

AXL library family
AXL

Product Information

Full information about other AJINEXTEK products is available by visiting our Web Site at:

Home Page : www.ajinextek.com

E-Mail : support@ajinextek.com

Useful Contact Information

Customer Support Seoul

Tel : 82-031-436-2180~2

Fax: 82-031-436-2183

Customer Support Cheonan

Tel : 82-041-555-9771~2 Fax: 82-041-555-9773

Customer Support Deagu

Tel : 82-053-593-3700~2 Fax: 82-053-593-3703



AJINEXTEK's sales team is always available to assist you in making your decision the final choice of boards or systems is solely and wholly the responsibility of the buyer. AJINEXTEK's entire liability in respect of the board or systems is as set out in AJINEXTEK's standard terms and conditions of sale

Content

| | |
|---|-----------|
| CONTENT..... | 3 |
| AXL 개요..... | 6 |
| 라이브러리 초기화..... | 10 |
| 라이브러리 경로 설정..... | 10 |
| DLL 파일 복사하기..... | 18 |
| 축번호, 접점번호, 채널번호 할당 방법..... | 19 |
| 기본 원리..... | 19 |
| 축번호, 접점번호, 채널번호의 할당 예(PCI slot 장착 순서)..... | 20 |
| BASE COMMAND 매뉴얼 정보..... | 21 |
| 헤더 파일..... | 21 |
| 함수 용어..... | 21 |
| 매뉴얼의 함수 이름..... | 21 |
| 본 매뉴얼의 인자 이름..... | 21 |
| BASE COMMAND QUICK LIST..... | 22 |
| 라이브러리 초기화..... | 22 |
| 라이브러리 및 베이스 보드 정보..... | 22 |
| MECHATROLINK III COMMAND QUICK LIST..... | 23 |
| 공통 커맨드..... | 23 |
| 표준 I/O 기기 커넥션..... | 23 |
| 표준 I/O 프로파일 커맨드..... | 23 |
| Define 문..... | 25 |
| BASE COMMAND FUNCTION LIST..... | 26 |
| 라이브러리 초기화..... | 27 |

| | |
|-------------------------------|-----------|
| AxlOpen | 28 |
| AxlOpenNoReset | 30 |
| AxlClose | 32 |
| AxlIsOpened | 33 |
| AxlInterruptEnable..... | 35 |
| AxlInterruptDisable | 37 |
| 라이브러리 및 베이스 보드 정보..... | 38 |
| AxlGetBoardCount | 39 |
| AxlGetLibVersion | 41 |
| AxlGetModuleNodeStatus | 43 |
| AxlGetBoardStatus | 45 |
| AxlGetLockMode | 47 |
| AxlSetLogLevel | 49 |
| AxlGetLogLevel | 51 |

MECHATROLINK III COMMAND FUNCTION LIST 53

| | |
|--------------------------------------|-----------|
| 공통 커맨드..... | 54 |
| AxlM3SetStationNop..... | 55 |
| AxlM3GetStationParameter | 57 |
| AxlM3SetStationParameter..... | 59 |
| AxlM3GetStationIdRd | 61 |
| AxlM3SetStationConfig..... | 63 |
| AxlM3GetStationAlarm..... | 65 |
| AxlM3SetStationAlarmClear | 67 |
| AxlM3SetStationSyncSet..... | 69 |
| AxlM3SetStationConnect | 71 |
| AxlM3SetStationDisConnect | 73 |
| AxlM3GetStationStoredParameter | 75 |
| AxlM3SetStationStoredParameter | 77 |
| AxlM3GetStationMemory | 79 |
| AxlM3SetStationMemory | 81 |

표준 I/O 기기 커백션 오류! 책갈피가 정의되어 있지 않습니다.

| | |
|----------------------------------|----|
| AxlM3SetStationAccessMode | 84 |
| AxlM3GetStationAccessMode | 86 |
| AxlM3SetAutoSyncConnectMode..... | 88 |
| AxlM3GetAutoSyncConnectMode..... | 90 |
| AxlM3SyncConnectSingle..... | 92 |
| AxlM3SyncDisconnectSingle | 94 |
| AxlM3IsOnLine | 96 |

표준 I/O 프로파일 커맨드 오류! 책갈피가 정의되어 있지 않습니다.

| | |
|--------------------------|-----|
| AxlM3GetStationRWS..... | 99 |
| AxlM3SetStationRWS | 101 |
| AxlM3GetStationRWA | 103 |
| AxlM3SetStationRWA | 105 |

에러코드 테이블 (Error Code Table) 확인..... **107**

AXL 개요

아진엑스텍의 제어보드는 기본형 베이스 보드에 디지털 I/O모듈, 아날로그 I/O모듈, 통신모듈 등을 어떻게 조합하는가에 따라 다양한 기능을 포함하는 형태로 구성할 수 있다.

또한 모듈형태가 아닌 특화된 기능의 일체형 단일 보드들의 조합으로도 저렴하고 강력한 기능을 구현하는 형태로 구성할 수 있다. 라이브러리의 사용 통일성을 위해 일체형 단일 보드도 모듈 형태처럼 베이스 보드에 모듈이 장착되어 동작되는 형식을 가진, 보드에 붙박이 된 Module이라고 사용자는 이해하면 되겠다. 단일형 보드에는 네트워크 리소스를 제어하는 마스트 보드 제품군도 포함되며, 네트워크로 연결된 제어요소는 Slave Node로 기술한다.

이러한 아진엑스텍의 보드와 모듈을 제어하는 프로그램 라이브러리 API (Application Programming Interface)를 통틀어 AXL(Ajinextek Library)이라고 한다.

AXL을 사용할 수 있는 PC 시스템 실장형 제품군은 다음과 같다.

| 제품명 | 특징 | 비고 |
|---------------------|--|--|
| BPFR(Full size) | 모듈 장착형 베이스 보드(PCI bus) | 최대 4개의 모듈 장착 |
| BPHR(Half size) | 모듈 장착형 베이스 보드(PCI bus) | 최대 2개의 모듈 장착 |
| BIFR(Full size) | 모듈 장착형 베이스 보드(ISA bus) | 최대 4개의 모듈 장착 |
| BIHR(Half size) | 모듈 장착형 베이스 보드(ISA bus) | 최대 2개의 모듈 장착 |
| BPHD(Half size) | 모듈 장착형 베이스 보드, DB32P가 기본 포함됨(PCI bus) | 최대 2개 모듈 장착, 디지털 입력(16)/출력(16) 기본 포함 |
| PCI-DI64x | 일체형 디지털 입력 보드(PCI bus) | 디지털 입력 64 |
| PCI-DB64x | 일체형 디지털 입력 보드(PCI bus) | 디지털 입력 32/ 디지털 출력 32 |
| PCI-DO64x | 일체형 디지털 입력 보드(PCI bus) | 디지털 출력 64 |
| PCI-AIO1602HR | 일체형 아날로그 입출력 보드(PCI bus) | 아날로그 출력(2CH), 입력(16CH) |
| PCI-N804(Half size) | 일체형 모션 제어 보드(PCI bus) | 8 축 제어, Open loop |
| PCI-N404(Half size) | 일체형 모션 제어 보드(PCI bus) | 4 축 제어, Open loop |
| PCI-N204(Half size) | 일체형 모션 제어 보드(PCI bus) | 2 축 제어, Open loop |
| PCI-R1604 | 네트워크 마스터 보드(PCI bus) | RTEX(Real-Time Express)용 네트워크 마스터 보드 최대 32개 Slave Node 장착 기본 16축, 확장시 32축 제어 |
| PCI-R1604-MLII | 네트워크 마스터 보드(PCI bus) | Mechatrolink-II 마스터 보드 최대 30개 Slave Node 장착 기본 16축, 확장시 30축 제어 |
| PCI-R0804-MLII | 네트워크 마스터 보드(PCI bus) | Mechatrolink-II 마스터 보드 최대 24개 Slave Node 장착 기본 8축, 확장시 24축 제어 |
| PCI-R1604-SIIIIH | 네트워크 마스터 보드(PCI bus) | SSCNET III/H 마스터 보드 최대 32개 Slave Node 장착 기본 16축, 확장시 32축 제어 |
| PCI-R1600-MLIII | 네트워크 마스터 보드(PCI bus) | Mechatrolink-III 마스터 보드 최대 16개 Slave Node 장착 기본 16축 |
| PCI-R3200-MLIII | 네트워크 마스터 보드(PCI bus) | Mechatrolink-III 마스터 보드 |

| | | |
|--|--|--------------------------------|
| | | 최대 32개 Slave Node 장착 기본 32축 |
|--|--|--------------------------------|

AXL을 사용할 수 있는 베이스 보드 장착형 모듈 제품군은 다음과 같다.

| 제품명 | 특징 | 비고 |
|-------------|--------------|----------------------|
| SMC-2V03 | 모션 모듈 | 2축 제어, 펄스 출력형, (단종) |
| SMC-2V04 | 모션 모듈 | 2축 제어, 펄스 출력형 |
| SIO-DI32 | 디지털 입력 모듈 | 디지털 입력 32 점점 |
| SIO-DB32 | 디지털 입출력 모듈 | 디지털 입력(16)/출력(16) 점점 |
| SIO-DO32 | 디지털 출력 모듈 | 디지털 출력 32 점점 |
| SIO-AI4RB | 아날로그 입력 모듈 | 아날로그 입력 4CH |
| SIO-AO4RB | 아날로그 출력 모듈 | 아날로그 출력 4CH |
| SIO-AO4F | 아날로그 출력 모듈 | 아날로그 출력 4CH |
| SIO-AI8F | 아날로그 입력 모듈 | 아날로그 입력 8CH |
| SIO-AI8AO4F | 아날로그 입/출력 모듈 | 아날로그 입력8CH/출력4CH |
| SIO-CN2CH | 트리거 출력 모듈 | 카운터 입력, 트리거 출력 2CH |

AXL을 사용할 수 있는 네트워크 슬레이브 노드 제품군은 다음과 같다.

| 제품명 | 특징 | 비고(제조사) |
|--------------|-------------------------------------|-----------------------------|
| RTEX-PM | RTEX 모션 Slave Node | 1축, 펄스 출력형(Ajinextek) |
| RTEX-PM2Q | RTEX 모션 Slave Node | 2축, 펄스 출력형(Ajinextek) |
| RTEX-PM2Q | RTEX 모션 Slave Node | 4축, 펄스 출력형(Ajinextek) |
| RTEX-DI32 | RTEX 디지털 입력 Slave Node | 디지털 입력 32 점점(Ajinextek) |
| RTEX-DO32 | RTEX 디지털 출력 Slave Node | 디지털 출력 32 점점(Ajinextek) |
| RTEX-CNT2 | RTEX 트리거 출력 Slave Node | 카운터 입력, 트리거 출력 2ch |
| RTEX-AI8 | RTEX 아날로그 입력 Slave Node | 아날로그 입력, 8CH(Ajinextek) |
| RTEX-AI16 | RTEX 아날로그 입력 Slave Node | 아날로그 입력, 16CH(Ajinextek) |
| RTEX-AO4 | RTEX 아날로그 출력 Slave Node | 아날로그 출력, 4CH(Ajinextek) |
| RTEX-AO8 | RTEX 아날로그 출력 Slave Node | 아날로그 출력, 8CH(Ajinextek) |
| RTEX-A4N | RTEX 모션 Slave Node | 1축, 서버 드라이브(Panasonic) |
| RTEX-A5N | RTEX 모션 Slave Node | 1축, 서버 드라이브(Panasonic) |
| JEPMC-PL2910 | Mechatrolink-II 모션 Slave Node | 2축, 펄스 출력형(Yaskawa) |
| JEPMC-AN2900 | Mechatrolink-II 아날로그 입력 Slave Node | 아날로그 입력, 4CH(Yaskawa) |
| JEPMC-AN2910 | Mechatrolink-II 아날로그 출력 Slave Node | 아날로그 출력, 2CH(Yaskawa) |
| JEPMC-IO2330 | Mechatrolink-II 디지털 입/출력 Slave Node | 디지털 입출력 64/64 점점(Yaskawa) |
| SIO-DB96T | Mechatrolink-II 디지털 입/출력 Slave Node | 디지털 입출력 96/96 점점(Ajinextek) |
| CSDL-xxxx | Mechatrolink-II 모션 Slave Node | 1축, 서버 드라이브(Convex) |
| CSDH-xxxx | Mechatrolink-II 모션 Slave Node | 1축, 서버 드라이브(Convex) |
| SGDV-xxxx11A | Mechatrolink-II 모션 Slave Node | 1축, 서버 드라이브(Yaskawa) |
| MRJ4B-xxxx | SSCNET III/H 모션 Slave Node | 1축, 서버 드라이브(Mitsubishi) |

| | | |
|--------------|--------------------------------------|-----------------------------|
| SGDV-xxxx21A | Mechatrolink-III 모션 Slave Node | 1축, 서버 드라이브(Yaskawa) |
| N3MLIII-PMxQ | Mechatrolink-III 모션 Slave Node | 펄스 출력형(Ajinextek) |
| N3MLIII-DI32 | Mechatrolink-III 디지털 입력 Slave Node | 디지털 입력 32 점점(Ajinextek) |
| N3MLIII-DO32 | Mechatrolink-III 디지털 출력 Slave Node | 디지털 출력 32 점점(Ajinextek) |
| N3MLIII-DB32 | Mechatrolink-III 디지털 입/출력 Slave Node | 디지털 입출력 16/16 점점(Ajinextek) |
| N3MLIII-AI8 | Mechatrolink-III 아날로그 입력 Slave Node | 아날로그 입력, 8CH(Ajinextek) |
| N3MLIII-AO4 | Mechatrolink-III 아날로그 출력 Slave Node | 아날로그 출력, 4CH(Ajinextek) |
| N3MLIII-CNT2 | Mechatrolink-III 트리거 출력 Slave Node | 카운터 입력, 트리거 출력 2ch |

AXL 라이브러리는 기능별로 초기화 등의 기초함수와 관련된 AXL, Motion에 관련된 AXM, Digital I/O에 관련된 AXD, Analog I/O에 관련된 AXA에 대한 라이브러리로 나누어져 있다. 라이브러리를 헤더파일을 기준으로 분류하면 다음과 같다.

AXL (Base Command)

AXL은 장착된 모든 보드 및 모듈들을 자동으로 인식하고 초기화하고 인터페이스 하기 위한 기초 라이브러리이다.

AXM (Motion Command)

AXM은 모션제어 관련 라이브러리로서, 직선 및 원호 보간 기능을 포함한 상위 고급 모션 함수들을 제공하며, 프로그래밍에 좀더 편리한 인터페이스를 제공한다.

AXD (DIO Command)

AXD는 DIO 모듈 및 보드를 통하여 디지털 신호의 입출력을 제어하는 라이브러리이다.

AXA (AIO Command)

AXA는 AIO 모듈 및 보드를 통하여 아날로그 신호를 디지털 신호로 변환하거나, 디지털 신호를 아날로그 신호로 변환하는 것을 제어하는 라이브러리이다.

AXL 라이브러리의 OS 개발 환경 지원 내역은 다음과 같다.

| OS | 세부 내역 |
|------------------------------------|---------------------------------|
| Windows 98 | 사용자 환경에 따라 지원 가능 |
| Windows 2000 | SP4 이상 |
| Windows XP(32-bit) | SP3 이상 |
| Windows XP(32-bit) + RTX kernel | Windows XP(SP3 이상), RTX(7.0 이상) |
| Windows XP(64-bit) | 지원 |
| Windows 7(32-bit) | Professional 이상 |
| Windows 7(64-bit) | Professional 이상 |

*. Windows 98/2000/XP/7(Microsoft 사 OS), RTX(IntervalZero 사의 Real-Time eXtension kernel)

AXL 라이브러리의 개발 언어 지원 내역은 다음과 같다.

| 개발 언어 | 세부 내역 |
|--------------|---|
| Visual C++ | Visual studio 6.0, 2003, 2005, 2008, 2010 |
| Visual Basic | Visual studio 6.0, 2003, 2005, 2008, 2010 |

| | |
|------------|--------------------------------|
| C# | Visual studio 2005, 2008, 2010 |
| C++Builder | 6.0 이상 |
| Delphi | 7.0 이상 |

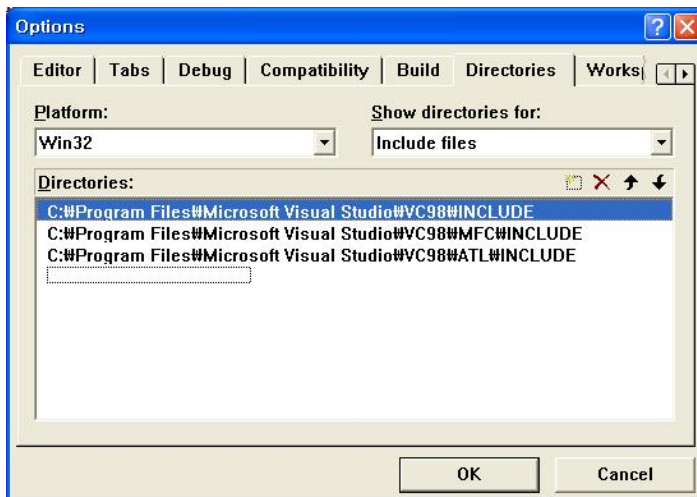
라이브러리 초기화

라이브러리 경로 설정

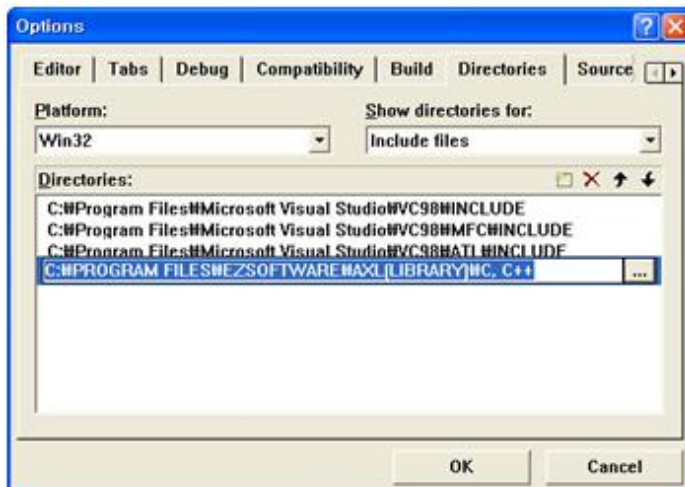
각 개발 프로그램 언어별로 라이브러리 경로를 설정할 수 있으며, 한번 설정된 경로정보는 새로운 프로젝트를 시작할 때마다 따로 자동으로 적용된다.

● Visual C++ 라이브러리 링크하기 (VS6.0 기준)

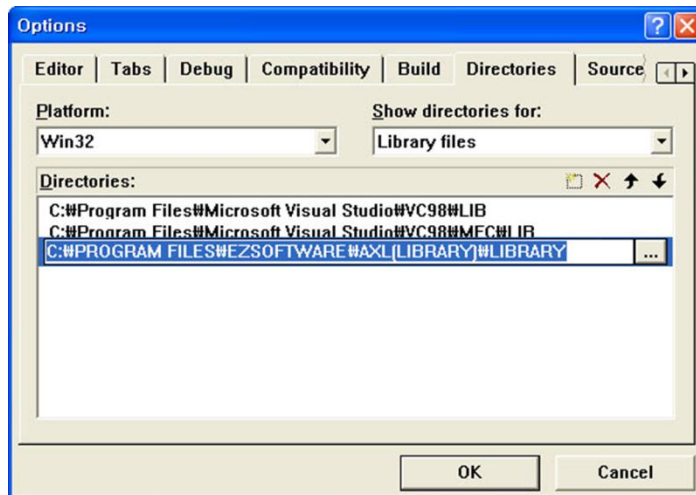
Visual C++를 실행한 후 시스템 메뉴에서 Tools -> Options..를 클릭하여 Directories 탭으로 이동한다. 여기서 Header 파일과 Library 파일을 지정한다.



Header 파일 지정은 Show directories for: 콤보에 Include files를 선택한 다음 Directories: 에 마지막 빈 라인을 클릭하면 오른쪽에 ... 버튼이 나타나는데 이 버튼을 클릭하면 Choose Directory다이얼로그가 나타난다. 이때 WEzSoftware\WAXL(Library)\WC 폴더를 지정한다.



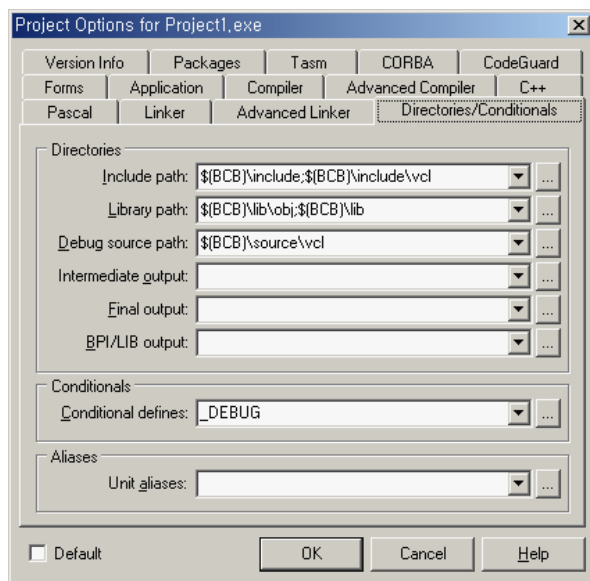
Library 파일 지정은 Show directories for: 콤보에 Library files 선택한 다음 Directories: 에 마지막 빈 라인을 클릭하면 오른쪽에 ... 버튼이 나타나는데 이 버튼을 클릭하면 Choose Directory 다이얼로그가 나타난다. 이때 WEzSoftware WAXL(Library)WLibrary 폴더를 지정한다.



Header 파일과 Library 파일 지정을 완료 했으면 OK 버튼을 클릭한다.

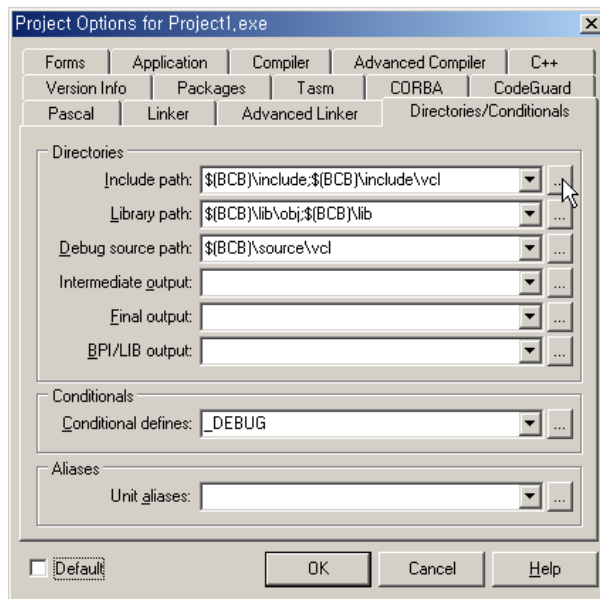
● C++ Builder 라이브러리 링크하기(CB6.0 기준)

C++ Builder를 실행한 후 시스템 메뉴에서 Project -> Options..를 클릭하여 Directories/ Conditionals 탭으로 이동한다. 여기서 Header 파일과 Library 파일을 지정한다.



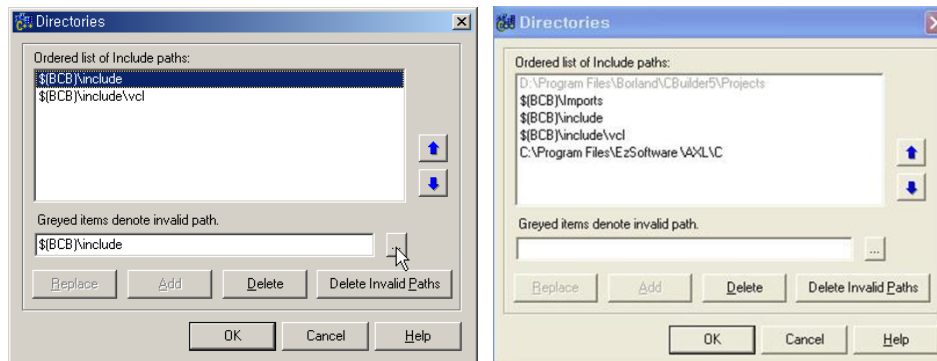
Header 파일 지정은 Directories 안의 Include path: 항목 오른쪽에 ... 버튼을 클릭하면 Directories 다이얼

로고가 표시된다.

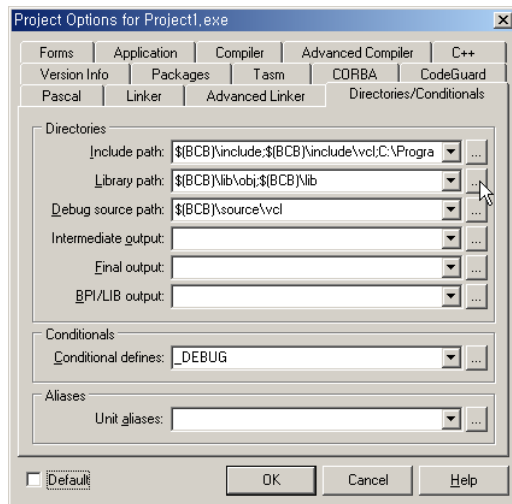


여기서 Greyed items denote invalid path. 항목 오른쪽에 ... 버튼을 클릭하면 폴더 찾아보기 다이얼로그가 표시난다.

이때 WEzSoftwareWAXL(Library)WC 폴더를 지정하고 OK 버튼을 클릭한다.

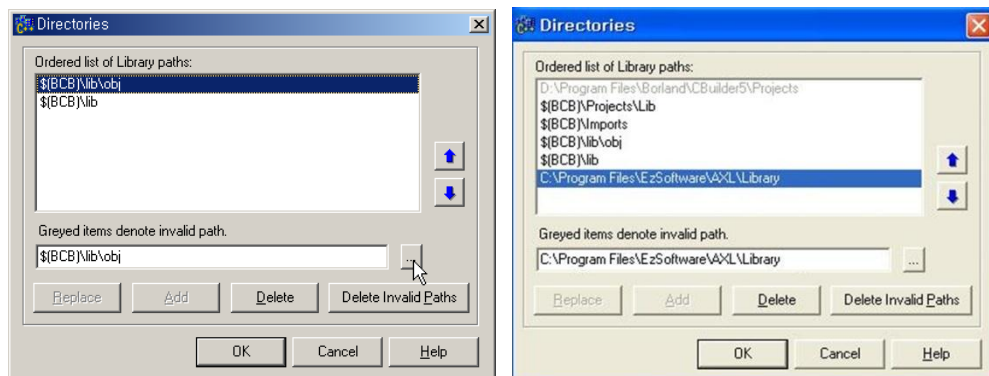


Library 파일 지정은 Directories 안의 Library path: 항목 오른쪽에 ... 버튼을 클릭하면 Directories 다이얼로그가 나온다.

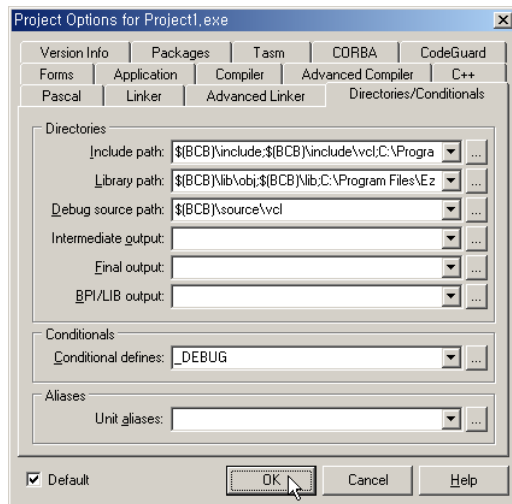


여기서 Greyed items denote invalid path. 항목 오른쪽에 ... 버튼을 클릭하면 폴더 찾아보기 다이얼로그가 나타난다.

이때 WEzSoftwareWAXL(Library)WLibrary 폴더를 지정하고 OK 버튼을 클릭한다.

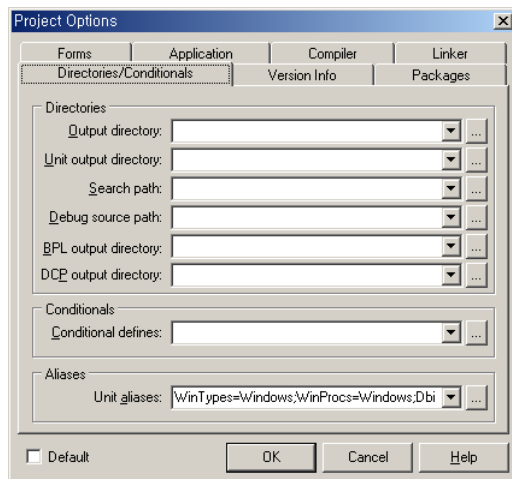


Header 파일과 Library 파일 지정을 완료했으면 왼쪽 하단에 있는 Default를 체크한 다음 OK 버튼을 클릭한다..

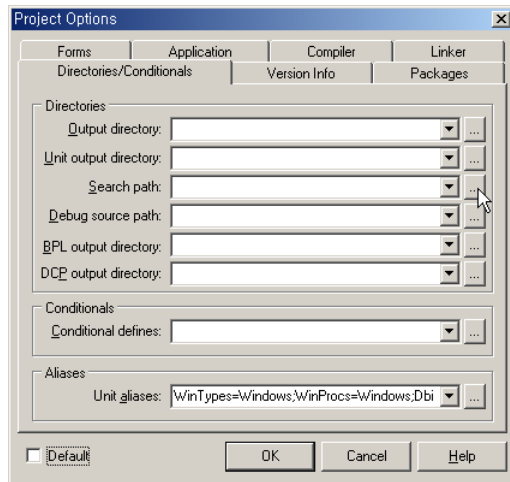



● Delphi (7.0 기준) 라이브러리 링크하기

Delphi를 실행한 후 시스템 메뉴에서 Project -> Options..를 클릭하여 Directories/ Conditionals 탭으로 이동한다. 여기서 Header 파일과 Library 파일을 지정한다.

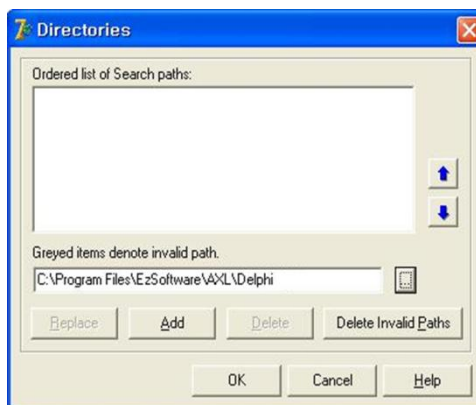
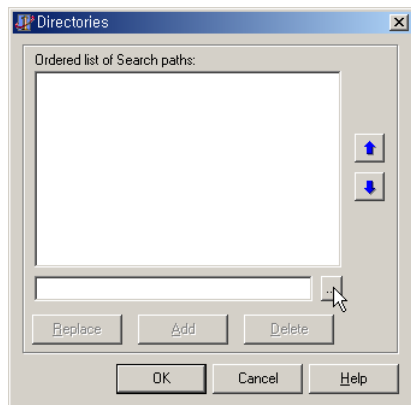


Header 파일 지정은 Directories 안의 Search path: 항목 오른쪽에 ... 버튼을 클릭하면 Directories 다이얼로그가 나온다.

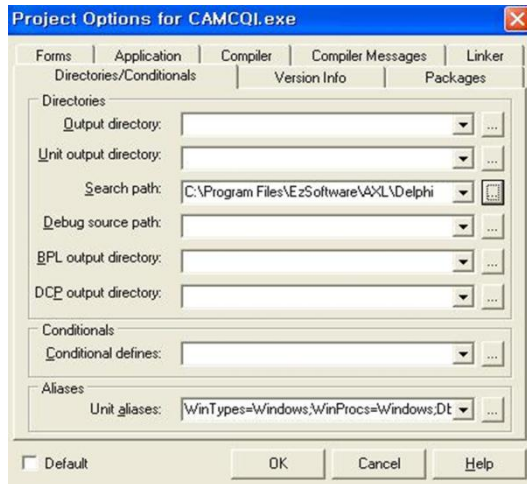


여기서 Greyed items denote invalid path. 항목 오른쪽에  버튼을 클릭하면 폴더 찾아보기 다이얼로그가 나타난다.

이때 WEzSoftwareWAXL(Library)WDelphi 폴더를 지정하고 OK 버튼을 클릭한다.

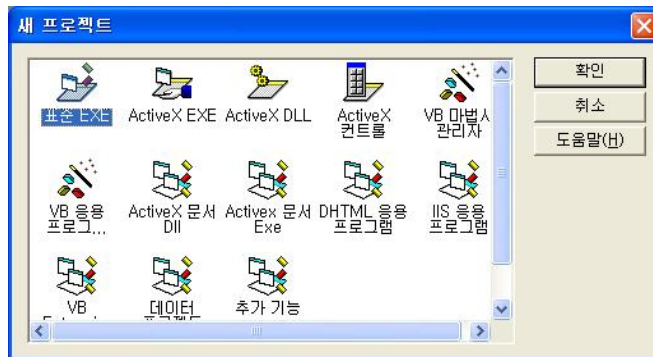


Header 파일 지정을 완료 했으면 왼쪽 하단에 있는 Default를 체크한 다음 OK 버튼을 클릭한다.



● Visual Basic 라이브러리 링크하기

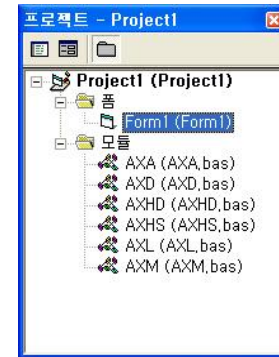
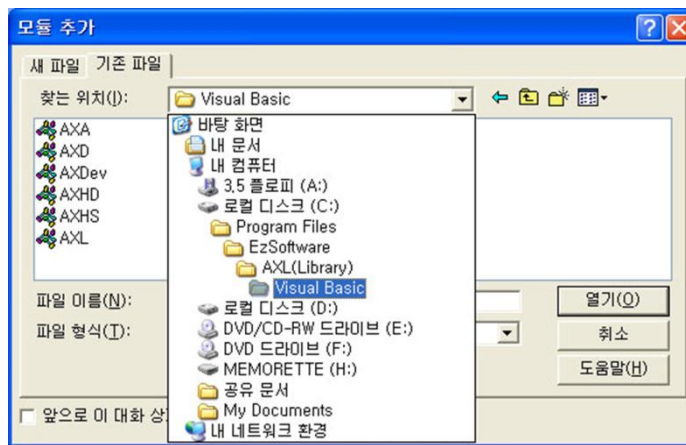
Visual Basic을 실행한 후 시스템 메뉴에서 파일(F) -> 새 프로젝트(N)를 클릭하여 표준 EXE 를 선택한 다음 확인 버튼을 클릭한다.



오른 쪽에 있는 프로젝트 창에서 마우스 오른쪽 버튼을 클릭하여 추가(A) -> 모듈(M)을 선택한다.



여기서 기존 파일 탭을 이동한 다음 찾는 위치(I)를 클릭하여 WEzSoftwareWAXLWVisual Basic 폴더를 지정한다.



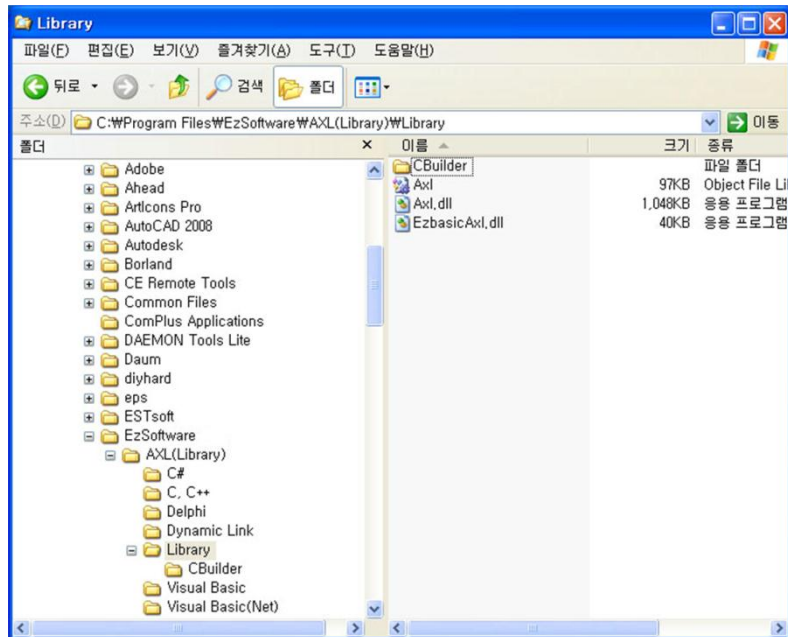
관계된 모든 모듈을 하나씩 열기를 수행한다.

모듈을 링크한 다음 해당 함수를 사용할 수 있다.

DLL 파일 복사하기

해당 언어 라이브러리 링크 작업을 완료하였다면 프로젝트 컴파일시 자동으로 생성되는 실행파일이 있는 폴더에 AXL.dll, EzBasicAXL.dll 총 2개의 파일이 있어야 된다.

파일 위치는 WEzSoftware\WAXL(Library)\WLibrary에 있으며, 사용자가 복사하여 사용하여야 한다. 단, Visual Basic에서는 해당 운영체제의 system (운영체제가 Windows 2000일 경우 WWINNT\system32) 폴더에 직접 복사하여 사용하여야 한다.



축번호, 접점번호, 채널번호 할당 방법

기본 원리

1. 같은 시스템에 여거가지 제품이 장착된 경우 보드 제품이 낮은 번호로 넘버링된다.

모션에 관계된 제어요소를 예를 들면, PCI-N804/404 제품에 연결된 모터부터 넘버링되며, 이어서 PCI-R1604 또는 PCI-R1604-MLII와 연결된 모터가 넘버링된다.

2. 같은 종류의 모듈끼리 개별적으로 넘버링된다.

모션에 관계된 제어 요소는 축번호로 넘버링되며, 디지털 입출력 제품은 모듈 번호 및 접점 번호로, 아날로그 입출력 제품은 모듈 번호 및 채널 번호로 넘버링된다.

3. 여러 개의 보드가 장착되어 있을 경우 CPU에 가까운 쪽 베이스보드에서부터 넘버링된다.

일반적인 PC또는 IPC 시스템에서 CPU에 가까운 쪽부터 PCI슬롯의 순서가 정해져 있는데, 시스템의 설정에 따라서 CPU모듈에서 먼 쪽부터 번호가 시작되거나 임의로 각 슬롯의 번호를 설정해줄 수 있는 시스템도 있다.

4. 같은 베이스보드에서는 J1,J2,J3,J4의 순서 또는 Network address 가 낮은 순서대로 넘버링 된다.

카드 타입 집중형 제어 보드의 넘버링 순서는 외부와 연결되는 커넥터 번호가 낮은 순서부터 넘버링되며, 분산 제어형 제어 보드는 연결된 슬레이브 노드에 설정된(H/W 스위치로 설정됨) 네트워크 주소 값이 작은 것부터 넘버링 된다.

5. EzConfig 의 정보 확인 창에서 자세한 내용을 확인 한다.

EzSoftware 를 설치하면 제어 리소스를 확인하고 축번호, 접점번호, 채널번호를 확인할 수 있는 EzConfig 프로그램을 사용할 수 있다.

주의 사항) PCI-R1604, PCI-R1604-MLII 제품은 Netowrk master 보드이므로, 슬레이스 노드를 장착하여야만 프로그램이 가능하다. 또한 슬레이브 노드의 추가 및 제거로 인해 제어 리소스의 번호가 바뀔 수 있으므로 주의 하여야 한다.

축번호, 접점번호, 채널번호의 할당 예(PCI slot 장착 순서)

| PCI slot #1 : BPFR(보드 번호 : 0) | | | |
|--------------------------------------|------------------------|--------------------------------------|-------------------|
| SIO-DB32(SUB1) : J1 connector | | SIO-DB32(SUB3) : J3 connector | |
| 모듈번호 : 0 | 접점번호(DI/DO): 0~15/0~15 | 모듈번호 : 2 | 접점번호(DI): 16 ~ 47 |
| SIO-DO32(SUB2) : J2 connector | | SIO-DB32(SUB4) : J4 connector | |
| 모듈번호 : 1 | 접점번호(DO): 16~47 | 모듈번호 : 3 | 접점번호(DI): 48 ~ 79 |

| PCI slot #2 : PCIN804(보드 번호 : 1) | | |
|---|-----------|--|
| SMC-4V04(SUB1) : Embedded(J1/J2 connector) | | |
| 모듈번호 : 0 | 축번호 : 0~3 | |
| SMC-4V04(SUB2) : Embedded(J3/J4 connector) | | |
| 모듈번호 : 1 | 축번호 : 4~7 | |

| PCI slot #3 : PCIR1604(보드 번호 : 3) | | | |
|-----------------------------------|-------------------|-------------------------|---------------------|
| RTEX-DI32(NODE0) | | RTEX-PM(NODE16) | |
| 모듈번호 : 7 | 접점번호(DI): 112~143 | 모듈번호 : 3 | 축번호: 9 |
| RTEX-DO32(NODE1) | | SIO-DB32(NODE17) | |
| 모듈번호 : 8 | 접점번호(DO): 128~159 | 모듈번호 : 9 | 접점번호(DI): 144 ~ 175 |
| RTEX-PM (NODE2) | | A4N(NODE18) | |
| 모듈번호 : 2 | 축번호 : 8 | 모듈번호 : 4 | 축번호: 10 |
| 나머지 Node 번호 비어있음. | | 나머지 Node 번호 비어있음. | |

| PCI slot #4 : BPHD(보드 번호 : 2) | | | |
|-------------------------------|-----------------------|--|-----------------------------|
| SIO-DB32(SUB1) | | SIO-DB32(SUB3) : Embedded(J3 connector) | |
| 모듈번호 : 4 | 접점번호(DI): 80~95/48~79 | 모듈번호 : 6 | 접점번호(DI/DO): 96~111/112~127 |
| SIO-DO32(SUB2) | | SIO-DB32(NODE17) | |
| 모듈번호 : 5 | 접점번호(DO): 80~111 | 모듈번호 : 9 | 접점번호(DI): 144 ~ 175 |

| PCI slot #5 : PCIR1604-MLII(보드 번호 : 4) | | |
|---|------------------------------|--|
| SGDV (NODE ADDR : 0x41) On Net#1 | | |
| 모듈번호 : 5 | 축번호: 11 | |
| SGDV (NODE ADDR : 0x41) On Net#2 | | |
| 모듈번호 : 6 | 축번호: 12 | |
| JEPMCIO2330 (NODE ADDR : 0x61) on NET#1 | | |
| 모듈번호 : 10 | 접점번호(DI/DO): 176~239/160~223 | |
| JEPMC-PL2910 (NODE ADDR : 0x61) on NET#2 | | |
| 모듈번호 : 7 | 축번호: 13~14 | |
| 나머지 Node 번호 비어있음. | | |

Base Command 매뉴얼 정보

본 매뉴얼은 BPHR/BPFR/BPHD 베이스 보드(Base Board), PCI-Nx04/PCI-Dx64 일체형 보드, PCI-PCI-R1604/ PCI-R1604-MLII 네트워크 마스트 보드의 기본 정보 확인 및 초기화등의 기능을 구동하기 위해 필요한 매뉴얼이다.

헤더 파일

C++

AXL.h

Visual Basic

AXL.bas

Delphi

AXL.pas

함수 용어

매뉴얼의 함수 이름

본 매뉴얼에서 사용된 함수 이름들은 접두어(Prefix)에 의해 동작을 구분할 수 있도록 되어있다.

※ 라이브러리 함수 Prefix

Axl : 베이스 보드 및 라이브러리 전용 함수임을 나타낸다. Axl로 시작되는 함수들은 모두 AXL.h 에 정의되어 있다.

Set : 라이브러리 내부 설정 값을 변경한다.

Get : Set과 한 쌍으로 Set함수에 의해 설정된 변수 값을 확인한다.

본 매뉴얼의 인자 이름

본 매뉴얼에서 사용된 함수들의 공통적인 인자들은 다음과 같은 의미를 가진다.

long lBoardNo : 초기화된 베이스 보드(Base Board)의 첫 번째 보드부터 오름차순으로 자동 정렬된다. 보드 번호는 '0'부터 시작한다.

Base Command Quick List

라이브러리 초기화

| Function | Description |
|-----------------------------|--------------------------|
| AxlOpen | 라이브러리를 초기화 한다. |
| AxlClose | 라이브러리 사용을 종료 한다. |
| AxlIsOpened | 라이브러리가 초기화 되어 있는 지 확인한다. |

라이브러리 및 베이스 보드 정보

| Function | Description |
|--|--|
| AxlGetBoardCount | 등록된 베이스 보드의 개수를 확인한다. |
| AxlGetLibVersion | 라이브러리 버전을 확인한다. |
| AxlGetModuleNodeStatus | 네트워크 마스트 보드와 연결된 슬레이브 노드의 연결 상태를 확인한다. |
| AxlGetBoardStatus | 해당 보드가 제어 가능한 상태인지 반환한다. |
| AxlGetLockMode | Network 제품의 Configuration Lock 상태를 반환한다. |
| AxlSetLogLevel | EzSpy 에 출력할 메시지 레벨을 설정한다. |
| AxlGetLogLevel | EzSpy 에 출력할 메시지 레벨을 반환한다. |

Mechatrolink III Command Quick List

공통 커맨드

| Function | Description |
|--------------------------------|--------------------------------|
| AxIM3SetStationNop | 슬레이브 기기의 무효 커맨드로 사용한다. |
| AxIM3GetStationParameter | 슬레이브 기기의 파라미터 설정 값을 반환한다. |
| AxIM3SetStationParameter | 슬레이브 기기의 파라미터 값을 설정한다. |
| AxIM3GetStationIdRd | 슬레이브 기기의 ID 값을 반환한다. |
| AxIM3SetStationConfig | 슬레이브 기기의 셋업을 실시한다. |
| AxIM3GetStationAlarm | 슬레이브 기기의 알람 및 경고 상태 값을 반환한다. |
| AxIM3SetStationAlarmClear | 슬레이브 기기의 알람 및 경고 상태를 해제한다. |
| AxIM3SetStationSyncSet | 슬레이브 기기와의 동기통신을 설정한다. |
| AxIM3SetStationConnect | 슬레이브 기기와의 연결을 설정한다. |
| AxIM3SetStationDisConnect | 슬레이브 기기와의 연결 끊음을 설정한다. |
| AxIM3GetStationStoredParameter | 슬레이브 기기의 비휘발성 파라미터 설정 값을 반환한다. |
| AxIM3SetStationStoredParameter | 슬레이브 기기의 비휘발성 파라미터 값을 설정한다. |
| AxIM3GetStationMemory | 슬레이브 기기의 메모리 설정 값을 반환한다. |
| AxIM3SetStationMemory | 슬레이브 기기의 메모리 값을 설정한다. |

표준 I/O 기기 커넥션

| Function | Description |
|-----------------------------|-------------------------------------|
| AxIM3SetStationAccessMode | 재정열된 슬레이브 기기의 자동 액세스 모드 값을 설정한다. |
| AxIM3GetStationAccessMode | 재정열된 슬레이브 기기의 자동 액세스 모드 설정 값을 반환한다. |
| AxIM3SetAutoSyncConnectMode | 슬레이브 기기의 동기 자동 연결 모드를 설정한다. |
| AxIM3GetAutoSyncConnectMode | 슬레이브 기기의 동기 자동 연결 모드 설정 값을 반환한다. |
| AxIM3SyncConnectSingle | 슬레이브 기기에 대한 단일 동기화 연결을 설정한다. |
| AxIM3SyncDisconnectSingle | 슬레이브 기기에 대한 단일 동기화 연결 끊음을 설정한다. |

표준 I/O 프로파일 커맨드

| Function | Description |
|--------------------|--|
| AxIM3GetStationRWS | 동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정값을 반환한다. |
| AxIM3SetStationRWS | 동기화 상태의 슬레이브 I/O 기기에 대한 데이터값을 설정한다. |

| | |
|------------------------------------|---|
| AxIM3GetStationRWA | 비동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정값을 반환한다. |
| AxIM3SetStationRWA | 비동기화 상태의 슬레이브 I/O 기기에 대한 데이터값을 설정한다. |

Define 문

AXL Library에서 사용하고 Define문 List는 다음과 같다

헤더파일 : #include "AXHS.h"

AXT_MLIII_SLAVE_MODULE_TYPE

| Definition | Value | Explanation |
|-------------------------|-------|--------------------|
| AXT_SMC_R1V04MLIIISV | 20h | SGDV 서보팩 |
| AXT_SMC_R1V04MLIIIPM | 21h | N3MLIII-PMxQ(펄스출력) |
| AXT_SMC_R1V04MLIIISV_MD | 22h | SigmaV-MD(다축 서보팩) |
| AXT_SIO_RDI32MLIII | 83h | N3MLIII-DI32 |
| AXT_SIO_RDO32MLIII | 84h | N3MLIII-DO32T |
| AXT_SIO_RDB32MLIII | 85h | N3MLIII-DB32T |
| AXT_SIO_RAI8MLIII | ABh | N3MLIII-AI16 |
| AXT_SIO_RAO4MLIII | ACh | N3MLIII-AO8 |
| AXT_SIO_RCNT2MLIII | B3h | N3MLIII-CNT |

Base Command Function List

라이브러리 초기화

[AxlOpen](#)

[AxlOpenNoReset](#)

[AxlClose](#)

[AxlIsOpened](#)

[AxlInterruptEnable](#)

[AxlInterruptDisable](#)

라이브러리 및 베이스 보드 정보

[AxlGetBoardCount](#)

[AxlGetLibVersion](#)

[AxlGetModuleNodeStatus](#)

라이브러리 초기화

| Function | Description |
|-------------------------------------|-------------------------------|
| AxlOpen | 라이브러리를 사용가능하도록 준비한다. |
| AxlOnpenNoReset | 초기화 과정없이 라이브러리를 사용가능하도록 준비한다. |
| AxlClose | 라이브러리 사용을 종료 한다. |
| AxlIsOpened | 라이브러리가 초기화 되어 있는 지 확인한다. |
| AxlInterruptEnable | 인터럽트 기능을 사용 가능하도록 설정한다. |
| AxlInterruptDisable | 인터럽트 기능 사용을 중지한다. |

AxlOpen

Purpose

라이브러리를 사용가능하도록 준비한다.

Format

C++

```
DWORD AxlOpen(long lIrqNo)
```

Visual Basic

```
Function AxlOpen(ByVal lIrqNo As Long) As Long
```

Delphi

```
function AxlOpen(lIrqNo : LongInt) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------|---|
| [>>]IrqNo | ISA Type 보드 사용시 IRQ 번호(0 ~ 64), PCI 보드 사용시 의미 없음. |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1001] AXT_RT_OPEN_ERROR : AXL 라이브러리 초기화 실패

[1002] AXT_RT_OPEN_ALREADY : AXL 라이브러리 사용 중임

* [See error code Table for more information on status error codes](#)

Description

라이브러리를 초기화하고 사용자가 ISA Type 보드에서 인터럽트를 사용할 경우 인터럽트를 처리할 IRQ 번호를 등록함으로써 인터럽트를 사용하게 해준다. 이때 PnP 기능이 지원되는 PCI 버스와 cPCI 버스인 경우 IRQ 번호는 자동 할당되므로 사용자가 설정한 IRQ 번호는 무시된다.

라이브러리의 초기화 과정 중 모션 제어 파라미터 및 H/W 에 설정되는 파라미터 들이 초기상태로 변경되며, 사용자는 이 함수를 실행하고 나서 각종 파라미터들을 올바르게 다시 설정하여야 한다.

IRQ 번호는 ISA 버스의 경우에는 베이스 보드에 있는 Irq 설정 Dip S/W 의 설정 값을 넣어주면 되고 PCI 버스나 cPCI 버스의 경우에는 PnP 기능이 지원되므로 nIrqNo 가 자동 할당되므로 아무 값이나 넣어주면 된다. 제품 출고 시 초기 값으로 Irq 7 로 설정되어 있다.

AXL 을 사용하기 위해서는 [AxlOpen](#) 또는 [AxlOpenNoReset](#) 함수를 반드시 호출한 후 사용해야 한다. 그리고 사용이 끝난 후에는 [AxlClose](#) 함수를 호출하여 할당 받은 메모리를 해제해 주어야 한다. 그렇지 않을 경우에 메모리 누수가 생겨 시스템에 문제가 발생할 수도 있다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 라이브러리를 초기화 한다.
// 7은 IRQ를 뜻한다. PCI에서 자동으로 IRQ가 설정된다.
```

```

if (AxlOpen(7) == AXT_RT_SUCCESS)
    AfxMessageBox("라이브러리가 초기화 되었습니다.");
else
    AfxMessageBox("라이브러리가 초기화 되지 못했습니다.");

```

Visual Basic

```

' AXL 라이브러리 사용을 위해 AXHS.bas와 AXL.bas 모듈을 추가 한다.

' 라이브러리를 초기화 한다.
' 7은 IRQ를 뜻한다. PCI에서 자동으로 IRQ가 설정된다.
If AxlOpen(7) = AXT_RT_SUCCESS Then
    MsgBox "라이브러리가 초기화 되었습니다.", vbOKCancel
Else
    MsgBox "라이브러리가 초기화 되지 못했습니다.", vbOKCancel
End If

```

Delphi

```

Uses
    AXL, AXHS      { 통합 라이브러리 추가 }

{ 라이브러리를 초기화 한다. }
{ 7은 IRQ를 뜻한다. PCI에서 자동으로 IRQ가 설정된다. }
if (AxlOpen(7) = AXT_RT_SUCCESS) then
    Application.MessageBox('라이브러리가 초기화 되었습니다.', 'Ajinextek', MB_OK)
else
    Application.MessageBox('라이브러리가 초기화 되지 못했습니다.', 'Ajinextek', MB_OK);
MessageBox

```

See Also

[AxlOpenNoReset](#), [AxlClose](#), [AxlIsOpened](#)

AxlOpenNoReset

Purpose

라이브러리 내부 파라미터 및 H/W 설정 파라미터의 초기화 과정 없이 라이브러리를 사용가능하도록 준비한다.

Format

C++

```
DWORD AxlOpenNoReset(long lIrqNo);
```

Visual Basic

```
Function AxlOpenNoReset(ByVal lIrqNo As Long) As Long
```

Delphi

```
function AxlOpenNoReset (lIrqNo : LongInt) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------|---|
| [>>]IrqNo | ISA Type 보드 사용시 IRQ 번호(0 ~ 64), PCI 보드 사용시 의미 없음. |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1001] AXT_RT_OPEN_ERROR : AXL 라이브러리 초기화 실패

[1002] AXT_RT_OPEN_ALREADY : AXL 라이브러리 사용 중임

* [See error code Table for more information on status error codes](#)

Description

라이브러리를 초기화하지 않고 사용가능하도록 제어 리소스를 연다. 즉 현재 설정된 라이브러리 내부 파라미터들이 유지되며, H/W 에 설정된 파라미터 또한 유지 된다. 사용자가 인터럽트를 사용할 경우 인터럽트를 처리할 IRQ 번호를 등록함으로써 인터럽트를 사용하게 해준다. 이때 PnP 기능이 지원되는 PCI 버스와 cPCI 버스인 경우 IRQ 번호는 자동 할당 되므로 사용자가 설정한 IRQ 번호는 무시된다.

IRQ 번호는 ISA 버스의 경우에는 베이스 보드에 있는 Irq 설정 Dip S/W 의 설정 값을 넣어주면 되고 PCI 버스나 cPCI 버스의 경우에는 PnP 기능이 지원되므로 nIrqNo 가 자동 할당되므로 아무 값이나 넣어주면 된다. 제품 출고 시 초기 값으로 Irq 7 로 설정되어 있다.

AXL 을 사용하기 위해서는 [AxlOpen/AxlOpenNoReset](#) 함수를 반드시 호출한 후 사용해야 한다. 그리고 사용이 끝난 후에는 [AxlClose](#) 함수를 호출하여 할당 받은 메모리를 해제해 주어야 한다. 그렇지 않을 경우에 메모리 누수가 생겨 시스템에 문제가 발생할 수도 있다

Examples

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 라이브러리를 초기화 한다.
// 7은 IRQ를 뜻한다. PCI에서 자동으로 IRQ가 설정된다.
```

```

if (AxlOpenNoReset(7) == AXT_RT_SUCCESS)
    AfxMessageBox("라이브러리가 초기화 되었습니다.");
else
    AfxMessageBox("라이브러리가 초기화 되지 못했습니다.");

```

Visual Basic

```

' 통합 라이브러리 사용을 위해 AXHS.bas와 AXL.bas 모듈을 추가 한다.

' 라이브러리를 초기화 한다.
' 7은 IRQ를 뜻한다. PCI에서 자동으로 IRQ가 설정된다.
If AxlOpenNoReset(7) = AXT_RT_SUCCESS Then
    MsgBox "라이브러리가 초기화 되었습니다.", vbOKCancel
Else
    MsgBox "라이브러리가 초기화 되지 못했습니다.", vbOKCancel
End If

```

Delphi

```

uses
    AXL, AXHS      { 통합 라이브러리 추가 }

{ 라이브러리를 초기화 한다. }
{ 7은 IRQ를 뜻한다. PCI에서 자동으로 IRQ가 설정된다. }
if (AxlOpenNoReset(7) = AXT_RT_SUCCESS) then
    Application.MessageBox('라이브러리가 초기화 되었습니다.', 'Ajinextek', MB_OK)
else
    Application.MessageBox('라이브러리가 초기화 되지 못했습니다.', 'Ajinextek', MB_OK);

```

See Also

[AxlOpen](#), [AxlClose](#), [AxlIsOpened](#)

AxlClose

Purpose

라이브러리 사용을 종료한다.

Format

C++

```
BOOL AxlClose();
```

Visual Basic

```
Function AxlClose() As Byte
```

Delphi

```
function AxlClose(): Boolean; stdcall;
```

Description

라이브러리 사용을 종료한다. 할당 받은 메모리를 해제하므로 라이브러리 사용이 종료되면 반드시 호출해줘야 한다. 다시 사용하기 위해서는 초기화하여야 한다.

Return Values

[0] AXL 라이브러리 종료를 실패했을 경우

[1] AXL 라이브러리 종료를 성공했을 경우

Example

C++

```
// 프로그램의 종료 부분에서 사용함  
AxlClose();
```

Visual Basic

```
\'프로그램의 종료 부분에서 사용함  
AxlClose
```

Delphi

```
{ 프로그램의 종료 부분에서 사용함 }  
AxlClose();
```

See Also

[AxlOpen](#), [AxlIsOpened](#)

AxlIsOpened

Purpose

라이브러리가 초기화 되어 있는 지 확인한다.

Format

C++

```
BOOL AxlIsOpened();
```

Visual Basic

```
Function AxlIsOpened() As Byte
```

Delphi

```
function AxlIsOpened():Boolean; stdcall;
```

Description

라이브러리가 초기화 되어있어 사용 가능한 지 확인한다. 만약 라이브러리가 초기화 되지 않은 상태에서 함수들을 사용하게 되면 여러 가지 문제를 야기하므로 반드시 이 함수를 이용해서 라이브러리가 초기화 되었는지 검사한 후 사용해야 한다

Return Values

[0] AXL 라이브러리가 초기화 되어 있지 않음

[1] AXL 라이브러리가 초기화 되어 있음

Example

C++

```
// 라이브러리가 초기화 되어 있는 지 확인한다.
if(AxlIsOpened())
    AfxMessageBox("라이브러리가 초기화 되어 있습니다.");
else
    AfxMessageBox("라이브러리가 초기화 되지 않았습니다.");
```

Visual Basic

```
⋄ 라이브러리가 초기화 되어 있는 지 확인한다.
If AxlIsOpened Then
    MsgBox "라이브러리가 초기화 되어 있습니다.", vbOKCancel
Else
    MsgBox "라이브러리가 초기화 되지 않았습니다.", vbOKCancel
End If
```

Delphi

```
{ 라이브러리가 초기화 되어 있는 지 확인한다. }
if(AxlIsOpened()) then
    Application.MessageBox('라이브러리가 초기화 되어 있습니다.', 'Ajinextek', MB_OK);
else
    Application.MessageBox('라이브러리가 초기화 되지 않았습니다.', 'Ajinextek', MB_OK);
```

See Also

[AxlOpen](#), [AxlClose](#)

AxlInterruptEnable

Purpose

AXL의 인터럽트 기능을 사용 할 수 있도록 설정한다.

Format

C++

```
DWORD AxlInterruptEnable();
```

Visual Basic

```
Function AxlInterruptEnable() As Long
```

Delphi

```
function AxlInterruptEnable() : DWord; stdcall;
```

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공.

* [See error code Table for more information on status error codes](#)

Description

AXL의 인터럽트 기능을 사용하기 위해 반드시 이 함수를 먼저 실행해야 한다. 즉 인터럽트 기능에 대한 Global mask 기능으로 이 함수를 실행하지 않으면 H/W 내부적으로 인터럽트가 발생하더라도 제어 프로그램으로 인터럽트를 전달하지 않는다.

Examples

C++

```
// 인터럽트 메시지를 받아오기 위하여 윈도우 메시지 사용
BEGIN_MESSAGE_MAP(CCAMCIPDlg, CDialog)
    //}AFX_MSG_MAP
    // 선언함
    ON_MESSAGE(WM_CAMCIP_INTERRUPT, OnTriggerInterrupt)
END_MESSAGE_MAP()

AxlInterruptEnable();
// 인터럽트 발생시 'OnTriggerInterrupt()' 호출 됨
AxmInterruptSetAxis(0, m_hWnd, NULL, NULL);
```

Visual Basic

```

    ' 인터럽트 메시지를 받아오기 위하여 윈도우 메시지 사용
    ' 선언
    Private Sub AxtMsg1_OnMessage1(ByVal wParam As Long, ByVal lParam As Long)
    ' 인터럽트 메시지 처리 구문
    End Sub

    ' 인터럽트 발생시 'AxtMsg1_OnMessage1()' 호출 됨
    AxtMsg1.Message1 = WM_CAMCIP_INTERRUPT
    AxlInterruptEnable
    ' 인터럽트 발생시 'OnTriggerInterrupt()' 호출 됨
```

```
AxmInterruptSetAxis 0, Me.hWnd, WM_CAMCIP_INTERRUPT, 0
```

Delphi

```
{ 인터럽트 메시지를 받아오기 위하여 윈도우 메시지 사용 }  
{ 선언 }  
procedure TForm1.OnInterruptMessage(var Msg : TMessage);  
begin  
    { 인터럽트 메시지 처리 구문 }  
    { 인터럽트 발생시 'OnInterruptMessage()' 호출 됨 }  
    AxlInterruptEnable();  
    AxmInterruptSetAxis(Form1.Handle, WM_CAMCIP_INTERRUPT, nil);  
end;
```

See Also

[AxlInterruptDisable](#)

AxlInterruptDisable

Purpose

AXL의 인터럽트 기능을 사용 중지 한다.

Format

C++

```
DWORD AxlInterruptDisable();
```

Visual Basic

```
Function AxlInterruptDisable() As Long
```

Delphi

```
function AxlInterruptDisable() : DWord; stdcall;
```

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공.

* [See error code Table for more information on status error codes](#)

Description

AxlInterruptEnable 함수를 이용해 인터럽트 사용후 중지 하기 위해 이 함수를 사용한다.

Examples

C++

```
// 인터럽트 사용한후 반드시 disable 로 설정한다.  
AxlInterruptDisable();
```

Visual Basic

```
‘인터럽트 사용한후 반드시 disable 로 설정한다.  
AxlInterruptDisable
```

Delphi

```
{ 인터럽트 사용한후 반드시 disable 로 설정한다.}  
AxlInterruptDisable();
```

See Also

[AxlInterruptEnable](#)

라이브러리 및 베이스 보드 정보

| Function | Description |
|--|--|
| AxlGetBoardCount | AXL 사용을 위해 등록된 보드의 개수를 확인한다. |
| AxlGetLibVersion | 라이브러리 버전을 확인한다. |
| AxlGetModuleNodeStatus | 네트워크 마스터 보드와 연결된 슬레이브 노드의 연결 상태를 확인한다. |
| AxlGetBoardStatus | 해당 보드가 제어 가능한 상태인지 반환 한다. |
| AxlGetKLockMode | Network 제품의 Configuration Lock 상태를 반환한다. |
| AxlSetLogLevel | EzSpy 에 출력할 메시지 레벨을 설정 한다. |
| AxlGetLogLevel | EzSpy 에 출력할 메시지 레벨을 반환 한다. |

AxlGetBoardCount

Purpose

시스템에 등록된 AXL 사용가능한 보드의 개수를 확인한다.

Format

C++

```
DWORD AxlGetBoardCount(long *lpBoardCounts);
```

Visual Basic

```
Function AxlGetBoardCount(ByRef lpBoardCounts As Long) As Long
```

Delphi

```
function AxlGetBoardCount(lpBoardCounts : PLongInt) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-------------------|--------------------------|
| [<<]lpBoardCounts | 베이스 보드 개수 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1053] AXT_RT_NOT_OPEN : AXL 라이브러리가 초기화 되지않음

* [See error code Table for more information on status error codes](#)

Description

등록된 베이스 보드의 개수를 확인한다.

Examples

C++

```
// 등록된 AXL 사용가능 보드의 개수를 확인한다.
long lBoardCounts;
CString strData;

AxlGetBoardCount(&lBoardCounts);
strData.Format("전체 보드 개수는 %d개 입니다.", lBoardCounts);

AfxMessageBox(strData);
```

Visual Basic

```
\' 등록된 보드의 개수를 확인한다.
Dim lBoardCounts As Long
Dim strData As String

AxlGetBoardCount(lBoardCounts)
strData = "전체 보드 개수는 " & CStr(lBoardCounts) & "개 입니다."
MsgBox strData
```

Delphi

```
{ 등록된 보드의 개수를 확인한다. }  
var  
  lBoardCounts : LongInt;  
  strData : String;  
  
begin  
  AxlGetBoardCount (@lBoardCounts);  
  strData := '전체 보드 개수는 ' + IntToStr(lBoardCounts) + '개 입니다.';  
  Application.MessageBox (PCHAR(strData), 'Ajinextek', MB_OK);  
End;
```

See Also

[AxlGetLibVersion](#)

AxlGetLibVersion

Purpose

라이브러리 버전을 확인한다.

Format

C++

```
DWORD AxlGetLibVersion(char *szVersion);
```

Visual Basic

```
Function AxlGetLibVersion(ByRef szVersion As Byte) As Long
```

Delphi

```
function AxlGetLibVersion(szVersion : PChar) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------|--------------------------|
| [>>]szVersion | 라이브러리 버전 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1053] AXT_RT_ERROR_VERSION_READ : AXL 라이브러리의 버전을 읽을 수 없을 때

* [See error code Table for more information on status error codes](#)

Description

라이브러리 버전을 확인한다. 라이브러리가 주기적으로 업데이트 되니 버전 정보를 확인하여 업데이트 여부를 결정할 수 있다.

Examples

C++

```
// 라이브러리 버전을 확인한다.
char szVersion[50];
CString strData;

AxlGetLibVersion(szVersion);
strData.Format("현재 라이브러리 버전은 %s 입니다.", szVersion);

AfxMessageBox(strData);
```

Visual Basic

```
라이브러리 버전을 확인한다.
Dim byVersion(30) As Byte

AxlGetLibVersion byVersion(0)
MsgBox StrConv(CStr(byVersion), vbUnicode)
```

Delphi

```
{ 라이브러리 버전을 확인한다. }  
Var  
    strData : String;  
    szVersion : array [0..29] of Char;  
begin  
    AxlGetLibVersion(@szVersion);  
    strData := '현재 라이브러리 버전은 ' + PCHAR(@szVersion) + '입니다.';  
    Application.MessageBox(PCHAR(strData), 'Ajintek', MB_OK);  
End;
```

See Also

[AxlGetBoardCount](#)

AxlGetModuleNodeStatus

Purpose

네트워크 마스터 보드와 연결된 슬레이브 노드의 현재 연결 상태를 확인한다.

Format

C++

```
DWORD AxlGetModuleNodeStatus(long lBoardNo, long lModulePos);
```

Visual Basic

```
Function AxlGetModuleNodeStatus(ByVal lBoardNo As Long, ByVal  
lModulePos As Long) As Long
```

Delphi

```
function AxlGetModuleNodeStatus(lBoardNo : LongInt; lModulePos :  
LongInt) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModulePos | 마스터 보드와 연결된 슬레이브 노드의 상대적인 위치에 대한 번호 예) PCI-R1604 : Node address 가 작은 것부터(0 ~ 31) 예) PCI-R1604-MLII : NET#1 에 장착된것을장착된 것 Node address 가 작은 것부터(0 ~ 29) |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1054] AXT_RT_NOT_SUPPORT_VERSION : 이 함수가 지원하지 않는 제품일 때

[1101] AXT_RT_INVALID_BOARD_NO : 베이스 보드의 위치값이 유효하지 않을 때

[1102] AXT_RT_INVALID_MODULE_POS : 모듈 위치값이 유효하지 않을 때

[1152] AXT_RT_NETWORK_ERROR : 네트워크 제품의 통신에러가 발생했을 때

[1153] AXT_RT_NETWORK_LOCK_MISMATCH : 네트워크 제품에서 Lock 데이터와 Scan 데이터가 일치하지 않을 때

* [See error code Table for more information on status error codes](#)

Description

네트워크 마스터 보드와 연결된 슬레이브 노드의 연결 상태를 확인한다. 함수의 Return 값이 AXT_RT_SUCCESS(0000)이면 정상적으로 연결된 것을 의미한다. 이외 리턴 코드에 의하여 여러가지 상황으로 해석될 수 있다.

Examples

C++

```
// 네트워크 마스터 보드와 연결된 슬레이브 노드의 연결 상태를 확인한다.  
DWORD dwReturn;  
dwReturn = AxlGetModuleNodeStatus (0, 0);
```

```

if (dwReturn)
    strData.Format ("0번째 보드의 0번째 노드에 연결 문제 발생 에러코드: %d.", dwReturn);
else
    strData.Format ("0번째 보드의 0번째 노드 정상적으로 연결됨");
AfxMessageBox (strData);

```

Visual Basic

```

'네트워크 마스터 보드와 연결된 슬레이브 노드의 연결 상태를 확인한다.
If AxlGetModuleNodeStatus (0,0) = AXT_RT_SUCCESS Then
    MsgBox "0번째 보드의 0번째 노드 정상적으로 연결됨.", vbOKCancel
Else
    MsgBox "0번째 보드의 0번째 노드에 연결 문제 발생.", vbOKCancel
End If

```

Delphi

```

{ 네트워크 마스터 보드와 연결된 슬레이브 노드의 연결 상태를 확인한다. }
var
    dwReturn : DWord;
    lBoardCounts : LongInt;
    strData : String;
begin
    dwReturn = AxlGetModuleNodeStatus (0,0);
    if (dwReturn )then
        strData := '0번째 보드의 0번째 노드에 연결 문제 발생.';
    else
        strData := '0번째 보드의 0번째 노드 정상적으로 연결됨.';
    end;
end;

```

See Also

[AxlGetBoardCount](#)

AxlGetBoardStatus

Purpose

해당 보드가 제어 가능한 상태인지 반환 한다.

Format

C++

```
DWORD AxlGetBoardStatus(long lBoardNo);
```

C#

```
uint AxlGetBoardStatus(int lBoardNo);
```

Visual Basic

```
function AxlGetBoardStatus (ByVal lBoardNo As Long) As Long
```

Delphi

```
function AxlGetBoardStatus(lBoardNo : LongInt) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1053] AXT_RT_NOT_OPEN : 라이브러리 초기화 실패

[1101] AXT_RT_INVALID_BOARD_NO : 베이스 보드의 위치값이 유효하지 않을 때

[1152] AXT_RT_NETWORK_ERROR : 네트워크 제품의 통신에러가 발생했을 때

[1153] AXT_RT_NETWORK_LOCK_MISMATCH : 네트워크 제품에서 Lock 데이터와 Scan 데이터가 일치하지 않을 때

[* See error code Table for more information on status error codes](#)

Description

해당 보드가 제어 가능한 상태인지 반환 하는 함수이다. 제어 가능하다면 라이브러리 초기화 성공을 Return 한다.

Examples

C++

```
// 0번 보드가 제어 가능한 상태인지 반환 한다.
DWORD dwReturn;
dwReturn = AxlGetBoardStatus (0);
if(dwReturn)
    strData.Format("0번째 보드 문제 발생 에러코드: %d.", dwReturn);
else
    strData.Format("0번째 보드가 정상적으로 초기화 됨.");
AfxMessageBox(strData);
```

Visual Basic

```
'0번 보드가 제어 가능한 상태인지 반환 한다.
If AxlGetBoardStatus (0) = AXT_RT_SUCCESS Then
    MsgBox "0번째 보드가 정상적으로 초기화 됨.", vbOKCancel
Else
    MsgBox "0번째 보드 문제 발생.", vbOKCancel
End If
```

Delphi

```
{ 0번 보드가 제어 가능한 상태인지 반환 한다. }
var
    dwReturn : DWord;
    strData : String;
begin
    dwReturn = AxlGetBoardStatus (0);
    if (dwReturn )then
        strData := '0번째 보드 문제 발생.';
    else
        strData := '0번째 보드가 정상적으로 초기화 됨.';
    end;
end;
```

C#

```
// 0번 보드가 제어 가능한 상태인지 반환 한다.
DWORD dwReturn;
dwReturn = CAXL.AxlGetBoardStatus (0);
if(dwReturn)
    String.Format("0번째 보드 문제 발생 에러코드: {0:0}", dwReturn);
else
    String.Format("0번째 보드가 정상적으로 초기화 됨.");
MessageBox.Show(strData);
```

See Also

[AxlGetBoardCount](#)

AxlGetLockMode

Purpose

Network 제품의 Configuration Lock 상태를 반환한다.

Format

C++

```
DWORD AxlGetLockMode(long lBoardNo, WORD* wpLockMode);
```

C#

```
uint AxlGetLockMode (int lBoardNo, ref uint upLockMode);
```

Visual Basic

```
function AxlGetLockMode (ByVal lBoardNo As Long, ByRef wpLockMode As Long) As Long
```

Delphi

```
function AxlGetLockMode (lBoardNo : LongInt; wpLockMode : PWord) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [<<]wpLockMode | Configuration Lock 상태 -[00h] DISABLE -[01h] ENABLE |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1053] AXT_RT_NOT_OPEN : 라이브러리 초기화 실패

[1152] AXT_RT_NETWORK_ERROR : 네트워크 제품의 통신에러가 발생했을 때

[1153] AXT_RT_NETWORK_LOCK_MISMATCH : 네트워크 제품에서 Lock 데이터와 Scan 데이터가 일치하지 않을 때

[* See error code Table for more information on status error codes](#)

Description

Network 제품의 Configuration Lock 상태를 반환 하는 함수이다. Configuration Lock/UnLock 은 Ezconfig Agent 에서 설정 할 수 있으며 사용자 함수로는 설정이 불가능 하다.

Examples

C++

```
// 장착된 보드의 Configuration Lock 상태를 확인한다.
DWORD dwReturn, dwLockStatus;
dwReturn = AxlGetLockMode (0, &dwLockStatus);
if(dwReturn)
{
```

```

if(dwLockStatus == ENABLE)
    AfxMessageBox("0번 보드가 Lock 상태 입니다.")
else
    AfxMessageBox("0번 보드가 UnLock 상태 입니다.");
}

```

Visual Basic

```

'장착된 보드의 Configuration Lock 상태를 확인한다.
Dim dwLockStatus As Long
If AxlGetLockMode 0, dwLockStatus = AXT_RT_SUCCESS Then
    If (dwLockStatus = ENABLE)
        MsgBox "0번째 보드가 Lock 상태 입니다.", vbOKCancel
    Else
        MsgBox "0번째 보드가 UnLock 상태 입니다.", vbOKCancel
    End If
End If

```

Delphi

```

{장착된 보드의 Configuration Lock 상태를 확인한다 }
var
    dwReturn : DWord;
    wLockStatus : Word;
    strData : String;
begin
    dwReturn = AxlGetLockMode (0, @ wLockStatus);
    if (wLockStatus)then
        strData := '0번째 보드가 Lock 상태입니다.';
    else
        strData := '0번째 보드가 UnLock 상태입니다.';
    end;
end;

```

C#

```

// 0번 보드가 제어 가능한 상태인지 반환 한다.
DWORD dwReturn;
WORD wLockStatus;
dwReturn = CAXL.AxlGetLockMode (0, ref wLockStatus);
if(dwReturn)
{
    if(wLockStatus == AXT_USE.ENABLE)
        MessageBox.Show("0번째 보드가 Lock 상태 입니다.");
    else
        MessageBox.Show("0번째 보드가 UnLock 상태 입니다.");
}

```

See Also

[AxlGetBoardCount](#)

AxlSetLogLevel

Purpose

EzSpy 에 출력할 메시지 레벨을 설정 한다.

Format

C++

```
DWORD AxlSetLogLevel (DWORD uLevel);
```

C#

```
uint AxlSetLogLevel (uint uLevel);
```

Visual Basic

```
function AxlSetLogLevel (ByVal uLevel As Long) As Long
```

Delphi

```
function AxlSetLogLevel (uLevel: DWord) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------|---|
| [>>]uLevel | Log 설정 레벨 -[00h] LEVEL_NONE 모든 메시지를 출력하지 않는다. -[01h] LEVEL_ERROR 에러가 발생한 메시지만 출력한다., -[02h] LEVEL_RUNSTOP 모션에서 Run/Stop 관련 메시지를 출력한다. -[03h] LEVEL_FUNCTION 모든 메시지를 출력한다. |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1053] AXT_RT_NOT_OPEN : 라이브러리 초기화 실패

[* See error code Table for more information on status error codes](#)

Description

EzSpy 의 Log Level 을 설정 하는 함수이다.

Examples

C++

```
// EzSpy LogLevel을 모든 메시지 출력 레벨로 설정한다.
DWORD dwReturn;
dwReturn = AxlSetLogLevel (3);
if(dwReturn)
{
    AfxMessageBox("EzPsy Log Level을 모든 메시지 출력으로 설정 합니다.")
}
```

Visual Basic

```
'EzSpy LogLevel을 모든 메시지 출력 레벨로 설정한다.  
If AxlGetLockMode 3 = AXT_RT_SUCCESS Then  
    MsgBox "EzPsy Log Level을 모든 메시지 출력으로 설정 합니다.", vbOKCancel  
End If
```

Delphi

```
{ EzSpy LogLevel을 모든 메시지 출력 레벨로 설정한다. }  
var  
    dwReturn : DWord;  
begin  
    dwReturn = AxlSetLogLevel (3);  
    strData := 'EzPsy Log Level을 모든 메시지 출력으로 설정 합니다.';  
end;
```

C#

```
// EzSpy LogLevel을 모든 메시지 출력 레벨로 설정한다.  
DWORD dwReturn;  
dwReturn = CAXL.AxlSetLogLevel (3);  
if(dwReturn)  
{  
    MessageBox.Show("EzPsy Log Level을 모든 메시지 출력으로 설정 합니다.");  
}
```

See Also

[AxlGetLogLevel](#)

AxlGetLogLevel

Purpose

EzSpy 에 출력할 메시지 레벨을 반환 한다.

Format

C++

```
DWORD AxlGetLogLevel (DWORD *upLevel);
```

C#

```
uint AxlGetLogLevel (ref uint upLevel);
```

Visual Basic

```
function AxlGetLogLevel (ByRef upLevel As Long) As Long
```

Delphi

```
function AxlGetLogLevel (upLevel : PDWORD) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------|--------------------------|
| [<<]upLevel | EzSpy 에 출력할 메시지 레벨 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 라이브러리 초기화 성공

[1053] AXT_RT_NOT_OPEN : 라이브러리 초기화 실패

[* See error code Table for more information on status error codes](#)

Description

EzSpy 에서 출력할 메시지 레벨을 반환하는 함수이다.

Examples

C++

```
// EzSpy출력 메시지 레벨을 반환한다.
DWORD dwReturn;
DWORD dwLevel;
dwReturn = AxlGetLogLevel (0, & dwLevel);
if(dwReturn)
    strData.Format("EzSpy 메시지 출력 Level : %d.", dwLevel);
AfxMessageBox(strData);
```

C#

```
// EzSpy출력 메시지 레벨을 반환한다.
DWORD dwReturn;
DWORD dwLevel;
dwReturn = CAXL.AxlGetLogLevel (0, ref dwLevel);
if(dwReturn)
```

```
String.Format("EzSpy 메시지 출력 Level : {0:0}", dwLevel);  
MessageBox.Show(strData);
```

See Also

[AxISetLogLevel](#)

Mechatrolink III Command Function List

공통 커맨드

[AxlM3SetStationNop](#)

[AxlM3GetStationParameter](#)

[AxlM3SetStationParameter](#)

[AxlM3GetStationIdRd](#)

[AxlM3SetStationConfig](#)

[AxlM3GetStationAlarm](#)

[AxlM3SetStationAlarmClear](#)

[AxlM3SetStationSyncSet](#)

[AxlM3SetStationConnect](#)

[AxlM3SetStationDisConnect](#)

[AxlM3GetStationStoredParameter](#)

[AxlM3SetStationStoredParameter](#)

[AxlM3GetStationMemory](#)

[AxlM3SetStationMemory](#)

표준 I/O 기기 커넥션

[AxlM3SetStationAccessMode](#)

[AxlM3GetStationAccessMode](#)

[AxlM3SetAutoSyncConnectMode](#)

[AxlM3GetAutoSyncConnectMode](#)

[AxlM3SyncConnectSingle](#)

[AxlM3SyncDisconnectSingle](#)

[AxlM3IsOnLine](#)

표준 I/O 프로파일 커맨드

[AxlM3GetStationRWS](#)

[AxlM3SetStationRWS](#)

[AxlM3GetStationRWA](#)

[AxlM3SetStationRWA](#)

공통 커맨드

| Function | Description |
|--|--------------------------------|
| AxIM3SetStationNop | 슬레이브 기기의 무효 커맨드로 사용한다. |
| AxIM3GetStationParameter | 슬레이브 기기의 파라미터 설정 값을 반환한다. |
| AxIM3SetStationParameter | 슬레이브 기기의 파라미터 값을 설정한다. |
| AxIM3GetStationIdRd | 슬레이브 기기의 ID 값을 반환한다. |
| AxIM3SetStationConfig | 슬레이브 기기의 셋업을 실시한다. |
| AxIM3GetStationAlarm | 슬레이브 기기의 알람 및 경고 상태 값을 반환한다. |
| AxIM3SetStationAlarmClear | 슬레이브 기기의 알람 및 경고 상태를 해제한다. |
| AxIM3SetStationSyncSet | 슬레이브 기기와의 동기통신을 설정한다. |
| AxIM3SetStationConnect | 슬레이브 기기와의 연결을 설정한다. |
| AxIM3SetStationDisConnect | 슬레이브 기기와의 연결 끊음을 설정한다. |
| AxIM3GetStationStoredParameter | 슬레이브 기기의 비휘발성 파라미터 설정 값을 반환한다. |
| AxIM3SetStationStoredParameter | 슬레이브 기기의 비휘발성 파라미터 값을 설정한다. |
| AxIM3GetStationMemory | 슬레이브 기기의 메모리 설정 값을 반환한다. |
| AxIM3SetStationMemory | 슬레이브 기기의 메모리 값을 설정한다. |

AxIM3SetStationNop

Purpose

슬레이브 기기의 무효 커맨드로 사용한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationNop (long lBoardNo, long lModuleNo, BYTE
bModuleType);
```

Visual Basic

```
Function AxIM3SetStationNop (ByVal lBoardNo As Long, ByVal lModuleNo
As Long, ByVal bModuleType As Byte) As Long
```

Delphi

```
function AxIM3SetStationNop (lBoardNo : LongInt; lModuleNo : LongInt;
bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공

[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러

* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 무효 커맨드로서 사용한다. 응답은 현재의 상태를 통지한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번째 보드에 0번 슬레이브 기기에 대한 무효 커맨드 실시한다.
AxIM3SetStationNop(0, 0, AXT_SMC_R1V04MLIIISV);
```

Visual Basic

‘ 0번째 보드에 0번 슬레이브 기기에 대한 무효 커맨드 실시한다.

```
AxIM3SetStationNop 0, 0, AXT_SMC_R1V04MLIISV
```

Delphi

Uses

AXL, AXHS { 통합 라이브러리 추가 }

{ 0번째 보드에 0번 슬레이브 기기에 대한 무효 커맨드 실시한다. }

```
AxIM3SetStationNop(0, 0, AXT_SMC_R1V04MLIISV);
```

See Also

[AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3GetStationParameter

Purpose

슬레이브 기기의 파라미터 설정 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationParameter(long lBoardNo, long lModuleNo, WORD wNo, BYTE bSize, BYTE bModuleType, BYTE *pbParam);
```

Visual Basic

```
Function AxIM3GetStationParameter(ByVal lBoardNo As Long, ByVal lModuleNo As Long, ByVal wNo As Integer, ByVal bSize As Byte, ByVal bModuleType As Byte, ByRef pbParam As Byte) As Long
```

Delphi

```
function AxIM3GetStationParameter(lBoardNo : LongInt; lModuleNo : LongInt; wNo : Word; bSize : Byte; bModuleType : Byte; pbParam : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wNo | 슬레이브 기기 공통 파라미터 번호(01h ~ 93h): * 메카트로링크 M3_표준 서보 프로파일 커맨드 설명서의 4.1 공통 파라미터 일람 Table 참조 |
| [>>]bSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]pbParam | 데이터 배열: - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공

[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러

[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 특정 파라미터를 값을 반환한다. 파라미터 번호와 데이터 사이즈를 지정하여 특정 파라미터 값을 반환한다. 존재하지 않는 파라미터를 지정한 경우와 같이 정상적으로 읽기가 완료되지 않은 경우에는 슬레이브 기기에서 검출하여 경고 상태가 된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE pbParam[1];

// 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 2 byte 값을 반환한다.
AxIM3GetStationParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

Visual Basic

```
Dim pbParam(0 To 1) As Byte

'0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 1 byte 값을 반환한다.
AxIM3GetStationParameter 0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..1] of Byte;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 1 byte 값을 반환한다. }
AxIM3GetStationParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

See Also

[AxIM3SetStationNop](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationParameter

Purpose

슬레이브 기기의 파라미터 값을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationParameter(long lBoardNo, long lModuleNo, WORD wNo, BYTE bSize, BYTE bModuleType, BYTE *pbParam);
```

Visual Basic

```
Function AxIM3SetStationParameter(ByVal lBoardNo As Long, ByVal lModuleNo As Long, ByVal wNo As Integer, ByVal bSize As Byte, ByVal bModuleType As Byte, ByRef pbParam As Byte) As Long
```

Delphi

```
function AxIM3SetStationParameter(lBoardNo : LongInt; lModuleNo : LongInt; wNo : Word; bSize : Byte; bModuleType : Byte; pbParam : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wNo | 슬레이브 기기 공통 파라미터 번호(01h ~ 93h): * 메카트로링크 M3_표준 서보 프로파일 커맨드 설명서의 4.1 공통 파라미터 일람 Table 참조 |
| [>>]bSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]pbParam | 데이터 배열: - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공

[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러

[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 파라미터 값을 설정한다. 파라미터 번호, 데이터 사이즈, 파라미터 값을 지정하고 파라미터를 저장합니다. 설정 후에 재 셋업이 필요한 오프라인 파라미터의 설정은 파라미터 저장 후 CONFIG 커맨드를 발행하여 셋업을 실시한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 2 byte 값을 설정한다.
BYTE pbParam[1];
pbParam[0] = 0x01;
pbParam[1] = 0x00;

AxIM3SetStationParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

Visual Basic

```
Dim pbParam(0 To 1) As Byte

pbParam(0) = 1
pbParam(1) = 0

'0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 1 byte 값을 반환한다.
AxIM3SetStationParameter 0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..1] of Byte;

pbParam [0] := 1;
pbParam [1] := 0;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 1 byte 값을 반환한다. }
AxIM3SetStationParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3GetStationIdRd

Purpose

슬레이브 기기의 ID 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationIdRd(long lBoardNo, long lModuleNo, BYTE
    bIdCode, BYTE bOffset, BYTE bSize, BYTE bModuleType, BYTE
    *pbParam);
```

Visual Basic

```
Function AxIM3GetStationIdRd(ByVal lBoardNo As Long, ByVal lModuleNo
    As Long, ByVal bIdCode As Byte, ByVal bOffset As Byte, ByVal bSize,
    ByVal bModuleType As Byte, ByRef pbParam As Byte) As Long
```

Delphi

```
function AxIM3GetStationIdRd(lBoardNo : LongInt; lModuleNo : LongInt;
    bIdCode : Byte; bOffset : Byte; bSize : Byte; bModuleType : Byte;
    pbParam : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bIdCode | 슬레이브 기기에 대한 ID 코드 번호 (01h ~ B8h): * 메카트로링크 M3_표준 서버 프로파일 커맨드 설명서의 ID_CODE 일람 Table 참조 |
| [>>]bOffset | 읽기 OFFSET |
| [>>]bSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]pbParam | 데이터 배열: - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
 [1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
 [1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
 * [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 ID 값을 반환한다. 제품정보를 ID 데이터로 읽는다. ID 데이터의 상세한 스펙은 메카트로링크 M3_표준 서버 프로파일 커맨드 설명서의 ID_CODE 일람 Table 참조한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE pbParam[4];

// 0번 보드의 0번 서보모션 슬레이브 기기에 대한 2번 ID CODE를 4 byte 값으로 반환한다.
AxIM3GetStationIdRd(0, 0, 2, 0, 4, AXT_SMC_R1V04MLIISV, pbParam);
```

Visual Basic

```
Dim pbParam(0 To 3) As Byte

'0번 보드의 0번 서보모션 슬레이브 기기에 대한 2번 ID CODE를 4 byte 값으로 반환한다.
AxIM3GetStationIdRd 0, 0, 2, 0, 4, AXT_SMC_R1V04MLIISV, pbParam
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..3] of Byte;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 대한 2번 ID CODE를 4 byte 값으로 반환한다. }
AxIM3GetStationIdRd(0, 0, 2, 0, 4, AXT_SMC_R1V04MLIISV, pbParam);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationConfig

Purpose

슬레이브 기기의 셋업을 실시한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationConfig(long lBoardNo, long lModuleNo, BYTE
bConfigMode, BYTE bModuleType);
```

Visual Basic

```
Function AxIM3SetStationConfig(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bConfigMode As Byte, ByVal bModuleType As
Byte) As Long
```

Delphi

```
function AxIM3SetStationConfig(lBoardNo : LongInt; lModuleNo :
LongInt; bConfigMode : Byte; bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|------------------|--|
| [>>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>>]bConfigMode | Config 모드: - [01h] 파라미터 재설계 및 셋업 - [02h] 비휘발성 메모리로의 일괄 저장 - [03h] 공장 출하시 파라미터 설정치로 초기화 |
| [>>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 셋업을 실시한다. 실시되는 처리 내용은 제품사양으로 규정된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 서보모션 슬레이브 기기에 셋업을 실시한다.
AxIM3SetStationConfig(0, 0, 1, AXT_SMC_R1V04MLIIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 셋업 실시한다.
AxIM3SetStationConfig 0, 0, 1, AXT_SMC_R1V04MLIIISV
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 서보모션 슬레이브 기기에 셋업 실시한다.}
AxIM3SetStationConfig(0, 0, 1, AXT_SMC_R1V04MLIIISV);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3GetStationAlarm

Purpose

슬레이브 기기의 알람 및 경고 상태 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationAlarm(long lBoardNo, long lModuleNo, WORD
wAlarmRdMod, WORD wAlarmIndex, BYTE bModuleType, WORD *pwAlarmData);
```

Visual Basic

```
Function AxIM3GetStationAlarm(ByVal lBoardNo As Long, ByVal lModuleNo
As Long, ByVal wAlarmRdMod As Integer, ByVal wAlarmIndex As Integer,
ByVal bModuleType As Byte, ByRef pwAlarmData As Integer) As Long
```

Delphi

```
function AxIM3GetStationAlarm(lBoardNo : LongInt; lModuleNo : LongInt;
wAlarmRdMod : Word; wAlarmIndex : Word; bModuleType : Byte;
pwAlarmData : PWord) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wAlarmRdMod | 슬레이브 기기에 대한 알람 읽기 - [00h] 현재의 알람 및 경고 상태 읽기 - [01h] 알람 이력의 읽기(최대 12건) - [02h] 현재 발생중인 알람 및 경고 상세 내용 개별 취득 - [03h] 알람 이력의 상세 내용 개별 취득 |
| [>>]wAlarmIndex | 슬레이브 기기에 대한 알람 발생순위 지정(0 ~ 11): |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]pwAlarmData | 데이터 배열: - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

[* See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 알람 및 경고 상태 값을 반환한다. 현재 발생한 알람 상태, 경고 상태를 알람 및 경고 코드로 반환된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE pwAlarmData[12];

// 0번 보드의 0번 서보모션 슬레이브 기기에 대한 0순위 알람 이력 값을 반환한다.
AxIM3GetStationAlarm(0, 0, 1, 0, AXT_SMC_R1V04MLIIISV, pwAlarmData);
```

Visual Basic

```
Dim pwAlarmData (0 To 11) As Byte

'0번 보드의 0번 서보모션 슬레이브 기기에 대한 0순위 알람 이력 값을 반환한다.
AxIM3GetStationAlarm 0, 0, 1, 0, AXT_SMC_R1V04MLIIISV, pwAlarmData
```

Delphi

```
Uses
  AXL, AXHS      { 통합 라이브러리 추가 }

pwAlarmData: array [0..11] of Byte;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 대한 0순위 알람 이력 값을 반환한다. }
AxIM3GetStationAlarm(0, 0, 1, 0, AXT_SMC_R1V04MLIIISV, pwAlarmData);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationAlarmClear

Purpose

슬레이브 기기의 알람 및 경고 상태를 해제한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationAlarmClear(long lBoardNo, long lModuleNo, WORD
wAlarmClrMod, BYTE bModuleType);
```

Visual Basic

```
Function AxIM3SetStationAlarmClear(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal wAlarmClrMod, ByVal bModuleType As Byte) As
Long
```

Delphi

```
function AxIM3SetStationAlarmClear(lBoardNo : LongInt; lModuleNo :
LongInt; wAlarmClrMod : Word; bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|------------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wAlarmClrMod | 슬레이브 기기에 대한 알람 읽기 - [00h] 현재의 알람 및 경고 상태를 클리어 - [01h] 알람이력의 클리어 |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
 [1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
 [1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
 * [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 알람 및 경고 상태를 해제한다. 이 함수는 슬레이브 기기의 상태를 변경하는 것이며, 요인을 해제하는 기능은 없다. 알람 및 경고 요인을 해소한 후에 본 함수로 상태를 해제한다.

동기 통신 중에 통신 에러, 동기통신 에러가 발생한 경우에는 본 함수를 실행한 후, AxIM3SetStationSyncSet() 함수에 의한 동기통신을 복구해야 한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 서보모션 슬레이브 기기에 대한 현재 알람을 클리어 한다.
AxIM3SetStationAlarmClear(0, 0, 0, AXT_SMC_R1V04MLIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 대한 현재 알람을 클리어 한다.
AxIM3SetStationAlarmClear 0, 0, 0, AXT_SMC_R1V04MLIISV
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 서보모션 슬레이브 기기에 대한 현재 알람을 클리어 한다. }
AxIM3SetStationAlarmClear(0, 0, 0, AXT_SMC_R1V04MLIISV);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationSyncSet

Purpose

슬레이브 기기와 동기통신을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationSyncSet(long lBoardNo, long lModuleNo, BYTE
bModuleType);
```

Visual Basic

```
Function AxIM3SetStationSyncSet(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte) As Long
```

Delphi

```
function AxIM3SetStationSyncSet(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기와 동기통신을 설정한다. 본 함수 완료 후에는 동기통신이 된다. 통신 에러 등에 의해 비동기 통신으로 이행한 경우는 본 함수에 의해 동기통신으로 복구한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"
```

```
// 0번 보드의 0번 서보모션 슬레이브 기기에 대한 동기통신을 설정한다.  
AxIM3SetStationSyncSet(0, 0, AXT_SMC_R1V04MLIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 대한 동기통신을 설정한다.  
AxIM3SetStationSyncSet 0, 0, AXT_SMC_R1V04MLIISV
```

Delphi

```
Uses  
AXL, AXHS      { 통합 라이브러리 추가 }  
  
{ 0번 보드의 0번 서보모션 슬레이브 기기에 대한 동기통신을 설정한다. }  
AxIM3SetStationSyncSet(0, 0, AXT_SMC_R1V04MLIISV);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationConnect

Purpose

슬레이브 기기와의 연결을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationConnect(long lBoardNo, long lModuleNo, BYTE bVer,
    BYTE bComMode, BYTE bComTime, BYTE bProfileType, BYTE bModuleType);
```

Visual Basic

```
Function AxIM3SetStationConnect(ByVal lBoardNo As Long, ByVal
    lModuleNo As Long, ByVal bVer As Byte, ByVal bComTime As Byte, ByVal
    bProfileType As Byte, ByVal bModuleType As Byte) As Long
```

Delphi

```
function AxIM3SetStationConnect(lBoardNo : LongInt; lModuleNo :
    LongInt; bVer : Byte; bComTime : Byte; bProfileType : Byte;
    bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-------------------|--|
| [>>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>>]bVer | Mechatrolink 어플리케이션층 버전(30H) |
| [>>>]bComMode | 통신모드 - [Bit 0] 0 : 동기통신 실시, 1 : 비동기통신 실시 - [Bit 3~2] 00 : 단송통신, 01 : 연송통신 - [Bit 7] 0 : 서버 커맨드 무효, 1 : 서버 커맨드 유효 |
| [>>>]bComTime | 통신주기 설정: 전송주기의 배수 ComTime = 통신주기(1ms) / 전송주기(0.5ms) = 2ms |
| [>>>]bProfileType | 슬레이브 기기에 사용될 프로파일 타입 - [10h] 표준 서버 프로파일 커맨드 설정 - [30h] 표준 I/O 프로파일 커맨드 설정 |
| [>>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
 [1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
 [1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

[* See error code Table for more information on status error codes](#)

Description

슬레이브 기기와의 연결을 설정한다. 본 함수 실행 후 Mechatrolink 통신에 의한 슬레이브 기기 제어를 시작한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 서보모션 슬레이브 기기에 대한 연결을 설정한다.
AxIM3SetStationConnect(0, 0, 0x30, 1, 2, 0x10, AXT_SMC_R1V04MLIIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 대한 연결을 설정한다.
AxIM3SetStationSyncSet 0, 0, &H30, 1, 2, &H10, AXT_SMC_R1V04MLIIISV
```

Delphi

```
Uses
  AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 서보모션 슬레이브 기기에 대한 연결을 설정한다. }
AxIM3SetStationSyncSet(0, 0, $30, 1, 2, $10, AXT_SMC_R1V04MLIIISV);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationDisconnect

Purpose

슬레이브 기기와의 연결 끊음을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationDisconnect(long lBoardNo, long lModuleNo, BYTE
bModuleType);
```

Visual Basic

```
Function AxIM3SetStationDisconnect(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte) As Long
```

Delphi

```
function AxIM3SetStationDisconnect(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|---|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_RT1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기와의 연결 끊음을 설정한다. 이 때, 슬레이브 기기는 현재의 처리를 중단하고, 연결 재확립을 위해 필요한 초기화를 실시한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"
```

```
// 0번 보드의 0번 서보모션 슬레이브 기기에 대한 연결 끊음을 설정한다.  
AxIM3SetStationConnect(0, 0, AXT_SMC_R1V04MLIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 대한 연결 끊음을 설정한다.  
AxIM3SetStationSyncSet 0, 0, AXT_SMC_R1V04MLIISV
```

Delphi

```
Uses  
AXL, AXHS      { 통합 라이브러리 추가 }  
  
{ 0번 보드의 0번 서보모션 슬레이브 기기에 대한 연결 끊음을 설정 한다. }  
AxIM3SetStationSyncSet(0, 0, AXT_SMC_R1V04MLIISV);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3GetStationStoredParameter

Purpose

슬레이브 기기의 비휘발성 메모리 상의 파라미터 설정 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationStoredParameter (long lBoardNo, long lModuleNo,
WORD wNo, BYTE bSize, BYTE bModuleType, BYTE *pbParam);
```

Visual Basic

```
Function AxIM3GetStationStoredParameter (ByVal lBoardNo As Long,
ByVal lModuleNo As Long, ByVal wNo As Integer, ByVal bSize As Byte,
ByVal bModuleType As Byte, ByRef pbParam As Byte) As Long
```

Delphi

```
function AxIM3GetStationStoredParameter (lBoardNo : LongInt;
lModuleNo : LongInt; wNo : Word; bSize : Byte; bModuleType : Byte;
pbParam : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wNo | 슬레이브 기기 공통 파라미터 번호(01h ~ 93h): * 메카트로링크 M3_표준 서버 프로파일 커맨드 설명서의 4.1 공통 파라미터 일람 Table 참조 |
| [>>]bSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]pbParam | 데이터 배열: - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공

[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러

[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 비휘발성 메모리 상의 파라미터 설정 값을 반환한다. 존재하지 않는 파라미터를 지정했을 경우 등 정상적으로 반환이 완료되지 않는 경우에는 슬레이브 기기에서 검출하여 경고 상태가 된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE pbParam[1];

// 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 2 byte 값을 반환한다.
AxIM3GetStationStoredParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

Visual Basic

```
Dim pbParam(0 To 1) As Byte

'0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 1 byte 값을 반환한다.
AxIM3GetStationStoredParameter 0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..1] of Byte;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 1 byte 값을 반환한다. }
AxIM3GetStationStoredParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisconnect](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3SetStationStoredParameter

Purpose

슬레이브 기기의 비휘발성 메모리 상의 파라미터 값을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationStoredParameter(long lBoardNo, long lModuleNo,
WORD wNo, BYTE bSize, BYTE bModuleType, BYTE *pbParam);
```

Visual Basic

```
Function AxIM3SetStationStoredParameter(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal wNo As Integer, ByVal bSize As Byte, ByVal
bModuleType As Byte, ByRef pbParam As Byte) As Long
```

Delphi

```
function AxIM3SetStationStoredParameter(lBoardNo : LongInt;
lModuleNo : LongInt; wNo : Word; bSize : Byte; bModuleType : Byte;
pbParam : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wNo | 슬레이브 기기 공통 파라미터 번호(01h ~ 93h): * 메카트로링크 M3_표준 서보 프로파일 커맨드 설명서의 4.1 공통 파라미터 일람 Table 참조 |
| [>>]bSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]pbParam | 데이터 배열: - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공

[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러

[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 비휘발성 메모리 상의 파라미터 값을 설정한다. 설정 범위 밖의 값을 쓴 경우 등 정상적으로 설정이 완료되지 않는 경우에는 슬레이브 기기가 경고 상태가 된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 2 byte 값을 설정한다.
BYTE pbParam[1];
pbParam[0] = 0x01;
pbParam[1] = 0x00;

AxIM3SetStationStoredParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

Visual Basic

```
Dim pbParam(0 To 1) As Byte

pbParam(0) = 1
pbParam(1) = 0

'0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 2 byte 값을 설정한다.
AxIM3SetStationStoredParameter 0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..1] of Byte;

pbParam [0] := 1;
pbParam [1] := 0;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 인코더 파라미터 2 byte 값을 설정한다. }
AxIM3SetStationStoredParameter(0, 0, 1, 2, AXT_SMC_R1V04MLIIISV, pbParam);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#), [AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#), [AxIM3SetStationConnect](#), [AxIM3SetStationDisconnect](#), [AxIM3GetStationStoredParameter](#), [AxIM3GetStationMemory](#), [AxIM3SetStationMemory](#)

AxIM3GetStationMemory

Purpose

슬레이브 기기의 가상 메모리 상의 설정 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationMemory(long lBoardNo, long lModuleNo, WORD wSize,
DWORD dwAddress, BYTE bModuleType, BYTE bMode, BYTE bDataType, BYTE
*pbData);
```

Visual Basic

```
Function AxIM3GetStationMemory(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal wSize As Integer, ByVal dwAddress As Long,
ByVal bModuleType As Byte, ByVal bMode, ByVal bDataType As Byte,
ByRef pbData As Byte) As Long
```

Delphi

```
function AxIM3GetStationMemory(lBoardNo : LongInt; lModuleNo :
LongInt; wSize : Word; dwAddress : DWord; bModuleType : Byte; bMode :
Byte; bDataType : Byte; pbData : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]dwAddress | 읽기 선두 어드레스 |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]bMode | 커맨드 파라미터 모드 - [01h] 휘발성 메모리 - [02h] 비휘발성 메모리 |
| [>>]bDataType | 커맨드 파라미터 데이터 타입 - [01h] Byte 타입 - [02h] Short 타입 - [03h] Long 타입 |
| [<<]pbParam | 데이터 배열: |

| | |
|--|--|
| | <ul style="list-style-type: none"> - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |
|--|--|

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
 [1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
 [1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
 * [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 가상 메모리 읽기 선두 어드레스, 데이터 사이즈를 지정하여 가상 메모리상의 설정값을 반환한다. 설정 범위 밖의 선두어드레스나 데이터 사이즈를 지정한 경우와 같이 정상적으로 읽기가 완료되지 않은 경우에는 슬레이브 기기에서 경고 상태가 된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE pbParam[4];

// 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 정축 소프트 리밋에 대한 4 Byte 메모리 값을 반환. 한다
AxIM3GetStationMemory(0, 0, 4, 0x10098, AXT_SMC_R1V04MLIISV, 1, 3, pbData);
```

Visual Basic

```
Dim pbParam(0 To 3) As Byte

'0번 보드의 0번 서보모션 슬레이브 기기에 설정된 정축 소프트 리밋에 대한 4 Byte 메모리 값을 반환한다.
AxIM3GetStationMemory 0, 0, 4, &H10098, AXT_SMC_R1V04MLIISV, 1, 3, pbData
```

Delphi

```
Uses
  AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..3] of Byte;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 정축 소프트 리밋에 대한 4 Byte 메모리 값을 반환한다. }
AxIM3GetStationMemory(0, 0, 4, $10098, AXT_SMC_R1V04MLIISV, 1, 3, pbData);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisConnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3SetStationMemory](#)

AxIM3SetStationMemory

Purpose

슬레이브 기기의 가상 메모리의 값을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationMemory(long lBoardNo, long lModuleNo, WORD wSize,
DWORD dwAddress, BYTE bModuleType, BYTE bMode, BYTE bDataType, BYTE
*pbData);
```

Visual Basic

```
Function AxIM3SetStationMemory(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal wSize As Integer, ByVal dwAddress As Long,
ByVal bModuleType As Byte, ByVal bMode, ByVal bDataType As Byte,
ByRef pbData As Byte) As Long
```

Delphi

```
function AxIM3SetStationMemory(lBoardNo : LongInt; lModuleNo :
LongInt; wSize : Word; dwAddress : DWord; bModuleType : Byte; bMode :
Byte; bDataType : Byte; pbData : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]wSize | 데이터 사이즈: - [01h] 1 Byte - [02h] 2 Byte - [03h] 3 Byte - [04h] 4 Byte |
| [>>]dwAddress | 읽기 전 두 어드레스 |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]bMode | 커맨드 파라미터 모드 - [01h] 휘발성 메모리 - [02h] 비휘발성 메모리 |
| [>>]bDataType | 커맨드 파라미터 데이터 타입 - [01h] Byte 타입 - [02h] Short 타입 - [03h] Long 타입 |
| [<<]pbParam | 데이터 배열: |

| | |
|--|--|
| | <ul style="list-style-type: none"> - pbParam[03h] h-hi-Byte - pbParam[02h] h-lo-Byte - pbParam[01h] l-hi-Byte - pbParam[00h] l-lo-Byte |
|--|--|

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
 [1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
 [1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
 * [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 가상 메모리의 값을 설정한다. 가상 메모리의 선두어드레스, 데이터 사이즈, 데이터를 지정하고 메모리상에 데이터를 설정한다. 설정범위 밖의 선두어드레스나 데이터 크기를 지정한 경우 등으로 정상적으로 쓰기가 완료되지 않은 경우에는 슬레이브 기기가 경고 상태가 된다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE pbParam[4];

// 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 정축 소프트 리밋에 대한 4 Byte 메모리 값을 설정한다.
AxIM3SetStationMemory(0, 0, 4, 0x10098, AXT_SMC_R1V04MLIISV, 1, 3, pbData);
```

Visual Basic

```
Dim pbParam(0 To 3) As Byte

'0번 보드의 0번 서보모션 슬레이브 기기에 설정된 정축 소프트 리밋에 대한 4 Byte 메모리 값을 설정한다.
AxIM3SetStationMemory 0, 0, 4, &H10098, AXT_SMC_R1V04MLIISV, 1, 3, pbData
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pbParam: array [0..3] of Byte;

{ 0번 보드의 0번 서보모션 슬레이브 기기에 설정된 정축 소프트 리밋에 대한 4 Byte 메모리 값을 설정한다. }
AxIM3SetStationMemory(0, 0, 4, $10098, AXT_SMC_R1V04MLIISV, 1, 3, pbData);
```

See Also

[AxIM3SetStationNop](#), [AxIM3GetStationParameter](#), [AxIM3SetStationParameter](#), [AxIM3GetStationIdRd](#),
[AxIM3SetStationConfig](#), [AxIM3GetStationAlarm](#), [AxIM3SetStationAlarmClear](#), [AxIM3SetStationSyncSet](#),
[AxIM3SetStationConnect](#), [AxIM3SetStationDisconnect](#), [AxIM3GetStationStoredParameter](#),
[AxIM3SetStationStoredParameter](#), [AxIM3GetStationMemory](#)

표준 I/O 기기 커백션

| Function | Description |
|---|-------------------------------------|
| AxIM3SetStationAccessMode | 재정열된 슬레이브 기기의 자동 액세스 모드 값을 설정한다. |
| AxIM3GetStationAccessMode | 재정열된 슬레이브 기기의 자동 액세스 모드 설정 값을 반환한다. |
| AxIM3SetAutoSyncConnectMode | 슬레이브 기기의 동기 자동 연결 모드를 설정한다. |
| AxIM3GetAutoSyncConnectMode | 슬레이브 기기의 동기 자동 연결 모드 설정 값을 반환한다. |
| AxIM3SyncConnectSingle | 슬레이브 기기에 대한 단일 동기화 연결을 설정한다. |
| AxIM3SyncDisconnectSingle | 슬레이브 기기에 대한 단일 동기화 연결 끊음을 설정한다. |
| AxIM3IsOnLine | 슬레이브 기기와 연결 상태를 확인한다. |

AxIM3SetStationAccessMode

Purpose

재정렬된 슬레이브 기기의 자동 액세스 모드 값을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationAccessMode(long lBoardNo, long lModuleNo, BYTE
bModuleType, BYTE bRWSMode);
```

Visual Basic

```
Function AxIM3SetAutoSyncConnectMode(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte, ByVal
dwAutoSyncConnectMode As Long) As Long
```

Delphi

```
function AxIM3SetAutoSyncConnectMode(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte; dwAutoSyncConnectMode : DWord) : DWord;
stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]bRWSMode | 자동 액세스 모드 - [00h] 비자동 모드 - [01h] 자동 모드 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

재정렬된 슬레이브 기기의 자동 액세스 모드 값을 설정한다.

Example

C++

```
#include "AXL.h"          // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 DI 슬레이브 기기에 자동 액세스 모드 설정한다.
AxlM3SetStationAccessMode(0, 0, AXT_SIO_RDI32MLIII, 1);
```

Visual Basic

```
'0번 보드의 0번 DI 슬레이브 기기에 자동 액세스 모드 설정한다.
AxlM3SetStationAccessMode 0, 0, AXT_SIO_RDI32MLIII, 1
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 DI 슬레이브 기기에 자동 액세스 모드 설정한다. }
AxlM3SetStationAccessMode(0, 0, AXT_SIO_RDI32MLIII, 1);
```

See Also

[AxlM3GetStationAccessMode](#), [AxlM3SetAutoSyncConnectMode](#), [AxlM3GetAutoSyncConnectMode](#),
[AxlM3SyncConnectSingle](#), [AxlM3SyncDisconnectSingle](#), [AxlM3IsOnLine](#)

AxIM3GetStationAccessMode

Purpose

재정렬된 슬레이브 기기의 자동 액세스 모드 설정 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationAccessMode(long lBoardNo, long lModuleNo, BYTE
bModuleType, BYTE bRWSMode);
```

Visual Basic

```
Function AxIM3GetAutoSyncConnectMode(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte, ByVal
dwAutoSyncConnectMode As Long) As Long
```

Delphi

```
function AxIM3GetStationAccessMode(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte; bRWSMode : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]bRWSMode | 자동 액세스 모드 - [00h] 비자동 모드 - [01h] 자동 모드 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

재정렬된 슬레이브 기기의 자동 액세스 모드 설정 값을 반환한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

BYTE bRWMode;

// 0번 보드의 0번 DI 슬레이브 기기에 자동 액세스 모드 설정한다.
AxIM3SetStationAccessMode(0, 0, AXT_SIO_RDI32MLIII, &bRWMode);
```

Visual Basic

```
BYTE bRWMode;

'0번 보드의 0번 DI 슬레이브 기기에 자동 액세스 모드 설정한다.
AxIM3SetStationAccessMode 0, 0, AXT_SIO_RDI32MLIII, bRWMode
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 DI 슬레이브 기기에 자동 액세스 모드 설정한다. }
AxIM3SetStationAccessMode(0, 0, AXT_SIO_RDI32MLIII, bRWMode);
```

See Also

[AxIM3SetStationAccessMode](#), [AxIM3SetAutoSyncConnectMode](#), [AxIM3GetAutoSyncConnectMode](#),
[AxIM3SyncConnectSingle](#), [AxIM3SyncDisconnectSingle](#), [AxIM3IsOnLine](#)

AxIM3SetAutoSyncConnectMode

Purpose

슬레이브 기기의 동기 자동 연결 모드를 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetAutoSyncConnectMode(long lBoardNo, long lModuleNo, BYTE
bModuleType, DWORD dwAutoSyncConnectMode);
```

Visual Basic

```
Function AxIM3SetAutoSyncConnectMode(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte, ByVal
dwAutoSyncConnectMode As Long) As Long
```

Delphi

```
function AxIM3SetAutoSyncConnectMode(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte; dwAutoSyncConnectMode : DWord) : DWord;
stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]dwAutoSyncConnectMode | 동기 자동 액세스 모드 - [00h] 동기 비자동 모드 - [01h] 동기 자동 모드 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 동기 자동 연결 모드를 설정한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

// 0번 보드의 0번 서보모션 슬레이브 기기에 자동 연결 모드 설정한다.
AxIM3SetStationAccessMode(0, 0, AXT_SMC_R1V04MLIIISV, 1);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 자동 연결 모드 설정한다.
AxIM3SetStationAccessMode 0, 0, AXT_SMC_R1V04MLIIISV, 1
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 서보모션 슬레이브 기기에 자동 연결 모드 설정한다. }
AxIM3SetStationAccessMode(0, 0, AXT_SMC_R1V04MLIIISV, 1);
```

See Also

[AxIM3SetStationAccessMode](#), [AxIM3GetStationAccessMode](#), [AxIM3GetAutoSyncConnectMode](#),
[AxIM3SyncConnectSingle](#), [AxIM3SyncDisconnectSingle](#), [AxIM3IsOnLine](#)

AxIM3GetAutoSyncConnectMode

Purpose

슬레이브 기기의 동기 자동 연결 모드 설정값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetAutoSyncConnectMode(long lBoardNo, long lModuleNo, BYTE bModuleType, DWORD *dwpAutoSyncConnectMode);
```

Visual Basic

```
Function AxIM3GetAutoSyncConnectMode(ByVal lBoardNo As Long, ByVal lModuleNo As Long, ByVal bModuleType As Byte, ByRef dwpAutoSyncConnectMode As Long) As Long
```

Delphi

```
function AxIM3GetAutoSyncConnectMode(lBoardNo : LongInt; lModuleNo : LongInt; bModuleType : Byte; dwpAutoSyncConnectMode : PDWord) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|---------------------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]dwAutoSyncConnectMode | 동기 자동 액세스 모드 - [00h] 동기 비자동 모드 - [01h] 동기 자동 모드 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기의 동기 자동 연결 모드 설정값을 반환한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

DWORD  dwpAutoSyncConnectMode;

// 0번 보드의 0번 서보모션 슬레이브 기기에 자동 연결 모드 설정값을 반환한다.
AxIM3SetStationAccessMode(0, 0, AXT_SMC_R1V04MLIIISV, &dwpAutoSyncConnectMode);
```

Visual Basic

```
Dim dwpAutoSyncConnectMode As LONG

'0번 보드의 0번 서보모션 슬레이브 기기에 자동 연결 모드 설정값을 반환한다.
AxIM3SetStationAccessMode 0, 0, AXT_SMC_R1V04MLIIISV, dwpAutoSyncConnectMode
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

dwpAutoSyncConnectMode : LongInt

{ 0번 보드의 0번 서보모션 슬레이브 기기에 자동 연결 모드 설정값을 반환한다. }
AxIM3SetStationAccessMode(0, 0, AXT_SMC_R1V04MLIIISV, dwpAutoSyncConnectMode);
```

See Also

[AxIM3SetStationAccessMode](#), [AxIM3GetStationAccessMode](#), [AxIM3SetAutoSyncConnectMode](#),
[AxIM3SyncConnectSingle](#), [AxIM3SyncDisconnectSingle](#), [AxIM3IsOnLine](#)

AxIM3SyncConnectSingle

Purpose

슬레이브 기기에 대한 단일 동기화 연결을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SyncConnectSingle(long lBoardNo, long lModuleNo, BYTE
bModuleType);
```

Visual Basic

```
Function AxIM3SyncConnectSingle(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte) As Long
```

Delphi

```
function AxIM3SyncConnectSingle(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기에 대한 단일 동기화 연결을 설정한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"
```

```
// 0번 보드의 0번 서보모션 슬레이브 기기에 단일 동기화 연결을 설정한다.
AxIM3SyncConnectSingle(0, 0, AXT_SMC_R1V04MLIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 단일 동기화 연결을 설정한다.
AxIM3SyncConnectSingle 0, 0, AXT_SMC_R1V04MLIISV
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 서보모션 슬레이브 기기에 단일 동기화 연결을 설정한다. }
AxIM3SyncConnectSingle(0, 0, AXT_SMC_R1V04MLIISV);
```

See Also

[AxIM3SetStationAccessMode](#), [AxIM3GetStationAccessMode](#), [AxIM3SetAutoSyncConnectMode](#),
[AxIM3GetAutoSyncConnectMode](#), [AxIM3SyncDisconnectSingle](#), [AxIM3IsOnLine](#)

AxIM3SyncDisconnectSingle

Purpose

슬레이브 기기에 대한 단일 동기화 연결을 해제한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SyncDisconnectSingle(long lBoardNo, long lModuleNo, BYTE
bModuleType);
```

Visual Basic

```
Function AxIM3SyncDisconnectSingle(ByVal lBoardNo As Long, ByVal
lModuleNo As Long, ByVal bModuleType As Byte) As Long
```

Delphi

```
function AxIM3SyncDisconnectSingle(lBoardNo : LongInt; lModuleNo :
LongInt; bModuleType : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기에 대한 단일 동기화 연결을 해제한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"
```

```
// 0번 보드의 0번 서보모션 슬레이브 기기에 단일 동기화 연결을 해제한다.
AxIM3SyncDisconnectSingle(0, 0, AXT_SMC_R1V04MLIIISV);
```

Visual Basic

```
'0번 보드의 0번 서보모션 슬레이브 기기에 단일 동기화 연결을 해제한다.
AxIM3SyncDisconnectSingle 0, 0, AXT_SMC_R1V04MLIIISV
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

{ 0번 보드의 0번 서보모션 슬레이브 기기에 단일 동기화 연결을 해제한다. }
AxIM3SyncDisconnectSingle(0, 0, AXT_SMC_R1V04MLIIISV);
```

See Also

[AxIM3SetStationAccessMode](#), [AxIM3GetStationAccessMode](#), [AxIM3SetAutoSyncConnectMode](#),
[AxIM3GetAutoSyncConnectMode](#), [AxIM3SyncConnectSingle](#), [AxIM3IsOnLine](#)

AxlM3IsOnLine

Purpose

슬레이브 기기와 연결 상태를 확인한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxlM3IsOnLine(long lBoardNo, long lModuleNo, DWORD *dwData);
```

Visual Basic

```
Function AxlM3IsOnLine(ByVal BoardNo As Long, ByVal lModuleNo As Long,
ByRef dwData As Long) As Long
```

Delphi

```
function AxlM3IsOnLine(lBoardNo : LongInt; lModuleNo : LongInt;
dwData : PDWord) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [0Eh] AXT_SMC_R1V04MLIIISV - [0Fh] AXT_SMC_R1V04MLIIIPM - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]dwData | 연결 상태 - [00h] 연결 안 되어 있음 - [01h] 연결되어 있음 |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공

[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러

[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어

* [See error code Table for more information on status error codes](#)

Description

슬레이브 기기와 연결 상태를 확인한다.

Example

C++


```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

DWORD dwData;

// 0번 보드의 0번 서보모션 슬레이브 기기의 연결 상태 설정값을 반환한다.
AxIM3SetStationAccessMode(0, 0, AXT_SMC_R1V04MLIIISV, &dwData);
```

Visual Basic

```
Dim dwData As LONG

'0번 보드의 0번 서보모션 슬레이브 기기의 연결 상태 설정값을 반환한다.
AxIM3SetStationAccessMode 0, 0, AXT_SMC_R1V04MLIIISV, dwData
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

dwData: LongInt

{ 0번 보드의 0번 서보모션 슬레이브 기기의 연결 상태 설정값을 반환한다.}
AxIM3SetStationAccessMode(0, 0, AXT_SMC_R1V04MLIIISV, dwData);
```

See Also

[AxIM3SetStationAccessMode](#), [AxIM3GetStationAccessMode](#), [AxIM3SetAutoSyncConnectMode](#),
[AxIM3GetAutoSyncConnectMode](#), [AxIM3SyncConnectSingle](#), [AxIM3SyncDisconnectSingle](#)

표준 I/O 프로파일 커맨드

| Function | Description |
|------------------------------------|---|
| AxIM3GetStationRWS | 동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정값을 반환한다. |
| AxIM3SetStationRWS | 동기화 상태의 슬레이브 I/O 기기에 대한 데이터값을 설정한다. |
| AxIM3GetStationRWA | 비동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정값을 반환한다. |
| AxIM3SetStationRWA | 비동기화 상태의 슬레이브 I/O 기기에 대한 데이터값을 설정한다. |

AxIM3GetStationRWS

Purpose

동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationRWS(long lBoardNo, long lModuleNo, BYTE
bModuleType, DWORD *pdwParam, BYTE bSize);
```

Visual Basic

```
Function AxIM3GetStationRWS(ByVal lBoardNo As Long, ByVal lModuleNo
As Long, ByVal bModuleType As Byte, ByVal pdwParam As Long, ByVal
bSize As Byte) As Long
```

Delphi

```
function AxIM3GetStationRWS(lBoardNo : LongInt; lModuleNo : LongInt;
bModuleType : Byte; pdwParam : PByte; bSize : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]pdwParam | 데이터 배열 |
| [>>]bSize | 데이터 사이즈(1~15) |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정 값을 반환한다.

Example

C++

```
#include "AXL.h"          // 통합 라이브러리 추가
#include "AXHS.h"

DWORD pdwParam[16];

// 0번 보드의 0번 동기화 상태의 I/O 슬레이브 기기에 대한 데이터 값을 반환한다.
AxIM3GetStationRWS(0, 0, AXT_SIO_RDI32MLIII, pdwParam, 1);
```

Visual Basic

```
Dim pdwParam As Long

'0번 보드의 0번 동기화 상태의 I/O 슬레이브 기기에 대한 데이터 값을 반환한다.
AxIM3GetStationRWS 0, 0, AXT_SIO_RDI32MLIII, pdwParam, 1
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pdwParam : LongInt

{ 0번 보드의 0번 동기화 상태의 I/O 슬레이브 기기에 대한 데이터 값을 반환한다. }
AxIM3GetStationRWS(0, 0, AXT_SIO_RDI32MLIII, pdwParam, 1);
```

See Also

[AxIM3SetStationRWS](#), [AxIM3GetStationRWA](#), [AxIM3SetStationRWA](#)

AxIM3SetStationRWS

Purpose

동기화 상태의 슬레이브 I/O 기기에 대한 데이터 값을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationRWS(long lBoardNo, long lModuleNo, BYTE
bModuleType, DWORD *pdwParam, BYTE bSize);
```

Visual Basic

```
Function AxIM3SetStationRWS(ByVal lBoardNo As Long, ByVal lModuleNo
As Long, ByVal bModuleType As Byte, ByRef pdwParam As Long, ByVal
bSize As Byte) As Long
```

Delphi

```
function AxIM3SetStationRWS(lBoardNo : LongInt; lModuleNo : LongInt;
bModuleType : Byte; pdwParam : PDWord; bSize : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]pdwParam | 데이터 배열 |
| [>>]bSize | 데이터 사이즈(1~15) |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

동기화 상태의 슬레이브 I/O 기기에 대한 데이터 값을 설정한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

DWORD pdwParam[16];

pdwParam [0] = 0x5555AAAA;

// 0번 보드의 0번 동기화 상태의 I/O 슬레이브 기기에 대한 1개의 데이터 값을 설정한다.
AxIM3SetStationRWS(0, 0, AXT_SMC_R1V04MLIISV, pdwParam, 1);
```

Visual Basic

```
Dim pdwParam(14) As Long

pdwParam(0) = &H5555AAAA

'0번 보드의 0번 동기화 상태의 I/O 슬레이브 기기에 대한 1개의 데이터 값을 설정한다.
AxIM3SetStationRWS 0, 0, AXT_SMC_R1V04MLIISV, pdwParam, 1
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pdwParam: array [0..14] of LongInt;

pdwParam[0] := $5555AAAA;

{ 0번 보드의 0번 동기화 상태의 I/O 슬레이브 기기에 대한 1개의 데이터 값을 설정한다. }
AxIM3SetStationRWS(0, 0, AXT_SMC_R1V04MLIISV, pdwParam, 1);
```

See Also

[AxIM3GetStationRWS](#), [AxIM3GetStationRWA](#), [AxIM3SetStationRWA](#)

AxIM3GetStationRWA

Purpose

비동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정 값을 반환한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3GetStationRWA(long lBoardNo, long lModuleNo, BYTE
bModuleType, DWORD *pdwParam, BYTE bSize);
```

Visual Basic

```
Function AxIM3GetStationRWA(ByVal lBoardNo As Long, ByVal lModuleNo
As Long, ByVal bModuleType As Byte, ByVal pdwParam As Long, ByVal
bSize As Byte) As Long
```

Delphi

```
function AxIM3GetStationRWA(lBoardNo : LongInt; lModuleNo : LongInt;
bModuleType : Byte; pdwParam : PByte; bSize : PByte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [<<]pdwParam | 데이터 배열 |
| [>>]bSize | 데이터 사이즈(1~15) |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

비동기화 상태의 슬레이브 I/O 기기에 대한 데이터 설정 값을 반환한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

DWORD pdwParam[16];

// 0번 보드의 0번 비동기화 상태의 I/O 슬레이브 기기에 대한 데이터 값을 반환한다.
AxIM3GetStationRWA(0, 0, AXT_SIO_RDI32MLIII, pdwParam, 1);
```

Visual Basic

```
Dim pdwParam As Long

'0번 보드의 0번 비동기화 상태의 I/O 슬레이브 기기에 대한 데이터 값을 반환한다.
AxIM3GetStationRWA 0, 0, AXT_SIO_RDI32MLIII, pdwParam, 1
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pdwParam : LongInt

{ 0번 보드의 0번 비동기화 상태의 I/O 슬레이브 기기에 대한 데이터 값을 반환한다. }
AxIM3GetStationRWA(0, 0, AXT_SIO_RDI32MLIII, pdwParam, 1);
```

See Also

[AxIM3GetStationRWS](#), [AxIM3SetStationRWS](#), [AxIM3SetStationRWA](#)

AxIM3SetStationRWA

Purpose

비동기화 상태의 슬레이브 I/O 기기에 대한 데이터 값을 설정한다.

주의사항 : MLIII 마스터 보드 전용 함수입니다.

Format

C++

```
DWORD AxIM3SetStationRWA(long lBoardNo, long lModuleNo, BYTE
bModuleType, DWORD *pdwParam, BYTE bSize);
```

Visual Basic

```
Function AxIM3SetStationRWA(ByVal lBoardNo As Long, ByVal lModuleNo
As Long, ByVal bModuleType As Byte, ByRef pdwParam As Long, ByVal
bSize As Byte) As Long
```

Delphi

```
function AxIM3SetStationRWA(lBoardNo : LongInt; lModuleNo : LongInt;
bModuleType : Byte; pdwParam : PDWord; bSize : Byte) : DWord; stdcall;
```

Parameters

| [in/out] Name | [Init Value] Explanation |
|-----------------|--|
| [>>]lBoardNo | 시스템에 장착된 AXL 지원 보드 중 네트워크 마스터 보드에 해당하는 보드 번호 |
| [>>]lModuleNo | 슬레이브 기기 번호(0 ~ (최대슬레이브수 - 1)) |
| [>>]bModuleType | 슬레이브 기기의 종류: AXT_MLIII_SLAVE_MODULE_TYPE - [83h] AXT_SIO_RDI32MLIII - [84h] AXT_SIO_RDO32MLIII - [85h] AXT_SIO_RDB32MLIII - [ABh] AXT_SIO_RAI8MLIII - [ACh] AXT_SIO_RAO4MLIII |
| [>>]pdwParam | 데이터 배열 |
| [>>]bSize | 데이터 사이즈(1~15) |

Return Values

[0000] AXT_RT_SUCCESS : AXL 함수 수행 성공
[1001] AXT_RT_M3_COMMUNICATION_FAILED : AXL MIII 통신 에러
[1054] AXT_RT_NOT_SUPPORT_VERSION : 지원하지 않는 하드웨어
* [See error code Table for more information on status error codes](#)

Description

비동기화 상태의 슬레이브 I/O 기기에 대한 데이터 값을 설정한다.

Example

C++

```
#include "AXL.h"           // 통합 라이브러리 추가
#include "AXHS.h"

DWORD pdwParam[16];

pdwParam [0] = 0x5555AAAA;

// 0번 보드의 0번 비동기화 상태의 I/O 슬레이브 기기에 대한 1개의 데이터 값을 설정한다.
AxIM3SetStationRWA(0, 0, AXT_SMC_R1V04MLIISV, pdwParam, 1);
```

Visual Basic

```
Dim pdwParam(14) As Long

pdwParam(0) = &H5555AAAA

'0번 보드의 0번 비동기화 상태의 I/O 슬레이브 기기에 대한 1개의 데이터 값을 설정한다.
AxIM3SetStationRWA 0, 0, AXT_SMC_R1V04MLIISV, pdwParam, 1
```

Delphi

```
Uses
AXL, AXHS      { 통합 라이브러리 추가 }

pdwParam: array [0..14] of LongInt;

pdwParam[0] := $5555AAAA;

{ 0번 보드의 0번 비동기화 상태의 I/O 슬레이브 기기에 대한 1개의 데이터 값을 설정한다. }
AxIM3SetStationRWA(0, 0, AXT_SMC_R1V04MLIISV, pdwParam, 1);
```

See Also

[AxIM3GetStationRWS](#), [AxIM3SetStationRWS](#), [AxIM3GetStationRWA](#)

에러코드 테이블 (Error Code Table) 확인

라이브러리 사용 중에 발생할 수 있는 에러를 체크하고 처리하려면 에러코드를 바로 반환하면 알 수 있다. 각 함수마다 반환하는 인자가 에러코드인데 에러가 났을 경우 이 함수가 왜 동작이 안되는지 알아보기 위해서 에러코드를 확인하면 된다. 시스템의 안전성이나 프로그램의 디버깅을 용이하게 하기 위해서 AXT_RT_SUCCESS가 나오지 않으면 확인해 보는 게 좋다.

| Error Code | Description |
|-------------------------------------|--|
| [0000] AXT_RT_SUCCESS | API 함수 수행 성공 |
| [1001] AXT_RT_OPEN_ERROR | 라이브러리 오픈 되지않음 |
| [1002] AXT_RT_OPEN_ALREADY | 라이브러리 오픈 되어있고 사용 중임 |
| [1053] AXT_RT_NOT_OPEN | 라이브러리 초기화 실패 |
| [1054] AXT_RT_NOT_SUPPORT_VERSION | 지원하지않는 하드웨어 |
| [1101] AXT_RT_INVALID_BOARD_NO | 유효하지않는 보드 번호 |
| [1102] AXT_RT_INVALID_MODULE_POS | 유효하지않는 모듈 위치 |
| [1103] AXT_RT_INVALID_LEVEL | 유효하지않는 레벨 |
| [1151] AXT_RT_ERROR_VERSION_READ | 라이브러리 버전을 읽을수 없음 |
| [1152] AXT_RT_NETWORK_ERROR | 네트워크에 연결된 슬레이브 노드 연결 되지 않았거나 연결상태가 불량함. |
| [1153] AXT_RT_NETWORK_LOCK_MISMATCH | 저장 된 연결정보와 실제 연결 정보가 맞지 않음. EzConfig 에서 해당 보드의 Lock 정보와 실제 연결상태가 일치 하지 않음. 슬레이브 재구성시 기존의 Lock 정보는 Ezconfig 에서 Unlock 으로 지워줘야 함. |
| [1160] AXT_RT_1ST_BELOW_MIN_VALUE | 첫번째 인자값이 최소값보다 더 작음 |
| [1161] AXT_RT_1ST_ABOVE_MAX_VALUE | 첫번째 인자값이 최대값보다 더 큼 |
| [1170] AXT_RT_2ND_BELOW_MIN_VALUE | 두번째 인자값이 최소값보다 더 작음 |
| [1171] AXT_RT_2ND_ABOVE_MAX_VALUE | 두번째 인자값이 최대값보다 더 큼 |
| [1180] AXT_RT_3RD_BELOW_MIN_VALUE | 세번째 인자값이 최소값보다 더 작음 |
| [1181] AXT_RT_3RD_ABOVE_MAX_VALUE | 세번째 인자값이 최대값보다 더 큼 |
| [1190] AXT_RT_4TH_BELOW_MIN_VALUE | 네번째 인자값이 최소값보다 더 작음 |
| [1191] AXT_RT_4TH_ABOVE_MAX_VALUE | 네번째 인자값이 최대값보다 더 큼 |
| [1200] AXT_RT_5TH_BELOW_MIN_VALUE | 다섯번째 인자값이 최소값보다 더 작음 |
| [1201] AXT_RT_5TH_ABOVE_MAX_VALUE | 다섯번째 인자값이 최대값보다 더 큼 |
| [1210] AXT_RT_6TH_BELOW_MIN_VALUE | 여섯번째 인자값이 최소값보다 더 작음 |
| [1211] AXT_RT_6TH_ABOVE_MAX_VALUE | 여섯번째 인자값이 최대값보다 더 큼 |
| [1220] AXT_RT_7TH_BELOW_MIN_VALUE | 일곱번째 인자값이 최소값보다 더 작음 |
| [1221] AXT_RT_7TH_ABOVE_MAX_VALUE | 일곱번째 인자값이 최대값보다 더 큼 |
| [1230] AXT_RT_8TH_BELOW_MIN_VALUE | 여덟번째 인자값이 최소값보다 더 작음 |
| [1231] AXT_RT_8TH_ABOVE_MAX_VALUE | 여덟번째 인자값이 최대값보다 더 큼 |

| | |
|--|--|
| [1240] AXT_RT_9TH_BELOW_MIN_VALUE | 아홉번째 인자값이 최소값보다 더 작음 |
| [1241] AXT_RT_9TH_ABOVE_MAX_VALUE | 아홉번째 인자값이 최대값보다 더 큼 |
| [1250] AXT_RT_10TH_BELOW_MIN_VALUE | 열번째 인자값이 최소값보다 더 작음 |
| [1251] AXT_RT_10TH_ABOVE_MAX_VALUE | 열번째 인자값이 최대값보다 더 큼 |
| [2001] AXT_RT_AIO_OPEN_ERROR | AIO 모듈 오픈실패 |
| [2051] AXT_RT_AIO_NOT_MODULE | AIO 모듈 없음 |
| [2052] AXT_RT_AIO_NOT_EVENT | AIO 이벤트 읽지 못함 |
| [2101] AXT_RT_AIO_INVALID_MODULE_NO | 유효하지않은 AIO 모듈 |
| [2102] AXT_RT_AIO_INVALID_CHANNEL_NO | 유효하지않은 AIO 채널번호 |
| [2106] AXT_RT_AIO_INVALID_USE | AIO 함수 사용못함 |
| [2107] AXT_RT_AIO_INVALID_TRIGGER_MODE | 유효하지않은 트리거 모드 |
| [3001] AXT_RT_DIO_OPEN_ERROR | DIO 모듈 오픈실패 |
| [3051] AXT_RT_DIO_NOT_MODULE | DIO 모듈 없음 |
| [3052] AXT_RT_DIO_NOT_INTERRUPT | DIO 인터럽트 설정안됨 |
| [3101] AXT_RT_DIO_INVALID_MODULE_NO | 유효하지않은 DIO 모듈 번호 |
| [3102] AXT_RT_DIO_INVALID_OFFSET_NO | 유효하지않은 DIO OFFSET 번호 |
| [3103] AXT_RT_DIO_INVALID_LEVEL | 유효하지않은 DIO 레벨 |
| [3104] AXT_RT_DIO_INVALID_MODE | 유효하지않은 DIO 모드 |
| [3105] AXT_RT_DIO_INVALID_VALUE | 유효하지않은 값 설정 |
| [3106] AXT_RT_DIO_INVALID_USE | DIO 함수 사용못함 |
| [4001] AXT_RT_MOTION_OPEN_ERROR | 모션 라이브러리 Open 실패 |
| [4051] AXT_RT_MOTION_NOT_MODULE | 시스템에 장착된 모션 모듈이 없음 |
| [4052] AXT_RT_MOTION_NOT_INTERRUPT | 인터럽트 결과 읽기 실패 |
| [4053] AXT_RT_MOTION_NOT_INITIAL_AXIS_NO | 해당 축 모션 초기화 실패 |
| [4054] AXT_RT_MOTION_NOT_IN_CONT_INTERPOL | 연속 보간 구동 중이 아닌 상태에서 연속보간 중지 명령을 수행 하였음 |
| [4055] AXT_RT_MOTION_NOT_PARA_READ | 원점 구동 설정 파라미터 로드 실패 |
| [4101] AXT_RT_MOTION_INVALID_AXIS_NO | 해당 축이 존재하지 않음 |
| [4102] AXT_RT_MOTION_INVALID_METHOD | 해당 축 구동에 필요한 설정이 잘못됨 |
| [4103] AXT_RT_MOTION_INVALID_USE | 'uUse' 인자값이 잘못 설정됨 |
| [4104] AXT_RT_MOTION_INVALID_LEVEL | 'uLevel' 인자값이 잘못 설정됨 |
| [4105] AXT_RT_MOTION_INVALID_BIT_NO | 범용 입출력 해당 비트가 잘못 설정됨 |
| [4106] AXT_RT_MOTION_INVALID_STOP_MODE | 모션 정지 모드 설정값이 잘못됨 |
| [4107] AXT_RT_MOTION_INVALID_TRIGGER_MODE | 트리거 설정 모드가 잘못 설정됨 |
| [4108] AXT_RT_MOTION_INVALID_TRIGGER_LEVEL | 트리거 출력 레벨 설정이 잘못됨 |
| [4109] AXT_RT_MOTION_INVALID_SELECTION | 'uSelection' 인자가 COMMAND 또는 ACTUAL 이외의 값으로 설정되어 있음 |

| | |
|---|---|
| [4110] AXT_RT_MOTION_INVALID_TIME | Trigger 출력 시간값이 잘못 설정되어 있음 |
| [4111] AXT_RT_MOTION_INVALID_FILE_LOAD | 모션 설정값이 저장된 파일이 로드가 안됨 |
| [4112] AXT_RT_MOTION_INVALID_FILE_SAVE | 모션 설정값을 저장하는 파일 저장에 실패함 |
| [4113] AXT_RT_MOTION_INVALID_VELOCITY | 모션 구동 속도값이 0 으로 설정되어 모션 에러 발생 |
| [4114] AXT_RT_MOTION_INVALID_ACCELTIME | 모션 구동 가속 시간값이 0 으로 설정되어 모션 에러 발생 |
| [4115] AXT_RT_MOTION_INVALID_PULSE_VALUE | 모션 단위 설정 시 입력 펄스값이 0 보다 작은값으로 설정됨 |
| [4116] AXT_RT_MOTION_INVALID_NODE_NUMBER | 위치나 속도 오버라이드 함수가 모션 정지 중에 실행됨 |
| [4117] AXT_RT_MOTION_INVALID_TARGET | 다축 모션 정지 원인에 관한 플래그를 반환한다. |
| [4151] AXT_RT_MOTION_ERROR_IN_NONMOTION | 모션 구동중이어야 되는데 모션 구동중이 아닐 때 |
| [4152] AXT_RT_MOTION_ERROR_IN_MOTION | 모션 구동 중에 다른 모션 구동 함수를 실행함 |
| [4153] AXT_RT_MOTION_ERROR | 다축 구동 정지 함수 실행 중 에러 발생함 |
| [4154] AXT_RT_MOTION_ERROR_GANTRY_ENABLE | 겐트리 enable 이 되어있어 모션종일 때 또 겐트리 enable 을 눌렀을 때 |
| [4155] AXT_RT_MOTION_ERROR_GANTRY_AXIS | 겐트리 축이 마스터채널 |
| [4156] AXT_RT_MOTION_ERROR_MASTER_SERVOON | 마스터 축 서보온이 안되어있을 때 |
| [4157] AXT_RT_MOTION_ERROR_SLAVE_SERVOON | 슬레이브 축 서보온이 안되어있을 때 |
| [4158] AXT_RT_MOTION_INVALID_POSITION | 유효한 위치에 없을 때 |
| [4159] AXT_RT_ERROR_NOT_SAME_MODULE | 똑 같은 모듈내에 있지 않을경우 |
| [4160] AXT_RT_ERROR_NOT_SAME_BOARD | 똑 같은 보드내에 있지 아닐경우 |
| [4161] AXT_RT_ERROR_NOT_SAME_PRODUCT | 제품이 서로 다를경우 |
| [4162] AXT_RT_NOT_CAPTURED | 위치가 저장되지 않을 때 |
| [4163] AXT_RT_ERROR_NOT_SAME_IC | 같은 칩내에 존재하지않을 때 |
| [4164] AXT_RT_ERROR_NOT_GEARMODE | 기어모드로 변환이 안될 때 |
| [4165] AXT_ERROR_CONTI_INVALID_AXIS_NO | 연속보간 축맵핑 시 유효한 축이 아닐 때 |
| [4166] AXT_ERROR_CONTI_INVALID_MAP_NO | 연속보간 맵핑 시 유효한 맵핑 번호가 아닐 때 |
| [4167] AXT_ERROR_CONTI_EMPTY_MAP_NO | 연속보간 맵핑 번호가 비워 있을 때 |
| [4168] AXT_RT_MOTION_ERROR_CACULATION | 계산상의 오차가 발생했을 때 |
| [4169] AXT_RT_ERROR_MOVE_SENSOR_CHECK | 연속보간 구동전 예러센서가(Alarm, EMG, Limit 등) 감지된 경우 |
| [4170] AXT_ERROR_HELICAL_INVALID_AXIS_NO | 헬리컬 축 맵핑 시 유효한 축이 아닐 때 |
| [4171] AXT_ERROR_HELICAL_INVALID_MAP_NO | 헬리컬 맵핑 시 유효한 맵핑 번호가 아닐 때 |
| [4172] AXT_ERROR_HELICAL_EMPTY_MAP_NO | 헬리컬 맵핑 번호가 비워 있을 때 |
| [4180] AXT_ERROR_SPLINE_INVALID_AXIS_NO | 스플라인 축 맵핑 시 유효한 축이 아닐 때 |
| [4181] AXT_ERROR_SPLINE_INVALID_MAP_NO | 스플라인 맵핑 시 유효한 맵핑 번호가 아닐 때 |
| [4182] AXT_ERROR_SPLINE_EMPTY_MAP_NO | 스플라인 맵핑 번호가 비워있을 때 |
| [4183] AXT_ERROR_SPLINE_NUM_ERROR | 스플라인 점숫자가 부적당할 때 |

| | |
|--|--|
| [4184] AXT_RT_MOTION_INTERPOL_VALUE | 보간할 때 입력 값이 잘못넣어졌을 때 |
| [4185] AXT_RT_ERROR_NOT_CONTIBEGIN | 연속보간 할 때 CONTIBEGIN 함수를 호출하지 않을 때 |
| [4186] AXT_RT_ERROR_NOT_CONTIEND | 연속보간 할 때 CONTIEND 함수를 호출하지 않을 때 |
| [4201] AXT_RT_MOTION_HOME_SEARCHING | 홈을 찾고 있는 중일 때 다른 모션 함수들을 사용할 때 |
| [4202] AXT_RT_MOTION_HOME_ERROR_SEARCHING | 홈을 찾고 있는 중일 때 외부에서 사용자나 혹은 어떤것에 의한 강제로 정지당할 때 |
| [4203] AXT_RT_MOTION_HOME_ERROR_START | 초기화 문제로 홈시작 불가할 때 |
| [4204] AXT_RT_MOTION_HOME_ERROR_GANTRY | 홈을 찾고 있는 중일 때 겐트리 enable 불가할 때 |
| [4251] AXT_RT_MOTION_POS_OUTOFBOUND | 설정한 위치값이 설정 최대값보다 크거나 최소값보다 작은값임 |
| [4252] AXT_RT_MOTION_PROFILE_INVALID | 구동 속도 프로파일 설정이 잘못됨 |
| [4253] AXT_RT_MOTION_VELOCITY_OUTOFBOUND | 구동 속도값이 최대값보다 크게 설정됨 |
| [4254] AXT_RT_MOTION_MOVE_UNIT_IS_ZERO | 구동 단위값이 0 으로 설정됨 |
| [4255] AXT_RT_MOTION_SETTING_ERROR | 속도, 가속도, 저크, 프로파일 설정이 잘못됨 |
| [4256] AXT_RT_MOTION_IN_CONT_INTERPOL | 연속 보간 구동 중 구동 시작 또는 재시작 함수를 실행하였음 |
| [4257] AXT_RT_MOTION_DISABLE_TRIGGER | 트리거 출력이 Disable 상태임 |
| [4258] AXT_RT_MOTION_INVALID_CONT_INDEX | 연속 보간 Index 값 설정이 잘못됨 |
| [4259] AXT_RT_MOTION_CONT_QUEUE_FULL | 모션 침의 연속 보간 큐가 Full 상태임 |
| [4260] AXT_RT_PROTECTED_DURING_SERVOON | 서보 온 상태에서 사용할 수 없는 함수 또는 입력값을 사용하였음. |
| [4261] AXT_RT_HW_ACCESS_ERROR | 보드의 H/W 접근 오류가 발생함. 시스템내의 보드 장착 상태를 확인하거나 전원 입력 상태를 확인하여 조치함. |
| [4262] AXT_RT_HW_DPRAM_CMD_WRITE_ERROR_LV1 | DPRAM Comamnd Write 실패 Level1 |
| [4263] AXT_RT_HW_DPRAM_CMD_WRITE_ERROR_LV2 | DPRAM Comamnd Write 실패 Level2 |
| [4264] AXT_RT_HW_DPRAM_CMD_WRITE_ERROR_LV3 | DPRAM Comamnd Write 실패 Level3 |
| [4265] AXT_RT_HW_DPRAM_CMD_READ_ERROR_LV1 | DPRAM Comamnd Read 실패 Level1 |
| [4266] AXT_RT_HW_DPRAM_CMD_READ_ERROR_LV2 | DPRAM Comamnd Read 실패 Level2 |
| [4267] AXT_RT_HW_DPRAM_CMD_READ_ERROR_LV3 | DPRAM Comamnd Read 실패 Level3 |
| [4300] AXT_RT_COMPENSATION_SET_PARAM_FIRST | 보정 파라미터 중 첫번째 값이 잘못 설정되었음 |
| [4400] AXT_RT_SEQ_NOT_IN_SERVICE | 순차 구동 함수 실행 중 자원 할당 실패 |
| [4401] AXT_ERROR_SEQ_INVALID_MAP_NO | 순차 구동 함수 실행 중 맵핑 번호 이상. |
| [4402] AXT_ERROR_INVALID_AXIS_NO | 함수 설정 인자중 축번호 이상. |
| [4403] AXT_RT_ERROR_NOT_SEQ_NODE_BEGIN | 순차 구동 노드 입력 시작 함수를 호출하지 않음. |
| [4404] AXT_RT_ERROR_NOT_SEQ_NODE_END | 순차 구동 노드 입력 종료 함수를 호출하지 않음. |
| [4405] AXT_RT_ERROR_NO_NODE | 순차 구동 노드 입력이 없음. |
| [4406] AXT_RT_ERROR_SEQ_STOP_TIMEOUT | |

| | |
|--|--|
| [4500] AXT_RT_M3_COMMUNICATION_FAILED | MLIII 통신 기준, 통신 실패 |
| [4501] AXT_RT_MOTION_ONE_OF_AXES_IS_NOT_M3 | MLIII 통신 기준, 구성된 ML3 노드 중에서 모션 노드 없음 |
| [4502] AXT_RT_MOTION_BIGGER_VEL_THEN_MAX_VEL | MLIII 통신 기준, 지정된 축의 설정된 최대 속도보다 큼 |
| [4503] AXT_RT_MOTION_SMALLER_VEL_THEN_MAX_VEL | MLIII 통신 기준, 지정된 축의 설정된 최대 속도보다 작음 |
| [4504] AXT_RT_MOTION_ACCEL_MUST_BIGGER_THEN_ZERO | MLIII 통신 기준, 지정된 축의 설정된 가속도가 0 보다 큼 |
| [4505] AXT_RT_MOTION_SMALL_ACCEL_WITH_UNIT_PULSE | MLIII 통신 기준, UnitPulse 가 적용된 가속도가 0 보다 큼 |
| [4506] AXT_RT_MOTION_INVALID_INPUT_ACCEL | MLIII 통신 기준, 지정된 축의 가속도 입력이 잘못됨 |
| [4507] AXT_RT_MOTION_SMALL_DECEL_WITH_UNIT_PULSE | MLIII 통신 기준, UnitPulse 가 적용된 감속도가 0 보다 큼 |
| [4508] AXT_RT_MOTION_INVALID_INPUT_DECEL | MLIII 통신 기준, 지정된 축의 감속도 입력이 잘못됨 |
| [4509] AXT_RT_MOTION_SAME_START_AND_CENTER_POS | MLIII 통신 기준, 원호보간의 시작점과 중심점이 같음 |
| [4510] AXT_RT_MOTION_INVALID_JERK | MLIII 통신 기준, 지정된 축의 저크 입력이 잘못됨 |
| [4511] AXT_RT_MOTION_INVALID_INPUT_VALUE | MLIII 통신 기준, 지정된 축의 입력값이 잘못됨 |
| [4512] AXT_RT_MOTION_NOT_SUPPORT_PROFILE | MLIII 통신 기준, 제공되지 않는 속도 프로파일임 |
| [4513] AXT_RT_MOTION_INPOS_UNUSED | MLIII 통신 기준, 인포지션 사용하지 않음 |
| [4514] AXT_RT_MOTION_AXIS_IN_SLAVE_STATE | MLIII 통신 기준, 지정된 축이 슬레이브 상태가 아님 |
| [4515] AXT_RT_MOTION_AXES_ARE_NOT_SAME_BOARD | MLIII 통신 기준, 지정된 축들이 같은 보드 내에 있지 않음 |
| [4516] AXT_RT_MOTION_ERROR_IN_ALARM | MLIII 통신 기준, 지정된 축이 알람 상태임 |
| [4517] AXT_RT_MOTION_ERROR_IN_EMGN | MLIII 통신 기준, 지정된 축이 비상정지 상태임 |
| [4518] AXT_RT_MOTION_CAN_NOT_CHANGE_COORD_NO | MLIII 통신 기준, 코디네이터 넘버 변환 불가임 |
| [4519] AXT_RT_MOTION_INVALID_INTERNAL_RADIOUS | MLIII 통신 기준, 원호보간의 X, Y 축 반지름 불일치 |
| [4521] AXT_RT_MOTION_CONTI_QUEUE_FULL | MLIII 통신 기준, 보간의 큐가 가득 참 |
| [4522] AXT_RT_MOTION_SAME_START_AND_END_POSITION | MLIII 통신 기준, 원호보간의 시작점과 종료점이 같음 |
| [4523] AXT_RT_MOTION_INVALID_ANGLE | MLIII 통신 기준, 원호보간의 각도가 360 도 초과됨 |
| [4524] AXT_RT_MOTION_CONTI_QUEUE_EMPTY | MLIII 통신 기준, 보간의 큐가 비어있음 |
| [4525] AXT_RT_MOTION_ERROR_GEAR_ENABLE | MLIII 통신 기준, 지정된 축이 이미 링크 설정 상태임 |
| [4526] AXT_RT_MOTION_ERROR_GEAR_AXIS | MLIII 통신 기준, 지정된 축이 링크축이 아님 |
| [4527] AXT_RT_MOTION_ERROR_NO_GANTRY_ENABLE | MLIII 통신 기준, 지정된 축이 겐트리 설정 상태가 아님 |
| [4528] AXT_RT_MOTION_ERROR_NO_GEAR_ENABLE | MLIII 통신 기준, 지정된 축이 링크 설정 상태가 아님 |
| [4529] AXT_RT_MOTION_ERROR_GANTRY_ENABLE_FULL | MLIII 통신 기준, 겐트리 설정 가득참 |
| [4530] AXT_RT_MOTION_ERROR_GEAR_ENABLE_FULL | MLIII 통신 기준, 링크 설정 가득참 |
| [4531] AXT_RT_MOTION_ERROR_NO_GANTRY_SLAVE | MLIII 통신 기준, 지정된 축이 겐트리 슬레이브 설정상태가 아님 |
| [4532] AXT_RT_MOTION_ERROR_NO_GEAR_SLAVE | MLIII 통신 기준, 지정된 축이 링크 슬레이브 설정상태가 아님 |
| [4533] AXT_RT_MOTION_ERROR_MASTER_SLAVE_SAME | MLIII 통신 기준, 마스터축과 슬레이브 축이 동일함 |
| [4534] AXT_RT_MOTION_NOT_SUPPORT_HOMESIGNAL | MLIII 통신 기준, 지정된 축의 홈신호는 지원되지 않음 |
| [4535] AXT_RT_MOTION_ERROR_NOT_SYNC_CONNECT | MLIII 통신 기준, 지정된 축이 싱크 연결 상태가 아님 |

| | |
|---|--|
| [4536] AXT_RT_MOTION_OVERFLOW_POSITION | MLIII 통신 기준, 지정된 축에 대한 구동 위치값이 오버플로우임 |
| [4537] AXT_RT_MOTION_ERROR_INVALID_CONTIMAPAXIS | MLIII 통신 기준, 보간작업을 위한 지정된 좌표계 축매핑이 없음 |
| [4538] AXT_RT_MOTION_ERROR_INVALID_CONTIMAPSIZE | MLIII 통신 기준, 보간작업을 위한 지정된 좌표계 축매핑 축사이즈가 잘못됨 |
| [4539] AXT_RT_MOTION_ERROR_IN_SERVO_OFF | MLIII 통신 기준, 지정된 축이 서보 OFF 되어 있음 |
| [4540] AXT_RT_MOTION_ERROR_POSITIVE_LIMIT | MLIII 통신 기준, 지정된 축이 (+)리미트 ON 되어 있음 |
| [4541] AXT_RT_MOTION_ERROR_NEGATIVE_LIMIT | MLIII 통신 기준, 지정된 축이 (-)리미트 ON 되어 있음 |
| [4542] AXT_RT_MOTION_ERROR_OVERFLOW_SWPROFILE_NUM | MLIII 통신 기준, 지정된 축들에 대한 지원 프로파일 개수가 오버플로우됨 |
| [4543] AXT_RT_PROTECTED_DURING_INMOTION | in_motion 되어 있는 상태에서 사용 못 함 |
| [5000] AXT_RT_DATA_FLASH_NOT_EXIST | |
| [5001] AXT_RT_DATA_FLASH_BUSY | |