 현재 설정된 윈도우 타이머의 분해능에 따라서 Sleep(1)이 최대 16msec 까지 지연을 시킬 수도 있습니다. 이를 방지하기 위해서는 타이머 분해능을 1msec 로 설정해야 합니다.

## PARAMETER

▶ IsEnable : 분해능을 1msec 로 조정하는 기능의 활성화 여부. 이 설정은 기본적으로 1 로 설정되어 있습니다.

Value	Meaning	Default
0	윈도우 타이머 분해능을 조정하지 않음	
1	Sleep()을 실행할 때에 윈도우 타이머를 1msec 로 정함	Y

▶ 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

- 분해능 1msec 설정 기능을 활성화 하였어도 ComiECAT SDK 에서 Sleep()을 사용할 때에만 윈도우 타이머 분해능을 1msec 로 설정하고, Sleep() 루틴이 끝나면 다시 원래의 분해능으로 복원합니다.

# General Utility 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.





## 7 General Utility Functions

### 7.1 함수 요약

Summary of Functions
□ void ecUtl_ShowError_A([in] HWND hParentWnd, [out] char *szFormat, ...) 아스키 스트링 형식의 에러 메시지 대화상자를 디스플레이합니다.
□ void ecUtl_ShowError_W([in] HWND hParentWnd, [out] WCHAR *szFormat, ...) 유니코드 스트링 형식의 에러 메시지 대화상자를 디스플레이합니다.
□ void ecUtl_ShowMessage_A([in] HWND hParentWnd, [out] char *szFormat, ...) 아스키 스트링 형식의 일반 메시지 대화상자를 디스플레이합니다.
□ void ecUtl_ShowMessage_W([in] HWND hParentWnd, [out] WCHAR *szFormat, ...) 유니코드 스트링 형식의 일반 메시지 대화상자를 디스플레이합니다.
□ t_ui32 ecUtl_GetCntDiff_UI32([in] t_ui32 PrvVal, [in] t_ui32 CurVal) 두 개의 부호없는 32 비트 정수의 차이를 계산하여 반환합니다.
□ t_ui16 ecUtl_GetCntDiff_UI16([in] t_ui16 PrvVal, [in] t_ui16 CurVal) 두 개의 부호없는 16 비트 정수의 차이를 계산하여 반환합니다.
□ t_bool ecUtl_IsTimeOut([in] t_ui32 dwStartTime_ms, [in] t_ui32 dwTimeOutVal_ms) 지정된 시간을 기준으로 지정한 제한 시간이 경과되었는지를 반환합니다.

**7.2** 함수 설명

NAME	INFORMATION
ecUtl_ShowError_A	General Function
ecUtl_ShowError_W	VC++/BCB/.NET
- 에러 메시지 대화상자 디스플레이	Level 1
	☺ 위험 요소 없음

**SYNOPSIS**

- ❑ void ecUtl\_ShowError\_A([in] HWND hParentWnd, [out] char \*szFormat, ...)
- ❑ void ecUtl\_ShowError\_W([in] HWND hParentWnd, [out] WCHAR \*szFormat, ...)

**DESCRIPTION**

ecUtl\_ShowError\_A 함수는 아스키 스트링 형식의 에러 메시지를 팝업 대화상자로 디스플레이해주는 함수입니다.

ecUtl\_ShowError\_W 함수는 유니코드 스트링 형식의 에러 메시지를 팝업 대화상자로 디스플레이해주는 함수입니다.

이 함수는 printf() 함수처럼 형식지정자(Format Specifier)를 사용할 수 있다는 것이 특징입니다.

**PARAMETER**

- ▶ hParentWnd : 부모 윈도우 핸들
- ▶ szFormat : 스트링의 형식 지정자(Format Specifier). 이 곳에 사용하는 형식 지정자의 문법은 printf() 함수를 참고하십시오.

**EXAMPLE**


---

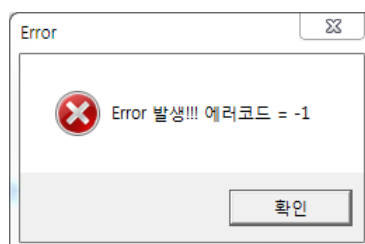
C/C++

---

```
ecUtl_ShowError_A(GetSafeHwnd(), "Error 발생!!! 에러코드 = %d", nErrCode);
```

---

< 실행 결과 >





**NAME**

ecUtl\_ShowMessage\_A  
 ecUtl\_ShowMessage\_W  
 -

**INFORMATION**

General Function  
 VC++/BCB/.NET  
 Level 1  
 위험 요소 없음

**SYNOPSIS**

- void ecUtl\_ShowMessage\_A([in] HWND hParentWnd, [out] char \*szFormat, ...)
- void ecUtl\_ShowMessage\_A([in] HWND hParentWnd, [out] WCHAR \*szFormat, ...)

**DESCRIPTION**

ecUtl\_ShowMessage\_A 함수는 아스키 스트링 형식의 일반 메시지를 팝업 대화상자로 디스플레이해주는 함수입니다.

ecUtl\_ShowMessage\_W 함수는 유니코드 스트링 형식의 일반 메시지를 팝업 대화상자로 디스플레이해주는 함수입니다.

이 함수는 printf() 함수처럼 형식지정자(Format Specifier)를 사용할 수 있다는 것이 특징입니다.

**PARAMETER**

- ▶ hParentWnd : 부모 윈도우 핸들
- ▶ szFormat : 스트링의 형식 지정자(Format Specifier). 이 곳에 사용하는 형식 지정자의 문법은 printf() 함수를 참고하십시오.

**EXAMPLE**


---

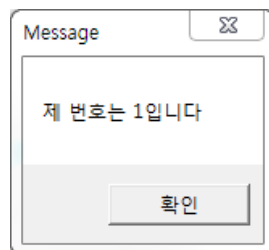
C/C++

---

```
t_i32 no = 1;
ecUtl_ShowMessage_A (GetSafeHwnd(), "제 번호는 %d 입니다", no);
```

---

< 실행 결과 >





## NAME


ecUtl\_GetCntDiff\_UI32  
 ecUtl\_GetCntDiff\_UI16  
 - 두 정수의 차이 값 계산

## INFORMATION

 General Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_ui32 ecUtl\_GetCntDiff\_UI32([in] t\_ui32 PrvVal, [in] t\_ui32 CurVal)

❑ t\_ui16 ecUtl\_GetCntDiff\_UI16([in] t\_ui16 PrvVal, [in] t\_ui16 CurVal)

## DESCRIPTION

ecUtl\_GetCntDiff\_UI32 함수는 두 개의 부호 없는 32 비트 정수의 편차 값을 계산하여 반환합니다.  
 ecUtl\_GetCntDiff\_UI16 함수는 두 개의 부호 없는 16 비트 정수의 편차 값을 계산하여 반환합니다.  
 이 함수들은 정수의 오버플로우(Overflow)까지 감안하여 두 정수의 편차값을 계산합니다.

## PARAMETER

- ▶ PrvVal : 카운트의 이전 값
- ▶ CurVal : 카운트의 현재 값

## RETURN VALUE

\* (CurVal - PrvVal)의 결과값을 반환합니다. 이때 PrvVal 에서 CurVal 로 넘어갈 때 오버플로우가 발생하였어도 문제가 없도록 두 카운트의 차를 반환합니다.

## EXAMPLE

C/C++


```
#include <mmsystem.h>


UINT startTime = timeGetTime();
...
UINT endTime = timeGetTime();
UINT elpsTime = ecUtl_GetCntDiff_UI32(startTime, endTime);
```

**NAME**


ecUtl\_IsTimeOut

- 지정된 시간을 기준으로 지정한 제한 시간이 경과되었는 지를 반환합니다.

**INFORMATION**
 General Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

```
□ t_bool ecUtl_IsTimeOut([in] t_ui32 dwStartTime_ms, [in] t_ui32 dwTimeOutVal_ms)
```

**DESCRIPTION**

지정된 시간으로부터 지정한 제한 시간이 경과되었는 지를 반환해주는 함수입니다. 단, 기준이 되는 시작 시간은 윈도우 API 함수 중에서 `timeGetTime()` 함수를 사용해서 얻어진 값이어야 합니다.

이 함수는 타이머의 오버플로우까지 감안하여 오버플로우가 발생하여도 문제가 없도록 구현되었습니다.

**PARAMETER**

- ▶ `dwStartTime_ms` : 타임아웃을 체크하기 위한 기준(시작) 시간. 이 값은 윈도우 API 함수 중에서 `timeGetTime()` 함수를 사용해서 얻어진 값이어야 합니다.
- ▶ `dwTimeOutVal_ms` : 제한 시간을 msec 단위로 지정합니다.

**RETURN VALUE**

제한 시간이 경과되었는지의 여부를 반환합니다.

Value	Meaning
0	제한 시간이 경과되지 않았음
1	제한 시간이 경과되었음

**EXAMPLE**


---

```
C/C++
```

---

```
#include <mmsystem.h>
```

```
UINT startTime = timeGetTime();
```

```
while(ecUtl_IsTimeOut(startTime, 1000)){ // 1000msec 가 경과되기 전까지 대기...
```

```
    ...
}
```

---

# Dubug logging 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.







## 8 Debug Logging Functions

### 8.1 함수 요약

Summary of Functions	
□ t_success ecDlog_SetFilePath([in] char* szLogFilePath, [out] t_i32 *ErrCode)	라이브러리를 사용자(使用者) 응용프로그램의 사용을 위해 로드(Load) 합니다.
□ t_success ecDlog_GetFilePath([out] char* pszLogFilePath, t_i32 BufLen, t_i32 *ErrCode)	라이브러리를 사용자(使用者) 응용프로그램의 사용 종료(終了)를 위해 언로드(Unload) 합니다.
□ t_success ecDlog_SetLogType([in] int DebugLogType, [out] t_i32 *ErrCode)	라이브러리가 로드 되었는지 알 수 있는 역할을 합니다.
□ INT ecDlog_GetLogType();	라이브러리가 로드 되었는지 알 수 있는 역할을 합니다.
□ t_success ecDlog_SetLogLevel([in] int DebugLogLevel, [out] t_i32 *ErrCode)	라이브러리가 로드 되었는지 알 수 있는 역할을 합니다.
□ t_success ecDlog_GetLogLevel()	라이브러리가 로드 되었는지 알 수 있는 역할을 합니다.



**8.2** 함수 설명

NAME	INFORMATION
ecDlog_SetFilePath	 Debug Logging Function
ecDlog_GetFilePath	 VC++/BCB/.NET
- 로그 파일 생성 위치 설정/반환	 Level 1
	 위험 요소 없음

**SYNOPSIS**

- t\_success ecDlog\_SetFilePath([in] char\* szLogFilePath, [out] t\_i32 \*ErrCode)
- t\_success ecDlog\_GetFilePath([out] char\* pszLogFilePath, [in] t\_i32 Buflen, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

ecDlog\_SetFilePath() 함수는 로그 파일을 생성할 위치를 설정 합니다.  
 ecDlog\_GetFilePath() 함수는 로그 파일을 생성할 위치를 반환 합니다.

**PARAMETER**

- ▶ szLogFilePath : ecDlog\_SetFilePath 함수의 인자이며, 로그 파일의 생성 위치 설정 값입니다.
- ▶ pszLogFilePath : ecDlog\_GetFilePath 함수의 인자이며, 로그 파일의 생성 위치 반환 값입니다.
- ▶ Buflen : ecDlog\_GetFilePath 함수의 인자이며, 로그 파일 위치값을 받아 올 버퍼의 크기 값입니다.
- ▶ 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	장치 정보를 가져오는데 실패했음을 의미합니다..
1	장치 정보를 성공적으로 가져왔음을 의미합니다.

## EXAMPLE

---

```

C/C++

#include "ComiEcatSdk_Api.h"

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

void LogFile_Setting(void)
{
    CHAR IN_File_Path[20] = "Log.txt"; //Log 파일을 저장 할 경로
    CHAR OUT_FILE_PATH[20]; //Log 파일 저장 경로를 받아올 변수

    if(!ecDlog_SetFilePath(IN_File_Path, &Error_Num)) //LOG 저장 파일 설정
    {
        //Log 파일 저장 경로 설정 실패 시 예외 처리
    }

    if(!ecDlog_GetFilePath(OUT_FILE_PATH, 20, &Error_Num)) //LOG 저장 경로 반환
    {
        //Log 파일 저장 경로 로드 실패시 예외 처리
    }

    if(!ecDlog_SetLogType(ecDLOG_TYPE_FILE, &Error_Num)) //Log 타입 설정
    {
        //Log 타입 설정 실패 시 예외 처리
    }

    if(!ecDlog_SetLogLevel(ecDLOG_LEVEL_ERR, &Error_Num)) //Log 레벨 설정
    {
        //Log 레벨 설정 실패 시 예외 처리
    }
}

```

---

<b>NAME</b>	<b>INFORMATION</b>
ecDlog_SetLogType	Debug Logging Function
ecDlog_GetLogType	VC++/BCB/.NET
- 로그 타입 설정/반환	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_success ecDlog\_SetLogType([in] int DebugLogType, [out] t\_i32 \*ErrCode)
- t\_success ecDlog\_GetLogType()

## DESCRIPTION

ecDlog\_SetLogType() 함수는 텍스트 파일, 디버그 스트링 뷰어 등 로그 타입에 대해 설정합니다.  
 ecDlog\_GetLogType() 함수는 텍스트 파일, 디버그 스트링 뷰어 등 로그 타입을 반환 합니다.

## PARAMETER

▶ DebugLogType : ecDlog\_SetLogType 함수의 인자이며, 설정하고자 하는 로그 타입 값입니다.

Type	Meaning
0 or ecDLOG_TYPE_FILE	텍스트 파일에 로그 결과를 기록하는 모드
1 or ecDLOG_TYPE_TRACE	디버그 스트링 뷰어를 통해서 로그 결과를 보여 주는모드
2 or ecDLOG_TYPE_COMITOOl:	(향후 지원 예정)

▶ 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	설정/ 반환에 실패 했음을 의미합니다.
1	설정/ 반환에 성공 했음을 의미합니다.

## EXAMPLE

\* ecDlog\_SetFilePath EXAMPLE 참조

NAME	INFORMATION
ecDlog_SetLogLevel	 Debug Logging Function
ecDlog_GetLogLevel	 VC++/BCB/.NET
- 로그 레벨 설정/반환	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecDlog\_SetLogLevel([in] int DebugLogLevel, [out] t\_i32 \*ErrCode)
- t\_success ecDlog\_GetLogLevel()

## DESCRIPTION

ecDlog\_SetLogLevel() 함수는 로그를 기록할 기준 레벨을 설정합니다.  
 ecDlog\_GetLogLevel() 함수는 로그를 기록할 기준 레벨을 반환합니다.

## PARAMETER

▶ DebugLogLevel : ecDlog\_SetLogType 함수의 인자이며, 설정하고자 하는 로그 레벨 값입니다.

Type	Meaning
0 or ecDLOG_LEVEL_DISABLE	No Debug Logging
1 or ecDLOG_LEVEL_ERR	에러 발생 시 Logging
2 or ecDLOG_LEVEL_CMD	Err 레벨을 포함해 Set 함수 및 각종 CMD 함수 Logging
3 or ecDLOG_LEVEL_GET	CMD 레벨을 포함해 Get 함수 Logging
4 or ecDLOG_LEVEL_ALL	모든 함수에 대해서 Logging

▶ 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	설정/ 반환에 실패 했음을 의미합니다.
1	설정/ 반환에 성공 했음을 의미합니다.

EXAMPLE

\* ecDlog\_SetFilePath EXAMPLE 참조

# Net Interface 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.



## 9 Net Interface Functions





### 9.1 함수 요약

Summary of Functions
<p>□ t_bool ecNet_IsBootCompt([in] t_i32 NetID, [out] t_i32 *ErrCode)</p>
<p>□ t_i32 ecNet_GetDevIdx([in] t_i32 NetID) NetID 의 Device Index 를 반환 합니다.</p>
<p>□ t_i32 ecNet_GetLocNetIdx([in] t_i32 NetID)</p>
<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)</p>
<p>□ t_success ecNet_SetFastFuncType([in] t_i32 NetID, [in] t_ui8 FastCmdType, [out] t_i32 *ErrCode) FF(Fast Function) 함수의 모드를 설정 합니다.</p>
<p>□ t_ui8 ecNet_GetFastFuncType([in] t_i32 NetID, [out] t_i32 *ErrCode) FF(Fast Function) 함수의 모드를 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_success ecNet_SetCfgSlaveCount([in] t_i32 NetID, [in] t_ui32 SlaveCount, [out] t_i32 *ErrCode) 장치에 연결되어 있는 슬레이브의 개수를 설정 합니다.</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>□ t_cmdidx ecNet_SetAlState_FF([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_bool ecNet_SlvComErrSum_SetEnable ([in] t_i32 NetID, [in] t_bool IsEnable, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecNet_SlvComErrSum_GetEnable ([in] t_i32 NetID, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecNet_SlvComErrSum_ClearAll ([in] t_i32 NetID, [out] t_i32 *ErrCode)</p>
<p>□ t_byte ecNet_InPDO_GetData_B ([in] t_i32 NetID, [in] t_i32 OfsPos, [out] t_i32 *ErrCode) InPDO 버퍼의 데이터를 반환 합니다.</p>

<p>□ t_word ecNet_InPDO_GetData_W ([in] t_i32 NetID, [in] t_i32 OfsPos, [out] t_i32 *ErrCode) InPDO 버퍼의 데이터를 반환 합니다.</p>
<p>□ t_dword ecNet_InPDO_GetData_D([in] t_i32 NetID, [in] t_i32 OfsPos, [out] t_i32 *ErrCode) InPDO 버퍼의 데이터를 반환 합니다.</p>
<p>□ t_i32 ecNet_OutPDO_SetData ([in] t_i32 NetID, [in] t_i32 OfsPos, [in] t_ui16 Size, [out] void *pBuf, [out] t_i32 *ErrCode) OutPDO 버퍼에 데이터를 삽입합니다.</p>
<p>□ t_success ecNet_OutPDO_SetData_B ([in] t_i32 NetID, [in] t_i32 OfsPos, [in] t_byte Data, [out] t_i32 *ErrCode) OutPDO 버퍼에 데이터를 삽입합니다.</p>
<p>□ t_success ecNet_OutPDO_SetData_W ([in] t_i32 NetID, [in] t_i32 OfsPos, [in] t-word Data, [out] t_i32 *ErrCode) OutPDO 버퍼에 데이터를 삽입합니다.</p>
<p>□ t_success ecNet_OutPDO_SetData_D ([in] t_i32 NetID, [in] t_i32 OfsPos, [in] t-dword Data, [out] t_i32 *ErrCode) OutPDO 버퍼에 데이터를 삽입합니다.</p>
<p>□ void* ecNet_InPDO_GetBufPtr ([in] t_i32 NetID, [in] t_i32 OfsPos, [out] t_i32 *ErrCode) InPDO 주소 값을 반환합니다.</p>
<p>□ void* ecNet_OutPDO_GetBufPtr ([in] t_i32 NetID, [in] t_i32 OfsPos, [out] t_i32 *ErrCode) OutPDO 주소 값을 반환합니다.</p>



**9.2** 함수 설명

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecNet_IsBootCompt</p> <p style="margin: 0;">- ComiECAT 마스터장치의 부팅 완료 체크</p>	<h3 style="margin: 0;">I N F O R M A T I O N</h3> <p style="margin: 0;"> Net Interface Function</p> <p style="margin: 0;"> VC++/BCB/.NET</p> <p style="margin: 0;"> Level 1</p> <p style="margin: 0;"> 위험 요소 없음</p>
---	---

## SYNOPSIS

```
□ t_bool ecNet_IsBootCompt([in] t_i32 NetID, [out] t_i32 *ErrCode)
```

### DESCRIPTION

지정한 네트워크를 담당하는 ComiECAT 마스터장치의 부팅이 완료되었는지를 체크해보는 함수입니다.

### PARAMETER

- ▶ NetID : 이더넷 Network 번호. 이 함수에서는 여러 개의 마스터 장치가 존재하는 경우에 각 장치를 구분하기 위해서 사용됩니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

### RETURN VALUE

지정한 네트워크를 담당하는 ComiECAT 마스터장치의 부팅이 완료되었는지 여부를 반환합니다.

Value	Meaning
0	부팅이 완료되지 않음
1	부팅이 완료됨


## NAME


ecNet\_GetDevIdx


- 지정한 네트워크를 담당하는 장치 번호 얻기

### INFORMATION

 Net Interface Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_i32 ecNet\_GetDevIdx([in] t\_i32 NetID)

### DESCRIPTION





지정한 NetID 를 담당하는 Device Index 를 반환합니다.

### PARAMETER

▶ NetID : 사용자가 사용 할 Network 번호

### RETURN VALUE

\* 지정한 NetID 를 담당하는 Device Index 를 반환합니다.

NAME	I N F O R M A T I O N
ecNet_GetLocNetIdx - 지정한 네트워크의 로컬 네트워크 번호 반환	 Net Interface Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

□ t\_i32 ecNet\_GetLocNetIdx ([in] t\_i32 NetID)

### DESCRIPTION

지정한 NetID 가 로컬 네트워크 번호로는 몇 번인지를 반환하는 함수입니다.

로컬 네트워크 번호는 하나의 마스터 장치내에서 사용되는 네트워크 번호를 말합니다.

COMI-LX551, COMI-LX552 제품은 하나의 네트워크만 제공하기 때문에 로컬 네트워크가 0 번만 존재하지만, COMI-LX554 제품은 두 개의 네트워크를 제공하여서 0 번과 1 번 두 개의 로컬 네트워크 번호를 제공합니다.

### PARAMETER

▶ NetID : 사용자가 사용 할 Network 번호

### RETURN VALUE

\* NetID 로 지정된 전역 네트워크 번호가 로컬 네트워크 번호로는 몇 번인지를 반환합니다.

<b>NAME</b>	<b>I N F O R M A T I O N</b>
ecNet_GetVerInfo	 Net Interface Function
-	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

```

    □ 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))
    
```

## DESCRIPTION

ComiECAT 프로그램들의 버전을 읽는 함수입니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 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 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.

<h2 style="margin: 0;">NAME</h2> <p style="margin: 10px 0;">ecNet_SetFastFuncType</p> <p style="margin: 10px 0;">ecNet_GetFastFuncType</p> <p style="margin: 10px 0;">- FF(Fast Function) 함수 모드 설정/반환</p>	<h3 style="margin: 0;">INFORMATION</h3> <ul style="list-style-type: none"> <li style="border-bottom: 1px solid black; padding: 5px 0;"> Net Interface Function</li> <li style="border-bottom: 1px solid black; padding: 5px 0;"> VC++/BCB/.NET</li> <li style="border-bottom: 1px solid black; padding: 5px 0;"> Level 1</li> <li style="border-bottom: 1px solid black; padding: 5px 0;"> 위험 요소 없음</li> </ul>
---	--

---

## SYNOPSIS

□ t\_success ecNet\_SetFastFuncType([in] t\_i32 NetID,[in] t\_ui8 FastCmdType, [out] t\_i32 \*ErrCode)

□ t\_ui8 ecNet\_GetFastFuncType([in] t\_i32 NetID, [out] t\_i32 \*ErrCode)

---

### DESCRIPTION

ecNet\_SetFastFuncType() 함수는 ComiECAT 의 FF(Fast Function) 함수들의 모드를 설정합니다.

ecNet\_GetFastFuncType() 함수는 ComiECAT 의 FF(Fast Function) 함수들의 모드를 반환합니다.

	<p>‘Fast Function’ 함수는 함수명에 ‘_FF’ 접미어가 붙은 함수들을 말하며 FF 함수와 관련된 자세한 내용은 ‘4.1.2.1 FAST FUNCTION 모드’ 단원을 참고하시기 바랍니다.</p>
--	--

### PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ FastCmdType : ‘Fast Function’의 형식을 지정합니다.

Value	Meaning	Default
0	Ecat Master 장치에 커맨드를 쓰기만 하고 바로 리턴하는 방식. 함수의 수행 속도가 가장 빠름	Y
1	Ecat Master 장치에 커맨드를 쓰고 커맨드 전달이 잘되었음을 확인한 후에 리턴하는 방식.	

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

### RETURN VALUE

\* ecNet\_SetFastFuncType 함수의 반환 값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecNet\_GetFastFuncType 함수의 반환 값: 현재 설정되어 있는 Fast Function 형식을 반환합니다.


Value	Meaning
0	Ecat Master 장치에 커맨드를 쓰기만 하고 바로 리턴하는 방식. 함수의 수행 속도가 가장 빠름
1	Ecat Master 장치에 커맨드를 쓰고 커맨드 전달이 잘되었음을 확인한 후에 리턴하는 방식.


## 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.

<b>NAME</b>	<b>INFORMATION</b>
ecNet_SetCfgSlaveCount	 Net Interface Function
ecNet_GetCfgSlaveCount	 VC++/BCB/.NET
- 마스터가 관리하는 슬레이브 수 설정 / 반환	 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.

<h1>NAME</h1> <p>ecNet_SetAlState</p> <p>ecNet_GetAlState</p> <p>- 마스터 AL-STATE 설정 / 반환</p>	<b>INFORMATION</b>
	<p> Net Interface Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

## 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. - 슬레이브 레지스터 읽기/쓰기 가능 - SDO 읽기/쓰기 불가능 - PDO 읽기/쓰기 불가능
2 (ecAL_STATE_PREOP)	PreOP State. - 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"> <li>- 슬레이브 레지스터 읽기/쓰기 가능</li> <li>- SDO 읽기/쓰기 가능</li> <li>- Input PDO 읽기 가능</li> <li>- Output PDO 쓰기 불가능</li> </ul>
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	INFORMATION
ecNet_SlvComErrSum_SetEnable	 Net Interface Function
ecNet_SlvComErrSum_GetEnable	 VC++/BCB/.NET
- 통신 에러카운트 누적 기능 활성화	 Level 1
	 위험 요소 없음

## SYNOPSIS

- ❑ t\_bool ecNet\_SlvComErrSum\_SetEnable([in] t\_i32 NetID, [in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)
- ❑ t\_success ecNet\_SlvComErrSum\_GetEnable ([in] t\_i32 NetID, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecNet\_SlvComErrSum\_SetEnable 함수는 마스터에서 각 슬레이브의 통신 에러 카운트값을 누적 관리하는 기능의 활성화 속성을 설정합니다.

ecNet\_SlvComErrSum\_GetEnable 함수는 마스터에서 각 슬레이브의 통신 에러 카운트값을 누적 관리하는 기능의 활성화 속성 값의 현재 설정 상태를 반환합니다.

대부분의 이더넷 슬레이브는 통신 에러를 카운트하는 카운터가 레지스터를 제공합니다. 그런데 그 카운터 레지스터는 1 바이트 크기의 레지스터라서 255 개의 에러만 카운트하고 그 이상이 되면 0 부터 다시 카운트하게 됩니다.

마스터의 통신에러 카운트 누적 기능은 주기적으로 슬레이브들의 에러카운트 값을 읽어서 누적하여서 슬레이브의 카운터 레지스터가 오버플로우되어서 카운트값이 리셋되는 현상을 보완하는 기능입니다.

단, 통신 에러 카운트 누적 기능을 활성화하면 마스터가 주기적으로 슬레이브들의 레지스터를 읽고, 클리어하는 작업을 수행해야 하므로 네트워크에 부하를 가중하게 됩니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ IsEnable : 활성화, 비활성화를 선택합니다.

Value	Meaning
0	에러 카운트 누적 기능을 비활성화합니다.
1	에러 카운트 누적 기능을 활성화합니다.

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

## RETURN VALUE

\* ecNet\_SlvComErrSum\_SetEnable 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecNet\_SlvComErrSum\_GetEnable 함수의 반환값 : 통신에러카운트 누적 기능의 활성화 여부





Value	Meaning
0	에러 카운트 누적 기능을 비활성화됨.
1	에러 카운트 누적 기능을 활성화됨.

## REFERENCE

□ 슬레이브들의 누적 에러카운트값은 ecSlv\_GetComErrSum 함수와 ecSlv\_GetComErrSum\_A 함수를 통해서 읽을 수 있습니다.

## SEE ALSO

\* ecNet\_SlvComErrSum\_ClearAll, ecSlv\_GetComErrSum, ecSlv\_GetComErrSum\_A

<h2>NAME</h2> <p>ecNet_SlvComErrSum_ClearAll</p> <p>-</p>	<b>INFORMATION</b>
	 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 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.

NAME	INFORMATION
ecNet_InPDO_GetData ecNet_InPDO_GetData_B ecNet_InPDO_GetData_W ecNet_InPDO_GetData_D - InPDO 버퍼 데이터 반환	Net Interface Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_i32 ecNet\_InPDO\_GetData ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [in] t\_ui16 Size, [out] void \*pBuf, [out] t\_i32 \*ErrCode)
- t\_byte ecNet\_InPDO\_GetData\_B ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [out] t\_i32 \*ErrCode)
- t\_word ecNet\_InPDO\_GetData\_W ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [out] t\_i32 \*ErrCode)
- t\_dword ecNet\_InPDO\_GetData\_D ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수들은 네트워크 전역으로 관리되는 Input PDO 버퍼의 데이터를 반환하는 함수들입니다.

Input PDO 데이터는 슬레이브로부터 마스터에게 주기적으로 전달되는 입력 데이터를 말합니다. 이것은 슬레이브 입장에서 전송하는 것이므로 Tx PDO 라고도 합니다.

ecNet\_InPDO\_GetData 함수는 가변 크기의 데이터를 읽을 수 있는 함수입니다.

ecNet\_InPDO\_GetData\_B 함수는 1 바이트에 해당하는 데이터를 읽는 함수입니다.

ecNet\_InPDO\_GetData\_W 함수는 2 바이트에 해당하는 데이터를 읽는 함수입니다.

ecNet\_InPDO\_GetData\_D 함수는 4 바이트에 해당하는 데이터를 읽는 함수입니다.

## PARAMETER

- ▶ NetID : 이더넷 Network 번호
- ▶ OfsPos : 읽고자 하는 데이터의 Input PDO 버퍼상의 바이트 오프셋 위치
- ▶ Size : 버퍼에서 읽을 데이터의 크기

▶ pBuf : 데이터를 저장 할 버퍼

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

## RETURN VALUE

ecNet\_InPDO\_GetData 함수는 실제로 버퍼를 통해서 반환되는 데이터의 크기를 반환합니다.

ecNet\_InPDO\_GetData\_B 함수는 1 바이트의 데이터를 반환합니다.

ecNet\_InPDO\_GetData\_W 함수는 2 바이트의 데이터를 반환합니다.


ecNet\_InPDO\_GetData\_D 함수는 4 바이트의 데이터를 반환합니다.




## NAME


ecNet\_OutPDO\_SetData  
 ecNet\_OutPDO\_SetData\_B  
 ecNet\_OutPDO\_SetData\_W  
 ecNet\_OutPDO\_SetData\_D  
 - OutPDO 버퍼에 데이터 삽입

## INFORMATION

 Net Interface Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_i32 ecNet\_OutPDO\_SetData ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [in] t\_ui16 Size, [out] void \*pBuf, [out] t\_i32 \*ErrCode)
- t\_success ecNet\_OutPDO\_SetData\_B ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [in] t-byte Date [out] t\_i32 \*ErrCode)
- t\_success ecNet\_OutPDO\_SetData\_W ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [in] t-word Data [out] t\_i32 \*ErrCode)
- t\_success ecNet\_OutPDO\_SetData\_D ([in] t\_i32 NetID, [in] t\_i32 OfsPos, [in] t-dword Data [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수들은 네트워크 전역으로 관리되는 Output PDO 버퍼에 데이터를 쓰는 함수들입니다.

Output PDO 데이터는 마스터가 슬레이브에게 주기적으로 전달하는 출력 데이터를 말합니다. 이것은 슬레이브 입장에서는 수신하는 것이므로 Rx PDO 라고도 합니다.

ecNet\_OutPDO\_SetData 함수는 가변 크기의 데이터를 쓰는 함수입니다.

ecNet\_OutPDO\_SetData\_B 함수는 1 바이트에 해당하는 데이터를 쓰는 함수입니다.

ecNet\_OutPDO\_SetData\_W 함수는 2 바이트에 해당하는 데이터를 쓰는 함수입니다.

ecNet\_OutPDO\_SetData\_D 함수는 4 바이트에 해당하는 데이터를 쓰는 함수입니다.

## PARAMETER

- ▶ NetID : 이더넷 Network 번호
- ▶ OfsPos : 쓰고자 하는 데이터의 Output PDO 버퍼상의 바이트 오프셋 위치
- ▶ Size : 쓰고자 하는 데이터의 크기
- ▶ pBuf : 가변 크기의 데이터를 쓰고자 할 때 데이터를 전달하는 버퍼
- ▶ Data : 쓰고자 하는 데이터 값

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

### RETURN VALUE

\* **ecNet\_OutPDO\_SetData** 함수의 반환값 : 실제 쓰기에 성공한 데이터의 크기

\* 나머지 함수들의 반환값 : 함수 수행의 성공 여부를 반환합니다.


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


## NAME


ecNet\_InPDO\_GetBufPtr  
- InPDO 버퍼 주소 값 반환

### INFORMATION

 Net Interface Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```
void* ecNet_InPDO_GetBufPtr ([in] t_i32 NetID[in] t_i32 OfsPos,[out] t_i32 *ErrCode)
```

### DESCRIPTION

이 함수는 Input PDO 영역에서 지정한 바이트 오프셋에 해당하는 메모리의 주소 값을 반환합니다.

### PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ OfsPos : 읽고자 하는 데이터의 Input PDO 버퍼 상의 바이트 오프셋 위치
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

### RETURN VALUE


\* InPDO 의 주소 값은 void\* 형식으로 반환 합니다. 따라서 원하는 데이터 형에 맞게 형변환을 해서 사용하셔야 합니다.


## NAME


ecNet\_OutPDO\_GetBufPtr  
- OutPDO 버퍼 주소 값 반환

### INFORMATION

 Net Interface Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```
void* ecNet_OutPDO_GetBufPtr ([in] t_i32 NetID,[in] t_i32 OfsPos,[out] t_i32
*ErrCode)
```

## DESCRIPTION

이 함수는 Output PDO 영역에서 지정한 바이트 오프셋에 해당하는 메모리의 주소 값을 반환합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ OfsPos : Output PDO 버퍼 내에서의 바이트 오프셋 위치
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* OutPDO 의 주소 값은 void\* 형식으로 반환 합니다. 따라서 원하는 데이터 형에 맞게 형변환을 해서 사용하셔야 합니다.

## Slave I/f 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.



## 10 Slave I/F Functions

### 10.1 함수 요약





Summary of Functions
<p>□ t_success ecSlv_GetProdInfo_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] TEcSlvProdInfo *pProdInfoBuf, [out] t_i32 *ErrCode)            슬레이브 인덱스에 맞는 슬레이브의 Product ID 정보를 반환합니다.</p>
<p>□ t_success ecSlv_GetProdInfo ([in] t_i32 NetID, [in] t_i32 SlvPhysAddr, [in] TEcSlvProdInfo *pProdInfoBuf, [out] t_i32 *ErrCode)            물리적 주소에 맞는 슬레이브의 Product ID 정보를 반환합니다.</p>
<p>□ t_success ecSlv_GetTypeInfo_A ([in] t_i32 NetID, [in] t_i32 SlaveIndex, [in] TEcSlvProdInfo *pProdInfoBuf, [out] t_i32 *ErrCode)            슬레이브 인덱스에 맞는 슬레이브의 장치 형식을 반환합니다.</p>
<p>□ t_success ecSlv_GetTypeInfo([in] t_i32 NetID, [in] t_i32 SlvPhysAddr, [in] TEcSlvProdInfo *pProdInfoBuf, [out] t_i32 *ErrCode)            물리적 주소에 맞는 슬레이브의 장치 형식을 반환합니다.</p>
<p>□ EEcAlState ecSlv_GetAlState([in] t_i32 NetID, [in] t_i32 SlvPhysAddr, [out] t_i32 *ErrCode)            물리적 주소에 맞는 Slave 의 AlState 를 반환합니다.</p>
<p>□ EEcAlState ecSlv_GetAlState_A([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] t_i32 *ErrCode)            슬레이브 인덱스에 맞는 Slave 의 AlState 를 반환합니다.</p>
<p>□ t_i32 ecSlv_SlvIdx2PhysAddr([in] t_i32 NetID, [in] t_ui16 SlaveIndex, [out] t_i32 *ErrCode)            슬레이브의 인덱스를 물리적 주소로 변경해줍니다.</p>
<p>□ t_i32 ecSlv_PhysAddr2SlvIdx([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [out] t_i32 *ErrCode)            물리적 주소를 슬레이브 인덱스로 변경해줍니다.</p>
<p>□ t_i32 ecSlv_ReadReg([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 RegAddr, [in] t_i32 DataSize, [out] void *pBuf, [out] t_i32 *ErrCode)            물리적주소에 해당하는 슬레이브의 레지스터 값 확인</p>
<p>□ t_i32 ecSlv_ReadReg_A([in] t_i32 NetID, [in] t_i32 SlaveIdx, [in] t_i32 RegAddr, [in] t_i32 DataSize, [out] void *pBuf, [out] t_i32 *ErrCode)            슬레이브 인덱스에 해당하는 슬레이브의 레지스터 값 확인</p>
<p>□ t_cmdidx ecSlv_WriteReg([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 RegAddr, [in] t_i32 DataSize, [in] void *pData, [out] t_i32 *ErrCode)            물리적주소에 해당하는 슬레이브의 레지스터 값 저장</p>
<p>□ t_cmdidx ecSlv_WriteReg_FF([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 RegAddr, [in] t_i32 DataSize, [in] void *pData, [out] t_i32 *ErrCode)            물리적주소에 해당하는 슬레이브의 레지스터 값 저장</p>

<p>□ t_cmdidx ecSlv_WriteReg_A([in] t_i32 NetID, [in] t_i32 SlaveIdx, [in] t_i32 RegAddr, [in] t_i32 DataSize, [in] void *pData, [out] t_i32 *ErrCode) 슬레이브 인덱스에 해당하는 슬레이브의 레지스터 값 저장</p>
<p>□ t_cmdidx ecSlv_WriteReg_A_FF([in] t_i32 NetID, [in] t_i32 SlaveIdx, [in] t_i32 RegAddr, [in] t_i32 DataSize, [in] void *pData, [out] t_i32 *ErrCode) 슬레이브 인덱스에 해당하는 슬레이브의 레지스터 값 저장</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) 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) 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) 물리적 주소의 슬레이브 Sdo 값 저장</p>
<p>□ t_cmdidx ecSlv_WriteCoeSdo_FF([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 Index, [in] t_i32 SubIdx, [in] t_i32 IsComptAccess, [in] t_i32 DataSize, [in] void* pData, [out] t_i32 *ErrCode) 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) 슬레이브 인덱스의 슬레이브 Sdo 값 확인</p>
<p>□ t_cmdidx ecSlv_WriteCoeSdo_A_FF([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) 슬레이브 인덱스의 슬레이브 Sdo 값 확인</p>
<p>□ void* ecSlv_InPDO_GetBufPtr([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 InPdoUnitIdx, [out] t_i32 *ErrCode) 해당 슬레이브의 Input PDO 버퍼의 시작 주소 반환</p>
<p>□ t_i32 ecSlv_InPDO_GetBufLen([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 InPdoUnitIdx, [out] t_i32 *ErrCode) 해당 슬레이브의 Input PDO 버퍼의 길이 반환</p>
<p>□ void* ecSlv_OutPDO_GetBufPtr([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 OutPdoUnitIdx, [out] t_i32 *ErrCode) 해당 슬레이브의 Output PDO 버퍼의 시작 주소 반환</p>
<p>□ t_i32 ecSlv_OutPDO_GetBufLen([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 OutPdoUnitIdx, [out] t_i32 *ErrCode) 해당 슬레이브의 Output PDO 버퍼의 길이 반환</p>
<p>□ t_success ecSlv_OutPDO_SetInitValMode([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 OutPdoUnitIdx, [in] t_i32 OutPDOInitMode, [out] t_i32 *ErrCode) Output PDO 데이터의 초기값 설정 모드 속성을 설정</p>

<p>□ t_i32 ecSlv_OutPDO_GetInitValMode([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 OutPdoUnitIdx, [out] t_i32 *ErrCode); Output PDO 데이터의 초기값 설정 모드 속성을 반환</p>
<p>□ t_success ecSlv_OutPDO_SetInitVal([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 OutPdoUnitIdx, [in] t_i32 OfsPos, [in] t_i32 DataSize, [in] void *pData, [out] t_i32 *ErrCode) Output PDO 데이터의 초기값을 설정</p>
<p>□ t_i32 ecSlv_OutPDO_GetInitVal([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8 OutPdoUnitIdx, [in] t_i32 OfsPos, [in] t_i32 BufSize, [in] void *pBuf, [out] t_i32 *ErrCode) Output PDO 데이터의 초기값을 반환</p>
<p>□ t_ui16 ecSlv_GetComErrSum([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [out] t_i32 *ErrCode) 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) 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) 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) 슬레이브의(순서 지정 방식) 누적 통신 에러 카운트 값을 0 으로 리셋</p>
<p>□ t_success ecSlv_SetAlState_A ([in] t_i32 NetID, [in] t_ui16 SlaveIdx, [in] EEcAlState AlState, [out] t_i32 *ErrCode) 슬레이브 인덱스에 해당하는 슬레이브의 AlState 변경</p>



**10.2** 함수 설명

NAME	I N F O R M A T I O N
ecSlv_GetProdInfo	 Slave I/F Function
ecSlv_GetProdInfo_A	 VC++/BCB/.NET
- 슬레이브 Product ID 정보 확인	 Level 1
	 위험 요소 없음

**SYNOPSIS**

- t\_success ecSlv\_GetProdInfo([in] t\_i32 NetID, [in] t\_i32 SlavePhysAddr, [out] TecSlvProdInfo \*pProdInfoBuf, [out] t\_i32 \*ErrCode)
- t\_success ecSlv\_GetProdInfo\_A([in] t\_i32 NetID, [in] t\_i32 SlaveIndex, [out] TecSlvProdInfo \*pProdInfoBuf, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

ecSlv\_GetProdInfo (),ecSlv\_GetProdInfo\_A() 함수는 하나의 슬레이브에 대한 Product ID 정보를 반환 합니다.

**PARAMETER**

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소
- ▶ SlaveIndex : 슬레이브 번호
- ▶ pProdInfoBuf : Product ID 에 대한 구조체 변수.

Parameter	Meaning
VendId	Slave 제조사 ID
ProdCode	Slave 를 구별하는 Code Number
RevNo	Reserve Number.
SerNo	Service Number

- ▶ 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	정보를 가져오는데 실패 했음을 의미합니다.
1	정보를 가져오는데 성공 했음을 의미합니다.

EXAMPLE

```

C/C++

#include "ComiEcatSdk_Api.h"

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

void Slave_Load(void)
{
    TEcSlvProdInfo pProdInfoBuf // Product ID 에 대한 구조체 변수.

    TEcSlvTypeInfo pTypeInfoBuf // 장치 형식에 대한 구조체 변수

    t_ui8 Alstate = 0;
    //Device_Info 변수는 6 장 참고

    if(!ecSlv_GetProdInfo(Device_Info.NetIdx, SLAVE_PHYADDR, &pProdInfoBuf,
    &Error_Num))
        // SLAVE_PHYADDR -> 슬레이브의 물리적 주소를 입력
        // ecSlv_GetProdInfo_A 함수를 이용하여 물리적 주소 대신 슬레이브 번호를 입력하여
        사용 가능
        // 이 함수를 이용하여 슬레이브의 Product Code 구조체를 얻어 옴

        {
            //슬레이브 Product ID 정보 확인 실패 시 예외 처리
        }

    if(!ecSlv_GetTypeInfo(Device_Info.NetIdx, SLAVE_PHYADDR,&pTypeInfoBuf,
    &Error_Num))
        // SLAVE_PHYADDR -> 슬레이브의 물리적 주소를 입력
        // ecSlv_GetTypeInfo_A 함수를 이용하여 물리적 주소 대신 슬레이브 번호를
        입력하여
        사용 가능
        {
            //슬레이브 장치 정보 로드 실패시 예외 처리
        }

    else
    {
        //*****
        장치 정보 로드 성공 시 pTypeInfoBuf 구조체에는 각 각 연결 된 모듈의 시작
        채널 및 채널 개수 구조체 형태로 저장
        *****//
    }
}

```

```

Alstate = ecSlv_GetAlState(Device_Info.NetIdx, SLAVE_PHYADDR, &Error_Num);
//물리적 주소 위치에 있는 슬레이브 AlState 확인
}
    
```

NAME	INFORMATION
ecSlv_GetTypeInfo	Slave I/F Function
ecSlv_GetTypeInfo_A	VC++/BCB/.NET
- 슬레이브 Product ID 정보 확인	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_success ecSlv\_GetProdInfo([in] t\_i32 NetID, [in] t\_i32 SlavePhysAddr, [out] TecSlvTypeInfo \*pTypeInfoBuf, [out] t\_i32 \*ErrCode)
- t\_success ecSlv\_GetProdInfo\_A([in] t\_i32 NetID, [in] t\_i32 SlaveIndex, [out] TecSlvTypeInfo \*pTypeInfoBuf, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecSlv\_GetTypeInfo (),ecSlv\_GetTypeInfo\_A() 함수는 슬레이브 모듈의 장치 형식에 대한 정보를 반환 합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ SlaveIndex : 슬레이브 번호(슬레이브가 연결되어 있는 순서를 인덱스로 구분)
- ▶ pTypeInfoBuf : 장치 형식에 대한 구조체 변수.

Struct	Parameter
Servo	StaAxisNo(시작 채널 번호), NumAxes(채널 개수)
DI	StaGlobChannel(시작채널번호), NumChannels(채널 개수)
DO	StaGlobChannel(시작채널번호), NumChannels(채널 개수)
AI	StaGlobChannel(시작채널번호), NumChannels(채널 개수)
AO	StaGlobChannel(시작채널번호), NumChannels(채널 개수)

- ▶ 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	정보를 가져오는데 실패 했음을 의미합니다.
1	정보를 가져오는데 성공 했음을 의미합니다.

EXAMPLE

\* ecSlv\_GetProdInfo 참조

NAME	I N F O R M A T I O N
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>1 ( ecAL_STATE_INITIAL)</p>	<p>Initial State.</p> <ul style="list-style-type: none"> <li>- 슬레이브 레지스터 읽기/쓰기 가능</li> <li>- SDO 읽기/쓰기 불가능</li> <li>- PDO 읽기/쓰기 불가능</li> </ul>
<p>2 ( ecAL_STATE_PREOP)</p>	<p>PreOP State.</p> <ul style="list-style-type: none"> <li>- 슬레이브 레지스터 읽기/쓰기 가능</li> <li>- SDO 읽기/쓰기 가능</li> <li>- PDO 읽기/쓰기 불가능</li> </ul>
<p>4 ( ecAL_STATE_SAFEOP)</p>	<p>SafeOP State.</p> <ul style="list-style-type: none"> <li>- 슬레이브 레지스터 읽기/쓰기 가능</li> <li>- SDO 읽기/쓰기 가능</li> <li>- Input PDO 읽기 가능</li> <li>- Output PDO 쓰기 불가능</li> </ul>
<p>8 ( ecAL_STATE_OP)</p>	<p>OP State. 모든 종류의 데이터 읽기/쓰기 가능</p>

NAME	INFORMATION
ecSlv_SlvIdx2PhysAddr ecSlv_PhysAddr2SlvIdx - 슬레이브 인덱스를 물리적 주소로 변경 - 물리적 주소를 슬레이브 인덱스로 변경	Slave I/F Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_i32 ecSlv\_SlvIdx2PhysAddr ([in] t\_i32 NetID, [in] t\_i32 SlaveIndex, [out] t\_i32 \*ErrCode)
- t\_i32 ecSlv\_PhysAddr2SlvIdx ([in] t\_i32 NetID, [in] t\_i32 SlvPhysAddr, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecSlv\_SlvIdx2PhysAddr () 함수는 슬레이브의 인덱스(연결 순서)를 슬레이브의 물리적 주소로 변환하여 주는 함수입니다.

ecSlv\_PhysAddr2SlvIdx() 함수는 슬레이브의 물리적 주소를 슬레이브 인덱스(연결 순서)로 변환하여 주는 함수입니다.

## 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

\* ecSlv\_SlvIdx2PhysAddr(): 이 리턴값은 슬레이브의 물리적 주소 값을 반환 합니다.

\* ecSlv\_PhysAddr2SlvIdx(): 이 리턴값은 슬레이브 인덱스(연결 순서) 값을 반환 합니다.

## EXAMPLE

NAME	I N F O R M A T I O N
ecSlv_ReadReg	 Slave I/F Function
ecSlv_ReadReg_A	 VC++/BCB/.NET
- 슬레이브의 Register 읽기	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_i32 ecSlv\_ReadReg ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 RegAddr, [in] t\_i32 DataSize, [out] void \*pBuf, [out] t\_i32 \*ErrCode)
- t\_i32 ecSlv\_ReadReg\_A ([in] t\_i32 NetID, [in] t\_ui16 SlaveIndex, [in] t\_i32 RegAddr, [in] t\_i32 DataSize, [out] void \*pBuf, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 두 함수는 슬레이브의 ESC(EtherCAT Slave Controller)에서 제공하는 레지스터를 읽어서 반환하는 함수들입니다.

ecSlv\_ReadReg 함수는 슬레이브를 지정할 때 주소값을 사용하여 슬레이브를 지정합니다.

ecSlv\_ReadReg\_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
- ▶ RegAddr : ESC 레지스터 주소
- ▶ DataSize : 읽을 데이터의 크기(바이트 단위)
- ▶ pBuf: 읽은 데이터를 전달 받을 버퍼 포인터
- ▶ 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 읽은 데이터의 크기를 반환하는 것입니다. 반환 값이 음수이면 에러코드를 반환하는 것입니다.



NAME	INFORMATION
ecSlv_WriteReg	 Slave I/F Function
ecSlv_WriteReg_A	 VC++/BCB/.NET
- 슬레이브의 Register 쓰기	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_i32 ecSlv\_WriteReg ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 RegAddr, [in] t\_i32 DataSize, [in] void \*pBuf, [out] t\_i32 \*ErrCode)
- t\_i32 ecSlv\_WriteReg\_A ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 RegAddr, [in] t\_i32 DataSize, [in] void \*pBuf, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 두 함수는 슬레이브의 ESC(EtherCAT Slave Controller)에서 제공하는 레지스터에 값을 쓰는 함수들입니다.

ecSlv\_WriteReg 함수는 슬레이브를 지정할 때 주소값을 사용하여 슬레이브를 지정합니다.





ecSlv\_WriteReg\_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
- ▶ RegAddr : ESC 레지스터 주소
- ▶ DataSize : 쓰기할 데이터의 크기(바이트 단위)
- ▶ pBuf: 쓰기할 데이터를 전달 받을 버퍼 포인터
- ▶ 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 쓰기할 데이터의 크기를 반환하는 것입니다. 반환 값이 음수이면 에러코드를 반환하는 것입니다.

<b>NAME</b>  ecSlv_ReadCoeSdo  ecSlv_ReadCoeSdo_A  - SDO 데이터 읽기	<b>INFORMATION</b>
	 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 SlaveIdx, [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.
- ▶ **pBuf**: 읽은 데이터를 전달 받을 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 읽은 데이터의 크기를 반환하는 것입니다. 반환 값이 음수이면 에러코드를 반환하는 것입니다.

REFERENCE

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

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_i32 errCode;
t_ui32 maxMotorSpeed;
ecSlv_ReadCoeSdo (NETID, SLAVE_ADDR, 0x6080, 0, false, 4, &maxMotorSpeed, &errCode);
printf("maxMotorSpeed 값은 %d 입니다!", maxMotorSpeed);
```

NAME	INFORMATION
ecSlv_WriteCoeSdo ecSlv_WriteCoeSdo_A - SDO 데이터 쓰기	Slave I/F Function VC++/BCB/.NET 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 SlaveIdx, [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 • 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\_InPDO\_GetBufPtr  
- InPDO 의 포인터 주소 반환

## INFORMATION

-  Slave I/F Function
-  VC++/BCB/.NET
-  Level 1
-  위험 요소 없음

## SYNOPSIS

```
void* ecSlv_InPDO_GetBufPtr ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8
InPdoUnitIdx, [out] t_i32 *ErrCode)
```

## DESCRIPTION

해당 슬레이브의 Input PDO 데이터가 제공되는 메모리 주소값(버퍼 포인터)을 반환합니다. 사용자는 이 주소값을 이용해서 해당 슬레이브에서 제공하는 Input PDO 데이터를 직접 액세스(Access)할 수 있습니다.

Input PDO 데이터는 슬레이브로부터 마스터에게 주기적으로 전달되는 입력 데이터를 말합니다. 이 것은 슬레이브 입장에서 전송하는 것이므로 Tx PDO 라고도 합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ InPdoUnitIdx: 이 값은 해당 슬레이브가 Input PDO 용 Sync. Manager(SM)를 여러 개 제공할 때에 각 SM 을 구분하기 위한 매개변수입니다. 일반적으로 대부분의 슬레이브 장치들은 Input SM 을 하나만 사용하므로 특별한 경우가 아니면 이 값은 0으로 설정하면 됩니다. Sync. Manager 는 ESC(EtherCAT Slave Controller)에서 제공하는 메모리 관리자인데 그 내용을 설명하기에는 본 설명서의 범주를 벗어나므로 설명을 생략합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 해당 슬레이브의 Input PDO 데이터가 제공되는 메모리 주소값(버퍼 포인터)을 반환합니다.

## EXAMPLE

< 예제 설명 >

슬레이브 주소가 0x201 인 32 채널 디지털입력 슬레이브 모듈이 있다고 가정합니다. 또한 해당 슬레이브는 32 비트(4 바이트)의 Input PDO 데이터를 제공하며 각각의 비트는 디지털입력 채널에 대응되는 상태값이라 가정합니다. 이 때 해당 슬레이브의 Input PDO 버퍼를 직접 액세스해서 각 채널의 디지털 접점 값을 사용하는 예제입니다.

---

< 예제 코드 >





```
#define NETID          0
#define SLAVE_ADDR    0x201

t_ui32 *pdwPtr;
pdwPtr = ecSly_InPDO_GetBufPtr (NETID, SLAVE_ADDR, 0, NULL);

if(pdwPtr != NULL){
    printf("D/I CH0 Value = %d\n", (*pdwPtr) & 0x1);
    printf("D/I CH1 Value = %d\n", ((*pdwPtr) >> 1) & 0x1);
    printf("D/I CH2 Value = %d\n", ((*pdwPtr) >> 2) & 0x1);
    printf("D/I CH3 Value = %d\n", ((*pdwPtr) >> 3) & 0x1);
}
```

---



NAME	I N F O R M A T I O N
ecSlv_InPDO_GetBufLen - InPDO 의 길이 반환	<div style="border-bottom: 1px solid black; padding: 2px;">  Slave I/F Function         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  VC++/BCB/.NET         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  Level 1         </div> <div style="padding: 2px;">  위험 요소 없음         </div>

## SYNOPSIS

□ t\_i32 ecSlv\_InPDO\_GetBufLen ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_ui8 InPdoUnitIdx, [out] t\_i32 \*ErrCode)

### DESCRIPTION





해당 슬레이브에서 제공하는 Input PDO 데이터의 크기를 바이트 단위로 반환합니다.

### PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ InPdoUnitIdx: 이 값은 해당 슬레이브가 Input PDO 용 Sync. Manager(SM)를 여러 개 제공할 때에 각 SM 을 구분하기 위한 매개변수입니다. 일반적으로 대부분의 슬레이브 장치들은 Input SM 을 하나만 사용하므로 특별한 경우가 아니면 이 값은 0 으로 설정하면 됩니다. Sync. Manager 는 ESC(EtherCAT Slave Controller)에서 제공하는 메모리 관리자인데 그 내용을 설명하기에는 본 설명서의 범주를 벗어나므로 설명을 생략합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

### RETURN VALUE

\* 해당 슬레이브에서 제공하는 Input PDO 데이터의 크기를 바이트 단위로 반환합니다.

<h2>NAME</h2> <p>ecSlv_OutPDO_GetBufPtr</p> <p>- OutPDO 의 포인터 주소 반환</p>	<h2>INFORMATION</h2> <ul style="list-style-type: none"> <li> Slave I/F Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>
---	---

## SYNOPSIS

```
void* ecSlv_OutPDO_GetBufPtr ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_ui8
OutPdoUnitIdx, [out] t_i32 *ErrCode)
```

## DESCRIPTION

해당 슬레이브의 Output PDO 데이터가 존재하는 메모리 주소값(버퍼 포인터)을 반환합니다. 사용자는 이 주소값을 이용해서 해당 슬레이브에게 출력하는 Output PDO 데이터를 직접 액세스(Access)하여 출력 데이터의 값을 설정할 수 있습니다.

Output PDO 데이터는 마스터가 슬레이브에게 주기적으로 전달하는 출력 데이터를 말합니다. 이 것은 슬레이브 입장에서는 수신하는 것이므로 Rx PDO 라고도 합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ OutPdoUnitIdx: 이 값은 해당 슬레이브가 Output PDO 용 Sync. Manager(SM)를 여러 개 제공할 때에 각 SM 을 구분하기 위한 매개변수입니다. 일반적으로 대부분의 슬레이브 장치들은 Output SM 을 하나만 사용하므로 특별한 경우가 아니면 이 값은 0 으로 설정하면 됩니다. Sync. Manager 는 ESC(EtherCAT Slave Controller)에서 제공하는 메모리 관리자인데 그 내용을 설명하기에는 본 설명서의 범주를 벗어나므로 설명을 생략합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 해당 슬레이브의 Output PDO 데이터가 존재하는 메모리 주소값(버퍼 포인터)을 반환합니다.

## EXAMPLE

< 예제 설명 >

슬레이브 주소가 0x211 인 32 채널 디지털출력 슬레이브 모듈이 있다고 가정합니다. 또한 해당 슬레이브는 32 비트(4 바이트)의 Output PDO 데이터를 제공하며 각각의 비트는

디지털출력 채널에 대응되는 상태값이라 가정합니다. 이 때 해당 슬레이브의 Output PDO 버퍼를 직접 액세스해서 0 번 채널의 디지털 출력 값을 토글하는 예제입니다.

---





< 예제 코드 >

```
#define NETID          0
#define SLAVE_ADDR    0x211

t_ui32 *pdwPtr;
static t_bool outVal = 0;
pdwPtr = ecSlv_InPDO_GetBufPtr (NETID, SLAVE_ADDR, 0, NULL);

if(pdwPtr != NULL){
    outVal = (outVal) ? 0: 1; // 출력 값 토글
    *pdwPtr = (*pdwPtr & 0xffffffe) | outVal; // 0 번 비트 값을 outVal 값으로 셋팅
    // *pdwPtr 의 값은 이더넷의 통신 사이클마다 해당 슬레이브에게 전달됨 //
}
```

---

NAME	I N F O R M A T I O N
ecSlv_OutPDO_GetBufLen - InPDO 의 길이 반환	<div style="border-bottom: 1px solid black; padding: 2px;">  Slave I/F Function         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  VC++/BCB/.NET         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  Level 1         </div> <div style="padding: 2px;">  위험 요소 없음         </div>

## SYNOPSIS

t\_i32 ecSlv\_OutPDO\_GetBufLen ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_ui8 OutPdoUnitIdx, [out] t\_i32 \*ErrCode)

## DESCRIPTION





해당 슬레이브에서 제공하는 Output PDO 데이터의 크기를 바이트 단위로 반환합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ OutPdoUnitIdx: 이 값은 해당 슬레이브가 Output PDO 용 Sync. Manager(SM)를 여러 개 제공할 때에 각 SM 을 구분하기 위한 매개변수입니다. 일반적으로 대부분의 슬레이브 장치들은 Output SM 을 하나만 사용하므로 특별한 경우가 아니면 이 값은 0 으로 설정하면 됩니다. Sync. Manager 는 ESC(EtherCAT Slave Controller)에서 제공하는 메모리 관리자인데 그 내용을 설명하기에는 본 설명서의 범주를 벗어나므로 설명을 생략합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecSlv\_OutPDO\_GetBufPtr(): OutPDO 의 길이를 반환합니다.

NAME	INFORMATION
ecSlv_OutPDO_SetInitValMode	 Slave I/F Function
ecSlv_OutPDO_GetInitValMode	 VC++/BCB/.NET
- Output PDO 데이터의 초기화 모드 속성을 설정/반환	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecSlv\_OutPDO\_SetInitValMode ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_ui8 OutPdoUnitIdx, [in] t\_i32 OutPDOInitMode, [out] t\_i32 \*ErrCode)
- t\_i32 ecSlv\_OutPDO\_GetInitValMode ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_ui8 OutPdoUnitIdx, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecSlv\_OutPDO\_SetInitValMode 함수는 지정한 슬레이브의 Output PDO 데이터의 초기화 모드 속성을 설정합니다.

Output PDO 데이터는 해당 슬레이브의 AL-STATE 가 'OP' 상태로 전환될 때 처음 전달되게 되는데, 이 처음 전달되는 데이터의 초기값을 어떻게 할 것인지를 결정하는 것입니다. 이 초기값은 해당 슬레이브가 'OP' 상태로 전환될 때 출력되는 출력값에 영향을 미칠 수 있습니다.

Output PDO 데이터의 초기값은 사용자가 Output PDO 데이터를 변경하기 전까지는 그대로 유지됩니다.

ecSlv\_OutPDO\_GetInitValMode 함수는 지정한 슬레이브의 Output PDO 데이터의 초기화 모드 속성을 반환합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용 할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ OutPdoUnitIdx: 이 값은 해당 슬레이브가 Output PDO 용 Sync. Manager(SM)를 여러 개 제공할 때에 각 SM 을 구분하기 위한 매개변수입니다. 일반적으로 대부분의 슬레이브 장치들은 Output SM 을 하나만 사용하므로 특별한 경우가 아니면 이 값은 0 으로 설정하면 됩니다. Sync. Manager 는 ESC(EtherCAT Slave Controller)에서 제공하는 메모리 관리자인데 그 내용을 설명하기에는 본 설명서의 범주를 벗어나므로 설명을 생략합니다.
- ▶ OutPDOInitMode: 지정한 슬레이브의 Output PDO 데이터의 초기값 설정 속성을 설정합니다. 이 매개변수에 설정할 수 있는 값은 아래의 표와 같습니다.

Value	Meaning	Default
-------	---------	---------

0 (ecOPDO_INIT_KEEPLAST)	이 전의 'OP' 상태에서 출력했던 마지막 출력값을 그대로 유지합니다. 단, 컴퓨터가 재부팅되면 마지막 출력값은 모두 0이 됩니다.	Y
1 (ecOPDO_INIT_ZERO)	해당 슬레이브의 Output PDO 데이터를 모두 0으로 설정합니다.	
2 (ecOPDO_INIT_USERDEF)	사용자가 미리 정의한 초기값으로 초기화 합니다. 사용자는 ecSlv_OutPDO_SetInitVal 함수를 이용하여 초기값을 정의할 수 있습니다.	





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

RETURN VALUE

\* ecSlv\_OutPDO\_SetInitValMode 함수의 반환값: 함수 수행의 성공 여부를 반환합니다.

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

\* ecSlv\_OutPDO\_GetInitValMode 함수의 반환값: 현재 설정되어 있는 Output PDO 데이터 초기화 모드 속성값을 반환합니다.

NAME	INFORMATION
ecSlv_OutPDO_SetInitVal ecSlv_OutPDO_GetInitVal - Output PDO 데이터의 사용자 정의 초기값 설정/반환	<div style="border-bottom: 1px solid black; padding: 2px;">  Net Interface Function         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  VC++/BCB/.NET         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  Level 1         </div> <div style="padding: 2px;">  위험 요소 없음         </div>

## SYNOPSIS

- t\_success ecSlv\_OutPDO\_SetInitVal ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_ui8 OutPdoUnitIdx, [in] t\_i32 OfsPos, [in] t\_i32 DataSize, [in] void \*pData, [out] t\_i32 \*ErrCode)
- t\_i32 ecSlv\_OutPDO\_GetInitVal ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_ui8 OutPdoUnitIdx, [in] t\_i32 OfsPos, [in] t\_i32 BufSize, [out] void \*pBuf, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecSlv\_OutPDO\_SetInitVal 함수는 지정한 슬레이브의 Output PDO 데이터의 사용자 정의 초기값을 설정합니다. 사용자 정의 초기값은 ecSlv\_OutPDO\_SetInitValMode 함수의 OutPDOInitMode 인자를 2(ecOPDO\_INIT\_USERDEF)로 설정했을 때 사용됩니다.

ecSlv\_OutPDO\_GetInitVal 함수는 현재 설정되어 있는 Output PDO 데이터의 사용자 정의 초기값을 pBuf 인자를 통해서 반환합니다.

## PARAMETER

- ▶ NetID : 사용자가 사용할 Network 번호
- ▶ SlvPhysAddr: 슬레이브의 물리적 주소(ex. 서보 로터리 스위치로 선택한 축 번호)
- ▶ OutPdoUnitIdx: 이 값은 해당 슬레이브가 Output PDO 용 Sync. Manager(SM)를 여러 개 제공할 때에 각 SM을 구분하기 위한 매개변수입니다. 일반적으로 대부분의 슬레이브 장치들은 Output SM을 하나만 사용하므로 특별한 경우가 아니면 이 값은 0으로 설정하면 됩니다. Sync. Manager는 ESC(EtherCAT Slave Controller)에서 제공하는 메모리 관리자인데 그 내용을 설명하기에는 본 설명서의 범주를 벗어나므로 설명을 생략합니다.
- ▶ OfsPos: 설정하고자 하는 데이터의 Output PDO 영역 상의 오프셋 위치(바이트 단위). 이때 Output PDO 영역이라 함은 네트워크 전체 영역이 아니라 해당 슬레이브만의 Output PDO 영역을 말합니다.
- ▶ DataSize: 설정하고자 하는 데이터의 크기(바이트 단위).

- ▶ BufSize: 데이터를 전달 받을 버퍼의 크기(바이트 단위)
- ▶ pData: 설정하고자 하는 초기값을 전달하는 포인터
- ▶ pBuf: 읽은 데이터를 전달 받을 버퍼의 포인터
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.





RETURN VALUE

\* ecSlv\_OutPDO\_SetInitVal 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecSlv\_OutPDO\_GetInitVal 함수의 반환값 : 반환되는 데이터의 크기



NAME	I N F O R M A T I O N
ecSlv_GetComErrSum	 Slave I/F Function
ecSlv_GetComErrSum_A	 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	I N F O R M A T I O N
ecSlv_ClearComErrSum	 Slave I/F Function
ecSlv_ClearComErrSum_A	 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. 서버 로터리 스위치로 선택한 축 번호)
- ▶ SlvPhysAddr : 슬레이브의 물리적 주소(ex. 서버 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)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.



## 11 Digital I/O Functions





### 11.1 함수 요약

Summary of Functions
<p>□ t_i32 ecdiGetSlaveIndex([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode) 대상 디지털 입력(Digital Input) 채널의 슬레이브 인덱스를 반환합니다.</p>
<p>□ t_i32 ecdoGetSlaveIndex([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode) 대상 디지털 출력(Digital Output) 채널의 슬레이브 인덱스를 반환합니다.</p>
<p>□ t_i32 ecdiGetSlaveID([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode) 대상 디지털 입력(Digital Input) 채널의 물리적 주소를 반환합니다.</p>
<p>□ t_i32 ecdoGetSlaveID([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode) 대상 디지털 출력(Digital Output) 채널의 물리적 주소를 반환합니다.</p>
<p>□ t_i32 ecdiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 Digital Input 채널 개수를 반환합니다.</p>
<p>□ t_i32 ecdoGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 Digital Output 채널 개수를 반환합니다.</p>
<p>□ t_bool ecdiGetOne([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode) 대상 디지털 입력(Digital Input) 채널에 대해 입력 상태를 확인합니다.</p>
<p>□ t_dword ecdiGetMulti([in] t_i32, NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode) 다중 디지털 입력(Digital Input) 채널에 대해 입력 상태를 확인합니다.</p>
<p>□ t_bool ecdoGetOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [out] t_i32 *ErrCode) 대상 디지털 출력(Digital Output) 채널에 대해 출력 상태를 확인합니다.</p>
<p>□ t_dword ecdoGetMulti([in] t_i32, NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode) 다중 디지털 출력(Digital Output) 채널에 대해 출력 상태를 확인합니다.</p>
<p>□ t_success ecdoPutOne([in] t_i32 NetID, [in] t_ui32 DoChannel, [in] t_bool OutState, [out] t_i32 *ErrCode) 대상 디지털 출력(Digital Output) 채널에 대해 출력을 발생합니다.</p>
<p>□ t_success ecdoPutMulti([in] t_i32, NetID, [in] t_ui32 IniChannel, [in] t_ui8 NumChannels, [in] t_dword dwOutStates, [out] t_i32 *ErrCode) 다중 디지털 출력(Digital Output) 채널에 대해 출력을 발생합니다.</p>
<p>□ t_dword ecdiGetLogicAddr([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode) 대상 디지털 입력(Digital Input) 채널의 논리비트주소를 반환합니다.</p>
<p>□ t_bool ecdiGetOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode) 물리적 주소와 로컬채널로 대상 디지털 입력(Digital Input) 채널에 대한 입력 상태를 확인합니다.</p>

<p>□ t_bool ecdoGetOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode) 물리적 주소와 로컬채널로 대상 디지털 출력(Digital Output)채널에 대한 출력 상태를 확인합니다.</p>
<p>□ t_bool ecdiGetMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 IniLocalChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode) 물리적 주소와 로컬채널로 다중 디지털 입력(Digital Input)채널에 대한 입력 상태를 확인합니다.</p>
<p>□ t_bool ecdoGetMulti_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 IniLocalChannel, [in] t_ui8 NumChannels, [out] t_i32 *ErrCode) 물리적 주소와 로컬채널로 다중 디지털 출력(Digital Output)채널에 대한 출력 상태를 확인합니다.</p>
<p>□ t_bool ecdoPutOne_L([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [in] t_bool OutState, [out] t_i32 *ErrCode) 슬레이브 인덱스와 로컬채널로 대상 디지털 출력(Digital Output)채널에 대한 출력을 발생합니다.</p>
<p>□ t_bool ecdoPutMulti_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 로컬채널로 다중 디지털 출력(Digital Output)채널에 대한 출력을 발생합니다.</p>
<p>□ t_dword ecdiLogBitAddr_FromGlobalChannel([in] t_i32 NetID, [in] t_i32 GlobalChannel, [out] t_i32 *ErrCode) Global Input Channel 을 논리비트주소로 변환</p>
<p>□ t_dword ecdoLogBitAddr_FromGlobalChannel([in] t_i32 NetID, [in] t_i32 GlobalChannel, [out] t_i32 *ErrCode) Global Output Channel 을 논리비트주소로 변환</p>
<p>□ t_dword ecdiLogBitAddr_FromLocalChannel([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode) Local Input Channel 을 논리비트주소로 변환</p>
<p>□ t_dword ecdoLogBitAddr_FromLocalChannel([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode) Local Output Channel 을 논리비트주소로 변환</p>
<p>□ t_dword ecdiLogBitAddr_FromOnboardChannel([in] t_i32 NetID, [in] t_i32 OnboardChannel, [out] t_i32 *ErrCode) Onboard Input Channel 을 논리비트주소로 변환</p>
<p>□ t_dword ecdoLogBitAddr_FromOnboardChannel([in] t_i32 NetID, [in] t_i32 OnboardChannel, [out] t_i32 *ErrCode) Onboard Output Channel 을 논리비트주소로 변환</p>
<p>□ t_i32 ecdiLogBitAddr_ToGlobalChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_i32 *ErrCode) 논리비트주소를 Global Input Channel 로 변환</p>
<p>□ t_i32 ecdoLogBitAddr_ToGlobalChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_i32 *ErrCode) 논리비트주소를 Global Output Channel 로 변환</p>

<p>□ t_i32 eccliLogBitAddr_ToLocalChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_ui16 *SlvPhysAddr, [out] t_i32 *ErrCode) D/I 논리비트주소를 Local Input Channel 로 변환</p>
<p>□ t_i32 ecdoLogBitAddr_ToLocalChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_ui16 *SlvPhysAddr, [out] t_i32 *ErrCode) D/O 논리비트주소를 Local Output Channel 로 변환</p>
<p>□ t_i32 eccliLogBitAddr_ToOnboardChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_i32 *ErrCode) D/I 논리비트주소를 Onboard Input Channel 로 변환</p>
<p>□ t_i32 ecdoLogBitAddr_ToOnboardChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_i32 *ErrCode) D/O 논리비트주소를 Onboard Output Channel 로 변환</p>
<p>□ EEcIOChanType eccliLogBitAddr_GetChanType([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_i32 *ErrCode) D/I 논리비트주소의 채널형식을 반환합니다.</p>
<p>□ EEcIOChanType ecdoLogBitAddr_GetChanType([in] t_i32 NetID, [in] t_dword LogBitAddr, [out] t_i32 *ErrCode) D/O 논리비트주소의 채널형식을 반환합니다.</p>

**11.2** 함수 설명

NAME	I N F O R M A T I O N
ecdiGetSlaveIndex	 Digital I/O Function
ecdoGetSlaveIndex	 VC++/BCB/.NET
- 대상 채널의 슬레이브 인덱스 확인	 Level 1
	 위험 요소 없음

**SYNOPSIS**

- t\_i32 ecdiGetSlaveIndex([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [out] t\_i32 \*ErrCode)
- t\_i32 ecdoGetSlaveIndex([in] t\_i32 NetID, [in] t\_ui32 DoChannel, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

ecdiGetSlaveIndex 함수는 지정된 디지털입력 채널(전역 채널)이 속한 슬레이브 장치의 슬레이브 인덱스(연결 순서)를 반환합니다.

ecDoGetSlaveIndex 함수는 지정된 디지털출력 채널(전역 채널)이 속한 슬레이브 장치의 슬레이브 인덱스(연결 순서)를 반환합니다.

**PARAMETER**

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

**RETURN VALUE**

\* 대상 채널이 속한 슬레이브 장치의 슬레이브 인덱스(연결 순서)를 반환합니다.

**REFERENCE**

- 슬레이브 인덱스를 슬레이브 주소값으로 변환하려면 ecSlv\_SlvIdx2PhysAddr 함수를 사용하면 됩니다.
- 해당 채널이 속한 슬레이브 주소값을 얻으려면 ecdiGetSlaveID / ecdoGetSlaveID 함수를 사용하십시오.

NAME	INFORMATION
ecdiGetSlaveID ecdoGetSlaveID - 대상 채널의 물리적 주소 확인	Digital I/O Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_i32 ecdiGetSlaveID([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [out] t\_i32 \*ErrCode)
- t\_i32 ecdoGetSlaveID([in] t\_i32 NetID, [in] t\_ui32 DoChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiGetSlaveIndex 함수는 지정된 디지털입력 채널(전역 채널)이 속한 슬레이브 장치의 슬레이브 주소를 반환합니다.

ecDoGetSlaveIndex 함수는 지정된 디지털출력 채널(전역 채널)이 속한 슬레이브 장치의 슬레이브 주소를 반환합니다.

## PARAMETER

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

## RETURN VALUE

- \* 대상 채널이 속한 슬레이브 장치의 슬레이브 주소를 반환합니다.

## REFERENCE


- 슬레이브 주소를 슬레이브 인덱스로 변환하려면 ecSlv\_PhysAddr2SlvIdx 함수를 사용하면 됩니다.
- 해당 채널이 속한 슬레이브 주소값을 얻으려면 ecdiGetSlaveIndex / ecdoGetSlaveIndex 함수를 사용하십시오.





**NAME**


ecdiGetNumChannels  
 ecdoGetNumChannels  
 - 디지털 입·출력 채널 갯수 반환

**INFORMATION**

 Digital I/O Function

 VC++/BCB/.NET

 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 지정된 네트워크에 연결되어 있는 전체 디지털 입력 채널의 개수를 반환합니다.

ecdoGetNumChannels 함수는 지정된 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: 지정된 네트워크에 연결되어 있는 디지털 입력 채널의 개수를 반환 합니다.

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

**EXAMPLE**


---

C/C++

---

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

```
t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수
//*****
9 장에서 각각의 슬레이브 당 채널을 얻어 오는 것 과는 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
}
```

---

### 11.2.1 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 0~63 의 번호로 구성됩니다. 따라서 두 번째 모듈의 첫 번째 채널은 전역채널 번호로는 32 번 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.

NAME	INFORMATION
ecdiGetOne - 단일 채널에 대한 입력 상태 반환	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
```

---

<b>NAME</b>  <b>ecdiGetMulti</b>  - 다중 채널에 대한 입력 상태 반환	<b>INFORMATION</b>
	 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, NUMCHANNELS, &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_GetMulti 함수 종료
```

---

<b>NAME</b>  <b>ecdiLtc_AddChannel</b>  - 래치 모드로 감시할 채널을 등록	<b>INFORMATION</b>
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_success ecdiLtc\_AddChannel ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

래치(Latch) 모드로 감시할 D/I 채널을 등록하는 함수입니다.

래치(Latch) 모드로 등록된 채널에 대해서는 ComiECAT 마스터 장치가 지속적으로 감시하면서 OFF 에서 ON 으로 바뀐 횟수를 기록하게 됩니다. 따라서 사용자는 입력 채널의 상태를 계속 감시하지 않아도 입력 채널의 상태가 OFF 에서 ON 으로 바뀐 횟수를 알 수 있습니다.

## 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

함수 수행의 성공 여부를 반환합니다.

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

## EXAMPLE

---

C/C++

---

```
#define NETID 0
#define LTC_CH 0
```

---



---

```
// OnStartLatchMon(): 래치모니터를 시작하는 가상의 사용자 함수 //
void OnStartLatchMon()
{
    t_i32 errCode;
    if(!ecdiLtc_AddChannel(NETID, LTC_CH, &errCode)){
        // 에러처리
    }
}

// OnTimer (): 래치 상태를 주기적으로 체크하는 가상의 사용자 함수 //
void OnTimer()
{
    if(ecdiLtc_GetOnCount(NETID, LTC_CH, true, NULL) > 0){
        // 래치 발생함!
        ...
    }
}
}
```


---

**NAME**


ecdiLtc\_DelChannel

- 래치 모드로 등록된 채널을 등록 해제

**INFORMATION**
 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

```

□ t_success ecdiLtc_DelChannel ([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32
*ErrCode)

```

**DESCRIPTION**

래치(Latch) 모드로 등록된 디지털입력 채널을 감시 대상 리스트에서 제거합니다.

**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**

\* 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.

**REFERENCE**

□ ecdiLtc\_DelChannel 함수를 실행하지 않아도 사용자 프로그램이 종료될 때에 해당 프로그램에서 등록된 래치 채널들은 자동으로 리스트에서 삭제됩니다. 따라서 이 함수의 실행을 생략해도 무방합니다.

**EXAMPLE**


---



---



---

C/C++

---



---



---

---

```
#define NETID 0
#define LTC_CH 0

// OnStartLatchMon(): 래치모니터를 시작하는 가상의 사용자 함수 //
void OnStartLatchMon()
{
    t_i32 errCode;
    if(!ecdiLtc_AddChannel(NETID, LTC_CH, &errCode)){
        // 에러처리
    }
}

// OnTimer (): 래치 상태를 주기적으로 체크하는 가상의 사용자 함수 //
void OnTimer()
{
    if(ecdiLtc_GetOnCount(NETID, LTC_CH, true, NULL) > 0){
        // 래치 발생함!
        ...
    }
}

// OnProgramExit (): 프로그램의 종료 시에 호출되는 가상의 사용자 함수 //
void OnProgramExit()
{
    ecdiLtc_DelChannel(NETID, LTC_CH, NULL)
}


```


---

**NAME**


**ecdiLtc\_SetFilter / ecdiLtc\_GetFilter**  
 - 래치 입력의 필터 설정 / 설정값 반환

**INFORMATION**

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecdiLtc\_SetFilter ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [in] t\_i32 FilterCycles, [out] t\_i32 \*ErrCode)

□ t\_i32 ecdiLtc\_GetFilter ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

ecdiLtc\_SetFilter 함수는 지정한 채널의 래치 모드에서 적용할 필터의 사이클 카운트값 설정합니다.

ecdiLtc\_GetFilter 함수는 현재 설정되어 있는 래치 모드 필터의 사이클 카운트값을 반환합니다.

래치 모드 필터는 노이즈와 같은 잘못된 입력 신호를 필터하기 위해서 일정 시간동안 ON 상태를 유지하는 것만 실제 ON으로 인정하도록 하는 것입니다. 필터 시간으로 등록된 시간보다 짧거나 같은 유지시간을 가지는 신호는 ON으로 인정되지 않습니다.

**PARAMETER**

▶ NetID : Network 번호

▶ DiChannel : 디지털 입력 채널 번호

▶ FilterCycles : 필터 시간을 결정하는 인자입니다. 이 값의 단위는 이더넷 통신사이클 카운트입니다. 여기서 지정한 통신사이클 횟수보다 짧거나 같은 유지 시간을 가지는 신호는 ON으로 인정되지 않습니다. 예를 들어서 이더넷 통신 주기가 1msec 인 환경에서 이 값을 1로 설정하였다면 1msec 이하의 신호는 ON으로 인정되지 않습니다. 결국 2 통신사이클(2msec) 이상의 유지 시간을 가지는 신호만 ON으로 인정되게 됩니다.

이 값을 0으로 하면 필터를 적용하지 않습니다.

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

**RETURN VALUE**

\* ecdiLtc\_SetFilter 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* `ecdiLtc_GetFilter` 함수의 반환값 : 현재 설정되어 있는 래치 필터의 사이클 카운트값을 반환합니다.


## REFERENCE


□ 래치 모드로 새로 등록되면 `FilterCycles` 는 0 으로 설정됩니다.

## NAME


ecdiLtc\_SetLogicInvert /  
ecdiLtc\_GetLogicInvert  
- 래치 입력 로직의 반전 속성 설정/반환

## INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_success ecdiLtc\_SetLogicInvert ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [in] t\_bool IsInvertLogic, [out] t\_i32 \*ErrCode)

❑ t\_bool ecdiLtc\_GetLogicInvert ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiLtc\_SetLogicInvert 함수는 지정한 채널의 래치 입력 로직의 반전 속성을 설정합니다.

ecdiLtc\_GetLogicInvert 함수는 현재 설정되어 있는 래치 입력 로직의 반전 속성 값을 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ DiChannel : 디지털 입력 채널 번호
- ▶ IsInvertLogic : 입력 로직을 반전하여 래치 입력으로 사용할 것인지를 설정합니다.

Value	Meaning	Default
0	입력 로직을 반전하지 않고 그대로 사용합니다. 따라서 입력 신호의 값이 OFF->ON 이 될 때 래치 카운트도 증가합니다.	Y
1	입력 로직을 반전하여 사용합니다. 따라서 입력 신호의 값이 ON->OFF 가 될 때 래치 카운트도 증가합니다.	

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

## RETURN VALUE

\* ecdiLtc\_SetLogicInvert 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* `ecdiLtc_GetLogicInvert` 함수의 반환값 : 현재 설정되어 있는 래치 입력 로직의 반전 속성 값을 반환합니다.


## REFERENCE


□ 입력 신호가 OFF 인 상태에서 래치 입력 로직을 반전하는 것으로 설정하면 그 순간에 래치 카운트가 1 증가할 수 있습니다. 이 것을 카운트에서 제외하려면 `ecdiLtc_SetLogicInvert` 함수를 실행한 후에 `ecdiLtc_ResetOnCount` 함수를 실행해주면 됩니다.


**NAME**


**ecdiLtc\_GetOnCount**  
- 래치 카운트 반환

**INFORMATION**

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_i32 ecdiLtc\_GetOnCount ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [in] t\_bool IsResetOnCount, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

지정된 디지털입력 채널의 래치 카운트를 반환합니다. 여기서 래치 카운트란 OFF 에서 ON 으로 변경된 회수를 의미합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ DiChannel : 디지털 입력 채널 번호
- ▶ IsResetOnCount : 래치 카운트를 읽은 후에 마스터에서 관리하는 래치 카운트를 0 으로 리셋할 것인지를 설정하는 인자입니다.

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

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

**RETURN VALUE**

지정한 채널의 입력 신호가 OFF 에서 ON 으로 변경된 회수를 반환합니다.

**EXAMPLE**

C/C++

```
#define NETID 0
#define LTC_CH 0
```

```
// OnStartLatchMon(): 래치모니터를 시작하는 가상의 사용자 함수 //
```





---

```
void OnStartLatchMon()
{
    t_i32 errCode;
    if(!ecdiLtc_AddChannel(NETID, LTC_CH, &errCode)){
        // 에러처리
    }
}

// OnTimer (): 래치 상태를 주기적으로 체크하는 가상의 사용자 함수 //
void OnTimer()
{
    if(ecdiLtc_GetOnCount(NETID, LTC_CH, true, NULL) > 0){
        // 래치 발생함!
        ...
    }
}
```

---

<h2>NAME</h2> <p><b>ecdiLtc_ResetOnCount</b></p> <p>- 래치 카운트 리셋</p>	<b>INFORMATION</b>
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_success ecdiLtc\_ResetOnCount ([in] t\_i32 NetID, [in] t\_ui32 DiChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

지정한 채널의 래치 카운트 값을 0으로 리셋합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ DiChannel : 디지털 입력 채널 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.

## REFERENCE


□ 입력 신호의 상태가 래치 로직 기준으로 ON인 상태에서 래치 채널을 등록하면 래치 카운트는 1이 됩니다. 만일 래치 채널로 등록한 이후에 상승에지(Rising Edge)가 발생하였을 때만 래치 카운트로 인정하려면 래치 채널을 등록한 후에 ecdiLtc\_ResetOnCount 함수를 실행해 주면 됩니다.


## NAME


ecdoPutOne / ecdoGetOne


- 단일 채널에 대한 출력 발생 및 출력 상태  
확인

### 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 발생립니다.
- ▶ 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)
{
    TEcSrvTypeInfo 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)
{
    TEcSrvTypeInfo 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 style="margin: 0;">NAME</h2> <p style="margin: 0;"><b>ecdoPutMulti / ecdoGetMulti</b></p> <p style="margin: 0;">- 다중 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 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

```


---


## NAME


`ecdiGetLogicAddr`


- 대상 채널의 논리비트주소 반환

### INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```
□ t_dword ecdiGetLogicAddr([in] t_i32 NetID, [in] t_ui32 DiChannel, [out] t_i32 *ErrCode)
```

## DESCRIPTION

이 함수는 대상 채널의 논리비트주소를 반환합니다..

## 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 논리비트주소를 반환합니다.







### 11.2.2 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.

 안내	<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	INFORMATION
<p><code>ecdiGetOne_L</code></p> <p>- 대상 채널의 입·출력상태 확인</p>	<p> Digital I/O Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

## 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.


## NAME

`ecdiGetMulti_L`


- 다중 채널의 입·출력상태 확인

### 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.

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

## SYNOPSIS

□ t\_success ecdiLtc\_AddChannel\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

래치(Latch) 모드로 감시할 D/I 채널을 등록하는 함수입니다.

래치(Latch) 모드로 등록된 채널에 대해서는 ComiECAT 마스터 장치가 지속적으로 감시하면서 OFF 에서 ON 으로 바뀐 횟수를 기록하게 됩니다. 따라서 사용자는 입력 채널의 상태를 계속 감시하지 않아도 입력 채널의 상태가 OFF 에서 ON 으로 바뀐 횟수를 알 수 있습니다.

## 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

함수 수행의 성공 여부를 반환합니다.

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

## EXAMPLE

C/C++

---

```
#define NETID      0
#define SLV_ADDR  0x204
#define LTC_CH    0

// OnStartLatchMon(): 래치모니터를 시작하는 가상의 사용자 함수 //
void OnStartLatchMon()
{
    t_i32 errCode;
    if(!ecdiLtc_AddChannel_L(NETID, SLV_ADDR, LTC_CH, &errCode)){
        // 에러처리
    }
}


// OnTimer (): 래치 상태를 주기적으로 체크하는 가상의 사용자 함수 //
void OnTimer()
{
    if(ecdiLtc_GetOnCount_L(NETID, SLV_ADDR, LTC_CH, true, NULL) > 0){
        // 래치 발생함!
        ...
    }
}
```


---


**NAME**


ecdiLtc\_DelChannel\_L

- 단일 채널에 대한 출력 발생 및 출력 상태  
확인

**INFORMATION**
 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

□ t\_success ecdiLtc\_DelChannel\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

래치(Latch) 모드로 등록된 디지털 입력 채널을 감시 대상 리스트에서 제거합니다.

**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**

\* 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.

**REFERENCE**

□ ecdiLtc\_DelChannel 함수를 실행하지 않아도 사용자 프로그램이 종료될 때에 해당 프로그램에서 등록한 래치 채널들은 자동으로 리스트에서 삭제됩니다. 따라서 이 함수의 실행을 생략해도 무방합니다.

**EXAMPLE**

---

C/C++

---

```
#define NETID 0
#define LTC_CH 0

#define NETID 0
#define SLV_ADDR 0x204
#define LTC_CH 0





// OnStartLatchMon(): 래치모니터를 시작하는 가상의 사용자 함수 //
void OnStartLatchMon()
{
    t_i32 errCode;
    if(!ecdiLtc_AddChannel_L(NETID, SLV_ADDR, LTC_CH, &errCode)){
        // 에러처리
    }
}

// OnTimer(): 래치 상태를 주기적으로 체크하는 가상의 사용자 함수 //
void OnTimer()
{
    if(ecdiLtc_GetOnCount_L(NETID, SLV_ADDR, LTC_CH, true, NULL) > 0){
        // 래치 발생함!
        ...
    }
}

// OnProgramExit(): 프로그램의 종료 시에 호출되는 가상의 사용자 함수 //
void OnProgramExit()
{
    ecdiLtc_DelChannel_L (NETID, SLV_ADDR, LTC_CH, NULL)
}

```

---

NAME	INFORMATION
<p><b>ecdiLtc_SetFilter_L / ecdiLtc_GetFilter_L</b></p> <p>- 단일 채널에 대한 출력 발생 및 출력 상태 확인</p>	<p> Digital I/O Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

## SYNOPSIS

- ▣ t\_success ecdiLtc\_SetFilter\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [in] t\_i32 FilterCycles, [out] t\_i32 \*ErrCode)
- ▣ t\_i32 ecdiLtc\_GetFilter\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiLtc\_SetFilter 함수는 지정한 채널의 래치 모드에서 적용할 필터의 사이클 카운트값 설정합니다.

ecdiLtc\_GetFilter 함수는 현재 설정되어 있는 래치 모드 필터의 사이클 카운트값을 반환합니다.

래치 모드 필터는 노이즈와 같은 잘못된 입력 신호를 필터하기 위해서 일정 시간동안 ON 상태를 유지하는 것만 실제 ON으로 인정하도록 하는 것입니다. 필터 시간으로 등록된 시간보다 짧거나 같은 유지시간을 가지는 신호는 ON으로 인정되지 않습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr: 슬레이브 주소
- ▶ LocalChannel: 해당 디지털 입력 채널의 지역 채널 번호
- ▶ FilterCycles : 필터 시간을 결정하는 인자입니다. 이 값의 단위는 이더넷 통신사이클 카운트입니다. 여기서 지정한 통신사이클 횟수보다 짧거나 같은 유지 시간을 가지는 신호는 ON으로 인정되지 않습니다. 예를 들어서 이더넷 통신 주기가 1msec 인 환경에서 이 값을 1로 설정하였다면 1msec 이하의 신호는 ON으로 인정되지 않습니다. 결국 2통신사이클(2msec) 이상의 유지 시간을 가지는 신호만 ON으로 인정되게 됩니다. 이 값을 0으로 하면 필터를 적용하지 않습니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecdiLtc\_SetFilter 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.







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

\* `ecdiLtc_GetFilter` 함수의 반환값 : 현재 설정되어 있는 래치 필터의 사이클 카운트값을 반환합니다.

## REFERENCE

래치 모드로 새로 등록되면 `FilterCycles` 는 0 으로 설정됩니다.

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecdiLtc_SetLogicInvert_L / ecdiLtc_GetLogicInvert_L</p> <p style="margin: 0;">- 래치 입력 로직의 반전 속성 설정/반환</p>	<b>INFORMATION</b>
	 Digital I/O Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- ❑ t\_success ecdiLtc\_SetLogicInvert\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [in] t\_bool IsInvertLogic, [out] t\_i32 \*ErrCode)
- ❑ t\_bool ecdiLtc\_GetLogicInvert\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiLtc\_SetLogicInvert\_L 함수는 지정한 채널의 래치 입력 로직의 반전 속성을 설정합니다.  
ecdiLtc\_GetLogicInvert\_L 함수는 현재 설정되어 있는 래치 입력 로직의 반전 속성 값을 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr: 슬레이브 주소
- ▶ LocalChannel: 해당 디지털 입력 채널의 지역 채널 번호
- ▶ IsInvertLogic : 입력 로직을 반전하여 래치 입력으로 사용할 것인지를 설정합니다.

Value	Meaning	Default
0	입력 로직을 반전하지 않고 그대로 사용합니다. 따라서 입력 신호의 값이 OFF->ON 이 될 때 래치 카운트도 증가합니다.	Y
1	입력 로직을 반전하여 사용합니다. 따라서 입력 신호의 값이 ON->OFF 가 될 때 래치 카운트도 증가합니다.	

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

## RETURN VALUE

\* ecdiLtc\_SetLogicInvert 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* `ecdiLtc_GetLogicInvert` 함수의 반환값 : 현재 설정되어 있는 래치 입력 로직의 반전 속성 값을 반환합니다.

## REFERENCE

□ 입력 신호가 OFF 인 상태에서 래치 입력 로직을 반전하는 것으로 설정하면 그 순간에 래치 카운트가 1 증가할 수 있습니다. 이것을 카운트에서 제외하려면 `ecdiLtc_SetLogicInvert_L` 함수를 실행한 후에 `ecdiLtc_ResetOnCount_L` 함수를 실행해주면 됩니다.

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;"><b>ecdiLtc_GetOnCount_L</b></p> <p style="margin: 0;">- 단일 채널에 대한 출력 발생 및 출력 상태 확인</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left; padding: 2px;">INFORMATION</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"> Digital I/O 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> </tbody> </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\_i32 ecdiLtc\_GetOnCount\_L ([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [in] t\_bool IsResetOnCount, [out] t\_i32 \*ErrCode)

### DESCRIPTION

지정된 디지털입력 채널의 래치 카운트를 반환합니다. 여기서 래치 카운트란 OFF 에서 ON 으로 변경된 회수를 의미합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr: 슬레이브 주소
- ▶ LocalChannel: 해당 디지털 입력 채널의 지역 채널 번호
- ▶ IsResetOnCount : 래치 카운트를 읽은 후에 마스터에서 관리하는 래치 카운트를 0 으로 리셋할 것인지를 설정하는 인자입니다.

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

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

### RETURN VALUE

지정한 채널의 입력 신호가 OFF 에서 ON 으로 변경된 회수를 반환합니다.

### EXAMPLE

---

C/C++

---

```
#define NETID    0
```

---

---

```
#define SLV_ADDR 0x204
#define LTC_CH 0

// OnStartLatchMon(): 래치모니터를 시작하는 가상의 사용자 함수 //
void OnStartLatchMon()
{
    t_i32 errCode;
    if(!ecdiLtc_AddChannel_L(NETID, SLV_ADDR, LTC_CH, &errCode)){
        // 에러처리
    }
}

// OnTimer(): 래치 상태를 주기적으로 체크하는 가상의 사용자 함수 //
void OnTimer()
{
    if(ecdiLtc_GetOnCount_L(NETID, SLV_ADDR, LTC_CH, true, NULL) > 0){
        // 래치 발생함!
        ...
    }
}
```


---


## NAME


`ecdiLtc_ResetOnCount_L`


- 단일 채널에 대한 출력 발생 및 출력 상태  
확인

## INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

`t_success ecdiLtc_ResetOnCount_L ([in] t_i32 NetID, [in] t_ui16 SlvPhysAddr, [in] t_i32 LocalChannel, [out] t_i32 *ErrCode)`

## DESCRIPTION

지정한 채널의 래치 카운트 값을 0으로 리셋합니다.

## 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

함수 수행의 성공 여부를 반환합니다.

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

## REFERENCE


□ 입력 신호의 상태가 래치 로직 기준으로 ON 인 상태에서 래치 채널을 등록하면 래치 카운트는 1이 됩니다. 만일 래치 채널로 등록한 이후에 상승에지(Rising Edge)가 발생하였을 때만 래치 카운트로 인정하려면 래치 채널을 등록한 후에 `ecdiLtc_ResetOnCount_L` 함수를 실행해 주면 됩니다.


**NAME**

ecdoPutOne\_L


ecdoGetOne\_L

- 단일 채널의 DO 출력 발생 및 출력상태 확인

**INFORMATION**
 Digital I/O Function

 VC++/BCB/.NET

 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 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.

**NAME**


ecdoPutMulti\_L  
 ecdoGetMulti\_L  
 - 다중 채널의 DO 출력

**INFORMATION**

 Digital I/O Function

 VC++/BCB/.NET

 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.



### 11.2.3 논리비트주소(Logic Bit Address) 관리 함수군

NAME	INFORMATION
ecdiLogBitAddr_FromGlobalChannel	 Digital I/O Function
ecdoLogBitAddr_FromGlobalChannel	 VC++/BCB/.NET
- 통합 채널을 논리비트주소로 변환	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_dword ecdiLogBitAddr\_FromGlobalChannel([in] t\_i32 NetID, [in] t\_i32 GlobalChannel, [out] t\_i32 \*ErrCode)
- t\_dword ecdoLogBitAddr\_FromGlobalChannel([in] t\_i32 NetID, [in] t\_i32 GlobalChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiLogBitAddr\_FromGlobalChannel(), ecdoLogBitAddr\_FromGlobalChannel() 함수는 Global Channel 을 논리비트주소로 변환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ GlobalChannel : 통합 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 논리비트주소를 반환합니다.

NAME	INFORMATION
ecdiLogBitAddr_FromLocalChannel ecdoLogBitAddr_FromLocalChannel - 로컬 채널을 논리비트주소로 변환	Digital I/O Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_dword ecdiLogBitAddr\_FromLocalChannel([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [out] t\_i32 \*ErrCode)
- t\_dword ecdoLogBitAddr\_FromLocalChannel([in] t\_i32 NetID, [in] t\_ui16 SlvPhysAddr, [in] t\_i32 LocalChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION





ecdiLogBitAddr\_FromLocalChannel(), ecdoLogBitAddr\_FromLocalChannel() 함수는 Local Channel 을 논리비트주소로 변환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ SlvPhysAddr : 물리적 주소
- ▶ LocalChannel : 슬레이브 내에서의 채널번호
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

- \* 이 리턴값은 논리비트주소를 반환합니다.

NAME	I N F O R M A T I O N
ecdiLogBitAddr_FromOnBoardChannel ecdoLogBitAddr_FromOnBoardChannel - Onboard 채널을 논리비트주소로 변환	 Digital I/O Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

- t\_dword ecdiLogBitAddr\_FromOnBoardChannel([in] t\_i32 NetID, [in] t\_i32 OnboardChannel, [out] t\_i32 \*ErrCode)
- t\_dword ecdoLogBitAddr\_FromOnBoardChannel([in] t\_i32 NetID, [in] t\_i32 OnboardChannel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiLogBitAddr\_FromOnBoardChannel(), ecdoLogBitAddr\_FromOnBoardChannel() 함수는 OnboardChannel 을 논리비트주소로 변환합니다.

## PARAMETER

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


## RETURN VALUE


- \* 이 리턴값은 논리비트주소를 반환합니다.


**NAME**


ecdiLogBitAddr\_ToGlobalChannel  
 ecdoLogBitAddr\_ToGlobalChannel  
 - 논리비트주소를 통합채널로 변환

**INFORMATION**

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

- ❑ t\_i32 ecdiLogBitAddr\_ToGlobalChannel([in] t\_i32 NetID, [in] t\_dword LogBitAddr, [out] t\_i32 \*ErrCode)
- ❑ t\_i32 ecdoLogBitAddr\_ToGlobalChannel([in] t\_i32 NetID, [in] t\_dword LogBitAddr, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

ecdiLogBitAddr\_ToGlobalChannel(), ecdoLogBitAddr\_ToGlobalChannel() 함수는 논리비트주소를 Global Channel 로 변환합니다.

**PARAMETER**

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


**RETURN VALUE**


- \* 이 리턴값은 통합 채널을 반환합니다.


## NAME


ecdiLogBitAddr\_ToLocalChannel  
 ecdoLogBitAddr\_ToLocalChannel  
 - 논리비트주소를 로컬 채널로 변환

## INFORMATION

 Digital I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```

  □ t_i32 ecdiLogBitAddr_ToLocalChannel([in] t_i32 NetID, [in] t_dword LogBitAddr, [out]
  t_ui16 *SlvPhysAddr, [out] t_i32 *ErrCode)
  □ t_i32 ecdoLogBitAddr_ToLocalChannel([in] t_i32 NetID, [in] t_dword LogBitAddr,
  [out] t_ui16 *SlvPhysAddr, [out] t_i32 *ErrCode)
  
```

## DESCRIPTION





ecdiLogBitAddr\_ToLocalChannel(), ecdoLogBitAddr\_ToLocalChannel() 함수는 논리비트주소를 Local Channel 로 변환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ LogBitAddr : 변환할 논리비트주소
- ▶ SlvPhysAddr : 로컬 채널로 변경된 슬레이브의 물리적 주소
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 이 리턴값은 로컬 채널을 반환합니다.

NAME	I N F O R M A T I O N
ecdiLogBitAddr_ToOnboardChannel ecdoLogBitAddr_ToOnboardChannel - 논리비트주소를 Onboard 채널로 변환	 Digital I/O Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

- t\_i32 ecdiLogBitAddr\_ToOnboardChannel([in] t\_i32 NetID, [in] t\_dword LogBitAddr, [out] t\_i32 \*ErrCode)
- t\_i32 ecdoLogBitAddr\_ToOnboardChannel([in] t\_i32 NetID, [in] t\_dword LogBitAddr, [out] t\_i32 \*ErrCode)

## DESCRIPTION





ecdiLogBitAddr\_ToOnboardChannel(), ecdoLogBitAddr\_ToOnboardChannel() 함수는 논리비트주소를 OnboardChannel 로 변환합니다.

## PARAMETER

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

## RETURN VALUE

\* 이 리턴값은 OnboardChannel 을 반환합니다.

NAME	I N F O R M A T I O N
ecdiLogBitAddr_GetChanType	 Digital I/O Function
ecdoLogBitAddr_GetChanType	 VC++/BCB/.NET
-	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_i32 ecdiLogBitAddr\_GetChanType([in] t\_i32 NetID, [in] t\_dword LogBitAddr, [out] t\_i32 \*ErrCode)
- t\_i32 ecdoLogBitAddr\_GetChanType ([in] t\_i32 NetID, [in] t\_dword LogBitAddr, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecdiLogBitAddr\_GetChanType(), ecdoLogBitAddr\_GetChanType () 함수는

## PARAMETER

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

## RETURN VALUE

- \* 이 리턴값은

# 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.









## 12 Analog I/O Functions

### 12.1 함수 요약

Summary of Functions
<p>□ t_i32 ecaiGetSlaveIndex([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode) 대상 아날로그 입력채널이 포함된 슬레이브 인덱스를 반환합니다.</p>
<p>□ t_i32 ecaoGetSlaveIndex([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode) 대상 아날로그 출력채널이 포함된 슬레이브 인덱스를 반환합니다.</p>
<p>□ t_i32 ecaiGetSlaveID([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode); 대상 아날로그 입력채널이 포함된 물리적 주소를 반환합니다.</p>
<p>□ t_i32 ecaoGetSlaveID([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode); 대상 아날로그 출력채널이 포함된 물리적 주소를 반환합니다.</p>
<p>□ t_i32 ecaiGetNumChannels([in] t_i32 NetID, [out] t_i32 *ErrCode) 장치에 연결되어 있는 아날로그 입력 채널 개수를 반환합니다.</p>
<p>□ t_i32 ecaoGetNumChannels([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_f64 ecaiGetChanVal_FS([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_f32 ScaleMin, [in] t_f32 ScaleMax, [out] t_i32 *ErrCode) 대상 아날로그 입력 채널에 대하여 AD 결과를 지정된 Full-Scale 값을 반영한 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_success ecaoSetChanVal_FS([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_f64 OutData, [in] t_f64 ScaleMin, [in] t_f64 ScaleMax, [out] t_i32 *ErrCode) 대상 아날로그 출력 채널을 통해 지정된 Full-Scale 값을 반영한 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>
<p>□ t_f64 ecaoGetOutValue_FS([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_f32 ScaleMin, [in] t_f32 ScaleMax, [out] t_i32 *ErrCode) 대상 아날로그 출력 채널에 대하여 AD 결과를 지정된 Full-Scale 값을 반영한 값으로 반환합니다.</p>

**12.2** 함수 설명

NAME	INFORMATION
ecaiGetSlaveIndex ecaoGetSlaveIndex - 대상 채널의 슬레이브 인덱스 확인	 Analog I/O Function <hr/>  VC++/BCB/.NET <hr/>  Level 1 <hr/>  위험 요소 없음
SYNOPSIS	
<pre>           □ t_i32 ecaiGetSlaveIndex([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode)           □ t_i32 ecaoGetSlaveIndex([in] t_i32 NetID, [in] t_ui32 Channel, [out] t_i32 *ErrCode)         </pre>	

## DESCRIPTION

ecaiGetSlaveIndex(),ecaoGetSlaveIndex() 함수는 대상 채널의 슬레이브 인덱스를 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 아날로그 입·출력 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 채널의 슬레이브 인덱스를 반환합니다..


## NAME


ecaiGetSlaveID


ecaoGetSlaveID


- 대상 채널의 물리적 주소 확인

## INFORMATION

 Analog I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_i32 ecaiGetSlaveID([in] t\_i32 NetID, [in] t\_ui32 Channel, [out] t\_i32 \*ErrCode)

❑ t\_i32 ecaoGetSlaveID([in] t\_i32 NetID, [in] t\_ui32 Channel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecaiGetSlaveID(), ecaoGetSlaveID() 함수는 대상 채널의 물리적 주소를 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 아날로그 입·출력 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 채널의 물리적 주소를 반환합니다..

**NAME**

ecaiGetNumChannels

ecaoGetNumChannels

- 아날로그 입·출력 채널 갯수 반환

**INFORMATION** Analog I/O Function VC++/BCB/.NET 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 연결 된 모든 AI 채널 개수 저장 할 변수
    t_i32 All_AOCount = 0;         //마스터에 연결 된 모든 AO 채널 개수 저장 할
    변수
    All_AICount = ecaiGetNumChannels (nNetID, &Error_Num);
    // 마스터에 연결 된 모든 AI 채널 개수를 로드
    All_AOCount = ecaoGetNumChannels (nNetID, &Error_Num);
}

```

---

```
// 마스터에 연결 된 모든 AO 채널 개수를 로드  
}
```


---


## NAME


ecaiGetChanVal\_I


- 대상 아날로그 입력 채널의 입력 Digit 값 반환

### INFORMATION

 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.

<b>NAME</b>  ecaiGetChanVal_F - 대상 아날로그 입력 채널의 입력 Volt 값 반환	<b>INFORMATION</b>
	 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	INFORMATION
ecaiGetChanVal_FS - 대상 아날로그 입력 채널의 입력 Volt 값 반환	<div style="border-bottom: 1px solid black; padding-bottom: 2px;"> Analog I/O Function</div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;"> VC++/BCB/.NET</div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;"> Level 1</div> <div style="padding-bottom: 2px;"> 위험 요소 없음</div>

## SYNOPSIS

```

❑ t_f64 ecaiGetChanVal_FS([in] t_i32 NetID, [in] t_ui32 Channel, [in] t_f32 ScaleMin, [in]
t_f32 ScaleMax, [out] t_i32 *ErrCode)

```

## DESCRIPTION

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

일반적으로 A/D 변환의 1 차적인 데이터는 정수값으로 표현됩니다. 이 정수 값은 일반적으로 계측 신호로 사용되는 전압이나 전류와 단위가 일치하지 않습니다. 이 때 입력 채널의 스케일 정보(입력 전압 또는 전류의 범위)를 알면 이 정수값을 소수점 수의 전압값이나 전류값으로 환산할 수 있습니다.

ecaiGetChanVal\_FS 함수는 사용자가 입력 채널의 전압(/전류) 범위를 알고 있을 때에 그 정보를 제공하여 현재 계측된 A/D 변환의 1 차 데이터인 정수값을 소수점 수의 전압(/전류) 값으로 반환하는 함수입니다.

## PARAMETER

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

## RETURN VALUE

\* ScaleMin 값과 ScaleMax 값을 입력 범위 조건으로 적용하여 현재 A/D 변환된 정수값을 해당 스케일 범위의 값으로 환산하여 반환합니다. 만일 ScaleMin/Max 값이 입력 전압 범위의 최소/최대값을 의미하는 것이면 반환 값은 전압 값이 되는 것이고, ScaleMin/Max 값이 입력 전류 범위의 최소/최대값을 의미하는 것이면 반환 값은 전류 값이 되는 것입니다.



## EXAMPLE

---

< 예제 설명 >

-10 ~ 10 V 의 입력 범위를 가지는 CH0 의 A/D 값을 volt 값으로 입력 받는 예

---

< 예제 코드 >

```
#define NETID    0  
#define AI_CH    0
```


```
t_f64 volt = ecaiGetChanVal_FS (NETID, AI_CH, -10.0f, 10.0f, NULL);  
printf("A/D CH0 Voltage = %.3f\n", volt);
```


---


**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.

<h2>NAME</h2> <p><b>ecaoSetChanVal_FS</b></p> <p>- 대상 아날로그 출력 채널을 통해 Volt 값 출력</p>	<b>INFORMATION</b>
	 Analog I/O Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_success ecaoSetChanVal\_FS([in] t\_i32 NetID, [in] t\_ui32 Channel, [in] t\_f64 OutData, [in] t\_f32 ScaleMin, [in] t\_f32 ScaleMax, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 대상 아날로그 출력 채널에 대하여 지정한 Volt 값을 출력합니다. ScaleMin, ScaleMax 매개변수를 이용하여 출력할 Volt 값의 Scale 을 설정합니다.

ScaleMin 은 해당 채널을 통해서 출력할 수 있는 최소 전압(/전류) 값을 의미하며, ScaleMax 는 해당 채널을 통해서 출력할 수 있는 최대 전압(/전류)값을 의미합니다.

일반적으로 아날로그출력 장치는 출력 데이터값을 정수값으로 전달받습니다. 따라서 사용자가 요구한 출력 전압(/전류)값을 아날로그출력 장치가 원하는 정수값으로 변환하여 전달해야 합니다. 이 변환을 위해서는 해당 채널의 출력 범위를 알아야 하며, 그 출력 범위의 정보가 ScaleMin 과 ScaleMax 값입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 출력 채널 번호
- ▶ OutData : 출력하고자 하는 전압(/전류) 값
- ▶ ScaleMin : 대상 채널의 최소 출력 전압(/전류)
- ▶ ScaleMax : 대상 채널의 최대 출력 전압(/전류)
- ▶ 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.

## EXAMPLE

---

< 예제 설명 >

-10 ~ 10 V 의 출력 범위를 가지는 AO CH0 채널을 통해서 3 volt 의 전압을 출력하는 예

---

< 예제 코드 >

```
#define NETID 0
#define AO_CH 0
#define OUT_VOLT 3.0
```

```
ecaoSetChanVal_FS (NETID, AO_CH, 3.0, -10.0f, 10.0f, NULL);
```


---


## 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\_i32 ecaoGetOutValue\_I([in] t\_i32 NetID, [in] t\_ui32 Channel, [out] t\_i32 \*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


ecaoGetOutValue\_F


- 대상 아날로그 출력 채널의 출력 Volt 값 반환

### INFORMATION

 Analog I/O Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_f64 ecaoGetOutValue\_F([in] t\_i32 NetID, [in] t\_ui32 Channel, [out] t\_i32 \*ErrCode)

### 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.

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

## SYNOPSIS

□ t\_f64 ecaoGetOutValue\_FS([in] t\_i32 NetID, [in] t\_ui32 Channel, [in] t\_f32 ScaleMin, [in] t\_f32 ScaleMax, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 대상 아날로그 출력 채널의 출력 범위를 알고 있을 때에 출력 범위의 최소값(ScaleMin)과 최대값(ScaleMax) 정보를 인자로 제공하고 현재 출력되고 있는 출력 값을 전압(/전류) 값으로 반환하도록 하는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Channel : 대상 아날로그 채널
- ▶ ScaleMin : 대상 아날로그 채널의 출력 전압(/전류) 범위의 최소 값
- ▶ ScaleMax : 대상 아날로그 채널의 출력 전압(/전류) 범위의 최대 값
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

- \* 현재 출력되고 있는 출력 값을 전압(/전류) 값으로 반환합니다.



# Motion General 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.







## 13 Motion General Functions

### 13.1 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) 대상 네트워크에 연결되어 있는 Servo 축 List 를 반환합니다.</p>
<p>□ TEcmInPDO_Header* ecmGn_GetInPDOHeader ([in] t_i32 NetID, [out] t_i32 *ErrCode) Input PDO 영역의 Motion Section 의 헤더 정보에 대한 포인터를 반환합니다.</p>
<p>□ t_i32 ecmGn_GetAxisList([in] t_i32 NetID, [out] t_ui8 AxisListBuf[], [in] t_ui8 AxisListBufSize, t_i32 *ErrCode) 축 번호를 이용하여 SlaveIndex 를 반환 합니다.</p>
<p>□ t_i32 ecmGn_AxisToSlaveID ([in] t_i32 NetID, [in] int SlaveIndex, [out] t_i32 *ErrCode) 축 번호를 이용하여 SlaveID 를 반환 합니다.</p>
<p>□ t_i32 ecmGn_AxisToSlaveID ([in] t_i32 NetID, [in] int Axis, [out] t_i32 *ErrCode) SlaveIndex 를 이용하여 Axis 를 반환 합니다.</p>
<p>□ t_i32 ecmGn_SlaveIDToAxis ([in] t_i32 NetID, [in] int SlaveID, [out] t_i32 *ErrCode) SlaveIndex 를 이용하여 Axis 를 반환 합니다.</p>
<p>□ t_success ecmGn_InitFromFile ([in] t_i32 NetID, [in] t_i32 AxisCnt, [in] char *szMotCfgFile, [out] t_i32 *ErrCode)</p>

**13.2** 함수 설명

NAME	I N F O R M A T I O N
<p><b>ecmGn_GetAxisList</b></p> <p>- 네트워크 축 리스트 반환</p>	<p> Motion      General</p> <p>Function</p> <hr/> <p> VC++/BCB/.NET</p> <hr/> <p> Level 1</p> <hr/> <p> 위험 요소 없음</p>

**SYNOPSIS**

```

□ t_i32 ecmGn_GetAxisList([in] t_i32 NetID, [out] t_ui8 AxisListBuf[], [in] t_ui8
AxisListBufSize, t_i32 *ErrCode)

```

**DESCRIPTION**

이 함수는 대상 네트워크에 연결되어 있는 서버 축 리스트를 반환합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ AxisListBuf : 서버 축 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	I N F O R M A T I O N
ecmGn_GetInPDOHeader -	<div style="border-bottom: 1px solid black; padding-bottom: 2px;"> <span style="font-size: 0.8em;">Motion</span>      <span style="float: right;">General</span> </div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;"> <span style="font-size: 0.8em;">Function</span> </div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;"> <span style="font-size: 0.8em;">VC++/BCB/.NET</span> </div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;"> <span style="font-size: 0.8em;">Level 1</span> </div> <div style="padding-bottom: 2px;"> <span style="font-size: 0.8em;">위험 요소 없음</span> </div>

## SYNOPSIS

```
□ TEcmInPDO_Header* ecmGn_GetInPDOHeader([in] t_i32 NetID, [out] t_i32 *ErrCode)
```

### DESCRIPTION

이 함수는 Input PDO 영역에서 Motion Control 영역의 헤더 정보를 가리키는 포인터를 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

\* Input PDO 영역에서 Motion Control 영역의 헤더 정보가 들어 있는 메모리 주소를 반환합니다. 이 메모리 주소는 TEcmInPDO\_Header 데이터형 포인터로 형변환하여 사용할 수 있습니다. TEcmInPDO\_Header 구조체는 'ComiEcatSdk.h' 파일에서 아래와 같이 정의되어 있습니다.

```
typedef struct{
    t_ui32 HEmgState:1; ///< Motion Hard-EMG 상태
    t_ui32 SEmgState:1; ///< Motion Soft-EMG 상태
    t_ui32 Resv: 30;
}TEcmInPDO_Header;
```

### REFERENCE

- Hardware Emergency 상태는 TEcmInPDO\_Header ::HEmgState 값 대신에 ecmHEMG\_GetState 함수를 통해서도 확인할 수 있습니다.
- Software Emergency 상태는 TEcmInPDO\_Header ::SEmgState 값 대신에 ecmSEMG\_GetState 함수를 통해서도 확인할 수 있습니다.

### EXAMPLE

---

```
C/C++
```

---

---

---

```
TEcmInPDO_Header *pMotIPDOHeader = ecmGn_GetInPDOHeader(0, NULL);  
if(pMotIPDOHeader && pMotIPDOHeader->HEmgState == ON){  
    ecUtl_ShowMessage_A (GetSafeHwnd(), "Hard-Emergency 스위치가 ON 되었습니다.");  
}
```

---

NAME	INFORMATION
<p><code>ecmGn_AxisToSlaveIndex</code> - Axis 로 SlaveIndex 반환</p>	<p>Motion      General Function</p> <hr/> <p>VC++/BCB/.NET</p> <hr/> <p>Level 1</p> <hr/> <p>위험 요소 없음</p>

## SYNOPSIS

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

### DESCRIPTION





이 함수는 Axis 인자를 통해서 지정된 축이 속해 있는 슬레이브 장치의 슬레이브 인덱스(연결 순서)를 반환 합니다.

### PARAMETER

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

### RETURN VALUE

\* 지정된 축이 속해 있는 슬레이브 장치의 슬레이브 인덱스를 반환합니다.

<h2>NAME</h2> <p>ecmGn_AxisToSlaveID</p> <p>- Axis 로 SlaveID 반환</p>	<b>INFORMATION</b>	
	 Motion	General
	Function	
	 VC++/BCB/.NET	
	 Level 1	
 위험 요소 없음		

## SYNOPSIS

□ t\_i32 ecmGn\_AxisToSlaveID ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 Axis 인자를 통해서 지정된 축이 속해 있는 슬레이브 장치의 슬레이브 주소를 반환합니다.

### PARAMETER

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

### RETURN VALUE


\* 지정된 축이 속해 있는 슬레이브 장치의 슬레이브 주소를 반환합니다.


## NAME


`ecmGn_SlaveIndexToAxis`  
- SlaveIndex 로 Axis 반환

### INFORMATION

 Motion General Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

`t_i32 ecmGn_AxisToSlaveID ([in] t_i32 NetID, [in] t_i32 SlaveIndex, [out] t_i32 *ErrCode)`

### DESCRIPTION

이 함수는 `SlaveIndex` (슬레이브 인덱스, 연결 순서)인자로 지정된 슬레이브에서 제공하는 첫 번째 축에 해당하는 전역 축 번호를 반환 합니다.





### PARAMETER

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

### RETURN VALUE

\* 이 함수는 `SlaveIndex` (슬레이브 인덱스, 연결 순서)인자로 지정된 슬레이브에서 제공하는 첫 번째 축에 해당하는 전역 축 번호를 반환 합니다.



<h2>NAME</h2> <p>ecmGn_SlaveIDToAxis</p> <p>- Axis 로 SlaveIndex 반환</p>	<b>INFORMATION</b>	
	 Motion	General
	Function	
	 VC++/BCB/.NET	
	 Level 1	
 위험 요소 없음		

## SYNOPSIS

□ t\_i32 ecmGn\_SlaveIDToAxis ([in] t\_i32 NetID, [in] t\_i32 SlaveID, [out] t\_i32 \*ErrCode)

### DESCRIPTION





이 함수는 SlaveID (슬레이브 아이디, 슬레이브 주소)인자로 지정된 슬레이브에서 제공하는 첫 번째 축에 해당하는 전역 축 번호를 반환 합니다.

### PARAMETER

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

### RETURN VALUE

\* 이 함수는 SlaveID (슬레이브 아이디, 슬레이브 주소)인자로 지정된 슬레이브에서 제공하는 첫 번째 축에 해당하는 전역 축 번호를 반환 합니다.

NAME	I N F O R M A T I O N
ecmGn_InitFromFile -	<div data-bbox="1102 347 1406 414">  Motion      General           </div> <div data-bbox="1102 414 1406 481">  VC++/BCB/.NET           </div> <div data-bbox="1102 481 1406 548">  Level 1           </div> <div data-bbox="1102 548 1406 560">  위험 요소 없음           </div>

## SYNOPSIS

t\_i32 ecmGn\_InitFromFile ([in] t\_i32 NetID, [in] t\_i32 AxisCnt, [in] char \*szMotCfgFile, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

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

## RETURN VALUE

\* 이 리턴값은 대상의 SlaveIndex 가 반환 됩니다.

# 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)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.



## 14 Motion Single Axis Functions

### 14.1 함수 요약

Summary of Functions
<p>□ t_cmdidx ecmSxCfg_SetMioProp([in]t_i32 NetID, [in] t_i32 Axis, [in] EEcmMioPropId PropId, [in] t_i32 PropVal, [out] t_i32 *ErrCode) 모션 입출력 신호의 환경설정을 구성합니다</p>
<p>□ t_i32 ecmSxCfg_GetMioProp([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmMioPropId PropId, [out] t_i32 *ErrCode) 모션 입출력 신호의 환경설정 값을 반환합니다.</p>
<p>□ t_cmdidx ecmSxCfg_SetSpeedPatt([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) 모션 이송의 전역 기준속도를 설정합니다. 이 속도의 비율을 통해 모션 이송의 실제 속도를 설정할 수 있습니다.</p>
<p>□ 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) 모션 이송의 전역 기준속도를 설정합니다. 이 속도의 비율을 통해 모션 이송의 실제 속도를 설정할 수 있습니다.</p>
<p>□ 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) 모션 이송의 전역 기준 속도를 반환합니다. 반환된 이 속도의 비율을 통해 모션 이송의 실제 속도가 설정됩니다.</p>
<p>□ t_cmdidx ecmSxCfg_SetSpeedPatt_Time ([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) 모션의 이송 전역 기준속도를 설정합니다. 가속도를 초단위로 설정합니다.</p>
<p>□ 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) 모션의 이송 전역 기준속도를 설정합니다. 가속도를 초단위로 설정합니다.</p>
<p>□ 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) 모션의 이송 전역 기준속도를 반환합니다. 가속도를 초단위로 반환합니다.</p>
<p>□ t_cmdidx ecmSxCfg_SetSpeedRatio([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 SpeedRatio, [out] t_i32 *ErrCode) 모션의 이송 전역 기준속도를 비율로 설정합니다..</p>
<p>□ t_cmdidx ecmSxCfg_SetSpeedRatio_FF([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 SpeedRatio, [out] t_i32 *ErrCode) 모션의 이송 전역 기준속도를 비율로 설정합니다..</p>
<p>□ t_f64 ecmSxCfg_GetSpeedRatio([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 모션의 이송 전역 기준속도를 비율로 반환합니다..</p>

<p>□ t_cmdidx emcSxCfg_SetJerkRatio([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode)                  지정된 축이 S-Curve 속도 프로파일을 사용하는 경우에 가/감속 구간의 시간에 대해서 Jerk 가 0 이 아닌 구간의 시간 비율을 설정합니다.</p>
<p>□ t_cmdidx emcSxCfg_SetJerkRatio_FF([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode)                  지정된 축이 S-Curve 속도 프로파일을 사용하는 경우에 가/감속 구간의 시간에 대해서 Jerk 가 0 이 아닌 구간의 시간 비율을 설정합니다.</p>
<p>□ t_f64 emcSxCfg_GetJerkRatio([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  지정된 축의 현재 설정된 Jerk 시간 비율을 반환합니다.</p>
<p>□ t_success ecmSxCfg_SetMinAccDecTime ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 MinAccTime, [in] t_f64 MinDecTime, [in] t_f64 LowLimitVel, [out] t_i32 *ErrCode)                  최소 가감속 시간 설정</p>
<p>□ t_success ecmSxCfg_GetMinAccDecTime)( [in] t_i32 NetID, [in] t_i32 Axis, [out] t_f64 *MinAccTime, [out] t_f64 *MinDecTime, [out] t_f64 *LowLimitVel, [out] t_i32 *ErrCode)                  최소 가감속 시간 설정값 읽기</p>
<p>□ t_cmdidx ecmSxCfg_SetTouchProbeFunc ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui8 TouchProbeIndex, [in] t_byte TouchProbeFuncVal, [out] t_i32 *ErrCode)                  터치프루브(Touch-probe) 기능 설정</p>
<p>□ t_byte ecmSxCfg_GetTouchProbeFunc([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui8 TouchProbeIndex, [out] t_i32 *ErrCode)                  터치프루브(Touch-probe) 기능 설정 읽기</p>
<p>□ t_success ecmSxCfg_SetUnitDist([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 UnitDist, [out] t_i32 *ErrCode)                  논리적 단위 거리에 대한 펄스 수를 설정합니다.</p>
<p>□ t_f64 ecmSxCfg_GetUnitDist([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  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 출력속도(PPS)를 설정합니다</p>
<p>□ t_f64 ecmSxCfg_GetUnitDist([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  논리적 단위 속도에 대한 실제 펄스 출력속도(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 기능 설정(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 읽기(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 기능 설정(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 읽기(위치 값을 펄스 단위로 읽음)</p>

<p>□ t_success ecmSxCfg_SetSoftLimit ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_f64 NegLimit, [in] t_f64 PosLimit, [in] t_bool IsDecelStop, [out] t_i32 *ErrCode) SoftLimit 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) 설정된 SoftLimit 값을 확인합니다.</p>
<p>□ t_success ecmSxCfg_SetOperMode ([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmOperMode OperMode, [out] t_i32 *ErrCode); 모터의 제어 모드 설정</p>
<p>□ t_i32 ecmSxCfg_GetOperMode([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 현재 설정된 모터의 제어 모드 설정값 읽기</p>
<p>□ t_success ecmSxCfg_PosCorr_SetTableSize ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TableSize, [out] t_i32 *ErrCode) 위치 보정 값을 저장하기 위한 테이블 크기를 조절 하기 위한 함수입니다.</p>
<p>□ t_i32 ecmSxCfg_PosCorr_GetTableSize ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 위치 보정 값을 저장하기 위한 테이블 크기를 반환 하기 위한 함수입니다.</p>
<p>□ t_i32 ecmSxCfg_PosCorr_SetTableFromFile ([in] t_i32 NetID, [in] t_i32 Axis, [in] char *szFilePath, [out] t_i32 *ErrCode) 테이블 데이터 값을 파일에 저장 되어 있는 값을 로드하여 테이블에 저장하는 함수입니다.</p>
<p>□ t_success ecmSxCfg_PosCorr_SetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [out] t_i32 *ErrCode) 축의 상태를 위치 보정 테이블에 저장 된 값에 대한 조정을 Enable 해주는 함수입니다.</p>
<p>□ t_bool ecmSxCfg_PosCorr_GetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 축의 상태를 위치 보정 테이블에 저장 된 값에 대한 조정을 Disnable 하는 함수입니다</p>
<p>□ t_success ecmSxCfg_PosCorr_ClearTable ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 위치 보정 테이블을 해제 해주는 함수 입니다.</p>
<p>□ t_success ecmSxCfg_Ring_SetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable [out] t_i32 *ErrCode) 링 카운터 사용 여부를 설정 하는 함수 입니다.</p>
<p>□ t_bool ecmSxCfg_Ring_GetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 링 카운터 사용 여부를 반환 하는 함수 입니다.</p>
<p>□ t_success ecmSxCfg_Ring_SetPosRange ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 RingPosRange, [out] t_i32 *ErrCode) 링 카운터의 범위를 설정 하는 함수 입니다.</p>
<p>□ t_f64 ecmSxCfg_Ring_GetPosRange ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 링 카운터의 범위를 반환 하는 함수 입니다.</p>

<p>□ t_success ecmSxCfg_Ring_SetDirMode ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 RingDirMode, [out] t_i32 *ErrCode) 링카운트의 모드를 설정을 하는 함수입니다..</p>
<p>□ t_i32 ecmSxCfg_Ring_GetDirMode ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 링카운트의 모드를 반환을 하는 함수입니다..</p>
<p>□ t_success ecmSxCfg_ExtStop_SetEnable([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 활성화 속성 설정</p>
<p>□ t_bool ecmSxCfg_ExtStop_GetEnable([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 활성화 속성 읽기</p>
<p>□ t_success ecmSxCfg_ExtStop_SetEnv([in] t_i32 NetID, [in] t_i32 Axis, [in] TEcLogicAddr SigAddr, [in] t_i32 ActiveState, [in] t_i32 DelayTime, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정(환경설정방식 1)</p>
<p>□ t_i32 ecmSxCfg_ExtStop_GetEnv([in] t_i32 NetID, [in] t_i32 Axis, [out] TEcLogicAddr *pSigAddr, [out] t_i32 *pActiveState, [out] t_i32 *pDelayTime, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정 읽기(환경설정방식 1)</p>
<p>□ t_success ecmSxCfg_ExtStop_SetEnv2([in] t_i32 NetID, [in] t_i32 Axis, [in] TEcLogicAddr SigAddr, [in] t_i32 ActiveLogic, [in] t_f64 OfsDist, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정</p>
<p>□ t_i32 ecmSxCfg_ExtStop_GetEnv2([in] t_i32 NetID, [in] t_i32 Axis, [out] TEcLogicAddr *pSigAddr, [out] t_i32 *ActiveLogic, [out] t_f64 *pOfsDist, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정(환경설정방식 2)</p>
<p>□ t_success ecmSxCfg_ExtStop_SetEnv3([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui8 TouchProbeIndex, [in] t_i32 ActiveLogic, [in] t_f64 OfsDist, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정 읽기(환경설정방식 2)</p>
<p>□ t_i32 ecmSxCfg_ExtStop_GetEnv3([in] t_i32 NetID, [in] t_i32 Axis, [out] t_ui8 *TouchProbeIndex, [out] t_i32 *ActiveLogic, [out] t_f64 *pOfsDist, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정(환경설정방식 3)</p>
<p>□ t_success ecmSxCfg_ExtStop_SetOfsDistMode([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsHardOfsDistMode, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능의 환경 설정 읽기(환경설정방식 3)</p>
<p>□ t_bool ecmSxCfg_ExtStop_GetOfsDistMode([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 외부센서스탑(External Sensor Stop) 기능에서 센서 감지 후 정지 까지의 거리 조건 설정</p>
<p>□ t_success ecmSxCfg_AutoTorq_SetValMode ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 ValMode, [in] t_i32 NumMultiVals, [out] t_i32 *ErrCode) 자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)를 설정 합니다..</p>
<p>□ t_i32 ecmSxCfg_AutoTorq_GetValMode ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *NumMultiVals, [out] t_i32 *ErrCode) 자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)를 반환 합니다..</p>

<p>□ t_success ecmSxCfg_AutoTorq_SetMultiVal ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 ValIndex, [in] t_i32 TorqVal, [in] t_i32 Duration, [out] t_i32 *ErrCode)                  자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)가 'Multi-Value Mode'로 설정된 경우에만 사용되는 함수로서, Multi-Value 리스트의 아이템 각각의 토크 값과 출력 시간을 정의하는 함수입니다.</p>
<p>□ t_success ecmSxCfg_AutoTorq_GetMultiVal ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 ValIndex, [out] t_i32 *TorqVal, [out] t_i32 *Duration, [out] t_i32 *ErrCode)                  자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)가 'Multi-Value Mode'로 설정된 경우에만 사용되는 함수로서, Multi-Value 리스트의 아이템 각각의 토크 값과 출력 시간을 정의를 반환하는 함수입니다.</p>
<p>□ t_success ecmSxCfg_AutoTorq_SetValue ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 OutTorqVal, [out] t_i32 *ErrCode)                  자동토크출력 기능에서의 토크 값 설정 모드(Torque Value Mode)가 'Single-Value Mode'로 설정된 경우에만 사용하는 함수로서, 토크 모드로 전환되었을 때 출력되어야 하는 토크 값을 정의하는 함수입니다.</p>
<p>□ t_i32 ecmSxCfg_AutoTorq_GetValue ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  자동토크출력 기능에서의 토크 값 설정 모드(Torque Value Mode)가 'Single-Value Mode'로 설정된 경우에만 사용하는 함수로서, 토크 모드로 전환되었을 때 출력되어야 하는 토크 값을 정의를 반환하는 함수입니다..</p>
<p>□ t_success ecmSxCfg_AutoTorq_SetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_bool IsOneShotEnable, [out] t_i32 *ErrCode)                  자동토크출력 기능을 활성화/비 활성화 하는 함수입니다.</p>
<p>□ t_bool ecmSxCfg_AutoTorq_GetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool* IsOneShotEnable, [out] t_i32 *ErrCode)                  자동토크출력 기능을 상태를 반환 하는 함수입니다.</p>
<p>□ t_success ecmSxCfg_SetMaxTorqOfCV([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 MaxTorq, [out] t_i32 *ErrCode)                  속도제어(Cyclic Velocity) 모드에서 토크의 최대값을 설정</p>
<p>□ t_ui16 ecmSxCfg_GetMaxTorqOfCV([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  속도제어(Cyclic Velocity) 모드에서 토크의 최대값 설정을 읽기</p>
<p>□ t_success ecmSxCfg_SetMaxSpdOfCT([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui32 MaxSpd, [out] t_i32 *ErrCode)                  토크제어(Cyclic Torque) 모드에서 속도의 최대값을 설정</p>
<p>□ t_ui32 ecmSxCfg_GetMaxSpdOfCT([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  토크제어(Cyclic Torque) 모드에서 속도의 최대값 설정 읽기</p>
<p>□ t_success ecmSxCfg_SetMaxProfSpdOfCT([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 MaxProfSpd, [out] t_i32 *ErrCode)                  토크제어(Cyclic Torque) 모드에서 프로파일 속도의 최대값 설정 읽기</p>
<p>□ t_f64 ecmSxCfg_GetMaxProfSpdOfCT([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  토크제어(Cyclic Torque) 모드에서 프로파일 속도의 최대값 설정 읽기</p>



<p>□ t_cmdidx ecmSxCfg_SetSyncOtherEnv ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SyncAxis, [in] t_i32 SyncType, [in] t_i32 PosSyncMethod, [in] t_f64 SyncPosition, [out] t_i32 *ErrCode) 한 축과 또 다른 축을 동기화 하여 이동하는 함수입니다.</p>
<p>□ t_success ecmSxCfg_GetSyncOtherEnv ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32* SyncAxis, [out] t_i32* SyncType, [out] t_i32* PosSyncMethod, [out] t_f64* SyncPosition, [out] t_i32 *ErrCode) 한 축과 또 다른 축을 동기화 하여 이동하는 함수입니다.</p>
<p>□ t_cmdidx ecmSxCfg_SetSyncOtherEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_bool IsOneShot, [out] t_i32 *ErrCode) 자동토크출력 기능을 활성화/비활성화 하는 함수입니다.</p>
<p>□ t_success ecmSxCfg_GetSyncOtherEnable ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_bool* IsEnable, [out] t_bool* IsOneShot, [out] t_i32 *ErrCode) 자동토크출력 기능 상태를 반환 하는 함수입니다</p>
<p>□ t_cmdidx ecmSxCtl_SetSvon([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SvonVal, [out] t_i32 *ErrCode) 서보 드라이브에 SERVO ON 신호 출력을 인가 혹은 차단 합니다.</p>
<p>□ t_cmdidx ecmSxCtl_SetSvon_FF([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 SvonVal, [out] t_i32 *ErrCode) 서보 드라이브에 SERVO ON 신호 출력을 인가 혹은 차단 합니다.</p>
<p>□ t_cmdidx ecmSxCtl_SetAlmRst([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 IsSetAlmRst, [out] t_i32 *ErrCode) 알람 리셋 신호 출력을 제어 합니다,</p>
<p>□ t_cmdidx ecmSxCtl_SetAlmRst_FF([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 IsSetAlmRst, [out] t_i32 *ErrCode) 알람 리셋 신호 출력을 제어 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 ecmSxCtl_ResetAlm_FF([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_VMoveStart_FF([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_cmdidx ecmSxMot_MoveStart_FF([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_cmdidx ecmSxMot_MoveToStart_FF([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 ecmSxMulTorq1_Start([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i16 InitialTorq, [in] TEcmMTQ1Item NextTorqList[], [in] t_ui16 NumNextTorqItems, [out] t_i32 *ErrCode)  멀티토크 출력 모드를 시작합니다</p>
<p>□ t_cmdidx ecmSxMulTorq1_Stop([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i16 StopTorq, [in] t_bool IsApplyStopTorq, [out] t_i32 *ErrCode)  멀티토크 출력 모드를 중지합니다.</p>
<p>□ t_success ecmSxMot_SetTargTorq([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargTorq, [in] [out] t_i32 *ErrCode)  토크제어모드에서 목표 토크 값을 설정합니다.</p>
<p>□ t_i32 ecmSxMot_GetTargTorq([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)  현재 설정된 목표 토크 값을 반환합니다.</p>

<p>□ t_success ecmSxMot_SetTargVel([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 TargVel, [out] t_i32 *ErrCode) 속도제어 모드에서 목표 속도를 설정합니다.</p>
<p>□ t_f64 ecmSxMot_GetTargVel([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 현재 설정된 목표 속도를 반환합니다.</p>
<p>□ t_success ecmSxCmpOne_SetHandler([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 HandlerType, [in] HANDLE Handler, [in] UINT nMessage, [in] LPARAM lParam, [out] t_i32 *ErrCode) 'One-shot' 모드의 위치비교출력 기능의 이벤트 핸들러를 설정합니다.</p>
<p>□ t_success ecmSxCmpOne_SetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_i32 GdoChanIdx, [out] t_i32 *ErrCode) 'One-shot' 모드의 위치비교출력 기능을 활성화 합니다.</p>
<p>□ t_success ecmSxCmpOne_GetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_i32 GdoChanIdx, [out] t_i32 *ErrCode) 'One-shot' 모드의 위치비교출력 기능을 비 활성화 합니다.</p>
<p>□ t_success ecmSxCmpOne_SetEnv ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 CmpCntrType, [in] t_i32 CmpMethod, [in] t_f64 CmpRefPos, [in] t_i32 OutSigLogic, [in] t_i32 OutSigOnTime, [out] t_i32 *ErrCode) 'One-shot' 모드 위치비교출력 기능을 위한 비교 조건 및 기타 환경을 설정한다.</p>
<p>□ t_success ecmSxCmpOne_GetEnv ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32* CmpCntrType, [out] t_i32* CmpMethod, [out] t_f64* CmpRefPos, [out] t_i32* OutSigLogic, [out] t_i32* OutSigOnTime, [out] t_i32 *ErrCode) 'One-shot' 모드 위치비교출력 기능을 위한 비교 조건 및 기타 환경을 반환한다.</p>
<p>□ t_success ecmSxCmpOne_GetOutSts ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *OutCount, [out] t_f64 *LastOutPos, [out] t_i32 *ErrCode) 함수는 'One-shot' 모드 위치비교출력 기능의 동작 상태를 매개 변수들을 통해서 반환해주는 함수 입니다.</p>
<p>□ t_success ecmSxCmpOne_ResetOutSts ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) ecmSxCmpOne_GetOutSts() 함수를 통해서 반환되는 CMP 신호가 출력된 횟수 카운트값(OutCount)과 마지막 트리거 위치(LastOutPos)를 모두 0 으로 리셋합니다..</p>
<p>□ t_success ecmSxCmpCont_SetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [in] t_i32 GdoChanIdx, [out] t_i32 *ErrCode) Continuous 모드의 위치비교출력 기능을 활성화 한다.</p>
<p>□ t_success ecmSxCmpCont_GetEnable ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32* GdoChanIdx, [out] t_i32 *ErrCode) Continuous 모드의 위치비교출력 기능을 비 활성화 한다.</p>
<p>□ t_success ecmSxCmpCont_SetTableSize ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TableSize, [out] t_i32 *ErrCode) Continuous 모드 위치비교출력 기능을 위한 데이터 테이블의 크기를 설정하는 함수입니다</p>
<p>□ t_i32 ecmSxCmpCont_GetTableSize ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) Continuous 모드 위치비교출력 기능을 위한 데이터 테이블의 크기를 설정을 반환 하는 함수입니다.</p>





<p>□ t_success ecmSxCmpCont_SetTableData ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TableIndex, [in] t_i32 CmpCntrType, [in] t_i32 CmpMethod, [in] t_f64 CmpRefPos, [in] t_i32 OutSigLogic, [in] t_i32 OutSigOnTime, [out] t_i32 *ErrCode);          ‘Continuous’ 모드 위치비교출력 기능을 위한 비교 조건 데이터 테이블의 데이터를 정의하는 함수이다.</p>
<p>□ t_success ecmSxCmpCont_GetTableData ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TableIndex, [out] t_i32* CmpCntrType, [out] t_i32* CmpMethod, [out] t_f64* CmpRefPos, [out] t_i32* OutSigLogic, [out] t_i32* OutSigOnTime, [out] t_i32 *ErrCode);          ‘Continuous’ 모드 위치비교출력 기능을 위한 비교 조건 데이터 테이블의 데이터를 반환하는 함수이다.</p>
<p>□ t_success ecmSxCmpCont_SetActTblIdx ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TableIndex, [out] t_i32 *ErrCode)          Continuous 모드 위치비교출력 기능의 데이터 테이블에 있는 아이템들 중에서 실행 비교기에 로드되어야 하는 아이템의 인덱스를 지정하는 함수입니다.</p>
<p>□ t_i32 ecmSxCmpCont_GetActTblIdx ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)          Continuous 모드 위치비교출력 기능의 데이터 테이블에 있는 아이템들 중에서 실행 비교기에 로드되어야 하는 아이템의 인덱스를 반환 하는 함수입니다.</p>
<p>□ t_success ecmSxCmpCont_GetOutSts ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *OutCount, [out] t_f64 *LastOutPos, [out] t_i32 *ErrCode)          ‘Continuous’ 모드 위치비교출력 기능의 동작 상태를 매개 변수들을 통해서 반환해주는 함수입니다.</p>
<p>□ t_success ecmSxCmpCont_ResetOutSts (t_i32 NetID, t_i32 Axis, t_i32 *ErrCode)          ecmSxCmpOne_GetOutSts() 함수를 통해서 반환되는 CMP 신호가 출력된 횟수 카운트값(OutCount)과 마지막 트리거 위치(LastOutPos)를 모두 0 으로 리셋합니다..</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)          단축 이송을 정지 합니다.</p>
<p>□ t_cmdidx ecmSxMot_Stop_FF([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 IsDecStop, [out] t_i32 *ErrCode)          단축 이송을 정지 합니다.</p>
<p>□ t_bool ecmSxSt_IsBusy([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)          단축 이송의 완료를 확인합니다</p>
<p>□ t_success ecmSxSt_WaitCompt([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)          단축 이송의 완료 시점까지 대기 합니다</p>
<p>□ t_success ecmSxSt_WaitCompt_NB([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)          단축 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 모션 채널의 지정한 카운터의 값을 전달된 매개변수를 통해 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 모션 채널의 지정한 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 채널의 지정한 카운터의 값을 전달된 매개변수를 통해 설정합니다. 단, 이때 지정하는 카운터값의 단위는 논리적인 거리 단위입니다.</p>
<p>□ t_f64 ecmSxSt_GetPosition([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode)                  대상 채널의 지정한 카운터 값을 반환합니다. 단, 이때 반환되는 카운터값의 단위는 논리적인 거리 단위입니다.</p>
<p>□ t_f64 ecmSxSt_GetCurSpeed([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TargCntr, [out] t_i32 *ErrCode)                  대상 채널의 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 채널의 토크값을 확인하여 반환합니다.</p>
<p>□ t_i32 ecmSxSt_GetMotState([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *SubErrData)                  대상 모션 채널의 모션 상태를 확인합니다.</p>
<p>□ t_word ecmSxSt_GetFlags ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  단일 축에 대해서 모션 상태에 관련된 상태 Flag 값들이 반환됩니다</p>
<p>□ t_bool ecmSxSt_IsConnected([in] t_i32 NetID, [in] t_i32 Axis, [out] EEcmConnStsDetail *ConnStsDetail, [out] t_i32 *ErrCode)                  지정한 축의 연결 상태를 반환합니다.</p>
<p>□ void* ecmSxSt_GetInPDOPtr([in] t_i32 NetID, [in] t_i32 Axis, [out] t_ui8 *PDODataType, [out] t_i32 *ErrCode)                  지정한 축의 Input PDO 데이터의 포인터를 반환합니다.</p>
<p>□ t_word ecmSxSt_GetDI([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  지정한 축에서 제공하는 디지털입력 상태 값을 반환합니다.</p>
<p>□ t_byte ecmSxSt_GetTouchProbeSts([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TouchProbeIndex, [out] t_i32 *ErrCode)                  지정한 축에서 제공하는 터치프루브(Touch-probe) 기능의 상태를 반환합니다.</p>
<p>□ t_f64 ecmSxSt_GetTouchProbePos([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 TouchProbeIndex, [out] t_i32 *ErrCode)                  지정한 축의 터치프루브(Touch-probe) 위치 값을 반환합니다.</p>
<p>□ t_ui8 ecmSxSt_GetOpModeDisp([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  지정한 축의 현재 설정된 운전 모드(Operation Mode)를 반환합니다.</p>
<p>□ t_i32 ecmSxSt_GetLmMapIdx([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode);                  지정한 축이 속한 리스트모션 맵 번호를 반환합니다.</p>
<p>□ t_i32 ecmSxSt_GetPtmMapIdx([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode);                  지정한 축이 속한 PT-모션 맵 번호를 반환합니다.</p>
<p>□ t_success ecmSxSD_SetInputEnv([in] t_i32 NetID, [in] t_i32 Axis, [in] t_dword LogBitAddr, [in] t_bool IsInvertLogic, [in] t_i32 FilterCount, [out] t_i32 *ErrCode)                  SD 입력 신호 환경 설정</p>
<p>□ t_success ecmSxSD_GetInputEnv([in] t_i32 NetID, [in] t_i32 Axis, [out] t_dword *LogBitAddr, [out] t_bool *IsInvertLogic, [out] t_i32 *FilterCount, [out] t_i32 *ErrCode)                  SD 입력 신호 환경 설정 값 반환</p>

<p>□ t_success ecmSxSD_SetSpeed([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 SdSpeed, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) 속도를 설정합니다.</p>
<p>□ t_f64 ecmSxSD_GetSpeed([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) 속도 설정 값을 반환합니다.</p>
<p>□ t_success ecmSxSD_SetOffset([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 OffsetMode, [in] t_f64 Offset, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) Offset 값을 설정합니다.</p>
<p>□ t_success ecmSxSD_GetOffset([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *OffsetMode, [out] t_f64 *Offset, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) Offset 설정 값을 반환합니다.</p>
<p>□ t_success ecmSxSD_SetLatchMode([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsLatchMode, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) 신호의 Latch 모드를 설정합니다.</p>
<p>□ t_bool ecmSxSD_GetLatchMode([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  지정한 축의 SD 신호의 Latch 모드 설정 값을 반환합니다.</p>
<p>□ t_success ecmSxSD_SetEnable([in] t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) 기능의 활성화 속성을 설정합니다.</p>
<p>□ t_bool ecmSxSD_GetEnable([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  지정한 축의 SD(Slow Down) 기능의 활성화 속성을 반환합니다.</p>
<p>□ t_success ecmSxSD_RestoreSpeed ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  Slow Down 속도로 진행 중인 경우 원래의 속도로 복귀</p>
<p>□ t_bool ecmSxSD_GetInputStatus([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  SD 신호의 상태 확인</p>
<p>□ t_bool ecmSxSD_GetActStatus([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)                  SD 속도로 진행중인지에 대한 상태 반환</p>

## 14.2 함수 설명

### 14.2.1 기본 환경 설정

NAME	INFORMATION
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	EL 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 신호의 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	ECAT 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	ELN 신호를 무시하라는 옵션에 대한 설정 0 : Disable(Default) 1 : Enable
71 or ecmMPID_IGNORE_ELP	ELP 신호를 무시하라는 옵션에 대한 설정 0 : Disable(Default) 1 : Enable

▶ 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 단위 거리에 대한 펄스 수를 설정합니다. 여기서 논리적 단위 거리라 함은 Move 함수에서 사용하는 거리 또는 위치에 대한 단위량을 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 출력되어져야 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 기본적으로는 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, (쥬커미조아 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.

$$D_u = P_r / L_r$$

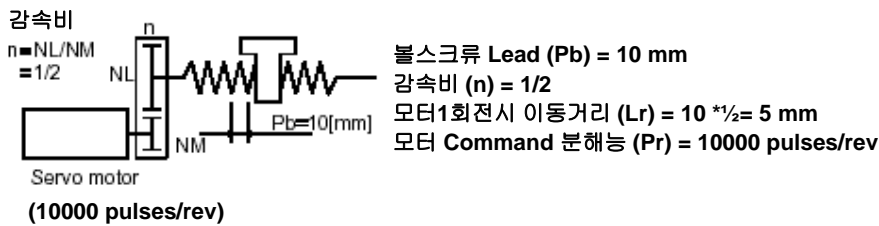
Pr : 모터 1 회전에 필요한 펄스수 (모터의 Command 분해능)

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.



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);
```

---

<h2>NAME</h2> <p>ecmSxCfg_SetUnitSpeed</p> <p>ecmSxCfg_GetUnitSpeed</p> <p>- 논리적 속도 단위 설정 및 반환</p>	<h3>INFORMATION</h3>
	<ul style="list-style-type: none"> <li> Single Axis Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>

## 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

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

## 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 값을 읽는 함수입니다.

### 인포지션(Inposition) 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 Inposition) 이란?

마스터 인포지션이라는 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 서보드라이버 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.

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 NULL 을 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.

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	INFORMATION
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 설정합니다.
- ▶ 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	I N F O R M A T I O N
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 SoftLimit 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.

<h2>NAME</h2> <p>ecmSxCfg_SetSpeedPatt ecmSxCfg_GetSpeedPatt</p> <p>- 모션 이송 기준 속도 설정 및 반환</p>	<h3>INFORMATION</h3>
	<p> Single Axis Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>

## SYNOPSIS

- ❑ t\_cmdidx ecmSxCfg\_SetSpeedPatt([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 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  $V_{ini}=0$  이라면  
 $acc = V_{work} / T_{acc}$ .

또는  
 $T_{acc} = V_{work} / acc$

▶ **Dec** : 이 인자는 감속도를 의미합니다. 참고로 감속도(dec)와 감속시간(Tacc)의 관계는 아래와 같습니다.  
 $dec = (V_{work} - V_{end}) / T_{dec}$ .

만일  $V_{end}=0$  이라면  
 $dec = V_{work} / T_{dec}$ .

또는  
 $T_{dec} = V_{work} / dec$

▶ **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.
- CONSTANT** 속도 모드  
 Constant 속도 모드에서는 **Motion** 을 수행할 때 가속/감속을 적용하지 않고 일정속도로 **Motion** 을 수행합니다. 여기서 적용되는 일정 속도는 **Vwork** 에서 주어진 값이 적용됩니다.
- CONSTANT** 속도 모드  
 Constant speed mode 에서는 **Motion** 을 수행할 때 가속/감속을 적용하지 않고 일정속도로 **Motion** 을 수행합니다. 여기서 적용되는 일정 속도는 **WorkSpeed** 에서 주어진 값이 적용됩니다.

**EXAMPLE**

\* **ecmSxMot\_MoveStart** EXAMPLE 참고

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmSxCfg_SetSpeedPatt_Time</p> <p style="margin: 0;">ecmSxCfg_GetSpeedPatt_Time</p> <p style="margin: 0;">- 모션 이송 기준 속도 시간 설정 및 반환</p>	<b>I N F O R M A T I O N</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

- t\_cmdidx ecmSxCfg\_SetSpeedPatt\_Time([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 시간단위로 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 값 입니다.
- ▶ VEnd : 이송 완료 시 속도에 대한 설정 및 반환 값 입니다.
- ▶ VWork : 작업 속도에 대한 설정 및 반환 값 입니다.
- ▶ 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` 함수의 반환 값 : 이 리턴값은 함수의 `cmdidx` 를 반환합니다.

\* `ecmSxCfg_GetSpeedPatt()` : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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 ecmSxCfg_GetSpeedRatio - 모션 기준 속도 비율 설정	<div style="border-bottom: 1px solid black; padding: 2px;">  Single Axis Function         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  VC++/BCB/.NET         </div> <div style="border-bottom: 1px solid black; padding: 2px;">  Level 1         </div> <div style="padding: 2px;">  위험 요소 없음         </div>

## SYNOPSIS

- t\_cmdidx ecmSxCfg\_SetSpeedRatio([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 함수는 지정한 축의 '이송 속도 비'를 설정합니다. 이송 속도 비는 이송할 때 적용되는 속도를 해당 축의 정격 속도





## 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
ecmSxCfg_SetJerkRatio	 Single Axis Function
ecmSxCfg_SetJerkRatio_FF	 VC++/BCB/.NET
ecmSxCfg_GetJerkRatio	 Level 1
- Jerk 시간 비율 설정 및 반환	 위험 요소 없음

## SYNOPSIS

- ❑ t\_cmdidx ecmSxCfg\_SetJerkRatio([in] t\_i32 NetID, [in] t\_i32 Axis, [in]t\_f64 JerkTimeRatio, [out] t\_i32 \*ErrCode)
- ❑ 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(), ecmSxCfg\_SetJerkRatio() : 이 리턴값은 함수의 cmdidx 를 반환합니다.

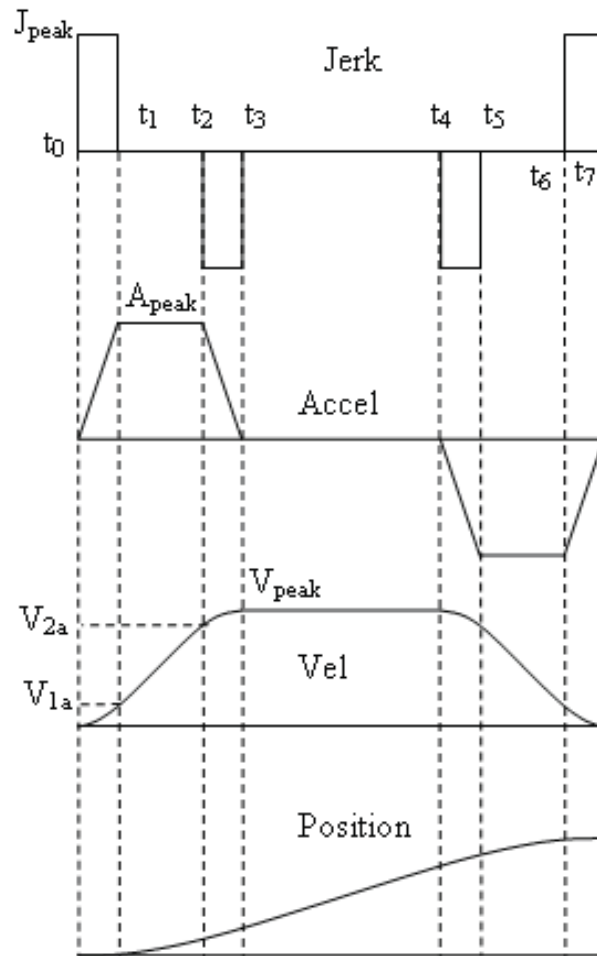
\* ecmSxCfg\_GetJerkRatio() : 이 리턴값은 JerkRatio 값을 반환합니다.

## EXAMPLE

- \* ecmSxMot\_MoveStart EXAMPLE 참고

REFERENCE





□ 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(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 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.



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

## SYNOPSIS

- t\_success ecmSxCfg\_SetMinAccDecTime ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 MinAccTime, [in] t\_f64 MinDecTime, [in] t\_f64 LowLimitVel, [out] t\_i32 \*ErrCode)
- t\_success ecmSxCfg\_GetMinAccDecTime ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_f64 \*MinAccTime, [out] t\_f64 \*MinDecTime, [out] t\_f64 \*LowLimitVel, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_SetMinAccDecTime 함수는 지정한 축의 최소 가속 시간과 감속 시간을 설정합니다.

ecmSxCfg\_GetMinAccDecTime 함수는 지정한 축의 최소 가속 시간과 감속 시간을 읽는 함수입니다.

최소 가/감속 시간은 이송 시에 지켜져야 하는 가속 구간과 감속 구간의 최소 시간을 말합니다. ecmSxCfg\_SetSpeedPatt() 함수 또는 ecmSxCfg\_SetSpeedPatt\_Time() 함수를 통해서 적절한 가속시간을 설정하였다더라도 이송거리가 너무 짧은 경우에는 작업속도 보정이 이루어져서 실제 적용되는 가속시간은 매우 짧아질 수 있습니다. 그런 경우에는 시스템 관성에 따라서 의도치 않은 진동이 발생할 수 있습니다.

이때 적절한 최소 가속 시간과 감속시간을 정의하여 주면 이러한 진동을 줄여줄 수 있습니다. 현재의 속도 패턴 설정을 기준으로 구동하였을 때 최소 가/감속 시간보다 짧은 가속 시간과 감속 시간이 형성될 것 같으면 모션제어기에서 자동으로 가/감속도를 조정하여 최소 가/감속 시간이 보장되도록 합니다.

단, 속도모드가 CONSTANT 모드로 설정된 경우에는 최소 가감속 시간 설정은 적용되지 않습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ MinAccTime : 최소 가속 시간. 시간의 단위톤 초(sec)입니다.
- ▶ MinDecTime : 최소 감속 시간. 시간의 단위톤 초(sec)입니다.
- ▶ LowLimitVel : 이 인자는 최소 가/감속 시간을 적용하는 최저 속도 한계를 의미합니다. 작업속도가 이 값보다 낮은 경우에는 최소 가/감속 시간 적용을 생략합니다.

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

## EXAMPLE

## 14.2.2 상태 제어 명령

NAME	INFORMATION
ecmSxCtl_SetSvon	 Single Axis Function
ecmSxCtl_GetSvon	 VC++/BCB/.NET
- 서보 On/Off 제어 / 상태반환	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmSxCtl\_SetSvon([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 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.");
}
}





```

---

## NAME

ecmSxCtl\_SetAlmRst  
 ecmSxCtl\_SetAlmRst\_FF  
 - 알람 리셋 신호 출력 제어

### INFORMATION

	DLL Function
	VC++/BCB/.NET
	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmSxCtl\_SetAlmRst([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 IsSetAlmRst, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmSxCtl\_SetAlmRst\_FF([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 IsSetAlmRst, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 지정한 채널(축)의 알람 리셋 출력을 제어합니다., 서보 드라이버를 사용하실 때는 외부에서 디지털 접점을 이용하여, 서보에서 발생한 알람 상황을 초기화 할 수 있도록 서보 드라이브 상에서 요구하는 입력 신호가 존재하는 데, 이 신호 입력에 대응하기 위해, 모션 보드에서는 기본으로 제공되는 디지털 출력 신호를 통해 디지털 출력 신호를 발생하게 되며, 이 신호를 알람 리셋 신호라 합니다. 이 함수를 통해 서보에 발생한 알람을 해제시킬 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsSetAlmRst : 출력 상태를 설정합니다. 출력 상태는 다음과 같이 설정할 수 있습니다.

Value	Meaning
0	출력 상태를 비 활성화 합니다.
1	출력 상태를 활성화 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 함수의 cmdidx 가 반환됩니다.

## NAME


`ecmSxCtl_ResetAlm`  
- 대상 서보 드라이버 알람 리셋

## INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- `t_cmdidx ecmSxCtl_ResetAlm([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)`
- `t_cmdidx ecmSxCtl_ResetAlm_FF([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)`

## DESCRIPTION

이 함수는 대상 채널(축)의 서보 드라이버에 발생한 알람을 리셋합니다.

## 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 함수는 `cmdidx` 가 반환됩니다. `CmdIdx` 는 ComiECAT SDK 라이브러리에서 발생시킨 명령들의 일련 번호입니다. 단, 이 값이 0 이면 함수의 수행이 실패했음을 의미합니다.





## EXAMPLE

```
C/C++
//본 예제는 ecmSxMot_Move 함수들을 사용하여 첫번 째 축을 (+)5000 이동하는 예 입니다.
#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);
    //0 번 축에 알람을 리셋 합니다.
}
```

### 14.2.3 이송 명령

<b>NAME</b>  <b>ecmSxMot_VMoveStart</b>  <b>- 단축 연속 속도 이송</b>	<b>INFORMATION</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_cmdidx ecmSxMot\_VMoveStart([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_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);

}

//*****
* 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;">ecmSxMot_MoveStart</p> <p style="margin: 0;">ecmSxMot_Move</p> <p style="margin: 0;">- 단축 상대 좌표 이송</p>	<b>INFORMATION</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_cmdidx ecmSxMot\_MoveStart([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 Distance, [out] t\_i32 \*ErrCode)

□ t\_success ecmSxMot\_Move([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

## RETURN VALUE

\* ecmSxMot\_MoveStart() 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 첫번 째 축을 (+)5000 이동하는 예 입니다.
#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(!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, 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 펄수 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 반환 값
Speed 설정 관련 매개 변수 값 */

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

t_f64 JerkTimeRatio = 1.0 //Jeck 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;
}
}
}

```

---

NAME	INFORMATION
ecmSxMot_MoveToStart	 Single Axis Function
ecmSxMot_MoveTo	 VC++/BCB/.NET
- 단축 절대 좌표 이송	 Level 1
	 위험 요소 없음

## SYNOPSIS

❑ t\_cmdidx ecmSxMot\_MoveToStart([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 Position, [out] t\_i32 \*ErrCode)

❑ t\_success ecmSxMot\_MoveTo([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 Position, [out] t\_i32 \*ErrCode)

## DESCRIPTION

하나의 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 블록킹(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 () 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 함수입니다.
* 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;
    }
}

```

---

<h2>NAME</h2> <p><b>ecmSxMot_OverrideSpeed</b> - 단축속도 오버라이딩 실행</p>	<b>INFORMATION</b>
	 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 오버라이딩하고자 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 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_i32 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
}

```

---



<b>NAME</b>  ecmSxMot_Stop - 단축 이송 정지	<b>INFORMATION</b>
	 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
}

```

---

## 14.2.4 모션 상태 감시

<h3>NAME</h3> <p><b>ecmSxSt_IsBusy</b> - 단축 모션 이송 확인</p>	<h3>INFORMATION</h3> <ul style="list-style-type: none"> <li> Single Axis Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>
<h3>SYNOPSIS</h3> <pre> t_bool ecmSxSt_IsBusy([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) </pre>	

### DESCRIPTION

이 함수는 단일 축에 대하여 모션 이송중인지를 확인합니다.

### 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	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
*****/

/*****
* DO_Busy: 작업 명령 시에 호출되는 가상의 함수입니다.
*****/





void DO_Busy()
{
    //cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
    //그대로 유지 하면서 두 축을 상대 위치 만큼 +5000 이동

    ecmSxMot_MoveStart(Device_Info.NetIdx, AXIS0, 5000, &Error_Num);
    //Move 두 축을 각각 절대 좌표 5000 으로 이동

    While(1)
    {
        if(!ecmSxSt_IsBusy (Device_Info.NetIdx, AXIS0, &Error_Num)
        {
            //이송 중이 아니면 break;
            break;
        }
    }
}

```

---

<b>NAME</b>  ecmSxSt_WaitCompt  ecmSxSt_WaitCompt_NB  - 단축 모션 완료 대기	<b>INFORMATION</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxSt\_WaitCompt([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)
- 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 하드웨어 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 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 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 이 리턴값은 함수의 `cmdidx` 가 반환됩니다.

**NAME**

ecmSxSt\_GetCount

- 사용자정의 하드웨어 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 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


`ecmSxSt_SetPosition`


- 사용자정의 논리적 카운트 값 설정

## INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## 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 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 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 이 리턴값은 함수의 `cmdidx` 가 반환됩니다.



**NAME**

ecmSxSt\_GetPosition

- 사용자정의 하드웨어 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 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 or Feedback 속도를 읽어서 논리적 속도 단위로 반환됩니다. TargCntr 매개변수에 따라서 Command or 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	I N F O R M A T I O N
ecmSxSt_GetCurTorque - 논리적 토크 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.

<b>NAME</b>	<b>INFORMATION</b>
ecmSxSt_GetMotState - 단축 이송 속도 상태 반환	Single Axis Function VC++/BCB/.NET Level 1 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 속도 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
}
```

---

<h1>NAME</h1> <p><b>ecmSxSt_GetFlags</b></p> <p>- 모션 상태 Flag 값 반환</p>	<b>INFORMATION</b>	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
	 Level 1	
 위험 요소 없음		

## SYNOPSIS

□ t\_word ecmSxSt\_GetFlags ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 단일 축에 대해서 모션 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 TEcmSxSt\_Flags 구조체를 이용하여 받아들일 수 있습니다. 구조체 안 매개 변수는 아래와 같습니다.

Parameter	Value & Meaning
RdyToSwOn	0 : Ready To Switch off 1 : Ready To Switch On
SwOn	0 : Switch Off 1 : Switch On
OperEnabled	0 : Servo-off 1 : Servo-on
ServoFault	Servo driver 의 Fault 발생 상태
VoltEnabled	Voltage Enabled
QuickStop	Quick Stop 발생 상태
SwOnDisabled	Switch On Disabled
ServoWarn	Servo driver 의 Warning 발생 상태
CtlrFault	Controller 의 Fault 발생 상태

HomeError	Homing중에 에러 발생 체크 상황
OMS1	Operation Mode Specific 1
IntLimit	Internal Limit Active 확인
OMS2	Operation Mode Specific 2
HomeBusy	Homing operation이 진행 체크 상황
HomeAttained	0 : Homing 동작이 실패 1 : Homing 동작이 성공적으로 완료되었음을 나타냄

## EXAMPLE

본 예제는 ecmSxSt\_GetFlags 함수를 이용하여 Servo On/Off 상태를 얻어 오는 예제입니다.

---

```

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

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





void Set_HomeFlag()
{
    TEcmSxSt_Flags MotFlag; //모션 상태 관련 구조체

    MotFlag.word = ecmSxSt_GetFlags(Device_Info.NetIdx, Axis0, &Error_Num);

    if(MotFlag.bit.OperEnabled) // Servo On/Off 상태 체크
        //Servo On 시 처리
    else
        //Servo Off 시 처리
}

```

---

<h1>NAME</h1> <p>ecmSxSt_IsConnected</p> <p>-</p>	<b>INFORMATION</b>
	 Home      Operation Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_bool ecmSxSt\_IsConnected ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] EEcmConnStsDetail \* ConnStsDetail, [out] t\_i32 \*ErrCode)

### DESCRIPTION

지정한 축의 연결상태를 반환하는 함수입니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ ConnStsDetail: 해당 축의 연결 상태에 대한 상세 정보를 반환 받을 포인터. 이 포인터를 통해서 반환 받은 값의 의미는 다음과 같습니다.

Value	Meaning
-1 (ecmCONN_STS_NOCONFIG)	해당 Slave 가 Network Configuration 에서 정의되어 있지 않음.
0 (ecmCONN_STS_DISCONN)	해당축이 환경설정되어 있으나 통신이 끊어진 상태
1 (ecmCONN_STS_INITIAL)	해당축의 통신이 연결되어 있으며, AL State 가 INITIAL 단계임.
2 (ecmCONN_STS_PREOP)	해당축의 통신이 연결되어 있으며, AL State 가 PreOP 단계임.
4 (ecmCONN_STS_SAFEOP)	해당축의 통신이 연결되어 있으며, AL State 가 SafeOP 단계임.
8 (ecmCONN_STS_OP)	해당축의 통신이 연결되어 있으며, AL State 가 OP 단계임.

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

### RETURN VALUE

\* 해당 축이 연결되어 있는지 여부를 반환합니다.

Value	Meaning
0	해당 축이 연결되어 있지 않음
1	해당 축이 연결되어 있음



## EXAMPLE

---

 C/C++
 

---

```

    // 1) 축의 연결 상태 및 AL-STATE 체크 및 표시 //
    EEcmConnStsDetail ConnStsDetail;
    t_bool IsConnected = ecmSxSt_IsConnected(NETID, Axis, &ConnStsDetail, NULL);

    // IsConnected 는 연결되었는지 안되었는지에 대한 상태를 true/false 로 반환하는데,
    ConnStsDetail 변수를 참조하면 좀더 자세하게 연결상태를 알아볼 수 있다. //
    if(ConnStsDetail == ecmCONN_STS_NOCONFIG){
        GetDlgItem(IDC_edtAlStateMon)->SetWindowText(_T("Invalid Axis"));
        return CDialog::OnTimer(nIDEvent);
    }else if(ConnStsDetail == ecmCONN_STS_DISCONNECT){
        GetDlgItem(IDC_edtAlStateMon)->SetWindowText(_T("Disconnected"));
        return CDialog::OnTimer(nIDEvent);
    }else{
        TCHAR szTemp[128];
        GetAlStateText(ConnStsDetail, szTemp, 50);
        GetDlgItem(IDC_edtAlStateMon)->SetWindowText(szTemp);
    }
  
```

---

NAME	INFORMATION
ecmSxSt_GetInPDOPtr -	<div style="display: flex; justify-content: space-between;"> <span>📁 Home</span> <span>Operation</span> </div> <hr/> <div style="display: flex; justify-content: space-between;"> <span></span> <span>Function</span> </div> <hr/> <div style="display: flex; justify-content: space-between;"> <span>✍️ VC++/BCB/.NET</span> </div> <hr/> <div style="display: flex; justify-content: space-between;"> <span>📖 Level 1</span> </div> <hr/> <div style="display: flex; justify-content: space-between;"> <span>😊 위험 요소 없음</span> </div>

## SYNOPSIS

```
void* ecmSxSt_GetInPDOPtr ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_ui8
*PDODataType, [out] t_i32 *ErrCode)
```

## DESCRIPTION

이 함수는 지정한 축의 Input PDO 영역에 대한 포인터를 반환하는 함수입니다. 사용자는 이 함수가 반환하는 포인터를 직접 액세스하여 해당 축의 모션 상태를 참조할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ PDODataType: ComiECAT에서는 각 축의 PDO 데이터 형식을 4가지로 형식으로 제공합니다. 이 인자는 지정한 축에서 현재 사용하고 있는 PDO 데이터 형식을 나타냅니다. PDO 데이터 형식은 네트워크 환경설정 프로그램에서 해당 슬레이브의 Input PDO 맵핑 설정에 의해서 결정됩니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 이 리턴값은 Input PDO 영역에 대한 포인터를 반환합니다.

## EXAMPLE

NAME	INFORMATION
ecmSxSt_GetDI - 모터드라이버에서 제공하는 디지털입력의 상태를 반환	<div style="display: flex; justify-content: space-between;"> <span>Home</span> <span>Operation</span> </div> <hr/> Function <hr/> <div style="display: flex; align-items: center;"> <span>VC++/BCB/.NET</span> </div> <hr/> <div style="display: flex; align-items: center;"> <span>Level 1</span> </div> <hr/> <div style="display: flex; align-items: center;"> <span>위험 요소 없음</span> </div>

## SYNOPSIS

□ t\_word ecmSxSt\_GetDI ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

### 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.

<h1>NAME</h1> <p><b>ecmSxSt_GetTouchProbeSts</b></p> <p>- 터치프루브 기능의 동작 상태를 반환</p>	<b>INFORMATION</b>	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
 Level 1		
 위험 요소 없음		

## SYNOPSIS

□ t\_byte ecmSxSt\_GetTouchProbeSts ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TouchProbeIndex, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 터치프루브 기능의 동작 상태를 반환합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ TouchProbeIndex: 터치프루브 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 터치프루브의 상태를 나타내는 바이트 값입니다. 이 값의 각 비트는 다음과 같은 의미를 가집니다.

Bit description

bit	value	Note	
0	0	Touch probe 1 is switch off	Touch probe 1 operation stop
	1	Touch probe 1 is enabled	Touch probe 1 is in operation
1	0	Touch probe 1 no positive edge value stored	Rising edge Touch probe 1 is incomplete status
	1	Touch probe 1 positive edge value stored	Rising edge Touch probe 1 is completion status
2	0	Touch probe 1 no negative edge value stored	Falling edge Touch probe 1 is incomplete status
	1	Touch probe 1 negative edge value stored	Falling edge Touch probe 1 is completion status
3-5	-	Reserved	Not used
6-7	-	Not Supported	Not used

## NAME

ecmSxSt\_GetTouchProbePos

- 지정한 터치프루브의 래치된 위치를 반환

### INFORMATION

Home Operation

Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ t\_f64 ecmSxSt\_GetTouchProbePos ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TouchProbeIndex, [out] t\_i32 \*ErrCode)

### DESCRIPTION





이 함수는 지정한 터치프루브의 래치된 위치 값을 반환합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ TouchProbeIndex: 터치프루브 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 터치프루브 센서가 감지되었을 때에 래치(Latch)된 모터의 위치값을 반환합니다.

<h1>NAME</h1> <p><b>ecmSxSt_GetOpModeDisp</b></p> <p>- 현재 설정된 모터의 운전 모드를 반환</p>	<b>INFORMATION</b>	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
	 Level 1	
 위험 요소 없음		

## SYNOPSIS

□ t\_ui8 ecmSxSt\_GetOpModeDisp ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 현재 설정된 모터의 운전 모드를 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 현재 설정된 모터의 운전 모드를 반환합니다. 이 값의 의미는 모터 드라이버 제조사마다 다를 수 있지만 일반적으로는 아래와 같은 값을 가지게 됩니다.

Value	Modes of operation	Abbreviation	Support *1)
-128 - -1	Reserved	-	No
0	No mode change / no mode assigned	-	Yes
1	Profile position mode	pp	Yes
2	Velocity mode	vl	No
3	Profile velocity mode	pv	Yes
4	Torque profile mode	tq	Yes
6	Homing mode	hm	Yes
7	Interpolated position mode	ip	(No)
8	Cyclic synchronous position mode	csp	Yes
9	Cyclic synchronous velocity mode	csv	Yes
10	Cyclic synchronous torque mode	est	Yes
11 - 127	Reserved	-	No

## NAME


ecmSxSt\_GetLmMapIndex


- 지정된 축의 리스트모션 맵 번호 반환


### INFORMATION

 Home      Operation

Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_i32 ecmSxSt\_GetLmMapIndex ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 지정된 축이 관련된 리스트모션 맵 번호를 반환합니다. 만일 지정된 축이 리스트모션에 참여하지 않고 있다면 -1 을 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 지정된 축이 관련된 리스트모션 맵 번호를 반환합니다. 만일 지정된 축이 리스트모션에 참여하지 않고 있다면 -1 을 반환합니다.

## NAME


`ecmSxSt_GetPtmMapIndex`


- 지정한 축의 PT 모션 맵 번호 반환


### INFORMATION

 Home      Operation

Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```
int_t ecmSxSt_GetPtmMapIndex ([in] int_t NetID, [in] int_t Axis, [out] int_t *ErrCode)
```

### DESCRIPTION

이 함수는 지정한 축이 관련된 PT 모션 맵 번호를 반환합니다. 만일 지정한 축이 PT 모션에 참여하지 않고 있다면 -1을 반환합니다.

### 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

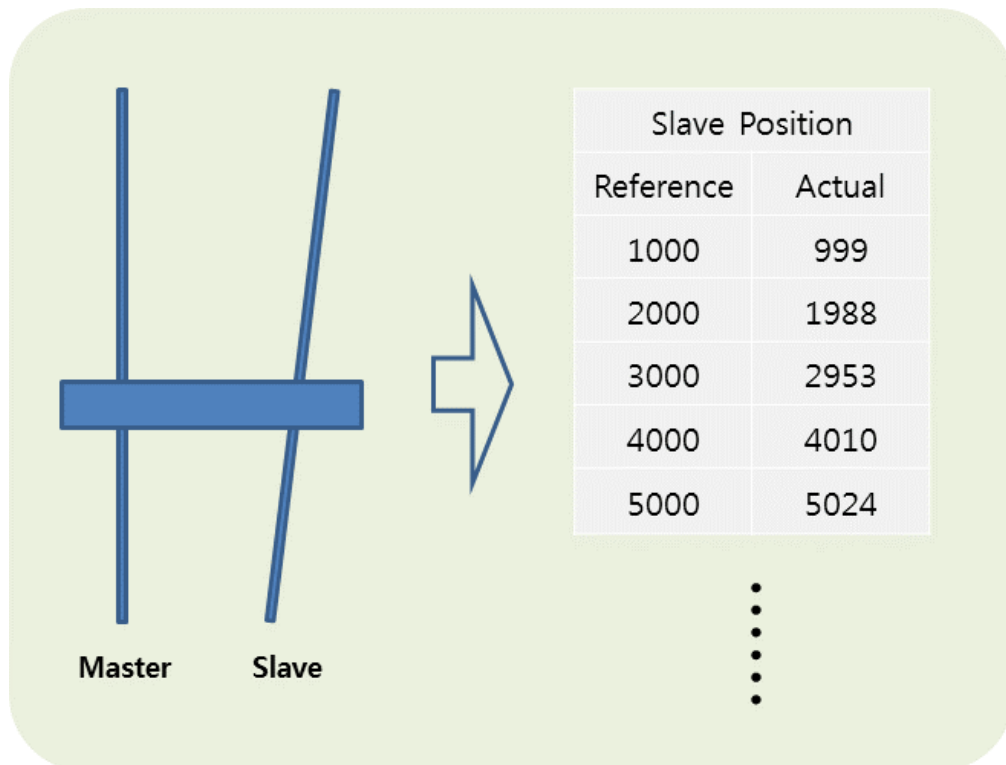
\* 이 리턴값은 지정한 축이 관련된 PT 모션 맵 번호를 반환합니다. 만일 지정한 축이 PT 모션에 참여하지 않고 있다면 -1을 반환합니다.







### 14.2.5 1 차원 위치 보정 (1D Position Correction) 기능 환경 설정

위치 보정 기능은 논리적인 위치 값에 대응되는 실제 모터의 위치에 편차가 존재할 때 이 편차를 보정해주는 기능입니다. 여기서 논리적인 위치값이라 함은 사용자가 지령하는 위치값을 의미합니다. 여러 포인트에 대해서 논리적인 위치값과 그에 대응되는 실제 모터의 위치값을 테이블로 등록한 후에 위치보정 기능을 활성화해주면 마스터 장치는 해당 축을 이송해야할 때에 목표하는 논리위치에 대응되는 실제 모터의 위치값을 테이블에서 찾거나 계산<sup>1</sup> 해서 그 위치로 모터를 이송합니다.

이 기능은 주로 겐트리(Master/Slave 구조) 시스템에서 슬레이브축의 위치를 보정할 때 사용됩니다. 겐트리 시스템에서 마스터 축(Master Axis)과 슬레이브 축(Slave Axis)은 논리적인 위치는 동일한 위치값을 가져야 하지만 여러가지 이유로 인해서 두 모터가 동일한 논리 위치(정렬된 위치)에 있는 경우에 실제 모터의 엔코더값을 읽어보면 편차가 존재하게 됩니다. 따라서 이러한 경우에는 여러 구간으로 나누어서 마스터 축의 위치에 대응되는 슬레이브 축의 위치를 위치보정 테이블로 등록하고 슬레이브 축의 위치 보정 기능을 활성화하면 슬레이브 축은 마스터 축의 위치에 대응되는 보정 위치로 이송됩니다.



<sup>1</sup> 테이블에 등록된 포인트들과 일치하지 않고 그 사이값에 해당하는 위치는 1차 보간 기법을 이용해서 계산

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmSxCfg_PosCorr_SetTableSize</p> <p style="margin: 0;">ecmSxCfg_PosCorr_GetTableSize</p> <p style="margin: 0;">- 위치 보정 테이블 Size 설정/반환</p>	<b>I N F O R M A T I O N</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

- t\_success ecmSxCfg\_PosCorr\_SetTableSize ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableSize, [out] t\_i32 \*ErrCode)
- t\_i32 ecmSxCfg\_PosCorr\_GetTableSize ([in] t\_i32 NetID, [in] t\_i32 Axis , [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_PosCorr\_SetTableSize 함수는 위치 보정 값을 저장하기 위한 테이블 크기를 설정하는 함수입니다.

ecmSxCfg\_PosCorr\_GetTableSize 함수는 현재 설정된 위치 보정 테이블의 크기를 반환하는 함수입니다.

## PARAMETER





- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TableSize : 저장 할 테이블 크기. 여기서 Size 의 단위는 보정 데이터 쌍의 개수를 말합니다. 한 쌍의 보정 데이터는 기준위치데이터와 그에 대응되는 보정위치데이터로 이루어 집니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecmSxCfg\_PosCorr\_SetTableSize 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_PosCorr\_GetTableSize 함수의 반환값: 현재 설정된 보정 테이블의 크기.

NAME	INFORMATION
ecmSxCfg_PosCorr_SetTableData ecmSxCfg_PosCorr_GetTableData - 위치 보정 테이블에 데이터 설정/반환	 Single Axis Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCfg\_PosCorr\_SetTableData ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableIndex, [in] t\_f64 RefCmdPos, [in] t\_f64 ActMotorPos, [out] t\_i32 \*ErrCode)

□ t\_success ecmSxCfg\_PosCorr\_GetTableData ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableIndex, [out] t\_f64 \*RefCmdPos, [out] t\_f64 \*ActMotorPos, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 위치 보정 테이블에 데이터를 설정/반환 하는 함수 입니다.





## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TableIndex : 데이터를 설정/반환 할 테이블 상의 순서
- ▶ RefCmdPos: 기준 위치(논리적인 위치 데이터)
- ▶ ActMotorPos: 해당 기준 위치에 대응되는 실제 모터의 위치
- ▶ 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.

<h2>NAME</h2> <p><b>ecmSxCfg_PosCorr_SetTableFromFile</b> - 위치 보정 테이블에 파일 데이터 값 저장</p>	<b>INFORMATION</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_i32 ecmSxCfg\_PosCorr\_SetTableFromFile ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] char \*szFilePath, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 테이블 데이터 값을 파일에 저장 되어 있는 값을 로드하여 테이블에 저장하는 함수 입니다.

이 함수를 사용하면 사용자는 ecmSxCfg\_PosCorr\_SetTableSize 함수와 ecmSxCfg\_PosCorr\_SetTableData 함수를 따로 사용할 필요가 없습니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TableSize : 파일 경로
- ▶ 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

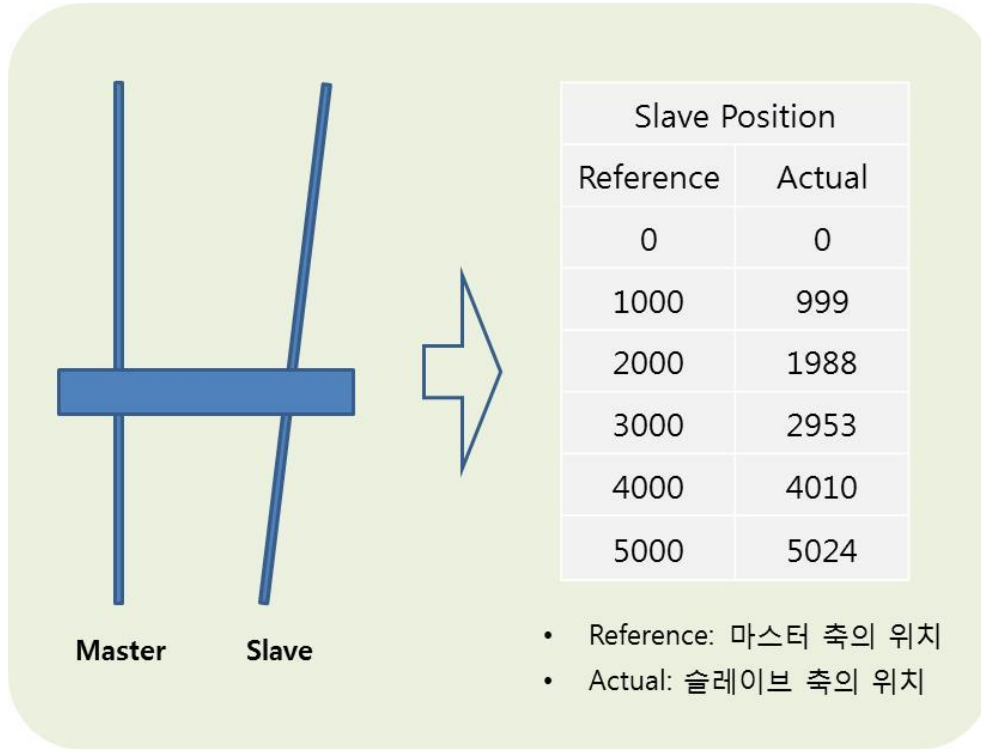
이 함수에 전달되는 파일은 아스키 텍스트 파일이며 그 내용은 아래와 같이 “기준위치, 실제모터데이터” 쌍을 세로로 나열하면 됩니다.

RefPos1, ActMotorPos1

RefPos2, ActMotorPos2

...

예를 들어서 아래 그림과 같이 겐트리(Gantry) 구성되어 있는 구조가 있다고 가정하고, 마스터와 슬레이브가 완전히 정렬이 되는 위치의 관계가 아래와 같다고 가정합니다.



[그림 #] 겐트리 위치 보정의 예시

이런 경우에 보정테이블 파일의 내용은 아래와 같이 구성합니다.

```
0, 0
1000, 999
2000, 1988
3000, 2953
4000, 4010
5000, 5024
```

위와 같이 구성하면 ecmSxCfg\_PosCorr\_SetTableFromFile 함수 안에서는 유효한 위치 데이터쌍의 라인을 카운트하여 테이블 크기를 자동으로 설정하므로 테이블 크기 설정을 따로 할 필요 없습니다.

보충

< 논리위치와 실제모터위치 값 읽기 >

ecmSxSt\_GetPosition() 함수는 다음 표와 같이 TargCntr 매개변수의 값에 따라서 논리위치값을 반환할 수도 있고, 실제모터위치값을 반환할 수도 있습니다.

TargCntr 값	의 미
0 또는 ecmCNT_COMM	논리적인 지령(Command) 위치 값
1 또는 ecmCNT_FEED	논리적인 궤환(Feedback) 위치 값
2 또는 ecmCNT_COMM_CORR	모터에 전달된 실제 지령(Command) 위치 값
3 또는 ecmCNT_FEED_CORR	모터에서 전달받은 실제 궤환(Feedback) 위치 값

EXAMPLE

< 예제 설명 >

위의 REFERENCE 에서 예시한 보정테이블 파일을 “C:\PosCorr.txt” 파일로 저장했다고 가정하고 이를 적용하는 예제입니다.

```

#define NETID 0
#define AXIS 0

// 1차원위치보정테이블을 파일을 통해서 설정 //
ecmSxCfg_PosCorr2D_SetTableFromFile (NETID, AXIS, "c:\PosCorr.txt", NULL);

// 2차원보정 기능 활성화 //
ecmSxCfg_PosCorr2D_SetEnable(NETID, TARGET_AXIS, true, NULL);

// 이후에 AXIS 가 이송(단축, 보간, 마스터-슬레이브 등 어떤 이송이든 모두 적요)될 때는
// 보정테이블이 적용되어서 논리좌표에 대응되는 보정좌표로 모터가 이송되게 됨//
...





ecmSxMot_MoveTo (NETID, AXIS, 5000, NULL);

t_f64 fLogicPos = ecmSxSt_GetPosition (NETID, AXIS, ecmCNT_COMM, NULL);
t_f64 fActualPos = ecmSxSt_GetPosition (NETID, AXIS, ecmCNT_COMM_CORR, NULL);

printf("Logical Commnd Position = %.0f\n Actual Motor Command Position = %.0f\n",
fLogicPos fActualPos);
    
```

< 실행 결과 >

Logical Commnd Position = 5000  
Actual Motor Command Position = 5024

<h2>NAME</h2> <p>ecmSxCfg_PosCorr_SetEnable</p> <p>ecmSxCfg_PosCorr_GetEnable</p> <p>- 위치 보정 테이블 사용 여부 설정/반환</p>	<b>INFORMATION</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCfg\_PosCorr\_SetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)

□ t\_bool ecmSxCfg\_PosCorr\_GetEnable ( [in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_PosCorr\_SetEnable 함수는 지정된 축에 대해서 1 차원 위치보정 기능의 활성화 속성을 설정합니다.

ecmSxCfg\_PosCorr\_GetEnable 함수는 지정된 축에 설정된 1 차원 위치보정 기능의 활성화 속성을 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable : 1 차원 위치보정 기능의 활성화 속성
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecmSxCfg\_PosCorr\_SetEnable 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.





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

\* ecmSxCfg\_PosCorr\_SetEnable 함수의 반환값 : 지정된 축에 설정된 1 차원 위치보정 기능의 활성화 속성을 반환합니다

REFERENCE

□ `ecmSxCfg_PosCorr_SetTableSize` 함수나 `ecmSxCfg_PosCorr_SetTableFromFile` 함수를 실행하면 1 차원위치보정 기능은 자동으로 비활성화됩니다. 따라서 위의 두 함수를 실행하기 전에 1 차원위치보정 기능을 활성화하였더라도 해당 두 함수를 실행하였으면 다시 활성화해주어야 합니다.



NAME	I N F O R M A T I O N
ecmSxCfg_PosCorr_ClearTable - 축에 저장 된 테이블 해제	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCfg\_PosCorr\_ClearTable ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 위치 보정 테이블을 위한 동적 메모리를 해제 해주는 함수 입니다.

## 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

\* ecmSxCfg\_PosCorr2D\_SetTableData 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

## REFERENCE

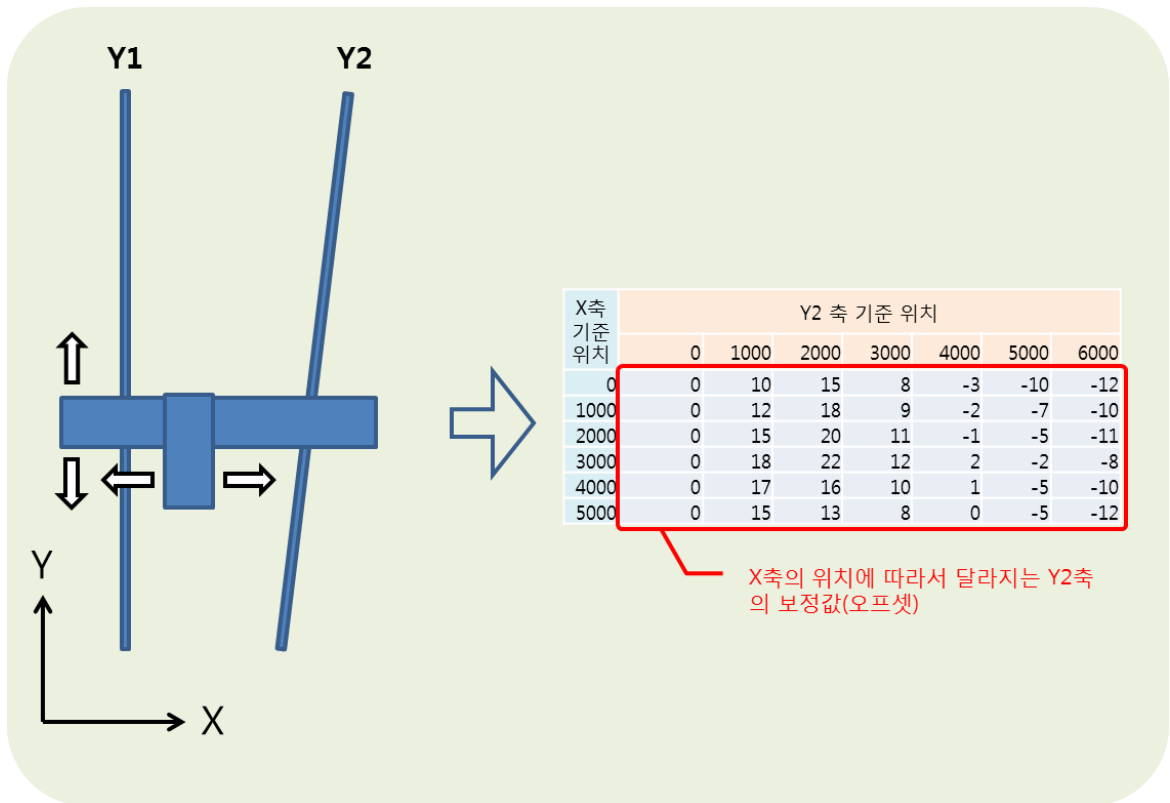
- 1 차원 위치 보정 기능을 사용한 축에 대해서는 마스터 장치 내의 동적 메모리 활용의 효율성을 위해서 1 차원 위치 보정기능을 더 이상 사용하지 않거나 프로그램이 종료되는 시점에 이 함수를 실행해주는 것이 바람직합니다.
- ecmSxCfg\_PosCorr\_SetTableSize 함수와 ecmSxCfg\_PosCorr\_SetTableFromFile 함수 안에서는 ecmSxCfg\_PosCorr\_ClearTable 함수를 자동으로 실행해주므로 위의 두 함수 앞에서는 이 함수를 따로 실행하지 않아도 됩니다.

### 14.2.6 2 차원 위치 보정 (2D Position Correction) 기능 환경 설정

앞서 ‘14.2.1 위치 보정’ 단원에서 설명한 바와 같이 위치 보정 기능은 논리적인 위치 값에 대응되는 실제 모터의 위치에 편차가 존재할 때 이 편차를 보정해주는 기능입니다.

‘1 차원 위치 보정’ 기능에서는 다른 축의 위치와 상관없이 하나의 논리위치에 대응되는 실제 모터 위치가 하나로 정의됩니다.

그런데 만일 하나의 논리위치에 대응되는 실제 모터 위치가 다른 축의 위치에 따라서 달라져야 한다면 ‘1 차원 위치 보정’ 기능으로는 완전하게 보정할 수가 없습니다. 이러한 경우에 사용하는 것이 ‘2 차원 위치 보정’ 기능입니다.



아래와 같은 용어들을 숙지하시면 2 차원 위치 보정 함수들의 설명을 이해하는데 도움이 될 수 있습니다.

- **대상 축(Target Axis):** 위치 보정을 수행할 축을 의미합니다.
- **참조 축(Reference Axis):** 보정대상축의 보정값에 영향을 미치는 축을 의미합니다.
- **논리 위치(좌표):** 사용자 지령 위치를 의미합니다. 겐트리(마스터/슬레이브) 시스템에서는 마스터 축의 위치가 논리 위치에 해당하게 됩니다.
- **실제 모터 위치(좌표):** 보정 대상 축이 논리 위치에 해당하는 좌표에 위치하기 위해서 실제 모터가 위치해야 하는 위치
- **대상 축의 기준 위치(좌표):** 보정 테이블을 정의할 때에는 대상 축의 논리 위치를 여러 구간으로 나누어서 보정 값을 정의하게 되는데, 이때 보정 값이 정의된 각 논리 위치를 말합니다. 이 값은 보정 테이블에서 각 열(Column)의 기준이 되는 위치에 해당합니다.
- **참조 축의 기준 위치(좌표):** 2D 위치 보정 테이블에는 참조 축의 논리 위치를 여러 구간으로 나누어서 참조 축이 각 논리 위치에 있을 때의 보상값 배열을 따로 정의하게 되는데, 이렇게 대상축의 보정테이블 값을


정의한 참조 축의 각 논리 위치를 '참조 축의 기준 위치'라 합니다. 이 값은 보정 테이블에서 각 행(Row)의 기준이 되는 위치에 해당합니다.


**NAME**


ecmSxCfg\_PosCorr2D\_Reset  
- 2 차원 위치보정 초기화

**INFORMATION**

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmSxCfg\_PosCorr2D\_Reset ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 2 차원 위치보정 관련 설정값들을 초기화 하는 함수입니다. 이 함수를 실행하면 ecmSxCfg\_PosCorr2D\_SetTableHeader() 함수나 ecmSxCfg\_PosCorr2D\_SetTableFromFile() 함수에서 생성한 2 차원 보정테이블 메모리가 삭제됩니다. 따라서 만일 사용자 프로그램에서 2 차원 보정 기능을 사용했다면 2 차원 보정 기능을 더 이상 사용하지 않거나 프로그램이 종료되는 시점에서 이 함수를 호출하여 메모리를 초기화해 주는 것이 바람직합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ Axis : 보정 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**


\* 함수 수행의 성공 여부를 반환합니다.


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


## NAME


ecmSxCfg\_PosCorr2D\_SetTableHeader  
 ecmSxCfg\_PosCorr2D\_GetTableHeader  
 - 보정테이블 헤더 정보 설정/읽기

### INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_PosCorr2D\_SetTableHeader ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] TEcmPC2DHeader PC2DHeader, [out] t\_i32 \*ErrCode)
- t\_success ecmSxCfg\_PosCorr2D\_GetTableHeader ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] TEcmPC2DHeader\* pPC2DHeader, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_PosCorr2D\_SetTableHeader() 함수는 보정 테이블의 헤더 정보를 설정합니다. 보정테이블 헤더 정보의 내용에 대해서는 PC2DHeader 매개변수 설명편을 참고하십시오.

ecmSxCfg\_PosCorr2D\_GetTableHeader() 함수는 현재 설정되어 있는 보정 테이블의 헤더 정보를 읽어 들입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 보정 대상 축 번호.
- ▶ PC2DHeader : 보정 테이블 헤더 정보 구조체. 이 구조체는 'ComiEcatSdk\_def.h' 헤더파일에 아래와 같이 정의되어 있습니다.

```
typedef struct{
    struct{
        t_f64 StartPos;
        t_f32 StepSize;
        t_ui32 StepCount;
    }TA;
    struct{
        t_ui32 Axis;
        t_f64 StartPos;
        t_f32 StepSize;
        t_ui32 StepCount;
    }RA;
}TEcmPC2DHeader;
```

멤버 변수명	Meaning
TA	대상축(Target Axis)의 정보를 모아 놓은 구조체 멤버 변수

TA::StartPos	대상축 기준좌표의 시작 위치
TA::StepSize	대상축 기준좌표의 간격. 이 값은 각 열(Column)의 기준 좌표의 간격을 의미합니다. 테이블의 첫 번째 열은 StartPos 에 해당하는 위치의 보정값이고, 두 번째 열은 (StartPos+StepSize)에 해당하는 위치의 보정값이며, 세 번째 열은 (StartPos+2*StepSize)에 해당하는 위치의 보정값이 되는 것입니다.
TA::StepCount	보정값을 정의한 대상축의 논리좌표 갯수. 이 값은 결국 보정테이블의 열(Column)의 갯수를 의미합니다
RA	참조축(Reference Axis)의 정보를 모아 놓은 구조체 멤버 변수
RA::Axis	참조축의 축번호
RA::StartPos	참조축 기준좌표의 시작 위치
RA::StepSize	참조축 기준좌표의 간격. 이 값은 각 행(Row)의 기준 좌표의 간격을 의미합니다.
RA::StepCount	보정값을 정의한 참조축의 논리좌표 갯수. 이 값은 결국 보정테이블의 행(Row)의 갯수를 의미합니다

- ▶ pPC2DHeader : 보정 테이블 헤더 정보 구조체 값을 반환 받을 포인터.
- ▶ 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 실행되면 마스터 장치에서는 보정테이블에 해당하는 동적 메모리가 할당됩니다.


## NAME


ecmSxCfg\_PosCorr2D\_SetTableData  
 ecmSxCfg\_PosCorr2D\_GetTableData  
 - 보정테이블의 데이터를 쓰기/읽기

## INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_PosCorr2D\_SetTableData ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 Row, [in] t\_i32 Col, [in] t\_f32 CorrOfsData, [out] t\_i32 \*ErrCode)
- t\_f32 ecmSxCfg\_PosCorr2D\_SetTableData ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 Row, [in] t\_i32 Col, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_PosCorr2D\_SetTableData () 함수는 보정 테이블의 지정된 행과 열에 해당하는 위치에 보정데이터 값을 설정합니다.

ecmSxCfg\_PosCorr2D\_GetTableHeader() 함수는 보정 테이블에서 지정된 행과 열에 해당하는 위치에 보정데이터 값을 읽어서 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ Row : 보정테이블의 행(Row) 번호
- ▶ Col : 보정테이블의 열(Column) 번호
- ▶ CorrOfsData : 보정테이블 상의 지정한 행, 열에 설정하고자 하는 보정데이터 값. 이 값은 오프셋값으로 설정합니다. 오프셋값은 논리위치와 실제 모터 위치간의 편차값에 해당하는 값을 말합니다. 모터의 실제 위치와 논리위치간의 관계는  

$$\text{모터실제위치} = \text{논리위치} + \text{CorrOfsData}$$
 의 관계가 됩니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecmSxCfg\_PosCorr2D\_SetTableData 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* `ecmSxCfg_PosCorr2D_GetTableHeader` 함수의 반환값 : 보정 테이블 상의 지정한 행, 열에 등록된 보정 데이터 값.





## NAME


**ecmSxCfg\_PosCorr2D\_SetTableFromFile**  
 - 보정테이블 정보를 파일을 통해서 설정

## INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCfg\_PosCorr2D\_SetTableFromFile ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] char \*szFilePath, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 보정 테이블의 헤더 정보 및 보정데이터 값을 파일을 통해서 설정하도록 하는 함수입니다.

이 함수는 내부에서 ecmSxCfg\_PosCorr2D\_SetTableHeader() 함수와 ecmSxCfg\_PosCorr2D\_SetTableData() 함수를 사용하여 헤더 정보와 보정데이터를 모두 설정합니다. 따라서 이 함수를 사용하면 사용자는 ecmSxCfg\_PosCorr2D\_SetTableHeader() 함수와 ecmSxCfg\_PosCorr2D\_SetTableData() 함수를 사용할 필요가 없습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ szFilePath : 보정테이블 정보가 들어 있는 파일의 경로
- ▶ 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 함수에 전달되는 파일은 xml 형식의 텍스트 파일이어야 하며, 보정테이블 헤더 정보와 보정데이터를 모두 포함하고 있어야 합니다. 2 차원보정에 사용되는 xml 파일의 내용은 아래의 예와 같이 작성하면 됩니다.

<PC-2D>

```
<Header>
  <TargetAxis StartPos="0.0" StepSize="100000.0" StepCount="10">0</TargetAxis>
  <ReferAxis StartPos="0.0" StepSize="50000.0" StepCount="3">1</ReferAxis>
</Header>

<Table>
  0, 10, 25, 30, 20, 5, -10, -20, -5, 3
  10, 20, 35, 40, 30, 15, -5, -15, 0, 13
  20, 30, 45, 50, 40, 25, 10, -5, 5, 23
</Table>
```

</PC-2D>

파일의 내용을 설명하기 전에 이해를 돕기 위해서 xml 에서 사용되는 몇 가지 용어에 대해서 먼저 설명하겠습니다.

- 엘리먼트(Element): 데이터 항목 노드를 말하며, <엘리먼트명> ~ </엘리먼트명>으로 표기합니다.
- 엘리먼트내용(Element Content): 해당 엘리먼트의 값을 말하며, <엘리먼트명> 엘리먼트내용</엘리먼트명> 과 같이 <엘리먼트명>과 </엘리먼트명>사이에 위치합니다. 이 값을 표기할 때는 인용부호(“)를 붙이지 않습니다.
- 엘리먼트속성(Element Attribute): 엘리먼트내용값 이외에도 엘리먼트는 속성이라는 항목을 통해서 데이터를 표현할 수가 있습니다. <엘리먼트명 Attribute1="data1" Attribute2="data2">와 같이 <엘리먼트> 태그 안에서 속성명="데이터"의 형식으로 정의합니다. 속성과 속성간에는 공백으로 분리합니다.

위와 같은 XML 에 대한 간단한 용어를 숙지했다는 전제하에 아래에서 2 차원 보정데이터 파일의 내용에 대해서 설명하도록 하겠습니다

1. <PC-2D> ~ </PC-2D>

2 차원보정에 사용되는 XML 파일의 내용은 ‘<PC-2D>’로 시작해서 ‘</PC-2D>’ 끝나야 합니다. 2 차원보정 테이블과 관련된 정보는 모두 이 블록 내에서 정의되어야 합니다.

1-1. <Header> ~ </Header>

보정테이블의 헤더 정보가 들어가는 영역입니다.

1-1-1. <TargetAxis> ~ </TargetAxis>

대상축의 헤더 정보를 설정하는 영역입니다.

이 엘리먼트의 내용(Element Content) 값은 대상축에 해당하는 축 번호를 설정합니다. 이 값은 ecmSxCfg\_PosCorr2D\_SetTableFromFile() 함수의 Axis 매개 변수값과 일치해야 합니다.

이 엘리먼트에서 정의해야 하는 속성(Attribute)은 아래와 같습니다.

속성명	설 명
StartPos	대상축 기준좌표의 시작 위치를 의미합니다.

StepSize	대상축 기준좌표의 간격을 의미합니다. 이 값은 각 열(Column)의 기준 좌표의 간격을 의미합니다. 테이블의 첫 번째 열은 StartPos 에 해당하는 위치의 보정값이고, 두 번째 열은 (StartPos+StepSize)에 해당하는 위치의 보정값이며, 세 번째 열은 (StartPos+2*StepSize)에 해당하는 위치의 보정값이 되는 것입니다.
StepCount	보정값을 정의한 대상축의 논리좌표 개수를 의미합니다. 이 값은 결국 보정테이블의 열(Column)의 갯수를 의미합니다

1-1-2. <ReferAxis> ~ </ReferAxis>

참조축의 헤더 정보를 설정하는 영역입니다.

이 엘리먼트의 내용(Element Content) 값은 참조축에 해당하는 축 번호를 설정합니다.

이 엘리먼트에서 정의해야 하는 속성(Attribute)은 아래와 같습니다.

속성명	설명
StartPos	참조축 기준좌표의 시작 위치를 의미합니다.
StepSize	참조축 기준좌표의 간격을 의미합니다. 이 값은 각 행(Row)의 기준 좌표의 간격을 의미합니다. 테이블의 첫 번째 행은 StartPos 에 해당하는 위치의 보정값이고, 두 번째 행은 (StartPos+StepSize)에 해당하는 위치의 보정값이며, 세 번째 행은 (StartPos+2*StepSize)에 해당하는 위치의 보정값이 되는 것입니다.
StepCount	보정값을 정의한 참조축의 논리좌표 개수를 의미합니다. 이 값은 결국 보정테이블의 행(Row)의 갯수를 의미합니다

1-2. <Table> ~ </Table>


보정테이블 데이터가 정의되는 영역입니다. 보정데이터는 대상축의 기준위치에서의 보정오프셋값으로 정의되며, 각 기준위치에서 모터의 실제 위치와 기준위치의 관계는

$$\text{모터실제위치} = \text{기준위치} + \text{보정오프셋}$$

의 관계가 됩니다.

보정테이블은 2 차원 배열로 정의되며 각 데이터는 쉼표(,)로 분리됩니다. 열의 개수는 TargetAxis의 StepCount 속성에서 정의한 값과 같아야 하며, 행의 개수는 ReferAxis의 StepCount 속성에서 정의한 값과 같아야 합니다.

공백이나 탭문자는 무시됩니다.

	<p>ecmSxCfg_PosCorr2D_SetEnable()를 사용하여 2차원 위치보정 기능을 비활성화 한 상태에서 이 함수를 실행하고 나서 2차원 위치보정 기능을 활성화해주어야 합니다.</p>
---	--

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"
#define NETID 0
#define TARGET_AXIS 0

// 2차원보정 기능 비활성화 //
    
```

---

```
ecmSxCfg_PosCorr2D_SetEnable(NETID, TARGET_AXIS, false, NULL);  
// 2 차원보정테이블을 파일을 통해서 설정 //  
ecmSxCfg_PosCorr2D_SetTableFromFile (NETID, TARGET_AXIS, "c:\\PosCorr2D.xml",  
NULL);  
// 2 차원보정 기능 활성화 //  
ecmSxCfg_PosCorr2D_SetEnable(NETID, TARGET_AXIS, true, NULL);  
  
// 이후에 TARGET_AXIS 가 이송(단축, 보간, 마스터-슬레이브 등 어떤 이송이든 모두  
적용)될 때는 보정테이블이 적용되어서 논리좌표에 대응되는 보정좌표로 모터가 이송되게  
됨//  
...  
  
// 프로그램의 종료 시점이나 2 차원 위치보정 기능이 필요없는 경우에는 Reset 함수  
실행해주는 것이 바람직함 //  
ecmSxCfg_PosCorr2D_Reset(NETID, TARGET_AXIS, NULL);
```

---

**NAME**

ecmSxCfg\_PosCorr2D\_SetEnable

ecmSxCfg\_PosCorr2D\_GetEnable

- 2 차원 위치보정 기능의 활성화 설정/반환

**INFORMATION**

Single Axis Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

**SYNOPSIS**

❑ t\_success ecmSxCfg\_PosCorr2D\_SetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)

❑ t\_bool ecmSxCfg\_PosCorr2D\_GetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

ecmSxCfg\_PosCorr2D\_SetEnable () 함수는 2 차원 위치보정 기능을 활성화 상태를 설정합니다.

ecmSxCfg\_PosCorr2D\_GetEnable () 함수는 2 차원 위치보정 기능의 활성화 상태를 반환합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable : 2 차원 위치보정 기능의 활성화 상태를 설정합니다.

Value	Meaning
0	2 차원 위치보정 기능을 활성화합니다.
1	2 차원 위치보정 기능을 비활성화합니다.

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


**RETURN VALUE**

\* ecmSxCfg\_PosCorr2D\_SetEnable 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_PosCorr2D\_SetEnable 함수의 반환값 : 2 차원 위치보정 기능의 활성화 상태를 반환합니다.

REFERENCE

 <p>주의</p>	<p>아래와 같은 함수들을 호출할 때에는 2 차원 위치보정 기능이 비활성화된 상태이어야 합니다.</p> <p>ecmSxCfg_PosCorr2D_SetTableHeader, ecmSxCfg_PosCorr2D_SetTableData, ecmSxCfg_PosCorr2D_SetTableFromFile</p>
---	--

 <p>보충</p>	<p>&lt; 논리위치와 실제모터위치 값 읽기 &gt;</p> <p>ecmSxSt_GetPosition() 함수는 다음 표와 같이 TargCntn 매개변수의 값에 따라서 논리위치값을 반환할 수도 있고, 실제모터위치값을 반환할 수도 있습니다.</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">TargCntn 값</th> <th style="text-align: left;">의미</th> </tr> </thead> <tbody> <tr> <td>0 또는 ecmCNT_COMM</td> <td>논리적인 지령(Command) 위치 값</td> </tr> <tr> <td>1 또는 ecmCNT_FEED</td> <td>논리적인 궤환(Feedback) 위치 값</td> </tr> <tr> <td>2 또는 ecmCNT_COMM_PRECORR</td> <td>모터에 전달된 실제 지령(Command) 위치 값</td> </tr> <tr> <td>3 또는 ecmCNT_FEED_PRECORR</td> <td>모터에서 전달받은 실제 궤환(Feedback) 위치 값</td> </tr> </tbody> </table>	TargCntn 값	의미	0 또는 ecmCNT_COMM	논리적인 지령(Command) 위치 값	1 또는 ecmCNT_FEED	논리적인 궤환(Feedback) 위치 값	2 또는 ecmCNT_COMM_PRECORR	모터에 전달된 실제 지령(Command) 위치 값	3 또는 ecmCNT_FEED_PRECORR	모터에서 전달받은 실제 궤환(Feedback) 위치 값
TargCntn 값	의미										
0 또는 ecmCNT_COMM	논리적인 지령(Command) 위치 값										
1 또는 ecmCNT_FEED	논리적인 궤환(Feedback) 위치 값										
2 또는 ecmCNT_COMM_PRECORR	모터에 전달된 실제 지령(Command) 위치 값										
3 또는 ecmCNT_FEED_PRECORR	모터에서 전달받은 실제 궤환(Feedback) 위치 값										

EXAMPLE

```

C/C++
#include "ComiEcatSdk_Api.h"
#define NETID 0
#define TARGET_AXIS 0
#define REFER_AXIS 1

// 2 차원보정 기능 비활성화 //
ecmSxCfg_PosCorr2D_SetEnable(NETID, TARGET_AXIS, false, NULL);
// 2 차원보정테이블을 파일을 통해서 설정. 이때 보정테이블에서 참조축의 위치가 0 일때, 대상축
100000 의 보정값이 +30 이라고 가정함. //
ecmSxCfg_PosCorr2D_SetTableFromFile (NETID, TARGET_AXIS, "c:\\PosCorr2D.xml", NULL);
// 2 차원보정 기능 활성화 //
ecmSxCfg_PosCorr2D_SetEnable(NETID, TARGET_AXIS, true, NULL);

ecmSxMot_MoveTo (NETID, REFER_AXIS, 0, NULL);
ecmSxMot_MoveTo (NETID, TARGET_AXIS, 100000, NULL);

t_f64 fComPos_Logic = ecmSxSt_GetPosition (NETID, TARGET_AXIS, ecmCNT_COMM, NULL);
t_f64 fComPos_Motor = ecmSxSt_GetPosition (NETID, TARGET_AXIS,
ecmCNT_COMM_PRECORR, NULL);

printf("Logical Commnd Position = %.0f\n Actual Motor Command Position = %.0f\n", fComPos_Logic,
fComPos_Motor);
    
```

위의 결과로 화면에 표시되는 것은

```

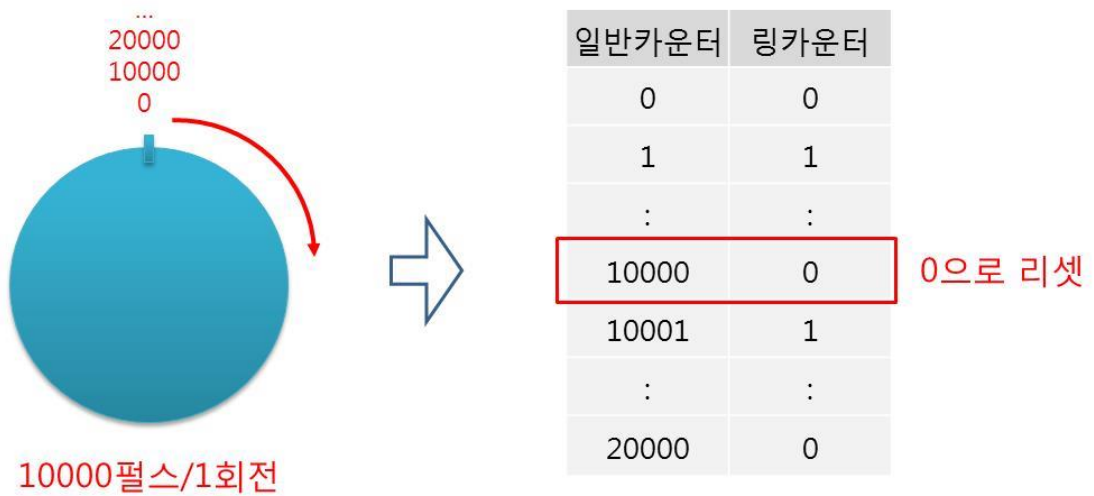
Logical Commnd Position = 100000
Actual Motor Command Position = 100030
    
```

으로 표시됩니다.

### 14.2.7 링카운터(Ring Counter) 기능 환경 설정

링카운터(Ring Counter) 기능은 위치카운트의 범위를 설정하고 위치 값이 해당 범위에 다다르면 다시 0으로 리셋되어서 0부터 다시 카운트되는 기능입니다.

예를 들어서 아래와 같이 1 회전에 10000 펄스에 해당하는 기구물이 있다고 가정합니다. 일반적인 카운터 모드에서는 이 기구물의 0 위치가 한 바퀴돌면 10000 이 되고 두 바퀴 돌면 20000 이 됩니다. 그런데 이때 링카운터 범위를 10000 으로 설정하여 링카운터 모드를 활성화하면 몇 바퀴를 돌아도 0 의 위치는 0 이 됩니다. 마찬가지로 몇 바퀴를 돌아도 같은 각도에 해당하는 위치에서는 같은 위치 값으로 표현되게 됩니다.



[그림 #] 링카운터 예시



## NAME

ecmSxCfg\_Ring\_SetEnable  
 ecmSxCfg\_Ring\_GetEnable  
 - 링 카운터 사용 여부 설정/반환

## INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_Ring\_SetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable [out] t\_i32 \*ErrCode)
- t\_bool ecmSxCfg\_Ring\_GetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 링 카운터 사용 여부를 설정/반환 하는 함수 입니다

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable : 링 카운터 사용 여부를 결정하는 파라미터

Value	Meaning	Default
0	일반 카운트 모드로 설정됩니다.	Y
1	링카운트 모드로 설정됩니다.	

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

## RETURN VALUE

- \* ecmSxCfg\_Ring\_SetEnable : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다

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

- \* ecmSxCfg\_Ring\_GetEnable. : 이 리턴 값은 링 카운터의 사용 여부가 반환 됩니다

**NAME**

ecmSxCfg\_Ring\_SetPosRange  
 ecmSxCfg\_Ring\_GetPosRange  
 - 링 카운터 범위 설정/반환

**INFORMATION**

Single Axis Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

**SYNOPSIS**

- t\_success ecmSxCfg\_Ring\_SetPosRange ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 RingPosRange , [out] t\_i32 \*ErrCode)
- t\_f64 ecmSxCfg\_Ring\_GetPosRange ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 링 카운터의 범위를 설정/반환 하는 함수 입니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ RingPosRange : 링 카운터의 범위를 설정 합니다. 이 값의 단위는 Unit Distance 가 적용된 논리적 거리 단위로 설정합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

**RETURN VALUE**

\* ecmSxCfg\_Ring\_SetPosRange 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_Ring\_GetPosRange 함수의 반환값 : 현재 설정되어 있는 링 카운터의 범위를 반환 합니다.

**REFERENCE**

RingPosRange 값과 위치 표현의 관계

링카운터 모드로 설정되면 위치의 표현은 0 ~ Prmax 로 표현됩니다. 이때 Prmax 값은 (RingPosRange - 1 펄스)에 해당하는 값입니다.

예를 들어서 1 회전에 10000 펄스, UnitDistance=1, RingPosRange=10000 으로 설정하였다면 위치는 0 ~ 9999 로 표현됩니다. 일반 카운터일 때 10000 으로 표현되던 위치는 0 으로 표현됩니다.

만일 1 회전에 10000 펄스, UnitDistance=200 이라면 RingPosRange 는  $50(10000/200=50)$ 으로 설정하여야 합니다. 이 조건에서 Prmax =  $50 - (1/200) = 49.995$  이므로 표현되는 위치의 범위는 0 ~ 49.995 가 됩니다.

만일 1 회전에 10000 펄스인 모터에서 위치의 표현을 360 도를 한 주기로 하는 각도(10000 -> 360)로 표현하고 싶다면 UnitDistance= $10000/360=27.7778$  로 설정합니다. 그리고 RingPosRange 는 360 으로 설정합니다. 그러면 이 조건에서는 Prmax =  $360 - (1/27.7778) = 359.964$  가 되므로 표현되는 실제 위치는 0 ~ 359.964 가 됩니다.

## NAME

`ecmSxCfg_Ring_SetDirMode`  
`ecmSxCfg_Ring_GetDirMode`  
 - 링 카운터 모드 설정/반환

## INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ `t_success ecmSxCfg_Ring_SetDirMode ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 RingDirMode, [out] t_i32 *ErrCode)`

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

## DESCRIPTION

`ecmSxCfg_Ring_SetDirMode` 함수는 링카운트모드에서 절대좌표이송의 이송방향모드를 설정합니다.

`ecmSxCfg_Ring_GetDirMode` 함수는 링카운트모드에서 현재 설정된 절대좌표이송의 이송방향모드를 반환합니다.

### 링카운트 모드에서 절대좌표이송의 이송방향모드

링카운트 모드에서는전체 영역이 항상 0 ~ Prmax(링카운트 최대 위치값)의 범위에서 위치가 표현되기 때문에 특정 위치로 이송할 때에 (+)방향으로 이송해서 갈 수도 있고 (-)방향으로 이송해서 갈 수도 있습니다.

예를 들어서 링카운트 범위가 10000(0 ~ 9999) 펄스로 설정되었다고 가정합니다. 이 시스템에서 현재의 위치가 0 이라 할 때 3000 의 위치로 이동하려면 (+)방향으로 3000 만큼 이송할 수도 있고, (-)방향으로 7000 만큼 이송하여도 3000 의 위치로 이동할 수 있습니다.

상대좌표이송은 이송명령에 어느 방향으로 이송할 지가 명시되지만 절대좌표이송에서는 그 방향이 명시적으로 지정되지 않습니다. `ecmSxCfg_Ring_SetDirMode` 함수는 링카운트 모드를 사용할 때에 절대좌표모드에서도 이송방향모드를 명시할 수 있도록 해주는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ RingDirMode : 절대좌표이송 시에 적용되는 이송 방향 모드(EEcmRingCntrDir 공용체 참조).

Value	Meaning	Default
0 (ecmRING_DIR_NEG)	(-)방향으로 이송하여 지정한 절대 위치로 이송	

1 (ecmRING_DIR_POS)	(+)방향으로 이송하여 지정한 절대 위치로 이송	
2 (ecmRING_DIR_NEAR)	(-)방향과 (+)방향 중에서 현재 위치로부터 지정한 절대 위치까지 거리가 더 가까운 방향으로 이송	Y
3 (ecmRING_DIR_FAR)	(-)방향과 (+)방향 중에서 현재 위치로부터 지정한 절대 위치까지 거리가 더 먼 방향으로 이송	

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

## RETURN VALUE

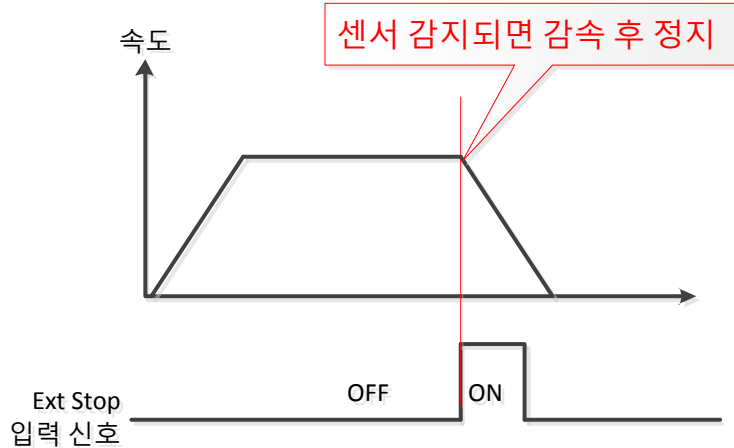
\* ecmSxCfg\_Ring\_SetDirMode 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_Ring\_GetDirMode 함수의 반환값 : 현재 설정된 절대좌표이송의 이송방향모드를 반환합니다

### 14.2.8 외부정지(External Stop) 기능 환경 설정

앞서 ‘14.2.1 위치 보정’ 단원에서 설명한 바와 같이 위치 보정 기능은 논리적인 위치 값에 대응되는 실제 모터의 위치에 편차가 존재할 때 이 편차를 보정해주는 기능입니다.



#### 상대좌표/절대좌표 이송모드에서의 외부정지 기능

외부정지 기능이 활성화된 상태에서 목표좌표가 정해져 있는 상대좌표 이송이나 절대좌표 이송을 수행하였을 때는 어떻게 될까요? 그것은 목표좌표에 도달하기 전에 센서가 감지되느냐 못하느냐에 따라서 동작이 달라집니다.

목표좌표에 도달할 때까지 센서가 감지되지 않으면 목표좌표에서 정지됩니다.

목표좌표에 도달하기 전에 센서가 감지되면 센서가 감지된 시점에 감속 후 정지합니다.

#### 외부정지 기능의 3 가지 환경설정 모드





외부정지 기능은 3 가지 모드로 동작 환경을 설정합니다.

모드번호	입력신호	오프셋 모드
1	범용 디지털입력 접점	시간 오프셋
2	범용 디지털입력 접점	거리 오프셋
3	서보드라이버 터치프루브 입력	거리 오프셋

입력 신호에서 터치프루브(Touch Probe) 입력은 서보드라이버에서 제공하는 입력 단자입니다.

오프셋 모드에서 ‘시간 오프셋’은 센서가 감지된 후에 지정된 시간 오프셋을 적용하여 감속을 시작하는 방식입니다.

오프셋 모드에서 ‘거리 오프셋’은 센서가 감지된 후에 이송되는 거리를 제한하는 방식입니다.

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

## SYNOPSIS

- ❑ t\_success ecmSxCfg\_ExtStop\_SetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)
- ❑ t\_bool ecmSxCfg\_ExtStop\_GetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_ExtStop\_SetEnable 함수는 외부정지 기능의 활성화 속성을 설정합니다.

ecmSxCfg\_ExtStop\_GetEnable 함수는 현재 설정된 외부정지 기능의 활성화 속성을 반환합니다.

## PARAMETER





- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable: 활성화 여부 (0: 활성화 1: 비활성화)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecmSxCfg\_ExtStop\_SetEnable 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_ExtStop\_SetEnable 함수의 반환값 : 현재 설정된 외부정지 기능의 활성화 속성을 반환합니다.

<h2>NAME</h2> <p>ecmSxCfg_ExtStop_SetEnv</p> <p>ecmSxCfg_ExtStop_GetEnv</p> <p>-</p>	<b>INFORMATION</b>
	<ul style="list-style-type: none"> <li> Single Axis Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>

## SYNOPSIS

- t\_success ecmSxCfg\_ExtStop\_SetEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] TEcLogicAddr SigAddr, [in] t\_i32 ActiveState, [in] t\_i32 DelayTime, [out] t\_i32 \*ErrCode)
- t\_i32 ecmSxCfg\_ExtStop\_GetEnv([in] t\_i32 NetID, [in] t\_i32 Axis, [out] TEcLogicAddr \*pSigAddr, [out] t\_i32 \*ActiveState, [out] t\_i32 \*DelayTime, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_ExtStop\_SetEnv 함수는 외부정지 기능의 환경을 1 번 모드 조건으로 설정합니다. 1 번 모드는 ExtStop 센서 입력은 범용디지털 입력을 통해서 감시하고, 오프셋 방식은 시간 오프셋을 적용하는 모드입니다.

ecmSxCfg\_ExtStop\_GetEnv 함수는 현재 설정된 외부정지 기능의 환경설정 상태를 읽습니다. 단, 이 함수는 직전에 ecmSxCfg\_ExtStop\_SetEnv 함수를 통해서 외부정지 기능의 환경이 설정되었을 때에만 사용할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ SigAddr: 외부정지입력으로 사용할 입력 채널에 해당하는 Input PDO 영역의 주소를 지정하는 인자입니다. 이 인자는 TEcLogicAddr 형식의 구조체로서 아래와 같은 멤버들로 구성됩니다. 이 인자의 값을 얻는 방법은 REFERENCE 문단을 참고하십시오.

Variable Name	Meaning
AddrOfs	네트워크의 Input PDO 영역에서 해당 접점 데이터가 위치하는 바이트 오프셋 주소
BitIdx	AddrOfs 이 가리키는 바이트에서 접점 데이터가 위치하는 비트의 번호

- ▶ ActiveState: 외부정지 기능이 동작하는 입력 신호의 조건을 설정합니다.

Value	Meaning
0	Falling Edge(ON -> OFF)가 발생하면 동작합니다. 만일 이송이 시작될 때 이미 OFF 이면 ON 이 되었다가 다시 OFF 될 때 동작합니다.



1	Rising Edge(OFF -> ON)가 발생하면 동작합니다. 만일 이송이 시작될 때 이미 ON 이면 OFF 되었다가 다시 ON 될 때 동작합니다.
2	신호가 OFF 상태가 되면 동작합니다. 만일 이송이 시작될 때 신호가 이미 OFF 이면 외부정지동작이 바로 작동합니다.
3	신호가 ON 상태가 되면 동작합니다. 만일 이송이 시작될 때 신호가 이미 ON 이면 외부정지동작이 바로 작동합니다.

▶ DelayTime: 센서가 감지된 이후에 실제 감속이 시작되는 지연 시간을 msec 단위로 설정합니다.

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

## RETURN VALUE

\* ecmSxCfg\_ExtStop\_SetEnv 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_ExtStop\_GetEnv 함수의 반환값 : 마지막에 설정된 외부정지 기능의 환경설정 모드를 반환합니다.

## REFERENCE

### □ <전역채널번호를 이용한 SigAddr 설정 방법>

입력점점의 전역 채널 번호를 알고 있을 때에는 ecdiGetLogicAddr 함수를 이용해서 SigAddr 값을 설정할 수 있습니다. ecdiGetLogicAddr 함수의 원형은 다음과 같습니다.

```
t_dword ecdiGetLogicAddr (t_i32 NetID, t_ui32 DiChannel, t_i32 *ErrCode)
```

예를 들어서 전역 채널 번호 10 번의 디지털입력을 사용해서 외부정지 기능을 적용한다면 아래와 같이 SigAddr 를 설정할 수 있습니다.

```
TEcLogicAddr SigAddr;
t_dword dwSigAddr = ecdiGetLogicAddr (NETID, 10, NULL);
memcpy(&SigAddr, &dwSigAddr, sizeof(t_dword));
ecmSxCfg_ExtStop_SetEnv (NETID, AXIS, SigAddr, 1, 0, NULL);
```

### □ <로컬채널번호를 이용한 SigAddr 설정 방법>

입력점점의 로컬 채널 번호를 알고 있을 때에는 ecdiLogBitAddr\_FromLocalChannel 함수를 이용해서 SigAddr 값을 설정할 수 있습니다. ecdiLogBitAddr\_FromLocalChannel 함수의 원형은 다음과 같습니다.

```
t_dword ecdiLogBitAddr_FromLocalChannel (t_i32 NetID, t_ui16 SlvPhysAddr, t_i32 LocalChannel, t_i32 *ErrCode)
```





예를 들어서 슬레이브 주소가 0x201 인 슬레이브의 로컬채널 10 번의 입력을 사용해서 외부정지 기능을 적용한다면 아래와 같이 SigAddr 를 설정할 수 있습니다.

## CHAPTER 14:: MOTION MULTIPLE AXES FUNCTIONS

```
TEcLogicAddr SigAddr;  
t_dword dwSigAddr = ecdiLogBitAddr_FromLocalChannel (NETID, 0x201, 10, NULL);  
SigAddr.AddrOfs = (dwSigAddr & 0x3fff); // dwSigAddr의 하위 14비트가 Offset Address임  
SigAddr.BitIdx = (dwSigAddr >> 16) & 0xffff; // dwSigAddr의 상위 16비트가 BitIdx임  
ecmSxCfg_ExtStop_SetEnv (NETID, AXIS, SigAddr, 1, 0, NULL);
```

### EXAMPLE

```
#define AXIS      0  
#define SLV_ADDR 0x201 // Slave Address  
#define LOC_CH    0 // ExtStop 신호 입력의 로컬 채널 번호  
TEcLogicAddr SigAddr;  
t_dword dwSigAddr = ecdiLogBitAddr_FromLocalChannel (NETID, SLV_ADDR, LOC_CH, NULL);  
SigAddr.AddrOfs = (dwSigAddr & 0x3fff); // dwSigAddr의 하위 14비트가 Offset Address임  
SigAddr.BitIdx = (dwSigAddr >> 16) & 0xffff; // dwSigAddr의 상위 16비트가 BitIdx임  
ecmSxCfg_ExtStop_SetEnv (NETID, AXIS, SigAddr, 1, 0, NULL);  
  
ecmSxMot_Move(NETID, AXIS, 1000000, NULL); // 1000000 거리만큼 이송 진행. 이송 중에 ExtStop  
신호 감지되면 그 시점에 감속후 정지
```

<h2>NAME</h2> <p>ecmSxCfg_ExtStop_SetEnv2</p> <p>ecmSxCfg_ExtStop_GetEnv2</p> <p>-</p>	<b>INFORMATION</b>
	 Single Axis Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

- t\_success ecmSxCfg\_ExtStop\_SetEnv2([in] t\_i32 NetID, [in] t\_i32 Axis, [in] TEcLogicAddr SigAddr, [in] t\_i32 ActiveLogic, [in] t\_f64 OfsDist, [out] t\_i32 \*ErrCode)
- t\_i32 ecmSxCfg\_ExtStop\_GetEnv2([in] t\_i32 NetID, [in] t\_i32 Axis, [out] TEcLogicAddr \*SigAddr, [out] t\_i32 \*ActiveLogic, [out] t\_f64 \*OfsDist, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_ExtStop\_SetEnv2 함수는 외부정지 기능의 환경을 2 번 모드 조건으로 설정합니다. 2 번 모드는 ExtStop 센서 입력은 범용디지털 입력을 통해서 감시하고, 오프셋 방식은 거리 오프셋을 적용하는 모드입니다.

ecmSxCfg\_ExtStop\_GetEnv2 함수는 현재 설정된 외부정지 기능의 환경설정 상태를 읽습니다. 단, 이 함수는 직전에 ecmSxCfg\_ExtStop\_SetEnv2 함수를 통해서 외부정지 기능의 환경이 설정되었을 때에만 사용할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ SigAddr: 외부정지입력으로 사용할 입력 채널에 해당하는 Input PDO 영역의 주소를 지정하는 인자입니다. 이 인자는 TEcLogicAddr 형식의 구조체로서 아래와 같은 멤버들로 구성됩니다. 이 인자의 값을 얻는 방법은 ecmSxCfg\_ExtStop\_SetEnv 함수 설명편의 REFERENCE 문단을 참고하십시오.

Variable Name	Meaning
AddrOfs	네트워크의 Input PDO 영역에서 해당 접점 데이터가 위치하는 바이트 오프셋 주소
BitIdx	AddrOfs 이 가리키는 바이트에서 접점 데이터가 위치하는 비트의 번호

- ▶ ActiveLogic: 외부정지 기능이 동작하는 입력 신호의 조건을 설정합니다.

Value	Meaning
0	Falling Edge(ON -> OFF)가 발생하면 동작합니다. 만일 이송이 시작될 때 이미 OFF 이면 ON 이 되었다가 다시 OFF 될 때 동작합니다.

1	Rising Edge(OFF -> ON)가 발생하면 동작합니다. 만일 이송이 시작될 때 이미 ON 이면 OFF 되었다가 다시 ON 될 때 동작합니다.
2	신호가 OFF 상태가 되면 동작합니다. 만일 이송이 시작될 때 신호가 이미 OFF 이면 외부정지동작이 바로 작동합니다.
3	신호가 ON 상태가 되면 동작합니다. 만일 이송이 시작될 때 신호가 이미 ON 이면 외부정지동작이 바로 작동합니다.

▶ OfsDist: 센서가 감지된 이후에 정지가 완료되는 위치까지의 거리.

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

RETURN VALUE

\* ecmSxCfg\_ExtStop\_SetEnv2 (): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

\* ecmSxCfg\_ExtStop\_GetEnv2 ():

REFERENCE

□ ecmSxCfg\_ExtStop\_SetEnv 함수의 DelayTime 인자는 센서가 감지된 이후에 실제 감속을 시작하는 시점까지의 지연시간을 정의하는 것에 반해서, ecmSxCfg\_ExtStop\_SetEnv2 함수의 OfsDist 인자는 센서가 감지되고 그 지점으로부터 정지되는 시점까지의 거리를 정의하게 된다.

□ PTP 이송(이송 거리가 정해진 이송)에서 OfsDist 가 적용되는 방식은 ‘Offset Distance Mode’에 따라서 영향을 받을 수 있습니다. ‘Offset Distance Mode’ 속성은 ecmSxCfg\_ExtStop\_SetOfsDistMode 함수를 통해서 설정됩니다.

EXAMPLE

<b>NAME</b>	<b>INFORMATION</b>
ecmSxCfg_ExtStop_SetEnv3	Single Axis Function
ecmSxCfg_ExtStop_GetEnv3	VC++/BCB/.NET
-	Level 1
	위험 요소 없음

## SYNOPSIS


- ▣ t\_success ecmSxCfg\_ExtStop\_SetEnv3 ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui8 TouchProbeIndex, [in] t\_i32 ActiveLogic, [in] t\_f64 OfsDist, [out] t\_i32 \*ErrCode)
- ▣ t\_i32 ecmSxCfg\_ExtStop\_GetEnv3 ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_ui8 \*TouchProbeIndex, [out] t\_i32 \*ActiveLogic, [out] t\_f64 \*OfsDist, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_ExtStop\_SetEnv3 함수는 외부정지 기능의 환경을 2 번 모드 조건으로 설정합니다. 2 번 모드는 ExtStop 센서 입력은 터치프루브(Touch Probe) 입력을 통해서 감시하고, 오프셋 방식은 거리 오프셋을 적용하는 모드입니다.

ecmSxCfg\_ExtStop\_GetEnv3 함수는 현재 설정된 외부정지 기능의 환경설정 상태를 읽습니다. 단, 이 함수는 직전에 ecmSxCfg\_ExtStop\_SetEnv3 함수를 통해서 외부정지 기능의 환경이 설정되었을 때에만 사용할 수 있습니다.

터치프루브는 서보드라이버에서 직접 신호 입력을 받아서 센서가 감지되는 시점의 위치를 저장하여 그 위치를 마스터 제어가 알 수 있도록 해주는 기능입니다. 터치프루브를 사용하면 범용입력을 사용했을 때보다 상대적으로 보다 정밀한 센서 감지 위치를 적용할 수 있습니다.

	<p>터치프루브를 ExtStop 입력으로 사용하려면 터치프루브 상태 오브젝트(60B9h)와 Touch Probe pos# value 오브젝트(60BAh / 60BB / 60BCh / 60BDh)<sup>2</sup>가 Input PDO 에 포함되어 있어야 합니다.</p>
---	---

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TouchProbeIndex: ExtStop 센서 입력으로 사용되는 터치프루브 번호

<sup>2</sup> 사용하는 터치프루브 번호와 Active Logic 에 따라서 Input PD 에 포함되어야 하는 오브젝트가 달라짐

- ▶ ActiveLogic: 외부정지 기능이 동작하는 입력 신호의 조건을 설정합니다.

Value	Meaning
0	Falling Edge(ON -> OFF)가 발생하면 동작합니다. 만일 이송이 시작될 때 이미 OFF 이면 ON 이 되었다가 다시 OFF 될 때 동작합니다.
1	Rising Edge(OFF -> ON)가 발생하면 동작합니다. 만일 이송이 시작될 때 이미 ON 이면 OFF 되었다가 다시 ON 될 때 동작합니다.
2	신호가 OFF 상태가 되면 동작합니다. 만일 이송이 시작될 때 신호가 이미 OFF 이면 외부정지동작이 바로 작동합니다.
3	신호가 ON 상태가 되면 동작합니다. 만일 이송이 시작될 때 신호가 이미 ON 이면 외부정지동작이 바로 작동합니다.

- ▶ OfsDist: 센서가 감지된 이후에 정지가 완료되는 위치까지의 거리.

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

## REFERENCE





□ 터치프루브를 이용해서 ExtStop 기능을 적용하려면 먼저 ecmSxCfg\_SetTouchProbeFunc 함수를 사용해서 해당 터치프루브의 환경을 적절히 설정해야 합니다.

## RETURN VALUE

\* ecmSxCfg\_ExtStop\_SetEnv3 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_ExtStop\_GetEnv3 함수의 반환값: 마지막에 설정된 외부정지 기능의 환경설정 모드를 반환합니다.

NAME	INFORMATION
ecmSxCfg_ExtStop_SetOfsDistMode	 Single Axis Function
ecmSxCfg_ExtStop_GetOfsDistMode	 VC++/BCB/.NET
-	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCfg\_ExtStop\_SetOfsDistMode ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsHardOfsDistMode, [out] t\_i32 \*ErrCode)

□ t\_bool ecmSxCfg\_ExtStop\_GetOfsDistMode ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_ExtStop\_SetOfsDistMode 함수는 ecmSxCfg\_ExtStop\_SetEnv2 함수와 ecmSxCfg\_ExtStop\_SetEnv3 함수의 OfsDist 인자(거리오프셋)의 적용모드를 설정합니다.

ecmSxCfg\_ExtStop\_GetOfsDistMode 함수는 현재 설정된 거리오프셋의 적용모드를 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsHardOfsDistMode: PTP 이송(이송거리가 정해진 이송) 모드에서 거리오프셋이 적용되는 모드를 설정합니다.

Value	Meaning	Default
0	연성(Sfot) 모드: 센서가 감지된 이후에 미리 정해진 거리오프셋(OfsDist)보다 목표좌표에 먼저 도달하면 해당 목표좌표에서 이송을 종료합니다.	
1	강성(Hard) 모드: 센서가 감지된 이후에 미리 정해진 거리오프셋(OfsDist)보다 목표좌표에 먼저 도달하더라도 해당 목표좌표는 무시되고, 정해진 거리오프셋만큼을 무조건 이송합니다.	Y

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

## RETURN VALUE

\* ecmSxCfg\_ExtStop\_SetOfsDistMode 함수의 반환값: 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxCfg\_ExtStop\_GetOfsDistMode 함수의 반환값: 현재 설정된 OfsDistMode 를 반환합니다.

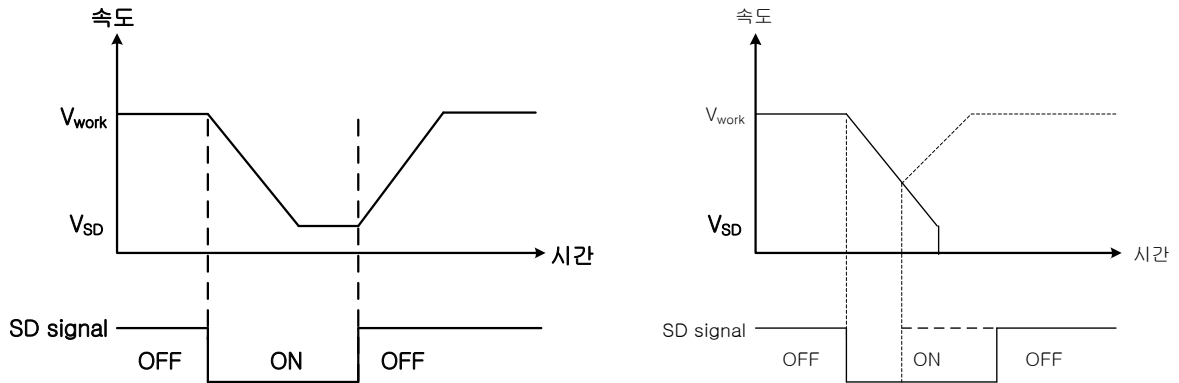
Value	Meaning
0	연성(Sfot) 모드
1	강성(Hard) 모드



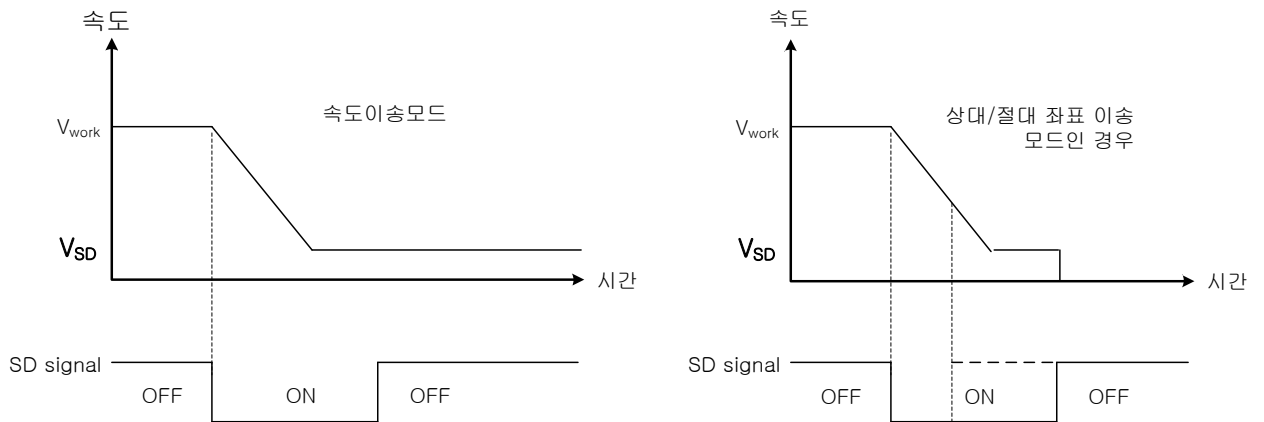
### 14.2.9 SD(Slow Down) 기능

SD (Slow Down)기능이 활성화된 상태에서 단축 이송을 수행하는 중에 SD 입력 신호가 ON 되면 저속으로 감속하여 운전하는 기능을 말합니다.

감속된 이송 속도는 래치모드 설정에 따라서 SD 신호가 OFF 되면 다시 원래의 속도로 복구될 수 있습니다.



[그림 #] 래치모드=1 인 SD 운전 모드



[그림 #] 래치모드=0 인 SD 운전 모드

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmSxSD_SetInputEnv</p> <p style="margin: 0;">ecmSxSD_GetInputEnv</p> <p style="margin: 0;">- SD 입력 신호에 대한 설정 및 반환</p>	<b>INFORMATION</b>
	Home      Operation Function
	VC++/BCB/.NET
	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_success ecmSxSD\_SetInputEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_dword LogBitAddr, [in] t\_bool IsInvertLogic, [in] t\_i32 FilterCount, [out] t\_i32 \*ErrCode)
- t\_success ecmSxSD\_GetInputEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_dword \*LogBitAddr, [out] t\_bool \*IsInvertLogic, [out] t\_i32 \*FilterCount, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxSD\_SetInputEnv 함수는 SD 입력 신호의 환경을 설정하는 함수입니다.

ecmSxSD\_GetInputEnv 함수는 현재 설정된 SD 입력 신호의 환경을 읽는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ LogBitAddr: 논리비트주소를 나타내는 인자입니다. 논리비트주소에 대한 자세한 내용은 “11.2.3 논리비트주소 관리 함수군” 단원을 참고하십시오.
- ▶ IsInvertLogic: 신호의 입력 로직을 반대로 적용할 것인지를 나타내는 인자입니다.

Value	Meaning
0	입력 값이 0 이면 SD 기능이 활성화 됩니다.
1	입력 값이 1 이면 SD 기능이 활성화 됩니다.

▶ FilterCount: SD 입력 신호에 적용할 필터의 주기를 설정하는 인자입니다. 이 값의 단위는 통신 사이클의 카운트 단위입니다. 입력 신호의 ACTIVE 상태가 지정한 사이클 카운트 이상의 시간 동안 유지되지 않으면 해당 입력신호는 무효 처리됩니다.

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

## EXAMPLE

---

C/C++

---

```

#define NETID      0
#define AXIS       0

void OnInitialSetup_SD ()
{
    #define SD_CH    0 // D/I CH0 을 HEMG 신호로 사용하는 것으로 가정함
    #define FILTER_COUNT  1 // 필터카운트값을 1 로 설정 => 2 회의 통신 사이클 동안
    입력상태를 유지하지 않으면 노이즈로 처리
    t_i32 errCode;
    t_dword dwLogBitAddr;

    // HEMG 로 사용할 입력채널 설정 //
    dwLogBitAddr = ecdiLogBitAddr_FromGlobalChannel (NETID, SD_CH, &errCode); //
    전역채널 -> 논리비트주소로 변환
    ecmSxSD_SetInputEnv (NETID, AXIS, dwLogBitAddr, false, FILTER_COUNT,
    &errCode); // SD 입력 채널 등록
    ecmSxSD_SetLatchMode (NETID, AXIS, false, &errCode); // SD 래치 모드 설정
}


void OnStartMove_SD()
{
    #define NORM_SPEED      500000
    #define SD_SPEED        30000
    t_i32 errCode;

    ecmSxSD_SetSpeed (NETID, AXIS, SD_SPEED, &errCode); // SD 속도 설정
    ecmSxSD_SetEnable (NETID, AXIS, true, &errCode); // SD 속도 설정

    ecmSxCfg_SetSpeedPatt (NETID, AXIS, ecmSMODE_SCURVE, 0, 0, NORM_SPEED,
    NORM_SPEED*5, NORM_SPEED*5, &errCode);
    ecmSxMot_VMoveStart (NETID, AXIS, ecmDIR_P, &errCode);
    // 이 이후에 축의 이송은 SD 신호가 감지되지 않으면 NORM_SPEED 속도로
    이송되다가, SD 신호가 ON 으로 감지되면 SD_SPEED 속도로 이송하게 된다
}

```

---

<h2 style="margin: 0;">NAME</h2> <p style="margin: 5px 0 0 20px;">ecmSxSD_SetSpeed</p> <p style="margin: 5px 0 0 20px;">ecmSxSD_GetSpeed</p> <p style="margin: 5px 0 0 20px;">- SD 속도 설정 및 반환</p>	<b>I N F O R M A T I O N</b>
	 Home      Operation Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxSD\_SetSpeed ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 SdSpeed, [out] t\_i32 \*ErrCode)
- t\_f64 ecmSxSD\_GetSpeed ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxSD\_SetSpeed 함수는 SD 센서가 활성화되었을 때 적용할 SD 속도를 설정합니다.  
 ecmSxSD\_GetSpeed 함수는 현재 설정된 SD 속도를 반환합니다.

## PARAMETER





- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ SdSpeed: SD 센서가 활성화되었을 때 적용할 SD 속도를 나타내는 인자입니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* ecmSxSD\_SetSpeed 함수의 반환값 : 함수 수행의 성공 여부를 반환합니다.

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

\* ecmSxSD\_GetSpeed 함수의 반환값 : 현재 설정된 SD 속도를 반환합니다.

NAME	INFORMATION
ecmSxSD_SetOffset	 Home      Operation
ecmSxSD_GetOffset	Function
- SD Offset 설정 및 반환	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxSD\_SetOffset ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 OffsetMode, [in] t\_f64 Offset, [out] t\_i32 \*ErrCode)

□ t\_success ecmSxSD\_GetOffset ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*OffsetMode, [out] t\_f64 \*Offset, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxSD\_SetOffset 함수는 SD Offset 에 관한 환경을 설정하는 함수입니다.

ecmSxSD\_GetOffset 함수는 현재 설정된 SD Offset 에 관한 환경을 읽는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ OffsetMode: SD 오프셋 모드를 나타내는 인자입니다.





Value	Meaning
0 (ecmSD_OFS_NONE)	OFFSET 적용 안함
1 (ecmSD_OFS_TIME)	시간 오프셋을 적용함. Offset 으로 지정한 시간만큼 지연되어서 SD 동작이 시작됩니다.
2 (ecmSD_OFS_CMDPOS)	SD 센서가 감지된 후에 커맨드 위치 기준으로 Offset 거리만큼 더 지나고 나서 SD 동작이 시작됩니다.
3 (ecmSD_OFS_FEEDPOS)	SD 센서가 감지된 후에 피드백 위치 기준으로 Offset 거리만큼 더 지나고 나서 SD 동작이 시작됩니다.

- ▶ Offset: SD 센서가 감지된 이후에 실제 SD 동작이 시작되는데 까지 걸리는 시간 또는 거리
- ▶ 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.

<h2>NAME</h2> <p>ecmSxSD_SetLatchMode</p> <p>ecmSxSD_GetLatchMode</p> <p>- SD 신호의 Latch 모드 설정 및 반환</p>	<b>INFORMATION</b>	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
	 Level 1	
	 위험 요소 없음	

## SYNOPSIS

- t\_success ecmSxSD\_SetLatchMOde ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsLatchMode, [out] t\_i32 \*ErrCode)
- t\_bool ecmSxSD\_GetLatchMode ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxSD\_SetLatchMOde SD 함수는 센서가 감지되고 나서 다시 SD 센서가 OFF 되었을 때 속도를 원래의 속도로 복원할 것인지 아니면 계속 SD 속도를 유지할 것인지를 설정하는 함수입니다.

ecmSxSD\_GetLatchMOde SD 함수는 SD 래치 모드를 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ IsLatchMode: SD 래치 모드를 나타내는 인자입니다.





Value	Meaning	Default
0	SD 신호 꺼지면 다시 원래의 속도로 복귀	
1	SD 신호 꺼져도 SD 속도 유지	Y

- ▶ 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.

<h2>NAME</h2> <p>ecmSxSD_SetEnable</p> <p>ecmSxSD_GetEnable</p> <p>- SD 신호의 사용 여부 설정 및 반환</p>	<b>INFORMATION</b>	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
	 Level 1	
	 위험 요소 없음	

## SYNOPSIS

- t\_success ecmSxSD\_SetOffset ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)
- t\_bool ecmSxSD\_GetOffset ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 SD Enable 여부를 설정 및 반환하는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ IsEnable: 사용 여부(0:Disable 1: Enable)
- ▶ 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


ecmSxSD\_RestoreSpeed


- 원래속도로 복귀 명령


### INFORMATION

 Home      Operation

Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxSD\_RestoreSpeed ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 SD 속도로 진행중인 경우에 원래의 속도로 복귀하는 함수입니다.





## 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	설정에 실패하였음을 의미합니다.
1	설정에 성공하였음을 의미합니다.

<h2>NAME</h2> <p>ecmSxSD_InputStatus</p> <p>- SD 신호 상태 반환</p>	INFORMATION	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
	 Level 1	
 위험 요소 없음		

## SYNOPSIS

□ t\_bool ecmSxSD\_GetInputStatus ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 SD 신호상태를 반환하는 함수입니다.

## 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	SD 신호 입력이 들어오지 않음.
1	SD 신호 입력이 들어옴

**NAME**

ecmSxSD\_GetActStatus  
- SD 속도로 진행여부 확인

**INFORMATION**

 Home	Operation
	Function
 VC++/BCB/.NET	
 Level 1	
 위험 요소 없음	

**SYNOPSIS**

□ t\_success ecmSxSD\_GetActStatus ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 SD 속도로 진행여부를 확인하는 함수입니다.

**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	SD 속도로 진행되고 있지 않음
1	SD 속도로 진행 중에 있음

### 14.2.10 오토토크(Auto Torque) 기능 환경 설정

<b>NAME</b>  ecmSxCfg_AutoTorq_SetValMode  ecmSxCfg_AutoTorq_GetValMode  - 토크 값 설정모드 설정/반환	<b>INFORMATION</b>
	Single Axis Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_AutoTorq\_SetValMode ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 ValMode, [in] t\_i32 NumMultiVals, [out] t\_i32 \*ErrCode)
- t\_i32 ecmSxCfg\_AutoTorq\_GetValMode ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*NumMultiVals, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)를 설정 합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ RingDirMode : 대상 축에 링 카운터 모드.
- ValMode: 해당 축의 AutoTorq 기능의 출력 토크 설정 모드를 Single-Value 모드로 사용할 지, Multi-Value 모드로 사용할 지를 설정하는 매개변수입니다..
- 이 매개변수에 설정할 수 있는 값은 아래 표와 같다. 아래 표에서 정의된 값에 대해서는 ComiEcatSdk\_Def.h 파일에서 EEcmAutoTorqValMode 선언문을 참고하기 바랍니다.

정의된 값	숫자	Meaning
ecmATRQ_VAL_SINGLE	0	자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)를 'Single Value 모드' 로 설정합니다.
ecmATRQ_VAL_MULTI	1	자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)를 'Multi-Value 모드' 로 설정합니다.

- ▶ NumMultiVals : 이 매개변수는 ValMode 가 ecmATRQ\_VAL\_MULTI 으로 설정되었을 때만 사용되는 매개변수로서 출력 토크의 값을 몇 개 사용할 지를 설정합니다.
- 이 함수를 통해서 Value Mode 를 'Multi-Value 모드'로 설정하였다면, ecmSxCfg\_AutoTorq\_SetMultiVal() 함수를 실행하여 NumMultiVals 매개변수를 통해서 설정한 수만큼 'Multi-

Value' 각각의 단계에 해당하는 토크값과 해당 토크가 출력되어야 하는 시간을 설정해주어야 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	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## EXAMPLE

\* `ecmSxCfg_AutoTorq_SetValue` Examplpe 참고

## NAME

ecmSxCfg\_AutoTorq\_SetMultiVal  
 ecmSxCfg\_AutoTorq\_GetMultiVal  
 - 토크 값과 출력 시간 정의 설정/반환

## INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCfg\_AutoTorq\_SetMultiVal ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 ValIndex, [in] t\_i32 TorqVal, [in] t\_i32 Duration, [ou] t\_i32 \*ErrCode)

□ t\_success ecmSxCfg\_AutoTorq\_GetMultiVal ([in] t\_i32 NetID, [in] t\_i32 Axis, [ou] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 자동토크출력 기능에서의 토크값 설정 모드(Torque Value Mode)가 'Multi-Value Mode'로 설정된 경우에만 사용되는 함수로서, Multi-Value 리스트의 아이템 각각의 토크 값과 출력 시간을 정의하는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ ValIndex : 설정하고자 하는 값의 Multi-Value 리스트내에서의 순서이다. 이 값은 0 부터 설정될 수 있으며, ecmSxCfg\_AutoTorq\_SetValMode() 함수의 NumMultiVals 매개변수를 통해서 정의한 값보다 작아야 합니다.
- ▶ TorqVal : 출력 토크값을 설정한다. 토크값은 0.1% 단위의 값으로 설정한다. 다시 말해서 이 값을 1로 하면 0.1%의 토크를 의미하며, 10은 1%의 토크를 의미합니다.
- ▶ Duration : 해당 토크값을 출력하는 지속 시간을 msec 단위로 설정합니다.
- ▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## EXAMPLE

---

```

C/C++
// Single-Value 모드로 설정하여 자동토크출력기능을 사용하는 예 제입니다.
#include "ComiEcatSdk_Api.h"

#define AXIS      0

#define NUM_TORQ_VALS 5
t_i32 nErrCode;
t_i32 nTroqList[NUM_TORQ_VALS] = {100, 90, 80, 70, 50};
t_i32 nTimeList[NUM_TORQ_VALS] = {10, 10, 10, 10, 1000};
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
    }
    //ALState OP 모드로 전환
    ecNet_SetALState(Device_Info.NetIdx, ecAL_STATE_OP, &Error_Num);

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

/*****
* OnMuiltTorq: 이 함수는 OnMuiltTorq 사용 입니다
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnMuiltTorq ()
{
    // 1) 토크값설정모드(Torque Value Mode) 설정 //

```

---

---

```

    if(!ecmSxCfg_AutoTorq_SetValMode (Device_Info.NetIdx , AXIS,
    ecmATRQ_VAL_MULTI, NUM_TORQ_VALS, &Error_Num))
    {
        //실패 시 에러처리
    }
    // 2) Mutliple 출력 토크값 설정 //
    for(int i=0; i<NUM_MULT_TORQ_VALS; i++) {
        ecmSxCfg_AutoTorq_SetMultiVal(Device_Info.NetIdx , AXIS, i, nTroqList[i],
    nTimeList[i], &Error_Num);
    }

    // 3) 자동토크출력 기능 활성화 //
    if(!ecmSxCfg_AutoTorq_SetEnable (Device_Info.NetIdx , AXIS, true, true, &Error_Num))
    {
        //실패 시 예외처리
    }

    // 4) PTP 이송을 수행한다. => 이 이송이 완료되면 자동으로 토크 출력됨. //
    if(!ecmSxMot_Move (Device_Info.NetIdx , AXIS, 100000, &Error_Num))
    {
        //실패 시 예외처리
    }

    // 5) 자동토크출력 기능이 완료될 때까지 기다린다.
    // 참고로, 토크모드로 전환되면 ecmSxSt_GetMotState() 함수는
    ecmMST_IN_AUTO_TORQ_MDOE(15) 값을 반환한다.
    if(!ecmSxSt_WaitCompt (Device_Info.NetIdx , AXIS, &Error_Num))
    {
        //실패 시 예외처리
    }
    // 6) 앞서 ecmSxCfg_AutoTorq_SetValue() 함수를 실행할 때 IsOneShotEnable 매개변수의
    값을 true 로 하였으므로 여기에서 비활성화하지 않아도 자동으로 비활성화된다. 그렇지
    않으면 여기에서 ecmSxCfg_AutoTorq_SetValue() 함수를 이용해서 비활성화할 필요 있다. //
}

```


---





## NAME


ecmSxCfg\_AutoTorq\_SetValue  
 ecmSxCfg\_AutoTorq\_GetValue  
 - 토크 값 정의 설정/반환

## INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_AutoTorq\_SetValue ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 OutTorqVal, [out] t\_i32 \*ErrCode)
- t\_i32 ecmSxCfg\_AutoTorq\_GetValue ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 자동토크출력 기능에서의 토크 값 설정 모드(Torque Value Mode)가 'Single-Value Mode'로 설정된 경우에만 사용하는 함수로서, 토크 모드로 전환되었을 때 출력되어야 하는 토크 값을 정의하는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ OutTorqVal : 출력 토크값을 설정한다. 토크값은 0.1% 단위의 값으로 설정한다. 다시 말해서 이 값을 1로 하면 0.1%의 토크를 의미하며, 10은 1%의 토크를 의미합니다.
- ▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## EXAMPLE

---

```

C/C++
// Single-Value 모드로 설정하여 자동토크출력기능을 사용하는 예 제입니다.
#include "ComiEcatSdk_Api.h"

#define AXIS      0
#define TORQ_VAL 50 // 출력 토크값. 값의 단위는 0.1%이다. 따라서 50 은 5%를 의미한다

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

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

/*****
* OnSigleTorq: 이 함수는 OnSigleTorq 사용 입니다
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnSigleTorq ()
{
    // 1) 토크값설정모드(Torque Value Mode) 설정 //
    if(!ecmSxCfg_AutoTorq_SetValMode (Device_Info.NetIdx , AXIS,
    ecmATRQ_VAL_SINGLE, 1, &nErrCode))
    {
        //실패 시 예외처리
    }

    // 2) 출력 토크값 설정 //
    if(!ecmSxCfg_AutoTorq_SetValue (Device_Info.NetIdx , AXIS, TORQ_VAL, &nErrCode))

```

---

---

```

    {
        //실패 시 예외처리
    }

    // 3) 자동토크출력 기능 활성화 //
    if(!ecmSxCfg_AutoTorq_SetEnable (Device_Info.NetIdx , AXIS, true, true, &nErrCode))
    {
        //실패 시 예외처리
    }

    // 4) PTP 이송을 수행한다. => 이 이송이 완료되면 자동으로 토크 출력됨. //
    if(!ecmSxMot_Move (Device_Info.NetIdx , AXIS, 100000, &nErrCode))
    {
        //실패 시 예외처리
    }

    // 5) 자동토크출력 기능이 완료될 때까지 기다린다.
    // 참고로, 토크모드로 전환되면 ecmSxSt_GetMotState() 함수는
    ecmMST_IN_AUTO_TORQ_MDOE(15) 값을 반환한다.
    if(!ecmSxSt_WaitCompt (Device_Info.NetIdx , AXIS, &nErrCode))
    {
        //실패 시 예외 처리
    }
}
// 6) 앞서 ecmSxCfg_AutoTorq_SetValue() 함수를 실행할 때 IsOneShotEnable 매개변수의
값을 true 로 하였으므로 여기에서 비활성화하지 않아도 자동으로 비활성화된다. 그렇지
않으면 여기에서 ecmSxCfg_AutoTorq_SetValue() 함수를 이용해서 비활성화할 필요 있다. //
}

```

---

# NAME

ecmSxCfg\_AutoTorq\_SetLimit  
 ecmSxCfg\_AutoTorq\_GetLimit  
 - 토크 출력 완료 조건 설정/반환

## INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

# SYNOPSIS

- t\_success ecmSxCfg\_AutoTorq\_SetLimit ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui32 LimitMask, [in] t\_f64 HighSpeedLimit, [in] t\_f64 LowSpeedLimit, [in] t\_i32 TimeLimit,[out] t\_i32 \*ErrCode)
- t\_success ecmSxCfg\_AutoTorq\_GetLimit ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_ui32\* LimitMask, [out] t\_f64\* HighSpeedLimit, [out] t\_f64\* LowSpeedLimit, [out] t\_i32\* TimeLimit,[out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 자동토크출력 기능에서의 토크출력 모드를 자동으로 완료하는 완료 조건을 설정하는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ LimitMask : 적용하고자 하는 완료 조건의 종류를 각 비트별로 설정한다. 단, 이 값을 0으로 하면 자동 완료 조건을 설정하지 않게 된다. 따라서 그러한 경우에는 ecmSxCfg\_AutoTorq\_SetEnable() 함수를 통해서만 토크출력모드를 완료할 수 있다. LimitMask 값의 각 비트의 의미는 아래 와 같다. 아래 표에서 “Bit 번호 정의값” 은 ComiEcatSdk\_Def.h 파일의 EEcmAtrqLimMask 선언문을 참고하기 바랍니다.

Bit 번호	Bit 번호 정의값	Meaning
BIT00	ecmATRQ_LMBIT_HIGHSPD	HighSpeedLimit 의 적용 여부 (0 : 적용안함, 1: 적용함)
BIT01	ecmATRQ_LMBIT_LOWSPD	LowSpeedLimit 의 적용 여부 (0 : 적용안함, 1: 적용함)
BIT02	ecmATRQ_LMBIT_TIME	TimeLimit 의 적용 여부 (0 : 적용안함, 1: 적용함)

▶ HighSpeedLimit : 해당 축의 이송속도의 최대값을 설정한다. 현재 모터의 이송속도(피드백 속도)가 여기에서 정의한 값보다 같거나 크게 되면 토크 출력모드를 자동으로 완료합니다.. 단, 이 값의 적용 여부는 LimitMask 매개변수의 bit0 의 설정에 의해서 결정됩니다.

▶ LowSpeedLimit : 해당 축의 이송속도의 최소값을 설정한다. 현재 모터의 이송속도(피드백 속도)가 여기에서 정의한 값보다 같거나 작게 되면 토크 출력모드를 자동으로 완료합니다. 단, 이 값의 적용 여부는 LimitMask 매개변수의 bit1 의 설정에 의해서 결정됩니다.

▶ **TimeLimit**: 토크출력 모드의 최대 유지 시간을 설정한다. 토크모드로 전환된 이후에 여기서 정의한 시간이 경과되면 토크출력모드를 자동으로 완료합니다. 단, 이 값의 적용 여부는 LimitMask 매개변수의 bit2 의 설정에 의해서 결정됩니다.





▶ **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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## 14.2.11 기타 환경 설정

NAME	INFORMATION
ecmSxCfg_SetTouchProbeFunc	 Single Axis Function
ecmSxCfg_GetTouchProbeFunc	 VC++/BCB/.NET
-	 Level 1
-	 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmSxCfg\_SetTouchProbeFunc([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui8 TouchProbeIndex, [in] t\_byte TouchProbeFuncVal, [out] t\_i32 \*ErrCode)
- t\_byte ecmSxCfg\_GetTouchProbeFunc ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui8 TouchProbeIndex, [out] t\_i32 \*ErrCode)

## DESCRIPTION

ecmSxCfg\_SetTouchProbeFunc 함수는 서보드라이버에서 제공하는 터치프루브(Touch Probe) 기능에 대한 환경을 설정합니다.

ecmSxCfg\_GetTouchProbeFunc 함수는 서보드라이버에서 제공하는 터치프루브(Touch Probe) 기능의 현재 설정되어 있는 환경을 반환합니다.

터치프루브(Touch Probe) 기능은 일부 서보드라이버에서 제공하는 기능으로서 터치프루브 센서 입력에 동기하여 모터의 위치를 래치(임시 저장)하여 놓고 그 값을 사용자가 읽어갈 수 있도록 하는 기능입니다.

터치프루브 기능은 모든 서보드라이버가 지원하는 기능은 아니므로 서보드라이버의 매뉴얼을 참고하여서 해당 서보드라이버가 터치프루브 기능을 지원하는 경우에만 터치프루브 기능을 사용할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TouchProbeIndex : 터치프루브 번호. 보통 하나의 서보드라이버에서는 2 개의 터치프루브 입력을 지원합니다. 이 번호는 0 부터 시작합니다.
- ▶ TouchProbeFuncVal : 터치프루브 기능을 설정하는 값입니다. 이 값은 CiA 402 프로파일의 60B8h 오브젝트에 해당하는 값입니다. 다만 60B8h 오브젝트에서는 2 채널의 터치프루브의 속성을 2 바이트를 통해서 정의하는데 이 함수에서는 TouchProbeIndex 로 설정 대상 터치프루브를 지정하고 1 바이트의 값으로 속성을 정의합니다. 파나소닉 서보드라이버를 사용하는 경우에 이 인자의 비트 구성은 다음과 같습니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\*ecmSxCfg\_SetTouchProbeFunc():

\*ecmSxCfg\_GetTouchProbeFunc():

## EXAMPLE

NAME	INFORMATION
ecmSxCfg_SetOperMode ecmSxCfg_GetOperMode -	Single Axis Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_SetOperMode([in] t\_i32 NetID, [in] t\_i32 Axis, [in] EEcmOperMode OperMode, [out] t\_i32 \*ErrCode)
- t\_i32 ecmSxCfg\_GetOperMode([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수들은.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ OperMode: 활성화 여부 (0: 활성화 1: 비활성화)
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE





\*ecmSxCfg\_SetOperMode(): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

\*ecmSxCfg\_GetOperMode():

## EXAMPLE



NAME	I N F O R M A T I O N
ecmSxCfg_SetMaxTorqOfCV ecmSxCfg_GetMaxTorqOfCV -	<div style="border-bottom: 1px solid black; padding-bottom: 2px;">  Single Axis Function         </div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;">  VC++/BCB/.NET         </div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;">  Level 1         </div> <div style="border-bottom: 1px solid black; padding-bottom: 2px;">  위험 요소 없음         </div>

## SYNOPSIS

- t\_success ecmSxCfg\_SetMaxTorqOfCV([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui16 MaxTorq, [out] t\_i32 \*ErrCode)
- t\_ui16 ecmSxCfg\_GetMaxTorqOfCV([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수들은

## PARAMETER

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

## RETURN VALUE

\* ecmSxCfg\_SetMaxTorqOfCV(): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

\* ecmSxCfg\_GetMaxTorqOfCV():

## EXAMPLE

<b>NAME</b>	<b>INFORMATION</b>
ecmSxCfg_SetMaxSpdOfCT	Single Axis Function
ecmSxCfg_GetMaxSpdOfCT	VC++/BCB/.NET
-	Level 1
	위험 요소 없음

## SYNOPSIS

- ❑ t\_success ecmSxCfg\_SetMaxSpdOfCT([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui32 MaxSpd, [out] t\_i32 \*ErrCode)
- ❑ t\_ui32 ecmSxCfg\_GetMaxOfCT([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수들은

## PARAMETER

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





## RETURN VALUE

\* ecmSxCfg\_SetMaxSpdOfCT (): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

\* ecmSxCfg\_GetMaxSpdOfCT ():

## EXAMPLE

<b>NAME</b>	<b>INFORMATION</b>
ecmSxCfg_SetMaxProfSpdOfCT	 Single Axis Function
ecmSxCfg_GetMaxProfSpdOfCT	 VC++/BCB/.NET
-	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCfg\_SetMaxProfSpdOfCT([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 MaxProfSpd, [out] t\_i32 \*ErrCode)
- t\_f64 ecmSxCfg\_GetMaxProfSpdOfCT([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수들은

## PARAMETER

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





## RETURN VALUE

\* ecmSxCfg\_SetMaxProfSpdOfCT (): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

\* ecmSxCfg\_GetMaxProfSpdOfCT ():

## EXAMPLE

<h1>NAME</h1> <p>ecmSxCfg_SetSyncOtherEnv ecmSxCfg_GetSyncOtherEnv - 다른 축 동기화 이동</p>	<b>INFORMATION</b>
	<ul style="list-style-type: none"> <li> Single Axis Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>

## SYNOPSIS

- t\_cmdidx ecmSxCfg\_SetSyncOtherEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 SyncAxis, [in] t\_i32 SyncType, [in] t\_i32 PosSyncMethod, [in] t\_f64 SyncPosition, [out] t\_i32 \*ErrCode)
- t\_success ecmSxCfg\_GetSyncOtherEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32\* SyncAxis, [out] t\_i32\* SyncType, [out] t\_i32\* PosSyncMethod, [out] t\_f64\* SyncPosition, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 한 축과 또 다른 축을 동기화 하여 이동하는 함수 입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ SyncAxis : Axi 축과 동기화 할 축
- ▶ SyncType : 동기화 타입 아래 표를 참고 합니다.

Value	Meaning
ecmSYNC_OTHER_START	지정 축이 시작 할 때 동기화 시작
ecmSYNC_OTHER_ACC_INI	지정 축 초기 가속도에 맞추어서 동기화 시작
ecmSYNC_OTHER_ACC_END	지정 축 종료 가속도에 맞추어서 동기화 시작
ecmSYNC_OTHER_DEC_INI	지정 축 초기 감속도에 맞추어서 동기화 시작
ecmSYNC_OTHER_DEC_END	지정 축 종료 감속도에 맞추어서 동기화 시작
ecmSYNC_OTHER_POSITION	지정 축 위치에 맞추어서 동기화 시작

- ▶ PosSyncMethod : 동기화 모드 아래 표를 참고 합니다.

Value	Meaning
ecmPOS_SYNC_GT	position 보다 큰(Greater Than) 경우
ecmPOS_SYNC_GE	position 보다 크거나 같은(Greater or Equal) 경우

ecmPOS_SYNC_LT	position 보다 작은(Less Than) 경우
ecmPOS_SYNC_LE	position 보다 크거나 같은(Less or Equal) 경우
ecmPOS_SYNC_POS_CROSS	position 보다 작은 값에서 큰 값으로 변하는 경우
ecmPOS_SYNC_NEG_CROSS	position 보다 큰 값에서 작은 값으로 변하는 경우

▶ SyncPosition : 동기화를 시작 할 위치

▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.


**NAME**


ecmSxCfg\_SetSyncOtherEnable  
 ecmSxCfg\_GetSyncOtherEnable  
 - 자동 토크 기능 활성화/비활성화

**INFORMATION**

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

- t\_cmdidx ecmSxCfg\_SetSyncOtherEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [in] t\_bool IsOneShot, [out] t\_i32 \*ErrCode)
- t\_success ecmSxCfg\_GetSyncOtherEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_bool\* IsEnable, [out] t\_bool\* IsOneShot, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 자동토크출력 기능을 활성화 또는 비활성화하는 함수입니다.

**PARAMETER**





- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ IsEnable : 이 값을 0 으로 하면 동기화 기능을 비활성화되고, 1 로 하면 활성화 됩니다.
- ▶ IsOneShot : 이 값은 0 으로 하면 OneShot 기능이 비활성화 되고, 1 로 하면 활성화 됩니다.
- ▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.



<h1>NAME</h1> <p><b>ecmSxMot_OverrideMove</b></p> <p>- 단축 상대위치 오버라이드 이송</p>	<b>INFORMATION</b>
	 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() 이송 함수를 통하여 수행되는 상대좌표 In-Position 모션에 대하여 상대좌표값, 즉 목표 거리값을 오버라이딩하는 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() 이송 함수를 통하여 수행되는 절대좌표 In-Position 모션에 대하여 목표 절대좌표값을 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	I N F O R M A T I O N
ecmSxMulTorq1_Start -	 Single Axis Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

```

□ t_cmdidx ecmSxMulTorq1_Start([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 InitialTorq,
[in] TEcmMTQ1Item NextTorqList[], [in] t_ui16 NumNextTorqItems, [out] t_i32
*ErrCode)

```

## DESCRIPTION

이 함수는





## PARAMETER

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

## RETURN VALUE

\*이 리턴은

## EXAMPLE

NAME	I N F O R M A T I O N
ecmSxMulTorq1_Stop -	 Single Axis Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

t\_cmdidx ecmSxMulTorq1\_Stop([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_ui16 StopTorq, [in] t\_bool IsApplyStopTorq, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는





## PARAMETER

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

## RETURN VALUE

\*이 리턴은

## EXAMPLE

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

## SYNOPSIS

❑ t\_cmdidx ecmSxMot\_SetTargTorq([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TargTorq, [out] t\_i32 \*ErrCode)

❑ t\_i32 ecmSxMot\_GetTargTorq([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

▶ NetID : Network 번호

▶ Axis : 대상 축 번호.

▶ TargTorq:





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

## RETURN VALUE

\* ecmSxMot\_SetTargTorq(): 이 리턴은

\* ecmSxMot\_GetTargTorq(): 이 리턴은

## EXAMPLE

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

## SYNOPSIS

- ❑ t\_success ecmSxMot\_SetTargVel([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 TargVel, [out] t\_i32 \*ErrCode)
- ❑ t\_i32 ecmSxMot\_GetTargVel([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

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

## RETURN VALUE


- \* ecmSxMot\_SetTargVel(): 이 리턴은
- \* ecmSxMot\_GetTargVel(): 이 리턴은

## EXAMPLE


**NAME**

ecmSxCmpOne\_SetHandler-

**INFORMATION**
 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

```

❑ t_success ecmSxCmpOne_SetHandler([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32
HandlerType, [in] HANDLE Handler, [in] UINT nMessage, [in] LPARAM lParam, [out]
t_i32 *ErrCode)

```

**DESCRIPTION**

이 함수는

**PARAMETER**

▶ NetID : Network 번호

▶ Axis : 대상 축 번호.

▶ HandlerType:

▶ Handler:

▶ nMessage:

▶ lParam:

▶ 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.

**EXAMPLE**



## NAME

ecmSxCmpOne\_SetEnable

ecmSxCmpOne\_GetEnable

- 위치 비교 출력 기능 활성화/비활성화

### INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCmpOne\_SetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [in] t\_i32 GdoChanIdx, [out] t\_i32 \*ErrCode)

□ t\_success ecmSxCmpOne\_GetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [in] t\_i32 GdoChanIdx, [out] t\_i32 \*ErrCode)

## DESCRIPTION

‘One-shot’ 모드의 위치비교출력 기능을 활성화 또는 비활성화합니다.

## PARAMETER

▶ NetID : Network 번호

▶ Axis : 대상 축 번호.

▶ IsEnable : 이 값을 0 으로 하면 ‘One-shot’ 모드의 위치비교출력 기능이 비활성화되고, 1 로 하면 활성화됩니다.

▶ GdoChanIdx : 위치비교신호(CMP)를 출력할 범용 디지털 채널 번호를 설정한다. 하나의 마스터 장치는 4 개의 범용 디지털 출력 채널을 제공합니다. 이 중에서 위치비교신호를 출력하는 용도로 사용할 채널을 선택하여 사용하게 된다. 채널 번호는 0 부터 시작합니다.

▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## EXAMPLE

---

```

C/C++
//P1(0) 에서 P2(2,000,000)로 이송하도록 하면서 500,000 의 위치를 지날 때에 트리거 신호가
출력되게 하는 예제입니다.
#include "ComiEcatSdk_Api.h"

#define AXIS      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 실패 시 예외 처리
    }
    //ALState OP 모드로 전환
    ecNet_SetALState(Device_Info.NetIdx, ecAL_STATE_OP, &Error_Num);

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

/*****
* OnCmp: 이 함수는 위치 비교 출력 함수 관련 예제
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnCmp ()
{
    // 1) 시작 위치로 이송 //
    if(!ecmSxMot_MoveTo(Device_Info.NetIdx , AXIS, 0, &ErrCode)){
        // 실패시 에러 처리
    }
    // 2) 위치비교출력 환경 설정 //
    if(!ecmSxCmpOne_SetEnv (NETID, AXIS, ecmCMP_CNTR_FEED,
    ecmCMP_METH_EQ_BIDIR, 500000/*CmpRefPos*/, ecmLOGIC_A, 100/*OutSigOnTime*/,
    &Error_Num)){
        //실패 시 에러 처리
    }
}

```

---

---

```

}
// 3) 위치비교출력 기능 활성화. CMP 신호는 GDO CH0 을 통해서 출력 //
if(!ecmSxCmpOne_SetEnable (Device_Info.NetIdx , AXIS, true/*IsEnable*/, 0/*GDO CH*/,
&Error_Num)){
    // 실패 시 에러처리
}
// 4) CMP 모니터 카운트값 리셋 //
ecmSxCmpOne_ResetOutSts(Device_Info.NetIdx, AXIS, &Error_Num);
// 5) 목표좌표로 이송 시작 //
if(!ecmSxMot_MoveToStart(Device_Info.NetIdx, AXIS, 2000000, &Error_Num)){
    //실패 시 에러 처리
}
// 6) 이송 완료 체크 & CMP 출력 상태 감시 //
t_i32 OutCount=0, PrvOutCount=0;
t_f64 LastOutPos;
while(ecmSxSt_IsBusy(Device_Info.NetIdx, AXIS, &Error_Num)){
    if(ecmSxCmpOne_GetOutSts(Device_Info.NetIdx, AXIS, &OutCount, &LastOutPos,
&Error_Num)){
        if(OutCount > PrvOutCount){
            PrvOutCount = OutCount;
            //"CMP 신호가 (%f)위치에서 출력되었습니다!"
        }
    }
    Sleep(1);
}

```

---

<h2>NAME</h2> <p>ecmSxCmpOne_SetEnv ecmSxCmpOne_GetEnv - 위치 비교 출력 기능 옵션 설정 /반환</p>	<b>INFORMATION</b>
	<p>Single Axis Function</p> <hr/> <p>VC++/BCB/.NET</p> <hr/> <p>Level 1</p> <hr/> <p>위험 요소 없음</p>

## SYNOPSIS

- t\_success ecmSxCmpOne\_SetEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 CmpCntrType, [in] t\_i32 CmpMethod, [in] t\_f64 CmpRefPos, [in] t\_i32 OutSigLogic, [in] t\_i32 OutSigOnTime, [out] t\_i32 \*ErrCode)
- t\_success ecmSxCmpOne\_GetEnv ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32\* CmpCntrType, [out] t\_i32\* CmpMethod, [out] t\_f64\* CmpRefPos, [out] t\_i32\* OutSigLogic, [out] t\_i32\* OutSigOnTime, [out] t\_i32 \*ErrCode)

## DESCRIPTION

‘One-shot’ 모드 위치비교출력 기능을 위한 비교 조건 및 기타 환경을 설정한다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ CmpCntrType : 비교대상이 되는 위치값의 종류를 선택한다. 이 매개변수에 설정할 수 있는 값들은 아래의 표와 같다. 아래 표에서 ‘정의된 값’ 은 ComiEcatSdk\_Def.h 파일에서 EEcmCmpCntrType 선언문을 참고하기 바란다.

정의된 값	숫자값	값의 의미
ecmCMP_CNTR_COMM	0	지령위치(Command Position)를 비교 대상 위치 값으로 설정한다.
ecmCMP_CNTR_FEED	1	궤환위치(Feedback Position)를 비교 대상 위치 값으로 설정한다.

- ▶ CmpMethod : 비교대상위치값고 기준위치값을 비교하는 방법을 설정한다. 이 매개변수에 설정할 수 있는 값들은 아래의 표와 같다. 아래 표에서 ‘정의된 값’ 은 ComiEcatSdk\_Def.h 파일에서 EEcmCmpMethod 선언문을 참고하기 바랍니다..

정의된 값	숫자값	값의 의미
ecmCMP_METH_EQ_NDIR	0	(-)방향 이송 중에 비교대상위치와 기준위치가 같아지면 신호 출력
ecmCMP_METH_EQ_PDIR	1	(+)방향 이송 중에 비교대상위치와 기준위치가 같아지면 신호 출력

ecmCMP_METH_EQ_BIDIR	2	이송 방향에 상관없이 비교대상위치와 기준위치가 같아지면 신호 출력
ecmCMP_METH_LT	3	비교대상위치가 기준위치보다 작으면 신호 출력
ecmCMP_METH_GT	4	비교대상위치가 기준위치보다 크면 신호 출력
ecmCMP_METH_LE	5	비교대상위치가 기준위치보다 작거나 같으면 신호 출력
ecmCMP_METH_GE	6	비교대상위치가 기준위치보다 크거나 같으면 신호 출력

▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

EXAMPLE

\* ecmSxCmpOne\_SetEnable Exampd 참고

## NAME

ecmSxCmpOne\_GetOutSts

- 위치 비교 출력 동작 상태 반환

### INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCmpOne\_GetOutSts ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*OutCount, [out] t\_f64 \*LastOutPos, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 'One-shot' 모드 위치비교출력 기능의 동작 상태를 매개 변수들을 통해서 반환해주는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ OutCount : 이 매개변수를 통해서 'One-shot' 모드 위치비교출력 기능을 통해서 출력된 신호(CMP)의 횟수를 반환 합니다.
- ▶ LastOutPos : 이 매개변수를 통해서 마지막에 CMP 신호가 출력되었을 때의 위치값을 반환 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	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## REFERENCE

□ 마스터장치가 처음 전원이 인가된 상태에서는 CMP 신호 카운트값은 0 이 되지만, ecmSxCmpOne\_SetEnable() 함수를 실행할 때마다 CMP 신호 카운트값이 0 으로 리셋되지는 않는다. CMP 신호 카운트값을 0 으로 리셋하려면 ecmSxCmpOne\_ResetOutSts () 함수를 실행해주어야 합니다.

EXAMPLE


\* ecmSxCmpOne\_SetEnable Exampd 참고


**NAME**


**ecmSxCmpOne\_ResetOutSts**  
- 위치 비교 출력 리셋

**INFORMATION**

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmSxCmpOne\_ResetOutSts ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 ecmSxCmpOne\_GetOutSts() 함수를 통해서 반환되는 CMP 신호가 출력된 횟수 카운트값(OutCount)과 마지막 트리거 위치>LastOutPos)를 모두 0으로 리셋합니다..

**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	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

**EXAMPLE**

\* ecmSxCmpOne\_SetEnable Exampd 참고



## NAME


ecmSxCmpCont\_SetEnable


ecmSxCmpCont\_GetEnable


- 위치 비교 출력 기능 활성화/비활성화

## INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_success ecmSxCmpCont\_SetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [in] t\_i32 GdoChanIdx, [out] t\_i32 \*ErrCode)

❑ t\_success ecmSxCmpCont\_GetEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32\* GdoChanIdx, [out] t\_i32 \*ErrCode)

## DESCRIPTION

Continuous 모드의 위치비교출력 기능을 활성화 또는 비활성화한다.

## PARAMETER

▶ NetID : Network 번호

▶ Axis : 대상 축 번호.

▶ IsEnable : 이 값이 0 으로 하면 Continuous 모드의 위치비교출력 기능이 비활성화되고, 1 로 하면 활성화됩니다.

▶ GdoChanIdx : 위치비교신호(CMP)를 출력할 범용 디지털 채널 번호를 설정한다. 하나의 마스터 장치는 4 개의 범용 디지털 출력 채널을 제공한다. 이 중에서 위치비교신호를 출력하는 용도로 사용할 채널을 선택하여 사용하게 된다. 채널 번호는 0 부터 시작합니다.

▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.


## NAME


ecmSxCmpCont\_SetTableSize  
 ecmSxCmpCont\_GetTableSize  
 - 위치 비교 출력 기능 위한 테이블 생성

### INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCmpCont\_SetTableSize ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableSize, [out] t\_i32 \*ErrCode)

□ t\_i32 ecmSxCmpCont\_GetTableSize ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 Continuous 모드 위치비교출력 기능을 위한 데이터 테이블의 크기를 설정하는 함수입니다.. 여기서 데이터 테이블이란 연속적인 위치비교출력을 하기 위해서 비교 조건 데이터 세트들을 저장해 놓는 테이블을 말합니다.. Continuous 모드 위치비교출력 기능은 이 데이터 테이블에 등록된 비교 조건 데이터 세트를 순차적으로 적용하여 위치비교출력을 수행하게 됩니다.

## PARAMETER

▶ NetID : Network 번호

▶ Axis : 대상 축 번호.

▶ TableSize : Continuous 모드 위치비교출력 기능을 위한 데이터 테이블의 크기를 설정한다. 이때 설정하는 값의 단위는 비교 조건 데이터 세트의 개수가 된다.

예를 들어서 TableSize 를 100 이라 설정하면 테이블의 크기는 비교 조건 데이터 세트를 100 개 저장할 수 있는 크기로 할당됩니다.

참고로 앞서 설명한 바와 같이 하나의 데이터 세트는 비교대상위치종류 (CmpCntrType), 비교방법(CmpMethod), 기준위치값(CmpRefPos), 출력신호로직(OutSigLogic), 출력신호 ON 시간(OutSigOnTime)의 데이터들로 구성됩니다.

▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

NAME	INFORMATION
ecmSxCmpCont_SetTableData	Single Axis Function
ecmSxCmpCont_GetTableData	VC++/BCB/.NET
- 위치 비교 출력 기능 위한 테이블 생성	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_success ecmSxCmpCont\_SetTableData ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableIndex, [in] t\_i32 CmpCntrType, [in] t\_i32 CmpMethod, [in] t\_f64 CmpRefPos, [in] t\_i32 OutSigLogic, [in] t\_i32 OutSigOnTime, [out] t\_i32 \*ErrCode);
- t\_success ecmSxCmpCont\_GetTableData ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableIndex, [out] t\_i32\* CmpCntrType, [out] t\_i32\* CmpMethod, [out] t\_f64\* CmpRefPos, [out] t\_i32\* OutSigLogic, [out] t\_i32\* OutSigOnTime, [out] t\_i32 \*ErrCode);

## DESCRIPTION

이 함수의 'Continuous' 모드 위치비교출력 기능을 위한 비교 조건 데이터 테이블의 데이터를 정의하는 함수이다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TableIndex : 설정하고자 하는 데이터 테이블의 아이템 인덱스(Index, 순서). 테이블 인덱스값은 0 부터 (TableSize-1) 사이의 값을 사용한다. 여기에서 TableSize 는 ecmSxCmpCont\_SetTableSize() 함수의 매개변수인 TableSize 를 말한다.
- ▶ CmpCntrType : 비교대상이 되는 위치값의 종류를 선택한다. 이 매개변수에 설정할 수 있는 값들은 아래의 표와 같습니다  
아래 표에서 정의된 값은 ComiEcatSdk\_Def.h 파일에서 EEcmCmpCntrType 선언문을 참고하기 바랍니다.

정의된 값	숫자값	값의 의미
ecmCMP_CNTR_COMM	0	지령위치(Command Position)를 비교 대상 위치 값으로 설정한다.
ecmCMP_CNTR_FEED	1	궤환위치(Feedback Position)를 비교 대상 위치 값으로 설정한다.

- ▶ CmpMethod : 비교대상위치값과 기준위치값을 비교하는 방법을 설정한다. 이 매개변수에 설정할 수 있는 값들은 아래의 표와 같다. 아래 표에서 '정의된 값' 은 ComiEcatSdk\_Def.h 파일에서 EEcmCmpMethod 선언문을 참고하기 바랍니다.

▶ **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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## NAME


ecmSxCmpCont\_SetActTblIdx


ecmSxCmpCont\_GetActTblIdx


- 위치 비교 출력 기능 위한 테이블 생성

### INFORMATION

 Single Axis Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCmpCont\_SetActTblIdx ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 TableIndex, [out] t\_i32 \*ErrCode)

□ t\_i32 ecmSxCmpCont\_GetActTblIdx ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 Continuous 모드 위치비교출력 기능의 데이터 테이블에 있는 아이템들 중에서 실행 비교기에 로드되어야 하는 아이템의 인덱스를 지정하는 함수입니다. 실행 비교기란 실제 위치비교출력 기능을 수행하는 모듈을 말한다. 데이터 테이블에 있는 비교데이터 아이템은 실행비교기에 로드되어서 위치비교출력 기능을 수행하게 됩니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ TableIndex : Active Item 으로 등록하고자 하는 아이템의 테이블 상 인덱스(Index, 순서)를 지정한다.
- ▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## NAME

ecmSxCmpCont\_GetOutSts

- 위치 비교 출력 동작 상태 반환

### INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCmpCont\_GetOutSts ([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*OutCount, [out] t\_f64 \*LastOutPos, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 'Continuous' 모드 위치비교출력 기능의 동작 상태를 매개 변수들을 통해서 반환해주는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ OutCount : 이 매개변수를 통해서 'Continuous' 모드 위치비교출력 기능을 통해서 출력된 신호(CMP)의 횟수를 반환합니다.
- ▶ LastOutPos : 이 매개변수를 통해서 마지막에 CMP 신호가 출력되었을 때의 위치 값을 반환합니다.
- ▶ 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 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

## REFERENCE

□ 마스터장치가 처음 전원이 인가된 상태에서는 CMP 신호 카운트값은 0 이 되지만, ecmSxCmpCont\_SetEnable() 함수를 실행할 때마다 CMP 신호 카운트값이 0으로 리셋되지는 않는다. CMP 신호 카운트값을 0으로 리셋하려면 ecmSxCmpCont\_ResetOutSts () 함수를 실행해주어야 합니다.





## NAME

ecmSxCmpCont\_ResetOutSts

- 카운트, 트리거 위치 리셋

### INFORMATION

Single Axis Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

□ t\_success ecmSxCmpCont\_ResetOutSts (t\_i32 NetID, t\_i32 Axis, t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 ecmSxCmpCont\_GetOutSts() 함수를 통해서 반환되는 CMP 신호가 출력된 횟수 카운트값(OutCount)과 마지막 트리거 위치(LastOutPos)를 모두 0으로 리셋합니다.

## 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	함수 수행에 실패 하였음을 의미합니다.
1	함수 수행에 성공 하였음을 의미합니다.

# Motion Multiple Axes 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.



## 15 Motion Multiple Axes Functions

### 15.1 함수 요약

Summary of Functions
<p>□ t_cmdidx ecmMxMot_VMoveStart([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_i32 DirList[], [out] t_i32 *ErrCode) 다축 연속속도 이송을 시작합니다. 이 구동함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmMxMot_VMoveStart_FF([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_i32 DirList[], [out] t_i32 *ErrCode) 다축 연속속도 이송을 시작합니다. 이 구동함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmMxMot_MoveStart([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_f64 DistList[], [out] t_i32 *ErrCode) 다축 상대좌표 이송을 시작합니다. 이 구동함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmMxMot_MoveStart_FF([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_f64 DistList[], [out] t_i32 *ErrCode) 다축 상대좌표 이송을 시작합니다. 이 구동함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmMxMot_MoveToStart([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_f64 PosList[], [out] t_i32 *ErrCode) 다축 절대좌표 이송을 시작합니다. 이 구동함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmMxMot_MoveToStart_FF([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_f64 PosList[], [out] t_i32 *ErrCode) 다축 절대좌표 이송을 시작합니다. 이 구동함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmMxMot_Stop([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode) 다축 이송을 정지 합니다.</p>
<p>□ t_cmdidx ecmMxMot_Stop_FF([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_i32 IsDecStop, [out] t_i32 *ErrCode) 다축 이송을 정지 합니다.</p>
<p>□ t_bool ecmMxSt_IsBusy([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [out] t_i32 *ErrCode) 다축 이송의 모션 완료를 확인합니다,</p>
<p>□ t_success ecmMxSt_WaitCompt([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [out] t_i32 *ErrCode) 다축 이송의 완료 시점까지 대기 합니다.</p>
<p>□ t_success ecmMxSt_WaitCompt_NB([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [out] t_i32 *ErrCode) 다축 이송의 완료 시점까지 대기 합니다.</p>

**15.2** 함수 설명

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmMxMot_VMoveStart</p> <p style="margin: 0;">ecmMxMot_VMoveStart_FF</p> <p style="margin: 0;">- 다축 연속속도 이송</p>	<b>INFORMATION</b>
	 Multiple Axes Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmMxMot\_VMoveStart([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_i32 DirList[], [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmMxMot\_VMoveStart\_FF([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_i32 DirList[], [out] t\_i32 \*ErrCode)

## DESCRIPTION

여러 개의 축에 대하여 Velocity Move 작업을 동시에 시작합니다. Velocity Move 는 작업속도까지 가속한 후에 작업속도를 유지하며 정지 함수가 호출될 때까지 지정한 방향으로의 모션을 계속 수행합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ NumAxes : 동시에 작업을 수행할 대상 축 개수
- ▶ AxisList : 동시에 작업을 수행할 대상 축 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다.
- ▶ DirList : 방향을 지시하는 값의 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다. 모션의 방향을 지시하는 값은 다음과 같습니다.

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 리턴값은 함수의 cmdidx 가 반환됩니다.

## EXAMPLE

---

```

C/C++
// 다음의 예제는 2 축을 “Jog 이동”을 하는 예입니다. 본 예제에서의
// “Jog 이동”은 버튼이 눌러진 상태에서는 Axis0,1 축의 이동을
// 수행하다가, 버튼이 풀리면 이동을 멈추는 예입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0
#define AXIS1 1

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

/*****
* MulitProgramInitial : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다. //5~9 장 사이 내용 참고
*****/
void MulitProgramInitial()
{
    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, AXIS0, &Error_Num);
    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXIS1, &Error_Num);

} MulitProgramInitial 함수 끝

/*****
* MulitMoveSetting : 이 함수는 다축 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 MulitMoveSetting ()
{
    ecmSxCfg_SetMioProp(Device_Info.NetIdx, AXIS0, propID, propID, &Error_Num)

```

---

---

```

    ecmSxCfg_SetUnitDist(Device_Info.NetIdx, AXIS0, UnitDist , &Error_Num);
    ecmSxCfg_SetUnitSpeed([Device_Info.NetIdx, AXIS0, UnitSpeed, &Error_Num);
    ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS0, 0, 0, 40000, 10000, 10000,
    &Error_Num );
    ecmSxCfg_SetJerkRatio (Device_Info.NetIdx, AXIS0, JerkTimeRatio, &Error_Num);

    ecmSxCfg_SetMioProp(Device_Info.NetIdx, AXIS1, propID, propID, &Error_Num)
    ecmSxCfg_SetUnitDist (Device_Info.NetIdx, AXIS1, UnitDist , &Error_Num);
    ecmSxCfg_SetUnitSpeed([Device_Info.NetIdx, AXIS1, UnitSpeed, &Error_Num);
    ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS1, 0, 0, 40000, 10000, 10000,
    &Error_Num );
    ecmSxCfg_SetJerkRatio (Device_Info.NetIdx, AXIS1, JerkTimeRatio, &Error_Num);
}

//*****
* MulitPlusButtonDown() : (+)Move 버튼이 눌렸을 때 호출되는 가상의 함수
* 이 함수에서 (+)방향으로 V-Move 를 시작합니다.
*****//
void MulitPlusButtonDown ()
{
    t_i32 AxisList[2] = {0,1};
    t_i32 DirList[2] = { ecmDIR_P, ecmDIR_P}

    ecmMxMot_VMoveStart(Device_Info.NetIdx, 2, AxisList DirList, &Error_Num)
    //Positive dir V-MOVE
}

//*****
* MulitPlusButtonUp() : (-)Move 버튼이 올라올 때 호출되는 가상의 함수
* 이 함수에서는 V-Move 를 종료합니다.
*****//
void MulitPlusButtonUp ()
{
    t_i32 AxisList[2] = {0,1};

    ecmMxMot_Stop(Device_Info.NetIdx, 2, AxisList , 0, 0, &Error_Num);
    // 두 축 정지
}

//*****
* MulitMinusButtonDown(): (-)Move 버튼이 눌렸을 때 호출되는 가상의 함수
* 이 함수에서 (+)방향으로 V-Move 를 시작합니다.
*****//
void MulitMinusButtonDown()
{
    t_i32 AxisList[2] = {0,1};
    t_i32 DirList[2] = { ecmDIR_N, ecmDIR_N}

    ecmMxMot_VMoveStart(Device_Info.NetIdx, 2, AxisList DirList, &Error_Num)
    //Negative dir V-MOVE
}

//*****
* MulitMinusButtonUp() : (-)Move 버튼이 올라올 때 호출되는 가상의 함수
* 이 함수에서는 V-Move 를 종료합니다.

```

---

```

*****//
void MultMinusButtonUp()
{
    ecmMxMot_Stop(Device_Info.NetIdx, 2, AxisList , 0, 0, &Error_Num);
    // 두 축 정지
}

```

NAME	INFORMATION
ecmMxMot_MoveStart	Multiple Axes Function
ecmMxMot_MoveStart_FF	VC++/BCB/.NET
- 다축 상대 좌표 이송	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmMxMot\_MoveStart([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_f64 DistList[], [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmMxMot\_MoveStart\_FF([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_f64 DistList[], [out] t\_i32 \*ErrCode)

## DESCRIPTION

여러 개의 축에 대하여 현재의 위치에서 지정한 거리만큼 이동을 동시에 시작합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다.,

## PARAMETER

- ▶ NetID : Network 번호
- ▶ NumAxes : 동시에 작업을 수행할 대상 축 개수
- ▶ AxisList : 동시에 작업을 수행할 대상 축 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다.
- ▶ DistList : 이동할 거리값의 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다. 이동 거리값은 현재의 위치에 대한 상대좌표이며 거리의 단위는 논리적 거리 단위를 사용합니다.
- ▶ 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 (+)5000 이동하는 예 입니다.
#include "ComiEcatSdk_Api.h"
#define AXIS0 0
#define AXIS1 1

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* MulitProgramInitial : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다. //5~9 장 사이 내용 참고
*****/
void MulitProgramInitial()
{
    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, AXIS0, &Error_Num);
    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXIS1, &Error_Num);

} OnProgramInitial 함수 끝

/*****
* MulitMoveSetting : 이 함수는 다축 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 MulitMoveSetting ()
{
    ecmSxCfg_SetMioProp(Device_Info.NetIdx, AXIS0, propID, propID, &Error_Num)
    ecmSxCfg_SetUnitDist( Device_Info.NetIdx, AXIS0, UnitDist , &Error_Num);
    ecmSxCfg_SetUnitSpeed([Device_Info.NetIdx, AXIS0, UnitSpeed, &Error_Num);
    ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS0, 0, 0, 40000, 10000, 10000,
    &Error_Num);

```

---



---

```

    ecmSxCfg_SetJerkRatio (Device_Info.NetIdx, AXIS0, JerkTimeRatio, &Error_Num);

    ecmSxCfg_SetMioProp(Device_Info.NetIdx, AXIS1, propID, propID, &Error_Num)
    ecmSxCfg_SetUnitDist (Device_Info.NetIdx, AXIS1, UnitDist , &Error_Num);
    ecmSxCfg_SetUnitSpeed ([Device_Info.NetIdx, AXIS1, UnitSpeed, &Error_Num);
    ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS1, 0, 0, 40000, 10000, 10000,
    &Error_Num);
    ecmSxCfg_SetJerkRatio (Device_Info.NetIdx, AXIS1, JerkTimeRatio, &Error_Num);
}

/*****
* DO_MOVE : 작업 명령 시에 호출되는 가상의 함수 입니다.
* cmsSetCfgSetSpeedPattern 으로 설정된 두개의 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 두 축을 상대 위치 만큼 +5000 이동
    t_i32 AxisList[2] = {0,1};
    t_f64 DistList[2] = {5000,5000};

    ecmMxMot_MoveStart(Device_Info.NetIdx, 2, AxisList, DistList, &Error_Num);
    //Move 두 축을 각각 상대 거리 +5000 이동

    if(!ecmMxSt_WaitCompt (Device_Info.NetIdx, 2, AxisList, &Error_Num)){
        //두 축이 +5000 만큼 이동 되는 동안 블로킹하는 함수
        // 모션 완료 대기 실패 예외 처리
        return;
    }
}

```

---

NAME	INFORMATION
ecmMxMot_MoveToStart ecmMxMot_MoveToStart_FF - 다축 절대 좌표 이송	Multiple Axes Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

□ t\_cmdidx ecmMxMot\_MoveToStart([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_f64 PosList[], [out] t\_i32 \*ErrCode)

□ t\_cmdidx ecmMxMot\_MoveToStart\_FF([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_f64 PosList[], [out] t\_i32 \*ErrCode)

## DESCRIPTION

여러 개의 축에 대하여 지정한 절대좌표로의 이동을 시작합니다. 이 함수를 사용하면 여러 개의 축이 동시에 작업을 시작합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ NumAxes : 동시에 작업을 수행할 대상 축 개수
- ▶ AxisList : 동시에 작업을 수행할 대상 축 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다.
- ▶ PosList : 절대좌표값의 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다. 절대좌표의 단위는 논리적 거리 단위를 사용합니다.
- ▶ 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

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

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

/*****
* DO_MOVETo : 작업 명령 시에 호출되는 가상의 함수 입니다.
* cmsSetCfgSetSpeedPattern 으로 설정된 두개의 축(Axis)의 속도모드를
* 그대로 유지 하면서 절대 좌표 +5000 으로 이동
*****/
void DO_MOVETo()
{
    //cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
    //그대로 유지 하면서 두 축을 상대 위치 만큼 +5000 이동
    t_i32 AxisList[2] = {0,1};
    t_f64 PosList[2] = {5000,5000};

    ecmMxMot_MoveToStart(Device_Info.NetIdx, 2, AxisList, PosList, &Error_Num);
    //Move 두 축을 각각 절대 좌표 5000 으로 이동

    if(!ecmMxSt_WaitCompt (Device_Info.NetIdx, 2, AxisList, &Error_Num)){
        //두 축이 절대 좌표 5000 으로 이동 되는 동안 블로킹하는 함수
        //모션 완료 대기 실패 예외 처리
        return;
    }
}

```

---

<h2>NAME</h2> <p>ecmMxMot_Stop</p> <p>ecmMxMot_Stop_FF</p> <p>- 다축 이송 정지</p>	<b>INFORMATION</b>
	 Multiple Axes Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmMxMot\_Stop([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_i32 IsDecStop, [in] t\_i32 IsWaitCompt, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmMxMot\_Stop\_FF([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [in] t\_i32 IsDecStop, [out] t\_i32 \*ErrCode)

## DESCRIPTION

지정한 모든 축에 대한 모션을 정지 합니다. IsDecStop 매개변수를 통해 감속정지, 급정지를 선택할 수 있으며, IsWaitCompt 매개변수를 통해 모션 완료시까지 대기할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ NumAxes : 동시에 작업을 수행할 대상 축 개수
- ▶ AxisList : 동시에 작업을 수행할 대상 축 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 합니다.
- ▶ 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

\* ecmMxMot\_VMoveStart EXAMPLE 참고

**NAME**

ecmMxSt\_IsBusy  
- 다축 모션 이송 확인

**INFORMATION**

-  Multiple Axes Function
-  VC++/BCB/.NET
-  Level 1
-  위험 요소 없음

**SYNOPSIS**

```
□ t_bool ecmMxSt_IsBusy([in] t_i32 NetID, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [out]
t_i32 *ErrCode)
```

**DESCRIPTION**

이 함수는 여러 축에 대하여 모션이 이송중인지를 확인합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ NumAxes : 동시에 작업을 수행할 대상 축 개수
- ▶ AxisList : 동시에 작업을 수행할 대상 축 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 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	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
#define AXIS1 1

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* 프로그램 앞 부분은 생략
*****/





/*****
* DO_Busy: 작업 명령 시에 호출되는 가상의 함수입니다.
*****/
void DO_Busy()
{
    //cmsSetCfgSetSpeedPattern 으로 설정된 첫번째 축(Axis)의 속도모드를
    //그대로 유지 하면서 두 축을 상대 위치 만큼 +5000 이동
    t_i32 AxisList[2] = {AXIS0, AXIS1};
    t_f64 PosList[2] = {5000,5000};

    ecmMxMot_MoveToStart(Device_Info.NetIdx, 2, AxisList, PosList, &Error_Num);
    //Move 두 축을 각각 절대 좌표 5000 으로 이동

    While(1)
    {
        if(!ecmMxSt_IsBusy(Device_Info.NetIdx, 2, AxisList, &Error_Num)
        {
            //이송 중이 아니면 break;
            break;
        }
    }
}

```

---

NAME	I N F O R M A T I O N
ecmMxSt_WaitCompt	 Multiple Axes Function
ecmMxSt_WaitCompt_NB	 VC++/BCB/.NET
- 다축 모션 완료 대기	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecmMxSt\_WaitCompt([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [out] t\_i32 \*ErrCode)
- t\_success ecmMxSt\_WaitCompt\_NB([in] t\_i32 NetID, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [out] t\_i32 \*ErrCode)

## DESCRIPTION

여러 축에 대하여 모션이 완료 될 때까지 기다립니다. ecmMxSt\_WaitCompt\_NB 는 완료 대기시 윈도우 메시지를 블로킹(Blocking) 하지 않습니다. 이 함수는 다축제어 뿐 아니라 원점복귀나 단축 모션제어 작업시에도 활용할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ NumAxes : 동시에 작업을 수행할 대상 축 개수
- ▶ AxisList : 동시에 작업을 수행할 대상 축 배열 주소값. 이 배열의 크기는 NumAxes 값과 일치하거나 커야 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	다축 모션이 완료되지 않았음을 의미합니다.
1	다축 모션이 완료되었음을 의미합니다.

## EXAMPLE



\* ecmMxMot\_MoveStart EXAMPLE 참고

# 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.



## 16 Motion Interpolation Functions

### 16.1 함수 요약

Summary of Functions
<p>□ 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) 보간 대상 축 그룹을 설정합니다</p>
<p>□ t_cmdidx ecmIxCfg_MapAxes_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 NumAxes, [in] t_i32 AxisList[], [in] t_i32 IxMode, [out] t_i32 *ErrCode) 보간 대상 축 그룹을 설정합니다</p>
<p>□ t_cmdidx ecmIxCfg_UnmapAxes([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode) 보간 대상 축 그룹을 해제합니다.</p>
<p>□ t_cmdidx ecmIxCfg_UnmapAxes_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode) 보간 대상 축 그룹을 해제합니다.</p>
<p>□ 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) 보간 이송 속도를 설정합니다. 보간 이송 속도는 마스터 속도모드와 벡터 속도모드를 설정 할 수 있습니다.</p>
<p>□ t_cmdidx ecmIxCfg_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) 보간 이송 속도를 설정합니다. 보간 이송 속도는 마스터 속도모드와 벡터 속도모드를 설정 할 수 있습니다.</p>
<p>□ 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) 설정된 보간 이송 속도를 반환합니다. 보간 이송 속도는 마스터 속도 혹은 벡터 모드에 해당하는 속도모드를 반환합니다.</p>
<p>□ t_cmdidx ecmIxCfg_SetJerkRatio([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode) 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>□ t_cmdidx ecmIxCfg_SetJerkRatio_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 JerkTimeRatio, [out] t_i32 *ErrCode) 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>□ t_f64 ecmIxCfg_GetJerkRatio([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode) 지정된 보간 맵의 현재 설정된 Jerk 시간 비율을 반환합니다.</p>
<p>□ t_cmdidx ecmIxCfg_SetSpeedPatt_MR([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 SpeedMode, [in] t_f64 MasterRatio, [out] t_i32 *ErrCode);</p>

<p>□ t_cmdidx ecmIxCfg_SetSpeedPatt_MR_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 SpeedMode, [in] t_f64 MasterRatio, [out] t_i32 *ErrCode);</p>
<p>□ t_success ecmIxCfg_GetSpeedPatt_MR([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *SpeedMode, [out] t_f64 *MasterRatio, [out] t_i32 *ErrCode);</p>
<p>□ t_cmdidx ecmIxCfg_SetMastWeight([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 *MastWeightList, [out] t_i32 *ErrCode);</p>
<p>□ t_success ecmIxCfg_GetMastWeight([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_f64 *MastWeightList, [out] t_i32 *ErrCode);</p>
<p>□ t_success ecmIxCfg_Spline_SetTrgOutEnv([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_bool IsEnableTrgOut, [in] t_i16 TrgOfsTime_ms, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecmIxCfg_Spline_GetTrgOutEnv([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [out] t_bool* IsEnableTrgOut, [out] t_i16* TrgOfsTime_ms, [out] t_i32 *ErrCode)</p>
<p>□ t_f64 ecmIxCfg_Spline_GetTrgOutPos([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_i32 Axis, [in] t_i32 PointIndex, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecmIxCfg_Spline_SetJsType([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_i32 JerkSmoothType, [out] t_i32 *ErrCode);</p>
<p>□ t_i32 ecmIxCfg_Spline_GetJsType([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecmIxCfg_Spline_SetJsProp_F([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_i32 PropId, [in] t_f64 PropVal, [out] t_i32 *ErrCode)</p>
<p>□ t_f64 ecmIxCfg_Spline_GetJsProp_F([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_i32 PropId, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecmIxCfg_Spline_SetJerkThresh([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_i32 Axis, [in] t_f64 JerkThreshold, [out] t_i32 *ErrCode)</p>
<p>□ t_f64 ecmIxCfg_Spline_GetJerkThresh([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_i32 Axis, [out] t_i32 *ErrCode)</p>
<p>□ t_cmdidx ecmIxMot_LineStart([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode) 보간 제어에 있어 직선 보간을 수행하며, 상대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmIxMot_LineStart_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode) 보간 제어에 있어 직선 보간을 수행하며, 상대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxMot_Line([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode) 보간 제어에 있어 직선 보간을 수행하며, 상대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxMot_Line_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 DistList[], [out] t_i32 *ErrCode) 보간 제어에 있어 직선 보간을 수행하며, 상대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>

<p>□ t_cmdidx ecmIxmMot_LineToStart([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)                  보간 제어에 있어 직선 보간을 수행하며, 절대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환됩니다.</p>
<p>□ t_cmdidx ecmIxmMot_LineToStart_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)                  보간 제어에 있어 직선 보간을 수행하며, 절대 좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환됩니다.</p>
<p>□ t_success ecmIxmMot_LineTo([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)                  보간 제어에 있어 직선 보간을 수행하며, 절대좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxmMot_LineTo_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 PosList[], [out] t_i32 *ErrCode)                  보간 제어에 있어 직선 보간을 수행하며, 절대좌표 이송을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxmMot_ArcAng_R_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmIxmMot_ArcAng_R_Start_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxmMot_ArcAng_R([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxmMot_ArcAng_R_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxmMot_ArcAng_A_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmIxmMot_ArcAng_A_Start_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxmMot_ArcAng_A([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>





<p>□ t_success ecmIxMot_ArcAng_A_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 각도를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxMot_ArcPos_R_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 XEndOffset, [in] t_f64 YEndOffset, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 종점 좌표를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmIxMot_ArcPos_R_Start_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 XEndOffset, [in] t_f64 YEndOffset, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 종점 좌표를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxMot_ArcPos_R([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 XEndOffset, [in] t_f64 YEndOffset, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 종점 좌표를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxMot_ArcPos_R_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentOffset, [in] t_f64 YCentOffset, [in] t_f64 XEndOffset, [in] t_f64 YEndOffset, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 상대적 중심 좌표와 종점 좌표를 통해 원호 보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxMot_ArcPos_A_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 XEndPos, [in] t_f64 YEndPos, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 종점 좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmIxMot_ArcPos_A_Start_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 XEndPos, [in] t_f64 YEndPos, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 종점 좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxMot_ArcPos_A([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 XEndPos, [in] t_f64 YEndPos, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 종점 좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxMot_ArcPos_A_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 XCentPos, [in] t_f64 YCentPos, [in] t_f64 XEndPos, [in] t_f64 YEndPos, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 절대적 중심 좌표와 종점 좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxMot_Arc3P_R_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XOffset, [in] t_f64 P2_YOffset, [in] t_f64 P3_XOffset, [in] t_f64 P3_YOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)                  보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 상대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>

<p>□ t_cmdidx ecmIxmot_Arc3P_R_Start_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XOffset, [in] t_f64 P2_YOffset, [in] t_f64 P3_XOffset, [in] t_f64 P3_YOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 상대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxmot_Arc3P_R([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XOffset, [in] t_f64 P2_YOffset, [in] t_f64 P3_XOffset, [in] t_f64 P3_YOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 상대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxmot_Arc3P_R_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XOffset, [in] t_f64 P2_YOffset, [in] t_f64 P3_XOffset, [in] t_f64 P3_YOffset, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 상대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxmot_Arc3P_A_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XPos, [in] t_f64 P2_YPos, [in] t_f64 P3_XPos, [in] t_f64 P3_YPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 절대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_cmdidx ecmIxmot_Arc3P_A_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XPos, [in] t_f64 P2_YPos, [in] t_f64 P3_XPos, [in] t_f64 P3_YPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 절대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환 됩니다.</p>
<p>□ t_success ecmIxmot_Arc3P_A([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XPos, [in] t_f64 P2_YPos, [in] t_f64 P3_XPos, [in] t_f64 P3_YPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 절대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_success ecmIxmot_Arc3P_A_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_f64 P2_XPos, [in] t_f64 P2_YPos, [in] t_f64 P3_XPos, [in] t_f64 P3_YPos, [in] t_f64 EndAngle, [out] t_i32 *ErrCode)</p> <p>보간 제어에 있어 원호보간을 수행하며, 현재 위치와 두개의 절대좌표를 통해 원호보간을 수행합니다. 이 구동 함수는 구동 시작 후 바로 반환되지 않습니다.</p>
<p>□ t_cmdidx ecmIxmot_Spline_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hSplineObj, [in] t_bool IsReverseDir, [out] t_i32 *ErrCode)</p>
<p>□ t_cmdidx ecmIxmot_MPRLin2X_Start([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_handle32 hMPRLin2XObj, [in] t_bool IsReverseDir, [out] t_i32 *ErrCode);</p>
<p>□ t_cmdidx ecmIxmot_MoveVia2X_Start([in] t_i32 NetID, [in] t_ui16 MapIndex, [in] t_f64 P2[], [in] t_f64 P3[], [in] t_bool IsAbsPosMode, [in] t_i32 RoundPosType, [in] t_f64 NormRadius, [in] t_f64 MinRadius, [out] t_i32 *ErrCode);</p>
<p>□ t_cmdidx ecmIxmot_Stop([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode);</p>
<p>□ t_cmdidx ecmIxmot_Stop_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [out] t_i32 *ErrCode);</p>

<p>□ t_cmdidx ecmIxMot_Stop([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [in] t_i32 IsWaitCompt, [out] t_i32 *ErrCode)          보간 제어 구동 이송을 정지 합니다.</p>
<p>□ t_cmdidx ecmIxMot_Stop_FF([in] t_i32 NetID, [in] t_i32 MapIndex, [in] t_i32 IsDecStop, [out] t_i32 *ErrCode)          보간 제어 구동 이송을 정지 합니다.</p>
<p>□ t_bool ecmIxSt_IsBusy([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)          보간 제어 구동 이송이 이송중인지 확인합니다</p>
<p>□ t_success ecmIxSt_WaitCompt([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)          보간 제어 구동 이송이 완료 될 때까지 대기합니다.</p>
<p>□ t_success ecmIxSt_WaitCompt_NB([in] t_i32 NetID, [in] t_i32 MapIndex, [out] t_i32 *ErrCode)          보간 제어 구동 이송이 완료 될 때까지 대기합니다.</p>



**16.2** 함수 설명

NAME	I N F O R M A T I O N
ecmlxCfg_MapAxes	 Interpolation Function
ecmlxCfg_MapAxes_FF	 VC++/BCB/.NET
- 보간 대상 축 그룹 설정	 Level 1
	 위험 요소 없음

**SYNOPSIS**

- t\_cmdidx ecmIxCfg\_MapAxes([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_i32 NumAxes, [in] t\_i32 AxisList[], [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxCfg\_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 ~ Bit31을 이용하여 그룹에 포함할 축들을 지정합니다. 각 비트의 값이 0이면 해당 축(비트의 순서와 일치하는 축)은 배제되는 것이며, 1이면 해당 축이 포함되는 것이다. 이 매개변수의 각 비트별 정보는 아래 표와 같다.(아래 표는 8축을 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 32 축 이상을 사용하고자 할 때 사용하는 매개변수이다. MapMask1 에 포함되지 못한 축들은 이 매개변수를 사용하여 지정한다.

▶ 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  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

□ t\_cmdidx ecmIxCfg\_UnmapAxes([in] t\_i32 NetID, [in] t\_i32 MapIndex, [out] t\_i32 \*ErrCode)

□ t\_cmdidx ecmIxCfg\_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.

<h2>NAME</h2> <p>ecmIxCfg_SetSpeedPatt                  ecmIxCfg_SetSpeedPatt_FF                  ecmIxCfg_GetSpeedPatt                  - 보간 이송 속도 설정 및 반환</p>	<h2>INFORMATION</h2> <p> Interpolation Function</p> <p> VC++/BCB/.NET</p> <p> Level 1</p> <p> 위험 요소 없음</p>
---	--

## SYNOPSIS

- 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)
- t\_cmdidx ecmIxCfg\_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 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)

## DESCRIPTION

ecmIxCfg\_SetSpeedPatt, ecmIxCfg\_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. ecmIxCfg\_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 값 입니다.

▶ VEnd : 이송 완료 시 속도에 대한 설정 및 반환 값 입니다.

▶ VWork : 작업 속도에 대한 설정 및 반환 값 입니다.

▶ Acc : 가속도에 대한 설정 및 반환 값 입니다.

▶ Dec : 감속도에 대한 설정 및 반환 값 입니다.

▶ 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

\* ecmIxBot\_LineStart EXAMPLE 참고

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

## SYNOPSIS

- t\_cmdidx ecmIxCfg\_SetJerkRatio([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 JerkTimeRatio, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxCfg\_SetJerkRatio\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 JerkTimeRatio, [out] t\_i32 \*ErrCode)
- t\_f64 ecmIxCfg\_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.

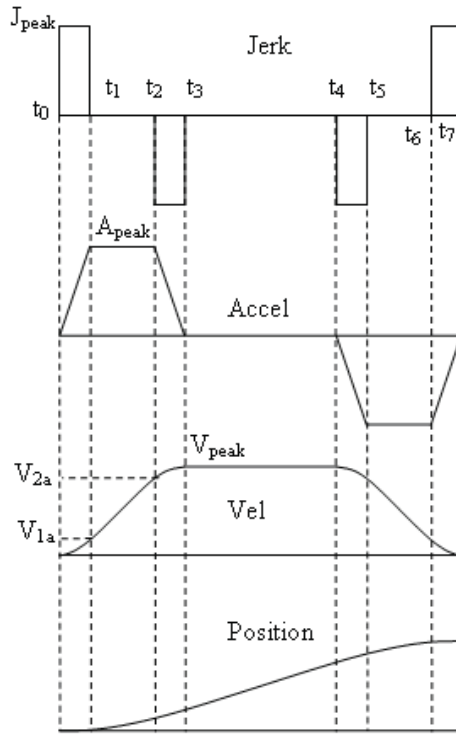
## RETURN VALUE

\* ecmIxCfg\_SetJerkRatio(), ecmIxCfg\_SetJerkRatio\_FF() : 이 리턴값은 함수의 cmdidx 를 반환합니다.

\* ecmIxCfg\_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 빼 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


ecmIxCfg\_SetSpeedPatt\_MR  
 ecmIxCfg\_SetSpeedPatt\_MR\_FF  
 ecmIxCfg\_GetSpeedPatt\_MR  
 - 보간 이송 속도 설정 및 반환

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxCfg\_SetSpeedPatt\_MR([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_i32 SpeedMode, [in] t\_f64 MasterRatio, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxCfg\_SetSpeedPatt\_MR\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_i32 SpeedMode, [in] t\_f64 MasterRatio, [out] t\_i32 \*ErrCode)
- t\_success ecmIxCfg\_GetSpeedPatt\_MR([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_i32 \*SpeedMode, [out] t\_f64 \*MasterRatio, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 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.

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

## RETURN VALUE

\* ecmIxCfg\_SetSpeedPatt\_MR(), ecmIxCfg\_SetSpeedPatt()\_MR\_FF : 이 리턴값은 함수의 cmdidx 를 반환합니다.



\* ecmIxCfg\_GetSpeedPatt\_MR() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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


## EXAMPLE


**NAME**


ecmIxCfg\_Spline\_ClearPool  
- 스플라인 보간 오브젝트 해제

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmIxCfg\_Spline\_ClearPool([in] t\_i32 NetID, [in] t\_i32 MapIndex, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

지정된 맵에 생성했던 스플라인 오브젝트를 삭제합니다. 이 함수를 수행하지 않고 ecmIxCfg\_Spline\_AddNewObj 함수를 반복 수행하면 마스터장치내의 메모리 부족현상이 일어날 수 있으므로 이 함수를 적절히 사용하여야 합니다.

**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	오브젝트 해제에 실패 했음을 의미합니다.
1	오브젝트 해제에 성공 했음을 의미합니다.


**EXAMPLE**


\* ecmIxCfg\_Spline\_Start EXAMPLE 참고


## NAME


ecmIxCfg\_Spline\_AddNewObj  
- 스플라인 보간 오브젝트 생성

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_handle ecmIxCfg\_Spline\_AddNewObj([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_ui32 NumRefPoints, [in] t\_bool IsAbsPosMode, [out] t\_i32 \*ErrCode)

## DESCRIPTION

지정된 보간맵에 스플라인 오브젝트를 생성합니다. 여기서 생성한 오브젝트 핸들값으로 스플라인 보간을 제어합니다. 하나의 보간 맵 Pool 에 여러 개의 오브젝트를 등록하여 사용할 수 있습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ NumRefPoints : 등록할 Reference 좌표 개수
- ▶ IsAbsPosMode

Value	Meaning
0	Reference 좌표가 상대좌표로 등록된다.
1	Reference 좌표가 절대좌표로 등록된다.

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

## RETURN VALUE

\* 이 리턴값은 오브젝트의 핸들값이 리턴됩니다..

## EXAMPLE


\* ecmIxCfg\_Spline\_Start EXAMPLE 참고


**NAME**


ecmIxCfg\_Spline\_SetRefPoint

- 스플라인 보간 좌표 설정

**INFORMATION**
 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

```

❑ t_success ecmIxCfg_Spline_SetRefPoint([in] t_i32 NetID, [in] t_i32 MapIndex, [in]
t_handle32 hSplineObj, [in] t_i32 PointIndex , [in] t_f64 Point[],[in] t_i32 NumPointData,
[out] t_i32 *ErrCode)

```

**DESCRIPTION**

이 함수는 스플라인 보간으로 이송하고자 하는 기준 포인트들의 좌표를 설정합니다. 이 함수는 ecmIxCfg\_Spline\_AddNewObj 함수에서 설정한 NumRefPoint 만큼 수행하여야 합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hSplineObj : 스플라인 보간 오브젝트 생성시 반환된 오브젝트 핸들값
- ▶ PointIndex : 포인트 순서 번호
- ▶ Point : 등록된 좌표 배열
- ▶ NumPointData : 좌표 데이터 개수
- ▶ 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	좌표 설정에 실패 했음을 의미합니다.
1	좌표 설정에 성공 했음을 의미합니다.

**EXAMPLE**


\* ecmIxMot\_Spline\_Start EXAMPLE 참고

**NAME**


ecmIxCfg\_Spline\_BuildObj  
- 스플라인 보간 오브젝트 빌드

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmIxCfg\_Spline\_BuildObj([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

지정된 보간맵의 스플라인 오브젝트를 빌드합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hSplineObj : 스플라인 보간 오브젝트 생성시 반환된 오브젝트 핸들값
- ▶ 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	빌드에 실패 했음을 의미합니다.
1	빌드에 성공 했음을 의미합니다.


**EXAMPLE**


\* ecmIxBot\_Spline\_Start EXAMPLE 참고

**NAME**


ecmIxCfg\_MPRLin2X\_ClearPool  
 - 2 축 Multi Point Rounded 보간 오브젝트 해제

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmIxCfg\_MPRLin2X\_ClearPool([in] t\_i32 NetID, [in] t\_i32 MapIndex, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

지정된 맵에 생성했던 MPRLin2X 오브젝트를 삭제합니다. 이 함수를 수행하지 않고 ecmIxCfg\_MPRLin2X\_AddNewObj 함수를 반복 수행하면 마스터장치내의 메모리 부족현상이 일어날 수 있으므로 이 함수를 적절히 사용하여야 합니다.

**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	오브젝트 해제에 실패 했음을 의미합니다.
1	오브젝트 해제에 성공 했음을 의미합니다.

**EXAMPLE**


\* ecmIxCfg\_MPRLin2X\_Start EXAMPLE 참고


**NAME**


ecmIxCfg\_MPRLin2X\_AddNewObj  
 - 2 축 Multi Point Rounded 보간 오브젝트 생성

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_handle ecmIxCfg\_MPRLin2X\_AddNewObj([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_ui32 NumRefPoints, [in] t\_bool IsAbsPosMode, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

지정된 보간맵에 MPRLin2X 오브젝트를 생성합니다. 여기서 생성한 오브젝트 핸들값으로 MPRLin2X 보간을 제어합니다. 하나의 보간 맵 Pool 에 여러 개의 오브젝트를 등록하여 사용할 수 있습니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ NumRefPoints : 등록할 Reference 좌표 개수
- ▶ IsAbsPosMode

Value	Meaning
0	Reference 좌표가 상대좌표로 등록된다.
1	Reference 좌표가 절대좌표로 등록된다.

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

**RETURN VALUE**

- \* 이 리턴값은 오브젝트의 핸들값이 리턴됩니다..

**EXAMPLE**

- \* ecmIxCfg\_MPRLin2X\_Start EXAMPLE 참고



<h2 style="margin: 0;">NAME</h2> <p style="margin: 10px 0 0 20px;">ecmIxCfg_MPRLin2X_SetRefPoint</p> <p style="margin: 0 0 0 20px;">- 2 축 Multi Point Rounded 보간 좌표 설정</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;"> Interpolation 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		Interpolation Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Interpolation Function											
VC++/BCB/.NET											
Level 1											
위험 요소 없음											

## SYNOPSIS

t\_success ecmIxCfg\_MPRLin2X\_SetRefPoint([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hMPRLin2XObj, [in] t\_i32 PointIndex , [in] t\_f64 Point[], [in] t\_64 RoundData, [in] t\_i32 RoundDataType, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 MPRLin2X 보간으로 이송하고자 하는 기준 포인트들의 좌표를 설정합니다. 이 함수는 ecmIxCfg\_MPRLin2X\_AddNewObj 함수에서 설정한 NumRefPoint 만큼 수행하여야 합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hMPRLin2XObj : 스플라인 보간 오브젝트 생성시 반환된 오브젝트 핸들값
- ▶ PointIndex : 포인트 순서 번호
- ▶ Point : 등록된 좌표 배열
- ▶ RoundData : 해당 포인트에 라운드를 삽입하고자 하는 경우 라운드의 크기 정보를 설정. 이 값은 RoundDataType 에 따라서 다릅니다.
- ▶ RoundData : 해당 포인트에 삽입되는 라운드 형식을 설정

Value	Meaning
0 or ecmROUND_DT_NONE	라운드를 삽입하지 않는다.
1 or ecmROUND_DT_RADIUS	라운드를 삽입하며, RoundData 정보가 라운드의 반지름을 나타냄.
2 or ecmROUND_DT_OFFSET	라운드를 삽입하며, RoundData 정보가 라운드가 시작되고 종료되는 위치를 해당 포인트에서 OffSet 값을 나타냄.

▶ **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	좌표 설정에 실패 했음을 의미합니다.
1	좌표 설정에 성공 했음을 의미합니다.


### EXAMPLE


\* `ecmIxMot_MPRLin2X_Start` EXAMPLE 참고


## NAME


ecmIxCfg\_MPRLin2X\_BuildObj  
 - 2 축 Multi Point Rounded 보간 오브젝트 빌드

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmIxCfg\_MPRLin2X\_BuildObj([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hMPRLin2XObj, [out] t\_i32 \*ErrCode)

## DESCRIPTION

지정된 보간맵의 MPRLin2X 오브젝트를 빌드합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hMPRLin2XObj : MPRLin2X 보간 오브젝트 생성시 반환된 오브젝트 핸들값
- ▶ 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	빌드에 실패 했음을 의미합니다.
1	빌드에 성공 했음을 의미합니다.

## EXAMPLE

\* ecmIxCfg\_MPRLin2X\_Start EXAMPLE 참고

## NAME

ecmIxMot\_LineStart

ecmIxMot\_LineStart\_FF


ecmIxMot\_Line


ecmIxMot\_Line\_NB


- 직선 보간 상대 좌표 이송

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxMot\_LineStart([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 DistList[], [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxMot\_LineStart\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 DistList[], [out] t\_i32 \*ErrCode)
- t\_success ecmIxMot\_Line([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 DistList[], [out] t\_i32 \*ErrCode)
- t\_success ecmIxMot\_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.  
 ecmIxMot\_LineStart, ecmIxMot\_LineStart\_FF 는 모션을 시작시킨 후에 바로 반환되고,  
 ecmIxMot\_Line, ecmIxMot\_Line\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다.  
 그리고 ecmIxMot\_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 함수입니다.
*****/

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 함수입니다.
* 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


ecmIxMot\_LineToStart  
 ecmIxMot\_LineToStart\_FF  
 ecmIxMot\_LineTo  
 ecmIxMot\_LineTo\_NB  
 - 직선 보간 절대 좌표 이송

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxMot\_LineToStart([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 PosList[], [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxMot\_LineToStart\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 PosList[], [out] t\_i32 \*ErrCode)
- t\_success ecmIxMot\_LineTo ([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 PosList[], [out] t\_i32 \*ErrCode)
- t\_success ecmIxMot\_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.  
 ecmIxMot\_LineToStart, ecmIxMot\_LineToStart\_FF 는 모션을 시작시킨 후에 바로 반환되고,  
 ecmIxMot\_LineTo, ecmIxMot\_LineTo\_NB 는 모션이 완료되기 전까지 함수가 반환되지  
 않습니다. 그리고 ecmIxMot\_LineTo\_NB 는 함수가 반환되지 않지만 윈도우 메시지를  
 블로킹(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


ecmIxmot\_ArcAng\_R\_Start  
 ecmIxmot\_ArcAng\_R\_Start\_FF  
 ecmIxmot\_ArcAng\_R  
 ecmIxmot\_ArcAng\_R\_NB  
 - 원호 보간 상대 좌표 이송  
 (상대적 중심 좌표와 각도)

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxmot\_ArcAng\_R\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxmot\_ArcAng\_R\_Start\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_success ecmIxmot\_ArcAng\_R([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_success ecmIxmot\_ArcAng\_R\_NB([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

## DESCRIPTION

중심좌표와 원호의 각도를 매개변수로 하여 원호보간 이송을 수행합니다. 이때 중심좌표는 상대 좌표로 표현됩니다.

ecmIxmot\_ArcAng\_R\_Start, ecmIxmot\_ArcAng\_R\_Start\_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmIxmot\_ArcAng\_R, ecmIxmot\_ArcAng\_R\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmIxmot\_ArcAng\_R\_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블록킹(Blocking) 하지 않습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ XCentOffset : 현재 위치(시작위치)로부터 원의 중심까지 X 축 상대좌표값. 거리의 단위는 논리적 거리를 적용합니다.

▶ YCentOffset : 현재 위치(시작위치)로부터 원의 중심까지 Y 축 상대좌표값. 거리의 단위는 논리적 거리를 적용합니다.

▶ EndAngle : 원호보간 이동을 완료할 목표지점의 현재 위치에 대한 각도값을 Degree 값으로 지정합니다. 각도의 부호가 (+) 이면 반시계방향, (-) 이면 시계방향으로의 이동을 의미합니다.

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

RETURN VALUE

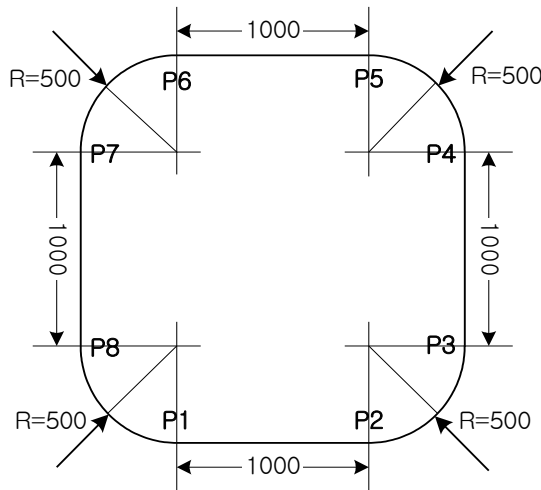
\* ecmIxMot\_ArcAng\_R\_Start(), ecmIxMot\_ArcAng\_R\_Start\_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

\* ecmIxMot\_ArcAng\_R (), ecmIxMot\_ArcAng\_R\_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

EXAMPLE

본 예제는 아래 그림과 같이 직선보간 이동과 원호보간 이동을 조합하는 Coordinated Motion 을 수행하는 예제입니다. P1 점으로부터 출발하여 P8 점을 거쳐 다시 P1 으로 복귀하는 작업입니다.



```

C/C++
C/C++

#include "ComiEcatSdk_Api.h"

#define AXISX      0
#define AXISY 1

#define MAP0      0
    
```

---

```

#define MAP1      1

t_i32 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);
    ecmIxCfg_MapAxes (Device_Info.NetIdx , MAP0, AXISX | AXISY, 0, ecmIX_MODE_LINEAR ,
    &Error_Num);

    ecmIxCfg_SetSpeedPatt( Device_Info.NetIdx , MAP0, 0, 0, 0, 40000, 10000, 10000,
    &Error_Num);
    ecmIxCfg_SetSpeedPatt( Device_Info.NetIdx , MAP1, 0, 0, 0, 40000, 10000, 10000,
    &Error_Num);

    //보간 그룹에 축을 설정하고
    //MAP0 은 직선 보간 모드 MAP1 은 원호 보간 모드로 설정

}
/*****

```

---

---

```

* OnDoMotion() : 작업명령시에 호출되는 가상의 함수
*****/
void OnDoMotion()
{
    double fDistList[2];

    // Move from P1 to P2 //
    fDistList[0]=1000; fDistList[1]=0;
    ecmIxMot_LineStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P2 to P3 //
    ecmIxMot_ArcAng_R_Start (Device_Info.NetIdx, MAP1, 0, 500, 90, &Error_Num);

    // Move from P3 to P4 //
    fDistList[0]=0; fDistList[1]=1000;
    ecmIxMot_LineStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P4 to P5 //
    ecmIxMot_ArcAng_R_Start (Device_Info.NetIdx, MAP1, -500, 0, 90, &Error_Num);

    // Move from P5 to P6 //
    fDistList[0]=-1000; fDistList[1]=0;
    ecmIxMot_LineStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P6 to P7 //
    ecmIxMot_ArcAng_R_Start Device_Info.NetIdx, MAP1, 0, -500, 90, &Error_Num);

    // Move from P7 to P8 //
    fDistList[0]=0; fDistList[1]=-1000;
    ecmIxMot_LineStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P8 to P1 //
    ecmIxMot_ArcAng_R_Start (Device_Info.NetIdx, MAP1, 500, 0, 90, &Error_Num);
}

```


---

## NAME


ecmIxMot\_ArcAng\_A\_Start  
 ecmIxMot\_ArcAng\_A\_Start\_FF  
 ecmIxMot\_ArcAng\_A  
 ecmIxMot\_ArcAng\_A\_NB  
 - 원호 보간 절대 좌표 이송  
 (절대적 중심 좌표와 각도)

### INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxMot\_ArcAng\_A\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxMot\_ArcAng\_A\_Start\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_success ecmIxMot\_ArcAng\_A([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_success ecmIxMot\_ArcAng\_A\_NB([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

## DESCRIPTION

중심좌표와 원호의 각도를 매개변수로 하여 원호보간 이송을 수행합니다. 이때 중심좌표는 절대 좌표로 표현됩니다.

ecmIxMot\_ArcAng\_A\_Start, ecmIxMot\_ArcAng\_A\_Start\_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmIxMot\_ArcAng\_A, ecmIxMot\_ArcAng\_A\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmIxMot\_ArcAng\_A\_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블록킹(Blocking) 하지 않습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ XCentPos : 중심점의 X 축 절대좌표. 좌표의 단위는 논리적 거리를 적용합니다.,
- ▶ YCentPos : 중심점의 Y 축 절대좌표. 좌표의 단위는 논리적 거리를 적용합니다.,

▶ **EndAngle** : 원호보간 이동을 완료할 목표지점의 현재 위치에 대한 각도값을 Degree 값으로 지정합니다. 각도의 부호가 (+) 이면 반시계방향, (-) 이면 시계방향으로의 이동을 의미합니다.

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

**RETURN VALUE**

\* `ecmIxMot_ArcAng_A_Start()`, `ecmIxMot_ArcAng_A_Start_FF()` : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

\* `ecmIxMot_ArcAng_A ()`, `ecmIxMot_ArcAng_A_NB()` : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

**EXAMPLE**


\* `ecmIxMot_ArcAng_R_Start` EXAMPLE 과 동일 상대, 절대만 차이

## NAME


ecmIxMot\_ArcPos\_R\_Start  
 ecmIxMot\_ArcPos\_R\_Start\_FF  
 ecmIxMot\_ArcPos\_R  
 ecmIxMot\_ArcPos\_R\_NB  
 - 원호 보간 상대 좌표 이송  
 (상대적 중심 좌표와 종점 좌표)

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_cmdidx ecmIxMot\_ArcPos\_R\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 XEndOffset, [in] t\_f64 YEndOffset, [out] t\_i32 \*ErrCode)

❑ t\_cmdidx ecmIxMot\_ArcPos\_R\_Start\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 XEndOffset, [in] t\_f64 YEndOffset, [out] t\_i32 \*ErrCode)

❑ t\_success ecmIxMot\_ArcPos\_R([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 XEndOffset, [in] t\_f64 YEndOffset, [out] t\_i32 \*ErrCode)

❑ t\_success ecmIxMot\_ArcPos\_R\_NB([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentOffset, [in] t\_f64 YCentOffset, [in] t\_f64 XEndOffset, [in] t\_f64 YEndOffset, [out] t\_i32 \*ErrCode)

## DESCRIPTION

중심좌표와 종점좌표를 매개변수로 하여 원호보간 이송을 수행합니다. 이때 중심좌표는 상대 좌표로 표현됩니다.

ecmIxMot\_ArcPos\_R\_Start, ecmIxMot\_ArcPos\_R\_Start\_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmIxMot\_ArcPos\_R, ecmIxMot\_ArcPos\_R\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmIxMot\_ArcPos\_R\_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블록킹(Blocking) 하지 않습니다.

## PARAMETER

▶ NetID : Network 번호

- ▶ MapIndex : 맵 번호
- ▶ XCentOffset : 현재 위치(시작위치)로부터 원의 중심까지 X 축 상대좌표값. 거리의 단위는 논리적 거리를 적용합니다.
- ▶ YCentOffset : 현재 위치(시작위치)로부터 원의 중심까지 Y 축 상대좌표값. 거리의 단위는 논리적 거리를 적용합니다.
- ▶ XEndOffset : 원호보간 이동을 완료할 목표지점의 현재 위치로부터 X 축상 거리값
- ▶ YEndOffset : 원호보간 이동을 완료할 목표지점의 현재 위치로부터 Y 축상 거리값
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

- \* ecmIxMot\_ArcPos\_R\_Start(), ecmIxMot\_ArcPos\_R\_Start\_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.
- \* ecmIxMot\_ArcPos\_R (), ecmIxMot\_ArcPos\_R\_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

EXAMPLE

- \* ecmIxMot\_ArcPos\_A\_Start 과 동일 상대, 절대만 차이



NAME	INFORMATION
ecmlxMot_ArcPos_A_Start ecmlxMot_ArcPos_A_Start_FF ecmlxMot_ArcPos_A ecmlxMot_ArcPos_A_NB - 원호 보간 절대 좌표 이송 (절대적 중심 좌표와 종점 좌표)	Interpolation Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmlxMot\_ArcPos\_A\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 XEndPos, [in] t\_f64 YEndPos, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmlxMot\_ArcPos\_A\_Start\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 XEndPos, [in] t\_f64 YEndPos, [out] t\_i32 \*ErrCode)
- t\_success ecmlxMot\_ArcPos\_A([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 XEndPos, [in] t\_f64 YEndPos, [out] t\_i32 \*ErrCode)
- t\_success ecmlxMot\_ArcPos\_A\_NB([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 XCentPos, [in] t\_f64 YCentPos, [in] t\_f64 XEndPos, [in] t\_f64 YEndPos, [out] t\_i32 \*ErrCode)

## DESCRIPTION

중심좌표와 종점좌표를 매개변수로 하여 원호보간 이송을 수행합니다. 이때 중심좌표는 절대 좌표로 표현됩니다.

ecmlxMot\_ArcPos\_A\_Start, ecmlxMot\_ArcPos\_A\_Start\_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmlxMot\_ArcPos\_A, ecmlxMot\_ArcPos\_A\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmlxMot\_ArcPos\_A\_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블로킹(Blocking) 하지 않습니다.

## PARAMETER

- ▶ NetID : Network 번호

- ▶ MapIndex : 맵 번호
- ▶ XCentPos : 중심점의 X 축 절대좌표 값
- ▶ YCentPos : 중심점의 Y 축 절대좌표 값
- ▶ XEndPos : 원호보간 이동을 완료할 목표지점(End Point)의 X 축 절대좌표 값
- ▶ YEndPos : 원호보간 이동을 완료할 목표지점(End Point)의 Y 축 절대좌표 값
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

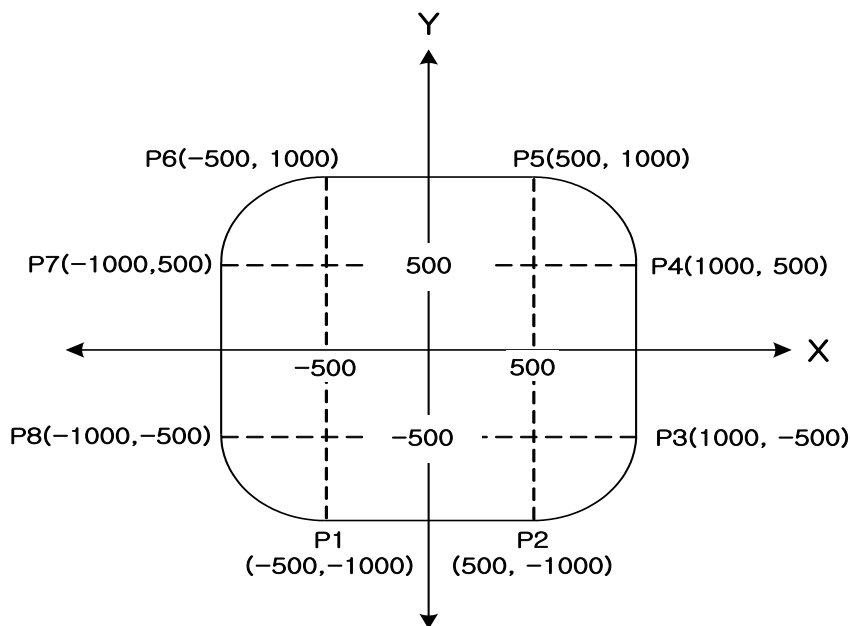
\* ecmIxMot\_ArcPos\_A\_Start(), ecmIxMot\_ArcPos\_A\_Start\_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

\* ecmIxMot\_ArcPos\_A(), ecmIxMot\_ArcPos\_A\_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

EXAMPLE

본 예제는 아래 그림과 같이 직선보간 이동과 원호보간 이동을 조합하는 Coordinated Motion 을 수행하는 예제입니다. P1 점으로부터 출발하여 P8 점을 거쳐 다시 P1 으로 복귀하는 작업입니다. 그리고 현재 위치가 P1 의 위치에 있다고 가정합니다.



---

```

C/C++
C/C++

#include "ComiEcatSdk_Api.h"

#define AXISX      0
#define AXISY 1

#define MAP0      0
#define MAP1      1

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

    ecmIxCfg_SetSpeedPatt( Device_Info.NetIdx , MAP0, 0, 0, 0, 40000, 10000, 10000,
    &Error_Num);
}

```

---

---

```

    ecmIxCfg_SetSpeedPatt( Device_Info.NetIdx , MAP1, 0, 0, 0, 40000, 10000, 10000,
    &Error_Num);

    //보간 그룹에 축을 설정하고
    //MAP0 은 직선 보간 모드 MAP1 은 원호 보간 모드로 설정

}
/*****
* OnDoMotion() : 작업명령시에 호출되는 가상의 함수
*****/
void OnDoMotion()
{
    double fDistList[2];

    // Move from P1 to P2 //
    fDistList[0]=1000; fDistList[1]=500;
    ecmIxMot_LineToStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P2 to P3 //
    ecmIxMot_ArcPos_A_Start (Device_Info.NetIdx, MAP1, 500, -500, 1000, -500,
    &Error_Num);

    // Move from P3 to P4 //
    fDistList[0]=1000; fDistList[1]=500;
    ecmIxMot_LineToStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P4 to P5 //
    ecmIxMot_ArcPos_A_Start (Device_Info.NetIdx, MAP1, 500, 500, 500, 1000,
    &Error_Num);

    // Move from P5 to P6 //
    fDistList[0]=-500; fDistList[1]=1000;
    ecmIxMot_LineToStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P6 to P7 //
    ecmIxMot_ArcPos_A_Start (Device_Info.NetIdx, MAP1, -500, 500, -1000, 500,
    &Error_Num);

    // Move from P7 to P8 //
    fDistList[0]=-1000; fDistList[1]=-500;
    ecmIxMot_LineToStart (Device_Info.NetIdx, MAP0, fDistList, &Error_Num);

    // Move from P8 to P1 //
    ecmIxMot_ArcPos_A_Start (Device_Info.NetIdx, MAP1, -500, -500, -500, -1000,
    &Error_Num);
}

```


---

## NAME


ecmIxMot\_Arc3P\_R\_Start  
 ecmIxMot\_Arc3P\_R\_Start\_FF  
 ecmIxMot\_Arc3P\_R  
 ecmIxMot\_Arc3P\_R\_NB  
 - 3 점을 통한 원호보간  
 (현재 좌표와 두 점의 상대 좌표)

### INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_cmdidx ecmIxMot\_Arc3P\_R\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XOffset, [in] t\_f64 P2\_YOffset, [in] t\_f64 P3\_XOffset, [in] t\_f64 P3\_YOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

❑ t\_cmdidx ecmIxMot\_Arc3P\_R\_Start\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XOffset, [in] t\_f64 P2\_YOffset, [in] t\_f64 P3\_XOffset, [in] t\_f64 P3\_YOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

❑ t\_success ecmIxMot\_Arc3P\_R([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XOffset, [in] t\_f64 P2\_YOffset, [in] t\_f64 P3\_XOffset, [in] t\_f64 P3\_YOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

❑ t\_success ecmIxMot\_Arc3P\_R\_NB([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XOffset, [in] t\_f64 P2\_YOffset, [in] t\_f64 P3\_XOffset, [in] t\_f64 P3\_YOffset, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

## DESCRIPTION

현재 좌표와 두점의 상대좌표를 통하여 원호보간을 이송을 수행합니다, ecmIxMot\_Arc3P\_R\_Start, ecmIxMot\_Arc3P\_R\_Start\_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmIxMot\_Arc3P\_R, ecmIxMot\_Arc3P\_R\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmIxMot\_Arc3P\_R\_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블록킹(Blocking) 하지 않습니다.

## PARAMETER

▶ NetID : Network 번호

▶ MapIndex : 맵 번호

- ▶ P2\_XOffset : 두번째 점 X 축 상대좌표 값.
- ▶ P2\_YOffset : 두번째 점 Y 축 상대좌표 값.
- ▶ P3\_XOffset : 세번째 점 X 축 상대좌표 값.
- ▶ P3\_YOffset : 세번째 점 Y 축 상대좌표 값.
- ▶ EndAngle : 현재 좌표에서 원하는 위치까지의 각도를 나타냅니다. 현재 좌표에서 세 점을 통해 만들어지는 원위를 이 파라미터 값만큼 보간이송 합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

\* ecmIxMot\_Arc3P\_R\_Start(), ecmIxMot\_Arc3P\_R\_Start\_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

\* ecmIxMot\_Arc3P\_R (), ecmIxMot\_Arc3P\_R\_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	원호보간 이송이 실패 했음을 의미합니다.
1	원호보간 이송이 성공 했음을 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
#define MAP1 1
t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수


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


/*****
* DO_Motion: 작업 명령 시에 호출되는 가상의 함수 입니다.
*****/
void DO_Motion()
{
    ecmIxMot_Arc3P_R_Start(Device_Info.NetIdx, MAP1, 300, -300, 500, -500, 90,
    &Error_Num);
}
    
```

## NAME


ecmIxmMot\_Arc3P\_A\_Start  
 ecmIxmMot\_Arc3P\_A\_Start\_FF  
 ecmIxmMot\_Arc3P\_A  
 ecmIxmMot\_Arc3P\_A\_NB  
 - 3 점을 통한 원호보간  
 (현재 좌표와 두 점의 절대 좌표)

## INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxmMot\_Arc3P\_A\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XPos, [in] t\_f64 P2\_YPos, [in] t\_f64 P3\_XPos, [in] t\_f64 P3\_YPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_cmdidx ecmIxmMot\_Arc3P\_A\_Start\_FF([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XPos, [in] t\_f64 P2\_YPos, [in] t\_f64 P3\_XPos, [in] t\_f64 P3\_YPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_success ecmIxmMot\_Arc3P\_A([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XPos, [in] t\_f64 P2\_YPos, [in] t\_f64 P3\_XPos, [in] t\_f64 P3\_YPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)
- t\_success ecmIxmMot\_Arc3P\_A\_NB([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 P2\_XPos, [in] t\_f64 P2\_YPos, [in] t\_f64 P3\_XPos, [in] t\_f64 P3\_YPos, [in] t\_f64 EndAngle, [out] t\_i32 \*ErrCode)

## DESCRIPTION

현재 좌표와 두점의 절대좌표를 통하여 원호보간을 이송을 수행합니다, ecmIxmMot\_Arc3P\_A\_Start, ecmIxmMot\_Arc3P\_A\_Start\_FF 는 모션을 시작시킨 후에 바로 반환되고, ecmIxmMot\_Arc3P\_A, ecmIxmMot\_Arc3P\_A\_NB 는 모션이 완료되기 전까지 함수가 반환되지 않습니다. 그리고 ecmIxmMot\_Arc3P\_A\_NB 는 함수가 반환되지 않지만 윈도우 메시지를 블록킹(Blocking) 하지 않습니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호

- ▶ P2\_XPos : 두번째 점 X 축 절대좌표 값.
- ▶ P2\_YPos : 두번째 점 Y 축 절대좌표 값.
- ▶ P3\_XPos : 세번째 점 X 축 절대좌표 값.
- ▶ P3\_YPos : 세번째 점 Y 축 절대좌표 값.
- ▶ EndAngle : 현재 좌표에서 원하는 위치까지의 각도를 나타냅니다. 현재 좌표에서 세 점을 통해 만들어지는 원위를 이 파라미터 값만큼 보간이송 합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

RETURN VALUE

\* ecmIxMot\_Arc3P\_A\_Start(), ecmIxMot\_Arc3P\_A\_Start\_FF() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

\* ecmIxMot\_Arc3P\_A(), ecmIxMot\_Arc3P\_A\_NB() : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	원호보간 이송이 실패 했음을 의미합니다.
1	원호보간 이송이 성공 했음을 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
#define MAP1 1
t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

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

/*****
* DO_Motion: 작업 명령 시에 호출되는 가상의 함수 입니다.
*****/
void DO_Motion()
{
    ecmIxMot_Arc3P_A_Start(Device_Info.NetIdx, MAP1, 300, -300, 500, -500, 90,
    &Error_Num);
}
    
```



<h2>NAME</h2> <p>ecmlxMot_Spline_Start</p> <p>- 스플라인 보간 이송</p>	<b>INFORMATION</b>
	 Interpolation Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_cmdidx ecmlxMot\_Spline\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_bool IsReverseDir, [out] t\_i32 \*ErrCode)

### DESCRIPTION

지정된 보간맵의 스플라인 보간 이송을 수행합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hSplineObj : 스플라인 보간 오브젝트 생성시 반환된 오브젝트 핸들값
- ▶ IsReverseDir : 이송 방향을 결정합니다

Value	Meaning
0	순방향으로 이송합니다.
1	역방향으로 이송합니다.

▶ 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 를 반환합니다.

### EXAMPLE

```

//*****
**
*이 예제는 15 개의 좌표를 등록하고 3 개의 축으로 스플라인 보간하는 예제입니다.
*사용자는 이 예제를 보고 응용하여 사용하기를 바랍니다.
    
```

---

```

*****
//

C/C++

#include "ComiEcatSdk_Api.h"

#define NUM_IX_SPLINE_AXES 3
t_i32 anAxisList[NUM_IX_SPLINE_AXES] = {0, 1, 2};
#define NUM_IX_SPLINE_REF_POINTS 15
#define POS_GAIN 1.0
t_f64 afPoint[NUM_IX_SPLINE_REF_POINTS][NUM_IX_SPLINE_AXES] =
{
    {0, 0.0, 0},
    {0, 200000.0, 200000.0},
    {0, 300000.0, 300000.0},
    {0, 400000.0, 400000.0},
    {1002.5, 420000.0, 420001},
    {6350.8, 450000.0, 450045},
    {26795, 500000.0, 500718},
    {80000, 560000.0, 565685},
    {112822, 580000.0, 590871},
    {137550, 590000.0, 605821},
    {200000, 600000.0, 632455},
    {250000, 600000.0, 682455},
    {350000, 600000.0, 782455},
    {500000, 600000.0, 932455},
    {600000, 600000.0, 1032455}
};

#define MAP2 2

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

t_handle32 hSplineObj = 0; // 스플라인 보간 오브젝트의 핸들 값 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 실패시 예외 처리
    }
}

```

---

---

```

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

ecmSxCtl_SetSvon(Device_Info.NetIdx, 0, &Error_Num);
ecmSxCtl_SetSvon(Device_Info.NetIdx, 1, &Error_Num);
ecmSxCtl_SetSvon(Device_Info.NetIdx, 2, &Error_Num);

ecmIxCfg_MapAxes(Device_Info.NetIdx, MAP2, 3, anAxisList, &Error_Num);
//3 축을 보간맵 2 번으로 그룹 설정

}

/*****
* OnSetting() : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 스플라인 보간 이송 설정 부분을 하는 함수 입니다.
* 속도 가속도 등 여러 정보 값들은 적절한 값이라고 가정합니다.
*****/
void OnSetting()
{
    ecmIxCfg_SetSpeedPatt(Device_Info.NetIdx, MAP2, IsVectorSpd, SpdMode, m_ini, m_end,
m_work, m_acc, m_dec, NULL);
    //속도, 가속도, 감속도 설정
    ecmIxCfg_SetJerkRatio(Device_Info.NetIdx, MAP2, m_jerk, NULL);
    //Jeck 값 설정
}

/*****
*OnClickedBuild() : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 빌드 버튼 클릭 시 스플라인 오브젝트 빌드를 하는 함수 입니다.
* 오브젝트 해제 -> 오브젝트 생성 -> 좌표 설명 -> 빌드 순서로 진행
*****/
void OnClickedBuild()
{
    if(!ecmIxCfg_Spline_ClearPool(Device_Info.NetIdx, MAP2, &Error_Num)
    //오브젝트 해제 함수
    {
        //오브젝트 해제 실패 시 예외 처리
    }

    hSplineObj = ecmIxCfg_Spline_AddNewObj (Device_Info.NetIdx, MAP2,
NUM_IX_SPLINE_REF_POINTS, g_Point, & Error_Num);
    // 스플라인 보간 오브젝트 생성

    for(int i=0; i<NUM_IX_SPLINE_REF_POINTS; i++){

        ecmIxCfg_Spline_SetRefPoint (Device_Info.NetIdx, MAP2, hSplineObj, i, afPoint[i],
NUM_IX_SPLINE_AXES, & Error_Num);
        //스플라인 보간 좌표 설정

```

---

---

```

    }
    ecmlxCfg_Spline_BuildObj (Device_Info.NetIdx, MAP2, hSplineObj, & Error_Num);
    //스플라인 보간 빌드 시작
}

/*****
* OnMotion_Minus(): 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 스플라인 보간 이송이 역방향으로 실행 되는 함수 입니다.
*****/
void OnMotion_Minus()
{
    ecmlxMot_LineTo(Device_Info.NetIdx, MAP2, afPoint[NUM_IX_SPLINE_REF_POINTS-
1], &Error_Num);
    if(Error_Num!= 0)
        return;

    ecmlxMot_Spline_Start (Device_Info.NetIdx, MAP2, hSplineObj, FALSE, & Error_Num);
}

/*****
* OnMotion_Plus (): 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 스플라인 보간 이송이 정방향으로 실행 되는 함수 입니다.
*****/
void OnMotion_Plus()
{
    ecmlxMot_LineTo(Device_Info.NetIdx, MAP2, afPoint[0], & Error_Num);
    if(Error_Num!= 0)
        return;

    ecmlxMot_Spline_Start (Device_Info.NetIdx, MAP2, hSplineObj, TURE, & Error_Num);
}

/*****
* OnDestory(): 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 스플라인 보간 이송 종료 후 실행 되는 함수 입니다.
*****/
void OnDestory()
{
    if(!ecmlxCfg_Spline_ClearPool(Device_Info.NetIdx, MAP2, &Error_Num)
//오브젝트 해제 함수
    {
        //오브젝트 해제 실패 시 예외 처리
    }
}
}

```

---

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmlxMot_MPRLin2X_Start</p> <p style="margin: 0;">- 2 축 Multi Point Rounded 보간 이송</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;"> Interpolation 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		Interpolation Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Interpolation Function											
VC++/BCB/.NET											
Level 1											
위험 요소 없음											

## SYNOPSIS

□ t\_success ecmlxMot\_MPRLin2X\_Start([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hMPRLin2XObj, [in] t\_bool IsReverseDir, [out] t\_i32 \*ErrCode)

### DESCRIPTION

지정된 보간맵의 MPRLin2X 오브젝트를 빌드합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hMPRLin2XObj : MPRLin2X 보간 오브젝트 생성시 반환된 오브젝트 핸들값
- ▶ IsReverseDir : 이송 방향을 결정합니다

Value	Meaning
0	순방향으로 이송합니다.
1	역방향으로 이송합니다.

▶ 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 를 반환합니다.

### EXAMPLE

```

//*****
**
*이 예제는 8 개의 좌표를 등록하고 2 개의 축으로 MRP 보간하는 예제입니다.
*사용자는 이 예제를 보고 응용하여 사용하기를 바랍니다.

```

---

```

*****
//

C/C++

#include "ComiEcatSdk_Api.h"

#define NUM_IX_MPRLin2X_REF_POINTS 8
#define POS_GAIN 1000.0
#define ROUND_VALUE 40
TMPRLin2XPoint aMPRLin2XPoint_Copy[NUM_IX_MPRLin2X_REF_POINTS];
TMPRLin2XPoint aMPRLin2XPoints[NUM_IX_MPRLin2X_REF_POINTS] =
{
    {{0, 0.0}, 0.0, ecmROUND_DT_NONE},
    {{500, 500}, ROUND_VALUE, ecmROUND_DT_RADIUS},
    {{600, 200}, ROUND_VALUE, ecmROUND_DT_RADIUS},
    {{800, 700}, ROUND_VALUE, ecmROUND_DT_RADIUS},
    {{100, 700}, ROUND_VALUE, ecmROUND_DT_RADIUS},
    {{100, 500}, ROUND_VALUE, ecmROUND_DT_RADIUS},
    {{250, 300}, ROUND_VALUE, ecmROUND_DT_RADIUS},
    {{0, 0}, 0, ecmROUND_DT_NONE}
    //{{500, 0}, 0, ecmROUND_DT_NONE}
};

#define MAP3 3

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

t_handle32 hMPRLin2XObj = 0; // 스플라인 보간 오브젝트의 핸들 값 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, 0, &Error_Num);
    ecmSxCtl_SetSvon(Device_Info.NetIdx, 1, &Error_Num);

```

---

---

```

t_i32 IxList[MAX_AXES] = {0, 1};
ecmIxCfg_MapAxes(Device_Info.NetIdx, MAP2, 3, IxList, &Error_Num);
//2 축을 보간맵 3 번으로 그룹 설정

}

/*****
* OnSetting() : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 스플라인 보간 이송 설정 부분을 하는 함수 입니다.
* 속도 가속도 등 여러 정보 값들은 적절한 값이라고 가정합니다.
*****/
void OnSetting()
{
    ecmIxCfg_SetSpeedPatt(Device_Info.NetIdx, MAP3, IsVectorSpd, SpdMode, m_ini, m_end,
m_work, m_acc, m_dec, NULL);
    //속도, 가속도, 감속도 설정
    ecmIxCfg_SetJerkRatio(Device_Info.NetIdx, MAP3, m_jerk, NULL);
    //Jeck 값 설정
}

/*****
* OnClickedBuild() : 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 빌드 버튼 클릭 시 스플라인 오브젝트 빌드를 하는 함수 입니다.
* 오브젝트 해제 -> 오브젝트 생성 -> 좌표 설정 -> 빌드 순서로 진행
*****/
void OnClickedBuild()
{
    for(int i=0; i< NUM_IX_MPRLin2X_REF_POINTS; i++){
        aMPRLin2XPoint_Copy[i].Pos[0] = aMPRLin2XPoints[i].Pos[0] * POS_GAIN;
        aMPRLin2XPoint_Copy[i].Pos[1] = aMPRLin2XPoints[i].Pos[1] * POS_GAIN;
        aMPRLin2XPoint_Copy[i].RoundData = aMPRLin2XPoints[i].RoundData * POS_GAIN;
        aMPRLin2XPoint_Copy[i].RoundDataType = aMPRLin2XPoints[i].RoundDataType;
        //MRP 오브젝트 카피 배열에 값 조절 후 저장
    }

    if(!ecmIxCfg_MPRLin2X_ClearPool (Device_Info.NetIdx, MAP3, &Error_Num)
//오브젝트 해제 함수
    {
        //오브젝트 해제 실패 시 예외 처리
    }

    hMPRLin2XObj = ecmIxCfg_MPRLin2X_AddNewObj (Device_Info.NetIdx, MAP3,
NUM_IX_MPRLin2X_REF_POINTS, g_Point, & Error_Num);
    // MRP 보간 오브젝트 생성

    for(int i=0; i< NUM_IX_MPRLin2X_REF_POINTS; i++){

```

---

---

```

    ecmIxCfg_MPRLin2X_SetRefPoint (Device_Info.NetIdx, MAP3, hMPRLin2XObj, i,
aMPRLin2XPoint_Copy[i].Pos, aMPRLin2XPoint_Copy[i].RoundData,
aMPRLin2XPoint_Copy[i].RoundDataType, & Error_Num);
    //MRP 보간 좌표 설정

}
    ecmIxCfg_MPRLin2X_BuildObj (Device_Info.NetIdx, MAP3, hMPRLin2XObj, &
Error_Num);
    //MRP 보간 빌드 시작
}

/*****
* OnMotion_Minus(): 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* MRP 보간 이송이 역방향으로 실행 되는 함수 입니다.
*****/
void OnMotion_Minus()
{
    ecmIxMot_MPRLin2X_Start (Device_Info.NetIdx, MAP3, hMPRLin2XObj, FALSE,
&Error_Num);
}

/*****
* OnMotion_Plus (): 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* MRP 보간 이송이 정방향으로 실행 되는 함수 입니다.
*****/
void OnMotion_Plus()
{
    ecmIxMot_MPRLin2X_Start (Device_Info.NetIdx, MAP3, hMPRLin2XObj, TRUE,
&Error_Num);
}

/*****
* OnDestory(): 이 함수는 가상의 함수로서 프로그램 초기화 루틴이
* 적용되는 부분을 의미합니다.
* 스플라인 보간 이송 종료 후 실행 되는 함수 입니다.
*****/
void OnDestory()
{
    if(!ecmIxCfg_MPRLin2X_ClearPool (Device_Info.NetIdx, MAP2, &Error_Num)
//오브젝트 해제 함수
    {
        //오브젝트 해제 실패 시 예외 처리
    }
}
}

```

---



<h2 style="margin: 0;">NAME</h2> <p style="margin: 5px 0 0 20px;">ecmlxMot_Stop</p> <p style="margin: 5px 0 0 20px;">ecmlxMot_Stop_FF</p> <p style="margin: 5px 0 0 20px;">- 보간 이송 정지</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;"> Interpolation 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		Interpolation Function		VC++/BCB/.NET		Level 1		위험 요소 없음	
INFORMATION											
Interpolation Function											
VC++/BCB/.NET											
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
}

```

---

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

---

## SYNOPSIS

---

□ t\_bool ecmIxSt\_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 NULL 을 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 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
#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(!ecmIxSt_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_32 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(!ecmSxSt_WaitCompt(Device_Info.NetIdx, Map0, &Error_Num)){
        // Map0 의 보간 이송 완료를 대기
        // 보간 이송 대기 실패 예외 처리
        return;
    }
}
```

---

NAME	I N F O R M A T I O N
ecmIxCfg_SetMastWeight	 Interpolation Function
ecmIxCfg_GetMastWeight	 VC++/BCB/.NET
-	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmIxCfg\_SetMastWeight([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_f64 MastWeightList, [out] t\_i32 \*ErrCode)
- t\_success ecmIxCfg\_GetMastWeight([in] t\_i32 NetID, [in] t\_i32 MapIndex, [out] t\_f64 \*MastWeightList, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

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

## RETURN VALUE

- \* ecmIxCfg\_SetMastWeight () : 이 리턴값은
- \* ecmIxCfg\_GetMastWeight () : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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


## EXAMPLE


**NAME**


ecmIxCfg\_Spline\_SetTrgOutEnv  
ecmIxCfg\_Spline\_GetTrgOutEnv

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmIxCfg\_Spline\_SetTrgOutEnv([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_bool IsEnableTrgOut, [in] t\_i16 TrgOfsTime\_ms, [out] t\_i32 \*ErrCode)

□ t\_success ecmIxCfg\_Spline\_GetTrgOutEnv([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [out] t\_bool \*IsEnableTrgOut, [out] t\_i16 \*TrgOfsTime\_ms, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ MapIndex : 맵 번호
- ▶ hSplineObj :
- ▶ IsEnableTrgOut :
- ▶ TrgOfsTime\_ms :
- ▶ 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.

**EXAMPLE**





## NAME


ecmIxCfg\_Spline\_SetTrgOutPos  
ecmIxCfg\_Spline\_GetTrgOutPos

### INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

❑ t\_f64 ecmIxCfg\_Spline\_SetTrgOutPos([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_i32 Axis, [in] t\_i32 PointIndex, [out] t\_i32 \*ErrCode)

헤더 부분에 없는 것 같아 추가하였습니다.

❑ t\_f64 ecmIxCfg\_Spline\_GetTrgOutPos([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_i32 Axis, [in] t\_i32 PointIndex, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

▶ NetID : Network 번호

▶ MapIndex : 맵 번호

▶ hSplineObj :





▶ Axis :

▶ PointIndex :

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

## RETURN VALUE

## EXAMPLE

NAME	I N F O R M A T I O N
ecmIxCfg_Spline_SetJsType	 Interpolation Function
ecmIxCfg_Spline_GetJsType	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_success ecmIxCfg\_Spline\_GetJsType([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_handle32 JerkSmoothType, [out] t\_i32 \*ErrCode)
- t\_i32 ecmIxCfg\_Spline\_GetJsType([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는

## PARAMETER

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

## RETURN VALUE

\* ecmIxCfg\_Spline\_GetJsType (): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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


## EXAMPLE


**NAME**


ecmIxCfg\_Spline\_SetJsProp\_F  
ecmIxCfg\_Spline\_GetJsProp\_F

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

- t\_success ecmIxCfg\_Spline\_SetJsProp\_F([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_i32 PropId, [in] t\_f64 PropVal, [out] t\_i32 \*ErrCode)
- t\_f64 ecmIxCfg\_Spline\_GetJsProp\_F([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_i32 PropId, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는

**PARAMETER**

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

**RETURN VALUE**

\* ecmIxCfg\_Spline\_SetJsProp\_F (): 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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


**EXAMPLE**


**NAME**


ecmIxCfg\_Spline\_SetJerkThresh  
ecmIxCfg\_Spline\_GetJerkThresh

**INFORMATION**

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

- t\_success ecmIxCfg\_Spline\_SetJerkThresh([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_i32 Axis, [in] t\_f64 JerkThreshold, [out] t\_i32 \*ErrCode)
- t\_f64 ecmIxCfg\_Spline\_GetJerkThresh([in] t\_i32 NetID, [in] t\_i32 MapIndex, [in] t\_handle32 hSplineObj, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는

**PARAMETER**

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

**RETURN VALUE**

\* ecmIxCfg\_Spline\_SetJerkThresh () : 이 리턴값은 불 형(Boolean Type) 을 가지고 있습니다.

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

**EXAMPLE**

## NAME

ecmIxBot\_MoveVia2X\_Start  
- 경유이송 시작

## INFORMATION

📁 Interpolation Function

✍ VC++/BCB/.NET

📖 Level 1

😊 위험 요소 없음

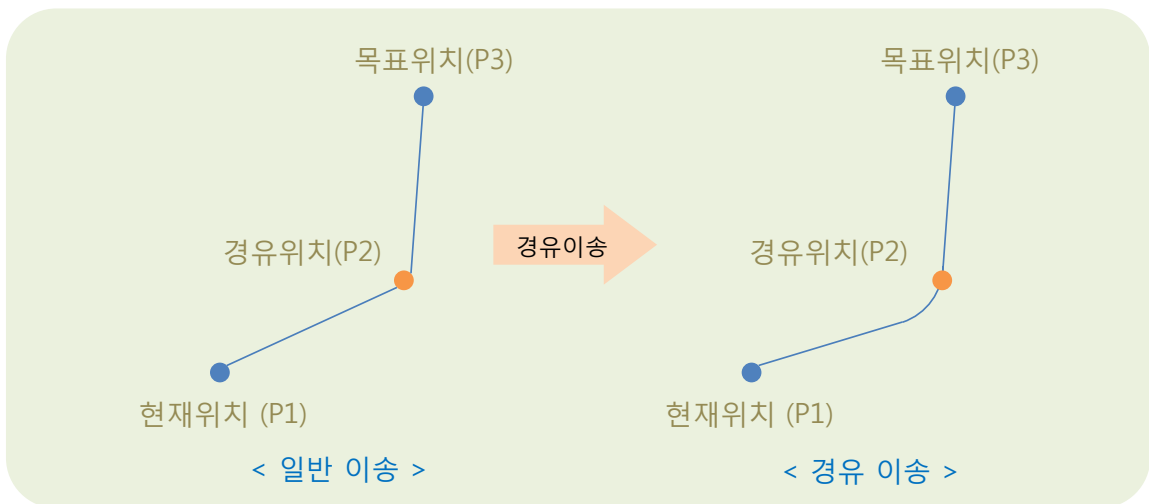
## SYNOPSIS

❑ t\_cmdidx ecmIxBot\_MoveVia2X\_Start([in] t\_i32 NetID, [in] t\_ui16 MapIndex, [in] t\_f64 P2[], [in] t\_f64 P3[], [in] t\_bool IsAbsPosMode, [in] t\_i32 RoundPosType, [in] t\_f64 NormRadius, [in] t\_f64 MinRadius, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 2 축을 대상으로 하는 경유이송(Move-Via)을 시작하는 함수입니다.

경유이송은 2 개의 축이 속도의 연속성을 유지하면서 현재 위치로부터 어느 한 점을 경유하여 목표 위치로 보간 이송을 수행하는 것을 말합니다.



경유점을 통과할 때에는 속도의 연속성을 확보하기 위해서 원호보간 이송이 자동으로 삽입됩니다. 경유이송이 단순한 원호삽입 이송과 다른 점은 반드시 지정된 경유점을 지나도록 원호가 삽입된다는 것입니다<sup>3</sup>.

## PARAMETER

<sup>3</sup> 반드시 경유점을 지난다는 것은 지령위치 기준으로 말하는 것이며 피드백 위치를 기준으로 말하는 것은 아닙니다. 지령위치가 경유점을 지나더라도 피드백 위치는 모터의 게인 튜닝 상태등과 같은 여러 가지 요인에 의해서 영향을 받아서 경유점을 지나지 않을 수도 있습니다.

- ▶ NetID : Network 번호
- ▶ MapIndex : 보간 맵 번호
- ▶ P2[] : 경유점 좌표 포인트(2 축의 좌표를 표현하는 배열)
- ▶ P3[] : 목표점 좌표 포인트(2 축의 좌표를 표현하는 배열)
- ▶ IsAbsPosMode : 지정된 좌표 데이터가 상대좌표인지 절대좌표인지를 나타내는 플래그

Value	Meaning
0	상대좌표값으로 P2, P3 데이터를 전달합니다.
1	절대좌표값으로 P2, P3 데이터를 전달합니다.

- ▶ RoundPosType : 삽입되는 라운드와 경유점과의 위치 관계를 나타내는 인자입니다 (EEcmRoundPosType 선언 참조).

Value	Meaning
0 (ecmROUND_PT_START)	경유점이 자동 삽입되는 원호의 시작점이 됩니다.
1 (ecmROUND_PT_END)	경유점이 자동 삽입되는 원호의 종점이 됩니다.

- ▶ NormRadius : 자동으로 삽입되는 라운드의 크기(원호의 반지름)를 나타내는 인자입니다.
- ▶ MinRadius : P1 이 라운드 원의 외부에 있을 때는 'NormRadius'를 적용하지만 P1 이 원의 내부에 있을 때는 P1 의 위치에 따라서 원의 크기가 조정됩니다. 이때 조정할 수 있는 원의 최소 크기(반지름)를 나타내는 인자입니다
- ▶ 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

```
#define NETID      0
#define IX_MAP_IDX 0
t_i32 axisList[2] = {2, 3}; // 축 번호 리스트
t_f64 P1[2] = {0, 0}; // 시작점
t_f64 P2[2] = {1000000, 500000}; // 경유점
t_f64 P3[2] = {1200000, 1000000}; // 종점

// 보간 축 맵핑 //
ecmIxCfg_MapAxes(NETID, IX_MAP_IDX, 2, axisList, NULL);
// 보간 속도 설정 //
ecmIxCfg_SetSpeedPatt(NETID, IX_MAP_IDX, ecmIXSPT_VECTOR, ecmSMODE_SCURVE, 0, 0, 500000,
2500000, 2500000, NULL);
// 시작점으로 이동 (이미 시작점에 있는 경우에는 이 라인 생략 가능) //
ecmIxBot_LineTo(NETID, IX_MAP_IDX, P1, NULL);
```


```
// 경유이송 시작 //
ecmIxmot_MoveVia2X_Start(NETID, IX_MAP_IDX, P2, P3, true, ecmROUND_PT_START, 10000, 5000,
NULL);
// 이송완료 대기 //
t_i32 ErrCode;
if(!ecmIxmot_WaitCompt (NETID, IX_MAP_IDX, &ErrCode)){
    // 에러처리! //
}
```


## NAME


**ecmIxMot\_MoveVia2X\_OverrideTP**  
 - 경유이송의 목표점 오버라이드

### INFORMATION

 Interpolation Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```

  □ t_cmdidx ecmIxMot_MoveVia2X_OverrideTP ([in] t_i32 NetID, [in] t_ui16 MapIndex,
  [in] t_f64 TP_new[], [in] t_f64 RoundRadius, [out] t_bool *IsIgnored, [out] t_i32
  *ErrCode)
  
```

## DESCRIPTION

이 함수는 경유이송의 목표점을 수정하는 함수입니다. 목표좌표가 바뀌어서 원래의 경로와 비교하여 경로가 바뀌는 곳에서는 속도의 연속성을 확보하기 위해서 원호가 자동삽입됩니다. 이때의 원호는 경유점에서 삽입되는 원호와 별개의 원호입니다.

이 함수는 경유이송이 진행되고 있는 상태에서만 실행가능합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MapIndex : 보간 맵 번호
- ▶ TP\_new[] : 새로운 목표 좌표 포인트(2 축의 좌표를 표현하는 배열)
- ▶ RoundRadius : 자동으로 삽입되는 원호의 반지름을 나타내는 인자입니다.
- ▶ IsIgnored : 목표점 오버라이드가 무시되었는지를 나타내는 플래그 값을 반환받을 포인터 인자입니다. 참고로 이미 이송이 완료된 상태에서는 오버라이드가 무시됩니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

\* 이 리턴값은

## EXAMPLE

```

#define NETID      0
#define IX_MAP_IDX 0

void OnMoveViaStart()
  
```



```

{
    t_i32 axisList[2] = {2, 3}; // 축 번호 리스트
    t_f64 P1[2] = {0, 0}; // 시작점
    t_f64 P2[2] = {1000000, 500000}; // 경유점
    t_f64 P3[2] = {1200000, 1000000}; // 종점

    // 보간 축 맵핑 //
    ecmIxCfg_MapAxes(NETID, IX_MAP_IDX, 2, axisList, NULL);
    // 보간 속도 설정 //
    ecmIxCfg_SetSpeedPatt(NETID, IX_MAP_IDX, ecmIXSPT_VECTOR, ecmSMODE_SCURVE, 0, 0, 500000,
        2500000, 2500000, NULL);
    // 시작점으로 이송 (이미 시작점에 있는 경우에는 이 라인 생략 가능) //
    ecmIxBot_LineTo(NETID, IX_MAP_IDX, P1, NULL);
    // 경유이송 시작 //
    ecmIxBot_MoveVia2X_Start(NETID, IX_MAP_IDX, P2, P3, true, ecmROUND_PT_START, 10000, 5000,
        NULL);
    // 이송완료 대기 //
    t_i32 ErrCode;
    if(!ecmIxBot_WaitCompt (NETID, IX_MAP_IDX, &ErrCode)){
        // 에러처리! //
    }
}

void OnOverrideViaTP()
{
    t_f64 TP_new[2] = {1300000, 800000};
    t_bool IsIgnored;
    t_i32 ErrCode;
    ecmIxBot_MoveVia2X_OverrideTP (NETID, IX_MAP_IDX, TP_new, 10000, &IsIgnored, &ErrCode);

    if(ErrCode != 0){
        // 에러처리! //
    }

    if(IsIgnored){
        // 오버라이드 무시된 것에 대한 처리! //
    }
}
}

```

# 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 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.







## 17 Homing Operation Functions

### 17.1 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_success ecmHomeCfg_SetEscDist([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 EscDist, [out] t_i32 *ErrCode) 지정된 축의 원점복귀 시 탈출 거리를 설정합니다.</p>
<p>□ t_f64 ecmHomeCfg_GetEscDist([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 지정된 축의 현재 설정된 탈출 거리를 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 ecmHomeCfg_SetOption([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmHomeOptID OptionID, [in] t_i32 OptionVal, [out] t_i32 *ErrCode) 지정된 축의 원점복귀와 관련된 기타 옵션 사항들을 설정합니다.</p>
<p>□ t_i32 ecmHomeCfg_GetOption([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmHomeOptID OptionID, [out] t_i32 *ErrCode) 지정된 축의 현재 설정된 원점복귀 옵션 사항들을 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_cmdidx ecmHomeMot_MoveStart_FF([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>

**17.2** 함수 설명

NAME	INFORMATION
ecmHomeCfg_SetMode	 Home      Operation Function
ecmHomeCfg_GetMode	 VC+ +/BCB/.NET
- 원점복귀 모드 설정 및 반환	 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	I N F O R M A T I O N
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 거리입니다.
- ▶ 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



NAME	I N F O R M A T I O N
ecmHomeCfg_SetEscDist	 Home      Operation Function
ecmHomeCfg_GetEscDist	 VC++/BCB/.NET
- 원점복귀 탈출거리 설정 및 반환	 Level 1
	 위험 요소 없음

## SYNOPSIS

- t\_cmdidx ecmHomeCfg\_SetEscDist([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_f64 EscDist, [out] t\_i32 \*ErrCode)
- t\_f64 ecmHomeCfg\_GetEscDist([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 원점 탈출 거리를 설정 및 반환합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ EscDist : 원점 탈출 거리. 이 값의 단위는 논리적 거리입니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE

- \* ecmHomeCfg\_SetEscDist() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.
- \* ecmHomeCfg\_GetEscDist() : 이 리턴값은 설정된 EscDist 값이 반환됩니다.

NAME	I N F O R M A T I O N
ecmHomeCfg_SetSpeedPatt	Home      Operation
ecmHomeCfg_GetSpeedPatt	Function
- 원점복귀 속도 설정 및 반환	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 값 입니다.
- ▶ Acc : 가속도에 대한 설정 및 반환 값 입니다.
- ▶ Dec : 감속도에 대한 설정 및 반환 값 입니다.
- ▶ HomeSpecVel : HomeSpec 속도의 설정 및 반환 값 입니다.

▶ **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

<b>NAME</b>	<b>INFORMATION</b>
ecmHomeCfg_SetOption ecmHomeCfg_GetOption - 원점복귀 관련 기타 옵션 설정	Home Operation Function VC++/BCB/.NET Level 1 위험 요소 없음

## SYNOPSIS

```

❑ t_cmdidx ecmHomeCfg_SetOption([in] t_i32 NetID, [in] t_i32 Axis, [in]
EEcmHomeOptID OptionID, [in] t_i32 OptionVal, [out] t_i32 *ErrCode)
❑ t_i32 ecmHomeCfg_GetOption([in] t_i32 NetID, [in] t_i32 Axis, [in] EEcmHomeOptID
OptionID, [out] t_i32 *ErrCode)
    
```

## DESCRIPTION

이 함수는 지정된 축의 원점복귀와 관련한 기타 옵션 사항들을 설정 및 반환합니다. 이 함수는 원점복귀와 관련된 여러가지 기타 옵션들을 설정할 때 공통적으로 사용할 수 있는 함수입니다.(현재는 기타 옵션에 ecmHOID\_TPROB\_EDGE\_SEL 옵션만 정의되어 있음)

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ OptionID : 설정하고자 하는 원점복귀 관련 옵션 종류를 구분하는 ID. 설정할 수 있는 값은 아래와 같습니다.

Value	Meaning
0 or ecmHOID_TPROB_EDGE_SEL	마스터 직접 구동 원점복귀 모드를 사용할 때 원점 위치를 래치하기 위하여 Touch Probe 기능을 사용하는데 이때 Touch Probe 센서의 상승에지(Rising Edge)를 사용하여 래치할 것인지, 하강에지(Falling Edge)를 사용하여 래치할 것인지를 설정하겠다는 것을 나타내는 아이디 값입니다.

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

## RETURN VALUE




\* ecmHomeCfg\_SetOption() : 이 리턴값은 함수의 cmdidx 가 반환됩니다.

\* ecmHomeCfg\_GetOption() : 이 리턴값은 설정된 EscDist 값이 반환됩니다.

REFERENCE

□ 설정가능한 OptionID 값과 각 OptionVal 값은 다음 표와 같습니다.

OptionID	OptionID 설명 및 설정가능한 OptionVal
<p><b>ecmHOLD_TPROB_EDGE_SEL</b> (또는 0)</p>	<p>마스터 직접 구동 원점복귀 모드를 사용할 때 원점 위치를 래치하기 위해서 Touch Probe 기능을 사용하는데 이때 Touch Probe센서의 상승에지(Rising Edge, Positive Edge)를 사용하여 래치할 것인지, 하강에지(Falling Edge, Negative Edge)를 사용하여 래치할 것인지를 설정하겠다는 것을 나타내는 아이디 값입니다. 이때 설정가능한 OptionVal 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>● <b>ecmEDGE_NEG (또는 0):</b> 하강에지 사용. 즉 Touch Probe가 ON에서 OFF로 변경될 때 Position을 래치함.</li> <li>● <b>ecmEDGE_POS (또는 1):</b> 상승에지 사용. 즉 Touch Probe가 OFF에서 ON으로 변경될 때 Position을 래치함.</li> </ul> <p>※ Touch Probe 신호는 무조건 A접점으로만 인식합니다.</p> <p>※ 원점센서를 사용하는 모드이거나 Z상을 이용해서 원점을 잡는 모드 (107, 109번 모드를 제외한 모든 원점복귀 모드)에서는 이 값을 ecmEDGE_POS 으로 설정합니다.</p> <p>※ 107번 원점복귀 모드에서는 리미트센서가 <b>A접점</b>인 경우에 <b>ecmEDGE_NEG</b> 값을 설정하고 <b>B접점</b>인 경우에 <b>ecmEDGE_POS</b> 값을 사용합니다.</p> <p>※ 109번 원점복귀 모드에서는 리미트센서가 <b>A접점</b>인 경우에 <b>ecmEDGE_POS</b> 값을 설정하고 <b>B접점</b>인 경우에 <b>ecmEDGE_NEG</b> 값을 사용합니다.</p>

<h2>NAME</h2> <p>ecmHomeMot_MoveStart</p> <p>ecmHomeMot_MoveStart_FF</p> <p>ecmHomeMot_Move</p> <p>- 단축 원점복귀 이송</p>	<b>INFORMATION</b>	
	 Home	Operation
	Function	
	 VC++/BCB/.NET	
	 Level 1	
 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>NAME</h2> <p>ecmHomeSt_IsBusy</p> <p>- 원점복귀 모션 이송 확인</p>	<b>INFORMATION</b>	
	 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 이송증인지를 확인합니다.

### 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	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_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* 프로그램 앞 부분은 생략
(설정 부분 다 생략)

```

---





```
*****/
/*****
* DO_Busy: 작업 명령 시에 호출되는 가상의 함수입니다.
*****/
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

NAME	I N F O R M A T I O N
ecmHomeSt_GetFlags - 원점복귀 상태 Flag 값 반환	<div style="display: flex; justify-content: space-between; align-items: center;"> <span> Home</span> <span>Operation</span> </div> <hr/> <div style="display: flex; justify-content: space-between; align-items: center;"> <span></span> <span>Function</span> </div> <hr/> <div style="display: flex; justify-content: space-between; align-items: center;"> <span> VC++/BCB/.NET</span> </div> <hr/> <div style="display: flex; justify-content: space-between; align-items: center;"> <span> Level 1</span> </div> <hr/> <div style="display: flex; justify-content: space-between; align-items: center;"> <span> 위험 요소 없음</span> </div>

## 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 받아들일 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

}

```

---

# Motion Master/Slave 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.







## 18 Motion Master/Slave Functions

### 18.1 함수 요약

Summary of Functions
<p>□ t_success ecmMsCfg_SetSlvEnv([in] t_i32 NetID, [in] t_i32 Axis, [in] t_i32 MasterAxis, [in] t_f64 PosRatio, [out] t_i32 *ErrCode) Master/Slave 기능의 Slave 로 동작할 축에 대한 Slave 동작 환경을 설정합니다.</p>
<p>□ t_success ecmMsCfg_GetSlvEnv([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32* MasterAxis, [out] t_f64* PosRatio, [out] t_i32 *ErrCode) Master/Slave 기능의 Slave 로 동작하는 축의 Slave 동작 환경 값을 반환합니다.</p>
<p>□ t_success ecmMsCtl_StartSlv([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) Master/Slave 기능의 Slave 로서의 동작을 활성화 합니다,</p>
<p>□ t_success ecmMsCtl_StopSlv([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) Master/Slave 기능의 Slave 로서의 동작을 비활성화 합니다.</p>
<p>□ t_bool ecmMsSt_IsSlvStarted([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 지정한 축이 Master/Slave 기능의 Slave 로서 동작이 활성화 되었는지를 확인합니다.</p>
<p>□ t_success ecmMsCtl_SynchSlv([in] t_i32 NetID, [in] t_i32 Axis, [in] t_f64 MastPosOfs, [in] t_bool IsWaitMoveCompt, [out] t_i32 *ErrCode);</p>

**18.2** 함수 설명

NAME	I N F O R M A T I O N
ecmMsCfg_SetSlvEnv	 Master/Slave Function
ecmMsCfg_GetSlvEnv	 VC++/BCB/.NET
- Slave 동작 환경 설정 및 반환	 Level 1
	 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmMsCfg\_SetSlvEnv([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 MasterAxis, [in] t\_f64 PosRatio, [out] t\_i32 \*ErrCode)

□t\_success ecmMsCfg\_GetSlvEnv([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32\* MasterAxis, [out] t\_f64\* PosRatio, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 Master/Slave 기능의 Slave 로 동작할 축에 대한 Slave 동작 환경을 설정 및 반환합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호.
- ▶ MasterAxis : 마스터로 동작하는 축 번호
- ▶ PosRatio : 마스터 축 이송거리와 슬레이브 축 이송거리의 비율을 나타냅니다. PosRatio 를 수식으로 표현하면 아래와 같습니다

$$\text{PosRatio} = D1 / D2 \text{ (여기서 } D1 \text{ 은 슬레이브축 이송거리, } D2 \text{ 는 마스터 축 이송거리)}$$

예를 들어서 PosRatio 가 0.5 이면 마스터축의 이송량이 1000 일 때 슬레이브축은 500 을 이송하게 됩니다.

- ▶ 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	설정 및 반환이 실패했음을 의미합니다,
1	설정 및 반환이 성공했음을 의미합니다,

## EXAMPLE

---

```

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

t_32 Error_Num = 0;//함수 별 에러 코드 저장 변수

/*****
* 프로그램 앞 부분은 생략
*****/
/*****
* OnSetSlv: 작업 명령 시에 호출되는 가상의 함수 입니다.
* 이 함수는 - Slave 동작 환경 설정을 하는 함수 입니다.
*****/
void OnSetSlv ()
{
    t_i32 g_MsAxis = //마스터로 설정 할 축 번호를 입력
    t_i32 g_SlvAxis = //슬레이브로 설정 할 축 번호를 입력
    t_i32 PosRatio = //슬레이브 이동거리 / 마스터 이동거리 계산해서 입력

    ecmMsCfg_SetSlvEnv(Device_Info.NetIdx, g_MsAxis, g_SlvAxis, PosRatio, &Error_Num);
}

```


---


**NAME**


ecmMsCtl\_StartSlv  
- 슬레이브 축 활성화

**INFORMATION**

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmMsCtl\_StartSlv([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 Slave 로 동작하는 축을 활성화 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	활성화 설정에 실패하였음을 의미합니다,
1	활성화 설정에 성공하였음을 의미합니다.

**REFERENCE**

\*슬레이브 축의 동작 상태는 ecmMsSt\_IsSlvStarted() 함수와 ecmSxSt\_GetMotState() 함수를 통해서 체크 할수 있습니다, 슬레이브 축이 활성화 된 상태에서 ecmSxSt\_GetMotState() 함수는 항상 ecmMST\_IN\_SLAVE\_MODE(==13) 값을 반환하게 됩니다.

**EXAMPLE**

C/C++

---





//본 예제는 ecmMxMot\_MoveStart 사용하여 두 축을 절대 좌표 5000 으로 이동하는 예  
입니다.

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

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

```
/******  
* 프로그램 앞 부분은 생략  
*****/  
/*****  
* OnEnable: 작업 명령 시에 호출되는 가상의 함수 입니다.  
* 이 함수는 - Slave 를 활성화하는 함수입니다.  
*****/  
void OnEnable()  
{  
    t_i32 g_SlvAxis = //슬레이브로 설정 할 축 번호를 입력  
    t_success ret = false ;  
  
    ret = ecmMsCtl_StartSlv(Device_Info.NetIdx, g_SlvAxis, &Error_Num);  
  
    if(!ret)  
        AfxMessageBox(L"StartSlv Fail");  
}
```

---

<h2>NAME</h2> <p>ecmMsCtl_StopSlv</p> <p>- 슬레이브 축 비활성화</p>	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_success ecmMsCtl\_StopSlv([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 Slave 로 동작하는 축을 비 활성화 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	비 활성화 설정에 실패하였음을 의미합니다,
1	비 활성화 설정에 성공하였음을 의미합니다.


**NAME**


ecmMsSt\_IsSlvStarted  
- 슬레이브 축 활성화 여부 반환

**INFORMATION**

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

❑ t\_bool ecmMsSt\_IsSlvStarted([in] t\_i32 NetID, [in] t\_i32 Axis, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는지정된 축이 Master/Slave 기능의 Slave 로서 동작이 활성화 되었는지에 대한 플래그값을 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 불 형(Boolean Type) 을 가지고 있습니다.

Value	Meaning
0	슬레이브 축이 비활성화 되어있음을 의미합니다.
1	슬레이브 축이 활성화 되어있음을 의미합니다.

<b>NAME</b>	<b>INFORMATION</b>
ecmMsCtl_SynchSlv	Master/Slave Function
-	VC++/BCB/.NET
	Level 1
	위험 요소 없음

## SYNOPSIS

□ t\_success ecmMsCtl\_SynchSlv([in] t\_i32 NetID, [in] t\_f64 MastPosOfs, [in] t\_bool IsWaitMoveCompt, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 마스터-슬레이브 기능의 슬레이브 축으로 등록된 축에 한해서 마스터의 현재 위치와 같은 위치로 가도록 이송 명령을 내려주는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ MastPosOfs : 마스터축의 위치 대비하여 일정량의 오프셋 위치를 가지고 싶다면 이 값을 0이 아닌 해당 오프셋 위치를 설정합니다. 대개는 이 값을 0으로 설정합니다.
- ▶ IsWaitMoveCompt : SynchSlv 이송이 완료될 때까지 기다릴 것인지를 나타내는 인자입니다.
- ▶ 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.

# Motion LIST Motion 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.



## 19 Motion LIST Motion Functions





### 19.1 함수 요약

Summary of Functions	
□ t_success ecmLmCtl_Begin ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [in] t_ui32 AxisMask1, [in] t_ui32 AxisMask2, [out] t_i32 *ErrCode)	하나의 리스트 모션맵에 대해서 리스트 모션 기능을 시작합니다.
□ t_success ecmLmCtl_End ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)	하나의 리스트 모션맵에 대해서 리스트 모션 기능을 종료합니다
□ t_success ecmLmCtl_Run ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)	하나의 리스트 모션맵에 대해서 리스트 모션으로 등록된 명령들의 실행을 시작합니다.
□ t_success ecmLmCtl_Stop ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [in] t_bool IsComptCurStep, [in] t_i32 DecelTime_ms, [out] t_i32 *ErrCode)	현재 실행되고 있는 리스트모션의 실행을 일시 정지합니다.
□ t_success ecmLmCtl_ClearQue ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)	리스트모션 큐 버퍼가 꽉 찬 상태에서 리스트모션에 등록되어야 할 함수가 호출 되었을 때 리스트모션 큐가 해당 함수를 등록할 수 있는 여유 공간이 확보될 때까지 기다릴지 아니면 바로 에러처리하고 리턴(Return)할 지를 결정하는 함수입니다.
□ t_i32 ecmLmCfg_SetQueFullMode ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)	리스트모션 큐 버퍼가 꽉 찬 상태에서 리스트모션에 등록되어야 할 함수가 호출 되었을 때 리스트모션 큐가 해당 함수를 등록할 수 있는 여유 공간이 확보될 때까지 기다릴지 아니면 바로 에러처리하고 리턴(Return)할 지를 결정 여부를 반환하는 함수입니다.
□ t_success ecmLmCfg_SetStepId ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)	리스트모션 테이블에 등록되는 각 명령들의 스텝 아이디(ID)값을 부여하는 함수입니다
□ t_i32 ecmLmCfg_GetStepId ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)	리스트모션 테이블에 등록되는 각 명령들의 스텝 아이디(ID)값을 반환하는 함수입니다.
□ t_success ecmLmCfg_SetStepParam ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [in] t_i32 StepParam1, [in] t_i32 StepParam2, [out] t_i32 *ErrCode)	각 명령 스텝들이 리스트모션 테이블에 등록될 때 함께 등록할 2 개의 데이터를 정의하는 것입니다. 이 2 개의 데이터 값은 추후에 해당 스텝이 실행될 때 그 값을 참고할 수 있습니다.
□ t_success ecmLmCfg_GetStepParam ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_dword* StepParam1, [out] t_dword* StepParam2, [out] t_i32 *ErrCode)	각 명령 스텝들이 리스트모션 테이블에 등록될 때 함께 등록할 2 개의 데이터를 정의를 반환하는 함수입니다.



<p>□ t_i32 ecmLmSt_GetRunSts ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [in] t_i32 *ErrCode)</p> <p>LmMapIndex 로 지정된 리스트모션 맵의 동작상태를 반환합니다.</p>
<p>□ t_i32 ecmLmSt_GetRemStepCount ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)</p> <p>리스트모션 테이블에 등록된 명령 스텝들 중에 실행되지 않고 남아 있는 명령의 수를 반환합니다.</p>
<p>□ t_success ecmLmSt_GetRunStepInfo ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *RunStepCount, [out] t_i32 *RunStepId, [out] t_i32 *RunStepSts, [out] t_i32 *ErrCode)</p> <p>현재 실행되고 있는 스텝에 대한 정보를 반환하는 함수입니다</p>
<p>□ t_success ecmLmSt_GetRunStepParam ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_dword *StepParam1, [out] t_dword *StepParam2, [out] t_i32 *ErrCode)</p> <p>현재 실행되고 있는 스텝의 스텝 파라미터값들을 반환하는 함수입니다</p>
<p>□ t_success ecmLmCmd_Delay ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [in] t_i32 DelayTime_ms, [out] t_i32 *ErrCode)</p> <p>리스트모션 실행 시에 시간 적용되어야 할 필요가 있는 경우에 사용될 수 있도록 시간 지연 명령을 리스트모션 테이블에 등록하는 것입니다</p>
<p>□ t_success ecmLmResetAll([in] t_i32 NetID, [out] t_i32 *ErrCode)</p>
<p>□ t_success ecmLmCfg_SetQueDepth([in] t_i32 NetID, [in] t_i32 LmMapIndex, [in] t_i32 QueDepth, [out] t_i32 *ErrCode)</p>
<p>□ t_i32 ecmLmCfg_GetQueDepth([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)</p>

19.2 함수 설명

<h1>NAME</h1> <p><b>ecmLmCtl_Begin</b></p> <p>- LIST-Motion 기능 시작</p>	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

□ t\_success ecmLmCtl\_Begin ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_ui32 AxisMask1, [in] t\_ui32 AxisMask2, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 하나의 리스트 모션맵에 대해서 리스트 모션 기능을 시작합니다. 리스트 모션이 시작 되면 해당 리스트 모션 맵에 포함된 축들의 이송 명령들은 리스트모션 테이블에 등록 되고 난 후에 **ecmLmCtl\_Run()** 함수가 실행된 이후에 일괄적으로 실행되게 됩니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ LmMapIndex : 시작하고자 하는 대상 LIST-Motion 맵 인덱스. LIST-Motion 맵은 최대 8 개까지 지원되므로 이 값은 0 ~ 7 의 숫자이어야 합니다..
- ▶ AxisMask1 : 0 ~ 31 번 축 중에서 LIST-Motion 에 참여하는 축을 선택하는 마스크 값입니다. 이 값의 각 비트 별 값의 의미는 아래 표와 같습니다.

Bit Number	Meaning
BIT00	0 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT01	1 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT02	2 번 축의 포함 여부 (0: 포함안함, 1: 포함)
:	
BIT31	31 번 축의 포함 여부 (0: 포함안함, 1: 포함)

- ▶ AxisMask2 : 32 ~ 63 번 축 중에서 LIST-Motion 에 참여하는 축을 선택하는 마스크 값입니다. 이 값의 각 비트 별 값의 의미는 아래 표와 같습니다.

Bit Number	Meaning
BIT00	32 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT01	33 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT02	34 번 축의 포함 여부 (0: 포함안함, 1: 포함)
:	
BIT31	63 번 축의 포함 여부 (0: 포함안함, 1: 포함)

▶ **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.

**REFERENCE**

이 함수를 실행한 이후에 리스트모션에 포함된 축들의 이송함수를 실행하면 해당 이송 명령은 리스트모션 테이블에 등록되게 됩니다. 그리고 실제 이송은 ecmLmCtl\_Run() 함수가 실행된 이후에 수행됩니다.

ecmSxMot\_VMoveStart() 함수와 같이 목표위치가 결정되지 않는 이송명령은 리스트모션 으로 등록할 수 없습니다.

다축동시구동 함수('ecmMxMot\_' 의 접두어로 이름 지어진 이송 함수)들은 리스트모션으로 등록할 수 없습니다.

리스트모션으로 등록 가능한 함수들은 아래 표와 같습니다.

리스트 모션으로 등록 가능한 함수 명
ecmSxCfg_SetSpeedPatt
ecmSxCfg_SetJerkRatio
ecmSxMot_MoveStart
ecmSxMot_MoveToStart
ecmIxCfg_MapAxes
ecmIxCfg_UnmapAxes
ecmIxCfg_SetSpeedPatt
ecmIxCfg_SetJerkRatio
ecmIxBot_LineStart
ecmIxBot_LineToStart
ecmIxBot_ArcAng_R_Start
ecmIxBot_ArcAng_A_Start
ecmIxBot_ArcPos_R_Start
ecmIxBot_ArcPos_A_Start
ecmIxBot_Arc3P_R_Start
ecmIxBot_Arc3P_A_Start

**EXAMPLE**

---

```
C/C++
//본 예제는 ListMoiton 을 등록하고 사용하는 예제입니다.
```

---

---

```

#include "ComiEcatSdk_Api.h"
#define AXIS      0

#define LM_MAP  0

#define VEL1     100000
#define VEL2     300000
#define VEL3     100000
#define ACC      800000
#define DEC      800000

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

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

/*****
* OnListMotion: 이 함수는 ListMotion 등록 및 시작 부분
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnListMotion ()
{
    // 1) ListMotion 기능 활성화(시작) //
    ecmLmCtl_Begin (Device_Info.NetIdx, LM_MAP, (0x1 << AXIS), 0, &Error_Num);

    // StepID 를 0 으로 설정 => 이후에 등록되는 스텝들은 모두 스텝아이디가 0 으로
    부여된다. //
    ecmLmCfg_SetStepId (Device_Info.NetIdx, LM_MAP, 0, &Error_Num);
}

```

---

---

```

// 2) MOVE#1 작업 등록 //
ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS, ecmSMODE_TRAPE, 0, VEL1, VEL1,
ACC, 0, &Error_Num);
ecmSxMot_MoveToStart(Device_Info.NetIdx, AXIS, 300000, &Error_Num);

// 3) 리스트모션 실행 시작 //
ecmLmCtl_Run (Device_Info.NetIdx, AXIS, &Error_Num);





// 4) MOVE#2 작업 등록 //
ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS, ecmSMODE_TRAPE, VEL1, VEL3,
VEL2, ACC, DEC, &Error_Num);
ecmSxMot_MoveToStart(Device_Info.NetIdx, AXIS, 800000, &Error_Num);

// 5) MOVE#3 작업 등록 //
ecmSxCfg_SetSpeedPatt(Device_Info.NetIdx, AXIS, ecmSMODE_TRAPE, VEL3, 0, VEL3, 0,
DEC, &Error_Num);
// StepID 를 10 으로 설정 => 이후에 등록되는 스텝들은 모두 스텝아이디가 10 으로 부여된다.
//
ecmLmCfg_SetStepId (Device_Info.NetIdx, LM_MAP, 0, &Error_Num);
// 마지막 스텝 등록, 이 스텝의 아이디는 10 이 된다. //
ecmSxMot_MoveToStart(Device_Info.NetIdx, AXIS, 800000, &Error_Num);

// 6) 리스트모션 실행 완료 체크 //
t_i32 LmRunSts, RunCount, RunStepId;
while(1){
    LmRunSts = ecmLmSt_GetRunSts(Device_Info.NetIdx, LM_MAP, &Error_Num);
    if(LmRunSts == ecmLM_STS_RUN_COMPT || LmRunSts ==
ecmLM_STS_DISABLED){
        break; // 등록된 리스트모션의 스텝들이 모두 완료되었으므로 상태감시 완료!
    }
    ecmLmSt_GetRunStepInfo (Device_Info.NetIdx, LM_MAP, &RunCount, &RunStepId,
NULL, &Error_Num);
    TRACE(“%d 번째 스텝이 실행되고 있습니다.\n”, RunCount);
    if(RunStepId == 10){
        TRACE(“마지막 스텝이 실행되고 있습니다.\n”);
    }
    Sleep(10);
}
// 7) 리스트모션 기능 비활성화(종료) //
ecmLmCtl_End(Device_Info.NetIdx, LM_MAP, &Error_Num);
}

```

---

NAME	I N F O R M A T I O N
ecmLmCtl_End - LIST-Motion 기능 정지	 Master/Slave Function  VC++/BCB/.NET  Level 1  위험 요소 없음

## SYNOPSIS

```

❑ t_success ecmLmCtl_End ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32
*ErrCode)

```

## DESCRIPTION

이 함수는 하나의 리스트 모션맵에 대해서 리스트 모션 기능을 종료합니다. 이 함수가 호출된 이후에는 해당 리스트모션 맵에 포함되었던 축들은 이송함수가 호출되었을 때 리스트 모션 방식으로 동작하지 않고 일반적인 이송 함수로 실행되게 됩니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ LmMapIndex : 종료하고자 하는 대상 LIST-Motion 맵 인덱스.
- ▶ 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.

## EXAMPLE


\* ecmLmCtl\_Begin Examp1 참조


## NAME


ecmLmCtl\_Run  
- LIST-Motion 시작

### INFORMATION

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmLmCtl\_Run ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 하나의 리스트 모션맵에 대해서 리스트 모션으로 등록된 명령들의 실행을 시작합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ LmMapIndex : Run 하고자 하는 LIST-Motion 맵 인덱스.
- ▶ 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.


## EXAMPLE


\* ecmLmCtl\_Begin Examp1 참조


## NAME


**ecmLmCtl\_Stop**  
- LIST-Motion 일시 정지

### INFORMATION

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

□ t\_success ecmLmCtl\_Stop ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_bool IsComptCurStep, [in] t\_i32 DecelTime\_ms, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 현재 실행되고 있는 리스트모션의 실행을 일시 정지합니다.

## PARAMETER

▶ NetID : Network 번호

▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.

▶ IsComptCurStep: 이 매개 변수에는 현재 실행되고 있는 스텝을 완료하고나서 리스트모션을 일시정지할 지를 설정합니다. 이 매개 변수에 설정할 수 있는 값과 그 의미는 아래의 표와 같습니다.

숫자값	값의 의미
0	현재 진행되고 있는 스텝이 완료되지 않았더라도 현재 상태에서 즉시 정지 작업에 들어간다.
1	현재 진행되고 있는 스텝이 있으면 현재 스텝이 완전히 완료되고 나서 정지 작업에 들어간다.

▶ 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.

## EXAMPLE

\* ecmLmCtl\_Begin Examl 참조



**NAME**

ecmLmCtl\_ClearQue

- LIST-Motion 등록 된 모든 스텝 제거

**INFORMATION**

Master/Slave Function

VC++/BCB/.NET

Level 1

위험 요소 없음

**SYNOPSIS**

```

❑ t_success ecmLmCtl_ClearQue ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *ErrCode)

```

**DESCRIPTION**

이 함수는 지정한 리스트모션 맵에 대해서 해당 리스트모션 테이블에 현재 등록되어 있는 모든 스텝들을 제거합니다

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ 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.

**EXAMPLE**

\* ecmLmCtl\_Begin Examp1 참조

<h2 style="margin: 0;">NAME</h2> <p style="margin: 0;">ecmLmCfg_SetQueFullMode</p> <p style="margin: 0;">ecmLmCfg_GetQueFullMode</p> <p style="margin: 0;">- LIST-Motion 공간 모드 설정/반환</p>	<b>INFORMATION</b>
	Master/Slave Function
	VC++/BCB/.NET
	Level 1
	위험 요소 없음

## SYNOPSIS

- t\_success ecmLmCfg\_SetQueFullMode ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_i32 QueFullMode, [out] t\_i32 \*ErrCode)
- t\_i32 ecmLmCfg\_SetQueFullMode ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 리스트모션 큐 버퍼가 꽉 찬 상태에서 리스트모션에 등록되어야 할 함수가 호출 되었을 때 리스트모션 큐가 해당 함수를 등록할 수 있는 여유 공간이 확보될 때까지 기다릴지 아니면 바로 에러처리하고 리턴(Return)할 지를 결정하는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ QueFullMode: 이 매개변수에 설정할 수 있는 값과 그 의미는 아래의 표와 같다. 아래 표에서 '정의된 값' 은 ComiEcatSdk\_Def.h 파일에서 EEcmLmQueFullMode 선언문을 참고하기 바랍니다.

정의된 값	숫자값	Meaning
ecmLM_QUEFULL_SKIP	0	리스트모션 큐 버퍼가 꽉 찬 상태에서 리스트모션에 등록되어야 할 함수가 호출되면 해당 함수는 바로 리턴(return)되도록 합니다. 이때 해당 함수는 eaERR_LM_QUE_FULL(-21110) 에러코드를 반환합니다. ecmLM_QUEFULL_SKIP 은 리스트모션의 QueFullMode 의 기본값(default)입니다.
ecmLM_QUEFULL_WAIT	1	리스트모션 큐 버퍼가 꽉 찬 상태에서 리스트모션에 등록되어야 할 함수가 호출 되었을 때 해당 함수 내부에서 리스트모션 큐가 해당 함수를 등록할 수 있는 여유 공간이 확보될 때까지 기다리도록 합니다.

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

## RETURN VALUE

\* ecmLmCfg\_SetQueFullMode : 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.

\* ecmLmCfg\_SetQueFullMode : 이 리턴값은 Mode 값을 가지고 있습니다.


#### EXAMPLE


\* ecmLmCtl\_Begin Examp1 참조


**NAME**


ecmLmCfg\_SetStepId  
 ecmLmCfg\_GetStepId  
 - LIST-Motion 스텝 ID 설정/반환

**INFORMATION**

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmLmCfg\_SetStepId ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [out] t\_i32 \*ErrCode)

□ t\_i32 ecmLmCfg\_GetStepId ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 리스트모션 테이블에 등록되는 각 명령들의 스텝 아이디(ID)값을 부여하는 함수입니다

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ 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.


**EXAMPLE**


\* ecmLmCtl\_Begin Examl 참조


## NAME


ecmLmCfg\_SetStepParam  
 ecmLmCfg\_GetStepParam  
 - LIST-Motion 테이블 파라미터 정의

### INFORMATION

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

- t\_success ecmLmCfg\_SetStepParam ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_i32 StepParam1, [in] t\_i32 StepParam2, [out] t\_i32 \*ErrCode)
- t\_success ecmLmCfg\_GetStepParam ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [out] t\_dword\* StepParam1, [out] t\_dword\* StepParam2, [out] t\_i32 \*ErrCode)

## DESCRIPTION

이 함수는 각 명령 스텝들이 리스트모션 테이블에 등록될 때 함께 등록할 2개의 데이터를 정의하는 것입니다. 이 2개의 데이터 값은 추후에 해당 스텝이 실행될 때 그 값을 참고할 수 있습니다.

이 함수에서 정의하는 스텝 파라미터는 이 후에 다시 그 값을 변경하기 전까지 이후에 등록되는 모든 스텝에 대해서 같은 파라미터값 들이 적용됩니다.

이 함수는 특별히 부가적인 데이터가 함께 저장되어야 할 필요가 있는 스텝에 대해서만 적용하면 됩니다.

## PARAMETER





- ▶ NetID : Network 번호
- ▶ LmMapIndex : 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ StepParam1 : 다음 스텝과 함께 저장하고자 하는 첫 번째 32 비트 파라미터.
- ▶ StepParam2 : 다음 스텝과 함께 저장하고자 하는 첫 번째 32 비트 파라미터.
- ▶ 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.
---	-----------------------

<h1>NAME</h1> <p><b>ecmLmSt_GetRunSts</b></p> <p>- LIST-Motion 테이블 파라미터 정의</p>	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_i32 ecmLmSt\_GetRunSts ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 LmMapIndex 로 지정된 리스트모션 맵의 동작상태를 반환합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ ErrCode : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 NULL 을 전달하면 에러 코드를 반환하지 않습니다.

### RETURN VALUE

\* LmMapIndex 로 지정된 리스트모션 맵의 동작상태를 반환합니다. 이 값은 아래 표에서 나타낸 값 들 중의 하나가 됩니다. 아래 표에서 '정의된 값 ' 은 ComiEcatSdk\_Def.h 파일에서 EEcmLmSts 선언문을 참고하기 바랍니다.


정의된 값	숫자값	값의 의미
ecmLM_STS_DISABLED	0	리스트모션이 종료된 상태. 즉, ecmLmCtl_Begin() 함수가 아직 실행되기 전이거나, ecmLmCtl_End() 함수가 실행된 상태임.
ecmLM_STS_PAUSED	1	List Motion 기능이 활성화(LmCtl_Begin()) 되었지만 아직 Run (LmCtl_Run()) 되지 않은 상태
ecmLM_STS_RUN_IDLE	2	List Motion 기능이 활성화(LmCtl_Begin()) 되고, 또한 Run (LmCtl_Run()) 된 상태이지만 아직 등록된 커맨드가 하나도 없는 경우.
ecmLM_STS_RUN_BUSY	3	List Motion 기능이 활성화(LmCtl_Begin()) 되고, 또한 Run (LmCtl_Run()) 된 상태에서 등록된 커맨드가 실행중에 있는 경우.
ecmLM_STS_RUN_COMPT	4	List Motion 기능이 활성화(LmCtl_Begin()) 되고, 또한 Run (LmCtl_Run()) 된 상태에서 등록된 커맨드가 모두 실행된 경우(참고로 ecmLM_STS_RUN_COMPT 은 1 개 이상의 등록된 커맨드가 있었던 경우에 해당하며, 등록된 커맨드가 아예 없는 경우는 ecmLM_STS_RUN_IDLE 상태로 반환됩니다)


## NAME

`ecmLmSt_GetRemStepCount`


- 테이블에 남아 있는 명령의 수 반환

### INFORMATION

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

`int_t ecmLmSt_GetRemStepCount ([in] int_t NetID, [in] int_t LmMapIndex, [out] int_t *ErrCode)`

## DESCRIPTION

이 함수는 리스트모션 테이블에 등록된 명령 스텝들 중에 실행되지 않고 남아 있는 명령의 수를 반환합니다.

## PARAMETER

- ▶ `NetID` : Network 번호
- ▶ `LmMapIndex` : 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ `ErrCode` : 이 매개 변수를 통하여 현재의 에러 코드를 반환합니다. 단, 이 매개 변수에 `NULL` 을 전달하면 에러 코드를 반환하지 않습니다.

## RETURN VALUE


\* `LmMapIndex` 로 지정된 리스트모션 맵의 테이블에 등록되었지만 아직 실행되지 않은 명령들의 개수를 반환합니다





## NAME


`ecmLmSt_GetRunStepInfo`  
- LIST-Motion 스텝 정보 반환

### INFORMATION

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

```
□ t_success ecmLmSt_GetRunStepInfo ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_i32 *RunStepCount, [out] t_i32 *RunStepId, [out] t_i32 *RunStepSts, [out] t_i32 *ErrCode)
```

## DESCRIPTION

이 함수는 현재 실행되고 있는 스텝에 대한 정보를 반환하는 함수입니다.

## PARAMETER

▶ NetID : Network 번호

▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.

▶ RunStepCount: 이 매개 변수를 통해서 현재 실행되고 있는 스텝의 스텝 카운트를 반환합니다. 스텝 카운트는 해당 스텝이 리스트모션 테이블에 등록된 순서를 말합니다. 스텝카운트 값은 1 부터 매겨지며, 명령이 하나씩 등록될 때마다 자동으로 1 씩 증가합니다.

▶ RunStepId : 이 매개를 통해서 현재 실행되고 있는 스텝의 스텝 아이디를 반환합니다. 스텝 아이디는 사용자가 해당 스텝에 부여하는 숫자값으로서 `ecmLmCfg_SetStepId()` 함수를 통해서 정의됩니다.

▶ RunStepSts: 이 매개를 통해서 현재 실행 스텝의 실행 상태를 반환합니다. 이 값을 통해서 반환되는 값은 아래의 표에 나타내었습니다. 아래 표에서 '정의된 값' 은 `ComiEcatSdk_Def.h` 파일에서 `EEcmLmCmdItemSts` 선언문을 참고하기 바랍니다.

정의된 값	숫자값	값의 의미
<code>ecmLM_CMDITEM_STS_READY</code>	0	해당 스텝이 실행되기 위한 준비를 하고 있는 상태임을 나타냅니다.
<code>ecmLM_CMDITEM_STS_BUSY</code>	1	해당 스텝이 현재 실행 중에 있음을 나타냅니다.
<code>ecmLM_CMDITEM_STS_PAUSED</code>	2	해당 스텝이 실행 중에 일시 정지된 상태임을 나타냅니다.
<code>ecmLM_CMDITEM_STS_COMPLETED</code>	3	해당 스텝의 실행이 완료되었음을 나타냅니다.

▶ 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.


## NAME


`ecmLmSt_GetRunStepParam`  
- LIST-Motion 파라미터 값 반환

### INFORMATION

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

## SYNOPSIS

`int t_success ecmLmSt_GetRunStepParam ([in] t_i32 NetID, [in] t_i32 LmMapIndex, [out] t_dword *StepParam1, [out] t_dword *StepParam2, [out] t_i32 *ErrCode)`

## DESCRIPTION

이 함수는 현재 실행되고 있는 스텝의 스텝 파라미터값들을 반환하는 함수입니다. 스텝 파라미터값은 사용자가 해당 스텝을 등록할 때에 함께 등록한 2 개의 32 비트 정수 값을 말합니다. 스텝 파라미터는 필요에 따라서 해당 스텝과 함께 참고해야 할 데이터가 있는 경우에 사용하는 데이터이며 꼭 사용해야 하는 것은 아닙니다.

## PARAMETER

- ▶ `NetID` : Network 번호
- ▶ `LmMapIndex`: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ `StepParam1`: 해당 스텝이 등록될 당시에 정의된 스텝 파라미터 1 에 해당 데이터를 반환합니다. 이와 관련해서는 `ecmLmCfg_SetStepParam()` 함수를 참고하시기 바랍니다.
- ▶ `StepParam2`: 해당 스텝이 등록될 당시에 정의된 스텝 파라미터 2 에 해당 데이터를 반환합니다. 이와 관련해서는 `ecmLmCfg_SetStepParam()` 함수를 참고하시기 바랍니다.
- ▶ `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.

**NAME**

eclmCmd\_Delay

- LIST-Motion 실행 시간 적용

**INFORMATION**

Master/Slave Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

**SYNOPSIS**

□ t\_success eclmCmd\_Delay ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_i32 DelayTime\_ms, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 리스트모션 실행 시에 시간 적용되어야 할 필요가 있는 경우에 사용될 수 있도록 시간 지연 명령을 리스트모션 테이블에 등록하는 것입니다.

리스트모션 실행 시에 이 명령이 실행되면 해당 시간만큼 리스트모션의 실행을 지연하게 됩니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ LmMapIndex: 이 매개 변수에는 리스트 모션의 맵 번호를 전달합니다.
- ▶ DelayTime\_ms: 지연하고자 하는 시간을 msec 단위로 설정합니다
- ▶ 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.

## NAME

ecmLmResetAll

- LIST-Motion 실행 시간 적용

### INFORMATION

Master/Slave Function

VC++/BCB/.NET

Level 1

위험 요소 없음

## SYNOPSIS

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

## DESCRIPTION

이 함수는

## 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 불 형(Boolean Type) 을 가지고 있습니다.


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


**NAME**


SetQueDepth


GetQueDepth

-

**INFORMATION**
 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

□ t\_success ecmLmCfg\_SetQueDepth ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [in] t\_i32 QueDepth, [out] t\_i32 \*ErrCode)

□ t\_success ecmLmCfg\_GetQueDepth ([in] t\_i32 NetID, [in] t\_i32 LmMapIndex, [out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는

**PARAMETER**

▶ NetID : Network 번호

▶ LmMapIndex :

▶ QueDepth :

▶ 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.

# Motion PT Motion 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.







## 20 Motion PT Motion Functions

### 20.1 함수 요약

Summary of Functions
<p>□ t_success ecmPtmCtl_Begin (t_i32 NetID, t_i32 PtmMapIndex, t_ui32 AxisMask1, t_ui32 AxisMask2, t_i32 *ErrCode) PtmMapIndex 로 지정된 PT-Motion 맵에 대해서 PT-Motion 기능을 시작하는 함수입니다.</p>
<p>□ t_success ecmPtmCtl_End (t_i32 NetID, t_i32 PtmMapIndex, t_i32 *ErrCode) PtmMapIndex 로 지정된 PT-Motion 맵에 대해서 PT-Motion 기능을 종료하는 함수입니다.</p>
<p>□ t_success ecmPtmCtl_SetHold (t_i32 NetID, t_i32 PtmMapIndex, t_bool IsHold, t_i32 *ErrCode) PtmMapIndex 로 지정된 PT-Motion 맵의 Hold 상태를 셋팅하는 함수입니다.</p>
<p>□ t_success ecmPtmCfg_AddItem_PT (t_i32 NetID, t_i32 PtmMapIndex, t_f64 DurTime, t_f64 PosDataList[], t_i32 NumPosData, t_bool IsAbsPos, t_i32 *ErrCode) PtmMapIndex 로 지정된 PT-Motion 맵에 위치(Position)와 시간(Time) 정보로 구성되는 이송 스텝을 등록합니다.</p>
<p>□ EEcmPtmSts ecmPtmSt_GetRunSts (t_i32 NetID, t_i32 PtmMapIndex, t_i32 *RunStepCount, t_i32 *RemStepCount, t_i32 *ErrCode) PtmMapIndex 로 지정된 PT-Motion 맵의 동작상태를 반환합니다.</p>
<p>□ t_success ecmPtmSt_GetRunStepInfo (t_i32 NetID, t_i32 PtmMapIndex, t_i32 *RunStepCount, t_i32 *RunStepSts, t_i32 *ErrCode) PtmMapIndex 로 지정된 PT-Motion 맵의 동작상태를 반환합니다.</p>
<p>□ t_success ecmPtmResetAll([in] t_i32 NetID, [out] t_i32 *ErrCode);</p>



**20.2** 함수 설명

<b>NAME</b>  ecmPtmCtl_Begin  - PT-Motion 기능 시작	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

**SYNOPSIS**

□ t\_success ecmPtmCtl\_Begin ([in] t\_i32 NetID, [in] t\_i32 PtmMapIndex, [in] t\_ui32 AxisMask1, [in] t\_ui32 AxisMask2,[out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 PtmMapIndex 로 지정된 PT-Motion 맵에 대해서 PT-Motion 기능을 시작하는 함수. 이 함수가 호출된 이후에 ecmPtmCfg\_AddItem\_PT() 함수를 실행하면 등록된 Position-Table 값에 해당하는 이송이 이루어진다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ PtmMapIndex : 시작하고자 하는 대상 PT-Motion 맵 인덱스. PT-Motion 맵은 최대 8 개까지 지원되므로 이 값은 0 ~ 7 의 숫자이어야 합니다.
- ▶ AxisMask1 : 0 ~ 31 번 축 중에서 PT-Motion 에 참여하는 축을 선택하는 마스크 값입니다. 이 값의 각 비트 별 값의 의미는 아래 표와 같습니다.

Bit Number	Meaning
BIT00	0 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT01	1 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT02	2 번 축의 포함 여부 (0: 포함안함, 1: 포함)
:	
BIT31	31 번 축의 포함 여부 (0: 포함안함, 1: 포함)

- ▶ AxisMask2 : 32 ~ 63 번 축 중에서 PT-Motion 에 참여하는 축을 선택하는 마스크 값입니다. 이 값의 각 비트 별 값의 의미는 아래 표와 같습니다.

Bit Number	Meaning
BIT00	32 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT01	33 번 축의 포함 여부 (0: 포함안함, 1: 포함)
BIT02	34 번 축의 포함 여부 (0: 포함안함, 1: 포함)
:	
BIT31	63 번 축의 포함 여부 (0: 포함안함, 1: 포함)

▶ **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.

EXAMPLE

---

```

C/C++
//본 예제는 PT Motion 관련 예제입니다.
#include "ComiEcatSdk_Api.h"

// SetBit() : 지정한 변수의 특정 비트의 값을 변경한다. 매크로이므로 함수명과 괄호가
떨어지면 않됨에 주의!
// v:대상 변수, n:bit 번호, s:변경값(state)
#define ecSetBit(v,n,s) if(s) (v) |= (1 << (n)); else (v) &= ~(1 << (n))

#define AXIS      0
#define PTM-MAP  0
#define VEL_HIGH  500000.0
#define ACC_TIME  100
#define DEC_TIME  100
#define VWORK_TIME  2000
#define DELTA_T   0.001 // Delta-T (sec)

#define NUM_AXES  3
int g_nAxisList[NUM_AXES] = {0, 1, 2};

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

    ecmSxCtl_SetSvon(Device_Info.NetIdx, AXIS0, &Error_Num);

} OnProgramInitial 함수 끝

/*****
* OnPTMotion: 이 함수는 PT 모션 등록 및 시작 부분
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnPTMotion ()
{
    t_f64 WorkVelList[NUM_AXES] = {300000, 400000, 500000};
    t_f64 Acc[NUM_AXES], Dec[NUM_AXES];
    t_f64 PosDataList[NUM_AXES];
    t_f64 InstVel;
    t_i32 RegStepCount = 0;
    t_i32 RemStepCount = 0;
    t_i32 RunStepCount = 0, RunStepSts = 0;
    t_ui32 g_dwAxisMask1, g_dwAxisMask2;

    // 1) PT-Motion 맵번호 //
    int nPtmMapIndex = PTM-MAP;

    // 2) 각 축의 가속도와 감속도 계산 //
    g_dwAxisMask1 = g_dwAxisMask2 = 0; // 축 마스크 리셋
    for(i=0; i < NUM_AXES; i++){
        Acc[i] = (WorkVelList[i] / (double)ACC_TIME);
        Dec[i] = (WorkVelList[i] / (double)DEC_TIME);
        // 축마스크 비트 셋팅 //
        if(g_nAxisList[i] < 32){ // 축번호 0~31 은 AxisMask1 을 통해서 전달.
            ecSetBit(g_dwAxisMask1, g_nAxisList[i], 1);
        }else{ // 축번호 32~63 은 AxisMask2 을 통해서 전달.
            ecSetBit(g_dwAxisMask2, (g_nAxisList[i]-32), 1);
        }
    }
}

// 3) PT-Motion 활성화 //
if(!ecmPtmCtl_Begin(Device_Info.NetIdx, nPtmMapIndex, g_dwAxisMask1,
g_dwAxisMask2, &Error_Num)){
    // 실패시 에러 처리
}

// 4) 가속구간의 Position Table 등록 //
for(i=0; i < ACC_TIME; i++){
    for(int j=0; j < NUM_AXES; j++){
        InstVel = (i+1) * Acc[j];
        // 가속도를 반영하여 각 미소구간에서 적용되어야 하는 평균 속도 계산
        PosDataList[j] = InstVel * DELTA_T; // 평균 속도에 미소시간을 곱하여 해당
        구간에서 이송해야 하는 이송 거리 계산
    }
}

```

---

---

```

    }
    // Position Table 등록 //
    ecmPtmCmd_AddItem_PT(Device_Info.NetIdx, nPtmMapIndex, DELTA_T,
    PosDataList, NUM_AXES, FALSE, & Error_Num);
    }

    // 5) 정속구간의 Position Table 등록 //
    for(i=0; i < VWORK_TIME; i++){
        for(int j=0; j < NUM_AXES; j++){
            PosDataList[j] = WorkVelList[j] * DELTA_T; // 평균 속도에 미소시간을 곱하여 해당
            구간에서 이송해야 하는 이송 거리 계산
        }
    }
    // Position Table 등록 //
    ecmPtmCmd_AddItem_PT(Device_Info.NetIdx, nPtmMapIndex, DELTA_T, PosDataList,
    NUM_AXES, FALSE, & Error_Num);
    }

    // 6) 감속구간의 Position Table 등록 //
    for(i=0; i < ACC_TIME; i++){
        for(int j=0; j < NUM_AXES; j++){
            InstVel = WorkVelList[j] - ((i+1) * Dec[j]);
            // 가속도를 반영하여 각 미소구간에서 적용되어야 하는 평균 속도 계산
            PosDataList[j] = InstVel * DELTA_T;
            // 평균 속도에 미소시간을 곱하여 해당 구간에서 이송해야 하는 이송 거리 계산
        }
    }
    // Position Table 등록 //
    ecmPtmCmd_AddItem_PT(Device_Info.NetIdx, nPtmMapIndex, DELTA_T,
    PosDataList, NUM_AXES, FALSE, & Error_Num);
    }

    // 7) PT-Motion 완료 체크 //
    MSG msg;
    EEcmPtmSts RunSts;
    while(1){
        RunSts = ecmPtmSt_GetRunSts(Device_Info.NetIdx, nPtmMapIndex, NULL, NULL,
    & Error_Num);
        if(RunSts == ecmPTM_STS_RUN_COMPT || RunSts ==
    ecmPTM_STS_DISABLED){
            break; // 등록된 PT 모션이 모두 완료되었으므로 상태감시 완료!
        }
        Sleep(10);

        while(PeekMessage(&msg, NULL, 0, 0, PM_REMOVE)){
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    // 8) PT-Motion 종료 //
    if(!ecmPtmCtl_End(Device_Info.NetIdx, nPtmMapIndex, &Error_Num)){
        ecUtl_ShowError_A (GetSafeHwnd(), "ecmPtmCtl_End() functions fail!, ec = %d",
    nErrCode);
        return;
    }
}

```

---

---

```

}

/*****
* OnPTHOLD: 이 함수는 PT 모션 HOLD 하는 부분
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnPTHOLD ()
{
    t_bool isHold = 0; //Hold 상태
    if(!ecmPtmCtl_SetHold(Device_Info.NetIdx, nPtmMapIndex, isHold, &Error_Num))
    {
        //실패시 에러 처리
    }
}

/*****
* OnPTEnd: 이 함수는 PT 모션 종료 하는 부분
* 많은 세팅 관련 함수가 있지만 경우에 따라 사용자가 선택해서 사용
*****/
void OnPTEnd ()
{
    if(!ecmPtmCtl_End(Device_Info.NetIdx, PTM-MAP, &Error_Num))
    {
        //실패시 에러 처리
    }
}

```

---

<h2>NAME</h2> <p><code>ecmPtmCtl_End</code></p> <p>- PT-Motion 기능 정지</p>	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

```

❑ t_success ecmPtmCtl_End ([in] t_i32 NetID,[in] t_i32 PtmMapIndex,[out] t_i32
*ErrCode)

```

## DESCRIPTION

이 함수는 PtmMapIndex 로 지정된 PT-Motion 맵에 대해서 PT-Motion 기능을 종료합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ PtmMapIndex : 종료하고자 하는 대상 PT-Motion 맵 인덱스.
- ▶ 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.

## EXAMPLE


\* `ecmPtmCtl_Begin` Exmale 참고


**NAME**


ecmPtmCtl\_SetHold  
- PT-Motion Hold 설정

**INFORMATION**

 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음

**SYNOPSIS**

□ t\_success ecmPtmCtl\_SetHold ([in] t\_i32 NetID,[in] t\_i32 PtmMapIndex,[in] t\_bool IsHold,[out] t\_i32 \*ErrCode)

**DESCRIPTION**

이 함수는 PtmMapIndex 로 지정된 PT-Motion 맵의 Hold 상태를 셋팅하는 함수입니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ PtmMapIndex : Hold 상태를 설정하고자 하는 대상 PT-Motion 맵 인덱스.
- ▶ IsHold : Hold 상태를 설정합니다.

Value	Meaning
0	해당 PT-Motion 맵의 이송이 중지 됩니다
1	해당 PT-Motion 맵의 이송이 다시 재개 됩니다.

▶ 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.

**EXAMPLE**

\* ecmPtmCtl\_Begin Exmale 참고

<b>NAME</b>  ecmPtmCfg_AddItem_PT  - PT-Motion 이송스텝 등록	<b>I N F O R M A T I O N</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

## SYNOPSIS

```

□ t_success ecmPtmCfg_AddItem_PT ([in] t_i32 NetID,[in] t_i32 PtmMapIndex,[in] t_f64
DurTime,[in] t_f64 PosDataList[],[in] t_i32 NumPosData,[in] t_bool IsAbsPos,[out] t_i32
*ErrCode)

```

## DESCRIPTION

이 함수는 PtmMapIndex 로 지정된 PT-Motion 맵에 위치(Position)와 시간(Time) 정보로 구성되는 이송스텝을 등록합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ PtmMapIndex : 이송스텝을 등록하고자 하는 대상 PT-Motion 맵 인덱스.
- ▶ DurTime : 이송스텝의 지속시간(Duration Time)을 sec 단위로 설정합니다. 각 축은 이 값으로 지정된 시간 동안 PosDataList 배열에서 지정한 이송 거리를 이송하게 됩니다..
- ▶ PosDataList : 각 축이 이번 스텝에서 이송해야 하는 이송거리 또는 목표좌표값을 배열로 전달하는 매개변수입니다.  
ecmPtmCtl\_Begin () 함수를 통해서 설정한 축의 수 만큼에 해당하는 위치데이터가 이 배열을 통해서 전달 되어야 합니다
- ▶ NumPosData : PosDataList 배열을 통해서 전달되는 위치데이터가 실제로 몇 개인지를 지정합니다.  
이 값이 ecmPtmCtl\_Begin () 함수를 통해서 설정 한 축의 수와 다를 경우에는 에러로 처리됩니다.
- ▶ IsAbsPos : PosDataList 배열을 통해서 전달되는 위치데이터가 상대좌표(거리)인지 아니면 절대좌표(목표좌표)인지를 설정 합니다..

Value	Meaning
0	PosDatList 배열의 위치 데이터가 상대 좌표 입니다.
1	PosDatList 배열의 위치 데이터가 절대 좌표 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 불 형(Boolean Type) 을 가지고 있습니다.

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

## EXAMPLE

\* ecmPtmCtl\_Begin Exmale 참고

<h1>NAME</h1> <p><b>ecmPtmSt_GetRunSts</b></p> <p>- PT-Motion 동작 상태 반환</p>	<b>INFORMATION</b>
	Master/Slave Function
	VC++/BCB/.NET
	Level 1
위험 요소 없음	

## SYNOPSIS

□ EEcmPtmSts ecmPtmSt\_GetRunSts ([in] t\_i32 NetID,[in] t\_i32 PtmMapIndex,[out] t\_i32 \*RunStepCount,[out] t\_i32 \*RemStepCount,[out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 PtmMapIndex 로 지정된 PT-Motion 맵의 동작상태를 반환합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ PtmMapIndex: 이송스텝을 등록하고자 하는 대상 PT-Motion 맵 인덱스.
- ▶ RunStepCount : 이 포인터를 통해서 “Run Step Count” 값을 반환합니다.

Value	Meaning
Run Step Count	사용자가 등록한 PT(또는 PV)이송스텝 중에서 현재까지 실행완료 되었거나 실행 중인 스텝의 개수를 의미 합니다. “Run Step Count” 값을 반환 받지 않으려면 이 매개변수의 값을 NULL 로 설정한다.

- ▶ RemStepCount : 이 포인터를 통해서 “Remained Step Count” 값을 반환 합니다.

Value	Meaning
Remained Step Count	사용자가 등록한 PT(또는 PV)이송스텝 중에서 아직까지 실행되지 않고 대기 중인 스텝의 개수를 말합니다. “Remained Step Count” 값을 반환 받지 않으려면 이 매개변수의 값을 NULL 로 설정합니다.

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





### RETURN VALUE

\* PtmMapIndex 로 지정된 PT-Motion 맵의 동작상태를 반환합니다. 이 값은 아래 표에서 나타낸 값 들 중의 하나가 됩니다.

Value	숫자값	Meaning
ecmPTM_STS_DISABLED	0	PT-Motion 이 종료된 상태. ecmPtmCtl_Begin() 함수가 아직 실행되기 전이거나, ecmPtmCtl_End() 함수가 실행된 상태임.
ecmPTM_STS_PAUSED	1	PT-Motion 기능이 활성화 되었지만 Hold 된 상태. 이 상태에서는 스텝의 등록은 가능하지만 이송은 진행되지 않는 경우
ecmPTM_STS_RUN_IDLE	2	PT-Motion 기능이 활성화 되었지만 아직 등록된 커맨드가 하나도 없는 경우.
ecmPTM_STS_RUN_BUSY	3	PT-Motion 기능이 활성화 되고, 등록된 커맨드가 실행 중에 있는 경우.
ecmPTM_STS_RUN_COMPT	4	PT-Motion 기능이 활성화된 상태에서 등록된 커맨드가 모두 실행 된 경우.

## EXAMPLE

\* ecmPtmCtl\_Begin Exmale 참고

<h1>NAME</h1> <p><b>ecmPtmSt_GetRunStepInfo</b> - PT-Motion 동작 상태 반환</p>	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

t\_success ecmPtmSt\_GetRunStepInfo ([in] t\_i32 NetID,[in] t\_i32 PtmMapIndex,[out] t\_i32 \*RunStepCount,[out] t\_i32 \*RunStepSts,[out] t\_i32 \*ErrCode)

### DESCRIPTION

이 함수는 PtmMapIndex 로 지정된 PT-Motion 맵의 동작상태를 반환합니다.

### PARAMETER

- ▶ NetID : Network 번호
- ▶ PtmMapIndex: 이송스텝을 등록하고자 하는 대상 PT-Motion 맵 인덱스.
- ▶ RunStepCount : 이 포인터를 통해서 현재 실행되고 있는 이송스텝의 일련번호를 반환한다. 이송스텝의 일련번호는 1 부터 매겨지는 스텝의 등록 순서를 말합니다. 일련 번호를 반환 받지 않으려면 이 매개변수의 값을 NULL 로 설정합니다..
- ▶ RunStepSts : 이 포인터를 통해서 현재 실행되고 있는 이송스텝의 명령 수행 상태를 반환합니다. 명령 수행 상태 값을 반환 받지 않으려면 이 매개변수의 값을 NULL 로 설정합니다.  
이송스텝의 명령 수행 상태 값의 의미는 아래 표와 같습니다.

Value	숫자값	Meaning
ecmPTM_CMDITEM_STS_READY	0	해당 이송 스텝이 실행을 대기하고 있는 상태
ecmPTM_CMDITEM_STS_BUSY	1	해당 이송 스텝이 이송을 진행하고 있는 상태
ecmPTM_CMDITEM_STS_PAUSED	2	해당 이송 스텝의 수행이 Hold 된 상태
ecmPTM_CMDITEM_STS_COMPLETED	3	해당 이송 스텝의 수행이 완료된 상태.

- ▶ 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.




EXAMPLE

\* ecmPtmCtl\_Begin Exmale 참고

**NAME**

ecmPtmResetAll

-

**INFORMATION** Master/Slave Function VC++/BCB/.NET Level 1 위험 요소 없음**SYNOPSIS**

```

❑ t_success ecmPtmResetAll([in] t_i32 NetID, [out] t_i32 *ErrCode)

```

**DESCRIPTION****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****EXAMPLE**



# Motion Motor Performance monitor 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.





## 21 Motion Motor Performance Monitor Functions

### 21.1 함수 요약

Summary of Functions
□ t_success ecmMPMon_AddAxis ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_DelAxis ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_SetAxisEnable ([in] (t_i32 NetID, [in] t_i32 Axis, [in] t_bool IsEnable, [out] t_i32 *ErrCode) 설명
□ t_i32 ecmMPMon_GetAxisStatus ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_ResetAxisData ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_ResetSectData ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 SectIdx, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_SetTorqThreshVal ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 SectIdx, [in] t_ui16 TorqThresholdVal, [out] t_i32 *ErrCode) 설명
□ t_ui16 ecmMPMon_GetTorqThreshVal ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 SectIdx, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_SetDevThreshVal ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 SectIdx, [in] t_ui32 DevThresholdVal, [out] t_i32 *ErrCode) 설명
□ t_ui32 ecmMPMon_GetDevThreshVal ([in] t_i32 NetID, [in] t_i32 Axis, [in] t_ui16 SectIdx, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_ReadAxisData ([in] t_i32 NetID, [in] t_i32 Axis, [in] TMPMonData *pBuffer, [out] t_i32 *ErrCode) 설명
□ t_success ecmMPMon_WriteAxisData ([in] t_i32 NetID, [in] t_i32 Axis, [in] TMPMonData *pMPMonData, [out] t_i32 *ErrCode) 설명

**21.2** 함수 설명

<b>NAME</b>	<b>INFORMATION</b>
ecmMPMon_AddAxis	 Master/Slave Function
- MPMon 관련 객체를 추가	 VC++/BCB/.NET
	 Level 1
	 위험 요소 없음

**SYNOPSIS**

`int t_success ecmMPMon_AddAxis ([in] t_i32 NetID, [in] t_i32 Axis, [out] t_i32 *ErrCode)`

**DESCRIPTION**

MPMon 리스트에 해당 축의 MPMon 관련 객체를 추가합니다.

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ Axis : MPMon 리스트에 추가할 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	함수 수행이 실패 했음을 의미 합니다.
1	함수 수행이 성공 했음을 의미 합니다.

**REFERENCE**

- 이 함수는 MPMon 리스트에 해당 축 객체를 추가하는 것입니다. 따라서 ecmMPMon\_GetStatus() 함수를 제외한 다른 MPMon 관련함수를 실행하기 전에 이 함수가 가장 먼저 선행되어야 합니다
- 이 함수를 실행해도 실제 Monitor 작업을 시작하는 것은 아닙니다. 실제 Monitor 작업을 시작하려면 이 함수를 실행한 이후에 IsEnable 매개변수의 값을 true 로 하여 ecmMPMon\_SetAxisEnable() 함수를 실행해야 합니다.
- 이 함수를 실행하면 해당 축의 MPMon 관련 데이터들은 초기화됩니다.

<h2>NAME</h2> <p>ecmMPMon_DelAxis</p> <p>- MPMon 리스트에서 해당 축의 MPMon 관련 객체를 삭제</p>	<h3>INFORMATION</h3> <ul style="list-style-type: none"> <li> Master/Slave Function</li> <li> VC++/BCB/.NET</li> <li> Level 1</li> <li> 위험 요소 없음</li> </ul>
--	--

## SYNOPSIS

□ t\_success ecmMPMon\_DelAxis ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_i32 \*ErrCode)

### DESCRIPTION

MPMon 리스트에서 해당 축의 MPMon 관련 객체를 삭제합니다.





### 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	함수 수행이 실패 했음을 의미 합니다.
1	함수 수행이 성공 했음을 의미 합니다.

<b>NAME</b>  ecmMPMon_SetAxisEnable  -	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_success ecmMPMon\_SetAxisEnable ([in] t\_i32 NetID, [in] t\_i32 Axis, [in] t\_bool IsEnable, [out] t\_i32 \*ErrCode)

## DESCRIPTION

하나의 축에 대해서 MPMon 기능을 활성화 또는 비활성화합니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ IsEnable : 활성화/비활성화 옵션 값  
0 - 해당 축의 MPMon 기능을 비활성화합니다.  
1 - 해당 축의 MPMon 기능이 비활성화합니다
- ▶ 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.

## EXAMPLE

- 이 함수를 실행하기 전에 ecmMPMon\_AddAxis() 함수가 선행되어야 합니다.
- 이 함수를 실행해도 데이터들은 초기화되지 않습니다. 데이터들을 초기화하려면 ecmMPMon\_ResetAxis() 함수를 실행하십시오.

<b>NAME</b>  ecmMPMon_GetAxisStatus  -	<b>I N F O R M A T I O N</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_i32 ecmMPMon\_GetAxisStatus (t\_i32 NetID, t\_i32 Axis, t\_i32 \*ErrCode)

### DESCRIPTION

하나의 축에 대해서 현재의 MPMon 기능과 관련한 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

\* 해당 축의 MPMon 기능과 관련한 상태를 반환합니다. 이 반환값의 의미는 아래의 표와 같습니다.

Value	Meaning
-1	해당 축이 MPMon 리스트에 등록되지 않은 상태(ecmMPMon_AddAxis() 함수를 실행하기 전 상태)를 의미합니다
0	해당 축이 MPMon 리스트에 등록은 되었으나 활성화되지 않은 상태(ecmMPMon_SetEnable(true)를 실행하지 않은 상태)를 의미합니다.
1	해당 축이 MPMon 리스트에 등록이 되었으며, 활성화도 되어 있는 상태를 의미합니다.

<b>NAME</b>  ecmMPMon_ResetAxisData  -	<b>INFORMATION</b>
	 Master/Slave Function
	 VC++/BCB/.NET
	 Level 1
 위험 요소 없음	

## SYNOPSIS

□ t\_success ecmMPMon\_ResetAxisData (t\_i32 NetID, t\_i32 Axis, t\_i32 \*ErrCode)

### DESCRIPTION

하나의 축에 대해서 MPMon 관련 데이터들을 초기화합니다.

### 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

\* 함수 수행의 성공 여부

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

**NAME**

ecmMPMon\_ResetSectData

-

**INFORMATION**

Master/Slave Function

VC++/BCB/.NET

Level 1

☺ 위험 요소 없음

**SYNOPSIS**

```

❑ t_success ecmMPMon_ResetSectData (t_i32 NetID, t_i32 Axis, t_ui16 SectIdx, t_i32
*ErrCode)

```

**DESCRIPTION**

특정 축의 특정 MPMon 섹션의 데이터들을 초기화합니다.

**PARAMETER**

▶ NetID : Network 번호

▶ Axis : 대상 축 번호

▶ SectIdx : MPMon 섹션 인덱스. 섹션 인덱스는 0 ~ 5 의 숫자를 입력할 수 있는데, ComiEcatSdk\_Def.h 파일에 서 아래와 같이 정의한 값을 사용할 수 있습니다.

```

enum EMPMonSectID{
    MPMSECT_ACC, /// < 0: 가속구간만을 누적하여 계산하는 구간
    MPMSECT_DEC, /// < 1: 감속구간만을 누적하여 계산하는 구간
    MPMSECT_CONST, /// < 2: 정속구간만을 누적하여 계산하는 구간
    MPMSECT_MOVING, /// < 3: 이송중인구간을 종합하여 계산하는 구간
    MPMSECT_IDLE, /// < 4: 정지해 있는 구간만 누적하여 계산하는 구간
    MPMSECT_TOTAL, /// < 5: MOVING 과 IDLE 을 합친 모든 구간
    MPMSECT_INVALID
};

```

▶ 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 수행이 실패 했음을 의미 합니다.
1	함수 수행이 성공 했음을 의미 합니다.

EXAMPLE

\*



NAME	INFORMATION
ecmMPMon_SetTorqThreshVal ecmMPMon_GetTorqThreshVal -	<div data-bbox="1102 349 1402 383">  Master/Slave Function         </div> <div data-bbox="1102 394 1402 427">  VC++/BCB/.NET         </div> <div data-bbox="1102 439 1402 472">  Level 1         </div> <div data-bbox="1102 483 1402 517">  위험 요소 없음         </div>

## SYNOPSIS

- t\_success ecmMPMon\_SetTorqThreshVal (t\_i32 NetID, t\_i32 Axis, t\_ui16 SectIdx, t\_ui16 TorqThresholdVal, t\_i32 \*ErrCode)
- t\_ui16 ecmMPMon\_GetTorqThreshVal (t\_i32 NetID, t\_i32 Axis, t\_ui16 SectIdx, t\_i32 \*ErrCode)

## DESCRIPTION

특정 축의 특정 섹션에 대해서 Torque Threshold 값을 설정합니다

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ SectIdx : 섹션 인덱스. 섹션 인덱스는 0 ~ 5 의 숫자를 입력할 수 있는데, ComiEcatSdk\_Def.h 파일에 서 아래와 같이 정의한 값을 사용할 수 있습니다.

```
enum EMPMonSectID{
    MPMSECT_ACC, /// < 0: 가속구간만을 누적하여 계산하는 구간
    MPMSECT_DEC, /// < 1: 감속구간만을 누적하여 계산하는 구간
    MPMSECT_CONST, /// < 2: 정속구간만을 누적하여 계산하는 구간
    MPMSECT_MOVING, /// < 3: 이송중인구간을 종합하여 계산하는 구간
    MPMSECT_IDLE, /// < 4: 정지해 있는 구간만 누적하여 계산하는 구간
    MPMSECT_TOTAL, /// < 5: MOVING 과 IDLE 을 합친 모든 구간
    MPMSECT_INVALID
};
```

- ▶ TorqThresholdVal : Torque Threshold 값을 설정합니다. Torque Threshold 값은 절대값으로 적용됩니다
- ▶ 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.

## REFERENCE

□ Threshold Value 는 ThreshCount 값을 카운트하는데 기준 값으로 사용되는 값입니다.  
Deviation ThreshCount 는 각 샘플링마다의 위치 편차값과 여기서 지정하는 Threshold 값을  
비교하여 각 샘플링의 위치편차 값이 Threshold 값보다 큰 경우의 횟수를 카운트하는  
것입니다.


**NAME**


ecmMPMon\_SetDevThreshVal


ecmMPMon\_GetDevThreshVal

-

**INFORMATION**
 Master/Slave Function

 VC++/BCB/.NET

 Level 1

 위험 요소 없음
**SYNOPSIS**

□ t\_success ecmMPMon\_SetDevThreshVal (t\_i32 NetID, t\_i32 Axis, t\_ui16 SectIdx, t\_ui32 DevThresholdVal, t\_i32 \*ErrCode)

□ t\_ui32 ecmMPMon\_GetDevThreshVal (t\_i32 NetID, t\_i32 Axis, t\_ui16 SectIdx, t\_i32 \*ErrCode)

**DESCRIPTION**

ecmMPMon\_SetDevThreshVal() 함수는  
ecmMPMon\_GetDevThreshVal() 함수는

**PARAMETER**

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ SectIdx :
- ▶ DevThresholdVal :
- ▶ 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.

**EXAMPLE**

\*

**DESCRIPTION**

특정 축의 특정 섹션에 대해서 Torque Threshold 값을 설정합니다

PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ SectIdx : 섹션 인덱스. 섹션 인덱스는 0 ~ 5 의 숫자를 입력할 수 있는데, ComiEcatSdk\_Def.h 파일에 서 아래와 같이 정의한 값을 사용할 수 있습니다.

```
enum EMPMonSectID{
    MPMSECT_ACC, /// < 0: 가속구간만을 누적하여 계산하는 구간
    MPMSECT_DEC, /// < 1: 감속구간만을 누적하여 계산하는 구간
    MPMSECT_CONST, /// < 2: 정속구간만을 누적하여 계산하는 구간
    MPMSECT_MOVING, /// < 3: 이송중인구간을 종합하여 계산하는 구간
    MPMSECT_IDLE, /// < 4: 정지해 있는 구간만 누적하여 계산하는 구간
    MPMSECT_TOTAL, /// < 5: MOVING 과 IDLE 을 합친 모든 구간
    MPMSECT_INVALID
};
```

- ▶ TorqThresholdVal : Torque Threshold 값을 설정합니다. Torque Threshold 값은 절대값으로 적용됩니다
- ▶ 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.

REFERENCE

□ Threshold Value 는 ThreshCount 값을 카운트하는데 기준 값으로 사용되는 값입니다. Deviation ThreshCount 는 각 샘플링마다의 위치 편차값과 여기서 지정하는 Threshold 값을 비교하여 각 샘플링의 위치편차 값이 Threshold 값보다 큰 경우의 횟수를 카운트하는 것입니다.

NAME	I N F O R M A T I O N
ecmMPMon_ReadAxisData	Master/Slave Function
ecmMPMon_WriteAxisData	VC++/BCB/.NET
-	Level 1
	☺ 위험 요소 없음

## SYNOPSIS

□ t\_success ecmMPMon\_ReadAxisData (t\_i32 NetID, t\_i32 Axis, TMPMonData \*pBuffer, t\_i32 \*ErrCode)

□ t\_success ecmMPMon\_WriteAxisData (t\_i32 NetID, t\_i32 Axis, TMPMonData \*pMPMonData, t\_i32 \*ErrCode)

## DESCRIPTION

특정 축의 MPMon 데이터를 읽어들이는 함수입니다.

## PARAMETER

- ▶ NetID : Network 번호
- ▶ Axis : 대상 축 번호
- ▶ pBuffer : MPMon 데이터를 전달 받을 버퍼
- ▶ 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.

## REFERENCE

□ 이 함수가 실행되면 마스터 장치에서는 보정테이블에 해당하는 동적 메모리가 할당됩니다.

EXAMPLE

□ TMPMonData 구조체는 6 개의 섹션데이터로 구성되어 있습니다. 6 개의 각 섹션데이터는 다음과 같은 의미를 가집니다.

Section[0]: 가속구간에 대해서만 모니터링한 데이터

Section[1]: 감속구간에 대해서만 모니터링한 데이터

Section[2]: 정속구간에 대해서만 모니터링한 데이터

Section[3]: 이송중인 상태에 대해서 모니터링한 데이터. 이송중인 상태는 가속, 정속, 감속의 세가지 섹션을 모두 모니터링하는 것이다.

Section[4]: 정지 상태에 대해서만 모니터링한 데이터

Section[5]: 이송중과 정지상태의 모든 구간에서 모니터링한 데이터

# ComiECAT Manual / Appendix

## 부록편 내용 요약

### A. ComiECAT Error Code

ComiECAT Error Code 는 고객(顧客) 여러분들께서 API 사용시 반환되는 Error Code 를 안내하고 있습니다. 이 내용은 전체적인 함수에 관련되어 있는 내용으로서 보다 정확하게 정리하였습니다. 필요한 사항들을 반드시 참조하시어, 모든 에러 상황에 대한 이해(理解)와 도움이 되시기를 바랍니다.

### B. Frequently Asked Questions (FAQ)

고객님들께서 자주 접하시게 되는 질문 사항이나 현상에 대해서, 보다 빠르고 정확하게 대응하실 수 있도록 빈번하게 질문되는 내용에 대해서 다루고 있습니다. 실제 모션 구동 상황에서 발생할 수 있는 사항을 비롯하여, 하드웨어, 소프트웨어에 대한 문의 사항들을 저희 (주)커미조아로 문의 하시기 전에 본 장을 통해서, 질문 사항에 대한 답변을 먼저 확인 하실 수 있도록 준비되어 있습니다.

### C. Index of ComiECAT Functions

커미조아의 모션 라이브러리 매뉴얼에서 보다 쉽고 빠르게 함수를 확인할 수 있는 안내 페이지를 제공하고 있습니다. PDF 화된 문서에서 보다 빠른 시간으로 함수를 찾을 수 있도록 하이퍼 링크(Hyper Link) 기능을 제공하고 있으며, 페이지와 함수명이 바로 연결되었으므로, 편리하게 원하는 함수의 설명을 확인 할 수 있습니다.

# ComiECAT Error Code

---

---

ComiECAT의 모든 함수에는 `ErrorCode` 를 반환하는 매개 변수가 있습니다. 이 매개변수를 통해 모션 동작시 나타날 수 있는 에러 상황들을 `ErrorCode` 로 표현하여 매개 변수로 반환하게 됩니다. 이번 장에서는 이 ComiECAT에서 반환되는 모든 `ErrorCode` 를 안내드리기 위해 준비하였습니다.





## I ComiECAT Error Code 안내

Error Code	Define	Meaning
0	ecERR_NONE	에러 없음
-5	ecERR_DEVICE_NOT_LOADED	지정한 디바이스가 로드 되지 않음
-6	ecERR_WDM_VER_READ_FAIL	WDM 드라이버에서 버전 정보를 읽는데 실패
-7	ecERR_FW_VER_READ_FAIL	Firmware 버전 정보를 읽는데 실패
-10	ecERR_INVALID_BOARDID	잘못된 Board ID 입력
-11	ecERR_INVALID_DEVIDX	잘못된 Device Index 입력됨
-12	ecERR_INVALID_VERSION	SDK(DLL), WDM 드라이버, Firmware 가 상호간에 버전이 호환되지 않음
-20	ecERR_INVALID_NETID	잘못된 Network ID 입력
-25	ecERR_INVALID_SLAVEID	잘못된 슬레이브 주소나 슬레이브 인덱스 입력
-30	ecERR_INVALID_CHANNEL	잘못된 축번호나 채널번호 입력
-40	ecERR_INVALID_IXMAP_IDX	잘못된 모션 보간 맵 번호가 입력
-50	ecERR_INVALID_IXMAP_AXES	IXMAP 에서 잘못된 축이 구성
-60	ecERR_INVALID_FUNC_ARG	함수의 매개변수가 유효하지 않는 값이 입력
-65	ecERR_INVALID_HANDLE	Invalid Handle Value 가 매개변수로 전달
-66	ecERR_INVALID_RESULT_DATA	유효하지 않은 결과 데이터
-67	ecERR_INVALID_SIZE_INFO	유효하지 않은 크기 관련 데이터 정보
-69	ecERR_NULL_DLLNETCTXT	SDK(DLL)에서 제공하는 공유메모리 포인터가 NULL 인 경우
-70	ecERR_NULL_WDMNETCTXT	WDM 드라이버에서 제공하는 공유메모리 포인터가 NULL 인 경우
-71	ecERR_NULL_INPDOLEM	WDM 드라이버에서 제공하는 InPDO 용 논리메모리 포인터가 NULL 인 경우
-72	ecERR_NULL_OUTPDOLEM	WDM 드라이버에서 제공하는 OutPDO 용 논리메모리 포인터가 NULL 인 경우
-73	ecERR_INVALID_AXIS_INPDO_TYPE	현재 설정되어 있는 Axis InputPDO 형식이 원하는 데이터를 지원하지 않음.
-80	ecERR_INVALID_SECTION_ID	PDO Section ID가 올바르지 않거나, 환경설정된 섹션 리스트에 없는 경우
-90	ecERR_INVALID_LOG_MEM_ADDR	Logical Memory의 주소값이 올바르지 않은 경우
-100	ecERR_INVALID_IO_CHAN_MAP_DATA	I/O 채널 맵핑 데이터의 일부 멤버값이 올바르지 않은 경우
-110	ecERR_INVALID_FILE_PATH	지정한 파일의 경로가 잘못되었거나 파일이 없는 경우
-111	ecERR_INTERNAL_PROCESS_FAULT	내부 연산 오류
-112	ecERR_MUTEX_CREATE_FAIL	Mutex 생성 실패
-120	ecERR_FILE_OPEN_FAIL	지정한 파일이 열리지 않은 경우
-125	ecERR_FILE_NOT_FOUND	지정한 파일이 검색되지 않은 경우
-130	ecERR_FILE_READ_ERROR	파일을 읽는 과정에서 에러가 발생한 경우

APPENDIX A :: COMIECAT ERROR CODE

-140	ecERR_FILE_VERIF_DATA_ERROR	CFG파일 Verification을 수행했을 때 상호간의 데이터가 일치하지 않는 경우
-150	ecERR_MEM_ALLOC_FAIL	Memory Allocation fail
-160	ecERR_XML_POS_SAVE_ERR	XML 데이터 읽기할 때 현재의 TREE position을 저장하는데 오류가 발생한 경우
-165	ecERR_XML_ELEM_NOT_FOUND	XML 데이터 읽기할 때 지정한 Element가 검색되지 않은 경우
-166	ecERR_INVALID_ODLIST_LEN_INFO	슬레이브로부터 전달된 SDO Info. OD List의 크기 정보가 유효하지 않은 경우
-180	ecERR_INVALID_FIRMWARE_SIGN	마스터장치에 다운로드 되어 있는 펌웨어의 signature가 유효하지 않은 경우
-181	ecERR_FW_VERIF_DATA_MISMATCH	F/W Download를 한 후에 Verify 과정에서 데이터가 일치하지 않은 경우
-1000	ecERR_MOTION_BASE	Motion 제어 에리코드 시작 번호
-1010	ecERR_MOT_SERVO_ALARM	Servo driver에 알람이 발생한 경우
-1020	ecERR_MOT_SEQ_SKIPPED	Servo driver에 알람이 발생한 경우

# Frequently Asked Questions

주로 자주 발생하는 질문이나, 고객(顧客) 기술 지원에 대한 내용은 고객(顧客)님의 제품 개발 및 모션 라이브러리 운용시에 많은 도움이 될 수 있습니다. 다양한 개발 환경에서 발생할 수 있는 여러가지 문제들을 세심하고 주의깊게 다루어, 고객(顧客)님들께 진정으로 도움이 될 수 있는 내용을 준비하였습니다.

**본** 부록에서는 ㈜커미조아의 ComiECAT 사용에 있어, 문의 사항이나 궁금하신 점을 FAQ로 안내하였습니다. 각 FAQ는 ㈜커미조아의 웹 사이트(<http://www.comizoa.com>)이나 별도로 등록된 고객(顧客)님의 정보를 통해 전달될 수 있도록 노력하겠습니다. 본 FAQ 장을 고객(顧客) 문의를 하시기 전에 미리 확인(確認)하시어, 좋은 참고가 되시기를 간절히 바라겠습니다.



## II Frequently Asked Questions (FAQ)

### III Visual Studio 2005

**Q** Microsoft ® Visual Studio 2005 에서 MFC WIZARD 선택 단계에서 [Use Unicode Libraries] 항목이 확인(確認)된 상태에서 프로젝트를 시작한 다음, 빌드 하였을 경우 아래와 같은 에러가 발생하게 됩니다. 에러 발생의 이유는 ComiECAT 은 표준 ANSI Libraries API 함수를 사용하는데, 현재 프로젝트의 설정이 Unicode 로 되어 있어 다음과 같은 에러가 발생할 수 있습니다.

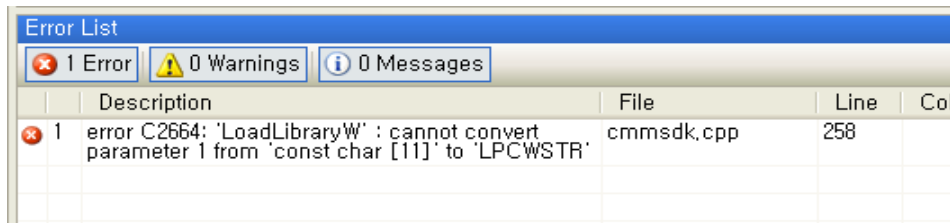


그림 21-1 Unicode 로 되어 있을 경우 발생하는 에러 화면

**A** 해당 문제를 해결 하는 방법은 다음과 같습니다. MS VC++ 의 Solution Explorer 에서 현재 작업중인 프로젝트를 선택합니다.

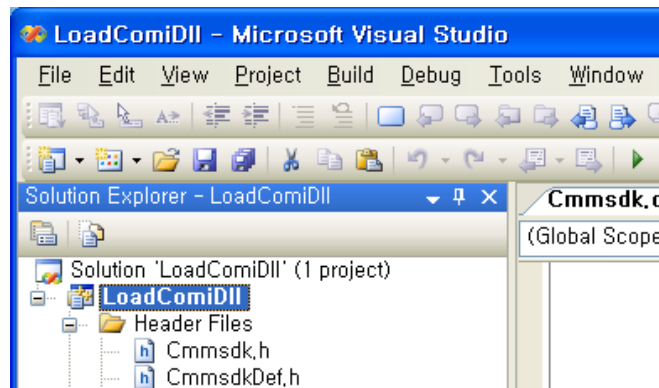


그림 21-2 사용자 생성 프로젝트 선택 화면

메뉴에서 [Project]>[Properties]를 선택하여 [Property Page]창을 엽니다.

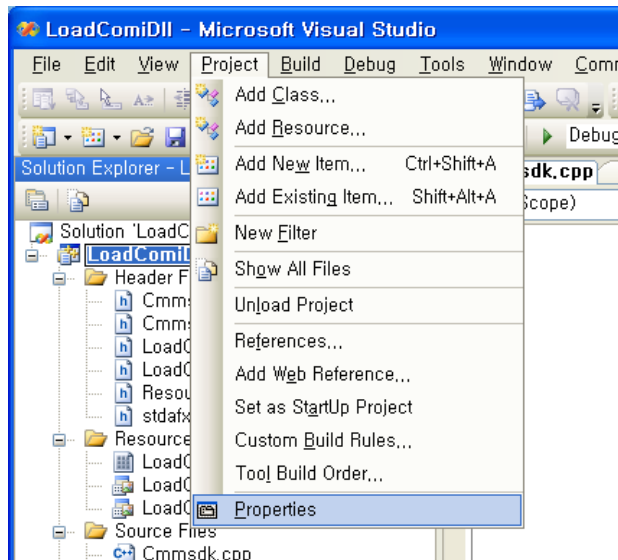


그림 21-3 사용자 생성 프로젝트의 등록정보 선택 화면

[Configuration Properties]아래의 [General]항목을 선택하면 오른쪽에 [Project Defaults]라는 항목이 표시 됩니다. [Project Defaults]항목의 [Character Set]이라는 항목을 [Use Unicode Character set]에서 [Not Set]으로 변경 하여 줍니다.

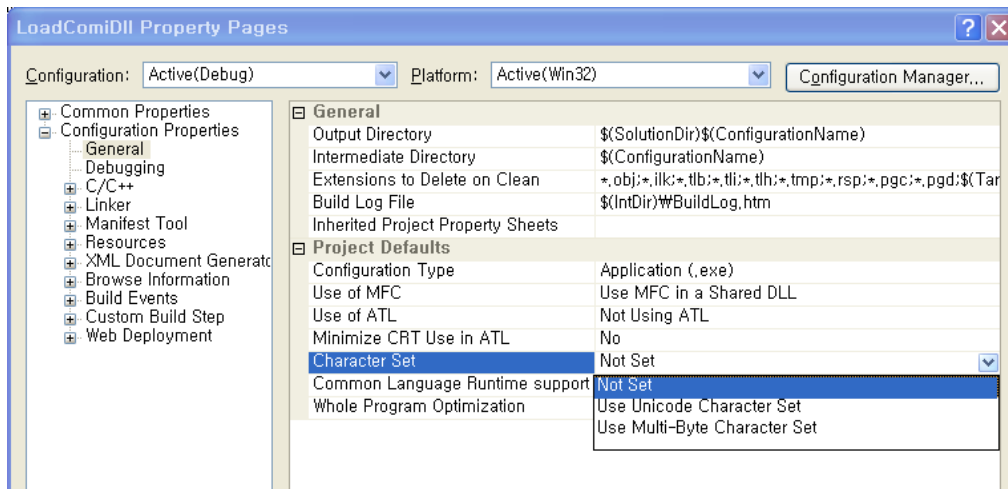


그림 21-4 문자 집합을 단일 바이트 타입으로 변경하는 화면

참고: Release Mode로 컴파일 할 경우에는 [Property Pages]창의 [Configuration]항목을 [Release]로 변경한 후 [Character Set]의 옵션을 [Not Set]으로 반드시 변경해야 합니다.

## IV Visual Basic

**Q** Visual Basic 6.0 에서 사용하던 COMIZOA SDK Library 를 Visual Studio .NET 에서 사용하면 오류가 발생한다.

**A** Microsoft 사가 인정한 Visual Studio 오류로서 디자인 타임 라이선스가 존재하지 않을 경우에 발생합니다. Visual Basic 6.0 으로 작성한 프로젝트를 Visual Studio.NET 으로 업그레이드하여 사용하고자 할 때 발생할 수 있는 오류 사항입니다. COMIZOA SDK Library 에는 라이선스가 없으므로 문제가 발생한다면 프로젝트에 포함되어있는 다른 ActiveX Control 로 인하여 문제가 발생할 수 있는 여지가 있습니다.

<http://support.microsoft.com/default.aspx?scid=kb;ko;318597>

마이크로 소프트웨어에서 제공하는 위의 주소를 통해 자세한 내용 및 해결방안을 찾아보실 수 있습니다.

## V Borland C++ Builder

**Q** C++ Builder 5 버전과 6 버전에서 COMIZOA OLE Components (ComiSliderCtrl.ocx)가 Import 되어 있는데 보이지 않는다.

**A** Visual Basic 으로 작성한 OCX 를 C++ Builder 에서 사용할 때 발생하는 문제입니다. ..\SDK\COMIZOA Components\Borland Package\C++ Builder\ 안에 각 버전별로 Component 관련 파일이 들어 있습니다. 이 파일들을 각각 위치에 맞게 복사해 주시면 됩니다.

새로운 환경에서 ComiSliderCtrl.ocx 사용하여 프로그램을 작성하시려면 다음과 같은 방법으로 등록하여 주시면 됩니다.

\* Builder 5 에서 ComiSliderCtrl.ocx 사용방법.

1. 윈도우 시스템폴더에 OCX 를 복사한 후 regsvr32 명령으로 OCX 를 등록시킵니다.  
(시작->실행 메뉴에서 : regsvr32 c:\windows\system32\ComiSliderCtrl.ocx)
2. ComiSliderCtrl.ocx 를 [..\CBuilder5\Bin\] 폴더로 복사합니다.  
tlibimp.exe 프로그램을 이용하여 Import Type Library File 을 생성합니다.

```
C:\Program Files\Borland\CBuilder5\BIN> tlibimp.exe -Yu -Ya ComiSliderCtrl.ocx
ComiSliderCtrl_TLB.h
ComiSliderCtrl_TLB.cpp
ComiSliderCtrl_OCX.h
ComiSliderCtrl_OCX.cpp
ComiSliderCtrl_OCX.dcr
총 5 가지 파일이 생성됩니다.
```

3. 계속하여, 시작-> 실행 메뉴를 통해 'cmd' 를 입력하여, 명령프롬프트를 실행합니다.

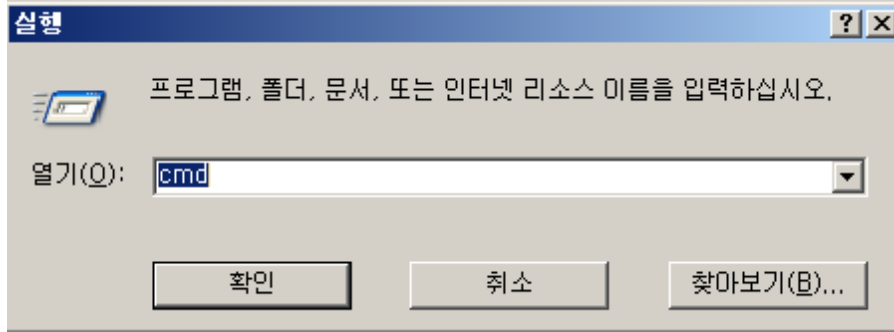


그림 21-5 [작업표시줄][시작][실행] 창에서 cmd 명령어 입력

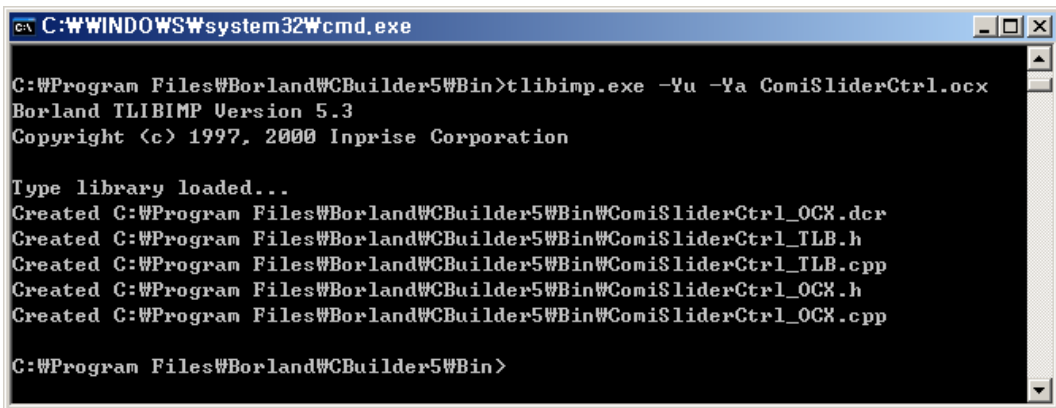


그림 21-6 Tlibimp.exe 파일을 이용 Import Type Library File 생성

4. Builder 5 에서 OCX 를 등록합니다.

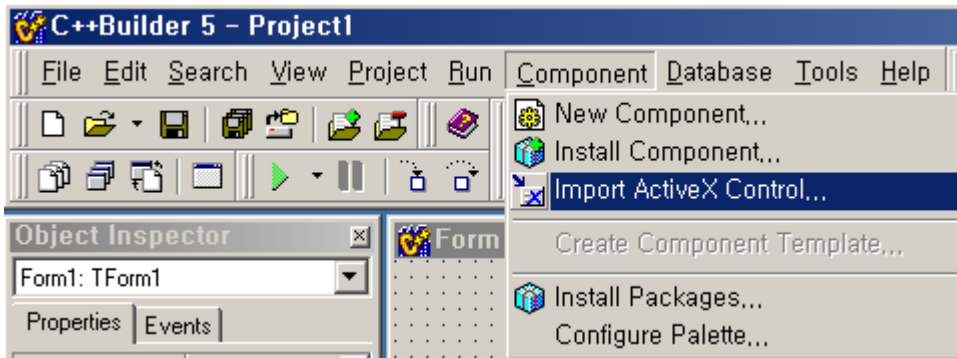


그림 21-7 C++ Builder 5 의 ActiveX Component 등록

5. Import ActiveX 윈도우가 화면에 나타나면, ComiSliderCtrl 을 선택하고 원하는 Palette Page 를 선택한뒤 Install 버튼을 눌러 설치를 시작합니다.

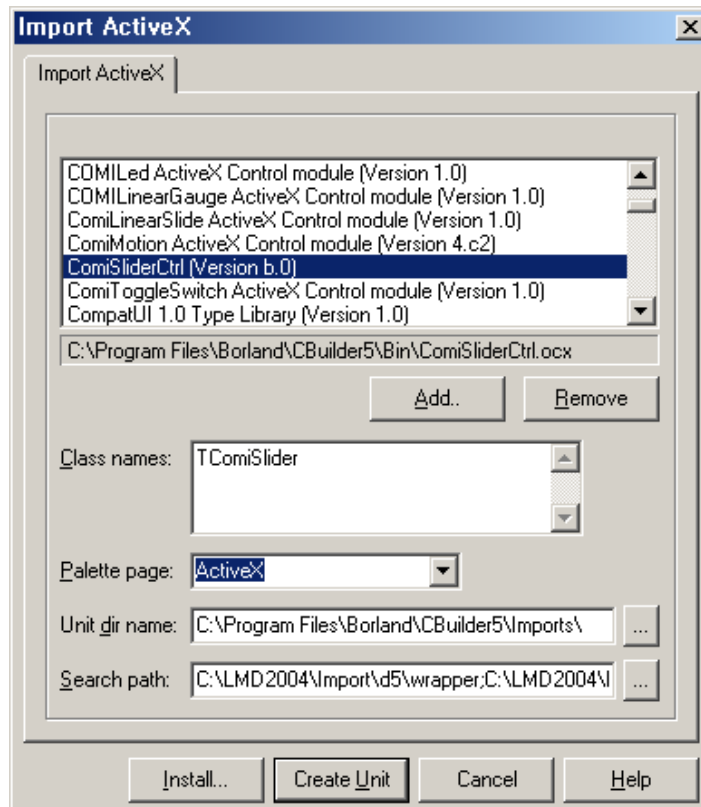


그림 21-8 C++ Builder 5의 ActiveX Component 를 등록하기 위한 Import Active X 화면

6. Install 창이 화면에 나타나면 Into new package 탭에서 생성을 원하는 Package 파일 이름 및 설명을 입력하여 새로운 bpk 를 생성합니다.

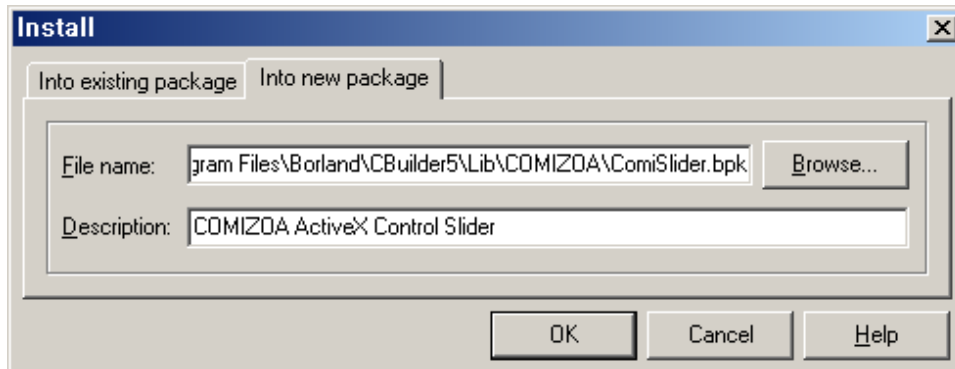


그림 21-9 C++ Builder 5의 ActiveX Component 설치 화면

7. Package 를 Install 한다는 확인(確認) 창이 화면에 표시되면, “No” 를 선택합니다.

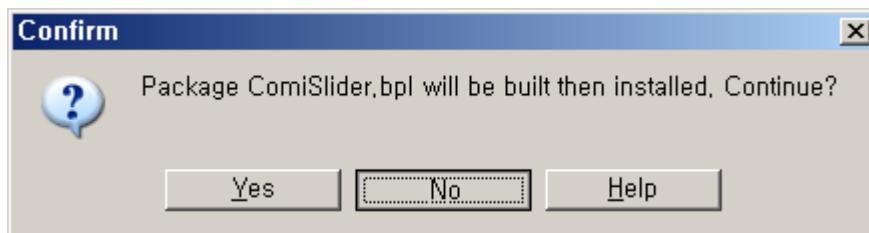


그림 21-10 C++ Builder 5에서 ActiveX Component 설치 화면



8. Tlibimp.exe 파일을 이용해 생성한 5 개의 파일을 ..\Borland\CBuilder5\Imports 에 덮어쓰기(Overwrite) 합니다

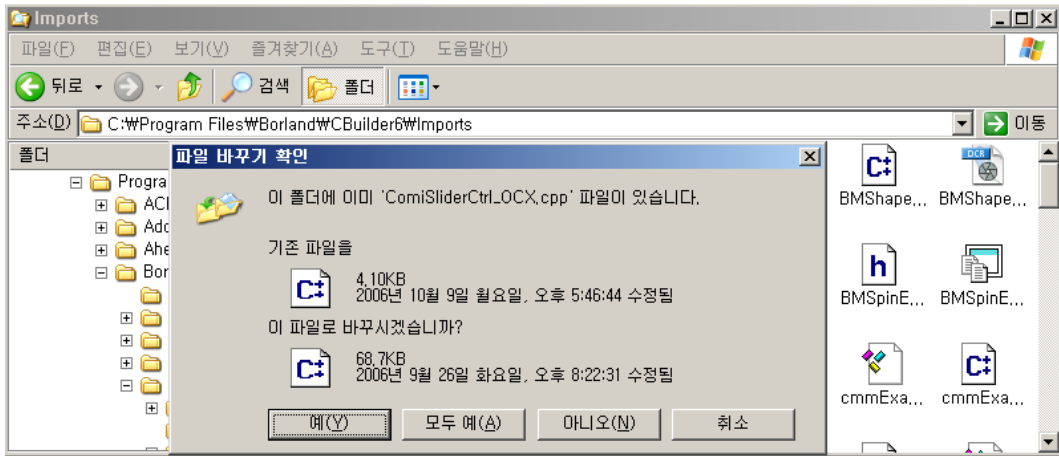


그림 21-11 Tlibimp.exe 가 생성한 파일을 통해 기존 파일을 대체하는 작업 화면

9. 덮어쓰기가 끝나면 생성한 Package 파일을 Install 합니다.

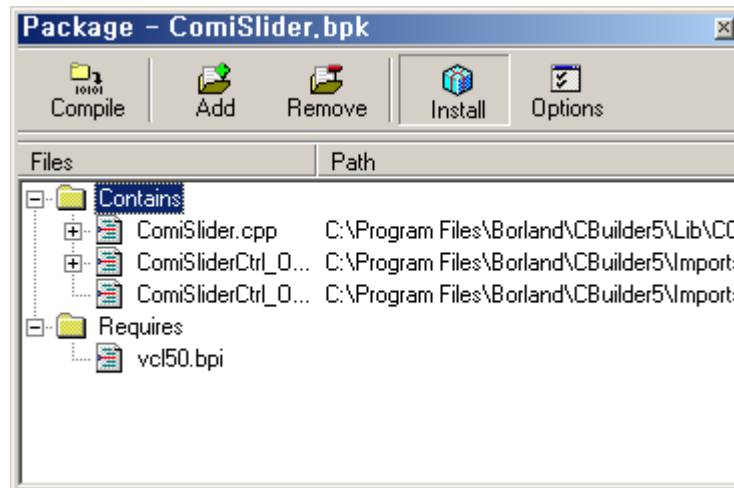


그림 21-12 C++ Builder 5 의 Package 표시 화면

10. Package 가 Install 되었다는 메시지와 함께 툴 팔레트에 OCX 가 정상적으로 등록된 것을 확인(確認)할 수 있습니다.

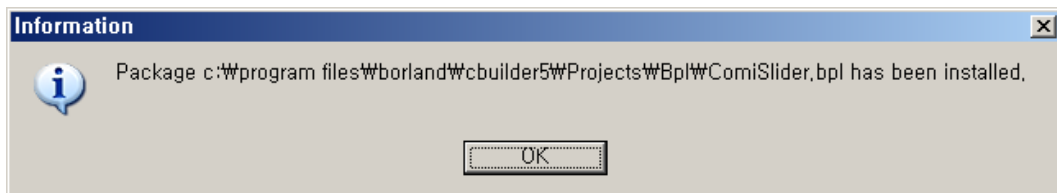


그림 21-13 C++ Builder 5 의 패키지 설치 완료 화면

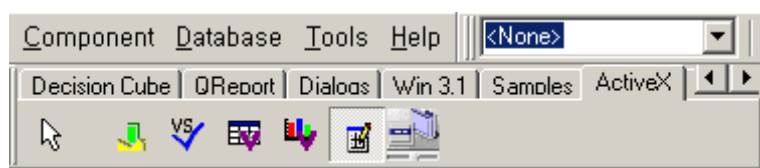


그림 21-14 C++ Builder 5 에서 팔레트(Palette) 에 등록된 ComiSlider Active X Control

\* Builder 6 에서 ComiSliderCtrl.ocx 사용방법.

11. [Tool][Environment Options..]를 선택합니다..

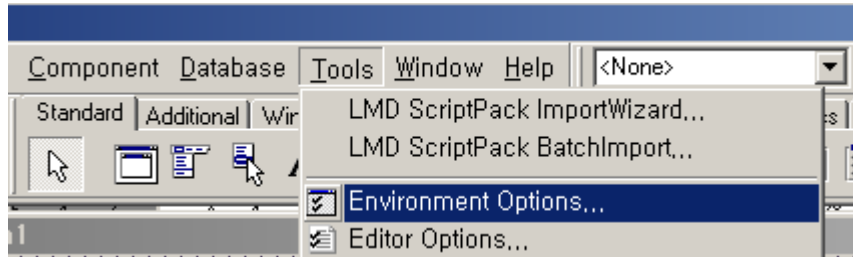


그림 21-15 C++ Builder 6 에서 ActiveX Component 등록 1

12. Environment Options 창의 Type Library 탭에서 “Ignore special CoClass Flags when importing”, “Can Create” - Check 한뒤 OK 버튼을 클릭합니다.

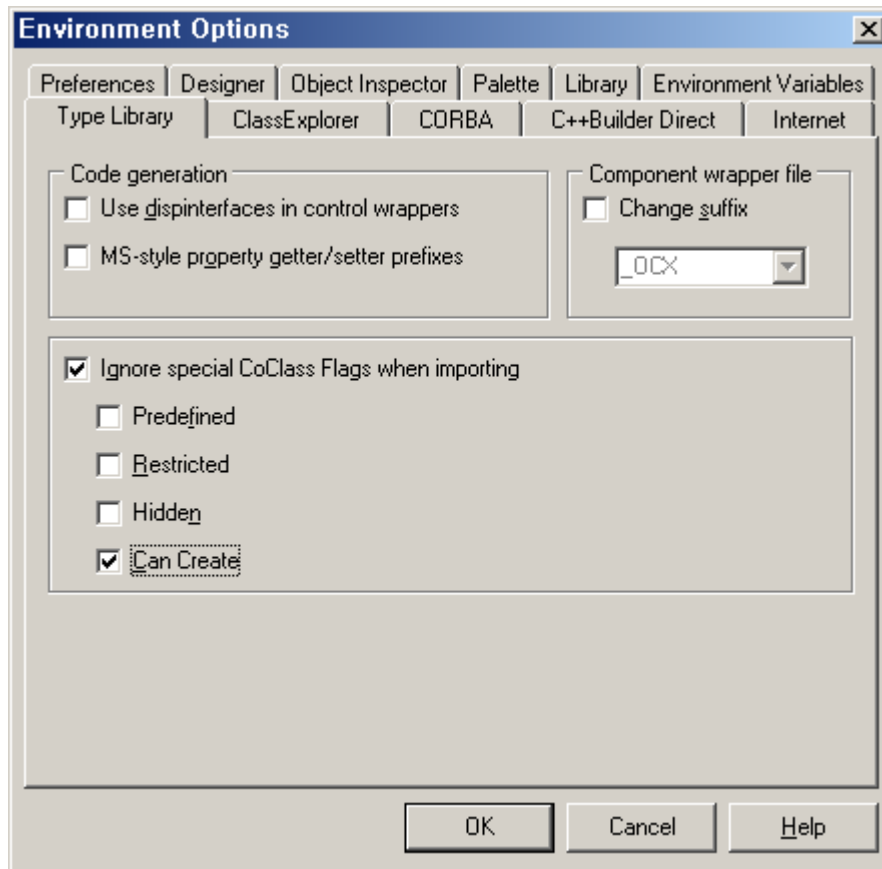


그림 21-16 C++ Builder 6 의 Environment Options 창의 화면

13. [Component][Import ActiveX Control..]를 선택한뒤 절차에 따라 OCX 를 등록하면 됩니다.

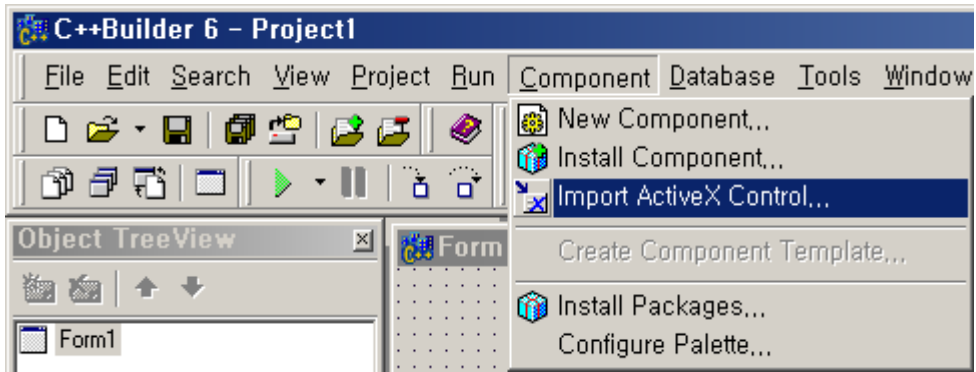


그림 21-17 C++ Builder 6에서 ActiveX Component 등록 3

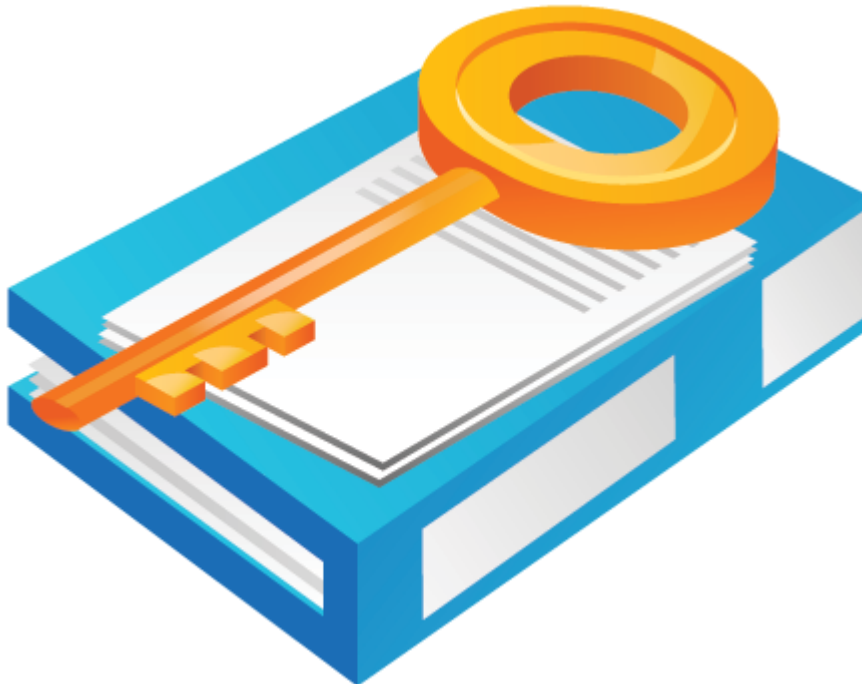
# Index of ComiECAT functions

---

---

필요한 함수(函數)를 가장 빠르고 쉽게 찾으십시오. ComiECAT 매뉴얼에서는 고객(顧客)님들께서 원하시는 함수들을 일목 요연하게 정리하였습니다. 필요한 함수는 온라인 문서에서 하이퍼 링크 기능으로 찾을 수 있도록 구성하였습니다.

**하** 이퍼 링크 기능(機能)을 통해 본장에서는 빠르고 정확하게 고객(顧客)님들께서 원하시는 함수(函數)를 찾으실 수 있도록 구성(構成)하였습니다. Adobe 社의 Acrobat Reader 와 같은 전자 문서 뷰어(Viewer) 를 통해 최단 시간내에 원하시는 함수(函數)를 찾을 수 있습니다.



## VI Index of ComiECAT Functions

## VII Quick Reference to ComiECAT Functions

<i>ecDll_Load</i>	54
<i>ecDll_Unload</i>	56
<i>ecDll_IsLoaded</i>	57
<i>ecGn_LoadDevices</i>	60
<i>ecGn_UnloadDevices</i>	62
<i>ecGn_IsDevLoaded</i>	63
<i>ecGn_GetNumDevices</i>	64
<i>ecGn_GetNumNetworks</i>	65
<i>ecGn_GetDevInfo</i>	66
<i>ecDlog_SetFilePath / ecDlog_GetFilePath</i>	81
<i>ecDlog_SetLogType / ecDlog_GetLogType</i>	83
<i>ecDlog_SetLogLevel/ ecDlog_GetLogLevel</i>	84
<i>ecNet_SetFastFuncType / ecNet_GetFastFuncType</i>	89
<i>ecNet_ScanSlave</i>	95
<i>ecNet_SetCfgSlaveCount / ecNet_GetCfgSlaveCount</i>	96
<i>ecNet_InPDO_GetData</i>	103
<i>ecNet_OutPDO_GetData</i>	105
<i>ecNet_InPDO_GetBufPtr</i>	107
<i>ecNet_OutPDO_GetBufPtr</i>	108
<i>ecSlv_GetProdInfo / ecSlv_GetProdInfo_A</i>	113
<i>ecSlv_GetTypeInfo / ecSlv_GetTypeInfo_A</i>	115
<i>ecSlv_GetAIState / ecSlv_GetAIState_A</i>	117
<i>ecdiGetOne</i>	148
<i>ecdiGetMulti</i>	150
<i>ecdiGetNumChannels</i>	145
<i>ecdoPutOne / ecdoGetOne</i>	163
<i>ecdoPutMulti / ecdoGetMulti</i>	166
<i>ecdoGetNumChannels</i>	오류! 책갈피가 정의되어 있지 않습니다.

<i>ecaiGetChanVal_I</i>	194
<i>ecaiGetChanVal_F</i>	199
<i>ecaiGetChanVal_FS</i>	200
<i>ecaiGetNumChannels</i>	오류! 책갈피가 정의되어 있지 않습니다.
<i>ecaoSetChanVal_I</i>	202
<i>ecaoSetChanVal_F</i>	203
<i>ecaoSetChanVal_FS</i>	204
<i>ecaoGetOutValue_I</i>	206
<i>ecaoGetOutValue_F</i>	207
<i>ecaoGetOutValue_FS</i>	210
<i>ecaoGetNumChannels</i>	오류! 책갈피가 정의되어 있지 않습니다.
<i>ecmGn_GetAxisList</i>	211
<i>ecmGn_AxisToSlaveInde</i>	214
<i>ecmGn_AxisToSlaveID</i>	215
<i>ecmGn_SlaveIndexToAxis</i>	216
<i>ecmGn_SlaveIDToAxis</i>	217
<i>ecmSxCfg_SetMioProp / ecmSxCfg_GetMioProp</i>	231
<i>ecmSxCfg_SetSpeedPatt / ecmSxCfg_SetSpeedPatt_FF / ecmSxCfg_GetSpeedPatt</i>	242
<i>ecmSxCfg_SetJerkRatio / ecmSxCfg_SetJerkRatio_FF / ecmSxCfg_GetJerkRatio</i>	247
<i>ecmSxCfg_SetUnitDist / ecmSxCfg_GetUnitDist</i>	233
<i>ecmSxCfg_SetUnitSpeed / ecmSxCfg_GetUnitSpeed</i>	235
<i>ecmSxCfg_PosCorr_SetTableData / ecmSxCfg_PosCorr_GetTableData</i>	291
<i>ecmSxCfg_Ring_SetEnable/ ecmSxCfg_Ring_GetEnable</i>	313
<i>ecmSxCfg_Ring_SetPosRange/ ecmSxCfg_Ring_GetPosRange</i>	314
<i>ecmSxCfg_Ring_SetDirMode / ecmSxCfg_Ring_GetDirMode</i>	316
<i>ecmSxCfg_AutoTorq_SetValMode/ ecmSxCfg_AutoTorq_GetValMode</i>	340
<i>ecmSxCfg_AutoTorq_SetMultiVal/ ecmSxCfg_AutoTorq_GetMultiVal</i>	342
<i>ecmSxCfg_AutoTorq_SetValue/ ecmSxCfg_AutoTorq_GetValue</i>	345
<i>ecmSxCtl_SetAlmRst / ecmSxCtl_SetAlmRst_FF</i>	오류! 책갈피가 정의되어 있지 않습니다.
<i>ecmSxCtl_ResetAlm / ecmSxCtl_ResetAlm_FF</i>	오류! 책갈피가 정의되어 있지 않습니다.
<i>ecmSxMot_VMovestart / ecmSxMot_VMovestart_FF</i>	255
<i>ecmSxMot_MoveStart / ecmSxMot_MoveStart_FF / ecmSxMot_Move / EcmSxMot_Move_NB</i>	258
<i>ecmSxMot_MoveToStart / ecmSxMot_MoveToStart_FF / ecmSxMot_MoveTo / EcmSxMot_MoveTo_NB</i>	261

<i>ecmSxMot_OverrideSpeed</i>	263
<i>ecmSxMot_OverrideMove</i>	360
<i>ecmSxMot_OverrideMoveTo</i>	362
<i>ecmSxMot_Stop / ecmSxMot_Stop_FF</i>	265
<i>ecmSxSt_IsBusy</i>	267
<i>ecmSxSt_WaitCompt / ecmSxSt_WaitCompt_NB</i>	269
<i>ecmSxSt_SetCount</i>	270
<i>ecmSxSt_GetCount</i>	271
<i>ecmSxSt_SetPosition</i>	272
<i>ecmSxSt_GetPosition</i>	273
<i>ecmSxSt_GetCurSpeed</i>	274
<i>ecmSxSt_GetCurTorque</i>	275
<i>ecmSxSt_GetMotState</i>	276
<i>ecmSxSt_GetFlags</i>	278
<i>ecmMxMot_VMoveStart / ecmMxMot_VMoveStart_FF</i>	388
<i>ecmMxMot_MoveStart / ecmMxMot_MoveStart_FF</i>	391
<i>ecmMxMot_MoveToStart / ecmMxMot_MoveToStart_FF</i>	394
<i>ecmMxMot_Stop / ecmMxMot_Stop_FF</i>	396
<i>ecmMxMot_IsBusy</i>	398
<i>ecmMxSt_WaitCompt / ecmMxSt_WaitCompt_NB</i>	400
<i>ecmIxCfg_MapAxes / ecmIxCfg_MapAxes_FF</i>	409
<i>ecmIxCfg_UnmapAxes / ecmIxCfg_UnmapAxes_FF</i>	411
<i>ecmIxCfg_SetSpeedPatt / ecmIxCfg_SetSpeedPatt_FF / ecmIxCfg_GetSpeedPatt</i>	412
<i>ecmIxCfg_SetJerkRatio / ecmIxCfg_SetJerkRatio_FF / ecmIxCfg_GetJerkRatio</i>	414
<i>ecmIxCfg_Spline_ClearPool</i>	416
<i>ecmIxCfg_Spline_AddNewObj</i>	419
<i>ecmIxCfg_Spline_SetRefPoint</i>	420
<i>ecmIxCfg_Spline_BuildObj</i>	422
<i>ecmIxCfg_MPRLin2X_ClearPool</i>	423
<i>ecmIxCfg_MPRLin2X_AddNewObj</i>	424
<i>ecmIxCfg_MPRLin2X_SetRefPoint</i>	425
<i>ecmIxCfg_MPRLin2X_BuildObj</i>	427
<i>ecmIxCfg_LineStart / ecmIxCfg_LineStart_FF / ecmIxCfg_Line / ecmIxCfg_Line_NB</i>	428

<i>ecmIxMot_LineToStart / ecmIxMot_LineToStart_FF / ecmIxMot_LineTo / ecmIxMot_LineTo_NB</i>	431
<i>ecmIxMot_ArcAng_R_Start / ecmIxMot_ArcAng_R_Start_FF / ecmIxMot_ArcAng_R / ecmIxMot_ArcAng_R_NB</i>	433
<i>ecmIxMot_ArcAng_A_Start / ecmIxMot_ArcAng_A_Start_FF / ecmIxMot_ArcAng_A / ecmIxMot_ArcAng_A_NB</i>	437
<i>ecmIxMot_ArcPos_R_Start / ecmIxMot_ArcPos_R_Start_FF / ecmIxMot_ArcPos_R / ecmIxMot_ArcPos_R_NB</i>	439
<i>ecmIxMot_ArcPos_A_Start / ecmIxMot_ArcPos_A_Start_FF / ecmIxMot_ArcPos_A / ecmIxMot_ArcPos_A_NB</i>	441
<i>ecmIxMot_Arc3P_R_Start / ecmIxMot_Arc3P_R_Start_FF / ecmIxMot_Arc3P_R / ecmIxMot_Arc3P_R_NB</i>	445
<i>ecmIxMot_Arc3P_A_Start / ecmIxMot_Arc3P_A_Start_FF / ecmIxMot_Arc3P_A / ecmIxMot_Arc3P_A_NB</i>	447
<i>ecmIxMot_Spline_Start</i>	449
<i>ecmIxMot_MPRLin2X_Start</i>	453
<i>ecmIxMot_Stop / ecmIxMot_Stop_FF</i>	457
<i>ecmIxSt_IsBusy</i>	459
<i>ecmIxSt_WaitCompt / ecmIxSt_WaitCompt_NB</i>	461
<i>ecmHomeCfg_SetMode / ecmHomeCfg_GetMode</i>	477
<i>ecmHomeCfg_SetOffset / ecmHomeCfg_GetOffset</i>	480
<i>ecmHomeCfg_SetEscDist / ecmHomeCfg_GetEscDist</i>	481
<i>ecmHomeCfg_SetSpeedPatt / ecmHomeCfg_GetSpeedPatt</i>	482
<i>ecmHomeCfg_SetOption / ecmHomeCfg_GetOption</i>	484
<i>ecmHomeMot_MoveStart / ecmHomeMot_MoveStart_FF / ecmHomeMot_Move</i>	486
<i>ecmHomeSt_IsBusy</i>	489
<i>ecmHomeSt_WaitCompt</i>	491
<i>ecmHomeSt_GetFlags</i>	492
<i>ecmMsCfg_SetSlvEnv / ecmMsCfg_GetS;vEnv</i>	496
<i>ecmMsCtl_StartSlv</i>	498
<i>ecmMsCtl_StopSlv</i>	500
<i>ecmMsSt_IsSlvStarted</i>	501
<i>ecmLmCtl_Begin</i>	506
<i>ecmLmCtl_End</i>	510
<i>ecmLmCtl_Run</i>	511
<i>ecmLmCtl_Stop</i>	512



<i>ecmLmCtl_ClearQue</i>	513
<i>ecmPtmCtl_Begin</i>	529
<i>ecmPtmCtl_End</i>	534
<i>ecmPtmCtl_SetHold</i>	535
<i>ecmPtmCfg_AddItem_PT</i>	536
<i>ecmPtmSt_GetRunSts</i>	538
<i>ecmPtmSt_GetRunStepInfo</i>	540

---

---

TEST & MEASUREMENT & AUTOMATION / COMIZOA

# ComiECAT Manual

---

저작권자 : ㈜커미조아

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

2013 년 07 월 22 일 1 판 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.

