

Library User Manual

AXL library family
AXLNET



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 theresponsibility 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

Contents

1. AXLNet 개요	6
1.1. AXLNet 개요	6
1.2. AXLNet 라이브러리 사용을 위한 환경 설정	7
Ethernet/IP 통신 관련 설정	7
Ethernet/IP 개요	7
Ethernet/IP 사용 환경 설정	8
1.3. AXLNET 라이브러리 구성 정보	10
2. AXLNet 사용하기	11
2.1. Visual C++ 라이브러리 링크하기	11
2.2. DLL 파일 및 Back-Ground process 복사하기	13
3. 라이브러리 초기화	14
3.1. Application Flow Chart	14
3.2. 함수 초기화 예제	15
4. Visual C++과 DIO 모듈을 이용한 초기화 예제	18
4.1. Win32 Console용 DIO 시작	18
4.2. Flow Chart 및 Source	21
4.3. 에러 조치	25
Error executing 발생	25
Cannot open include file 발생	25
fatal error LNK1120 발생	25
DLL 찾기 실패 발생	26
기타 에러 발생	26
5. Visual C++과 Motion 모듈을 이용한 초기화 예제	27
5.1. Win32 Console용 Motion 시작	27
5.2. Flow Chart 및 Source	30
5.3. 에러 조치	35
Error executing 발생	36
Cannot open include file 발생	36
fatal error LNK1120 발생	36
DLL 찾기 실패 발생	37
기타 에러 발생	37

6. Base Command Function List	38
6.1. 라이브러리 초기화	39
AxIOpen	40
AxIInit	41
AxIClose	42
AxIIsOpened	43
6.2. 라이브러리 및 베이스 보드 정보	44
AxIGetNodeCount	45
AxIGetNodeInfo	47
AxIGetNetInfo	48
AxIGetLibVersion	50
6.3. 로그 레벨	52
AxISetLogLevel	53
AxIGetLogLevel	55
6.4. 네트워크 상태 관련 함수	57
AxINetStatusRead	58
AxIGetNodeFirmVersion	60
AxISaveParamToFlash	61
AxILoadParamFromFlash	63
AxIResetParamFlash	65
에러코드 테이블(Error Code Table) 확인	67

Revision History

Manual	VERSION	Comments
Rev. 1.0 issue 1.0	Rev. 1.0	2008.07.23

1. AXLNet 개요

1.1. AXLNet 개요

아진엑스텍의 분산제어형 제품은 여러 종류의 통신 연결 모듈에 모션모듈, 디지털 I/O 모듈, 아날로그 I/O 모듈등을 어떻게 조합하는가에 따라 다양한 기능을 포함하는 형태로 구성할 수 있다. 분산형 제품에는 통신 방식에 따라 Ethernet/IP 용 제품군, RTEXT 용 제품군(미출시), EtherCat 제품군(미출시)으로 나뉘어 진다. 현재 AXLNet 라이브러리에서는 VC++ 개발 환경에서 Ethernet/IP 제품군만 지원한다.

분산제어용 제품군의 통신제어 및 모듈을 제어하는 프로그램 라이브러리 API(Application Programming Interface)를 통틀어 AXLNet(AjineXtek Library for Network resources)이라고 한다.

AXLNet 라이브러리는 기능별로 초기화 및 네트워크 상태, 제어기 구성 정보를 확인하는 AXL, Motion 에 관련된 AXM, Digital I/O 와 관련된 AXD, Analog I/O 와 관련된 AXA 에 대한 라이브러리로 나뉘어져 있다. 라이브러리 헤더파일을 기준으로 분류하면 다음과 같다.

AXL (Base Command)

AXL 은 네트워크에 연결된 EIP-CPU 모듈들을 자동으로 인식하고 초기화, 인터페이스 하기 위한 기초 라이브러리이다.

AXM (Motion Command)

AXM 은 모션제어 관련 라이브러리로서, 칩에서 제공하는 직선 및 원호 보간 기능을 포함한 상위 고급 모션 함수들을 제공하며, 프로그래밍에 좀더 편리한 인터페이스를 제공한다. AXM 은 2 축 전용 모션 제어 칩(CAMC-QI 이상)이 장착된 EIP-M2Q 버전 이상의 하드웨어를 지원한다.

AXD (DIO Command)

AXD 는 DIO 모듈 및 보드를 통하여 디지털 신호의 입출력을 제어하는 라이브러리이다.

AXA (AIO Command)

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

1.2. AXLNet 라이브러리 사용을 위한 환경 설정.

Ethernet/IP 통신 관련 설정

Ethernet/IP 개요

완전한 표준 Ethernet 을 사용하여 Application 계층에 CIP(Common Industrial Protocol)을 결합한 형태의 분산제어 통신 방식이다. Ethernet/IP 방식이 사용하는 통신 프로토콜은 모두 표준화되어 있는 기술이기 때문에 특정 하드웨어 및 소프트웨어에 제약을 받지 않으며, 기존 Ethernet 설치 기반의 네트워크 H/W 로 모든 기능을 구현할 수 있다. 제어 환경 요건으로는 Ethernet 을 사용하기 위한 기본 시스템 환경 수준이 필요한데 다음과 같은 최소 시스템 사양 이상의 제어 시스템이 필요하다.

시스템 : 일반적으로 IBM 호환 기종 컴퓨터와 산업용 컴퓨터인 IPC, cPCI 에서 운용할 수 있습니다.

CPU : Pentium II 450MHz 이상에서 사용 가능하지만 권장 사양은 Pentium III 850MHz 이상이면 안정적으로 작동됩니다.

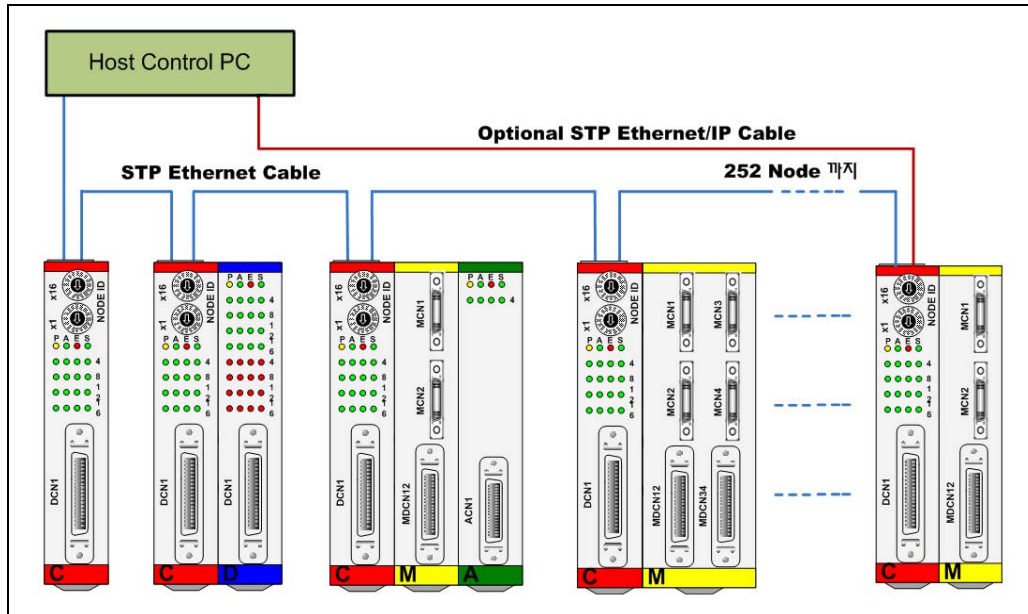
Memory : 최소 128MB 이상이면 사용 가능하고 권장 사양은 256MB 이상입니다.

OS : Microsoft 사 계열인 Windows98 이상이면 운용 가능하지만 시스템 안정성을 위해서는 Windows 2000 이상을 권장합니다. (Windows NT 경우는 별도 문의 바랍니다.)

LanAdapter : 통신 속도 100BaseTX(Full duplex) 이상 성능을 권장합니다.

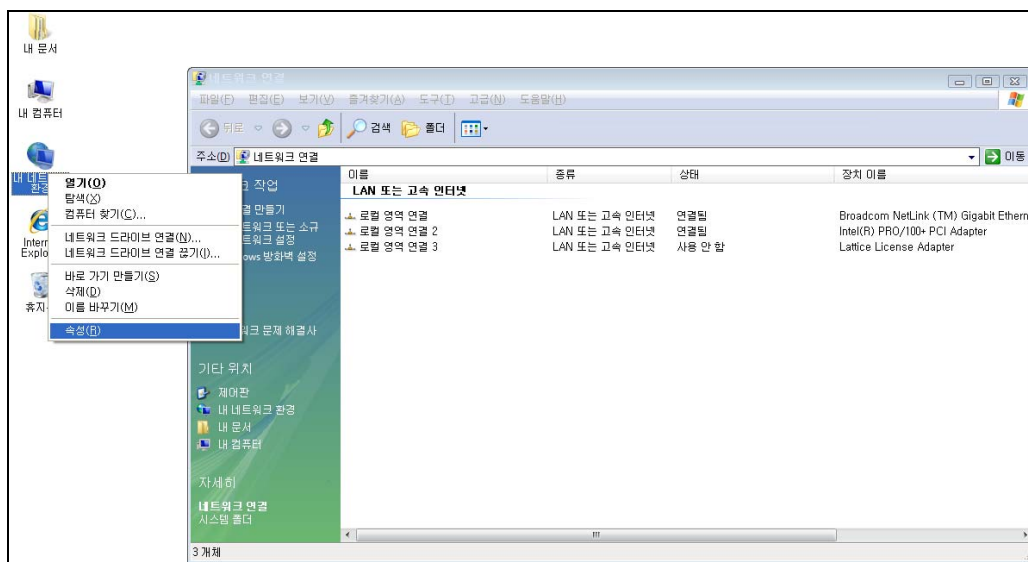
Ethernet/IP 사용 환경 설정.

아래의 그림과 같이 제어 시스템을 구성하고 연결하기 위해서는 Lan adapter 의 TCP/IP 설정등의 과정이 필요하다.



1

제어 시스템에 장착된 Lan adapter 가 장치관리자에 정상적으로 등록되었는지 확인한다. 정상적으로 등록되었다면 시스템내 네트워크 환경 등록정보로 현재 장착된 Lan adapter 의 상태를 확인할 수 있다.

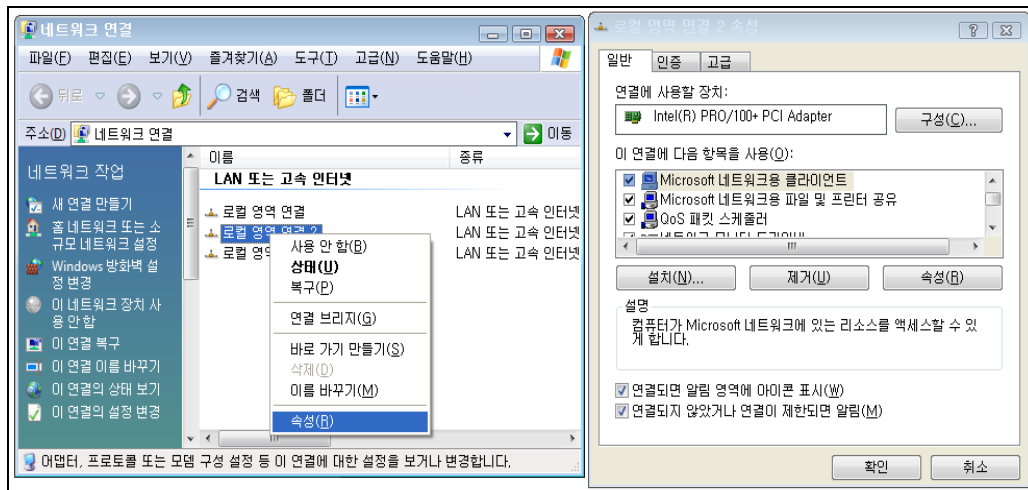


2

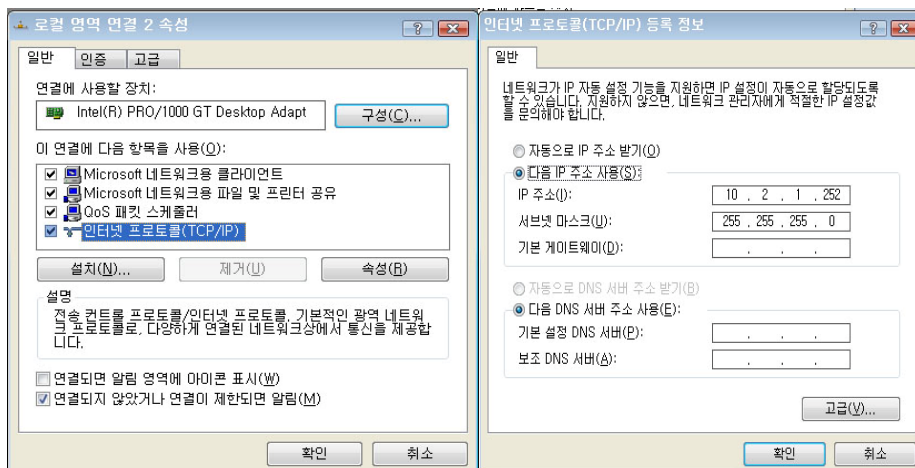
Ethernet/IP 통신을 위한 사용 예정 Lan adapter 의 IP 를 설정하여야 한다. 아진 엑스텍의 Ethernet/IP 기능 모듈의 IP 사용 범위는 제어 시스템과 하위 제어기 모두가 같은 범위인 10.2.1.2~10.2.1.254 이다. 즉 제어 시스템과 하위 제어기들의 IP 는 10.2.1.2~10.2.1.254 범위에서 하나로 결정하여야 한다. 제어 시스템의 IP 는 사용자가 직접 정하는데 일반적으로 10.2.1.250~10.2.1.254 범위로 사용하는 것이 좋다. 하위 제어기의 IP 은 NODE ID 설정 값으로 자동으로

IP 가 할당되기 때문에 NODE ID 설정을 중복되지 않게 설정하면 별도로 IP 설정은 불필요하다. 이때 상위 제어 시스템과 IP 충돌을 방지하기 위하여 NODE ID 를 0x00 ~ 0xF7 까지 사용할 것을 권장한다. 다음의 그림과 같이 제어 시스템의 IP 를 설정한다.

먼저 해당 Lan adapter 를 선택하고 오른쪽 마우스 클릭후 속성을 클릭한다.



연결 속성의 항목중 인터넷 프로토콜(TCP/IP) 를 선택하여 속성보기를 클릭한다. 아래의 과 같이 IP 주소는 10.2.1.250 ~ 10.2.1.254 중 하나로 설정하고 서브넷 마스크를 255.255.255.0 으로 설정한다. 나머지 항목은 비워둔다. 만일 Ring topology 를 위해 2 개의 Lan adapter 를 사용하였다면 첫번째로 설정한 Lan adapter 와 IP 를 다르게 하고 다른 부분은 동일하게 설정하면 된다.



3 제어 시스템의 IP 설정을 완료 하였다면 하위 제어기의 NODE ID 를 중복되지 않도록 설정하여야 한다. 이때 상위 제어 시스템과 IP 충돌을 방지하기 위하여 NODE ID 를 0x00 ~ 0xF7 까지 사용할 것을 권장한다.

1.3. AXLNET 라이브러리 구성 정보

각 제품에 대한 실행 함수 외형 및 프로그램에 필요한 정의 값들을 기술한 라이브러리 헤더 파일 및 라이브러리 링커에 필요한 .LIB 파일 그리고 실행에 필요한 .DLL 파일로 구성되어 있다.

다음은 AXLNet 라이브러리 구성 파일의 상세 설명이다.(각 개발 언어별 파일 이름이 다를 수 있습니다. 다음 표는 VC++ 개발 언어 기준입니다.)

파일명	해당 제품	설명
AXL.h	모든 Ethernet/IP 제품	Node 단위 제어에 관한 함수
AXHS.h	모든 Ethernet/IP 제품	AXL Define 함수
AXA.h	Analog 입출력 모듈	아날로그 입출력 기능 모듈과 관련된 기능 함수
AXM.h	Motion 모듈	모션 제어 기능 모듈과 관련된 기능 함수
AXD.h	Digital 입출력 모듈	디지털 입출력 기능 모듈과 관련된 기능 함수
AXLNet.lib	모든 Ethernet/IP 제품	AXLNet 라이브러리 VC++ link 라이브러리
AXLNet.dll	모든 Ethernet/IP 제품	AXLNet 실행 라이브러리
EzBasicAxlNet.dll	모든 Ethernet/IP 제품	Network 관련 제어 실행 라이브러리
EtIPScanner.dll	모든 Ethernet/IP 제품	Ethernet/IP 통신 관련 실행 라이브러리
EipBGProcess.exe	모든 Ethernet/IP 제품	EtherNet/IP 라이브러리를 사용하기 위한 BackGround 실행 프로그램.

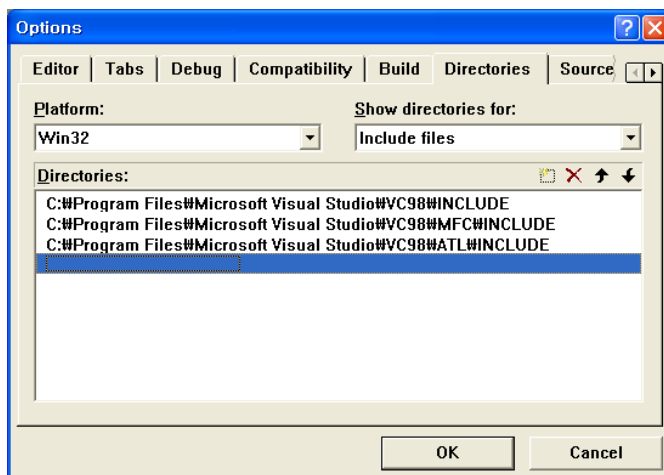
2. AXLNet 사용하기

각 언어별로 라이브러리를 링크하여 사용하면 새로운 프로젝트를 시작할 때마다 따로 지정할 필요가 없습니다. 사용자께서는 사용할 해당 언어만 참조하시면 됩니다.

2.1. Visual C++ 라이브러리 링크하기

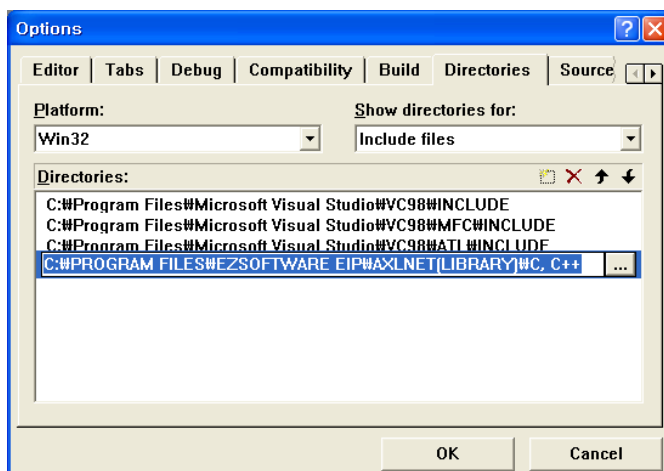
1

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



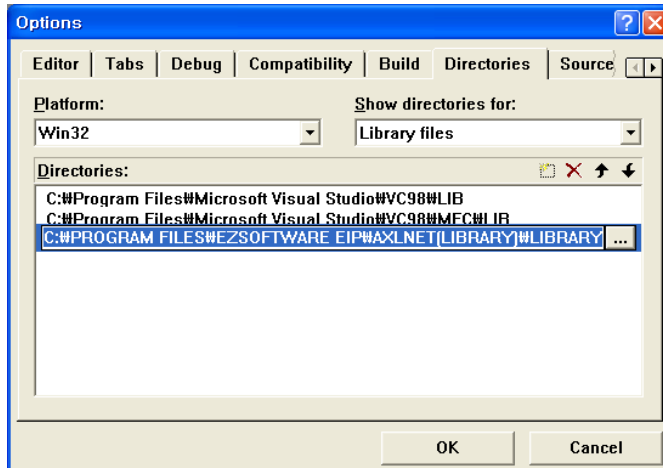
2

Header 파일 지정은 Show directories for: 콤보에 Include files 를 선택한 다음 Directories: 에 마지막 빈 라인을 클릭하면 오른쪽에 ... 버튼이 나타나는데 이 버튼을 클릭하면 Choose Directory 다이얼로그가 나타납니다. 이때 “라이브러리 제공 디렉토리”\WAXLNet(Library)\WInclude 폴더를 지정합니다.

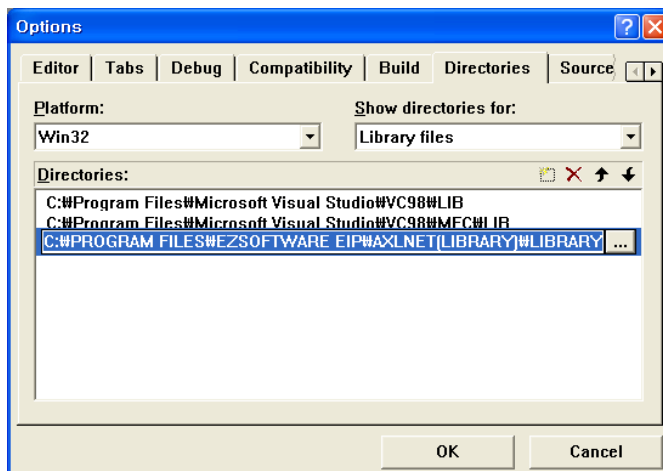


3 Library 파일 지정은 Show directories for: 콤보에 Library files 선택한 다음 Directories: 에 마지막 빈 라인을 클릭하면 오른쪽에 ...버튼이 나타나는데 이 버튼을 클릭하면 Choose Directory 다이얼로그가 나타납니다.

이때 “라이브러리 제공 디렉토리”WAXLNet(Library)WLibrary 폴더를 지정합니다.



4 Header 파일과 Library 파일 지정을 완료 했으면 OK 버튼을 클릭합니다.



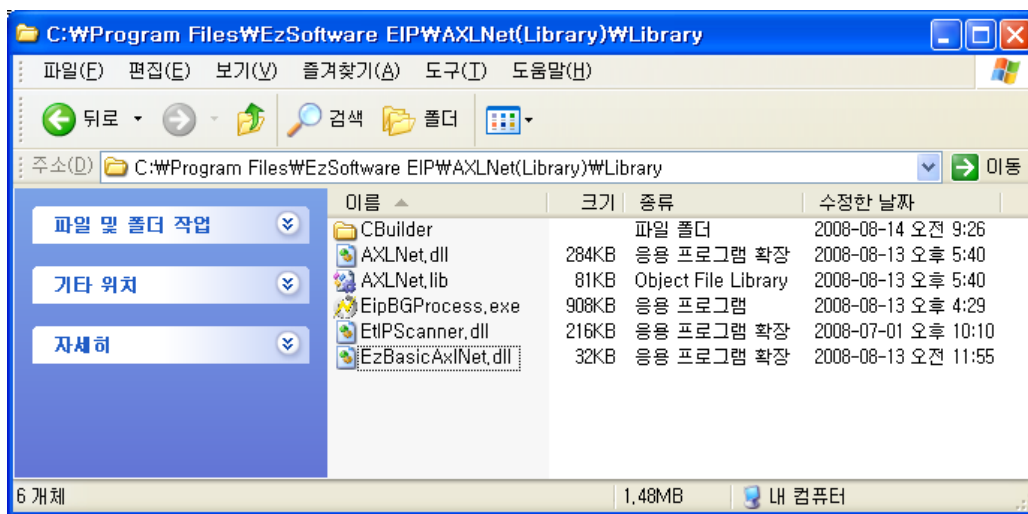
2.2. DLL 파일 및 Back-Ground process 복사하기

1

해당 언어 라이브러리 링크 작업을 완료 하였다면 프로젝트 컴파일 생성되는 실행파일이 있는 폴더에 AXLNet.dll, EzBasicAxINet.dll 두 개의 파일과 EzSoftware EIP 를 설치하지 않은 시스템에서 실행하기 위해서는 EtlPScanner.dll 과 EipBGProcess.exe 두개의 파일을 복사하여야 합니다.

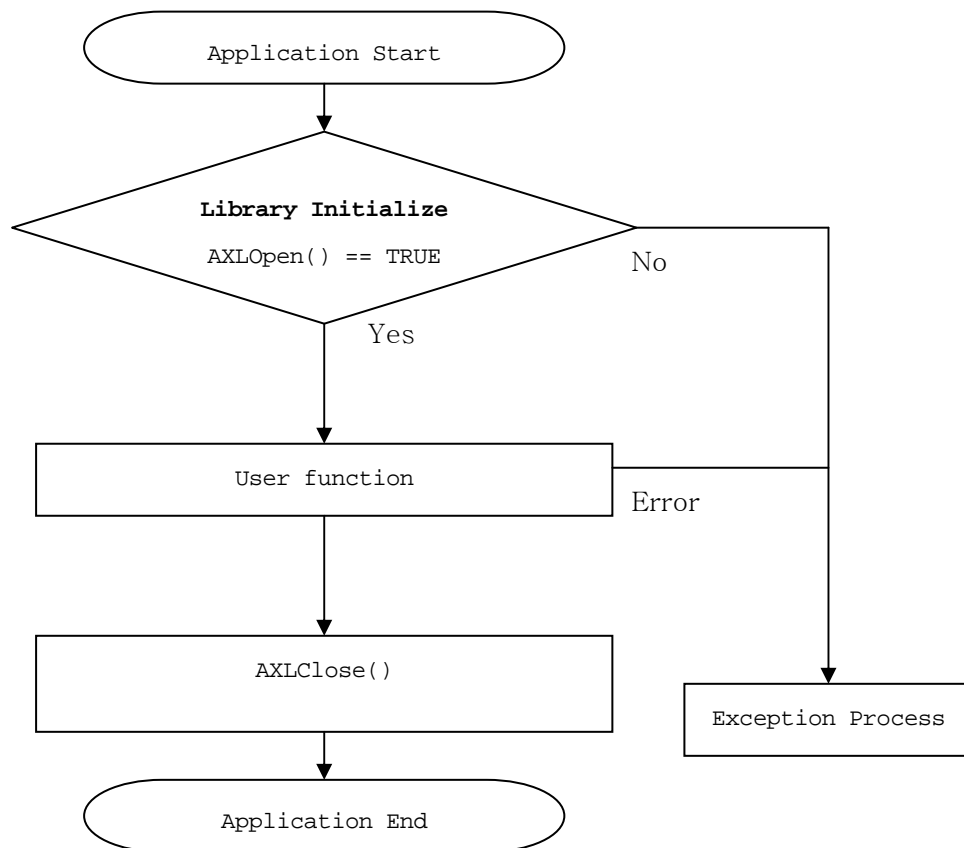
파일 위치는 “라이브러리 제공 디렉토리”WAXLNet(library)WLibrary 에 있습니다.

단, Visual Basic 에서는 해당 운영체제의 system (운영체제가 Windows 2000 일 경우 %WINNT%\system32) 폴더에 복사하여 사용하셔야 합니다.



3. 라이브러리 초기화

3.1.Application Flow Chart



주 : 모듈별 함수 초기화는 예제를 참조

3.2. 함수 초기화 예제

1

다음 소스코드는 각 모듈들의 초기화를 위한 함수 초기화 예를 들은 것입니다. 사용자께서는 이 초기화 루틴을 참고하시어 해당 프로그램에 삽입하시면 됩니다.

```
//++
// 라이브러리 초기화
// 윈도우 핸들 사용하지 않음
if (AxlOpen() == AXT_RT_SUCCESS)          // AXL.h
{
    DWORD dwStatus;

    // AIO 모듈이 있는지 확인한다.
    if (AxaInfoIsAIOModule(&dwStatus) == AXT_RT_SUCCESS)      // AXA.h
    {
        if (dwStatus == STATUS_EXIST)
        {
            printf("AIO 모듈이 있는 Node 존재합니다.\n");
        }
        else
        {
            printf("AIO 모듈이 있는 Node 존재하지 않습니다.\n");
        }
    }

    // Motion 모듈이 있는지 확인한다.
    if (AxmInfoIsMotionModule(&dwStatus) == AXT_RT_SUCCESS) // AXM.h
    {
        if (dwStatus == STATUS_EXIST)
        {
            printf("Motion 모듈이 있는 Node 존재합니다.\n");
        }
        else
        {
            printf("Motion 모듈이 있는 Node 존재하지 않습니다.\n");
        }
    }
}
```

```

    }

    // DIO 모듈이 있는지 확인한다.
    if (AxdInfoIsDIOModule (&dwStatus) == AXT_RT_SUCCESS)// AXD.h
    {
        if (dwStatus == STATUS_EXIST)
        {
            printf("DIO 모듈이 있는 Node 존재합니다.\n");
        }
        else
        {
            printf("DIO 모듈이 있는 Node 존재하지 않습니다.\n");
        }
    }
}
else
{
    printf("라이브러리를 초기화 하지 못했습니다.\n");
}

// 라이브러리가 사용 가능하지 (초기화가 되었는지)를 확인한다.
if (AxlIsOpened ())          // AXL.h
{
    // 라이브러리 리소스를 닫는다.
    AxlClose ();              // AXL.h
}
}

```

2

AxlOpen 은 통합라이브러리 초기화 하는 함수 입니다. 만일 Ring topology 를 위한 연결을 지원한다면 **AxlRingOpen** 을 사용하시기 바랍니다.

AxIsAIOModule 은 아날로그 입출력 모듈이 장착된 NODE 가 존재하는지 확인하는 함수 입니다. 만약 해당 모듈이 없다면 0 값이 반환됩니다.

AxIsMotionModule 은 모션 모듈이 장착된 NODE 가 존재하는지 확인하는 함수 입니다. 만약 해당 모듈이 없다면 0 값이 반환됩니다.

AxdlIsDIOModule 은 디지털 입출력 모듈이 장착된 NODE 가 존재하는지 확인하는 함수 입니다. 만약 해당 모듈이 없다면 0 값이 반환됩니다.

AxllsOpened 는 통합 라이브러리가 사용 가능하지 (초기화가 되었는지)를 확인합니다. 초기화 되었으면 'TRUE'를 반환하고 그렇지 않으면 'FALSE'를 반환합니다.

AxlClose 는 통합 라이브러리의 사용을 종료하는 함수 입니다. 프로그램 종료하기 전에 이 함수를 반드시 사용합니다.

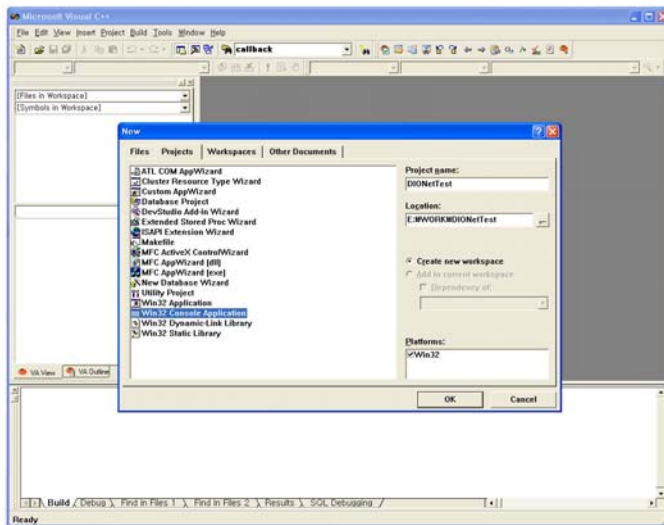
아주 기본적인 초기화에 관한 함수 설명만 했습니다. 자세한 함수 설명은 해당 모듈의 라이브러리 매뉴얼을 참조하시기 바랍니다.

4. Visual C++과 DIO 모듈을 이용한 초기화 예제

4.1. Win32 Console용 DIO 시작

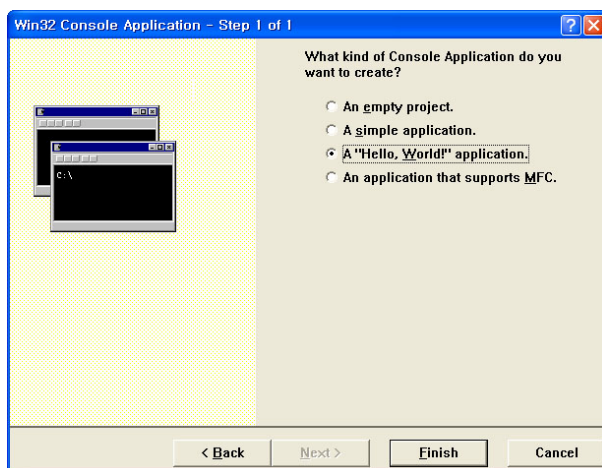
1

Visual C++를 실행시킨 후 새로운 프로젝트를 생성합니다. 이 때 나타나는 다이얼로그에서 Win32 Console Application 을 선택한 후 파일을 저장할 위치를 지정한 후 Project name 에 원하는 프로젝트의 이름을 적은 후 OK 버튼을 누릅니다.



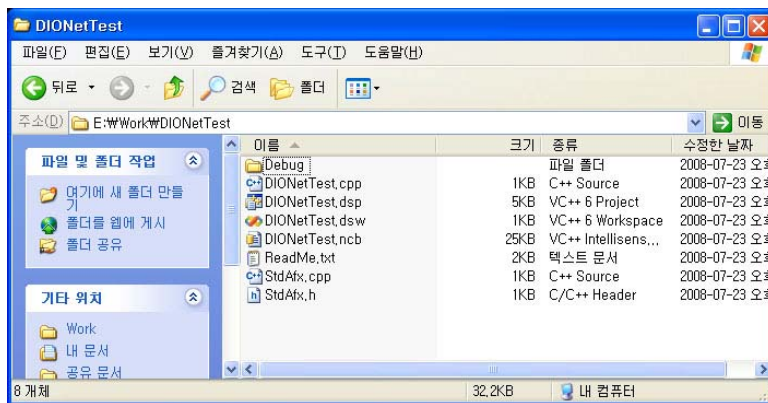
2

여러 형태의 프로젝트를 생성할 수 있는데 기본적인 코드가 생성되는 A “Hello, World!” application.을 선택한 후 “Finish” 버튼을 눌러 프로젝트를 생성합니다.



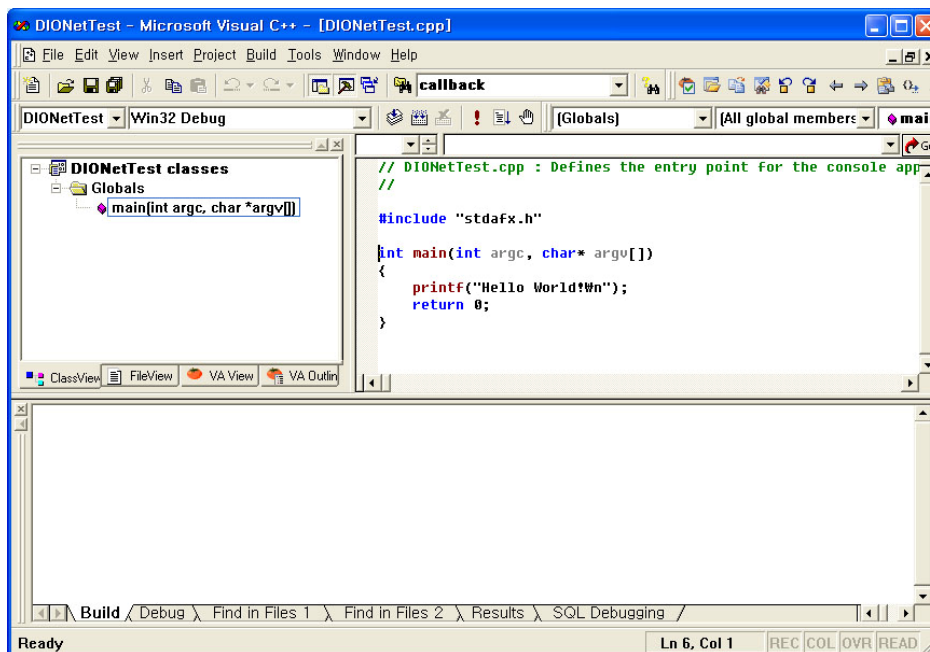
3

지정한 디렉토리에 지정한 이름으로 새로운 파일이 만들어진 것을 확인할 수 있습니다. 이 폴더에 ‘5.5. DLL 파일 복사’에 있는 파일들을 복사합니다.

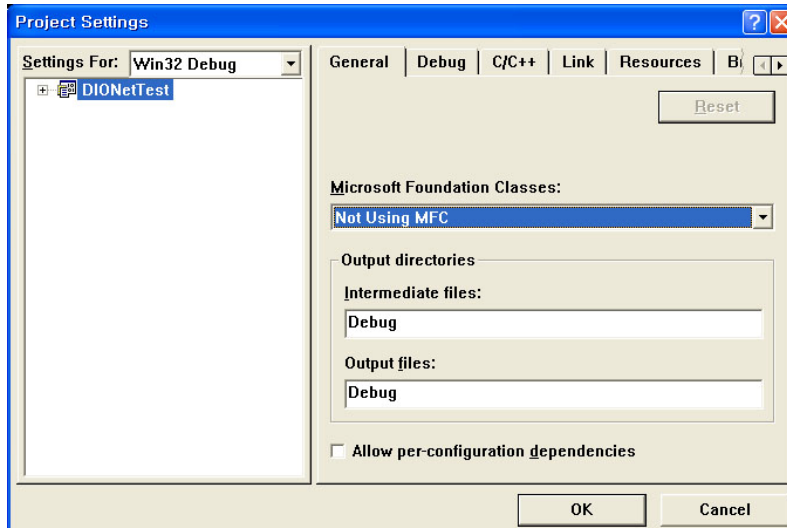


4

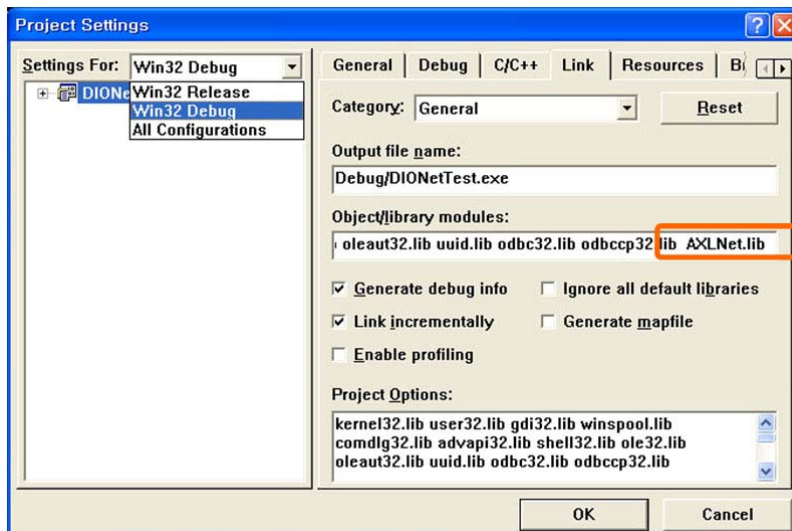
Visual C++의 Workspace 에서 ClassView 를 클릭하면 main() 함수가 나옵니다. 이 함수 안에 간단하게 DIO 초기화 및 테스트 함수를 사용 해 보겠습니다.



- 5** EtherNet/IP CPU NODE 및 DIO 모듈을 사용하기 위해선 AXLNet.lib 를 연결해야 합니다. 시스템 메뉴에서 Project -> Settings 을 클릭하면 Project Setting 화면이 나옵니다.



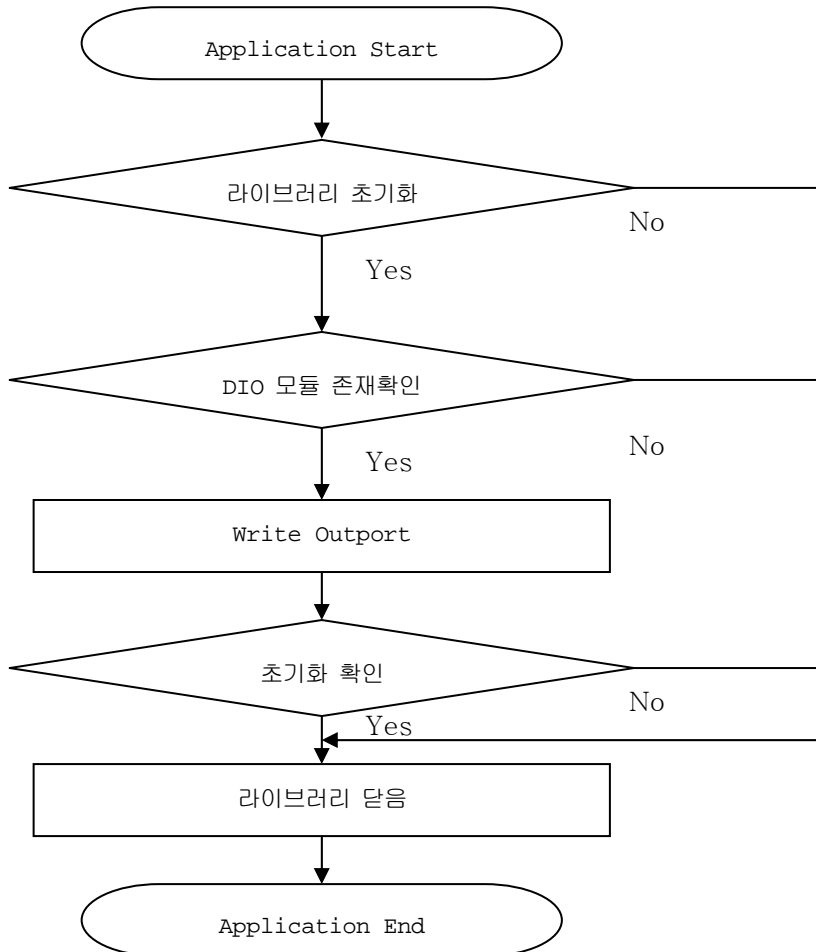
- 6** Settings For: 콤보에서 Win32 Debug 를 선택하고 우측 상단 탭에서 Link 를 클릭합니다. 그러면 Object/library modules: 항목이 있는데 이 항목 제일 뒤에 'AXLNet.lib'라고 입력합니다.
- 입력 완료 했으면 Settings For: 콤보에서 Win32 Release 를 선택하고 Object/library modules: 항목 제일 뒤에 'AXLNet.lib'라고 똑같이 입력하고 OK 버튼을 클릭합니다.



4.2.Flow Chart 및 Source

1

DIO 예제의 계략적인 Flow Chart는 다음과 같이 라이브러리 오픈, 네트워크 초기화, 모듈 초기화 순으로 합니다.



2

DioTest.cpp 에서 다음과 같은 코드를 입력합니다.

```

01 #include "stdafx.h"
02 #include "AXL.h"      // 네트워크 관련 함수
03 #include "AXD.h"      // DIO 모듈 관련 함수
04
05 int main(int argc, char* argv[])
06 {
07     WORDwOutput;
  
```

```

08     WORD wInput;
09     //++
10     // 라이브러리 초기화
11     // 윈도우 핸들 사용하지 않음
12     if (AxlOpen() == AXT_RT_SUCCESS )
13     {
14         DWORD dwStatus;
15         long lCount;
16
17         // DIO 모듈이 존재하는지 확인.
18         if (AxdInfoIsDIOModule (&dwStatus) == AXT_RT_SUCCESS)// AXD.h
19         {
20             if (dwStatus == STATUS_EXIST)
21             {
22                 printf("DIO 모듈이 존재합니다.\n");
23
24                 AxlGetNodeCount(&lCount);
25                 printf("\n 노드의 갯수는 [%d] 개입니다", lCount);
26
27                 AxdInfoGetModuleCount(&lCount);
28                 printf("\nDIO 모듈의 갯수는 [%d] 개입니다", lCount);
29                 printf("\n\n");
30
31                 wOutput = 0x1234;
32                 //++
33                 // 출력 되는 신호를 WORD 로 쓴다.
34                 AxdoWriteOutportWord(0, 0, wOutput);
35
36                 printf("16bit 신호 출력 하기 (0 ~ 15) : %04Xh\n", wOutput);
37
38                 //++
39                 // 출력된 신호를 WORD 로 읽는다.
40                 AxdoReadOutportWord(0, 0, &wInput);
41
42                 printf("16bit 입력 신호 읽기 (0 ~ 15) : %04Xh\n", wInput);
43             }
44         }
45     }
46     else

```

```
45         {
46             printf("DIO 모듈이 존재하지 않습니다.\n");
47         }
48     }
49 }
50 else
51 {
52     printf("라이브러리를 초기화 하지 못했습니다.\n");
53 }
54
55 // 통합 라이브러리가 사용 가능하지 (초기화가 되었는지)를 확인한다.
56 if (AxlIsOpened())
57 {
58     // 출력을 초기화 한다.
59     // AxdWritePortWord(0, 0, 0x0000);
60
61     // 라이브러리를 닫는다.
62     AxlClose();
63
64     printf("\n 라이브러리 및 DIO 모듈 종료 하였습니다.\n");
65     printf("\n 아무 키나 누르세요.\n");
66     getchar();
67 }
68
69 return 0;
70 }
```

3

실행 결과는 다음과 같이 나오면 성공적으로 작업이 완료 된 것입니다.

DIO Library 에 관한 자세한 설명은 해당 라이브러리 매뉴얼을 참조하시기 바랍니다.

```
D:\WORK\WABOUT_HARDWARE_DEVELOPING\WABOUT_DISTRIBUTED\WABOUT_...
DIO 모듈이 존재합니다.
노드 갯수는 [2] 개입니다
DIO 모듈의 갯수는 [2] 개입니다
16bit 신호 출력 하기 <0 ~ 15> : 1234h
16bit 입력 신호 읽기 <0 ~ 15> : 1234h
라이브러리 및 DIO 모듈 종료 하였습니다.
아무 키나 누르세요.
```


4.3.에러 조치

이 장 에러 처리를 읽으신다면 지금까지 제대로 따라 하지 않아서 발생한 에러들 입니다.

라이브러리와 관련된 에러는 크게 네 가지가 있으니 필히 숙지하시기 바랍니다.

Error executing 발생

D:\WMYPROJECTS\WVCWDIOTesst\WDIOTest.cpp(28) : error C2065: 'AxlOpen' : undeclared identifier

라는 Error 는 해당 함수의 헤더 파일이 없어서 발생한 에러 메시지 입니다.

main 함수 위에 AXL.h 와 AXD.h 헤더 파일을 Including 시켜주면 됩니다.

```

01 #include "stdafx.h"
02 #include "AXL.h"          // 네트워크 관련 함수
03 #include "AXD.h"          // DIO 모듈 관련 함수
04
05 int main(int argc, char* argv[])
06 {
    .
    .
    .

```

Cannot open include file 발생

D:\WMYPROJECTS\WVCWDIOTesst\WDIOTest.cpp(5) : fatal error C1083: Cannot open include file: 'AXL.h'

라는 Error 는 AXL.h 와 AXD.h 헤더 파일을 선언 해 줬으나 경로가 맞지 않아 발생한 에러 메시지 입니다.

'2.Visual C++ 라이브러리 링크하기'를 다시 참조하여 Include 파일과 Library 파일의 경로를 다시 설정 해주시기 바랍니다.

fatal error LNK1120 발생

DIOTest.obj : error LNK2001: unresolved external symbol _AXLOpen@0

Debug/DIOTest.exe : fatal error LNK1120: 1 unresolved externals

이 Error 는 AXLNet.lib 파일이 링크되지 못하여 발생한 에러 메시지 입니다.

‘4.1.Win32 Console 용 DIO 예제의 5 번 6 번’를 다시 참조하여 AXLNet.lib 파일을 다시 링크 시켜주시기 바랍니다.

DLL 찾기 실패 발생

DIOTest.exe – DLL 찾기 실패

Error 는 실행파일에 있는 폴더에 AXLNet.dll 파일이 없어서 발생한 에러 메시지 입니다.

2.2. DLL 파일 복사’를 다시 참조하여 해당 DLL 파일들을 작업 폴더에 다시 복사 하시기 바랍니다.

기타 에러 발생

위의 네가지 사항 이 외에 에러가 발생하였다면 AXLibrary 와는 관련없는 문법 에러나 잘못된 번지 참조로 발생한 에러가 대부분 일 것입니다.

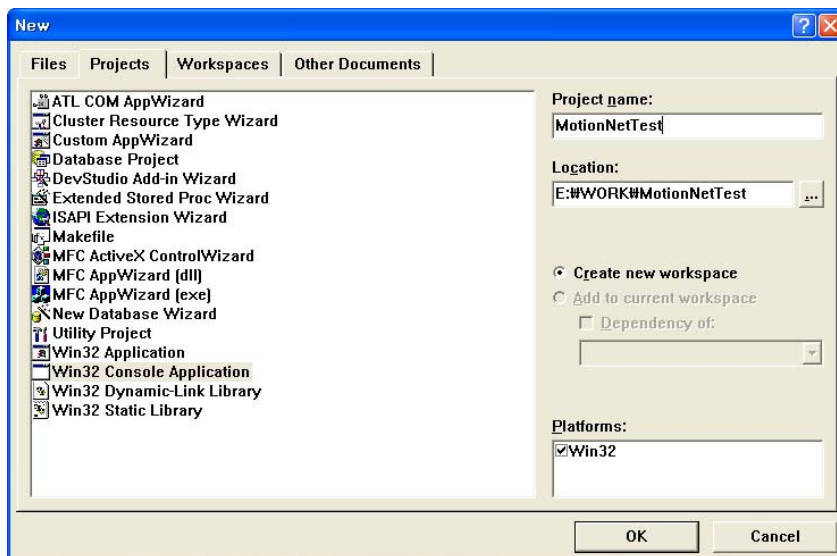
소스 내용을 다시 한번 더 확인 해보시기 바랍니다.

5. Visual C++과 Motion모듈을 이용한 초기화 예제

5.1.Win32 Console용 Motion 시작

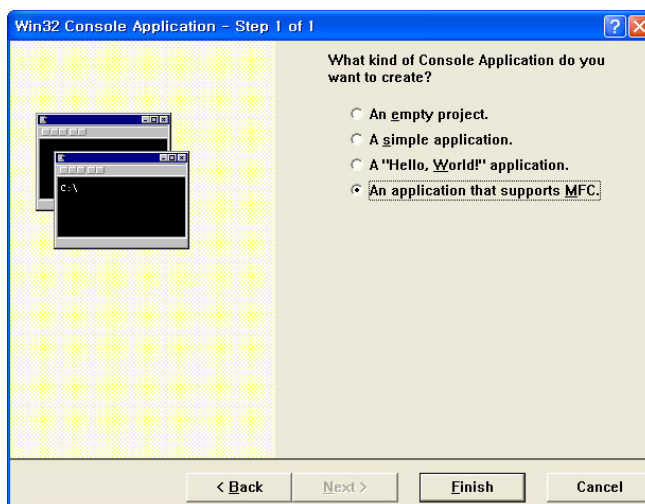
1

Visual C++를 실행시킨 후 새로운 프로젝트를 생성합니다. 이 때 나타나는 다이얼로그에서 Win32 Console Application 을 선택한 후 파일을 저장할 위치를 지정한 후 Project name 에 원하는 프로젝트의 이름을 적은 후 OK 버튼을 누릅니다.



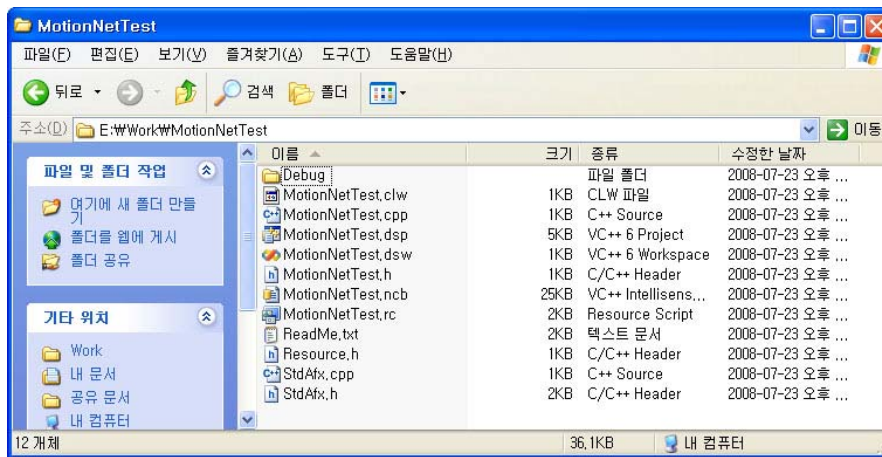
2

여러 형태의 프로젝트를 생성할 수 있는데 An application that supports MFC.를 선택한 후 “Finish”버튼을 눌러 프로젝트를 생성합니다.

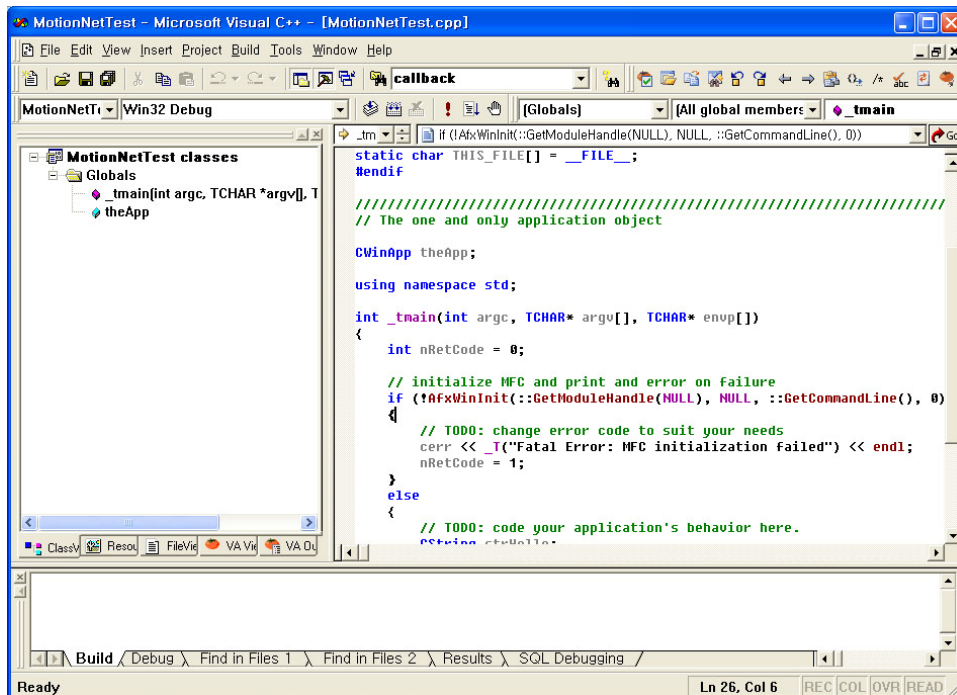


3

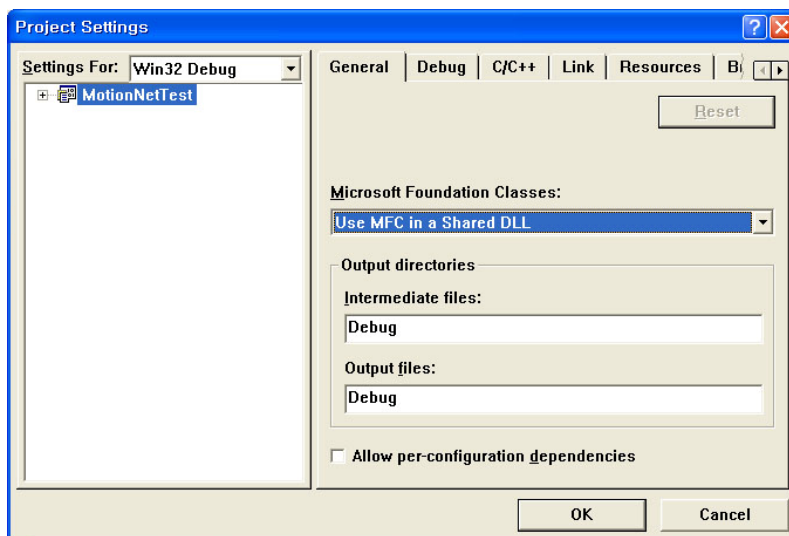
지정한 디렉토리에 지정한 이름으로 새로운 파일이 만들어진 것을 확인할 수 있습니다. 이 폴더에 '5.5. DLL 파일 복사'에 있는 파일들을 복사합니다.



4 Visual C++의 Workspace 에서 ClassView 를 클릭하면 `_tmain()` 함수가 나옵니다. 이 함수 안에 간단하게 Motion 초기화 및 테스트 함수를 사용 해 보겠습니다.

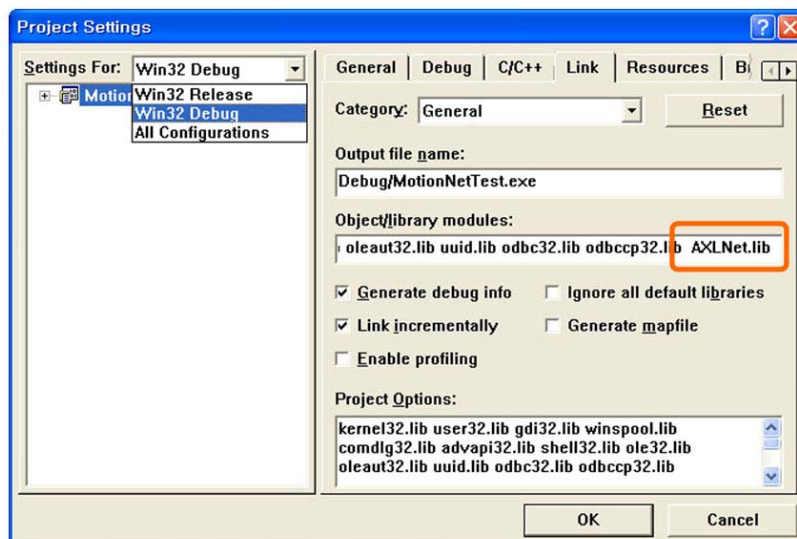


5 노드 및 모션 모듈을 사용하기 위해선 AXLNet.lib 를 연결해야 합니다. 시스템 메뉴에서 Project -> Settings 을 클릭하면 Project Setting 화면이 나옵니다.



6 Settings For: 콤보에서 Win32 Debug 를 선택하고 우측 상단 탭에서 Link 를 클릭합니다. 그러면 Object/library modules: 항목이 있는데 이 항목 제일 뒤에 'AXLNet.lib'라고 입력합니다.

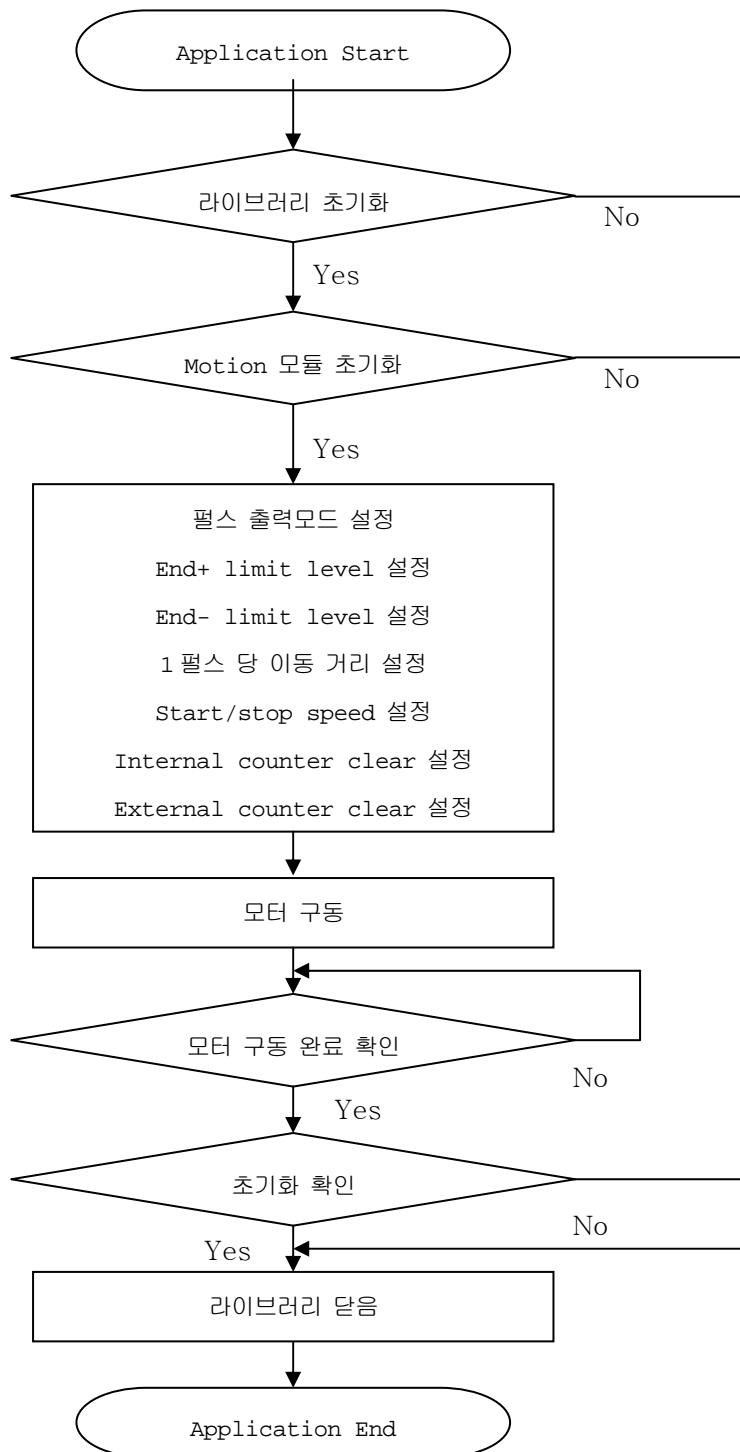
입력 완료 했으면 Settings For: 콤보에서 Win32 Release 를 선택하고 Object/library modules: 항목 제일 뒤에 'AXLNet.lib'라고 똑같이 입력하고 OK 버튼을 클릭합니다.



5.2.Flow Chart 및 Source

1

Motion 예제의 계략적인 Flow Chart 는 다음과 같이 라이브러리 오픈, EtherNet/IP CPU NODE 초기화, 모듈 초기화 순으로 합니다.



2

MotionTest.cpp 에 다음과 같은 코드를 입력합니다.

```
#include "stdafx.h"

#include "MotionTest.h"

#include "AXL.h"

#include "AXM.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////

// The one and only application object

CWinApp theApp;

using namespace std;

bool OpenDevice()
{
    //++
    // 라이브러리 초기화
    // 윈도우 핸들 사용하지 않음
    if (AxlOpen() == AXT_RT_SUCCESS)
    {
        DWORD dwStatus;

        // Motion 모듈이 존재하는지 확인.
        if (AxmInfoIsMotionModule (&dwStatus) == AXT_RT_SUCCESS)// AXM.h
        {
            if (dwStatus == STATUS_EXIST)
            {
                printf("Motion 모듈이 존재합니다.\n");

                // 0 축 설정
                // 펄스 출력모드 설정 : 2 펄스 모드
            }
        }
    }
}
```

```

    AxmMotSetPulseOutMethod(0, TwoCwCcwHigh);

    // End+ limit level : HIGH, End- limit level : HIGH
    AxmSignalSetLimit(0, EMERGENCY_STOP, HIGH, HIGH);

    AxmMotSetMoveUnitPerPulse(0, 1, 1); // 1 펄스 당 이동 거리
    AxmMotSetMinVel(0, 10.0);           // Start/stop speed

    // Internal counter clear
    AxmStatusSetCmdPos(0, 0.0);

    // External counter clear
    AxmStatusSetActPos(0, 0.0);

    // Inposition 사용 유무 : 사용하지 않음
    AxmSignalSetInpos(0, DISABLE);

    // Alarm 사용 유무 : 사용하지 않음
    AxmSignalSetServoAlarm(0, DISABLE);

    // 최대 속도 설정
    AxmMotSetMaxVel(0, 10000.0);

    // 1 축 설정
    // 펄스 출력모드 설정 : 2 펄스 모드
    AxmMotSetPulseOutMethod(1, TwoCwCcwHigh);

    // End+ limit level : HIGH, End- limit level : HIGH
    AxmSignalSetLimit(1, EMERGENCY_STOP, HIGH, HIGH);

    AxmMotSetMoveUnitPerPulse(1, 1, 1); // 1 펄스 당 이동 거리
    AxmMotSetMinVel(1, 10.0);           // Start/stop speed

    // Internal counter clear
    AxmStatusSetCmdPos(1, 0.0);

    // External counter clear
    AxmStatusSetActPos(1, 0.0);

    // Inposition 사용 유무 : 사용하지 않음
    AxmSignalSetInpos(1, DISABLE);

    // Alarm 사용 유무 : 사용하지 않음
    AxmSignalSetServoAlarm(1, DISABLE);

    // 최대 속도 설정
    AxmMotSetMaxVel(1, 10000.0);

}
else
{

```



```

        printf("Motion 모듈이 존재하지 않습니다.\n");
        return false;
    }
}
else
{
    printf("라이브러리를 초기화 하지 못했습니다.\n");
    return false;
}
return true;
}

bool CloseDevice()
{
    // 통합 라이브러리가 사용 가능하지 (초기화가 되었는지)를 확인한다.
    if (Ax1IsOpened())
    {
        // 라이브러리를 닫는다.
        Ax1Close();

        printf("\n 라이브러리 및 Motion 모듈 종료 하였습니다.\n");
        return true;
    }
    else
    {
        return false;
    }
}

bool RunMove()
{
    // 구동 테스트
    DWORD dwStatus;
    AxmStatusReadInMotion(0, &dwStatus);

    if (!dwStatus)
    {

```

```

        // 0 축에 절대 좌표로 1000 만큼 100 의 속도와 200 의 가속율로 구동
        printf("0 축 AxmMoveStartPos 실행\n");
        AxmMoveStartPos(0, 1000, 100, 200, 200);

        // 1 축에 절대 좌표로 1000 만큼 100 의 속도와 200 의 가속율로 구동
        printf("1 축 AxmMoveStartPos 실행\n");
        AxmMoveStartPos(1, 1000, 100, 200, 200);
    }
    else
    {
        return false;
    }

    return true;
}

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    // initialize MFC and print and error on failure
    if (!AfxWinInit(::GetModuleHandle(NULL), NULL, ::GetCommandLine(), 0))
    {
        // TODO: change error code to suit your needs
        cerr << _T("Fatal Error: MFC initialization failed") << endl;
        nRetCode = 1;
    }
    else
    {
        // 라이브러리 및 모션 초기화
        if(OpenDevice())
        {
            // 모터 구동
            RunMove();

            // 모터가 구동 완료될 때까지 대기한다.
            DWORD    dwInMotion = TRUE;

```

```

        do{
            AxmStatusReadInMotion(0, &dwInMotion);
        }
        while (dwInMotion);
    }

    // 라이브러리 및 모션 종료
    CloseDevice();

    getchar();

    return nRetCode;
}

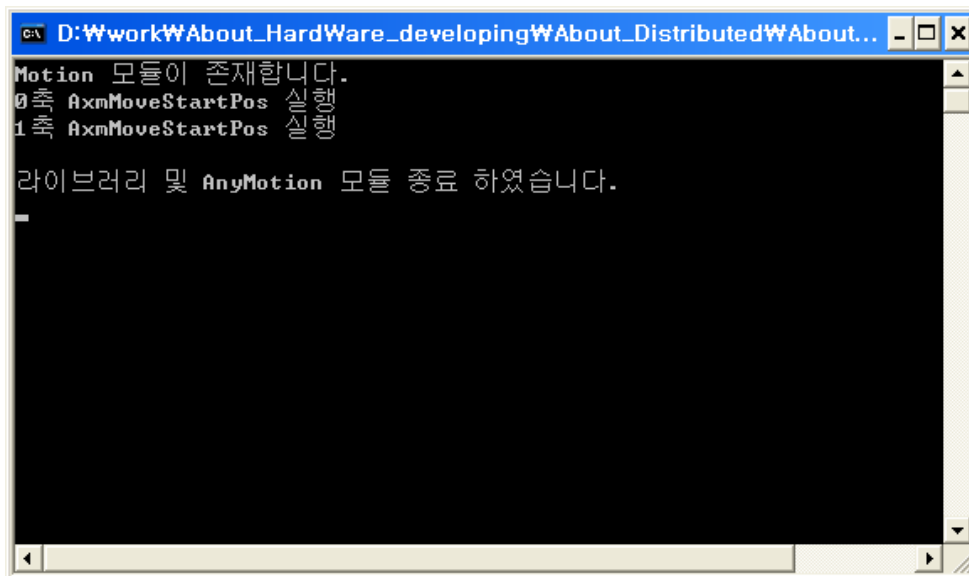
```

3

실행후 10 초 후에 다음과 같이 나오면 성공적으로 작업이 완료 된 것입니다.

만약, 에러가 나와 제대로 실행 되지 않았다면 '5.3. 에러 조치'에서 천천히 확인해보시기 바랍니다.

Motion Library 에 관한 자세한 설명은 해당 라이브러리 매뉴얼을 참조하시기 바랍니다.



5.3.에러 조치

이 장 에러 처리를 읽으신다면 지금까지 제대로 따라 하지 않아 발생한 에러를 입니다.

라이브러리와 관련된 에러는 크게 네 가지가 있으니 필히 숙지하시기 바랍니다.

Error executing 발생

D:\WMyProjects\WVCWMotionTest\WMotionTest.cpp(83) : error C2065: 'AxmMoveStartPos' : undeclared identifier

이 Error 는 해당 함수의 헤더 파일이 없어서 발생한 에러 메시지 입니다.

main 함수 위에 AXL.h 와 AXM.h 헤더 파일을 Including 시켜주면 됩니다.

```
001 #include "stdafx.h"
002 #include "AXL.h"          // 네트워크 관련 함수
003 #include "AXM.h"          // Motion 모듈 관련 함수
004
005 int main(int argc, char* argv[])
006 {
    .
    .
    .
```

Cannot open include file 발생

D:\WMyProjects\WVCWMotionTest\WMotionTest.cpp(5) : fatal error C1083: Cannot open include file: 'AXL.h'

이 Error 는 AXL.h 와 AXM.h 헤더 파일을 선언 해 줬으나 경로가 맞지 않아 발생한 에러 메시지 입니다.

'2.1.Visual C++ 라이브러리 링크하기'를 다시 참조하여 Include 파일과 Library 파일의 경로를 다시 설정 해주시기 바랍니다.

fatal error LNK1120 발생

MotionTest.obj : error LNK2001: unresolved external symbol _AXLOpen@2

이 Error 는 AXLNet.lib 파일이 링크되지 못하여 발생한 에러 메시지 입니다.

'4.1.Win32 Console 용 DIO 예제의 5 번 6 번'를 다시 참조하여 AXLNet.lib 파일을 다시 링크 시켜주시기 바랍니다.

DLL 찾기 실패 발생

MotionTest.exe – DLL 찾기 실패

이 Error 는 실행파일에 있는 폴더에 AXLNet.dll 파일이 없어서 발생한 에러 메시지 입니다.

2.2. DLL 파일 복사'를 다시 참조하여 해당 DLL 파일들을 작업 폴더에 다시 복사 하시기 바랍니다.

기타 에러 발생

위의 네 가지 사항 이 외에 에러가 발생하였다면 AXLibrary 와는 관련없는 문법 에러나 잘못된 번지 참조로 발생한 에러가 대부분 일 것입니다.

소스 내용을 다시 한번 더 확인 해보시기 바랍니다.

6. Base Command Function List

라이브러리 초기화

[AxlOpen](#)
[AxlInit](#)
[AxlClose](#)
[AxllsOpened](#)

라이브러리 및 베이스 보드 정보

[AxlGetNodeCount](#)
[AxlGetNodeInfo](#)
[AxlGetNetInfo](#)
[AxlGetLibVersion](#)

로그 레벨

[AxlSetLogLevel](#)
[AxlGetLogLevel](#)

네트워크 상태 관련 함수

[AxlNetStatusRead](#)
[AxlGetNodeFirmVersion](#)
[AxlSaveParamToFlash](#)
[AxlSaveParamFromFlash](#)
[AxlResetParamFlash](#)

6.1. 라이브러리 초기화

Function	Description	비고
AxlOpen	라이브러리를 초기화 한다.	
AxlInit	노드에 장착된 모듈을 초기화한다	
AxlClose	라이브러리 사용을 종료 한다.	
AxllsOpened	라이브러리가 초기화 되어 있는 지 확인한다.	

AxlOpen

Purpose

라이브러리를 초기화 한다.

Format

C

Function AxlOpen() As Byte

Visual Basic

Function AxlOpen() As Byte

Delphi

function AxlOpen() : Boolean; stdcall;

Input / Output

None

Return AXT_RT_SUCCESS(0000) : Successful execution of API
[* See error code Table more information on status error codes](#)

Description

라이브러리를 초기화를 시켜 준다.

네트워크 변경 사항이 있는 상황에서 Back Ground Process를 정지 시키고, 다시 실행 하게 되면, Configuration이 변경 되게 되므로 AxlOpen을 다시 하여야 한다.

C Example

```
// 라이브러리 사용을 초기화 한다.
AxlOpen();
```

VB Example

```
' 라이브러리 사용을 초기화 한다.
AxlOpen
```

Delphi Example

```
{ 라이브러리 사용을 초기화 한다. }
AxlOpen();
```

See Also

[AxlInit](#), [AxlClose](#), [AxllsOpened](#)

AxlInit

Purpose

노드에 장착된 모듈을 초기화 한다

Format

C

```
DWORD AxlInit(long INodeNum);
```

Visual Basic

```
Function AxlInitModules (ByVal INodeNum As Long) As Long
```

Delphi

```
function AxlInitModules (INodeNum: LongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	Long	-	노드 번호

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

노드에 장착된 모듈을 초기화 한다.

네트워크 변경 사항이 있는 상황에서 Back Ground Process를 정지 시키고, 다시 실행 하게 되면, Configuration이 변경 되게 되므로 AxlOpen을 다시 하여야 한다.

C Example

```
// 0 노드에 모듈을 초기화 한다
AxlInit(0);
```

VB Example

```
'0모듈에 모듈을 초기화한다.
AxlInit(0)
```

Delphi Example

```
{ 0노드에 모듈을 초기화한다. }
AxlInit(0);
```

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;
```

Input / Output

None

Return TRUE : 라이브러리가 정상적으로 종료 되었음
 FALSE : 라이브러리가 정상적으로 종료되지 않았음

Description

라이브러리 사용을 종료한다.

C Example

```
// 라이브러리 사용을 종료 한다.  
AxlClose()
```

VB Example

```
' 라이브러리 사용을 종료 한다.  
AxlClose
```

Delphi Example

```
{ 라이브러리 사용을 종료 한다. }  
AxlClose();
```

See Also

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

AxllsOpened

Purpose

라이브러리가 초기화 되어 있는 지 확인한다.

Format

C

```
BOOL AxllsOpened();
```

Visual Basic

```
Function AxllsOpened() As Byte
```

Delphi

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

Input / Output

None

Return

TRUE : 라이브러리가 정상적으로 종료 되었음

FALSE : 라이브러리가 정상적으로 종료되지 않았음

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

C Example

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

VB Example

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

Delphi Example

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

See Also

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

6.2. 라이브러리 및 베이스 보드 정보

Function	Description	비고
AxlGetNodeCount	등록된 노드의 개수를 확인한다.	
AxlGetNodeInfo	노드 정보를 확인한다	
AxlGetNetInfo	등록된 노드 정보를 배열로 제공한다	
AxlGetLibVersion	라이브러리 버전을 확인한다.	

AxlGetNodeCount

Purpose

등록된 노드의 개수를 확인한다

Format

C

```
DWORD AxlGetNodeCount(long * lNodeNum);
```

Visual Basic

```
Function AxlGetNodeCount(ByRef lNodeNum As Long) As Long
```

Delphi

```
function AxlGetNodeCount(lNodeNum : PLongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
lNodeNum	out	Long*	-	노드의 개수

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

등록된 노드의 개수를 확인한다.

C Example

```
// 등록된 베이스 보드의 개수를 확인한다.
long lNodeCounts;
CString strData;

AxlGetNodeCount (&lNodeCounts);
strData.Format( "전체 베이스 보드 개수는 %d개 입니다." , lNodeCounts);

AfxMessageBox(strData);
```

VB Example

```
' 등록된 베이스 보드의 개수를 확인한다.
Dim lNodeCounts As Long
Dim strData As String

AxlGetNodeCount (lNodeCounts)
strData = "전체 베이스 보드 개수는 " + CStr(lNodeCounts) + " 개 입니다."

MsgBox strData
```

Delphi Example

```
{ 등록된 베이스 보드의 개수를 확인한다. }
var
lNodeCounts : LongInt;
```

```
strData : String;  
  
begin  
  AxlGetNodeCount (@lNodeCounts);  
  strData := '전체 베이스 보드 개수는 ' + IntToStr(lNodeCounts) + '개 입니다.' ;  
  
  Application.MessageBox (PCHAR(strData), 'Ajintek', MB_OK);  
End;
```

See Also

[AxlGetNodeInfo](#), [AxlGetNetInfo](#), [AxlGetLibVersion](#)

AxlGetNodeInfo

Purpose

등록된 노드의 정보를 확인한다.

Format

C

```
DWORD AxlGetNodeInfo (long INodeNum, DWORD *dwModuleId);
```

Visual Basic

```
Function AxlGetNodeInfo (ByVal INodeNum As Long, ByRef dwModuleId As Long) As Long
```

Delphi

```
function AxlGetNodeInfo (INodeNum: LongInt; dwModuleId: PDword) : Dword; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	In	Long*	-	노드의 번호
dwModuleId	out	Dword*	-	노드 정보 (Array)

Return AXT_RT_SUCCESS(0000) : Successful execution of API
 * [See error code Table more information on status error codes](#)

Description

등록된 노드의 정보를 확인한다 dwModuleId는 반드시 Array (size = 17) 선언해서 사용한다.

C Example

```
// 0 노드의 정보를 확인
DWORD dwModuleId[17]
AxlGetNodeInfo (0, dwModuleId);
```

VB Example

```
' 0노드에 정보를 확인
Dim dwModuleId As Long
AxlGetNodeInfo 0, dwModuleId
```

Delphi Example

```
{ 0축에 Home 신호의 입력 확인 }
var
dwModuleId: DWord;
begin
AxlGetNodeInfo (0, @dwModuleId);
end;
```

See Also

[AxlGetNodeCount](#), [AxlGetNetInfo](#), [AxlGetLibVersion](#)

AxlGetNetInfo

Purpose

등록된 노드 정보를 배열로 제공한다.

Format

C

```
DWORD AxlGetNetInfo(DWORD *upNodeCount, DWORD *upArrayNodeID);
```

Visual Basic

```
Function AxlGetNetInfo (ByRef upNodeCount As Long, ByRef upArrayNodeID As Long) As Long
```

Delphi

```
function AxlGetNetInfo (upNodeCount: PDword; upArrayNodeID: PDword) : Dword; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
upNodeCount	In	DWORD *	–	노드 개수
upArrayNodeID	out	DWORD *	–	노드 배열

Return AXT_RT_SUCCESS(0000) : Successful execution of API
[* See error code Table more information on status error codes](#)

Description

등록된 노드 정보(노드 개수, 노드 ID Array)를 제공한다. 노드 배열은 반드시 250개 배열로 선언해야 된다

C Example

```
// 등록된 노드의 정보를 확인
DWORD upNodeCount;
DWORD upArrayNodeID[250];

AxlGetNetInfo(&upNodeCount, upArrayNodeID);
```

VB Example

```
// 등록된 노드의 정보를 확인
Dim upNodeCount As Long
Dim upArrayNodeID (0 To 250) As Long

AxlGetNetInfo upNodeCount, upArrayNodeID(0)
```

Delphi Example

```
// 등록된 노드의 정보를 확인
var
upNodeCount: DWord;
upArrayNodeID: array [0..250] of DWord;
begin
AxlGetNetInfo(@upNodeCount, @upArrayNodeID);
end;
```




See Also

[AxlGetNodeCount](#), [AxlGetNodeInfo](#), [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;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
szVersion	out	char *	-	라이브러리 버전

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

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

C Example

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

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

AfxMessageBox(strData);
```

VB Example

```
' 라이브러리 버전을 확인한다.
Dim byVersion(30) As Byte
AxlGetLibVersion byVersion(0)
MsgBox StrConv(CStr(byVersion), vbUnicode)
```

Delphi Example

```
{ 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), 'Ajinextek', MB_OK);  
End;
```

See Also

[AxlGetNodeCount](#), [AxlGetNodeInfo](#), [AxlGetNetInfo](#)

6.3.로그 레벨

Function	Description	비고
AxlSetLogLevel	EzSpy에 출력할 메시지 레벨을 설정한다.	
AxlGetLogLevel	EzSpy에 출력할 메시지 레벨을 확인한다.	

AxlSetLogLevel

Purpose

EzSpy에 출력할 메시지 레벨을 설정한다.

Format

C

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

Visual Basic

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

Delphi

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

Input / Output

Name	in/out	Format	Init Value	Explanation
uLevel	in	Long	-	로그 레벨

Log Level

enum	Value	Explanation
LEVEL_NONE	0	모든 메시지를 출력하지 않는다.
LEVEL_ERROR	1	에러가 발생한 메시지만 출력한다.
LEVEL_RUNSTOP	2	모션에서 Run / Stop 관련 메시지를 출력한다.
LEVEL_FUNCTION	3	모든 메시지를 출력한다.

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

EzSpy에 출력할 메시지 레벨을 설정한다. 로그 레벨은 Log Level Table을 참조한다.

C Example

```
// Error 로그 레벨을 설정한다.
AxlSetLogLevel(LEVEL_ERROR);
```

VB Example

```
'Error 로그 레벨을 설정한다.
AxlSetLogLevel(1)
```

Delphi Example

```
{ Error 로그 레벨을 설정한다. }
AxlSetLogLevel(1);
```

See Also

[AxlGetLogLevel](#)

AxlGetLogLevel

Purpose

EzSpy에 출력할 메시지 레벨을 확인한다.

Format

C

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

Visual Basic

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

Delphi

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

Input / Output

Name	in/out	Format	Init Value	Explanation
uLevel	in	Long	-	로그 레벨

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

EzSpy에 출력할 메시지 레벨을 확인한다.

C Example

```
// 로그 레벨을 확인한다.
DWORD dwLogLevel;
CString strData;

AxlGetLogLevel(&dwLogLevel);
strData.Format("설정된 로그 레벨은 %d 입니다.", dwLogLevel);

AfxMessageBox(strData);
```

VB Example

```
' 로그 레벨을 확인한다.
Dim lLogLevel As Long
Dim strData As String

AxlGetLogLevel(lLogLevel)
strData = "설정된 로그 레벨은 " + CStr(lLogLevel) + "입니다."

MsgBox strData
```

Delphi Example

```
{ 로그 레벨을 확인한다. }
Var
dwLogLevel : DWord;
```

```
strData : String;  
  
begin  
  AxlGetLogLevel(@dwLogLevel);  
  strData:= '설정된 로그 레벨은 ' + IntToStr(dwLogLevel) + '입니다.' ;  
  Application.MessageBox(PCHAR(strData), 'Ajinextek', MB_OK);  
End;
```

See Also

[AxlSetLogLevel](#)

6.4.네트워크 상태 관련 함수

Function	Description	비고
AxlNetStatusRead	해당 노드의 현재 네트워크 상태를 확인 한다.	
AxlGetNodeFirmVersion	해당 노드의 Firmware 정보를 확인한다	
AxlSaveParamToFlash	해당 노드에 있는 모듈들의 파라미터를 저장한다	
AxlLoadParamFromFlash	해당 노드에 있는 모듈들의 설정값을 로드한다	
AxlResetParamFlash	해당 노드에 있는 모듈들의 설정값의 저장 공간을 초기화 한다	

AxlNetStatusRead

Purpose

해당 노드의 현재 네트워크 상태를 확인 한다.

Format

C

```
DWORD AxlNetStatusRead (long INodeNum, DWORD *dwpFlag);
```

Visual Basic

```
Function AxlNetStatusRead (ByVal IpNodeNum As Long, ByRef dwpFlag Long) As Long
```

Delphi

```
function AxlNetStatusRead (IpNodeNum: LongInt; upMethod : dwpFlag) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IpNodeNum	in	long	—	노드 번호
dwpFlag	out	* DWORD	—	상태 플래그

dwFlag

emum	Explanation
0	Connect
1	Disconnect

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

해당 노드의 현재 네트워크 상태를 확인 한다.

C Example

```
// 0 노드의 정보를 확인
DWORD dwpFlag;
AxlNetStatusRead (0, &dwpFlag);
```

VB Example

```
' 0노드에 정보를 확인
Dim dwpFlag As Long
AxlNetStatusRead 0, dwpFlag
```

Delphi Example

```
{ 0노드에 정보를 확인 }
var
dwpFlag: DWord;
begin
AxlNetStatusRead (0, @dwpFlag);
```

```
end;
```

See Also

[AxlGetNodeFirmVersion](#), [AxlSaveParamToFlash](#), [AxlLoadParamFromFlash](#), [AxlResetParamFlash](#)

AxlGetNodeFirmVersion

Purpose

해당 노드의 Firmware Version을 확인한다.

Format

C

```
DWORD AxlGetNodeFirmVersion (long lNodeNum, char *szVersion);
```

Visual Basic

```
Function AxlGetNodeFirmVersion (ByVal lpNodeNum As Long, ByRef ByRef szVersion As Byte) As Long
```

Delphi

```
function AxlGetNodeFirmVersion (lpNodeNum: LongInt; szVersion : PChar) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
lpNodeNum	in	long	–	노드 번호
szVersion	out	* char	–	Firmware 정보(Array 값)

Return AXT_RT_SUCCESS(0000) : Successful execution of API
[* See error code Table more information on status error codes](#)

Description

해당 노드의 Firmware Version을 확인한다.

C Example

```
// 0 노드의 Firmware정보를 확인
char szVersion[255];
AxlGetNodeFirmVersion (0, szVersion);
```

VB Example

```
' 0노드에 Firmware정보를 확인
Dim As szVersion Byte
AxlGetNodeFirmVersion 0, szVersion
```

Delphi Example

```
{ 0노드에 Firmware 정보를 확인 }
var
szVersion: char;
begin
AxlGetNodeFirmVersion (0, @ szVersion);
end;
```

See Also

[AxlNetStatusRead](#), [AxlSaveParamToFlash](#), [AxlLoadParamFromFlash](#), [AxlResetParamFlash](#)

AxlSaveParamToFlash

Purpose

해당 노드에 설정되어 있는 모듈들의 파라미터를 저장 내장된 Flash 메모리에 저장 한다.

Format

C

```
DWORD AxlSaveParamToFlash (long INodeNum);
```

Visual Basic

```
Function AxlSaveParamToFlash (ByVal INodeNum As Long) As Long
```

Delphi

```
function AxlSaveParamToFlash (INodeNum: LongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	Long	-	노드 번호

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

해당 노드에 있는 모듈들의 파라미터를 내장된 Flash 메모리에 저장한다. 해당 노드의 전원이 재인가 될 때 Flash에 저장된 내용으로 파라미터들이 자동으로 설정된다. 이 함수를 실행하면 최대 1.5 Sec 이후에 정상적으로 Flash에 기록이 완료 되기 때문에 완료 이전에 또 다시 실행하면, Error Code를 리턴하고 무시된다

C Example

```
// 0 노드에 파라미터를 저장한다
AxlSaveParamToFlash (0);
```

VB Example

```
'0모듈에 파라미터를 저장한다.
AxlSaveParamToFlash (0)
```

Delphi Example

```
{ 0노드에 모듈들의 파라미터를 저장한다. }
AxlSaveParamToFlash (0);
```

See Also

[AxlNetStatusRead](#), [AxlGetNodeFirmVersion](#), [AxlLoadParamFromFlash](#), [AxlResetParamFlash](#)

★ Firmware에 저장되는 Parameter

신호관련 변수	관련 제품	초기 값	파라메타 설명
PulseOutMethod	모션 모듈	TwoCwCcwHigh (6)	모션보드에서 서보팩 지령으로 사용될 펄스의 출력방식을 설정
EnclInputMethod	모션 모듈	ObverseSqr4Mod e (3)	Encoder 신호의 입력방식을 설정
Unit/Pulse	모션 모듈	1/1	모터 1회전 시 이동되는 이동 량 [mm, 도, 등 등] / 모터 1회전에 필요한 펄스 수 설정 [서보 팩 전자 기어 비 참조]
ProfileMode	모션 모듈	ASYM_S_CURVE_M ODE (4)	초기화시 속도 프로파일 모드
MinVel	모션 모듈	1	초기화시 속도(속도의 단위는 Unit/pulse를 1/1로 한경우에 PPS[Pulses/sec])
MaxVel	모션 모듈	700000	모터 구동 시 최대구동 속도 설정
AbsRelMode	모션 모듈	POS_ABS_MODE (0)	초기화시 절대/상대 모드
HomeLevel	모션 모듈	1	원점 신호 레벨을 설정
ZphaseLevel	모션 모듈	1	엔코더 Z상 레벨을 설정
ServoOnLevel	모션 모듈	1	서보 온 신호 레벨을 설정
ServoAlarmResetLevel	모션 모듈	1	알람 클리어 설정
Inposition	모션 모듈	2	위치 결정 신호 설정
Servo Alarm	모션 모듈	1	서보 알람 신호 설정
Limit	모션 모듈	1	리미트 센서의 사용 여부에 대한 레벨을 설정
Soft Limit	모션 모듈	uUse(1), uStopMode(0), uSelection(0)	Soft 리미트 값 설정
Stop	모션 모듈	1	ESTOP, SSTOP 사용시 모드
Network Error Action	모션 모듈	0	AxmNetWorkErrorSetAct ion로 설정된 내용
Network Error Action	DIO 모듈	0	AxdoNetWorkErrorSetAct ion로 설정된 내용
Network Error Value	DIO 모듈	0	AxdoNetWorkErrorSetByteValue로 설정된 내용

AxlLoadParamFromFlash

Purpose

해당 노드의 Flash에 저장된 파라미터를 적용한다.

Format

C

```
DWORD AxlLoadParamFromFlash (long INodeNum);
```

Visual Basic

```
Function AxlLoadParamFromFlash (ByVal INodeNum As Long) As Long
```

Delphi

```
function AxlLoadParamFromFlash (INodeNum: LongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	Long	-	노드 번호

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

해당 노드의 Flash에 저장된 파라미터를 적용한다. 만일 해당 노드의 파라미터가 Flash에 저장되어 있지 않으면 무시된다.

(주의) 해당 노드에 모듈을 추가 하거나 변경하는 등의 H/W 변경 사항이 발생하면 Flash에 저장된 정보가 잘못된 파라미터로 설정 될 수 있기 때문에 이 경우는 AxlResetParamFlash 이후에 AxlSaveParamToFlash로 다시 설정하여야 한다.

C Example

```
// 0 노드에 파라미터를 로드한다
AxlLoadParamFromFlash (0);
```

VB Example

```
'0모듈에 파라미터를 로드한다.
AxlLoadParamFromFlash (0)
```

Delphi Example

```
{ 0노드에 모듈들의 파라미터를 로드한다. }
AxlLoadParamFromFlash (0);
```

See Also

[AxlNetStatusRead](#), [AxlGetNodeFirmVersion](#), [AxlSaveParamToFlash](#),
[AxlResetParamFlash](#)

★ Firmware에 저장되는 Parameter

신호관련 변수	관련 제품	초기 값	파라메타 설명
PulseOutMethod	모션 모듈	TwoCwCcwHigh (6)	모션보드에서 서보팩 지령으로 사용될 펄스의 출력 방식을 설정
EnclInputMethod	모션 모듈	ObverseSqr4Mode (3)	Encoder 신호의 입력방식을 설정
Unit/Pulse	모션 모듈	1/1	모터 1회전 시 이동되는 이동 량 [mm, 도, 등등] / 모터 1회전에 필요한 펄스 수 설정 [서보팩 전자 기어 비 참조]
ProfileMode	모션 모듈	ASYM_S_CURVE_MODE (4)	초기화시 속도 프로파일 모드
MinVel	모션 모듈	1	초기화시 속도(속도의 단위는 Unit/pulse를 1/1로 한경우에 PPS[Pulses/sec])
MaxVel	모션 모듈	700000	모터 구동 시 최대구동 속도 설정
AbsRelMode	모션 모듈	POS_ABS_MODE (0)	초기화시 절대/상대 모드
HomeLevel	모션 모듈	1	원점 신호 레벨을 설정
ZphaseLevel	모션 모듈	1	엔코더 Z상 레벨을 설정
ServoOnLevel	모션 모듈	1	서보 온 신호 레벨을 설정
ServoAlarmResetLevel	모션 모듈	1	알람 클리어 설정
Inposition	모션 모듈	2	위치 결정 신호 설정
Servo Alarm	모션 모듈	1	서보 알람 신호 설정
Limit	모션 모듈	1	리미트 센서의 사용 여부에 대한 레벨을 설정
Soft Limit	모션 모듈	uUse(1), uStopMode(0), uSelection(0)	Soft 리미트 값 설정
Stop	모션 모듈	1	ESTOP, SSTOP 사용시 모드
Network Error Action	모션 모듈	0	AxmNetWorkErrorSetAction로 설정된 내용
Network Error Action	DIO 모듈	0	AxdoNetWorkErrorSetAction로 설정된 내용
Network Error Value	DIO 모듈	0	AxdoNetWorkErrorSetByteValue로 설정된 내용

AxlResetParamFlash

Purpose

해당 노드에 있는 모듈들의 설정값의 저장 공간을 초기화한다.

Format

C

```
DWORD AxlResetParamFlash (long INodeNum);
```

Visual Basic

```
Function AxlResetParamFlash (ByVal INodeNum As Long) As Long
```

Delphi

```
function AxlResetParamFlash (INodeNum: LongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	Long	-	노드 번호

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

해당 노드의 파라미터 저장 FLASH를 초기화하여 전원 재인가 시 Default값으로 설정 되도록 한다. Default 값은 “Firmware에 저장되는 Parameter”표의 초기값과 같다.

C Example

```
// 0 노드에 있는 모듈들의 설정값의 저장 공간을 초기화한다
AxlResetParamFlash (0);
```

VB Example

```
'// 0 노드에 있는 모듈들의 설정값의 저장 공간을 초기화한다
AxlResetParamFlash (0)
```

Delphi Example

```
{ // 0 노드에 있는 모듈들의 설정값의 저장 공간을 초기화한다. }
AxlResetParamFlash (0)
```

See Also

[AxlNetStatusRead](#), [AxlGetNodeFirmVersion](#), [AxlSaveParamToFlash](#), [AxlLoadParamFromFlash](#),

★ Firmware 에 저장되는 Parameter

신호관련 변수	관련 제품	초기 값	파라메타 설명
PulseOutMethod	모션 모듈	TwoCwCcwHigh (6)	모션보드에서 서보팩 지령으로 사용될 펄스의 출력 방식을 설정
EncInputMethod	모션 모듈	ObverseSqr4Mode (3)	Encoder 신호의 입력방식을 설정
Unit/Pulse	모션 모듈	1/1	모터 1회전 시 이동되는 이동 량 [mm, 도, 등등] / 모터 1회전에 필요한 펄스 수 설정 [서보팩 전자 기어 비 참조]
ProfileMode	모션 모듈	ASYM_S_CURVE_MODE (4)	초기화시 속도 프로파일 모드
MinVel	모션 모듈	1	초기화시 속도(속도의 단위는 Unit/pulse를 1/1로 한경우에 PPS[Pulses/sec])
MaxVel	모션 모듈	700000	모터 구동 시 최대구동 속도 설정
AbsRelMode	모션 모듈	POS_ABS_MODE (0)	초기화시 절대/상대 모드
HomeLevel	모션 모듈	1	원점 신호 레벨을 설정
ZphaseLevel	모션 모듈	1	엔코더 Z상 레벨을 설정
ServoOnLevel	모션 모듈	1	서보 온 신호 레벨을 설정
ServoAlarmResetLevel	모션 모듈	1	알람 클리어 설정
Inposition	모션 모듈	2	위치 결정 신호 설정
Servo Alarm	모션 모듈	1	서보 알람 신호 설정
Limit	모션 모듈	1	리미트 센서의 사용 여부에 대한 레벨을 설정
Soft Limit	모션 모듈	uUse(1), uStopMode(0), uSelection(0)	Soft 리미트 값 설정
Stop	모션 모듈	1	ESTOP, SSTOP 사용시 모드
Network Error Action	모션 모듈	0	AxmNetWorkErrorSetAction로 설정된 내용
Network Error Action	DIO 모듈	0	AxdoNetWorkErrorSetAction로 설정된 내용
Network Error Value	DIO 모듈	0	AxdoNetWorkErrorSetByteValue로 설정된 내용

에러코드 테이블(Error Code Table) 확인

라이브러리 사용 중 오 동작 또는 동작을 안 하는 경우가 발생 할 수 있다. 이러한 경우 함수의 리턴 값을 체크하므로 라이브러리 사용상에서의 문제점을 알 수 있다. 문제를 빨리 해결하고 손쉬운 디버깅을 위해 많은 에러 코드가 제공 되고 있으니 에러 코드를 잘 활용 하시오

에러코드 (Error Code)	Description
AXT_RT_SUCCESS(0000)	API 함수 수행 성공
AXT_RT_OPEN_ERROR (1001)	라이브러리 오픈 되지않음
AXT_RT_OPEN_ALREADY (1002)	라이브러리 오픈 되어있거나 현재 기능이 사용 중임
AXT_RT_NOT_OPEN (1053)	라이브러리 초기화 실패
AXT_RT_NOT_SUPPORT_VERSION(1054)	지원하지않는 하드웨어
AXT_RT_INVALID_NODE_NO (1101)	유효하지않는 노드 번호
AXT_RT_INVALID_MODULE_POS (1102)	유효하지않는 모듈 위치
AXT_RT_INVALID_LEVEL (1103)	유효하지않는 레벨
AXT_RT_FLASH_BUSY(1150)	플래시가 Busy 상태 임
AXT_RT_ERROR_VERSION_READ(1151)	라이브러리 버전을 읽을수 없음
AXT_RT_ERROR_NETWORK(1152)	네트워크 에러
AXT_RT_ERROR_HW_ACCESS(1153)	하드웨어 에러
AXT_RT_ERROR_NETWORK_CHEKSUM(1154)	네트워크 CheckSum 에러
AXT_RT_1ST_BELOW_MIN_VALUE (1160)	첫번째 인자값이 최소값보다 더 작음
AXT_RT_1ST_ABOVE_MAX_VALUE (1161)	첫번째 인자값이 최대값보다 더 큼
AXT_RT_2ND_BELOW_MIN_VALUE (1170)	두번째 인자값이 최소값보다 더 작음
AXT_RT_2ND_ABOVE_MAX_VALUE (1171)	두번째 인자값이 최대값보다 더 큼
AXT_RT_3RD_BELOW_MIN_VALUE (1180)	세번째 인자값이 최소값보다 더 작음
AXT_RT_3RD_ABOVE_MAX_VALUE (1181)	세번째 인자값이 최대값보다 더 큼
AXT_RT_4TH_BELOW_MIN_VALUE (1190)	네번째 인자값이 최소값보다 더 작음
AXT_RT_4TH_ABOVE_MAX_VALUE (1191)	네번째 인자값이 최대값보다 더 큼
AXT_RT_5TH_BELOW_MIN_VALUE (1200)	다섯번째 인자값이 최소값보다 더 작음
AXT_RT_5TH_ABOVE_MAX_VALUE (1201)	다섯번째 인자값이 최대값보다 더 큼
AXT_RT_6TH_BELOW_MIN_VALUE (1210)	여섯번째 인자값이 최소값보다 더 작음
AXT_RT_6TH_ABOVE_MAX_VALUE (1211)	여섯번째 인자값이 최대값보다 더 큼
AXT_RT_7TH_BELOW_MIN_VALUE (1220)	일곱번째 인자값이 최소값보다 더 작음
AXT_RT_7TH_ABOVE_MAX_VALUE (1221)	일곱번째 인자값이 최대값보다 더 큼

AXT_RT_8TH_BELOW_MIN_VALUE (1230)	여덟번째 인자값이 최소값보다 더 작음
AXT_RT_8TH_ABOVE_MAX_VALUE (1231)	여덟번째 인자값이 최대값보다 더 큼
AXT_RT_9TH_BELOW_MIN_VALUE (1240)	아홉번째 인자값이 최소값보다 더 작음
AXT_RT_9TH_ABOVE_MAX_VALUE (1241)	아홉번째 인자값이 최대값보다 더 큼
AXT_RT_10TH_BELOW_MIN_VALUE (1250)	열번째 인자값이 최소값보다 더 작음
AXT_RT_10TH_ABOVE_MAX_VALUE (1251)	열번째 인자값이 최대값보다 더 큼
AXT_RT_AIO_OPEN_ERROR (2001)	AIO 모듈 오픈실패
AXT_RT_AIO_NOT_MODULE (2051)	AIO 모듈 없음
AXT_RT_AIO_INVALID_MODULE_NO (2101)	유효하지않은 AIO모듈
AXT_RT_AIO_INVALID_CHANNEL_NO (2102)	유효하지않은 AIO채널번호
AXT_RT_AIO_INVALID_USE (2106)	AIO 함수 사용못함
AXT_RT_AIO_INVALID_TRIGGER_MODE (2107)	유효하지않는 트리거 모드
AXT_RT_DIO_OPEN_ERROR (3001)	DIO 모듈 오픈실패
AXT_RT_DIO_NOT_MODULE (3051)	DIO 모듈 없음
AXT_RT_DIO_INVALID_MODULE_NO (3101)	유효하지않는 DIO 모듈 번호
AXT_RT_DIO_INVALID_OFFSET_NO (3102)	유효하지않는 DIO OFFSET 번호
AXT_RT_DIO_INVALID_LEVEL (3103)	유효하지않는 DIO 레벨
AXT_RT_DIO_INVALID_MODE (3104)	유효하지않는 DIO 모드
AXT_RT_DIO_INVALID_VALUE (3105)	유효하지않는 값 설정
AXT_RT_DIO_INVALID_USE (3106)	DIO 함수 사용못함
AXT_RT_MOTION_OPEN_ERROR(4001)	모션 라이브러리 Open 실패
AXT_RT_MOTION_NOT_MODULE(4051)	시스템에 장착된 모션 모듈이 없음
AXT_RT_MOTION_NOT_INITIAL_AXIS_NO(4053)	해당 축 모션 초기화 실패
AXT_RT_MOTION_NOT_PARA_READ(4055)	원점 구동 설정 파라미터 로드 실패
AXT_RT_MOTION_INVALID_AXIS_NO(4101)	해당 축이 존재하지 않음
AXT_RT_MOTION_INVALID_METHOD(4102)	해당 축 구동에 필요한 설정이 잘못됨
AXT_RT_MOTION_INVALID_USE(4103)	'uUse' 인자값이 잘못 설정됨
AXT_RT_MOTION_INVALID_LEVEL(4104)	'uLevel' 인자값이 잘못 설정됨
AXT_RT_MOTION_INVALID_BIT_NO(4105)	범용 입출력 해당 비트가 잘못 설정됨
AXT_RT_MOTION_INVALID_STOP_MODE(4106)	모션 정지 모드 설정값이 잘못됨
AXT_RT_MOTION_INVALID_TRIGGER_MODE(4107)	트리거 설정 모드가 잘못 설정됨
AXT_RT_MOTION_INVALID_TRIGGER_LEVEL(4108)	트리거 출력 레벨 설정이 잘못됨
AXT_RT_MOTION_INVALID_SELECTION(4109)	'uSelection' 인자가 COMMAND 또는 ACTUAL 이 외의 값으로 설정되어 있음
AXT_RT_MOTION_INVALID_TIME(4110)	Trigger 출력 시간값이 잘못 설정되어 있음
AXT_RT_MOTION_INVALID_FILE_LOAD(4111)	모션 설정값이 저장된 파일이 로드가 안됨

AXT_RT_MOTION_INVALID_FILE_SAVE(4112)	모션 설정값을 저장하는 파일 저장에 실패함
AXT_RT_MOTION_INVALID_VELOCITY(4113)	모션 구동 속도값이 0으로 설정되어 모션 에러 발생
AXT_RT_MOTION_INVALID_ACCELTIME(4114)	모션 구동 가속 시간값이 0으로 설정되어 모션 에러 발생
AXT_RT_MOTION_INVALID_PULSE_VALUE(4115)	모션 단위 설정 시 입력 펄스값이 0보다 작은값으로 설정됨
AXT_RT_MOTION_INVALID_NODE_NUMBER (4116)	위치나 속도 오버라이드 함수가 모션 정지 중에 실행됨
AXT_RT_MOTION_INVALID_TARGET(4117)	다축 모션 정지 원인에 관한 플래그를 반환한다.
AXT_RT_MOTION_ERROR_IN_NONMOTION(4151)	모션 구동중이어야 되는데 모션 구동중이 아닐 때
AXT_RT_MOTION_ERROR_IN_MOTION(4152)	모션 구동 중에 다른 모션 구동 함수를 실행함
AXT_RT_MOTION_ERROR(4153)	다축 구동 정지 함수 실행 중 에러 발생함
AXT_RT_MOTION_ERROR_GANTRY_ENABLE(4154)	겐트리 enable이 되어있어 모션중일 때 또 겐트리 enable을 눌렀을 때
AXT_RT_MOTION_ERROR_GANTRY_AXIS(4155)	겐트리 축이 마스터채널(축) 번호(0 ~ (최대축수 - 1))가 잘못 들어갔을 때
AXT_RT_MOTION_ERROR_MASTER_SERVOON(4156)	마스터 축 서보온이 안되어있을 때
AXT_RT_MOTION_ERROR_SLAVE_SERVOON(4157)	슬레이브 축 서보온이 안되어있을 때
AXT_RT_MOTION_INVALID_POSITION(4158)	유효한 위치에 없을 때
AXT_RT_ERROR_NOT_SAME_MODULE (4159)	똑 같은 모듈내에 있지 않을경우
AXT_RT_ERROR_NOT_SAME_PRODUCT (4161)	제품이 서로 다를경우
AXT_RT_NOT_CAPTURED(4162)	위치가 저장되지 않을 때
AXT_RT_ERROR_NOT_SAME_IC(4163)	같은 칩내에 존재하지않을 때
AXT_RT_ERROR_NOT_GEARMODE(4164)	기어모드로 변환이 안될 때
AXT_ERROR_CONTI_INVALID_AXIS_NO(4165)	보간 축매핑할 때 유효한축이 아닐 때
AXT_ERROR_CONTI_INVALID_MAP_NO(4166)	보간 맵핑 시 유효한 맵핑 번호가 아닐 때
AXT_ERROR_CONTI_EMPTY_MAP_NO(4167)	보간 맵핑 번호가 비워있을 때
AXT_RT_MOTION_ERROR_CACULATION(4168)	계산상의 오차가 발생했을 때
AXT_RT_ERROR_NOT_SAME_NODE(4169)	노드 번호가 틀릴 때
AXT_RT_MOTION_INTERPOL_VALUE(4184)	보간할 때 입력 값이 잘못넣어졌을 때
AXT_RT_MOTION_HOME_SEARCHING(4201)	홈을 찾고 있는 중일 때 다른 모션 함수들을 사용할 때
AXT_RT_MOTION_HOME_ERROR_SEARCHING(4202)	홈을 찾고 있는 중일 때 외부에서 사용자나 혹은 어떤것에 의한 강제로 정지당할 때
AXT_RT_MOTION_HOME_ERROR_START(4203)	초기화 문제로 홈시작 불가할 때
AXT_RT_MOTION_HOME_ERROR_GANTRY(4204)	홈을 찾고 있는 중일 때 겐트리 enable 불가할 때

AXT_RT_MOTION_POS_OUTOFBOUND(4251)	설정된 위치값이 설정 최대값보다 크거나 최소값보다 작은값임
AXT_RT_MOTION_PROFILE_INVALID(4252)	구동 속도 프로파일 설정이 잘못됨
AXT_RT_MOTION_VELOCITY_OUTOFBOUND(4253)	구동 속도값이 최대값보다 크게 설정됨
AXT_RT_MOTION_MOVE_UNIT_IS_ZERO(4254)	구동 단위값이 0으로 설정됨
AXT_RT_MOTION_SETTING_ERROR(4255)	속도, 가속도, 저크, 프로파일 설정이 잘못됨
AXT_RT_MOTION_DISABLE_TRIGGER(4257)	트리거 출력이 Disable 상태임
AXT_RT_INIT_DOSE_NOT_EXIST_LAN_CARD(4301)	랜카드가 존재하지 않을 때
AXT_RT_INIT_DOES_NOT_RESPONSE_SLAVE(4302)	슬레이브에서의 응답이 없을 때
AXT_RT_INIT_INVALID_HOST_IP_ADDRESS(4303)	마스터의 IP 주소가 틀릴 때
AXT_RT_INIT_INVALID_HOST_COUNT(4304)	랜카드 개수가 틀릴 때
AXT_RT_INIT_EIPSTART_FAIL(4305)	Ethernet/IP 통신이 초기화 되지 않을 때
AXT_RT_INIT_ALREADY_INITIALIZED(4306)	초기화가 이미 되어 있을 때
AXT_RT_INIT_INVALID_NET_TYPE(4307)	Network 타입이 틀릴 때
AXT_RT_INIT_PRODUCTID (4308)	모듈 ID가 존재하지 않을 때
AXT_RT_INIT_NOT_ENOUGH_MEMORY(4309)	메모리가 충분하지 않을 때
AXT_RT_INIT_DOES_NOT_RESPONSE_SLAVE_AT_RING(4310)	링 오픈시 응답이 없을 때
AXT_RT_INIT_NETWORK_ERROR(4311)	네트워크 초기화 에러
AXT_RT_INIT_BACKGROUND_NOT_STARTED(4312)	BackGround Process 에러
AXT_RT_INIT_BACKGROUND_START_FAIL(4313)	BackGround Process 초기화 실패

용어 설명 :**Motion :**

각종 스텝 모터, 서보 모터등의 제어 기능을 제공하는 아진엑스텍의 모션 제어 모듈을 통칭한다.

DIO :

각종 센서 접속 기능을 제공하는 아진엑스텍의 디지털 입출력 제어 모듈을 통칭한다.

AIO :

각종 센서 접속 기능을 제공하는 아진엑스텍의 아날로그 입출력 제어 모듈을 통칭한다.

이 설명서의 내용은 예고 없이 변경될 수 있습니다. 용례에 사용된 회사, 기관, 제품, 인물 및 사건 등은 실제 데이터가 아닙니다. 어떠한 실제 회사, 기관, 제품, 인물 또는 사건과도 연관시킬 의도가 없으며 그렇게 유추해서도 안됩니다. 해당 저작권법을 준수하는 것은 사용자의 책임입니다. 저작권에서의 권리와는 별도로, 이 설명서의 어떠한 부분도 (주)아진엑스텍의 명시적인 서면 승인 없이는 어떠한 형식이나 수단(전기적, 기계적, 복사기에 의한 복사, 디스크 복사 또는 다른 방법) 또는 다른 목적으로도 복제되거나, 검색 시스템에 저장 또는 도입되거나, 전송될 수 없습니다.

(주)아진엑스텍은 이 설명서 본 안에 관련된 특허권, 상표권, 저작권 또는 기타 지적 소유권 등을 보유할 수 있습니다. 서면 사용권 계약에 따라 (주)아진엑스텍으로부터 귀하에게 명시적으로 제공된 권리 이외에, 이 설명서의 제공은 귀하에게 이러한 특허권, 저작권 또는 기타 지적 소유권 등에 대한 어떠한 사용권도 허용하지 않습니다.