Make Life Easy :

라이브러리 매뉴얼

모션 컨트롤러

PMC-2HSP Series

MMD-PMC2HSP2HSN1-V1.3-2109KR

저희 (주)오토닉스 제품을 구입해 주셔서 감사합니다.
사용 전에 안전을 위한 주의 사항을 반드시 읽고 정확하게 사용하십시오.

www.autonics.com

Autonics

제품 구입 감사 안내문

㈜오토닉스 제품을 구입해 주셔서 감사합니다.

먼저 안전을 위한 주의사항을 반드시 읽고 제품을 올바르게 사용해 주십시오

본 라이브러리 매뉴얼은 제품에 대한 안내와 바른 사용 방법에 대한 내용을 담고 있으므로 사용자가 쉽게 찾아 볼 수 있는 장소에 보관하십시오.

라이브러리 매뉴얼 안내

- 라이브러리 매뉴얼의 내용을 충분히 숙지한 후에 제품을 사용하십시오.
- 라이브러리 매뉴얼은 사용자에게 제공되는 라이브러리 함수를 자세하게 설명한 것으로 라이브러리 매뉴얼 이외의 내용에 대해서는 보증하지 않습니다.
- 라이브러리 매뉴얼의 일부 또는 전부를 무단으로 편집 또는 복사하여 사용할 수 없습니다.
- 라이브러리 매뉴얼은 제품과 함께 제공하지 않습니다.
  당사 홈페이지(www.autonics.com)에서 다운로드하여 사용하십시오.
- 라이브러리 매뉴얼의 내용은 해당 제품의 성능 및 소프트웨어 개선에 따라 사전 예고없이 변경될 수 있으며, 업그레이드 공지는 당사 홈페이지를 통해 제공해 드립니다.
- 당사에서는 라이브러리 매뉴얼의 내용을 조금 더 쉽게, 정확하게 작성하고자 많은 노력을 기울였습니다. 그럼에도 불구하고 수정해야 될 부분이나 질문사항이 있으시면 당사 홈페이지를 통하여 의견을 주시기 바랍니다.
- 각 함수 별 사용 예제와 함수를 활용한 MFC 기반의 예제프로그램을 제공합니다.

라이브러리 매뉴얼의 공통 기호

기호	설명
 **Note**	해당 기능에 대한 보충 설명
 **Warning**	지시 사항을 위반할 경우 심각한 상해나 사망 사고의 위험이 있는 내용
 **Caution**	지시 사항을 위반할 경우 경미한 상해나 제품 손상이 발생할 수 있는 내용
 **Ex.**	해당 기능에 대한 예시
※1	주석 설명 표시

안전을 위한 주의사항

- ‘안전을 위한 주의사항’은 제품을 안전하고 올바르게 사용하여 사고나 위험을 미리 막기 위한 것이므로 반드시 지키십시오.
- 주의사항은 '경고'와 '주의'의 두 가지로 구분되어 있으며 '경고'와 '주의'의 의미는 다음과 같습니다.

 **Warning**	**경고**	지시 사항을 위반하였을 때, 심각한 상해나 사망 사고가 발생할 가능성이 있는 경우
 **Caution**	**주의**	지시 사항을 위반하였을 때, 경미한 상해나 제품 손상이 발생할 가능성이 있는 경우

 Warning

- 인명이나 재산상에 영향이 큰 기기(예: 원자력 제어 장치, 의료기기, 선박, 차량, 철도, 항공기, 연소장치, 안전장치, 방범/방재장치 등)에 사용할 경우에는 반드시 2 중으로 안전장치를 부착한 후 사용하십시오.
  인사사고, 재산상의 손실 및 화재 위험이 있습니다.
- 가연성/폭발성/부식성 가스, 다습, 직사광선, 복사열, 진동, 충격, 염분이 있는 환경에서 사용하지 마십시오.
  화재 및 폭발 위험이 있습니다.
- 판넬이나 DIN rail 에 설치하여 사용하십시오.
  화재 위험이 있습니다.
- 전원이 인가된 상태에서 결선, 점검 및 보수를 하지 마십시오.
  화재 위험이 있습니다.
- 배선 시, 접속도를 확인하고 연결하십시오.
  화재 위험이 있습니다.
- 임의로 제품을 개조하지 마십시오.
  화재 위험이 있습니다.
- 제품 동작 중에는 커넥터를 분리하거나 전원을 차단하지 마십시오.
  인사사고, 재산상의 손실 및 제품 고장 위험이 있습니다.
- 외부 전원 이상, 컨트롤러 고장 등의 문제가 발생해도 시스템 전체가 안전하게 동작하도록 컨트롤러의 외부에 안전 보호장치를 마련하십시오.
  인사사고, 재산상의 손실 및 화재 위험이 있습니다.

Caution

- 전원 입력단 배선 시 AWG 28-16(0.081~1.31mm^2) 이상을 사용하십시오.
- 전원 입력 측 회로에 반드시 절연 트랜스를 사용하십시오.
  인사사고 및 화재 위험이 있습니다.
- 정격/성능 범위 내에서 사용하십시오.
  화재 및 제품 고장 위험이 있습니다.
- 청소 시 마른 수건으로 닦으시고, 물, 유기용제는 사용하지 마십시오.
  화재 위험이 있습니다.
- 제품 내부로 금속체, 먼지, 배선 찌꺼기 등의 이물질이 유입되지 않도록 하십시오.
  화재 및 제품 고장 위험이 있습니다.
- 입/출력 배선에 리본 케이블을 사용 시 케이블을 바르게 연결하시고 케이블에 의한 접촉 불량이 발생하지 않도록 하십시오.
  오동작 위험이 있습니다.
- 이 기기는 업무용(A 급)으로 전자파 적합 등록을 한 기기입니다.
  또한 가정 이외의 지역에서의 사용을 목적으로 합니다.

※ **본 라이브러리 매뉴얼에 기재된 사양, 외형치수 등은 제품의 개선을 위하여 예고없이 변경될 수 있습니다**

취급 시 주의사항

- 취급 시 주의사항에 명기된 사항을 지키십시오.
  그렇지 않을 경우, 예기치 못한 사고가 일어날 수 있습니다.
- 24VDC 모델의 전원 입력은 절연되고 제한된 전압/전류 또는 Class2, SELV 전원 장치로 공급하십시오.
- 제품의 전원 입력 및 차단을 위해 스위치나 차단기를 조작이 편리한 곳에 설치하십시오.
- 서지, 유도성 노이즈 방지를 위해 고압선, 전력선 등과 분리하여 배선 작업하시고,
  배선 길이는 가능한 짧게 하십시오.
- 부하 및 사용환경 등에 따라 각 파라미터를 적절한 값으로 설정 후 구동하십시오.
- 전원 인가 전에 atMotion 의 Power On 기능 설정 여부를 확인하십시오.
- 신호 배선과 전원 전선은 반드시 10cm 이상 이격시키십시오.
- CN3, 4, 5 커넥터와 배선 연결 시 Twist pair 쉴드선 사용을 권장합니다.
  설치환경에 따라 필요 시 Shield 선을 접지하십시오.
- 통신 케이블은 제공된 케이블(RS232C, USB) 사용을 권장합니다.
- RS485 케이블 배선 시 Twist pair 선 사용을 권장하며, AWG 24($0.2mm^2$) 이상을 사용하십시오.
- 본 제품은 다음 환경조건에서 사용할 수 있습니다.
  ① 실내(정격/성능의 내환경성 조건 만족)
  ② 고도 2,000m 이하
  ③ 오염등급 2(Pollution Degree 2)
  ④ 설치 카테고리 II(Installation Category II)

Table of Contents

1 초기화

1.1 autpmc_Open

autpmc_Open 함수는 PMC-2HSP 에 통신 연결을 합니다.

(1) 함수명

```
int autpmc_Open(
int PortNum,
int BaudRate
);
```

(2) 파라미터

- PortNum

  접속하려는 Serial Port 숫자를 입력합니다.

- BaudRate

  Serial Port 의 Baudrate 을 입력합니다.

구분	입력	내용	상수값
PMC_BAUDRATE	FPMC_BAUD_9600	9,600bps	9600
	FPMC_BAUD_19200	19,200bps	19200
	FPMC_BAUD_38400	38,400bps	38400
	FPMC_BAUD_57600	57,600bps	57600
	FPMC_BAUD_115200	115,200bps	115200

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_PORT	1	입력한 Port 가 사용 중이거나 잘못된 Port 번호를 입력하였습니다.
	FPMC_INVALID_BAUDRATE	2	잘못된 Baudrate을 입력하였습니다.

1.2 autpmc_Reset

autpmc_Reset 함수는 PMC-2HSP 의 모션 IC 를 초기화합니다. (브로드캐스트 가능)

(1) 함수명

```
int autpmc_Reset(
int PortNum,
char nNodeId
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

  또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

1.3 autpmc_IsCon

autpmc_IsCon 함수는 PMC-2HSP 과 통신 접속을 해서 데이터를 정상적으로 주고 받는지 확인합니다.

(1) 함수명

```
int autpmc_IsCon(
int PortNum,
char nNodeId,
int *bOn
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- bOn
  명령 성공 시 데이터에 1 이 저장되고, 실패 시 0 이 저장됩니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID를 입력하였습니다.

1.4 autpmc_SetBaudrate

autpmc_SetBaudrate 함수는 PMC-2HSP 의 통신 속도(baudrate)를 변경합니다.
(브로드캐스트 가능)

(1) 함수명

```
int autpmc_SetBaudrate(
int PortNum,
char nNodeId,
int BaudRate
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.
- BaudRate
  변경하고자 하는 통신 속도를 입력합니다.

구분	입력	내용	상수값
PMC_BAUDRATE	FPMC_BAUD_9600	9,600bps	9600
	FPMC_BAUD_19200	19,200bps	19200
	FPMC_BAUD_38400	38,400bps	38400
	FPMC_BAUD_57600	57,600bps	57600
	FPMC_BAUD_115200	115,200bps	115200

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_BAUDRATE	2	잘못된 Baudrate 을 입력하였습니다.
	FPMC_INVALID_NODE	3	잘못된 노드 ID를 입력하였습니다.

1.5 autpmc_ClrINCPos

autpmc_ClrINCPos 함수는 PMC-2HSP 의 상대 위치를 초기화합니다.

(1) 함수명

```
int autpmc_ClrINCPos(
int PortNum,
char nNodeId,
char axis
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

2 정지, 종료

2.1 autpmc_Close

autpmc_Close 함수는 PMC-2HSP 의 통신 연결을 해제합니다.

(1) 함수명

```
int autpmc_Close(
int PortNum
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_PORT	1	유효한 Port 가 아닙니다.

2.2 autpmc_SlowStop

autpmc_SlowStop 함수는 PMC-2HSP 을 감속 정지합니다.

(1) 함수명

int autpmc_SlowStop(
int PortNum,
char nNodeId,
char axis
);

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

2.3 autpmc_EmgStop

autpmc_EmgStop 함수는 PMC-2HSP 을 긴급 정지합니다. (브로드캐스트 가능)

(1) 함수명

```
int autpmc_EmgStop(
int PortNum,
char nNodeId
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

3 파라미터

3.1 autpmc_GetParaAll

autpmc_GetParaAll 함수는 PMC-2HSP 의 저장된 모든 설정값을 가져옵니다.

(1) 함수명

```
int PMC_PARADATA *autpmc_GetParaAll(
int PortNum,
char nNodeId,
char axis,
PMC_PARADATA *pData
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- pData

  모든 파라미터의 정보를 한 번에 가져와서 저장합니다.

구조체명	변수 형식	내용	데이터 값
PMC_ PARADATA	int iErrorState	오류 상태 확인	0: 함수가 정상적으로 명령을 수행하였습니다. 3: 잘못된 노드 ID 를 입력하였습니다. 4: 잘못된 축을 입력하였습니다. 5: 잘못된 데이터를 입력하였습니다.
	int bLmtStopMod[2]	리미트 정지 모드 Enable/Disable (X 축: bLmtStopMod[0], Y 축: bLmtStopMod[1])	0: Instant 1: Slow
PMC_ PARADATA	int bLmtActLev[2]	리미트 신호 논리 레벨 Low/High (X 축: bLmtActLev [0], Y 축: bLmtActLev [1])	0: Low 1: High
	int bSCurve[2]	S 자 가감속의 사용 Enable/Disable (X 축: bSCurve [0], Y 축: bSCurve [1])	0: Disable 1: Enable
	int bEndPEnable[2]	드라이브 종료 펄스의 사용 여부 (X 축: bEndPEnable [0], Y 축: bEndPEnable [1])	0: Disable 1: Enable
	int bDecValue[2]	사다리꼴 가감속 드라이브의 대칭/비대칭 (X 축: bDecValue [0], Y 축: bDecValue [1])	0: Accel 1: Decel
	int bSofLmtEnable[2]	소프트웨어 리미트의 Enable/Disable (X 축: bSofLmtEnable [0], Y 축: bSofLmtEnable [1])	0: Enable 1: Disable
	int bPowHomStart[2]	파워 온 원점 복귀 자동 스타트 Enable/Disable (X 축: bPowHomStart [0], Y 축: bPowHomStart [1])	0: Disable 1: Enable
	int bPowPgmStart[2]	파워 온 프로그램 자동 스타트 Enable/Disable (X 축: bPowPgmStart [0], Y 축: bPowPgmStart [1])	0: Disable 1: Enable

구조체명	변수 형식	내용	데이터 값
	int bInput0Lev[2]	범용 입력 0 번의 액티브 레벨 Low/High (X 축: bInput0Lev [0], Y 축: bInput0Lev [1])	0: Low 1: High
	int bInput1Lev[2]	범용 입력 1 번의 액티브 레벨 Low/High (X 축: bInput1Lev [0], Y 축: bInput1Lev [1])	0: Low 1: High
	int iPulseType	펄스 입력 방식 : 1PULSE, 2PULSE	1: 1PULSE 2: 2PULSE
	int iSpdMul[2]	속도 배율 (X 축: iSpdMul [0], Y 축: iSpdMul [1])	1~500
	int iJrkSpd[2]	가가속도 (X 축: iJrkSpd [0], Y 축: iJrkSpd [1])	1~65535
	int iAccSpdRate[2]	가속률 (X 축: iAccSpdRate [0], Y 축: iAccSpdRate [1])	1~8000
	int iDecSpdRate[2]	감속률 (X 축: iDecSpdRate [0], Y 축: iDecSpdRate [1])	1~8000
	int iStrSpd[2]	초기 속도 (X 축: iStrSpd [0], Y 축: iStrSpd [1])	1~8000
	int iDrvSpd[2]	구동 속도 (X 축: iDrvSpd [0], Y 축: iDrvSpd [1])	1~8000
	int iDrvSpd1Pgm[2]	프로그램 모드에서 사용하는 구동 속도 1 (X 축: iDrvSpd1Pgm [0], Y 축: iDrvSpd1Pgm [1])	1~8000
	int iDrvSpd2Pgm[2]	프로그램 모드에서 사용하는 구동 속도 2 (X 축: iDrvSpd2Pgm [0], Y 축: iDrvSpd2Pgm [1])	1~8000
	int iDrvSpd3Pgm[2]	프로그램 모드에서 사용하는 구동 속도 3 (X 축: iDrvSpd3Pgm [0], Y 축: iDrvSpd3Pgm [1])	1~8000

구조체명	변수 형식	내용	데이터 값
	int iDrvSpd4Pgm[2]	프로그램 모드에서 사용하는 구동 속도 4 (X 축: iDrvSpd4Pgm [0], Y 축: iDrvSpd4Pgm [1])	1~8000
	int iTim1Pgm[2]	프로그램 모드에서 사용하는 포스트 타이머 1 (X 축: iTim1Pgm [0], Y 축: iTim1Pgm [1])	1~65535
	int iTim2Pgm[2]	프로그램 모드에서 사용하는 포스트 타이머 2 (X 축: iTim2Pgm [0], Y 축: iTim2Pgm [1])	1~65535
	int iTim3Pgm[2]	프로그램 모드에서 사용하는 포스트 타이머 3 (X 축: iTim3Pgm [0], Y 축: iTim3Pgm [1])	1~65535
	long lSofLmtP[2]	+방향 소프트웨어 리미트 (X 축: lSofLmtP [0], Y 축: lSofLmtP [1])	-8388608~8388607
	long lSofLmtM[2]	-방향 소프트웨어 리미트 (X 축: lSofLmtM [0], Y 축: lSofLmtM [1])	- 8388608 ~ 8388607
	int iEndPWidth[2]	드라이브 종료 펄스의 폭 (X 축: iEndPWidth [0], Y 축: iEndPWidth [1])	1~65535
	int iPulSclNum[2]	펄스 스케일의 분자값 (X 축: iPulSclNum [0], Y 축: iPulSclNum [1])	1~65535
	int iPulSclDen[2]	펄스 스케일의 분모값 (X 축: iPulSclDen [0], Y 축: iPulSclDen [1])	1~65535
	int bHomMod1[2]	원점 복귀 모드의 스텝 1 Enable/Disable (X 축: bHomMod1 [0], Y 축: bHomMod1 [1])	0: Disable 1: Enable
	int bHomMod1Dir[2]	원점 복귀 모드의 스텝 1 탐색 방향 (X 축: bHomMod1Dir [0], Y 축: bHomMod1Dir [1])	0: +방향 1: -방향

구조체명	변수 형식	내용	데이터 값
	int bHomMod2[2]	원점 복귀 모드의 스텝 2 Enable/Disable (X 축: bHomMod2 [0], Y 축: bHomMod2 [1])	0: Disable 1: Enable
	int bHomMod2Dir[2]	원점 복귀 모드의 스텝 2 탐색 방향 (X 축: bHomMod2Dir [0], Y 축: bHomMod2Dir [1])	0: +방향 1: -방향
	int bHomMod3[2]	원점 복귀 모드의 스텝 3 Enable/Disable (X 축: bHomMod3 [0], Y 축: bHomMod3 [1])	0: Disable 1: Enable
	int bHomMod3Dir[2]	원점 복귀 모드의 스텝 3 탐색 방향 (X 축: bHomMod3Dir [0], Y 축: bHomMod3Dir [1])	0: +방향 1: -방향
	int bHomMod4[2]	원점 복귀 모드의 스텝 4 Enable/Disable (X 축: bHomMod4 [0], Y 축: bHomMod4 [1])	0: Disable 1: Enable
	int bHomMod4Dir[2]	원점 복귀 모드의 스텝 4 탐색 방향 (X 축: bHomMod4Dir [0], Y 축: bHomMod4Dir [1])	0: +방향 1: -방향
	int bHomEndPosClr[2]	위치 카운터의 초기화 Enable/Disable (X 축: bHomEndPosClr [0], Y 축: bHomEndPosClr [1])	0: Disable 1: Enable
	int bHomSig0Lev[2]	원점 근접 신호(STOP 0) 논리 레벨 Low/High (X 축: bHomSig0Lev [0], Y 축: bHomSig0Lev [1])	0: Low 1: High
	int bHomSig1Lev[2]	원점 신호(STOP1) 논리 레벨 Low/High (X 축: bHomSig1Lev [0], Y 축: bHomSig1Lev [1])	0: Low 1: High
	int bHomSig2Lev[2]	엔코더 Z 상 신호(STOP2) 논리 레벨 Low/High (X 축: bHomSig2Lev [0], Y 축: bHomSig2Lev [1])	0: Low 1: High

구조체명	변수 형식	내용	데이터 값
	int iHomLowSpd[2]	저속 원점 복귀 속도 (X 축: iHomLowSpd [0], Y 축: iHomLowSpd [1])	1~8000
	int iHomHighSpd[2]	고속 원점 복귀 속도 (X 축: iHomHighSpd [0], Y 축: iHomHighSpd [1])	1~8000
	long lHomOffset[2]	원점 복귀 스텝 4 의 고속 오프셋 이동의 이동량 (X 축: lHomOffset [0], Y 축: lHomOffset [1])	0~8388607

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.2 autpmc_GetParaOPMAll

autpmc_GetParaOPMAll 함수는 PMC-2HSP 의 동작 모드 설정값을 가져옵니다.

(1) 함수명

```
int PMC_PARADATA *autpmc_GetParaOPMAll(
int PortNum,
char nNodeId,
char axis,
PMC_PARADATA *pData
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- pData

동작 모드에 관련된 모든 파라미터의 정보를 한 번에 가져와서 저장합니다.

구조체명	변수 형식	내용	데이터 값
PMC_ PARADATA	int iErrorState	오류 상태 확인	0: 함수가 정상적으로 명령을 수행하였습니다. 3: 잘못된 노드 ID 를 입력하였습니다. 4: 잘못된 축을 입력하였습니다. 5: 잘못된 데이터를 입력하였습니다.
	int bLmtStopMod[2]	리미트 정지 모드 즉시 정지(Instant)/ 감속 정지(Slow) (X 축: bLmtStopMod[0], Y 축: bLmtStopMod[1])	0: Instant 1: Slow
PMC_ PARADATA	int bLmtActLev[2]	리미트 신호 논리 레벨 Low/High (X 축: bLmtActLev [0], Y 축: bLmtActLev [1])	0: Low 1: High
	int bSCurve[2]	S 자 가감속의 사용 Enable/Disable (X 축: bSCurve [0], Y 축: bSCurve [1])	0: Disable 1: Enable
	int bEndPEnable[2]	드라이브 종료 펄스의 사용 여부 (X 축: bEndPEnable [0], Y 축: bEndPEnable [1])	0: Disable 1: Enable
	int bDecValue[2]	사다리꼴 가감속 드라이브의 대칭/비대칭 (X 축: bDecValue [0], Y 축: bDecValue [1])	0: Accel 1: Decel
	int bSofLmtEnable[2]	소프트웨어 리미트의 Enable/Disable (X 축: bSofLmtEnable [0], Y 축: bSofLmtEnable [1])	0: Enable 1: Disable

구조체명	변수 형식	내용	데이터 값
	int bPowHomStart[2]	파워 온 원점 복귀 자동 스타트 Enable/Disable (X 축: bPowHomStart [0], Y 축: bPowHomStart [1])	0: Disable 1: Enable
	int bPowPgmStart[2]	파워 온 프로그램 자동 스타트 Enable/Disable (X 축: bPowPgmStart [0], Y 축: bPowPgmStart [1])	0: Disable 1: Enable
	int bInput0Lev[2]	범용 입력 0 번의 액티브 레벨 Low/High (X 축: bInput0Lev [0], Y 축: bInput0Lev [1])	0: Low 1: High
	int bInput1Lev[2]	범용 입력 1 번의 액티브 레벨 Low/High (X 축: bInput1Lev [0], Y 축: bInput1Lev [1])	0: Low 1: High

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.3 autpmc_GetParaPMAll

autpmc_GetParaPMAll 함수는 PMC-2HSP 의 파라미터 설정값을 가져옵니다.

(1) 함수명

```
int PMC_PARADATA *autpmc_GetParaPMAll(
int PortNum,
char nNodeId,
char axis,
PMC_PARADATA *pData
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- pData

  파라미터에 관련된 모든 정보를 한 번에 가져와서 저장합니다.

구조체명	변수 형식	내용	데이터 값
PMC_ PARADATA	int iErrorState	오류 상태 확인	0: 함수가 정상적으로 명령을 수행하였습니다. 3: 잘못된 노드 ID 를 입력하였습니다. 4: 잘못된 축을 입력하였습니다. 5: 잘못된 데이터를 입력하였습니다.
	int iPulseType	펄스 입력 방식: 1PULSE, 2PULSE	1: 1PULSE 2: 2PULSE
	int iSpdMul[2]	속도 배율 (X 축: iSpdMul [0], Y 축: iSpdMul [1])	1~500
	int iJrkSpd[2]	가가속도 (X 축: iJrkSpd [0], Y 축: iJrkSpd [1])	1~65535
	int iAccSpdRate[2]	가속률 (X 축: iAccSpdRate [0], Y 축: iAccSpdRate [1])	1~8000
	int iDecSpdRate[2]	감속률 (X 축: iDecSpdRate [0], Y 축: iDecSpdRate [1])	1~8000
	int iStrSpd[2]	초기 속도 (X 축: iStrSpd [0], Y 축: iStrSpd [1])	1~8000
	int iDrvSpd[2]	구동 속도 (X 축: iDrvSpd [0], Y 축: iDrvSpd [1])	1~8000
	int iDrvSpd1Pgm[2]	프로그램 모드에서 사용하는 구동 속도 1 (X 축: iDrvSpd1Pgm [0], Y 축: iDrvSpd1Pgm [1])	1~8000
	int iDrvSpd2Pgm[2]	프로그램 모드에서 사용하는 구동 속도 2 (X 축: iDrvSpd2Pgm [0], Y 축: iDrvSpd2Pgm [1])	1~8000
	int iDrvSpd3Pgm[2]	프로그램 모드에서 사용하는 구동 속도 3 (X 축: iDrvSpd3Pgm [0], Y 축: iDrvSpd3Pgm [1])	1~8000

구조체명	변수 형식	내용	데이터 값
	int iDrvSpd4Pgm[2]	프로그램 모드에서 사용하는 구동 속도 4 (X 축: iDrvSpd4Pgm [0], Y 축: iDrvSpd4Pgm [1])	1~8000
	int iTim1Pgm[2]	프로그램 모드에서 사용하는 포스트 타이머 1 (X 축: iTim1Pgm [0], Y 축: iTim1Pgm [1])	1~65535
	int iTim2Pgm[2]	프로그램 모드에서 사용하는 포스트 타이머 2 (X 축: iTim2Pgm [0], Y 축: iTim2Pgm [1])	1~65535
	int iTim3Pgm[2]	프로그램 모드에서 사용하는 포스트 타이머 3 (X 축: iTim3Pgm [0], Y 축: iTim3Pgm [1])	1~65535
	long lSofLmtP[2]	+방향 소프트웨어 리미트 (X 축: lSofLmtP [0], Y 축: lSofLmtP [1])	-8388608 ~8388607
	long lSofLmtM[2]	-방향 소프트웨어 리미트 (X 축: lSofLmtM [0], Y 축: lSofLmtM [1])	-8388608 ~8388607
	int iEndPWidth[2]	드라이브 종료 펄스의 폭 (X 축: iEndPWidth [0], Y 축: iEndPWidth [1])	1~65535
	int iPulSclNum[2]	펄스 스케일의 분자값 (X 축: iPulSclNum [0], Y 축: iPulSclNum [1])	1~65535
	int iPulSclDen[2]	펄스 스케일의 분모값 (X 축: iPulSclDen [0], Y 축: iPulSclDen [1])	1~65535

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.4 autpmc_GetParaHSMAll

autpmc_GetParaHSMAll 함수는 PMC-2HSP 에 설정된 원점 복귀 모드에 관련된 파라미터를 모두 가져옵니다.

(1) 함수명

```
int PMC_PARADATA *autpmc_GetParaHSMAll(
int PortNum,
char nNodeId,
char axis,
PMC_PARADATA *pData
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- pData

원점 복귀에 관련된 모든 파라미터의 정보를 한 번에 가져와서 저장합니다.

구조체명	변수 형식	내용	데이터 값
PMC_ PARADATA	int iErrorState	오류 상태 확인	0: 함수가 정상적으로 명령을 수행하였습니다. 3: 잘못된 노드 ID 를 입력하였습니다. 4: 잘못된 축을 입력하였습니다. 5: 잘못된 데이터를 입력하였습니다.
	int bHomMod1[2]	원점 복귀 모드의 스텝 1 Enable/Disable (X 축: bHomMod1 [0], Y 축: bHomMod1 [1])	0: Disable 1: Enable
PMC_ PARADATA	int bHomMod1Dir[2]	원점 복귀 모드의 스텝 1 탐색 방향 (X 축: bHomMod1Dir [0], Y 축: bHomMod1Dir [1])	0: +방향 1: -방향
	int bHomMod2[2]	원점 복귀 모드의 스텝 2 Enable/Disable (X 축: bHomMod2 [0], Y 축: bHomMod2 [1])	0: Disable 1: Enable
	int bHomMod2Dir[2]	원점 복귀 모드의 스텝 2 탐색 방향 (X 축: bHomMod2Dir [0], Y 축: bHomMod2Dir [1])	0: + 1: -
	int bHomMod3[2]	원점 복귀 모드의 스텝 3 Enable/Disable (X 축: bHomMod3 [0], Y 축: bHomMod3 [1])	0: Disable 1: Enable
	int bHomMod3Dir[2]	원점 복귀 모드의 스텝 3 탐색 방향 (X 축: bHomMod3Dir [0], Y 축: bHomMod3Dir [1])	0: + 1: -
	int bHomMod4[2]	원점 복귀 모드의 스텝 4 Enable/Disable (X 축: bHomMod4 [0], Y 축: bHomMod4 [1])	0: Disable 1: Enable
	int bHomMod4Dir[2]	원점 복귀 모드의 스텝 4 탐색 방향 (X 축: bHomMod4Dir [0], Y 축: bHomMod4Dir [1])	0: + 1: -

구조체명	변수 형식	내용	데이터 값
	int bHomEndPosClr[2]	위치 카운터의 초기화 Enable/Disable (X 축: bHomEndPosClr [0], Y 축: bHomEndPosClr [1])	0: Disable 1: Enable
	int bHomSig0Lev[2]	원점 근접 신호(STOP 0) 논리 레벨 Low/High (X 축: bHomSig0Lev [0], Y 축: bHomSig0Lev [1])	0: Low 1: High
	int bHomSig1Lev[2]	원점 신호(STOP1) 논리 레벨 Low/High (X 축: bHomSig1Lev [0], Y 축: bHomSig1Lev [1])	0: Low 1: High
	int bHomSig2Lev[2]	엔코더 Z 상 신호(STOP2) 논리 레벨 Low/High (X 축: bHomSig2Lev [0], Y 축: bHomSig2Lev [1])	0: Low 1: High
	int iHomLowSpd[2]	저속 원점 복귀 속도 (X 축: iHomLowSpd [0], Y 축: iHomLowSpd [1])	1~8000
	int iHomHighSpd[2]	고속 원점 복귀 속도 (X 축: iHomHighSpd [0], Y 축: iHomHighSpd [1])	1~8000
	long lHomOffset[2]	원점 복귀 스텝 4 의 고속 오프셋 이동의 이동량 (X 축: lHomOffset [0], Y 축: lHomOffset [1])	0~8388607

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.5 autpmc_GetLmtStopMod

autpmc_GetLmtStopMod 함수는 PMC-2HSP 에 설정된 리미트 정지 모드를 가져옵니다.

(1) 함수명

```
int autpmc_GetLmtStopMod(
int PortNum,
char nNodeId,
char axis,
int *bStopType
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bStopType

  즉시 정지로 설정되어 있으면 FPMC_INSTANTSTOP(0)을 감속 정지로 설정되어 있으면 FPMC_SLOWSTOP(1)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.6 autpmc_GetLmtActLev

autpmc_GetLmtActLev 함수는 PMC-2HSP 에 설정된 리미트 신호 논리 레벨을 가져옵니다.

(1) 함수명

```
int autpmc_GetLmtActLev(
int PortNum,
char nNodeId,
char axis,
int *bLevel
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bLevel
  설정된 리미트 입력 신호 논리 레벨이 High 일 때는 FPMC_HIGH(1)을 Low 일 때는 FPMC_LOW(0)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.7 autpmc_GetSCurve

autpmc_GetSCurve 함수는 PMC-2HSP 에 설정된 S 자 가감속의 사용 여부를 가져옵니다.

(1) 함수명

```
int autpmc_GetSCurve(
int PortNum,
char nNodeId,
char axis,
int *bEnable
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable
  S 자 가감속이 설정되어 있으면 FPMC_ENABLE(1)을 설정되어 있지 않으면 FPMC_DISABLE(0)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.8 autpmc_GetEndPEnable

autpmc_GetEndPEnable 함수는 PMC-2HSP 에 설정된 드라이브 종료 펄스의 사용 여부를 가져옵니다.

(1) 함수명

```
int autpmc_GetEndPEnable(
int PortNum,
char nNodeId,
char axis,
int *bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  드라이브 종료 펄스가 설정되어 있으면 FPMC_ENABLE(1)을 설정되어 있지 않으면 FPMC_DISABLE(0)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.9 autpmc_GetDecValue

autpmc_GetDecValue 함수는 PMC-2HSP 에서 사다리꼴 가감속 드라이브의 설정값을 가져옵니다.

(1) 함수명

```
int autpmc_GetDecValue(
int PortNum,
char nNodeId,
char axis,
int *bDec
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDec

  가속도가 설정되어 있으면 FPMC_ACCEL(0)을 감속도가 설정되어 있으면 FPMC_DECEL(1)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.10 autpmc_GetSofLmtEnable

autpmc_GetSofLmtEnable 함수는 PMC-2HSP 에서 소프트웨어 리미트의 사용 여부를 가져옵니다.

(1) 함수명

```
int autpmc_GetSofLmtEnable(
int PortNum,
char nNodeId,
char axis,
int *bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  소프트웨어 리미트가 설정되어 있으면 FPMC_ENABLE(0)을 설정되어 있지 않으면 FPMC_DISABLE(1)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.11 autpmc_GetPowHomStart

autpmc_GetPowHomStart 함수는 PMC-2HSP 에서 파워 온 원점 복귀 자동 스타트의 사용 여부를 가져옵니다.

(1) 함수명

```
int autpmc_GetPowHomStart(
int PortNum,
char nNodeId,
char axis,
int *bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  파워 온 원점 복귀 자동 스타트가 설정되어 있다면 FPMC_ENABLE(1)을 설정되어 있지 않다면 FPMC_DISABLE(0)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.12 autpmc_GetPowPgmStart

autpmc_GetPowPgmStart 함수는 PMC-2HSP 에서 파워 온 프로그램 자동 스타트의 사용 여부를 가져옵니다.

(1) 함수명

```
int autpmc_GetPowPgmStart(
int PortNum,
char nNodeId,
char axis,
int *bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  파워 온 프로그램 자동 스타트가 설정되어 있다면 FPMC_ENABLE(1)을 설정되어 있지 않다면 FPMC_DISABLE(0)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.13 autpmc_GetInputLev

autpmc_GetInputLev 함수는 PMC-2HSP 에서 범용 입력 0, 1 번의 액티브 레벨을 가져옵니다.

(1) 함수명

```
int autpmc_GetInputLev(
int PortNum,
char nNodeId,
char axis,
int bInPort,
int *bActLev
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bInPort
  범용 입력 0 번을 선택하려면 FPMC_INPORT0(0)을 입력하고 1 번을 선택하려면 FPMC_INPORT1(1)을 입력합니다.
- bActLev
  액티브 레벨이 Low 로 설정되어 있다면 FPMC_LOW(0)을 High 로 설정되어 있다면 FPMC_HIGH(1)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.14 autpmc_GetPulseType

autpmc_GetPulseType 함수는 PMC-2HSP 에서 펄스 입력 방식(1PULSE/2PULSE)을 가져옵니다.

(1) 함수명

```
int autpmc_GetPulseType(
int PortNum,
char nNodeId,
int *iPulseType
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- iPulseType
  펄스 입력 방식이 1PULSE 입력 방식일 경우에는 FPMC_1PULSETYPE(1)을 2PULSE 입력 방식이 경우에는 FPMC_2PULSETYPE(2)를 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

3.15 autpmc_GetSpdMul

autpmc_GetSpdMul 함수는 PMC-2HSP 에서 속도 배율을 가져옵니다.

(1) 함수명

```
int autpmc_GetSpdMul(
int PortNum,
char nNodeId,
char axis,
int *iSpdMul
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iSpdMul

  설정된 속도 배율을 가져옵니다. (1~500)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.16 autpmc_GetJrkSpd

autpmc_GetJrkSpd 함수는 PMC-2HSP 에서 가가속도를 가져옵니다.

(1) 함수명

```
int autpmc_GetJrkSpd(
int PortNum,
char nNodeId,
char axis,
int *iJrkSpd
);
```

(2) 파라미터

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iJrkSpd

  설정된 가가속도를 가져옵니다. (1~65,535)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.17 autpmc_GetAccSpdRate

autpmc_GetAccSpdRate 함수는 PMC-2HSP 에서 가속률을 가져옵니다.

(1) 함수명

```
int autpmc_GetAccSpdRate(
int PortNum,
char nNodeId,
char axis,
int *iAccSpdRate
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iAccSpdRate

  설정된 가속률을 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.18 autpmc_GetDecSpdRate

autpmc_GetDecSpdRate 함수는 PMC-2HSP 에서 가속률을 가져옵니다.

(1) 함수명

```
int autpmc_SetDecSpdRate(
int PortNum,
char nNodeId,
char axis,
int *iDecSpdRate
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iDecSpdRate

  설정된 가속률을 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.19 autpmc_GetStrSpd

autpmc_GetStrSpd 함수는 PMC-2HSP 에서 초기 속도를 가져옵니다.

(1) 함수명

```
int autpmc_GetStrSpd(
int PortNum,
char nNodeId,
char axis,
int *iStrSpd
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iStrSpd

  설정된 초기 속도를 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.20 autpmc_ GetCurDrvSpd

autpmc_GetDrvSpd 함수는 PMC-2HSP 에서 현재 기동중인 구동 속도를 가져옵니다.

(1) 함수명

```
int autpmc_GetCurDrvSpd(
int PortNum,
char nNodeId,
char axis,
int *iDrvSpd_X,
int *iDrvSpd_Y
);
```

(2) 파라미터

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iDrvSpd_X

  설정된 X 축 구동 속도를 가져옵니다. (1~8,000)

- iDrvSpd_Y

  설정된 Y 축 구동 속도를 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.21 autpmc_GetDrvSpdPgm

autpmc_GetDrvSpdPgm 함수는 PMC-2HSP 에서 구동 속도를 가져오는 함수로서 프로그램 모드에서 사용됩니다.

(1) 함수명

```
int autpmc_GetDrvSpdPgm(
int PortNum,
char nNodeId,
char axis,
int nDrvIndex,
int *iDrvSpd
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nDrvIndex
  드라이브 속도 인덱스를 입력합니다. 이때 유효한 값은 1~4 입니다.
- iDrvSpd
  설정된 구동 속도를 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.22 autpmc_GetTimPgm

autpmc_GetTimPgm 함수는 PMC-2HSP 에서 포스트 타이머를 가져오는 함수로서 프로그램 모드에서 사용됩니다.

(1) 함수명

```
int autpmc_GetTimPgm(
int PortNum,
char nNodeId,
char axis,
int nIndex,
int *iPostTim
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nIndex
  포스트 타이머 인덱스를 입력합니다. 이때 유효한 값은 1~3 입니다.
- iPostTim
  설정된 포스트 타이머를 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

3.23 autpmc_GetSofLmt

autpmc_GetSofLmt 함수는 PMC-2HSP 에서 소프트웨어 리미트를 가져옵니다.
(펄스 스케일값에 비례합니다.)

(1) 함수명

```
int autpmc_GetSofLmt(
int PortNum,
char nNodeId,
char axis,
int iDirection,
long *lSoftLmt
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iDirection
  '+' 방향의 리미트를 설정하려면 FPMC_PLUS(0)을 '-' 방향의 리미트를 선택하려면 FPMC_MINUS(1)을 입력합니다.
- lSoftLmt
  설정된 소프트웨어 리미트를 가져옵니다. (-8,388,608~8,388,607)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.24 autpmc_GetEndPWidth

autpmc_GetEndPWidth 함수는 PMC-2HSP 에서 드라이브 종료 펄스의 폭을 가져옵니다.

(1) 함수명

```
int autpmc_GetEndPWidth(
int PortNum,
char nNodeId,
char axis,
int *iEndPWidth
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iEndPWidth

  설정된 드라이브 종료 펄스의 폭을 가져옵니다. (1~65,535)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.25 autpmc_GetPulSclNum

autpmc_GetPulSclNum 함수는 PMC-2HSP 에서 펄스 스케일의 분자값을 가져옵니다.

(1) 함수명

```
int autpmc_GetPulSclNum(
int PortNum,
char nNodeId,
char axis,
int *iPulScl
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iPulScl

  설정된 펄스 스케일의 분자값을 가져옵니다. (1~65,535)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.26 autpmc_GetPulSclDen

autpmc_GetPulSclDen 함수는 PMC-2HSP 에서 펄스 스케일의 분모값을 가져옵니다.

(1) 함수명

```
int autpmc_GetPulSclDen(
int PortNum,
char nNodeId,
char axis,
int *iPulScl
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iPulScl

  설정된 펄스 스케일의 분모값을 가져옵니다. (1~65,535)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.27 autpmc_GetHomMod

autpmc_GetHomMod 함수는 PMC-2HSP 에서 원점 복귀 모드를 가져옵니다.

(1) 함수명

```
int HOMMOD *autpmc_GetHomMod(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
HOMMOD *pMode
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  설정할 스텝 번호를 입력합니다. 이때 유효한 값은 1~4 입니다.

- pMode

  설정된 원점 복귀 모드의 ENABLE 여부와 방향을 가져옵니다.

구조체명	변수 형식	내용	데이터 값
HOMMODE	int bEnable[2] bEable[0]: X 축 bEable[1]: Y 축	원점복귀 사용	FPMC_ENABLE(1)/ FPMC_DISABLE(0)
	int bDirection[2] bDirection [0]: X 축 bDirection [1]: Y 축	원점복귀 검색 방향	FPMC_PLUS(0)/ FPMC_MINUS(1)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.28 autpmc_GetHomEndPosClr

autpmc_GetHomEndPosClr 함수는 PMC-2HSP 에서 원점 복귀 종료 시 위치 카운터의 사용 여부를 가져옵니다.

(1) 함수명

```
int autpmc_GetHomEndPosClr(
int PortNum,
char nNodeId,
char axis,
int *bClear
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bClear

  위치 카운터 초기화가 사용되고 있으면 FPMC_ENABLE(1)을 사용하지 않고 있으면 FPMC_DISABLE(0)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.29 autpmc_GetHomSigLev

autpmc_GetHomSigLev 함수는 PMC-2HSP 에서 원점 신호의 논리 레벨을 가져옵니다.

(1) 함수명

```
int autpmc_GetHomSigLev(
int PortNum,
char nNodeId,
char axis,
int nHomSigNo,
int *bLevel
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nHomSigNo

  원점 근접 신호를 선택하려면 FPMC_HSTOP0(0)을 원점 신호를 선택하려면 FPMC_HSTOP1(1)을 엔코더 Z 상 신호를 선택하려면 FPMC_HSTOP2(2)를 입력합니다.

- bLevel

  원점 신호의 논리 레벨이 Low 로 설정되어 있으면 FPMC_LOW(0)을 High 로 설정되어 있으면 FPMC_HIGH(1)을 가져옵니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.30 autpmc_GetHomSpd

autpmc_GetHomSpd 함수는 PMC-2HSP 에서 원점 복귀 속도를 가져옵니다.

(1) 함수명

```
int autpmc_GetHomSpd(
int PortNum,
char nNodeId,
char axis,
int bSpd,
int *iSpd
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bSpd

  저속 원점 복귀 속도를 가져올려면 FPMC_LOW(0)을 고속 원점 복귀 속도를 가져올려면 FPMC_HIGH(1)을 설정합니다.

- iSpd

  설정된 원점 복귀 속도를 가져옵니다. (1~8,000)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.31 autpmc_GetHomOffset

autpmc_GetHomOffset 함수는 PMC-2HSP 에서 원점 복귀 스텝 4 의 고속 오프셋 이동의 이동량을 가져옵니다.(펄스 스케일값에 비례됩니다.)

(1) 함수명

```
int autpmc_GetHomOffset(
int PortNum,
char nNodeId,
char axis,
long *lOffset
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- lOffset

  설정된 원점 복귀 스텝 4 의 고속 오프셋 이동량을 가져옵니다. (0~8,388,607)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.32 autpmc_SetLmtStopMod

autpmc_SetLmtStopMod 함수는 PMC-2HSP 의 리미트 정지 모드를 설정합니다.

(1) 함수명

```
int autpmc_SetLmtStopMod(
int PortNum,
char nNodeId,
char axis,
int bInstant
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bInstant

  Instant 모드일 때는 FPMC_INSTANT(0)을 Slow 모드일 때는 FPMC_SLOW(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.33 autpmc_SetLmtActLev

autpmc_SetLmtActLev 함수는 PMC-2HSP 의 리미트 신호 논리 레벨를 활성화합니다.

(1) 함수명

```
int autpmc_SetLmtActLev(
int PortNum,
char nNodeId,
char axis,
int bLmtActLev
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bLmtActLev

  리미트 입력 신호 논리 레벨을 High 일 때 활성화시킬 경우에는 FPMC_HIGH(1)을 Low 일 때 활성화시킬 경우에는 FPMC_LOW(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.34 autpmc_SetSCurve

autpmc_SetSCurve 함수는 PMC-2HSP 의 S 자 가감속의 사용 여부를 설정합니다.

(1) 함수명

```
int autpmc_SetSCurve(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  S 자 가감속을 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.35 autpmc_SetEndPEnable

autpmc_SetEndPEnable 함수는 PMC-2HSP 의 드라이브 종료 펄스의 사용 여부를 설정합니다.

(1) 함수명

```
int autpmc_SetEndPEnable(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  드라이브 종료 펄스를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.36 autpmc_SetDecValue

autpmc_SetDecValue 함수는 PMC-2HSP 의 사다리꼴 가감속 드라이브의 대칭/비대칭을 설정합니다.

(1) 함수명

```
int autpmc_SetDecValue(
int PortNum,
char nNodeId,
char axis,
int bDec
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDec

  가속도를 사용할 시 FPMC_ACCEL(0)을 감속도를 사용할 시 FPMC_DECEL(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.37 autpmc_SetSofLmtEnable

autpmc_SetSofLmtEnable 함수는 PMC-2HSP 의 소프트웨어 리미트의 사용 여부를 설정합니다.

(1) 함수명

```
int autpmc_SetSofLmtEnable(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  소프트웨어 리미트를 사용하려면 FPMC_ENABLE(0)을 사용하지 않으려면 FPMC_DISABLE(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.38 autpmc_SetPowHomStart

autpmc_SetPowHomStart 함수는 PMC-2HSP 의 파워 온 원점 복귀 자동 스타트의 사용 여부를 설정합니다.

(1) 함수명

```
int autpmc_SetPowHomStart(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  파워 온 원점 복귀 자동 스타트를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.39 autpmc_SetPowPgmStart

autpmc_SetPowPgmStart 함수는 PMC-2HSP 의 파워 온 프로그램 자동 스타트의 사용 여부를 설정합니다.

(1) 함수명

```
int autpmc_SetPowPgmStart(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  파워 온 프로그램 자동 스타트를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.40 autpmc_SetInputLev

autpmc_SetInputLev 함수는 PMC-2HSP 의 범용 입력 0, 1 번의 액티브 레벨을 설정합니다.

(1) 함수명

```
int autpmc_SetInputLev(
int PortNum,
char nNodeId,
char axis,
int bInPort,
int bActLev
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bInPort

  범용 입력 0 번을 설정할려면 FPMC_INPORT(0)을 범용 입력 1 번을 설정할려면 FPMC_INPORT1(1)을 설정합니다.

- bActLev

  액티브 레벨을 Low 로 설정하려면 FPMC_LOW(0)을 High 로 설정하려면 FPMC_HIGH(1)을 설정합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.41 autpmc_SetPulseType

autpmc_SetPulseType 함수는 PMC-2HSP 의 펄스 입력 방식(1PULSE/2PULSE)을 선택합니다.

(1) 함수명

```
int autpmc_SetPulseType(
int PortNum,
char nNodeId,
int iPulseType
);
```

(2) 파라미터

- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- iPulseType
  1PULSE 입력 방식일 경우에는 FPMC_SETPULSE1(1)을 2PULSE 입력 방식일 경우에는 FPMC_SETPULSE2(2)를 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.42 autpmc_SetSpdMul

autpmc_SetSpdMul 함수는 PMC-2HSP 의 속도 배율를 설정합니다.

(1) 함수명

```
int autpmc_SetSpdMul(
int PortNum,
char nNodeId,
char axis,
int iSpdMul
);
```

(2) 파라미터

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iSpdMul

  속도 배율을 입력합니다. 이때 유효한 값은 1~500 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.43 autpmc_SetJrkSpd

autpmc_SetJrkSpd 함수는 PMC-2HSP 의 가가속도를 설정합니다.

(1) 함수명

autpmc_SetJrkSpd(
int PortNum,
char nNodeId,
char axis,
int iJrkSpd
);

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iJrkSpd

  가가속도를 입력합니다. 이때 유효한 값은 1~65,535 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.44 autpmc_SetAccSpdRate

autpmc_SetAccSpdRate 함수는 PMC-2HSP 의 가속률을 설정합니다.

(1) 함수명

```
int autpmc_SetAccSpdRate(
int PortNum,
char nNodeId,
char axis,
int iAccSpdRate
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iAccSpdRate

  가속률을 입력합니다. 이때 유효한 값은 1~8,000 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.45 autpmc_SetDecSpdRate

autpmc_SetDecSpdRate 함수는 PMC-2HSP 의 감속률을 설정합니다.

(1) 함수명

```
int autpmc_SetDecSpdRate(
int PortNum,
char nNodeId,
char axis,
int iDecSpdRate
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iDecSpdRate

  감속률을 입력합니다. 이때 유효한 값은 1~8,000 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.46 autpmc_SetStrSpd

autpmc_SetStrSpdRate 함수는 PMC-2HSP 의 초기 속도를 설정합니다.

(1) 함수명

```
int autpmc_SetStrSpd(
int PortNum,
char nNodeId,
char axis,
int iStrSpd
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iStrSpd

  초기 속도를 입력합니다. 이때 유효한 값은 1~8,000 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.47 autpmc_SetCurDrvSpd

autpmc_SetCurDrvSpd 함수는 PMC-2HSP 의 현재 구동 속도를 설정합니다.

(1) 함수명

```
int autpmc_SetCurDrvSpd(
int PortNum,
char nNodeId,
char axis,
int iDrvSpd_X,
int iDrvSpd_Y
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iDrvSpd_X

  구동할 X 축 드라이버 속도를 입력합니다. 이때 유효한 값은 1~8,000 입니다.

- iDrvSpd_Y

  구동할 Y 축 드라이버 속도를 입력합니다. 이때 유효한 값은 1~8,000 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.48 autpmc_ SetDrvSpd

autpmc_SetDrvSpd 함수는 조그/연속/위치 운전 시 PMC-2HSP 의 파라미터에 설정된 구동 속도(DriveSpeed1 ~ DriveSpeed4)를 선택합니다

(1) 함수명

```
int autpmc_SetDrvSpd(
int PortNum,
char nNodeId,
char axis,
int nDrvIndex
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nDrvIndex

  드라이브 속도 인덱스를 입력합니다. 이때 유효한 값은 1~4 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.49 autpmc_SetDrvSpdPgm

autpmc_SetDrvSpdPgm 함수는 PMC-2HSP 의 구동 속도를 설정하는 함수로서 프로그램 모드에서 사용됩니다.

(1) 함수명

```
int autpmc_SetDrvSpdPgm(
int PortNum,
char nNodeId,
char axis,
int nDrvIndex,
int iDrvSpd
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nDrvIndex

  드라이브 속도 인덱스를 입력합니다. 이때 유효한 값은 1~4 입니다.

- iDrvSpd

  구동 속도를 입력합니다. 이때 유효한 값은 1~8,000 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.50 autpmc_SetTimPgm

autpmc_SetTimPgm 함수는 PMC-2HSP 의 포스트 타이머를 설정하는 함수로서 프로그램 모드에서 사용됩니다.

(1) 함수명

```
int autpmc_SetTimPgm(
int PortNum,
char nNodeId,
char axis,
int nIndex,
int iPostTim
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nIndex

  사용할 포스트 타이머를 선택합니다. 이때 유효한 값은 1~3 입니다.

- iPostTim

  포스트 타이머를 입력합니다. 이때 유효한 값은 1~65,535 이며, 값의 단위는 ms 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.51 autpmc_SetSofLmt

autpmc_SetSofLmt 함수는 PMC-2HSP 의 소프트웨어 리미트를 설정합니다.
(펄스 스케일값에 비례됩니다.)

(1) 함수명

```
int autpmc_SetSftLmt(
int PortNum,
char nNodeId,
char axis,
int iDirection,
long lSoftLmt
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iDirection
  각 축의 '+' 방향과 '–' 방향을 선택합니다.
  각 축의 소프트웨어 리미트 '+' 방향을 설정하실려면 FPMC_SOFLMT_PLUS(0)을 소프트웨어 리미트 '-' 방향을 설정하실려면 FPMC_SOFLMT_MINUS(1)을 입력합니다.
- lSoftLmt
  소프트웨어 리미트를 입력합니다. 이때 유효한 값은 -8,388,608~8,388,607 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.52 autpmc_SetEndPWidth

autpmc_SetEndPWidth 함수는 PMC-2HSP 의 드라이브 종료 펄스의 폭을 설정합니다.

(1) r 함수명

```
int autpmc_SetEndPWidth(
int PortNum,
char nNodeId,
char axis,
int iEndPWidth
);
```

(2) 파라미터

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iEndPWidth

  드라이브 종료 펄스의 폭을 입력합니다. 이때 유효한 값은 1~65,535 이며, 값의 단위는 ms 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.53 autpmc_SetPulSclNum

autpmc_SetPulScl 함수는 PMC-2HSP 의 펄스 스케일의 분자값을 설정합니다.

(1) 함수명

```
int autpmc_SetPulSclNum(
int PortNum,
char nNodeId,
char axis,
int iPulScl
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iPulScl

  펄스 스케일의 분자값을 입력합니다. 이때 유효한 값은 1~65,535 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.54 autpmc_SetPulSclDen

autpmc_SetPulScl 함수는 PMC-2HSP 의 펄스 스케일의 분모값을 설정합니다.

(1) 함수명

```
int autpmc_SetPulSclDen(
int PortNum,
char nNodeId,
char axis,
int iPulScl
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iPulScl

  펄스 스케일의 분모값을 입력합니다. 이때 유효한 값은 1~65,535 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.55 autpmc_SetHomMod

autpmc_SetHomMod 함수는 PMC-2HSP 의 원점 복귀 모드를 설정합니다.

(1) 함수명

```
int autpmc_SetHomMod(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int bEnable,
int bDirection
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  설정할 스텝 번호를 입력합니다. 이때 유효한 값은 1~4 입니다.

- bEnable

  원점 복귀를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

- bDirection

  원점 복귀 시 검색 방향을 '+'로 설정하려면 FPMC_PLUS(0)을 '-'로 설정하려면 FPMC_MINUS(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.56 autpmc_HomStop

autpmc_HomStop 함수는 PMC-2HSP 에서 원점 복귀 모드를 종료합니다.

(1) 함수명

```
int autpmc_HomStop(
int PortNum,
char nNodeId,
char axis
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 을 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

3.57 autpmc_Step1Enable

autpmc_Step1Enable 함수는 PMC-2HSP 의 원점 복귀 모드에서 스텝 1 의 사용 유무를 설정합니다.

(1) 함수명

```
int autpmc_Step1Enable(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  원점 복귀를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.58 autpmc_Step2Enable

autpmc_Step2Enable 함수는 PMC-2HSP 의 원점 복귀 모드에서 스텝 2 의 사용 유무를 설정합니다.

(1) 함수명

```
int autpmc_Step2Enable(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  원점 복귀를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.59 autpmc_Step3Enable

autpmc_Step3Enable 함수는 PMC-2HSP 의 원점 복귀 모드에서 스텝 3 의 사용 유무를 설정합니다.

(1) 함수명

```
int autpmc_Step3Enable(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  원점 복귀를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.60 autpmc_Step4Enable

autpmc_Step4Enable 함수는 PMC-2HSP 의 원점 복귀 모드에서 스텝 4 의 사용 유무를 설정합니다.

(1) 함수명

int autpmc_Step4Enable(
int PortNum,
char nNodeId,
char axis,
int bEnable
);

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable
  원점 복귀를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.61 autpmc_Step1Direction

autpmc_Step1Direction 함수는 PMC-2HSP 의 원점 복귀 시 스텝 1 의 검색 방향을 설정합니다.

(1) 함수명

```
int autpmc_Step1Direction(
int PortNum,
char nNodeId,
char axis,
int bDirection
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDirection

  원점 복귀 시 검색 방향을 '+'로 설정하려면 FPMC_PLUS(0)을 '-'로 설정하려면 FPMC_MINUS(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.62 autpmc_Step2Direction

autpmc_Step2Direction 함수는 PMC-2HSP 의 원점 복귀 시 스텝 2 의 검색 방향을 설정합니다.

(1) 함수명

```
int autpmc_Step2Direction(
int PortNum,
char nNodeId,
char axis,
int bDirection
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDirection

  원점 복귀 시 검색 방향을 '+'로 설정하려면 FPMC_PLUS(0)을 '-'로 설정하려면 FPMC_MINUS(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.63 autpmc_Step3Direction

autpmc_Step3Direction 함수는 PMC-2HSP 의 원점 복귀 시 스텝 3 의 검색 방향을 설정합니다.

(1) 함수명

```
int autpmc_Step3Direction(
int PortNum,
char nNodeId,
char axis,
int bDirection
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDirection

  원점 복귀 시 검색 방향을 '+'로 설정하려면 FPMC_PLUS(0)을 '-'로 설정하려면 FPMC_MINUS(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.64 autpmc_Step4Direction

autpmc_Step4Direction 함수는 PMC-2HSP 의 원점 복귀 시 스텝 4 의 검색 방향을 설정합니다.

(1) 함수명

```
int autpmc_Step4Direction(
int PortNum,
char nNodeId,
char axis,
int bDirection
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDirection

  원점 복귀 시 검색 방향을 '+'로 설정하려면 FPMC_PLUS(0)을 '-'로 설정하려면 FPMC_MINUS(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.65 autpmc_SetHomEndPosClr

autpmc_SetHomEndPosClr 함수는 PMC-2HSP 의 원점 복귀 종료 시 위치 카운터를 초기화합니다.

(1) 함수명

```
int autpmc_SetHomEndPosClr(
int PortNum,
char nNodeId,
char axis,
int bEnable
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bEnable

  위치 카운터 초기화를 사용하려면 FPMC_ENABLE(1)을 사용하지 않으려면 FPMC_DISABLE(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.66 autpmc_SetHomSigLev

autpmc_SetHomSigLev 함수는 PMC-2HSP 의 원점 신호의 논리 레벨을 설정합니다.

(1) 함수명

```
int autpmc_SetHomSigLev(
int PortNum,
char nNodeId,
char axis,
int nHomSigNo,
int bLevel
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nHomSigNo

  원점 근접 신호를 선택하려면 FPMC_HSTOP0(0)을 원점 신호를 선택하려면 FPMC_HSTOP1(1)을 엔코더 Z 상 신호를 선택하려면 FPMC_HSTOP2(2)를 입력합니다.

- bLevel

  Low 로 설정하려면 FPMC_LOW(0)을 High 로 설정하려면 FPMC_HIGH(1)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.67 autpmc_SetHomSpd

autpmc_SetHomSpd 함수는 PMC-2HSP 의 원점 복귀 속도를 설정합니다.

(1) 함수명

```
int autpmc_SetHomSpd(
int PortNum,
char nNodeId,
char axis,
int bSpd,
int iSpd
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bSpd

  저속 원점 복귀 속도를 설정하려면 FPMC_LOW(0)을 고속 원점 복귀 속도를 설정하려면 FPMC_HIGH(1)을 설정합니다.

- iSpd

  원점 복귀 속도를 입력합니다. 이때 유효한 값은 1~8,000 입니다.

(3) 리턴값

구분	정의	리턴값	내용
—	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

3.68 autpmc_SetHomOffset

autpmc_SetHomOffset 함수는 PMC-2HSP 의 원점 복귀 스텝 4 의 고속 오프셋 이동의 이동량을 설정합니다. (펄스 스케일값에 비례됩니다.)

(1) 함수명

```
int autpmc_SetHomOffset(
int PortNum,
char nNodeId,
char axis,
long lOffset
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- lOffset

  원점 복귀 스텝 4 의 고속 오프셋 이동의 이동량을 입력합니다. 이때 유효한 값은 0~8,388,607 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력했습니다.

4 I/O 제어

4.1 autpmc_GetParallelIO

autpmc_GetParallelIO 함수는 PMC-2HSP 의 Parallel I/F 커넥터(CN3) 입력 신호를 읽습니다.

(1) 함수명

```
int PARALLELSTATE *autpmc_GetParallelIO(
int PortNum,
char nNodeId,
PARALLELSTATE *pState
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- pState

  Parallel I/F 의 신호를 읽어 와서 현재 변수에 저장합니다.

구조체명	변수 형식	내용	데이터 값
PARALLELSTATE	int HOME;	원점복귀 시작	0: OFF/ 1: ON
	int STROBE;	드라이브 시작	0: OFF/ 1: ON
	int X;	X 축 지정/ 조그 2 모드 Y+	0: OFF/ 1: ON
	int Y;	Y 축 지정/ 조그 2 모드 Y+	0: OFF/ 1: ON
	int MODE0;	스텝 지정 0/ 런+/ 조그 2 모드 X+	0: OFF/ 1: ON
	int MODE1;	스텝 지정 1/ 런-/ 조그 2 모드 X-	0: OFF/ 1: ON
	int STEPSL0;	스텝 지정 2/ 드라이브 속도 지정 0	0: OFF/ 1: ON
	int STEPSL1;	스텝 지정 3/ 드라이브 속도 지정 1	0: OFF/ 1: ON
	int STEPSL2;	스텝 지정 4/ 조그 지정	0: OFF/ 1: ON
	int STEPSL3;	스텝 지정 5/ 드라이브 정지	0: OFF/ 1: ON
	int STEPSL4;	운전 모드 지정 0	0: OFF/ 1: ON
	int STEPSL5;	운전 모드 지정 1	0: OFF/ 1: ON

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

4.2 autpmc_GetAxisIO

autpmc_GetAxisIO 함수는 PMC-2HSP 의 X 축(CN4)과 Y 축(CN5)의 입/출력 커넥터 신호를 읽습니다.

(1) 함수명

```
int AXISSTATE *autpmc_GetAxisIO(
int PortNum,
char nNodeId,
char axis,
AXISSTATE *pState
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

▪ pState

CN4 번 X 축 또는 CN5 번 Y 축의 입/출력 신호를 읽어 와서 현재 변수에 저장을 합니다.

구조체명	변수 형식	내용	데이터 값
PARALLELSTATE	int bHomSig0[2];	원점 근접 (X 축: bHomSig0 [0], Y 축: bHomSig0 [1])	0: OFF/ 1: ON
	int bHomSig1[2];	원점 (X 축: bHomSig1 [0], Y 축: bHomSig1 [1])	0: OFF/ 1: ON
	int bHomSig2[2];	엔코더 Z 상 (X 축: bHomSig2 [0], Y 축: bHomSig2 [1])	0: OFF/ 1: ON
	int LmtP[2];	Limit+ (X 축: LmtP [0], Y 축: LmtP [1])	0: OFF/ 1: ON
	int LmtM[2];	Limit- (X 축: LmtM [0], Y 축: LmtM [1])	0: OFF/ 1: ON
	int EMG[2];	EMG (X 축: EMG [0], Y 축: EMG [1])	0: OFF/ 1: ON
구조체명	**변수 형식**	**내용**	**데이터 값**
PARALLELSTATE	int bInput0Lev[2];	범용 입력 0 (X 축: bInput0Lev [0], Y 축: bInput0Lev [1])	0: OFF/ 1: ON
	int bInput1Lev[2];	범용 입력 1 (X 축: bInput1Lev [0], Y 축: bInput1Lev [1])	0: OFF/ 1: ON

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

4.3 autpmc_SetUserOut

autpmc_SetUserOut 함수는 PMC-2HSP 의 CN4 커넥터의 X 축 범용 출력 0/1 핀 또는 CN5 커넥터의 Y 축 범용 출력 0/1 핀을 ON/OFF 합니다.

(1) 함수명

```
int autpmc_SetUserOut(
int PortNum,
char nNodeId,
char axis,
int bPort,
int bOn
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bPort
  범용 출력 0 핀을 설정하려면 FPMC_OUTPORT0(0)을 범용 출력 1 핀을 설정하려면 FPMC_OUTPORT1(1)을 입력합니다.
- bOn
  해당 범용 출력 포트 핀을 ON 하려면 FPMC_ON(1)을 OFF 하려면 FPMC_OFF(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

4.4 autpmc_GetCurPos

autpmc_GetCurPos 함수는 PMC-2HSP 의 현재 좌표를 읽습니다.

(1) 함수명

```
int autpmc_GetCurPos(
int PortNum,
char nNodeId,
char axis,
long *lCurPos_X,
long *lCurPos_Y
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- lCurPos_X

  현재 X 축 좌표를 읽습니다. (-8,388,608~8,388,607)

- lCurPos_Y

  현재 Y 축 좌표를 읽습니다. (-8,388,608~8,388,607)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

4.5 autpmc_GetCurPgmNo

autpmc_GetCurPgmNo 함수는 PMC-2HSP 의 현재 실행 중인 프로그램 스텝을 읽습니다.

(1) 함수명

```
int autpmc_GetCurPgmNo(
int PortNum,
char nNodeId,
char axis,
int *iCurPgmNo_X,
int *iCurPgmNo_Y
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iCurPgmNo_X
  현재 실행 중인 X 축 프로그램 스텝을 읽습니다. (0~199)
- iCurPgmNo_Y
  현재 실행 중인 Y 축 프로그램 스텝을 읽습니다. (0~199)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

4.6 autpmc_GetErrorSt

autpmc_GetErrorSt 함수는 PMC-2HSP 의 에러 상태를 읽습니다.

(1) 함수명

```
int PMC_ERRORSTATE *autpmc_GetErrorSt(
int PortNum,
char nNodeId,
char axis,
PMC_ERRORSTATE *pError
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- pError

  현재 에러 상태를 읽습니다.

구조체명	변수 형식	내용	데이터 값
PMC_ERRORSTATE	int iErrorState	오류 상태 확인	FPMC_OK(0) : 함수가 정상적으로 명령을 수행하였습니다. FPMC_INVALID_NODE(3) : 잘못된 노드 ID 를 입력하였습니다. FPMC_INVALID_AXIS(4) : 잘못된 축을 입력하였습니다. FPMC_INVALID_DATA)(5) : 잘못된 데이터를 입력하였습니다.
	int bSofLmtErrP[2] 1. bSofLmtErrP[0] -> X 축 2. bSofLmtErrP[1] -> Y 축	소프트웨어 리미트 + 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bSofLmtErrM[2] 1. bSofLmtErrM[0] -> X 축 2. bSofLmtErrM[1] -> Y 축	소프트웨어 리미트 - 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bHardLmtErrP[2] 1. bHardLmtErrP[0] -> X 축 2. bHardLmtErrP[1] -> Y 축	하드웨어 리미트 + 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bHardLmtErrM[2] 1. bHardLmtErrM[0] -> X 축 2. bHardLmtErrM[1] -> Y 축	하드웨어 리미트 - 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bEmgErr[2] 1. bEmgErr[0] -> X 축 2. bEmgErr[1] -> Y 축	긴급정지 시 발생하는 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bPgmErr[2] 1. bPgmErr[0] -> X 축 2. bPgmErr[1] -> Y 축	프로그램 모드 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bHomErr[2] 1. bHomErr[0] -> X 축 2. bHomErr[1] -> Y 축	원점복귀 모드 에러	FPMC_ON(1)/ FPMC_OFF(0)
	int bInxErr[2] 1. bInxErr[0] -> X 축 2. bInxErr[1] -> Y 축	인덱스 에러	FPMC_ON(1)/ FPMC_OFF(0)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

4.7 autpmc_IsRun

autpmc_IsRun 함수는 PMC-2HSP 의 현재 구동 상태를 읽어옵니다.

(1) 함수명

int PMC_RUNSTATE *autpmc_IsRun(
int PortNum,
char nNodeId,
char axis,
PMC_RUNSTATE *pRun
);

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_ AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- pRun

  현재 구동 상태를 읽습니다.

구조체명	변수 형식	내용	데이터 값
PMC_ RUN STATE	int iErrorState	오류 상태 확인	0: 함수가 정상적으로 명령을 수행하였습니다. 3: 잘못된 노드 ID 를 입력하였습니다. 4: 잘못된 축을 입력하였습니다. 5: 잘못된 데이터를 입력하였습니다.
	int bHomIsRun[2] 1. bHomIsRun[0] ->X 축 2. bHomIsRun[1] ->Y 축	X 축 원점복귀 모드 구동	1: ON 0: OFF
	int bJogIsRun[2] 1. bJogIsRun[0] ->X 축 2. bJogIsRun[1] ->Y 축	X 축 조그 모드 구동	1: ON 0: OFF
	int bPgmIsRun[2] 1. bPgmIsRun[0] ->X 축 2. bPgmIsRun[1] ->Y 축	X 축 프로그램 모드 구동	1: ON 0: OFF

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

4.8 autpmc_GetModName

autpmc_GetModName 함수는 PMC-2HSP 의 모델명을 읽어옵니다.

(1) 함수명

```
int PMC_SOFTVERSION *autpmc_GetModName(
int PortNum,
char nNodeId,
PMC_SOFTVERSION *pVersion
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- pVersion

  모델명을 읽습니다.

구조체명	변수 형식	내용	데이터 값
PMC_SOFTVERSION	int iErrorState	오류 상태 확인	0: 함수가 정상적으로 명령을 수행하였습니다. 3: 잘못된 노드 ID 를 입력하였습니다. 4: 잘못된 축을 입력하였습니다. 5: 잘못된 데이터를 입력하였습니다.
	char cModName[12];	모델명 읽어오기	PMC-2HSP-USB PMC-2HSP-485

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

4.9 autpmc_GetSofVer

autpmc_GetSofVer 함수는 PMC-2HSP 의 펌웨어 버전을 읽어옵니다.

(1) 함수명

```
int PMC_SOFTVERSION *autpmc_GetSofVer(
int PortNum,
char nNodeId,
PMC_SOFTVERSION *pVersion
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- pVersion
  현재 펌웨어 버전을 읽습니다. 버전 구성은 연도 4 자리, 월 2 자리, 일 2 자리로 연속으로 구성되어 있습니다.
  예) 20091009 2009 년 10 월 09 일 버전

구조체명	변수 형식	내용	데이터 값
PMC_ PARADATA	int iErrorState	오류 상태 확인	FPMC_OK(0) : 함수가 정상적으로 명령을 수행하였습니다. FPMC_INVALID_NODE(3) : 잘못된 노드 ID 를 입력하였습니다. FPMC_INVALID_AXIS(4) : 잘못된 축을 입력하였습니다. FPMC_INVALID_DATA)(5) : 잘못된 데이터를 입력하였습니다.
	char cSofVer[8];	펌웨어 버전	yyyymmdd

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.

5 동작 제어

5.1 autpmc_HomRun

autpmc_HomRun 함수는 PMC-2HSP 에서 원점 복귀 모드를 실행합니다.

(1) 함수명

int autpmc_HomRun(
int PortNum,
char nNodeId,
char axis
);

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

  또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.2 autpmc_ABSMove

autpmc_ABSMove 함수는 PMC-2HSP 에서 원점을 기준으로 지정된 거리를 절대 위치로 이동합니다. (펄스 스케일값에 비례됩니다.)

(1) 함수명

```
int autpmc_ABSMove(
int PortNum,
char nNodeId,
char axis,
long lPos_X,
long lPos_Y
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- lPos

  이동 위치를 절대값으로 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.3 autpmc_INCMove

autpmc_INCMove 함수는 PMC-2HSP 에서 현재 위치를 기준으로 지정된 거리를 상대 위치로 이동합니다. (펄스 스케일값에 비례됩니다.)

(1) 함수명

```
int autpmc_INCMove(
int PortNum,
char nNodeId,
char axis,
long lPos_X,
long lPos_Y
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- lPos

  이동 위치를 상대값으로 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.4 autpmc_ContMove

autpmc_ContMove 함수는 PMC-2HSP 의 정지 신호가 입력될 때까지 연속하여 드라이브 펄스를 출력합니다.(브로드캐스트 가능)

(1) 함수명

```
int autpmc_ContMove(
int PortNum,
char nNodeId,
char axis,
int bDirection
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 을 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- bDirection

  '+'로 이동하려면 FPMC_PLUS(1)을 '-'로 이동하려면 FPMC_MINUS(0)을 입력합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.5 autpmc_LIDMove

autpmc_LIDMove 함수는 PMC-2HSP 에서 현재 좌표로부터 종점 좌표를 향해 2 축 직선 보간을 실행합니다.

(1) 함수명

```
int autpmc_LIDMove(
int PortNum,
char nNodeId,
int bFLS,
long lXEndPos,
long lYEndPos
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- bFLS
  선속 일정을 사용하려면 FPMC_ON(1)을 사용하지 않으려면 FPMC_OFF(0)을 입력합니다.
- lXEndPos
  X 축 종점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lYEndPos
  Y 축 종점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.6 autpmc_CIDMove

autpmc_CIDMove 함수는 PMC-2HSP 에서 CW 방향(시계 방향)으로 원 보간 드라이브를 실행합니다.

(1) 함수명

```
int autpmc_CIDMove(
int PortNum,
char nNodeId,
int bFLS,
long lRadius,
long lMDP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- bFLS
  선속 일정을 사용하려면 FPMC_ON(1)을 사용하지 않으려면 FPMC_OFF(0)을 입력합니다.
- lRadius
  반지름을 입력합니다. 이때 유효한 범위는 0~8,388,607 입니다.
- lMDP
  매뉴얼 감속점을 입력합니다. 이때 유효한 범위는 0~268,435,455 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.7 autpmc_FIDMove

autpmc_FIDMove 함수는 PMC-2HSP 에서 CW 방향(시계 방향)으로 원호 보간 드라이브를 실행합니다.

(1) 함수명

```
int autpmc_FIDMove(
int PortNum,
char nNodeId,
int bFLS,
long lXCenPos,
long lYCenPos,
long lXEndPos,
long lYEndPos,
long lMDP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- bFLS
  선속 일정을 사용하려면 FPMC_ON(1)을 사용하지 않으려면 FPMC_OFF(0)을 입력합니다.
- lXCenPos
  X 축 중점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lYCenPos
  Y 축 중점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lXEndPos
  X 축 종점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lYEndPos
  X 축 종점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lMDP
  매뉴얼 감속점을 입력합니다. 이때 유효한 범위는 0~268,435,455 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

5.8 autpmc_RIDMove

autpmc_RIDMove 함수는 PMC-2HSP 에서 CCW 방향(반시계 방향)으로 원호 보간 드라이브를 실행합니다.

(1) 함수명

```
int autpmc_RIDMove(
int PortNum,
char nNodeId,
int bFLS,
long lXCenPos,
long lYCenPos,
long lXEndPos,
long lYEndPos,
long lMDP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- bFLS
  선속 일정을 사용하려면 FPMC_ON(1)을 사용하지 않으려면 FPMC_OFF(0)을 입력합니다.
- lXCenPos
  X 축 중점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lYCenPos
  Y 축 중점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lXEndPos
  X 축 종점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lYEndPos
  Y 축 종점 좌표를 입력합니다. 이때 유효한 범위는 -8,388,608~8,388,607 입니다.
- lMDP
  매뉴얼 감속점을 입력합니다. 이때 유효한 범위는 0~268,435,455 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6 프로그램 제어

6.1 autpmc_PgmRun

autpmc_PgmRun 함수는 PMC-2HSP 의 프로그램 모드를 실행합니다.
입력된 해당스텝 번호부터 드라이브를 시작합니다.(브로드캐스트 가능)

(1) 함수명

```
int autpmc_PgmRun(
int PortNum,
char nNodeId,
char axis,
int iStepNo
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 를 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iStepNo
  프로그램 시작 실행 번지를 입력합니다. 이때 유효한 범위는 0~199 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.2 autpmc_PgmStepRun

autpmc_PgmStepRun 함수는 PMC-2HSP 에서 프로그램 모드에서 한 스텝만 실행합니다.

(1) 함수명

```
int autpmc_PgmStepRun(
int PortNum,
char nNodeId,
char axis,
int iStepNo
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 을 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- iStepNo

  한 스텝만 실행할 프로그램 시작 번지를 입력합니다. 이때 유효한 범위는 0~199 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.3 autpmc_PgmPause

autpmc_PgmPause 함수는 PMC-2HSP 의 프로그램 모드를 일시 정지합니다.
(브로드캐스트 가능)

(1) 함수명

```
int autpmc_PgmPause(
int PortNum,
char nNodeId,
char axis
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 을 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

6.4 autpmc_PgmReRun

autpmc_PgmReRun 함수는 PMC-2HSP 의 프로그램 모드를 재실행하는 함수로서 일시 정지에서 정지된 스텝부터 다시 시작합니다.(브로드캐스트 가능)

(1) 함수명

```
int autpmc_PgmReRun(
int PortNum,
char nNodeId,
char axis
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 을 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

6.5 autpmc_PgmStop

autpmc_PgmStop 함수는 PMC-2HSP 의 프로그램 모드를 강제 종료합니다.
(브로드캐스트 가능)

(1) 함수명

```
int autpmc_PgmStop(
int PortNum,
char nNodeId,
char axis
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE 를 리턴합니다. 또한, Broadcast (0x80)을 입력하면 브로드캐스트 기능으로 현재 PC 와 연결된 모든 PMC-2HSP 에 데이터를 전송합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.

6.6 autpmc_DelPgmData

autpmc_DelPgmData 함수는 PMC-2HSP 의 프로그램 모드 내에서 지정한 스텝의 데이터를 삭제합니다.

(1) 함수명

```
int autpmc_DelPgmData(
int PortNum,
char nNodeId,
char axis,
int nStepNo
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.7 autpmc_DelPgmDataAll

autpmc_DelPgmDataAll 함수는 PMC-2HSP 의 프로그램 모드 내에서 모든 스텝의 데이터를 삭제합니다.

(1) 함수명

```
int autpmc_DelPgmDataAll(
int PortNum,
char nNodeId,
char axis
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.8 autpmc_PgmABS

autpmc_PgmABS 함수는 PMC-2HSP 의 프로그램 모드 내에서 원점을 기준으로 지정된 거리를 절대 위치로 이동합니다.

(1) 함수명

```
int autpmc_PgmABS(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
long lPos,
int nSpeed,
int nTimer,
int bEndP,
int bBoth
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.
- lPos
  절대 위치 좌표 (-8,388,608~8,388,607)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)

- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)
- bBoth
  X,Y 축 동시 동작 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.9 autpmc_PgmINC

autpmc_PgmINC 함수는 PMC-2HSP 의 프로그램 모드 내에서 현재 위치를 기준으로 지정된 거리를 상대 위치로 이동합니다.

(1) 함수명

```
int autpmc_PgmINC(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
long lPos,
int nSpeed,
int nTimer,
int bEndP,
int bBoth
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.
- lPos
  절대 위치 좌표(-8,388,608~8,388,607)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)

- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)
- bBoth
  X,Y 축 동시 동작 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.10 autpmc_PgmHOM

autpmc_PgmHOM 함수는 PMC-2HSP 의 프로그램 모드 내에서 원점 복귀 모드에서 설정되어 있는 순서에 따라 원점 복귀를 실행합니다.

(1) 함수명

```
int autpmc_PgmHOM(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int bEndP,
int bBoth
) ;
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.
- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)
- bBoth
  X,Y 축 동시 동작 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.11 autpmc_PgmLID

autpmc_PgmLID 함수는 PMC-2HSP 의 프로그램 모드 내에서 현재 좌표로부터 종점 좌표를 향해 2 축 직선 보간을 실행합니다.

(1) 함수명

```
int autpmc_PgmLID(
int PortNum,
char nNodeId,
int nStepNo,
long lXEndPos,
long lYEndPos,
int bFLS,
int nSpeed,
int nTimer,
int bEndP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 이며, 총 2 스텝이 할당됩니다.
- lXEndPos
  X 축 종점 좌표(-8,388,608~8,388,607)
- lYEndPos
  Y 축 종점 좌표(-8,388,608~8,388,607)
- bFLS
  선속 일정 사용 유무(0: 사용안함, 1: 사용)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)
- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.12 autpmc_PgmCID

autpmc_PgmCID 함수는 PMC-2HSP 의 프로그램 모드 내에서 X, Y 축의 CW 방향(시계 방향)으로 원 보간 드라이브를 실행합니다.

(1) 함수명

```
int autpmc_PgmCID(
int PortNum,
char nNodeId,
int nStepNo,
long lRadius,
int bFLS,
int nSpeed,
int nTimer,
int bEndP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 이며 총 2 스텝이 할당됩니다.
- lRadius
  반지름 좌표 0~8,388,607
- bFLS
  선속 일정 사용 유무(0: 사용안함, 1: 사용)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)
- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.13 autpmc_PgmFID

autpmc_PgmFID 함수는 PMC-2HSP 의 프로그램 모드 내에서 X, Y 축의 CW 방향(시계 방향)으로 원호 보간 드라이브를 실행합니다.

(1) 함수명

```
int autpmc_PgmFID(
int PortNum,
char nNodeId,
int nStepNo,
long XCenPos,
long XEndPos,
long YCenPos,
long YEndPos,
int bFLS,
int nSpeed,
int nTimer,
int bEndP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 이며, 총 2 스텝이 할당됩니다.
- XCenPos
  X 축 중점 좌표 -8,388,608~8,388,607
- XEndPos
  X 축 종점 좌표 -8,388,608~8,388,607
- YCenPos
  Y 축 중점 좌표 -8,388,608~8,388,607
- YEndPos
  Y 축 종점 좌표 -8,388,608~8,388,607
- bFLS
  선속일정 사용 유무(0: 사용안함, 1: 사용)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)
- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.14 autpmc_PgmRID

autpmc_PgmRID 함수는 PMC-2HSP 의 프로그램 모드 내에서 X, Y 축의 CCW 방향(반시계 방향)으로 원호 보간 드라이브를 실행합니다.

(1) 함수명

```
int autpmc_PgmRID(
int PortNum,
char nNodeId,
int nStepNo,
long XCenPos,
long XEndPos,
long YCenPos,
long YEndPos,
int bFLS,
int nSpeed,
int nTimer,
int bEndP
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 이며, 총 2 스텝이 할당됩니다.
- XCenPos
  X 축 중점 좌표 -8,388,608~8,388,607
- XEndPos
  X 축 종점 좌표 -8,388,608~8,388,607
- YCenPos
  Y 축 중점 좌표 -8,388,608~8,388,607
- YEndPos
  Y 축 종점 좌표 -8,388,608~8,388,607
- bFLS
  선속일정 사용 유무(0: 사용안함, 1: 사용)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)
- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.15 autpmc_PgmFRID

autpmc_PgmFRID 함수는 PMC-2HSP 의 프로그램 모드 내에서 X, Y 축의 CW/CCW 방향(시계/반시계 방향)으로 원호 보간 드라이브를 실행합니다.

(4) 함수명

```
int autpmc_PgmFRID(
int PortNum,
char nNodeId,
int nStepNo,
long XCenPos,
long XEndPos,
long YCenPos,
long YEndPos,
int bFLS,
int nSpeed,
int nTimer,
int bEndP
);
```

(5) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 이며, 총 2 스텝이 할당됩니다.
- XCenPos
  X 축 중점 좌표 -8,388,608~8,388,607
- XEndPos
  X 축 종점 좌표 -8,388,608~8,388,607
- YCenPos
  Y 축 중점 좌표 -8,388,608~8,388,607
- YEndPos
  Y 축 종점 좌표 -8,388,608~8,388,607
- bFLS
  선속일정 사용 유무(0: 사용안함, 1: 사용)
- nSpeed
  속도 인덱스(1~4)
- nTimer
  포스트 타이머 인덱스(1~3)
- bEndP
  드라이브 종료 펄스 유무(0: 사용안함, 1: 사용)

(6) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.16 autpmc_PgmICJ

autpmc_PgmICJ 함수는 PMC-2HSP 의 프로그램 모드 내에서 선택한 입력 포트가 활성화 상태라면 지정된 스텝으로 점프합니다. 입력 포트가 비활성화 상태라면 바로 다음 스텝을 실행합니다.

(1) 함수명

```
int autpmc_PgmICJ(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nJumpStep,
int nInputPtNo
);
```

(2) S 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.
- nJumpStep
  점프 할 스텝 번호(0~199)
- nInputPtNo
  입력 포트 번호(0~14)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.17 autpmc_PgmIRD

autpmc_PgmIRD 함수는 PMC-2HSP 의 프로그램 모드 내에서 선택한 입력 포트가 활성화 상태가 되면 다음 스텝으로 이동합니다. 입력 포트가 비활성화 상태라면 활성화 상태가 될 때까지 현재 스텝에서 대기합니다.

(1) 함수명

```
int autpmc_PgmIRD(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nInputPtNo
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.
- nInputPtNo
  입력 포트 번호(0~14)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.18 autpmc_PgmOPC

autpmc_PgmOPC 함수는 PMC-2HSP 의 프로그램 모드 내에서 선택한 출력 포트를 ON, OFF 합니다.

(1) 함수명

```
int autpmc_PgmOPC(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nInputPtNo,
int bON
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

- nInputPtNo

  입력 포트 번호(0~3)

- bON

  OFF(0)~ON(1)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.19 autpmc_PgmOPT

autpmc_PgmOPT 함수는 PMC-2HSP 의 프로그램 모드 내에서 선택한 출력 포트를 ON Time 설정 시간 동안 ON 합니다.

(1) 함수명

```
int autpmc_PgmOPT(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nOutPtNo,
int iOnTim,
int bNextStep
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

- nOutPtNo

  출력 포트 번호(0~3)

- iOnTim

  출력 포트를 ON 시키는 시간: 0~65,535ms

- bNextStep

  ON(1): 출력동작과 관계없이 다음 스텝으로 이동합니다.

  OFF(0)

  : ON 설정 시간 동안 선택한 출력 포트를 ON 시키고, 시간 종료 시 다음 스텝으로 이동합니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.20 autpmc_PgmJMP

autpmc_PgmJMP 함수는 PMC-2HSP 의 프로그램 모드 내에서 지정된 스텝으로 점프합니다.

(1) 함수명

```
int autpmc_PgmJMP(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nJumpStep
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

- nJumpStep

  점프 할 스텝 번호(0~199)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.21 autpmc_PgmREP

autpmc_PgmREP 함수는 PMC-2HSP 의 프로그램 모드 내에서 이 명령의 다음 스텝 부터 autpmc_PgmRPE 함수(반복 종료)를 만날 때까지 지정 횟수만큼 반복 실행합니다.

(1) 함수명

```
int autpmc_PgmREP(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nRepCnt
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

- nRepCnt

  반복 횟수(1~255)

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.22 autpmc_PgmRPE

autpmc_PgmREP 함수는 PMC-2HSP 의 프로그램 모드 내에서 autpmc_PgmREP(반복시작) 함수의 종료 함수입니다.

(1) 함수명

```
int autpmc_PgmRPE(
int PortNum,
char nNodeId,
char axis,
int nStepNo
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.23 autpmc_PgmEND

autpmc_PgmEND 함수는 PMC-2HSP 의 프로그램 모드 내에서 프로그램을 종료합니다. 프로그램의 마지막에 반드시 입력해야 합니다.

(1) 함수명

```
int autpmc_PgmEND(
int PortNum,
char nNodeId,
char axis,
int nStepNo
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.24 autpmc_PgmTIM

autpmc_PgmTIM 함수는 PMC-2HSP 의 프로그램 모드 내에서 지정 시간만큼 대기 명령을 수행합니다.

(1) 함수명

```
int autpmc_PgmTIM(
int PortNum,
char nNodeId,
char axis,
int nStepNo,
int nOnTim
);
```

(2) 파라미터

- PortNum
  명령어를 수행할 Serial Port 를 입력합니다.
- nNodeId
  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.
- axis
  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo
  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.
- nOnTim
  대기시간을 입력합니다. 이때 유효한 범위는 0~65,535ms 입니다.

(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

6.25 autpmc_PgmNOP

autpmc_PgmNOP 함수는 PMC-2HSP 의 프로그램 모드 내에서 아무 것도 처리하지 않습니다.

(1) 함수명

```
int autpmc_PgmNOP(
int PortNum,
char nNodeId,
char axis,
int nStepNo
);
```

(2) 파라미터

- PortNum

  명령어를 수행할 Serial Port 를 입력합니다.

- nNodeId

  설정할 노드 ID 를 선택합니다. 노드 ID 의 범위는 1~16 까지이며, ID 의 범위를 벗어났을 경우에는 FPMC_INVALID_NODE(3)을 리턴합니다.

- axis

  제어할 축(X 축, Y 축)을 선택합니다. 잘못된 축을 입력하였을 경우 FPMC_INVALID_AXIS(4)를 리턴합니다.

구분	정의	내용	상수값
PMC_AXIS	FPMC_X_AXIS	X 축	0
	FPMC_Y_AXIS	Y 축	1
	FPMC_X_Y_AXIS	X, Y 축	2

- nStepNo

  프로그램 스텝 번호를 입력합니다. 이때 유효한 범위는 0~199 입니다.

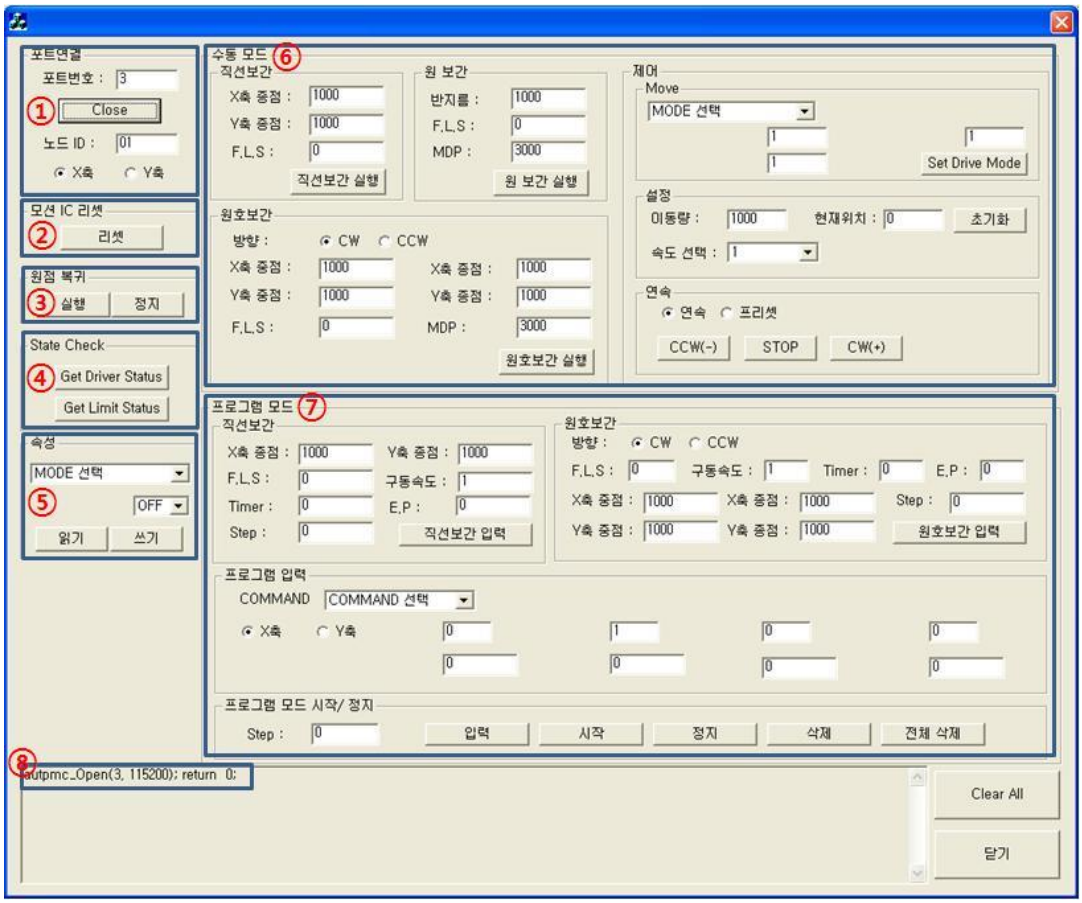
(3) 리턴값

구분	정의	리턴값	내용
정상	FPMC_OK	0	함수가 정상적으로 명령을 수행하였습니다.
입력에러	FPMC_INVALID_NODE	3	잘못된 노드 ID 를 입력하였습니다.
	FPMC_INVALID_AXIS	4	잘못된 축을 입력하였습니다.
	FPMC_INVALID_DATA	5	잘못된 데이터를 입력하였습니다.

7 라이브러리로 활용 가능한 MFC 예제 프로그램

(1) MFC 예제 프로그램

라이브러리 함수를 활용한 MFC 예제 프로그램입니다. 소스를 직접 확인 할 수 있으므로, 필요한 동작을 쉽게 이해하고 사용 할 수 있습니다.



(2) 설명

번호	구분	내용
①	포트연결	포트 연결 라이브러리 함수를 호출하여 통신 연결 및 해제를 수행합니다.
②	모션 IC 리셋	현재 연결되어 있는 PMC 모션 IC 를 리셋합니다.
③	원점 복귀	원점 복귀 라이브러리 함수를 호출하여 원점 복귀 실행 또는 정지합니다.
④	드라이버 상태 체크	현재 연결되어 있는 PMC 모션 드라이버의 상태나 오류를 확인합니다. (각 축의 원점 복귀모드, 조그 모드, 프로그램 모드가 구동하고 있는지 또는 오류 확인이 가능합니다.)
⑤	파라미터 속성 읽기/ 쓰기	현재 설정되어 있는 PMC 모션 드라이버의 파라미터값을 읽거나 쓸 수 있습니다.
⑥	수동 모드	통신을 통하여 패킷을 보내어 하나의 동작을 수행하는 모드입니다.
⑦	프로그램 모드	EEPROM 에 미리 프로그램 스텝을 저장해 놓고, 그 후 통신이 불가능한 환경에서 사용하거나 연속적인 동작을 수행하기 위한 모드입니다.
⑧	로그	실행한 기능에 대한 라이브러리 함수명과 세부 속성값을 확인 할 수 있습니다.

8 라이브러리 사용 예제

8.1 초기화

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{

    int stat=0; // 현재 접속 가능한 Comport 의 상태 확인 변수
    for(int i=0;i<COMPORT;i++)
    {
        switch(i)
        {
        case 0: stat = autpmc_Open( 0, FPMC_BAUD_115200 );
            break;
        case 1: stat = autpmc_Open( 1, FPMC_BAUD_115200 );
            break;
        case 2: stat = autpmc_Open( 2, FPMC_BAUD_115200 );
            break;
        case 3: stat = autpmc_Open( 3, FPMC_BAUD_115200 );
            break;
        case 4: stat = autpmc_Open( 4, FPMC_BAUD_115200 );
            break;
        case 5: stat = autpmc_Open( 5, FPMC_BAUD_115200 );
            break;
        case 6: stat = autpmc_Open( 6, FPMC_BAUD_115200 );
            break;
        case 7: stat = autpmc_Open( 7, FPMC_BAUD_115200 );
            break;
        case 8: stat = autpmc_Open( 8, FPMC_BAUD_115200 );
            break;
        case 9: stat = autpmc_Open( 9, FPMC_BAUD_115200 );
            break;
```

```
                case 10: stat = autpmc_Open(10, FPMC_BAUD_115200);
                        break;
                case 11: stat = autpmc_Open(11, FPMC_BAUD_115200);
                        break;
                case 12: stat = autpmc_Open(12, FPMC_BAUD_115200);
                        break;
                case 13: stat = autpmc_Open(13, FPMC_BAUD_115200);
                        break;
                case 14: stat = autpmc_Open(14, FPMC_BAUD_115200);
                        break;
                case 15: stat = autpmc_Open(15, FPMC_BAUD_115200);
                        break;
                }

                if (stat == FPMC_OK)
                {
                        printf("MESSAGE: Found and open 'PMC-2HSP (ID=%d)' ComPort\n",
i);
                }
        }

}
```

8.2 정지, 종료

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{

	int Flag=0; //명령어 상태 확인 변수


	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수


	Flag = autpmc_SlowStop(PORTNUM, Node01, FPMC_X_Y_AXIS); //감속 정지 명령 함수


	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}


	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.3 파라미터 설정

8.3.1 autpmc_GetParaAll( )

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수


	PMC_PARADATA Data;
	PMC_PARADATA *pData = &Data;


	autpmc_GetParaAll(PORTNUM, Node01, FPMC_X_Y_AXIS, pData);
	// PMC-2HSP 의 저장된 모든 파라미터 설정값을 구조체 변수에 저장.


	printf("X 축 Limit Stop Mode: %d\nY 축 Limit Stop Mode: %d\n", pData-
>bLmtStopMod[0], pData->bLmtStopMod[1]); //저장된 구조체 변수 값 확인.


	if(pData->iErrorState!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}


	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.3.2 autpmc_GetLmtActLev( )

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	int Flag=0;
	int bLevel=0;


	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수




	Flag = autpmc_GetLmtActLev (PORTNUM, Node01, FPMC_X_AXIS, &bLevel); //PMC-2HSP 에 설정된 리미트 신호 논리 레벨을 읽어와서 변수에 저장


	printf("X 축 bLevel: %d\n", bLevel); //변수에 저장된 값 확인


	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}


	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.3.3 autpmc_SetStrSpd( )

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	int Flag=0;

	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수



Flag = autpmc_SetStrSpd(PORTNUM, Node01, FPMC_X_Y_AXIS, 1000);
//초기 속도 1000pps 설정

	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}

	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.4 I/O 제어

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수


	PARALLELSTATE State;
	PARALLELSTATE *pState = &State;


	autpmc_GetParallelIO(PORTNUM, Node01, pState); // PMC-2HSP 의 저장된 Parallel I/F 커넥터(CN3) 입력 신호 설정값을 구조체 변수에 저장.


	printf("원점 복귀: %d\n", pState->HOME); //저장된 구조체 변수 값 확인.


	if(pState->bErrorState!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}


	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.5 동작 제어

8.5.1 autpmc_HomRun( )

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	int Flag=0; ; //명령어 상태 확인 변수

	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수

	Flag = autpmc_HomRun(PORTNUM, Node01, FPMC_X_Y_AXIS);
	//원점 복귀 실행 명령 함수

	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}

	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.5.2 autpmc_INCMove( )

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	int Flag=0; ; //명령어 상태 확인 변수


	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수


	autpmc_SetDrvSpd(PORTNUM, Node01, FPMC_X_AXIS, 1);
	//파라미터에 설정된 드라이브 속도(Drive Speed 1 ~ 4) 선택


	Flag = autpmc_INCMove(PORTNUM, Node01, FPMC_X_AXIS, 10000);
	//현재 위치를 기준으로 지정된 거리를 이동하는 상대 위치 이동 명령(10000pulse) 실행


	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}


	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.5.3 autpmc_FIDMove( )

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
    int Flag=0; ; //명령어 상태 확인 변수
    long ManualDecelPoint; //ManualDecelPoint
    int Range; // Speed Multiplier
    long Acceleration; // Acceleration rate
    long Deceleration; // Deceleration rate
    long StVelocity; // Start speed
    long DrVelocity; //Drive speed
    long CtXaxis; //X axis Center position
    long CtYaxis; //Y axis Center position
    long EndXaxis; //X axis End position
    long EndYaxis; //Y axis End position
    int CW=1; //Direction
    int bDecValue; //Fiexd Line Speed;

    Range = 10;
    Acceleration = 400;
    Deceleration = 400;
    StVelocity = 50;
    DrVelocity = 10;
    CtXaxis = 1000;
    CtYaxis = 1000;
    EndXaxis = 1000;
    EndYaxis = 1000;
    bDecValue = 1; //Enable;

    autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수

    autpmc_SetDrvSpd(PORTNUM, Node01, FPMC_X_AXIS, 1);
```

```
	//파라미터에 설정된 드라이브 속도(Drive Speed 1 ~ 4) 선택


ManualDecelPoint = CalculateManualDecelPoint(Range, Acceleration, Deceleration,
StVelocity, DrVelocity, CtXaxis, CtYaxis, EndXaxis, EndYaxis, CW, bDecValue); //매뉴얼 감속점
계산


	printf("매뉴얼 감속점: %ld\n", ManualDecelPoint); //매뉴얼 감속점 확인


	Flag = autpmc_FIDMove(PORTNUM, Node01, CW, CtXaxis, CtYaxis, EndXaxis, EndYaxis,
ManualDecelPoint); //CW 방향(시계 방향)으로 구동하는 원호 보간 드라이브 명령 실행


	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}


	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

8.6 프로그램 제어

```
#include <windows.h>
#include <stdio.h>
#include “lib/include/Library.h”
#pragma comment(lib, "lib\\x86\\PMCLibrary.lib")
#define PortNum 15


void main()
{
	int Flag=0; ; //명령어 상태 확인 변수

	autpmc_Open(PORTNUM, FPMC_BAUD_115200); //통신 연결 함수

	autpmc_PgmABS (PORTNUM, Node01, FPMC_X_Y_AXIS, 0, 10000, 1, 0, 0, 0);
	//프로그램 모드 0 스텝에 절대 위치 운전 명령어 입력

	Flag = autpmc_PgmRun(PORTNUM, Node01, FPMC_X_Y_AXIS, 0);
	//프로그램 모드 0 스텝 실행

	if(Flag!=FPMC_OK)
	{
		printf("error!\n");
		return;
	}

	autpmc_Close(PORTNUM); //통신 연결 해제 함수
}
```

Make Life Easy : Autonics

* 본 매뉴얼에 기재된 사양, 외형치수 등은 제품의 개선을 위해서 예고 없이 변경되거나 일부 모델이 단종될 수 있습니다.

MMD-PMC2HSP2HSN1-V1.2-1911KR