

AnyDIO Series
AXD

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

Contents

매뉴얼 정보	6
헤더 파일	6
함수 용어	6
본 매뉴얼의 함수 이름	6
본 매뉴얼의 인자 이름	7
Quick List	9
노드 및 모듈	9
입출력 포트	9
Function List	14
노드 및 모듈	14
AxdInfoIsDIOModule	15
AxdInfoGetModuleCount	17
AxdInfoGetInputCount	19
AxdInfoGetOutputCount	21
AxdInfoGetModule	23
입 출력 포트	25
AxdiLevelSetInportBit	29
AxdiLevelSetInportByte	31
AxdiLevelSetInportWord	33
AxdiLevelSetInportDword	35
AxdiLevelSetInport	37
AxdiLevelGetInportBit	38
AxdiLevelGetInportByte	40
AxdiLevelGetInportWord	42
AxdiLevelGetInport	44
AxdoLevelSetOutportBit	46
AxdoLevelSetOutportByte	48
AxdoLevelSetOutportWord	50
AxdoLevelSetOutportDWord	52
AxdoLevelSetOutport	54
AxdoLevelGetOutportBit	55
AxdoLevelGetOutportByte	57
AxdoLevelGetOutportWord	59

AxdoLevelGetOutput	61
AxdoWriteOutputBit	63
AxdoWriteOutputByte	65
AxdoWriteOutputWord	67
AxdoWriteOutput	69
AxdoReadOutputBit.....	70
AxdoReadOutputByte.....	72
AxdoReadOutputWord	74
AxdoReadOutput	76
AxdiReadInportBit.....	78
AxdiReadInportByte.....	80
AxdiReadInportWord	82
AxdiReadInport	84
AxdiNLevelSetInportBit	86
AxdiNLevelSetInportByte	88
AxdiNLevelSetInportWord.....	90
AxdiNLevelSetInportDword.....	92
AxdiNLevelSetInport.....	94
AxdiNLevelGetInportBit.....	96
AxdiNLevelGetInportByte.....	98
AxdiNLevelGetInportWord	100
AxdiNLevelGetInport.....	102
AxdoNLevelSetOutputBit	104
AxdoNLevelSetOutputByte	106
AxdoNLevelSetOutputWord.....	108
AxdoNLevelSetOutputDword.....	110
AxdoNLevelSetOutput.....	112
AxdoNLevelGetOutputBit.....	114
AxdoNLevelGetOutputByte.....	116
AxdoNLevelGetOutputWord	118
AxdoNLevelGetOutput	120
AxdoNWriteOutputBit.....	122
AxdoNWriteOutputByte.....	124
AxdoNWriteOutputWord	126
AxdoNWriteOutputDword	128
AxdoNWriteOutput.....	130
AxdoNReadOutputBit	132

AxdoNReadOutputByte	134
AxdoNReadOutputWord.....	136
AxdoNReadOutputDword.....	138
AxdoNReadOutput.....	140
AxdiNReadInportBit	142
AxdiNReadInportByte	144
AxdiNReadInportWord.....	146
AxdiNReadInport.....	148
AxdoNetWorkErrorSetAction	150
AxdoNetWorkErrorGetAction.....	152
AxdoNetWorkErrorSetByteValue	154
AxdoNetWorkErrorGetByteValue	156
에러코드 테이블(Error Code Table) 확인	158

Revision History

Manual	PCB	Comments
Rev. 1.0 issue 1.0	Rev. 1.0	2008.03.27.

매뉴얼 정보

본 매뉴얼은 EIP-CPU16B, EIP-DB32T 모듈을 Windows 98, Windows NT, Windows 2000 또는 Windows XP 의 OS 환경에서 Microsoft VC++6.0, Visual Basic, Borland C-Builder, Delphi 등의 언어에서 구동하기 위해 필요한 매뉴얼이며, 포함된 라이브러리 함수를 기능별로 분류하여 설명하였다.

헤더 파일

C

AXD.h

Visual Basic

AXD.bas

Delphi

AXD.pas

함수 용어

본 매뉴얼의 함수 이름

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

라이브러리 함수 Prefix

Axd : Axd 전용 함수임을 나타낸다. Axd 로 시작되는 함수들은 모두 Axd.h 에 정의 되어있다.

Set : 칩의 레지스터나 변수들을 설정한다.

Get : Set 과 한 쌍으로 Set 함수에 의해 설정된 변수 값을 확인하거나 칩 레지스터의 상태를 읽는다.

Read : 입출력 접점의 특정 레지스터에서 상태를 읽는다.

Write : 입출력 접점의 특정 레지스터에서 상태를 설정한다.

Port : 입력 및 출력 접점과 관련된 함수 군이다.

Bit : 1 개의 접점 단위를 읽거나 쓴다.

Byte : 8 개의 접점 단위를 읽거나 쓴다.

Word : 16 개의 점점 단위를 읽거나 쓴다.

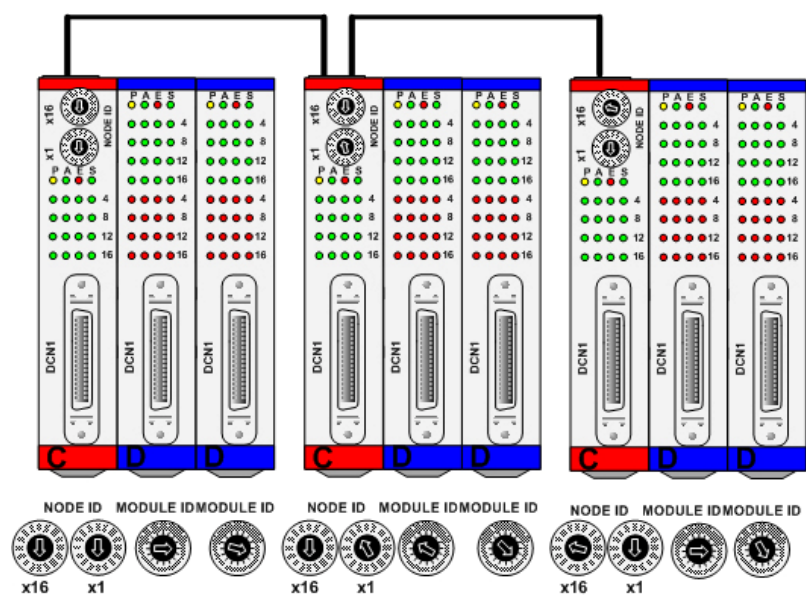
Dword : 32 개의 점점 단위를 읽거나 쓴다.

본 매뉴얼의 인자 이름

본 매뉴얼에서 사용된 함수들의 공통적인 인자들은 다음과 같은 의미를 가진다.

long lNodeNum: 초기화 된 노드들의 전면 NODE ID 설정 값.

long lModuleNo : DIO 모듈 초기화 시에 첫 번째 노드의 모듈부터 오름차순으로 자동 정렬된다. 모듈 번호는 '0'부터 시작한다.



	CPU	DIO	DIO	CPU	DIO	DIO	CPU	DIO	DIO
Node ID	0			7			6		
Module ID	0	1	2	0	10	4	0	1	4
Module No	0	1	2	3	5	4	6	7	8

long lModulePos : DIO 모듈상단의 MODULE ID 설정 값. CPU 모듈은 이 값이 0 이다. MODULE ID SW 값의 범위가 (1 ~16) 이므로 lModulePos 의 범위는 (0~16)이다.

DWORD uLevel : uLevel 은 ReadPort 및 WritePort 관련 함수들을 사용시 Level 을 설정 및 확인한다.

DWORD uOffset : uOffset 은 사용 모듈과 사용 함수에 따라 유효 범위가 달라지는데 아래의 Table 을 참조하여 설정한다.

EIP-DB32T(EIP-CPU16B) Offset 범위

Data Type	Offset	Value or Return	Related
All	0 ~ n-1	0(Off), 1(On)	
Bit	0 ~ 31	0(Off), 1(On)	Bit 계열 함수 군
Byte	0, 1, 2, 3	00h ~ FFh	Byte 계열 함수 군
Word	0, 1	0000h ~ FFFFh	Word 계열 함수 군
Dword	0	00000000h ~ FFFFFFFFh	Dword 계열 함수 군

Quick List

노드 및 모듈

노드 및 모듈 정보

Function	Description
AxdInfosDIOModule	DI0 모듈이 있는지 확인한다.
AxdInfoGetModuleCount	DI0 입출력 모듈의 개수를 확인한다.
AxdInfoGetInputCount	지정한 모듈의 입력 접점 개수를 확인한다.
AxdInfoGetOutputCount	지정한 모듈의 출력 접점 개수를 확인한다.
AxdInfoGetModule	지정한 모듈 번호로 베이스 노드 번호, 모듈 위치, 모듈 ID를 확인한다.

입출력 포트

입출력 레벨 설정 확인

Function	Description
AxdLevelSetInportBit	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다
AxdLevelSetInportByte	지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다
AxdLevelSetInportWord	지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.
AxdLevelSetInportDword	지정한 입력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
AxdLevelSetInport	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdLevelGetInportBit	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
AxdLevelGetInportByte	지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.
AxdLevelGetInportWord	지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을

	확인한다.
AxdiLevelGetInport	전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
AxdoLevelSetOutputBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputDword	지정한 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutput	전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelGetOutputBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
AxdoLevelGetOutputByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.
AxdoLevelGetOutputWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.
AxdoLevelGetOutput	전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

입출력 포트 쓰기 읽기

Function	Description
AxdoWriteOutputBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다
AxdoWriteOutputByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.
AxdoWriteOutputWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 출력한다.
AxdoWriteOutput	전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.
AxdoReadOutputBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

AxdoReadOutputByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.
AxdoReadOutputWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.
AxdoReadOutput	전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
AxdireadInputBit	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
AxdireadInputByte	지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.
AxdireadInputWord	지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.
AxdireadInput	전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

지정 노드에 대한 입력 레벨 확인

Function	Description
AxdinLevelSetInputBit	해당 노드의 지정한 입력 모듈에 bit단위로 Offset위치에 데이터 레벨을 설정한다
AxdinLevelSetInputByte	해당 노드의 지정한 입력 모듈에 byte단위로 Offset위치에 데이터 레벨을 설정한다
AxdinLevelSetInputWord	해당 노드의 지정한 입력 모듈에 word단위로 Offset위치에 데이터 레벨을 설정한다
AxdinLevelSetInputDword	해당 노드의 지정한 입력 모듈에 double word 단위로 Offset위치에 데이터 레벨을 설정한다
AxdinLevelSetInput	해당 노드의 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정 한다
AxdinLevelGetInputBit	해당 노드의 지정한 입력 모듈에 bit단위로 Offset위치에 데이터 레벨을 확인한다
AxdinLevelGetInputByte	해당 노드의 지정한 입력 모듈에 byte단위로 Offset위치에 데이터 레벨을 확인한다
AxdinLevelGetInputWord	해당 노드의 지정한 입력 모듈에 word단위로 Offset위치에 데이터 레벨을 확인한다
AxdinLevelGetInput	해당 노드의 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인 한다
AxdoNLevelSetOutputBit	해당 노드의 지정한 출력 모듈에 bit단위로 Offset위치에 데이터 레벨

	을 설정한다
AxdoNLevelSetOutputByte	해당 노드의 지정한 출력 모듈에 byte단위로 Offset위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutputWord	해당 노드의 지정한 출력 모듈에 word단위로 Offset위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutputDword	해당 노드의 지정한 출력 모듈에 double word 단위로 Offset위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutput	해당 노드의 전체 출력 모듈에 bit단위로 Offset위치에 데이터 레벨을 설정한다
AxdoNLevelGetOutputBit	해당 노드의 지정한 출력 모듈에 bit단위로 Offset위치에 데이터 레벨을 확인한다
AxdoNLevelGetOutputByte	해당 노드의 지정한 출력 모듈에 byte단위로 Offset위치에 데이터 레벨을 확인한다
AxdoNLevelGetOutputWord	해당 노드의 지정한 출력 모듈에 word단위로 Offset위치에 데이터 레벨을 확인한다
AxdoNLevelGetOutput	해당 노드의 전체 출력 점점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인 한다

지정 노드에 대한 출력 포트 쓰기 읽기

Function	Description
AxdoNWriteOutputBit	해당 노드의 지정한 출력 모듈에 bit단위로 Offset위치에 출력한다.
AxdoNWriteOutputByte	해당 노드의 지정한 출력 모듈에 byte단위로 Offset위치에 출력한다.
AxdoNWriteOutputWord	해당 노드의 지정한 출력 모듈에 word단위로 Offset위치에 출력한다.
AxdoNWriteOutputDword	해당 노드의 지정한 출력 모듈에 double word단위로 Offset위치에 출력한다.
AxdoNWriteOutput	해당 노드의 전체 출력 모듈에 bit단위로 Offset위치에 출력한다.
AxdoNReadOutputBit	해당 노드의 지정한 출력 모듈에 bit단위로 Offset위치에 읽는다.
AxdoNReadOutputByte	해당 노드의 지정한 출력 모듈에 byte단위로 Offset위치에 읽는다.
AxdoNReadOutputWord	해당 노드의 지정한 출력 모듈에 word단위로 Offset위치에 읽는다.
AxdoNReadOutputDword	해당 노드의 지정한 출력 모듈에 double word단위로 Offset위치에 읽는다.
AxdoNReadOutput	해당 노드의 전체 출력 모듈에 bit단위로 Offset위치에 읽는다.
AxdiNReadInportBit	해당 노드의 지정한 입력 모듈에 bit단위로 Offset위치에 읽는다.

AxdINReadInportByte	해당 노드의 지정한 입력 모듈에 byte단위로Offset위치에 읽는다.
AxdINReadInportWord	해당 노드의 지정한 입력 모듈에 word단위로Offset위치에 읽는다.
AxdINReadInport	해당 노드의 전체 입력 모듈에 bit단위로 Offset위치에 읽는다.

출력 포트의 네트워크 예외 처리

Function	Description
AxdoNetWorkErrorSetAction	지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어 졌을때 현재 출력 접점 상태에 대한 예외처리 방법 설정한다.
AxdoNetWorkErrorGetAction	지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어 졌을때 현재 출력 접점 상태에 대한 예외처리 방법 확인한다.
AxdoNetWorkErrorSetByteValue	Network연결이 끊어 졌을 경우 지정한 출력 접점의 상태를 설정 한다
AxdoNetWorkErrorGetByteValue	Network연결이 끊어 졌을 경우 지정한 출력 접점의 상태를 확인 한다

Function List

노드 및 모듈

노드 및 모듈 정보

Function	Description
AxdInfoIsDIOModule	DIO 모듈이 있는지 확인한다.
AxdInfoGetModuleCount	DIO 입출력 모듈의 개수를 확인한다.
AxdInfoGetInputCount	지정한 모듈의 입력 접점 개수를 확인한다.
AxdInfoGetOutputCount	지정한 모듈의 출력 접점 개수를 확인한다.
AxdInfoGetModule	지정한 모듈 번호로 노드 번호, 모듈 위치, 모듈 ID를 확인한다.

AxdInfoIsDIOModule

Purpose

DIO 모듈이 있는지 확인한다.

Format

C

```
DWORD AxdInfoIsDIOModule(DWORD *upStatus);
```

Visual Basic

```
Function AxdInfoIsDIOModule(ByRef upStatus As Long) As Long
```

Delphi

```
function AxdInfoIsDIOModule(upStatus : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
upStatus	out	DWORD*	–	STATUS_NOTEXIST(0) : DIO모듈이 존재하지 않음 STATUS_EXIST(1) : DIO모듈이 존재함

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

네트워크로 연결된 DIO 모듈의 존재 여부를 확인한다.

C Example

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

DWORD dwStatus;

// DIO 모듈이 있는지 확인한다.
AxdInfoIsDIOModule(&dwStatus);
if(dwStatus == STATUS_EXIST)
    AfxMessageBox("디지털 입출력 모듈들이 존재합니다.");
else
    AfxMessageBox("디지털 입출력모듈들이 존재하지 않습니다.");
```

VB Example

```

' 라이브러리 사용을 위해 AXD.bas와 AXL.bas, AXHS.bas 모듈을 추가 한다.
Dim lStatus As Long

' DIO 모듈이 있는지 확인한다.
AxdInfoIsDIOModule lStatus
If lStatus = STATUS_EXIST Then
    MsgBox "디지털 입출력모듈들이 존재합니다.", vbOKCancel
Else
    MsgBox "디지털 입출력모듈들이 존재하지 않습니다.", vbOKCancel
End If

```

Delphi Example

```

Uses
AXL, AXHS, AXD           { 라이브러리 추가 }
var
dwStatus : DWord;

{ DIO 모듈이 있는지 확인한다. }
begin
AxdInfoIsDIOModule(@dwStatus);
if(dwStatus = STATUS_EXIST) then
    Application.MessageBox ( '디지털 입출력모듈들이 존재합니다.' ,
        'Ajinextek' , MB_OK)
    else
    Application.MessageBox ( '디지털 입출력모듈들이 존재하지 않습니다.' ,
        'Ajinextek' , MB_OK);
end;

```

See Also

[AxdInfoGetModuleCount](#), [AxdInfoGetInputCount](#), [AxdInfoGetOutputCount](#), [AxdInfoGetModule](#)

AxdInfoGetModuleCount

Purpose

DIO 입출력 모듈의 개수를 확인한다.

Format

C

```
DWORD AxdInfoGetModuleCount(long *lpModuleCounts);
```

Visual Basic

```
Function AxdInfoGetModuleCount(ByRef lpModuleCounts As Long) As Long
```

Delphi

```
function AxdInfoGetModuleCount(lpModuleCounts : PLongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
lpModuleCounts	out	long	–	DIO 입출력 모듈의 개수

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

초기화 되어 있는 DIO 입출력 모듈의 개수를 확인한다.

C Example

```
// DIO 모듈의 개수를 확인한다.
long          lCount;
CString       strData;

AxdInfoGetModuleCount (&lCount);
strData.Format( "DIO 모듈의 개수는 %d개 입니다." , lCount);

AfxMessageBox(strData);
```

VB Example

```
' DIO 모듈의 개수를 확인한다.
Dim lCount As Long
Dim strData As String

AxdInfoGetModuleCount lCount
strData = "DIO 모듈의 개수는 " + CStr(lCount) + " 개 입니다."

MsgBox strData
End If
```

Delphi Example

```
{ DIO 모듈의 개수를 확인한다. }  
var  
  lCount : LongInt;  
  strData : String;  
  
begin  
  AxdInfoGetModuleCount (@lCount);  
  strData      := 'DIO 모듈의 개수는 ' + IntToStr(lCount) + '개 입니다.' ;  
  
  Application.MessageBox (PCHAR(strData), 'Ajinextek', MB_OK);  
end;
```

See Also

[AxdInfoIsDIOModule](#), [AxdInfoGetInputCount](#), [AxdInfoGetOutputCount](#), [AxdInfoGetModule](#)

AxdInfoGetInputCount

Purpose

지정한 모듈의 입력 접점 개수를 확인한다.

Format

C

```
DWORD AxdInfoGetInputCount(long IModuleNo, long *IpCounts);
```

Visual Basic

```
Function AxdInfoGetInputCount(ByVal IModuleNo As Long, ByRef IpCounts As Long) As Long
```

Delphi

```
function AxdInfoGetInputCount(IModuleNo : LongInt; IpCounts : PLongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IpCounts	out	long		입력 접점 개수

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 모듈의 입력 접점 개수를 확인한다.

Module	Input	Output
EIP-CPU16B	16	16
EIP-DB32T	16	16

Note) EIP- CPU16B접점 수는 16개 이하 모든 접점이 입력 또는 출력으로 사용될 수 있기 때문에 입력/출력 접점 개수를 라이브러리 에서는 모듈당 각각 16개로 인식한다. 즉 실제로 16 접점을 입력과 출력으로 나누어서 사용되며 입력으로만 사용시 최대 16개, 출력으로만 사용시 최대 16개로 사용될 수 있는 것이다.

C Example

```
// 0번째 모듈의 입력 접점 개수를 확인한다.
long          lCount;
CString       strData;

AxdInfoGetInputCount (0, &lCount);
strData.Format( "0번 모듈의 입력 접점 개수는 %d개 입니다." , lCount );

AfxMessageBox( strData );
```

VB Example

```
' 0번째 모듈의 입력 점점 개수를 확인한다.
Dim lCount As Long
Dim strData As String

AxdInfoGetInputCount 0, lCount
strData = "0번 모듈의 입력 점점 개수는 " + CStr(lCount) + " 개 입니다."

MsgBox strData
```

Delphi Example

```
{ 0번째 모듈의 입력 점점 개수를 확인한다. }
var
lCount : LongInt;
strData : String;

begin
AxdInfoGetInputCount (@lCount);
strData := '0번 모듈의 입력 점점 개수는 ' + IntToStr(lCount) + '개 입니
다.' ;

Application.MessageBox (PCHAR(strData), 'Ajinextek', MB_OK);
end;
```

See Also

[AxdInfoIsDIOModule](#), [AxdInfoGetModuleCount](#), [AxdInfoGetOutputCount](#), [AxdInfoGetModule](#)

AxdInfoGetOutputCount

Purpose

지정한 모듈의 출력 접점 개수를 확인한다.

Format

C

```
DWORD AxdInfoGetOutputCount(long IModuleNo, long *lpCounts);
```

Visual Basic

```
Function AxdInfoGetOutputCount(ByVal IModuleNo As Long, ByRef lpCounts As Long) As Long
```

Delphi

```
function AxdInfoGetOutputCount(IModuleNo : LongInt; lpCounts : PLongInt) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
lpCounts	out	long		출력 접점 개수

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 모듈의 출력 접점 개수를 확인한다.

Module	Input	Output
EIP-CPU16B	16	16
EIP-DB32T	16	16

Note) EIP- CPU16B접점 수는 16개 이하 모든 접점이 입력 또는 출력으로 사용될 수 있기 때문에 입력/출력 접점 개수를 라이브러리 에서는 모듈당 각각 16개로 인식한다. 즉 실제로 16 접점을 입력과 출력으로 나누어서 사용되며 입력으로만 사용시 최대 16개, 출력으로만 사용시 최대 16개로 사용될 수 있는 것이다.

C Example

```
// 0번째 모듈의 출력 접점 개수를 확인한다.
long          lCount;
CString       strData;

AxdInfoGetOutputCount (0, &lCount);
strData.Format( "0번 모듈의 출력 접점 개수는 %d개 입니다." , lCount );

AfxMessageBox( strData );
```

VB Example

```
' 0번째 모듈의 출력 점점 개수를 확인한다.
Dim lCount As Long
Dim strData As String

AxdInfoGetOutputCount 0, lCount
strData = "0번 모듈의 출력 점점 개수는 " + CStr(lCount) + " 개 입니다."

MsgBox strData
```

Delphi Example

```
{ 0번째 모듈의 출력 점점 개수를 확인한다. }
var
lCount : LongInt;
strData : String;

begin
AxdInfoGetOutputCount (@lCount);
strData := '0번 모듈의 출력 점점 개수는 ' + IntToStr(lCount) + '개 입니
다.' ;

Application.MessageBox (PCHAR(strData), 'Ajinextek', MB_OK);
end;
```

See Also

[AxdInfoIsDIOModule](#), [AxdInfoGetModuleCount](#), [AxdInfoGetInputCount](#), [AxdInfoGetModule](#)

AxdInfoGetModule

Purpose

지정한 모듈 번호로 노드 번호, 모듈 위치, 모듈 ID를 확인한다.

Format

C

```
DWORD AxdInfoGetModule (long IModuleNo, long *lpNodeNum, long *lpModulePos, DWORD *upModuleID);
```

Visual Basic

```
Function AxdInfoGetModule (ByVal IModuleNo As Long, ByRef lpNodeNum As Long, ByRef lpModulePos As Long, ByRef upModuleID As Long) As Long
```

Delphi

```
function AxdInfoGetModule (IModuleNo : LongInt; lpNodeNum: PLongInt; lpModulePos : PLongInt; upModuleID : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
lpNodeNum	out	Long*		노드 번호 (0 ~ n-1)
lpModulePos	out	Long*		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
upModuleID	out	DWORD*		모듈 ID

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

Description

사용자가 지정한 모듈 번호로 노드 번호, 모듈 위치, 모듈 ID를 확인한다.

Module	Input
EIP-CPU16B	0xB3
EIP-DB32T	0XB7

C Example

```
// 0번째 모듈의 노드 번호, 모듈 번호, 모듈 ID를 확인한다.
long lpNodeNum;
long lModulePos;
DWORD dwModuleID;
CString strData;

AxdInfoGetModule (0, &lpNodeNum, &lModulePos, &dwModuleID);
```

```
strData.Format("NodeNum=%d, ModulePos=%d, dwModuleID=%02Xh", lpNodeNum,
    lModulePos, dwModuleID);
AfxMessageBox(strData);
```

VB Example

```
' 0번째 모듈의 노드 번호, 모듈 번호, 모듈 ID를 확인한다.
Dim lpNodeNum As Long
Dim lModulePos As Long
Dim lModuleID As Long
Dim strData As String

AxdInfoGetModule 0, lpNodeNum, lModulePos, dwModuleID
strData = "NodeNum=" + CStr(lpNodeNum) + ", ModulePos=" + CStr(lModulePos)
    + ", ModuleID=" + CStr(dwModuleID)
MsgBox strData
```

Delphi Example

```
{ 0번째 모듈의 노드 번호, 모듈 번호, 모듈 ID를 확인한다. }
var
lBoardNo : LongInt;
lpNodeNum : LongInt;
dwModuleID : DWord;
strData : String;

begin
AxdInfoGetModule (0, @lpNodeNum, @lModulePos, @dwModuleID);
strData := 'pNodeNum=' + IntToStr(lpNodeNum) + ', ModulePos=' +
    IntToStr(nModulePos) + ', ModuleID=' + IntToStr(dwModuleID);
Application.MessageBox (PCHAR(strData), 'Ajinextek', MB_OK);
end;
```

See Also

[AxdInfoIsDIOModule](#), [AxdInfoGetModuleCount](#), [AxdInfoGetInputCount](#), [AxdInfoGetOutputCount](#),

입 출력 포트

입출력 레벨 쓰기 읽기

Function	Description
AxdiLevelSetInportBit	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportByte	지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportWord	지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportDword	지정한 입력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInport	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
AxdiLevelGetInportBit	지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.
AxdiLevelGetInportByte	지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.
AxdiLevelGetInportWord	전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdiLevelGetInport	전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
AxdoLevelSetOutputportBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputportByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputportWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputportDword	지정한 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputport	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
AxdoLevelGetOutputportBit	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.
AxdoLevelGetOutputportByte	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.

AxdoLevelGetOutputWord	지정 노드내의 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelGetOutput	지정 노드내의 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

입출력 포트 쓰기 읽기

Function	Description
AxdoWriteOutputBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다
AxdoWriteOutputByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다
AxdoWriteOutputWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 출력한다.
AxdoNWriteOutputDword	지정한 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터를 출력한다
AxdoWriteOutput	전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.
AxdoReadOutputBit	지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
AxdoReadOutputByte	지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.
AxdoReadOutputWord	지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.
AxdoReadOutput	전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다
AxdiReadInportBit	지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다
AxdiReadInportByte	지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다
AxdiReadInportWord	지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.
AxdiReadInport	전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

지정 노드에 대한 입력 레벨 확인

Function	Description
AxdiNLevelSetInportBit	해당 노드의 지정한 입력 모듈에 bit 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdiNLevelSetInportByte	해당 노드의 지정한 입력 모듈에 byte 단위로 Offset 위치에 데이터 레

	벨을 설정한다
AxdinLevelSetInportWord	해당 노드의 지정한 입력 모듈에 word 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdinLevelSetInportDword	해당 노드의 지정한 입력 모듈에 double word 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdinLevelSetInport	해당 노드의 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정 한다
AxdinLevelGetInportBit	해당 노드의 지정한 입력 모듈에 bit 단위로 Offset 위치에 데이터 레벨을 확인한다
AxdinLevelGetInportByte	해당 노드의 지정한 입력 모듈에 byte 단위로 Offset 위치에 데이터 레벨을 확인한다
AxdinLevelGetInportWord	해당 노드의 지정한 입력 모듈에 word 단위로 Offset 위치에 데이터 레벨을 확인한다
AxdinLevelGetInport	해당 노드의 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인 한다
AxdoNLevelSetOutputBit	해당 노드의 지정한 출력 모듈에 bit 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutputByte	해당 노드의 지정한 출력 모듈에 byte 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutputWord	해당 노드의 지정한 출력 모듈에 word 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutputDword	해당 노드의 지정한 출력 모듈에 double word 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdoNLevelSetOutput	해당 노드의 전체 출력 모듈에 bit 단위로 Offset 위치에 데이터 레벨을 설정한다
AxdoNLevelGetOutputBit	해당 노드의 지정한 출력 모듈에 bit 단위로 Offset 위치에 데이터 레벨을 확인한다
AxdoNLevelGetOutputByte	해당 노드의 지정한 출력 모듈에 byte 단위로 Offset 위치에 데이터 레벨을 확인한다
AxdoNLevelGetOutputWord	해당 노드의 지정한 출력 모듈에 word 단위로 Offset 위치에 데이터 레벨을 확인한다
AxdoNLevelGetOutput	해당 노드의 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인 한다

지정 노드에 대한 출력 포트 쓰기 읽기

Function	Description
AxdoNWriteOutputBit	해당 노드의 지정한 출력 모듈에 bit 단위로 Offset 위치에 출력한다.

AxdoNWriteOutputByte	해당 노드의 지정한 출력 모듈에 byte 단위로 Offset 위치에 출력한다.
AxdoNWriteOutputWord	해당 노드의 지정한 출력 모듈에 word 단위로 Offset 위치에 출력한다.
AxdoNWriteOutputDword	해당 노드의 지정한 출력 모듈에 double word 단위로 Offset 위치에 출력한다.
AxdoNWriteOutput	해당 노드의 전체 출력 모듈에 bit 단위로 Offset 위치에 출력한다.
AxdoNReadOutputBit	해당 노드의 지정한 출력 모듈에 bit 단위로 Offset 위치에 읽는다.
AxdoNReadOutputByte	해당 노드의 지정한 출력 모듈에 byte 단위로 Offset 위치에 읽는다.
AxdoNReadOutputWord	해당 노드의 지정한 출력 모듈에 word 단위로 Offset 위치에 읽는다.
AxdoNReadOutputDword	해당 노드의 지정한 출력 모듈에 double word 단위로 Offset 위치에 읽는다.
AxdoNReadOutput	해당 노드의 전체 출력 모듈에 bit 단위로 Offset 위치에 읽는다.
AxdiNReadInportBit	해당 노드의 지정한 입력 모듈에 bit 단위로 Offset 위치에 읽는다.
AxdiNReadInportByte	해당 노드의 지정한 입력 모듈에 byte 단위로 Offset 위치에 읽는다.
AxdiNReadInportWord	해당 노드의 지정한 입력 모듈에 word 단위로 Offset 위치에 읽는다.
AxdiNReadInport	해당 노드의 전체 입력 모듈에 bit 단위로 Offset 위치에 읽는다.

출력 포트의 네트워크 예외 처리

Function	Description
AxdoNetWorkErrorSetAction	지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어졌을 때 현재 출력 접점 상태에 대한 예외 처리 방법 설정한다.
AxdoNetWorkErrorGetAction	지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어졌을 때 현재 출력 접점 상태에 대한 예외 처리 방법 확인한다.
AxdoNetWorkErrorSetByteValue	Network 연결이 끊어졌을 경우 지정한 출력 접점의 상태를 설정한다.
AxdoNetWorkErrorGetByteValue	Network 연결이 끊어졌을 경우 지정한 출력 접점의 상태를 확인한다.

AxdiLevelSetInportBit

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiLevelSetInportBit(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiLevelSetInportBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiLevelSetInportBit(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	In	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportBit(0, 0, 1);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportBit 0, 0, 1
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }  
AxdilLevelSetInportBit(0, 0, 1);
```

See Also

[AxdilLevelSetInportByte](#), [AxdilLevelSetInportWord](#), [AxdilLevelSetInportDword](#), [AxdilLevelSetInport](#)

AxdiLevelSetInportByte

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiLevelSetInportByte(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiLevelSetInportByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiLevelSetInportByte(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	In	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportByte(0, 0, 0xFF);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportByte 0, 0, 0xFF
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다. }  
AxdiLevelSetInportByte (0, 0, 0xFF);
```

See Also

[AxdiLevelSetInportBit](#), [AxdiLevelSetInportWord](#), [AxdiLevelSetInportDword](#), [AxdiLevelSetInport](#)

AxdiLevelSetInportWord

Purpose

전체 DIO 모듈들의 순서 중 지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiLevelSetInportWord(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiLevelSetInportWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiLevelSetInportWord(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	In	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	00h ~ FFFFh

C Example

```
// 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportWord(0, 0, 0xFFFF);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportWord 0, 0, 0xFFFF
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다. }  
AxdiLevelSetInportWord (0, 0, 0xFFFF);
```

See Also

[AxdiLevelSetInportBit](#), [AxdiLevelSetInportByte](#), [AxdiLevelSetInportDword](#), [AxdiLevelSetInport](#)

AxdiLevelSetInportDword

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
현재 지원하지 않음

Format

C

```
DWORD AxdiLevelSetInportDword(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiLevelSetInportDword(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiLevelSetInportDword(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호(0 ~ n-1)
IOffset	In	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Double Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.
모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
지원하지 않음		

C Example

```
// 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportDword(0, 0, 0xFFFFFFFF);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
AxdiLevelSetInportDword 0, 0, 0xFFFFFFFF
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다. }  
Axdilevelsetinportdword (0, 0, 0xFFFFFFFF);
```

See Also

[Axdilevelsetinportbit](#), [Axdilevelsetinportbyte](#), [Axdilevelsetinportword](#), [Axdilevelsetinport](#)

AxdiLevelSetInport

Purpose

전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiLevelSetInport(long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiLevelSetInport (ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiLevelSetInport(IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		전체 입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 전체 입력 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdSetPortLevel(0, 1);
```

VB Example

```
' 전체 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdSetPortLevel 0, 1
```

Delphi Example

```
{ 전체 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }
AxdiLevelSetInport(0, 1);
```

See Also

[AxdiLevelSetInportBit](#), [AxdiLevelSetInportByte](#), [AxdiLevelSetInportWord](#), [AxdiLevelSetInportDword](#),

AxdiLevelGetInportBit

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdiLevelGetInportBit(long IModuleNo, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiLevelGetInportBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiLevelGetInportBit(IModuleNo : LongInt; IOffset : LongInt; upLevel : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiLevelGetInportBit(0, 0, &dwLevel);
```

VB Example

```
' 0번째 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
```

```
Dim lLevel As long  
AxdiLevelGetInportBit 0, 0, lLevel
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiLevelGetInportBit(0, 0, @dwLevel);  
end;
```

See Also

[AxdiLevelGetInportByte](#), [AxdiLevelGetInportWord](#), [AxdiLevelGetInport](#)

AxdiLevelGetInportByte

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdiLevelGetInportByte(long IModuleNo, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiLevelGetInportByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiLevelGetInportByte(IModuleNo : LongInt; IOffset : LongInt; upLevel : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 모듈의 offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiLevelGetInportByte(0, 0, &dwLevel);
```

VB Example

```
' 0번째 모듈의 offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.
```



```
Dim lLevel As long  
AxdiLevelGetInportByte 0, 0, lLevel
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiLevelGetInportByte(0, 0, @dwLevel);  
end;
```

See Also

[AxdiLevelGetInportBit](#), [AxdiLevelGetInportWord](#), [AxdiLevelGetInport](#)

AxdiLevelGetInportWord

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdiLevelGetInportWord(long IModuleNo, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiLevelGetInportWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiLevelGetInportWord(IModuleNo : LongInt; IOffset : LongInt; upLevel : PWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 모듈의 offset 0번지에서 word 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiLevelGetInportWord(0, 0, &dwLevel);
```

VB Example

```
' 0번째 모듈의 offset 0번지에서 word 단위로 데이터 레벨을 확인한다.
```

```
Dim lLevel As long  
AxdiLevelGetInportWord 0, 0, lLevel
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiLevelGetInportWord(0, 0, @dwLevel);  
end;
```

See Also

[AxdiLevelGetInportBit](#), [AxdiLevelGetInportByte](#), [AxdiLevelGetInport](#)

AxdiLevelGetInport

Purpose

전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdiLevelGetInport(long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiLevelGetInport(ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiLevelGetInport(IOffset : LongInt; upLevel : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 전체 입력모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiLevelGetInport(0, &dwLevel);
```

VB Example

```
' 전체 입력모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
Dim lLevel As long

AxdiLevelGetInport 0, lLevel
```

Delphi Example

```
{ 전체 입력모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }
var
dwLevel : DWord;

begin
AxdiLevelGetInport(0, @dwLevel);
```

```
end;
```

See Also

[AxdilevelGetInportBit](#), [AxdilevelGetInportByte](#), [AxdilevelGetInportWord](#)

AxdoLevelSetOutputBit

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다

Format

C

```
DWORD AxdoLevelSetOutputBit(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoLevelSetOutputBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoLevelSetOutputBit(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputBit (0, 0, 1);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputBit 0, 0, 1
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }  
AxdoLevelSetOutputBit (0, 0, 1);
```

See Also

[AxdoLevelSetOutputByte](#), [AxdoLevelSetOutputWord](#), [AxdoLevelSetOutputDword](#), [AxdoLevelSetOutput](#)

AxdoLevelSetOutputByte

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다

Format

C

```
DWORD AxdoLevelSetOutputByte(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoLevelSetOutputByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoLevelSetOutputByte(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputByte(0, 0, 0xFF);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputByte, 0, 0xFF
```


Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다. }  
AxdoLevelSetOutputByte (0, 0, 0xFF);
```

See Also

[AxdoLevelSetOutputBit](#), [AxdoLevelSetOutputWord](#), [AxdoLevelSetOutputDword](#), [AxdoLevelSetOutput](#)

AxdoLevelSetOutputWord

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다

Format

C

```
DWORD AxdoLevelSetOutputWord(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoLevelSetOutputWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoLevelSetOutputWord(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputWord(0, 0, 0xFFFF);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputWord 0, 0, 0xFFFF
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다. }  
AxdoLevelSetOutportWord (0, 0, 0xFFFF);
```

See Also

[AxdoLevelSetOutportBit](#), [AxdoLevelSetOutportByte](#), [AxdoLevelSetOutportDword](#), [AxdoLevelSetOutport](#)

AxdoLevelSetOutputDWord

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다
지원하지 않음

Format

C

```
DWORD AxdoLevelSetOutputDWord(long IModuleNo, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoLevelSetOutputDWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoLevelSetOutputDWord(IModuleNo : LongInt; IOffset : LongInt; uLevel : DWord) : DWord;
stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (double Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.
모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
지원하지 않음		

C Example

```
// 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputDword(0, 0, 0xFFFFFFFF);
```

VB Example

```
' 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutputDword 0, 0, 0xFFFFFFFF
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다. }  
AxdoLevelSetOutportDword (0, 0, 0xFFFFFFFF);
```

See Also

[AxdoLevelSetOutportBit](#), [AxdoLevelSetOutportByte](#), [AxdoLevelSetOutportWord](#), [AxdoLevelSetOutport](#)

AxdoLevelSetOutput

Purpose

전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다

Format

C

```
DWORD AxdoLevelSetOutput(long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoLevelSetOutput(ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoLevelSetOutput(LongInt: uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 전체 출력 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutput(0, 1);
```

VB Example

```
' 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoLevelSetOutput 0, 1
```

Delphi Example

```
{ 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }
AxdoLevelSetOutput (0, 1);
```

See Also

[AxdoLevelSetOutputBit](#), [AxdoLevelSetOutputByte](#), [AxdoLevelSetOutputWord](#),
[AxdoLevelSetOutputDword](#)

AxdoLevelGetOutputBit

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoLevelGetOutputBit(long IModuleNo, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdoLevelGetOutputBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoLevelGetOutputBit(IModuleNo : LongInt; IOffset : LongInt; upLevel : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoLevelGetOutputBit (0, 0, &dwLevel);
```

VB Example

```
' 0번째 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
```

```
Dim lLevel As long  
AxdoLevelGetOutportBit 0, 0, lLevel
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdoLevelGetOutportBit (0, 0, @dwLevel);  
end;
```

See Also

[AxdoLevelGetOutportByte](#), [AxdoLevelGetOutportWord](#), [AxdoLevelGetOutport](#)

AxdoLevelGetOutputByte

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoLevelGetOutputByte(long IModuleNo, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdoLevelGetOutputByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoLevelGetOutputByte(IModuleNo : LongInt; IOffset : LongInt; upLevel : PDWord) : DWord;
stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoLevelGetOutputByte (0, 0, &dwLevel);
```

VB Example

```
‘ 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.
Dim lLevel As long

AxdoLevelGetOutportByte 0, 0, lLevel
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다. }
var
dwLevel : DWord;

begin
AxdoLevelGetOutportByte (0, 0, @dwLevel);
end;
```

See Also

[AxdoLevelGetOutportBit](#), [AxdoLevelGetOutportWord](#), [AxdoLevelGetOutport](#)

AxdoLevelGetOutputWord

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoLevelGetOutputWord(long IModuleNo, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdoLevelGetOutputWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoLevelGetOutputWord(IModuleNo : LongInt; IOffset : LongInt; upLevel : PDWord) : DWord;
stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoLevelGetOutputWord (0, 0, &dwLevel);
```

VB Example

```
‘ 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다.
Dim lLevel As long

AxdoLevelGetOutportWord 0, 0, lLevel
```

Delphi Example

```
{ 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다. }
var
dwLevel : DWord;

begin
AxdoLevelGetOutportWord (0, 0, @dwLevel);
end;
```

See Also

[AxdoLevelGetOutportBit](#), [AxdoLevelGetOutportByte](#), [AxdoLevelGetOutport](#)

AxdoLevelGetOutput

Purpose

전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoLevelGetOutput(long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdoLevelGetOutput(ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoLevelGetOutput(IOffset : LongInt; upLevel : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 전체 출력모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoLevelGetOutput(0, &dwLevel);
```

VB Example

```
' 전체 출력모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
Dim lLevel As long

AxdoLevelGetOutput 0, lLevel
```

Delphi Example

```
{ 전체 출력모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }
var
dwLevel : DWord;

begin
AxdoLevelGetOutput(0, @dwLevel);
```

```
end;
```

See Also

[AxdoLevelGetOutportBit](#), [AxdoLevelGetOutportByte](#), [AxdoLevelGetOutportWord](#)

AxdoWriteOutportBit

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoWritePortBit(long IModuleNo, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoWritePortBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoWritePortBit(IModuleNo : LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoWriteOutportBit (0, 0, 1);
```

VB Example

```
' 0번째 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoWriteOutportBit 0, 0, 1
```

Delphi Example

```
{ 0번째 출력 점점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다. }  
AxdoWriteOutportBit (0, 0, 1);
```

See Also

[AxdoWriteOutportByte](#), [AxdoWriteOutportWord](#), [AxdoWriteOutport](#)

AxdoWriteOutportByte

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoWritePortByte(long IModuleNo, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoWritePortByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoWritePortByte(IModuleNo : LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 출력 접점 모듈의 Offset 0번지에서 byte 단위로 데이터를 출력한다.
AxdoWriteOutportByte (0, 0, 0xFF);
```

VB Example

```
' 0번째 출력 접점 모듈의 Offset 0번지에서 byte 단위로 데이터를 출력한다.
AxdoWriteOutportByte 0, 0, 0xFF
```

Delphi Example

```
{ 0번째 출력 점점 모듈의 offset 0번지에서 byte 단위로 데이터를 출력한다. }  
AxdoWriteOutportByte (0, 0, 0xFF);
```

See Also

[AxdoWriteOutportBit](#), [AxdoWriteOutportWord](#), [AxdoWriteOutport](#)

AxdoWriteOutportWord

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoWritePortWord(long IModuleNo, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoWritePortWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoWritePortWord(IModuleNo : LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 출력한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 출력 접점 모듈의 Offset 0번지에서 word 단위로 데이터를 출력한다.
AxdoWriteOutportWord (0, 0, 0xFFFF);
```

VB Example

```
' 0번째 출력 접점 모듈의 Offset 0번지에서 word 단위로 데이터를 출력한다.
AxdoWriteOutportWord 0, 0, 0xFFFF
```

Delphi Example

```
{ 0번째 출력 점점 모듈의 offset 0번지에서 word 단위로 데이터를 출력한다. }  
AxdoWriteOutportWord (0, 0, 0xFFFF);
```

See Also

[AxdoWriteOutportBit](#), [AxdoWriteOutportByte](#), [AxdoWriteOutport](#)

AxdoWriteOutport

Purpose

전체 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoWritePort(long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoWritePort(ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoWritePort(IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.

Offset은 0부터 총 출력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 0번째 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoWriteOutport (0, 1);
```

VB Example

```
' 0번째 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoWriteOutport 0, 1
```

Delphi Example

```
{ 0번째 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다. }
AxdoWriteOutport (0, 1);
```

See Also

[AxdoWriteOutportBit](#), [AxdoWriteOutportByte](#), [AxdoWriteOutportWord](#)

AxdoReadOutputBit

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoReadOutputBit(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoReadOutputBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoReadOutputBit(IModuleNo : LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 모듈에서 offset 0번지에 데이터를 bit 단위로 읽는다.
DWORD dwValue;

AxdoReadPortBit(0, 0, &dwValue);
```

VB Example

```
' 0번째 모듈에서 offset 0번지에 데이터를 bit 단위로 읽는다.
```

```
Dim lValue As Long  
AxdoReadPortBit 0, 0, lValue
```

Delphi Example

```
{ 0번째 모듈에서 offset 0번지에 데이터를 bit 단위로 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoReadPortBit(0, 0, @dwValue);  
end;
```

See Also

[AxdoReadOutportByte](#), [AxdoReadOutportWord](#), [AxdoReadOutport](#)

AxdoReadOutportByte

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoReadOutportByte(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoReadOutportByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoReadOutportByte(IModuleNo : LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0 ~ 1	00h ~FFh
EIP-DB32T	0 ~ 1	00h ~FFh

C Example

```
// 0번째 모듈에서 offset 0번지에 데이터를 byte 단위로 읽는다.
DWORD dwValue;

AxdoReadPortByte(0, 0, &dwValue);
```

VB Example

```
' 0번째 모듈에서 offset 0번지에 데이터를 byte 단위로 읽는다.
```



```
Dim lValue As Long  
AxdoReadPortByte 0, 0, lValue
```

Delphi Example

```
{ 0번째 모듈에서 offset 0번지에 데이터를 byte 단위로 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoReadPortByte(0, 0, @dwValue);  
end;
```

See Also

[AxdoReadOutputBit](#), [AxdoReadOutputWord](#), [AxdoReadOutput](#)

AxdoReadOutportWord

Purpose

지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoReadOutportWord(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoReadOutportWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoReadOutportWord(IModuleNo : LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0	0000h ~FFFFh
EIP-DB32T	0	0000h ~FFFFh

C Example

```
// 0번째 모듈에서 offset 0번지에 데이터를 word 단위로 읽는다.
DWORD dwValue;

AxdoReadPortWord(0, 0, &dwValue);
```

VB Example

```
' 0번째 모듈에서 offset 0번지에 데이터를 word 단위로 읽는다.
```

```
Dim lValue As Long  
AxdoReadPortWord 0, 0, lValue
```

Delphi Example

```
{ 0번째 모듈에서 offset 0번지에 데이터를 word 단위로 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoReadPortWord(0, 0, @dwValue);  
end;
```

See Also

[AxdoReadOutputBit](#), [AxdoReadOutputByte](#), [AxdoReadOutput](#)

AxdoReadOutput

Purpose

전체 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoReadOutput(long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoReadOutput(ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoReadOutput(IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Offset은 0부터 총 출력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 전체 출력 모듈의 offset 0번지에서 bit 단위로 데이터를 읽는다.
DWORD dwValue;

AxdoReadOutput(0, &dwValue);
```

VB Example

```
、 전체 출력 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
Dim lValue As Long

AxdoReadOutput 0, lValue
```

Delphi Example

```
{ 전체 출력 모듈의 offset 0번지에서 bit 단위로 데이터를 읽는다. }
var
dwValue : DWord;

begin
AxdoReadOutput(0, @dwValue);
```

```
end;
```

See Also

[AxdoReadOutputBit](#), [AxdoReadOutputByte](#), [AxdoReadOutputWord](#)

AxdiReadInportBit

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiReadInportBit(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiReadInportBit(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdiReadInportBit(IModuleNo : LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 모듈에서 offset 0번지에 데이터를 bit 단위로 읽는다.
DWORD dwValue;

AxdiReadInportBit(0, 0, &dwValue);
```

VB Example

```
' 0번째 모듈에서 offset 0번지에 데이터를 bit 단위로 읽는다.
```

```
Dim lValue As Long  
AxdiReadInportBit 0, 0, lValue
```

Delphi Example

```
{ 0번째 모듈에서 offset 0번지에 데이터를 bit 단위로 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdiReadInportBit (0, 0, @dwValue);  
end;
```

See Also

[AxdiReadInportByte](#), [AxdiReadInportWord](#), [AxdiReadInport](#)

AxdiReadInportByte

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiReadInportByte(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiReadInportByte(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdiReadInportByte(IModuleNo : LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0 ~ 1	00h ~FFh
EIP-DB32T	0 ~ 1	00h ~FFh

C Example

```
// 0번째 모듈에서 offset 0번지에 데이터를 byte 단위로 읽는다.
DWORD dwValue;

AxdiReadInportByte(0, 0, &dwValue);
```

VB Example

```
' 0번째 모듈에서 offset 0번지에 데이터를 byte 단위로 읽는다.
```



```
Dim lValue As Long  
AxdiReadInportByte 0, 0, lValue
```

Delphi Example

```
{ 0번째 모듈에서 offset 0번지에 데이터를 byte 단위로 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdiReadInportByte (0, 0, @dwValue);  
end;
```

See Also

[AxdiReadInportBit](#), [AxdiReadInportWord](#), [AxdiReadInport](#)

AxdiReadInportWord

Purpose

지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiReadInportWord(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiReadInportWord(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdiReadInportWord(IModuleNo : LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		모듈 번호 (0 ~ n-1)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.

모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Value
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 모듈에서 offset 0번지에 데이터를 word 단위로 읽는다.
DWORD dwValue;

AxdiReadInportWord(0, 0, &dwValue);
```

VB Example

```
' 0번째 모듈에서 offset 0번지에 데이터를 word 단위로 읽는다.
```

```
Dim lValue As Long  
AxdiReadInportWord 0, 0, lValue
```

Delphi Example

```
{ 0번째 모듈에서 offset 0번지에 데이터를 word 단위로 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdiReadInportWord (0, 0, @dwValue);  
end;
```

See Also

[AxdiReadInportBit](#), [AxdiReadInportByte](#), [AxdiReadInport](#)

AxdiReadInport

Purpose

전체 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdiReadInport(long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiReadInport(ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdiReadInport(IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IOffset	in	long		입력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.

C Example

```
// 전체 입력 모듈의 offset 0번지에서 bit 단위로 데이터를 읽는다.
DWORD dwValue;

AxdiReadInport (0, &dwValue);
```

VB Example

```
、 전체 입력 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
Dim lValue As Long

AxdiReadInport 0, lValue
```

Delphi Example

```
{ 전체 입력 모듈의 offset 0번지에서 bit 단위로 데이터를 읽는다. }
var
dwValue : DWord;

begin
AxdiReadInport (0, @dwValue);
```

```
end;
```

See Also

[AxdReadInpor tBit](#), [AxdReadInpor tByte](#), [AxdReadInpor tWord](#)

AxdiNLevelSetInportBit

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelSetInportBit(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiNLevelSetInportBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiNLevelSetInportBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdiNLevelSetInportBit (0, 0, 0, 1);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdinLevelSetInportBit 0, 0, 0, 1

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }
AxdinLevelSetInportBit (0, 0, 0, 1);

See Also

[AxdinLevelSetInportByte](#), [AxdinLevelSetInportWord](#), [AxdinLevelSetInportDword](#), [AxdinLevelSetInport](#)

AxdiNLevelSetInportByte

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelSetInportByte(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiNLevelSetInportByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiNLevelSetInportByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdiNLevelSetInportByte (0, 0, 0, 0xFF);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdinLevelSetInportByte 0, 0, 0, 0xFF

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 설정한다. }
AxdinLevelSetInportByte (0, 0, 0, 0xFF);

See Also

[AxdinLevelSetInportBit](#), [AxdinLevelSetInportWord](#), [AxdinLevelSetInportDword](#), [AxdinLevelSetInport](#)

AxdiNLevelSetInportWord

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelSetInportWord(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiNLevelSetInportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiNLevelSetInportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	0000h ~ FFFFh
EIP-DB32T	0 ~ 1	0000h ~ FFFFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdiNLevelSetInportWord (0, 0, 0, 0xFFFF);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdinLevelSetInportWord 0, 0, 0, 0xFFFF

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다. }
AxdinLevelSetInportWord (0, 0, 0, 0xFFFF);

See Also

[AxdinLevelSetInportBit](#), [AxdinLevelSetInportByte](#), [AxdinLevelSetInportDword](#), [AxdinLevelSetInport](#)

AxdiNLevelSetInportDword

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
현재 지원하지 않음

Format

C

```
DWORD AxdiNLevelSetInportDword(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiNLevelSetInportDword(ByVal INodeNum As Long, ByVal IModulePosAs Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiNLevelSetInportDword(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Double Word)

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
지원하지 않음		

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
AxdiNLevelSetInportDword (0, 0, 0, 0xFFFFFFFF);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
`AxdiNLevelSetInportDword 0, 0, 0, 0xFFFFFFFF`

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다. }
`AxdiNLevelSetInportDword (0, 0, 0, 0xFFFFFFFF);`

See Also

[AxdiNLevelSetInportBit](#), [AxdiNLevelSetInportByte](#), [AxdiNLevelSetInportWord](#), [AxdiNLevelSetInport](#)

AxdiNLevelSetInport

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelSetInport(long INodeNum, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdiNLevelSetInport(ByVal INodeNum As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdiNLevelSetInport(INodeNum: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		입력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
// 0 번째 노드, 전체 입력 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdiNLevelSetInport(0,0,1);
```

VB Example

```
// 0 번째 노드, 전체 입력 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdiNLevelSetInport(0,0,1);
```

Delphi Example

```
{ 0번째 노드, 전체 입력 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }
AxdiNLevelSetInport(0, 0, 1);
```

See Also

[AxdINLevelSetInportBit](#), [AxdINLevelSetInportByte](#), [AxdINLevelSetInportWord](#),
[AxdINLevelSetInportDword](#)

AxdiNLevelGetInportBit

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelGetInportBit(long INodeNum, long IModulePos, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiNLevelGetInportBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiNLevelGetInportBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiNLevelGetInportBit (0, 0, 0, &dwLevel);
```


VB Example

```
` 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.  
Dim lLevel As long  
  
AxdiNLevelGetInportBit 0, 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiNLevelGetInportBit (0, 0, 0, @dwLevel);  
end;
```

See Also

[AxdiNLevelGetInportByte](#), [AxdiNLevelGetInportWord](#), [AxdiNLevelGetInport](#)

AxdiNLevelGetInportByte

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelGetInportByte(long INodeNum, long IModulePos, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiNLevelGetInportByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiNLevelGetInportByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~FFh
EIP-DB32T	0 ~ 1	00h ~FFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiNLevelGetInportByte (0, 0, 0, &dwLevel);
```

VB Example

```
` 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.  
Dim lLevel As long  
  
AxdiNLevelGetInportByte 0, 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiNLevelGetInportByte (0, 0, 0, @dwLevel);  
end;
```

See Also

[AxdiNLevelGetInportByte](#), [AxdiNLevelGetInportWord](#), [AxdiNLevelGetInport](#)

AxdiNLevelGetInportWord

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelGetInportWord(long INodeNum, long IModulePos, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiNLevelGetInportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiNLevelGetInportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~FFFFh
EIP-DB32T	0	0000h ~FFFFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiNLevelGetInportWord (0, 0, 0, &dwLevel);
```

VB Example

```
` 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다.  
Dim lLevel As long  
  
AxdiNLevelGetInportWord 0, 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiNLevelGetInportWord (0, 0, 0, @dwLevel);  
end;
```

See Also

[AxdiNLevelGetInportBit](#), [AxdiNLevelGetInportByte](#), [AxdiNLevelGetInport](#)

AxdiNLevelGetInport

Purpose

지정한 노드에 대한 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdiNLevelGetInport (long INodeNum, long IOffset, DWORD *upLevel);
```

Visual Basic

```
Function AxdiNLevelGetInportBit(ByVal INodeNum As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdiNLevelGetInportBit(INodeNum: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		입력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
// 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdiNLevelGetInport (0, 0, &dwLevel);
```

VB Example

```
' 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
Dim lLevel As long

AxdiNLevelGetInport 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 전체 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdINLevelGetInport(0, 0, @dwLevel);  
end;
```

See Also

[AxdINLevelGetInportBit](#), [AxdINLevelGetInportByte](#), [AxdINLevelGetInportWord](#)

AxdoNLevelSetOutputBit

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdoNLevelSetOutputBit(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoNLevelSetOutputBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoNLevelSetOutputBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutputBit (0, 0, 0, 1);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutputPortBit 0, 0, 0, 1

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }
AxdoNLevelSetOutputPortBit (0, 0, 0, 1);

See Also

[AxdiNLevelSetInportByte](#), [AxdiNLevelSetInportWord](#), [AxdiNLevelSetInportDword](#), [AxdiNLevelSetInport](#)

AxdoNLevelSetOutputByte

Purpose

지정한 노드에 대한 지정 출력 점점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdoNLevelSetOutputByte(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoNLevelSetOutputByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoNLevelSetOutputByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 점점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 점점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutputByte (0, 0, 0, 0xFF);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 byte 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutputByte 0, 0, 0, 0xFF

Delphi Example

{0번째 노드에 대한 0번째 모듈의 offset 0번지에서 byte 단위로 데이터 레벨을 설정한다. }
AxdoNLevelSetOutputByte (0, 0, 0, 0xFF);

See Also

[AxdoNLevelSetOutputBit](#), [AxdoNLevelSetOutputWord](#), [AxdoNLevelSetOutputDword](#),
[AxdoNLevelSetOutput](#)

AxdoNLevelSetOutportWord

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdoNLevelSetOutportWord(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoNLevelSetOutportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoNLevelSetOutportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 설정한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutportWord (0, 0, 0, 0xFFFF);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutputWord 0, 0, 0, 0xFFFF

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 설정한다. }
AxdoNLevelSetOutputWord (0, 0, 0, 0xFFFF);

See Also

[AxdoNLevelSetOutputBit](#), [AxdoNLevelSetOutputByte](#), [AxdoNLevelSetOutputDword](#),
[AxdoNLevelSetOutput](#)

AxdoNLevelSetOutputDword

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.
현재 지원하지 않음

Format

C

```
DWORD AxdoNLevelSetOutputDword(long INodeNum, long IModulePos, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoNLevelSetOutputDword(ByVal INodeNum As Long, ByVal IModulePosAs Long, ByVal IOffset  
As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoNLevelSetOutputDword(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uLevel :  
DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
LNodeNum	in	long		노드 번호
LModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
LOffset	in	long		출력 접점에 대한 Offset 위치
ULevel	in	DWORD		레벨 값 (Double Word)

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터 레벨을 설정한다.

지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.

노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
지원하지 않음		

C Example

```
// 0번째 노드에 대한 0번째 모듈의 offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.  
AxdoNLevelSetOutputDword (0, 0, 0, 0xFFFFFFFF);
```

VB Example

‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutportDword 0, 0, 0, 0xFFFFFFFF

Delphi Example

{0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터 레벨을 설정한다. }
AxdoNLevelSetOutportDword (0, 0, 0, 0xFFFFFFFF);

See Also

[AxdoNLevelSetOutportBit](#), [AxdoNLevelSetOutportByte](#), [AxdoNLevelSetOutportWord](#),
[AxdoNLevelSetOutport](#)

AxdoNLevelSetOutput

Purpose

지정한 노드에 대한 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.

Format

C

```
DWORD AxdoNLevelSetOutput(long INodeNum, long IOffset, DWORD uLevel);
```

Visual Basic

```
Function AxdoNLevelSetOutputBit(ByVal INodeNum As Long, ByVal IOffset As Long, ByVal uLevel As Long) As Long
```

Delphi

```
function AxdoNLevelSetOutputBit(INodeNum: LongInt; IOffset : LongInt; uLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		출력 접점에 대한 Offset 위치
uLevel	in	DWORD		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 설정한다.
Offset은 0부터 총 출력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
// 0 번째 노드, 전체 출력 모듈, Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutput (0,0,1);
```

VB Example

```
' 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다.
AxdoNLevelSetOutput 0, 0, 1
```

Delphi Example

```
{0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 설정한다. }
AxdoNLevelSetOutput (0, 0, 1);
```

See Also

[AxdoNLevelSetOutportBit](#), [AxdoNLevelSetOutportByte](#), [AxdoNLevelSetOutportWord](#),
[AxdoNLevelSetOutportDword](#)

AxdoNLevelGetOutputBit

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoNLevelGetOutputBit(long INodeNum, long IModulePos, long IOffset, DWORD upLevel);
```

Visual Basic

```
Function AxdoNLevelGetOutputBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoNLevelGetOutputBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoNLevelGetOutputBit (0, 0, 0, &dwLevel);
```

VB Example

```
' 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.  
Dim lLevel As long  
  
AxdoNLevelGetOutportBit 0, 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdoNLevelGetOutportBit (0, 0, 0, @dwLevel);  
end;
```

See Also

[AxdoNLevelGetOutportByte](#), [AxdoNLevelGetOutportWord](#), [AxdoNLevelGetOutport](#)

AxdoNLevelGetOutportByte

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoNLevelGetOutportByte(long INodeNum, long IModulePos, long IOffset, DWORD upLevel);
```

Visual Basic

```
Function AxdoNLevelGetOutportByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal upLevel As Long) As Long
```

Delphi

```
function AxdoNLevelGetOutportByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터 레벨을 확인한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoNLevelGetOutportByte (0, 0, 0, &dwLevel);
```

VB Example

```
' 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다.  
Dim lLevel As long  
  
AxdoNLevelGetOutportByte 0, 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdoNLevelGetOutportByte (0, 0, 0, @dwLevel);  
end;
```

See Also

[AxdoNLevelGetOutportBit](#), [AxdoNLevelGetOutportWord](#), [AxdoNLevelGetOutport](#)

AxdoNLevelGetOutportWord

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoNLevelGetOutportWord(long INodeNum, long IModulePos, long IOffset, DWORD upLevel);
```

Visual Basic

```
Function AxdoNLevelGetOutportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoNLevelGetOutportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터 레벨을 확인한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	0000h ~ FFFFh
EIP-DB32T	0 ~ 1	0000h ~ FFFFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoNLevelGetOutportWord (0, 0, 0, &dwLevel);
```

VB Example

```
' 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다.  
Dim lLevel As long  
  
AxdoNLevelGetOutportWord 0, 0, 0, lLevel
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdoNLevelGetOutportWord (0, 0, 0, @dwLevel);  
end;
```

See Also

[AxdoNLevelGetOutportBit](#), [AxdoNLevelGetOutportByte](#), [AxdoNLevelGetOutport](#)

AxdoNLevelGetOutput

Purpose

지정한 노드에 대한 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.

Format

C

```
DWORD AxdoNLevelGetOutput (long INodeNum, long IOffset, DWORD upLevel);
```

Visual Basic

```
Function AxdoNLevelGetOutput(ByVal INodeNum As Long, ByVal IOffset As Long, ByRef upLevel As Long) As Long
```

Delphi

```
function AxdoNLevelGetOutput(INodeNum: LongInt; IOffset : LongInt; upLevel : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		출력 접점에 대한 Offset 위치
upLevel	out	DWORD*		레벨 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터 레벨을 확인한다.
Offset은 0부터 총 출력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
// 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
DWORD dwLevel;

AxdoNLevelGetOutput (0, 0, &dwLevel);
```

VB Example

```
' 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터 레벨을 확인한다.
Dim lLevel As long

AxdoNLevelGetOutput 0, 0, lLevel
```

Delphi Example


```
{ 0번째 노드에 대한 전체 모듈의 offset 0번지에서 bit 단위로 데이터 레벨을 확인한다. }  
var  
dwLevel : DWord;  
  
begin  
AxdiNLevelGetInport(0, 0, @dwLevel);  
end;
```

See Also

[AxdoNLevelGetOutportBit](#), [AxdoNLevelGetOutportByte](#), [AxdoNLevelGetOutportWord](#),

AxdoNWriteOutputBit

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoNWriteOutputBit(long INodeNum, long IModulePos, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoNWriteOutputBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoNWriteOutputBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoNWriteOutputBit (0, 0, 0, 1);
```

VB Example

‘0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoNWriteOutportBit 0, 0, 0, 1

Delphi Example

{0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 bit 단위로 데이터를 출력한다. }
AxdoNWriteOutportBit (0, 0, 0, 1);

See Also

[AxdoWriteOutportByte](#), [AxdoWriteOutportWord](#), [AxdoNWriteOutportDword](#), [AxdoWriteOutport](#)

AxdoNWriteOutputByte

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoNWriteOutputByte(long INodeNum, long IModulePos, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoNWriteOutputByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoNWriteOutputByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 출력한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 byte 단위로 데이터를 출력한다.
AxdoNWriteOutputByte (0, 0, 0, 0xFF);
```

VB Example

‘0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 byte 단위로 데이터를 출력한다.
AxdoNWriteOutportByte 0, 0, 0, 0xFF

Delphi Example

{0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 byte 단위로 데이터를 출력한다. }
AxdoNWriteOutportByte (0, 0, 0, 0xFF);

See Also

[AxdoWriteOutportBit](#), [AxdoWriteOutportWord](#), [AxdoNWriteOutportDword](#), [AxdoWriteOutport](#)

AxdoNWriteOutportWord

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoNWriteOutportWord(long INodeNum, long IModulePos, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoNWriteOutportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoNWriteOutportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 출력한다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 word 단위로 데이터를 출력한다.
AxdoNWriteOutportWord (0, 0, 0, 0xFFFF);
```

VB Example

‘0번째 노드에 대한 0번째 출력 접점 모듈의 Offset 0번지에서 word 단위로 데이터를 출력한다.
AxdoNWriteOutportWord 0, 0, 0, 0xFFFF

Delphi Example

{0번째 노드에 대한 0번째 출력 접점 모듈의 Offset 0번지에서 word 단위로 데이터를 출력한다. }
AxdoNWriteOutportWord (0, 0, 0, 0xFFFF);

See Also

[AxdoWriteOutputBit](#), [AxdoWriteOutputByte](#), [AxdoNWriteOutputDword](#), [AxdoWriteOutput](#)

AxdoNWriteOutputDword

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터를 출력한다.
현재 지원하지 않음

Format

C

```
DWORD AxdoNWriteOutputDword(long INodeNum, long IModulePos, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoNWriteOutputDword(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoNWriteOutputDword(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Double Word)

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터를 출력한다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
지원하지 않음		

C Example

```
// 0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 double word 단위로 데이터를 출력한다.  
AxdoNWriteOutputDword (0, 0, 0, 0xFFFFFFFF);
```


VB Example

'0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 double word 단위로 데이터를 출력한다.
`AxdoNWriteOutportDword 0, 0, 0, 0xFFFFFFFF`

Delphi Example

{0번째 노드에 대한 0번째 출력 접점 모듈의 offset 0번지에서 double word 단위로 데이터를 출력한다. }
`AxdoNWriteOutportDword (0, 0, 0, 0xFFFFFFFF);`

See Also

[AxdoWriteOutportBit](#), [AxdoWriteOutportByte](#), [AxdoWriteOutportWord](#), [AxdoWriteOutport](#)

AxdoNWriteOutput

Purpose

지정한 노드에 대한 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.

Format

C

```
DWORD AxdoNWriteOutput(long INodeNum, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoNWriteOutput(ByVal INodeNum As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoNWriteOutput(INodeNum: LongInt; IOffset : LongInt; uValue : DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 출력한다.
Offset은 0부터 총 출력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
// 0번째 노드에 대한 전체 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoNWriteOutput (0, 0, 1);
```

VB Example

```
'0번째 노드에 대한 전체 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다.
AxdoNWriteOutput 0, 0, 1
```

Delphi Example

```
{0번째 노드에 대한 전체 출력 접점 모듈의 Offset 0번지에서 bit 단위로 데이터를 출력한다. }
AxdoNWriteOutput(0, 0, 1);
```

See Also

[AxdoWriteOutportBit](#), [AxdoWriteOutportByte](#), [AxdoWriteOutportWord](#), [AxdoNWriteOutportDword](#)

AxdoNReadOutputBit

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoNReadOutputBit(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoNReadOutputBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long,
ByRef upValue As Long) As Long
```

Delphi

```
function AxdoNReadOutputBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upValue :
PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
DWORD dwValue;

AxdoNReadOutputBit(0,0,0, &dwValue);
```

VB Example

```
'0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.  
Dim lValue As Long  
  
AxdoNReadOutportBit 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoNReadOutportBit(0,0,0, @dwValue);  
end;
```

See Also

[AxdoNReadOutportByte](#), [AxdoNReadOutportWord](#), [AxdoNReadOutportDword](#), [AxdoNReadOutport](#)

AxdoNReadOutputByte

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoNReadOutputByte(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoNReadOutputByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoNReadOutputByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터를 읽는다.
DWORD dwValue;

AxdoNReadOutputByte(0,0,0, &dwValue);
```

VB Example

```
'0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터를 읽는다.  
Dim lValue As Long  
  
AxdoNReadOutportByte 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoNReadOutportByte(0,0,0, @dwValue);  
end;
```

See Also

[AxdoNReadOutportBit](#), [AxdoNReadOutportWord](#), [AxdoNReadOutportDword](#), [AxdoNReadOutport](#)

AxdoNReadOutportWord

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoNReadOutportWord(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoNReadOutportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoNReadOutportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터를 읽는다.
DWORD dwValue;

AxdoNReadOutportWord (0,0,0, &dwValue);
```


VB Example

```
' 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터를 읽는다.  
Dim lValue As Long  
  
AxdoNReadOutportWord 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈에서 Offset 0번지에서 word 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoNReadOutportWord (0,0,0, @dwValue);  
end;
```

See Also

[AxdoNReadOutportBit](#), [AxdoNReadOutportByte](#), [AxdoNReadOutportDword](#), [AxdoNReadOutport](#)

AxdoNReadOutportDword

Purpose

지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoNReadOutportDword(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoNReadOutportDword(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoNReadOutportDword(INodeNum: LongInt IModulePos: LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Double Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 double word 단위로 데이터를 읽는다. 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 설정한다. 노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
지원하지 않음		

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터를 읽는다.
DWORD dwValue;

AxdoNReadOutportDword (0,0,0, &dwValue);
```

VB Example

```
‘ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터를 읽는다.  
Dim lValue As Long  
AxdoNReadOutportDword 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 double word 단위로 데이터를 읽는  
  다. }  
var  
dwValue : DWord;  
  
begin  
AxdoNReadOutportDword (0,0,0, @dwValue);  
end;
```

See Also

[AxdoNReadOutportBit](#), [AxdoNReadOutportByte](#), [AxdoNReadOutportWord](#), [AxdoNReadOutport](#)

AxdoNReadOutput

Purpose

지정한 노드에 대한 전체 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdoNReadOutput(long INodeNum, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoNReadOutput(ByVal INodeNum As Long,ByVal IOffset As Long ByRef upValue As Long) As Long
```

Delphi

```
function AxdoNReadOutput(INodeNum: LongInt, IOffset : LongInt; upValue: PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 출력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
Offset은 0부터 총 출력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
/ 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
DWORD dwValue;

AxdoNReadOutput (0, 0, &dwValue);
```

VB Example

```
' 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
Dim lValue As Long

AxdoNReadOutput 0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 전체 모듈의 offset 0번지에서 bit 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdoNReadOutport(0,0, @dwValue);  
end;
```

See Also

[AxdoNReadOutportBit](#), [AxdoNReadOutportByte](#), [AxdoNReadOutportWord](#), [AxdoNReadOutportDword](#)

AxdiNReadInportBit

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiNReadInportBit(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiNReadInportBit(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long,
ByRef upValue As Long) As Long
```

Delphi

```
function AxdiNReadInportBit(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upValue : PDWord) :
DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 15	0(Off), 1(On)
EIP-DB32T	0 ~ 15	0(Off), 1(On)

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
DWORD dwValue;

AxdiNReadInportBit(0,0,0, &dwValue);
```

VB Example

```
'0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.  
Dim lValue As Long  
  
AxdiNReadInportBit 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdiNReadInportBit (0,0,0, @dwValue);  
end;
```

See Also

[AxdoNReadOutportByte](#), [AxdoNReadOutportWord](#), [AxdoNReadOutportDword](#), [AxdoNReadOutport](#)

AxdiNReadInportByte

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiNReadInportByte(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiNReadInportByte(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long,
ByRef upValue As Long) As Long
```

Delphi

```
function AxdiNReadInportByte(INodeNum: LongInt; IModulePos: LongInt; IOffset : LongInt; upValue :
PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Byte)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 byte 단위로 데이터를 읽는다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터를 읽는다.
DWORD dwValue;

AxdiNReadInportByte(0,0,0, &dwValue);
```


VB Example

```
'0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터를 읽는다.  
Dim lValue As Long  
  
AxdiNReadInportByte 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 byte 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdiNReadInportByte (0,0,0, @dwValue);  
end;
```

See Also

[AxdiNReadInportBit](#), [AxdiNReadInportWord](#), [AxdiNReadInport](#)

AxdiNReadInportWord

Purpose

지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiNReadInportWord(long INodeNum, long IModulePos, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiNReadInportWord(ByVal INodeNum As Long, ByVal IModulePos As Long, ByVal IOffset As Long,
ByRef upValue As Long) As Long
```

Delphi

```
function AxdiNReadInportWord(INodeNum: LongInt; IModulePos: LongInt; IOffset: LongInt; upValue :
PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IModulePos	in	long		모듈 위치(사용자가 로터리 스위치로 정한 절대 위치)
IOffset	in	long		입력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Word)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 word 단위로 데이터를 읽는다.
지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 확인한다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

Module	Offsett	Level
EIP-CPU16B	0	0000h ~ FFFFh
EIP-DB32T	0	0000h ~ FFFFh

C Example

```
// 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터를 읽는다.
DWORD dwValue;

AxdiNReadInportWord (0,0,0, &dwValue);
```

VB Example

```
'0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터를 읽는다.  
Dim lValue As Long  
  
AxdiNReadInportWord 0,0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 0번째 모듈의 Offset 0번지에서 word 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdiNReadInportWord (0,0,0, @dwValue);  
end;
```

See Also

[AxdiNReadInportBit](#), [AxdiNReadInportWord](#), [AxdiNReadInport](#)

AxdiNReadInport

Purpose

지정한 노드에 대한 전체 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.

Format

C

```
DWORD AxdiNReadInport(long INodeNum, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdiNReadInportBit(ByVal INodeNum As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdiNReadInportBit(INodeNum: LongInt; IOffset : LongInt; upValue : PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
INodeNum	in	long		노드 번호
IOffset	in	long		입력 접점에 대한 Offset 위치
upValue	out	DWORD*		입력 접점 값 (Boolean)

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

사용자가 지정한 노드에 대한 지정 입력 접점 모듈의 Offset 위치에서 bit 단위로 데이터를 읽는다.
Offset은 0부터 총 입력 접점 수 - 1까지 사용할 수 있다.
노드 번호는 CPU모듈의 로터리 스위치 16진수 두자리 값이다.

C Example

```
// 0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
DWORD dwValue;

AxdiNReadInport (0,0, &dwValue);
```

VB Example

```
'0번째 노드에 대한 전체 모듈의 Offset 0번지에서 bit 단위로 데이터를 읽는다.
Dim lValue As Long

AxdiNReadInport 0,0, lValue
```

Delphi Example

```
{ 0번째 노드에 대한 전체 모듈의 offset 0번지에서 bit 단위로 데이터를 읽는다. }  
var  
dwValue : DWord;  
  
begin  
AxdinReadInport (0,0, @dwValue);  
end;
```

See Also

[AxdinReadInportBit](#), [AxdinReadInportByte](#), [AxdinReadInportWord](#),

AxdoNetWorkErrorSetAction

Purpose

지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어 졌을 때 현재 출력 접점 상태를 설정한다

Format

C

```
DWORD AxdoNetWorkErrorSetAction(long IModuleNo, DWORD dwNetErrorAct);
```

Visual Basic

```
Function AxdoNetWorkErrorSetAction (ByVal IModuleNo As Long, ByVal dwNetErrorAct As Long) As Long
```

Delphi

```
function AxdoNetWorkErrorSetAction (IModuleNo: LongInt; dwNetErrorAct: DWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		노드 번호
dwNetErrorAct	in	DWORD		접점의 상태 (자세한 내용은 아래의 표를 참고)

NetErrorAct

Value	Explanation
00h	동작 하지 않음
01h	AxdoSetNetWorkErrorByteValue()에 의하여 Setting 된 값으로 동작

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어 졌을 때 현재 출력 접점 상태를 설정한다

C Example

```
// 0 번째 모듈이 연결이 끊어 졌을 때 동작하지 않는다.
AxdoNetWorkErrorSetAction (0,0);
```

VB Example

```
' 0 번째 모듈이 연결이 끊어 졌을 때 동작하지 않는다.
AxdoNetWorkErrorSetAction 0, 0
```

Delphi Example

```
0 번째 모듈이 연결이 끊어 졌을 때 동작하지 않는다. }  
AxdoNetWorkErrorSetAction (0, 0);
```

See Also

[AxdoNetWorkErrorGetAction](#), [AxdoNetWorkErrorSetByteValue](#), [AxdoNetWorkErrorGetByteValue](#)

AxdoNetWorkErrorGetAction

Purpose

지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어 졌을 때 현재 출력 접점 상태를 확인한다

Format

C

```
DWORD AxdoNetWorkErrorGetAction(long IModuleNo, DWORD * dwpNetErrorAct);
```

Visual Basic

```
Function AxdoNetWorkErrorGetAction (ByVal IModuleNo As Long, ByRef dwpNetErrorAct As Long) As Long
```

Delphi

```
function AxdoNetWorkErrorGetAction (IModuleNo: LongInt; dwpNetErrorAct: PDWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		노드 번호
dwpNetErrorAct	out	DWORD*		접점의 상태 (자세한 내용은 아래의 표를 참고)

NetErrorAct

Value	Explanation
00h	동작 하지 않음
01h	AxdoSetNetWorkErrorByteValue()에 의하여 Setting 된 값으로 동작

Return

AXT_RT_SUCCESS(0000) : Successful execution of API

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

Description

지정한 출력 접점 모듈이 연결된 노드의 통신이 끊어 졌을 때 현재 출력 접점 상태를 확인한다.

C Example

```
// 0 번째 모듈이 연결이 끊어 졌을 때 동작하지 않는다.
DWORD dwpNetErrorAct;
AxdoNetWorkErrorGetAction(0, &dwpNetErrorAct);
```

VB Example

```
\' 0 번째 모듈이 연결이 끊어 졌을 때 동작하지 않는다.
Dim dwpNetErrorAct As Long
```



```
AxdoNetworkErrorGetAction 0, dwpNetErrorAct
```

Delphi Example

```
{ 0 번째 모듈이 연결이 끊어 졌을 때 동작하지 않는다. }  
var  
  dwpNetErrorAct: DWord;  
  
begin  
  AxdoNetworkErrorGetAction (0, @ dwpNetErrorAct);  
end;
```

See Also

[AxdoNetworkErrorSetAction](#), [AxdoNetworkErrorSetByteValue](#), [AxdoNetworkErrorGetByteValue](#)

AxdoNetWorkErrorSetByteValue

Purpose

Network연결이 끊어 졌을 경우 지정한 출력 접점의 상태를 byte 단위로 설정 한다.

Format

C

```
DWORD AxdoNetWorkErrorSetByteValue (long IModuleNo, long IOffset, DWORD uValue);
```

Visual Basic

```
Function AxdoNetWorkErrorSetByteValue (ByVal IModuleNo As Long, ByVal IOffset As Long, ByVal uValue As Long) As Long
```

Delphi

```
function AxdoNetWorkErrorSetByteValue (IModuleNo : LongInt; IOffset : LongInt; uValue : DWord) : DWord;
stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		노드 번호
IOffset	in	long		출력 접점에 대한 Offset 위치
uValue	in	DWORD		출력 접점의 값(Byte)

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 출력 접점 상태를 설정 한다.
 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 읽는다.
 모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈
 의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
// 0번째 출력 접점 모듈의 Offset 0번지에서 byte 단위로 출력 접점 상태를 설정 한다.
AxdoNetWorkErrorSetByteValue (0, 0, 0xFF);
```

VB Example

```
' 0번째 출력 접점 모듈의 Offset 0번지에서 byte 단위로 출력 접점 상태를 설정 한다.
AxdoNetWorkErrorSetByteValue 0, 0, 0xFF
```

Delphi Example

```
{ 0번째 출력 접점 모듈의 Offset 0번지에서 byte 단위로 출력 접점 상태를 설정 한다. }  
AxdoNetWorkErrorSetByteValue 0, 0, 0xFF
```

See Also

[AxdoNetWorkErrorSetAction](#), [AxdoNetWorkErrorGetAction](#), [AxdoNetWorkErrorGetByteValue](#)

AxdoNetWorkErrorGetByteValue

Purpose

Network연결이 끊어 졌을 경우 지정한 출력 접점의 상태를 byte 단위로 확인 한다.

Format

C

```
DWORD AxdoNetWorkErrorGetByteValue(long IModuleNo, long IOffset, DWORD *upValue);
```

Visual Basic

```
Function AxdoNetWorkErrorGetByteValue(ByVal IModuleNo As Long, ByVal IOffset As Long, ByRef upValue As Long) As Long
```

Delphi

```
function AxdoNetWorkErrorGetByteValue (IModuleNo : LongInt; IOffset : LongInt; upValue : PWord) : DWord; stdcall;
```

Input / Output

Name	in/out	Format	Init Value	Explanation
IModuleNo	in	long		노드 번호
IOffset	in	long		출력 접점에 대한 Offset 위치
upValue	out	DWORD*		출력 접점의 값(Byte)

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

Description

사용자가 지정한 출력 접점 모듈의 Offset 위치에서 byte 단위로 출력 접점 상태를 확인 한다.
 지정한 모듈이 어떤 모듈인지 확인하여 아래의 Table에서 Offset 사용 범위 참조하여 읽는다.
 모듈 번호는 가장 빠른 노드(노드 ID 값이 작은 CPU)의 CPU 부터 0으로 시작되며, 그 노드에 장착된 DIO 모듈의 모듈 위치(로터리 스위치 값)에 따라 차례대로 배열 된 후, 다음 빠른 노드로 순서대로 배열 된다.

Module	Offsett	Level
EIP-CPU16B	0 ~ 1	00h ~ FFh
EIP-DB32T	0 ~ 1	00h ~ FFh

C Example

```
/// 0번째 출력 접점 모듈의 Offset 0번지에서 byte 단위로 출력 접점 상태를 확인한다.
DWORD dwValue;

AxdoGetNetWorkErrorByteValue (0, 0, &dwValue);
```

VB Example

' 0번째 출력 점점 모듈의 Offset 0번지에서 byte 단위로 출력 점점 상태를 확인한다.

Dim lValue As Long

AxdoGetNetWorkErrorByteValue 0, 0, lValue

Delphi Example

{0번째 출력 점점 모듈의 Offset 0번지에서 byte 단위로 출력 점점 상태를 확인한다. }

var

dwValue : DWord;

begin

AxdoGetNetWorkErrorByteValue (0, 0, @dwValue);

end;

See Also

[AxdoNetWorkErrorSetAction](#), [AxdoNetWorkErrorGetAction](#), [AxdoNetWorkErrorSetByteValue](#)

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 사용중 임
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)	플래시 상태가 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_CHECKSUM(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 이외의 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)	연속보간 축맵핑할 때 유효한축이 아닐 때

AXT_ERROR_CONTI_INVALID_MAP_NO(4166)	유효한 맵핑 번호가 아닐 때
AXT_ERROR_CONTI_EMPTY_MAP_NO(4167)	맵핑 번호가 비워있을 때
AXT_RT_MOTION_ERROR_CALCULATION(4168)	계산상의 오차가 발생했을 때
AXT_RT_ERROR_NOT_SAME_NODE(4169)	노드 번호가 틀릴 때
AXT_ERROR_HELICAL_INVALID_AXIS_NO(4170)	헬리컬 축 번호가 존재하지 않을 때
AXT_ERROR_HELICAL_INVALID_MAP_NO(4171)	헬리컬 맵이 존재하지 않을 때
AXT_ERROR_HELICAL_EMPTY_MAP_NO(4172)	헬리컬 맵 번호가 비어 있을 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 갠트리 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_MOTION_INVALID_CONT_INDEX(4258)	연속 보간 Index값 설정이 잘못됨
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

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.