



# HX API TESTER SAMPLE MANUAL MFC

CSCAM

# 목차

---

<b>1. HX API TESTER SAMPLE 구성</b>	3
1.1 HX 제어기 설정	3
1.2 HX API	3
1.3 HX API TESTER MFCv2	3
1.4 Lib	3
1.5 윈도우 설정	3
<b>2. HX 제어기 설정</b>	5
<b>3. HX API TESTER 설정</b>	6
<b>4. HX API TESTER 프로그램 구성 및 사용 방법</b>	7
4.1 HX API TESTER SAMPLE 프로그램 구성	7
4.2 HX API 종류 및 설정	8
4.3 HX 제어기와 연결하기	9
<b>5. 메시지 알람 API 사용하기</b>	13
<b>6. MOTION 화면 구성 및 사용 방법</b>	14
6.1 좌표 모니터링	14
6.2 MODE 설정	18
6.2.1 MPG 모드	18
6.2.2 JOG 모드	19
6.2.3 ZRN 모드	21
6.2.4 AUTO 모드	22
6.2.5 MDI 모드	23
6.2.6 EDIT 모드	24
6.3 NC 제어 설정	25
6.3.1 NC READY	25
6.3.2 RESET	27
6.3.3 CYCLE START	27
6.3.4 CYCLE STOP	28

6.4 JOG 좌표 설정.....	29
<b>7. INFO 화면 구성 및 사용 방법 .....</b>	<b>31</b>
7.1 파라미터 설정.....	31
7.2 RESET 설정 .....	32
<b>8. IO 화면 구성 및 사용 방법 .....</b>	<b>34</b>
<b>9. NC LIST 화면 구성 및 사용 방법 .....</b>	<b>37</b>
9.1 NC LIST 불러오기 .....	37
9.2 SELECT .....	38
9.3 UPLOAD.....	41
9.4 DOWNLOAD.....	43
<b>10. MAP TEST 화면 구성 및 사용 방법.....</b>	<b>46</b>
10.1 GET 기능 .....	46
10.2 SET 기능 .....	47

---

# 1. HX API TESTER SAMPLE 구성

---

HX API TESTER SAMPLE 폴더 구성 목록입니다.

- HX 제어기 세팅
- HXApi
- HXApiTesterMFCv2
- Lib
- 윈도우 세팅

## 1.1 HX 제어기 세팅

---

HX 제어를 세팅할 때 필요한 파일이 들어있습니다.

## 1.2 HX Api

---

HX API TESTER SAMPLE 프로젝트를 실행하는데 필요한 파일이 들어있습니다. HXApiTesterMFCv2 프로젝트를 실행 시 참조하는 헤더파일이 들어있습니다.

## 1.3 HXApiTesterMFCv2

---

HXApiTesterMFCv2 프로젝트 소스가 들어있습니다.

## 1.4 Lib

---

HXApiTesterMFCv2 프로젝트에서 참조하는 HXApi.lib 파일이 들어있습니다.

## 1.5 윈도우 세팅

---

윈도우에서 HX API TESTER SAMPLE 프로그램을 실행하기 위해 필요한 윈도우 구성파일이 들어있습니다. HX API TESTER SAMPLE 프로그램을 실행 시

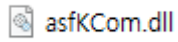


해당 메시지가 뜨는 경우 폴더 안에 있는 DLL 파일을 비트 사양에 맞게 아래 경로로 복사하길 바랍니다.

64 비트 : C:\Windows\SysWOW64

32 비트 : C:\Windows\System32

## 2. HX 제어기 설정



HX 제어기 세팅 폴더에 asfKCom.dll 파일이 들어있습니다.

1. asfKCom.dll 파일을 HX 제어기 실행파일이 있는 경로에 복사합니다.
2. Data 폴더 → KOR 폴더 → HMIData\_L 폴더에 있는 asf.txt 파일을 엽니다.
3. asfKCom 앞에 #이 붙은 경우 삭제바랍니다. (#이 삭제되어야 활성화합니다.)

#	43	0	2	asfBusEthernetCH	0	198	415	165	68	1024	768	[숨겨져서실행...]
	44	0	2	asfKCom	0	198	370	141	97	1024	768	[숨겨져서실행...]
#	44	0	2	asfBusChannel	0	198	370	141	97	1024	768	[숨겨져서실행...]

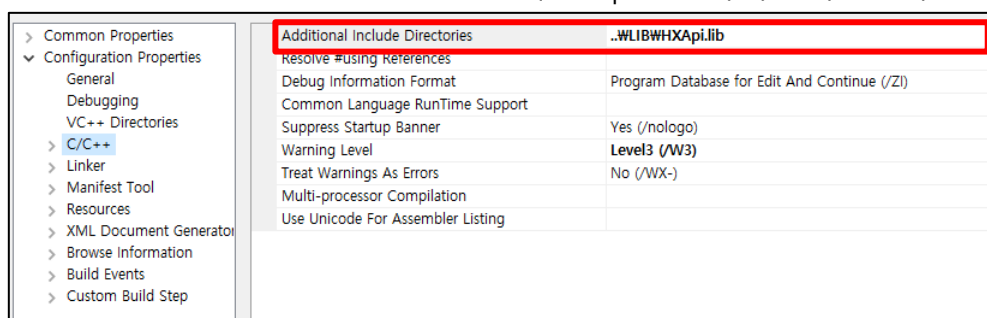
### 3. HX API TESTER SAMPLE 설정

1. Microsoft Visual Studio 2010 을 실행합니다.
2. HXApiTesterMFCv2 폴더에 있는 HXApiTesterMFCv2.sln 프로젝트를 열어 실행합니다.
3. 컴파일 및 빌드를 합니다.

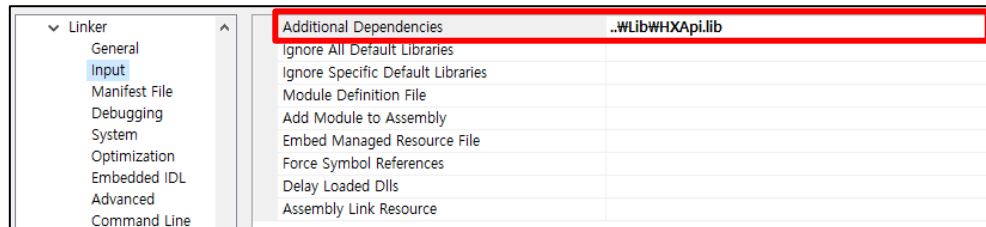
(HXApiTesterMFCv2WDebug, Release 에서 컴파일 및 빌드가 됩니다.)

#### \*\* 프로젝트 속성 설정

속성 - C/C++ - General – Additional include Directories 에 HXApi.lib 를 추가한 사항을 확인할 수 있습니다.



속성 – Linker – Input – Additional Dependencies 에 HXApi.lib 를 추가한 사항을 확인할 수 있습니다.



#### \*\* 64 비트일 경우

HXApiTesterMFCv2W64x 폴더에 Debug, Release 폴더에서 컴파일 및 빌드가 됩니다.

# 4.HX API TESTER SAMPLE 프로그램

## 구성 및 사용 방법

### 4.1 HX API TESTER SAMPLE 프로그램 구성

HXApiTesterMFC

CSCAM

Setup-Motion ( Motion 4 Axis)

2019-12-04 13:33:12

195000

현재 위치

X 0.000

Y 0.000

Z 0.000

A -1.867

B 335.369

기계 위치

X 0.000

Y 0.000

Z 0.000

A -1.867

B 335.369

상대 위치

X 0.000

Y 0.000

Z 0.000

A -1.867

B 335.369

남은 거리

X 0.000

Y 0.000

Z 0.000

A 0.000

B 0.000

MODE

MPG JOG ZRN

AUTO MDI EDIT

NO READY RESET CYCLE START CYCLE STOP

A+ Z+ Y-

X- X+

Y+ Z- A-

Select Axis : AXIS 06

Limit Value(한계 설정)

Absolute Position ☒ USE

Min : 0.000

Max : 0.000

Set Value

JOG Move

Velocity 8.000

Target Pos. Move

Velocity 0.000

Step Move

Velocity 6.000

Homing Move

Accel.(ms) 120.000

Velocity 60.000

1st Decel Velocity : 200.000

2nd Decel Velocity : 10.000

☒ 임의 환점

☐ DOG 사용

금지영역 : 0.000 mm,de

RESET

연결 중

Connection

Motion

IO

NC LIST

Map Test

TOP

- 알람 메시지 모니터링

MENU

- HX 제어기와 연결

- 화면 전환

MAIN SCREEN

- MOTION, INFORMATION, IO INPUT, IO OUTPUT, NC LIST, MAP TEST 화면 구성



### ▶ HX API 종류

1. Hx20Api
2. HXApi

HX API 설정은 HXApiTesterMFCv2 프로젝트 → HXApiTesterMFCDlg.cpp → OnInitDialog() 함수에서 수정할 수 있습니다.

### # 프로그램 코드

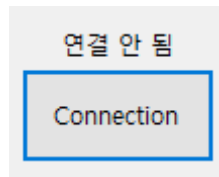
```
// 연결 작업  
m_bCreate = FALSE;  
m_IsConn = false;  
mConnectType = CONN_HX_20;
```

### # 예제 코드

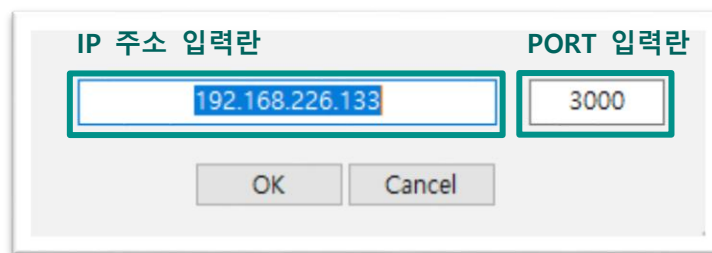
Hx20Api 사용할 경우	mConnectType = CONN_HX_20;
HXApi 사용할 경우	mConnectType = CONN_HX;

### 4.3 HX 제어기와 연결하기

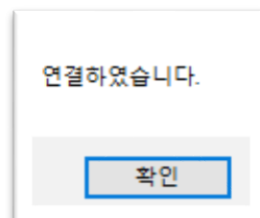
1. MENU 에 있는 Connection 버튼을 클릭하여 Connection 다이얼로그를 엽니다.



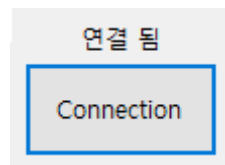
2. IP 주소와 PORT 를 입력합니다.



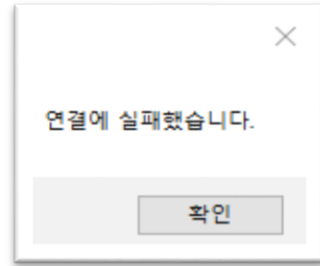
3. 입력 후 OK 버튼을 눌러 Connection 다이얼로그를 종료합니다. 종료와 동시에 연결 시도를 합니다.
4. 연결 성공 시 연결 메시지 박스창이 뜨며, Connection 버튼 클릭 시, 다이얼로그가 재차 열리지 않습니다.



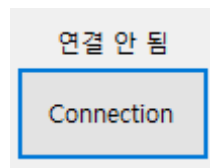
더불어 Connection 버튼 위 레이블에서 연결 상태를 확인할 수 있습니다.



5. 연결 실패 시 연결 실패 메시지 박스창이 뜨며, Connection 버튼을 다시 클릭하여 IP 주소와 PORT 를 기입합니다.



연결이 안 되어있을 시, Connetion 버튼 위 레이블 텍스트는 '연결 안 됨'으로 보여집니다.



# 프로그램 코드

```
if(mConnectType == CONN_HX_20)
{
    // HX_ETHERNET =0   /HX_RTX =1
    // 이니셜라이즈 및 연결하는 API
    // HxInitialize2(IP주소1, IP주소2, IP주소3, IP주소4, PORT번호);
    int res = HxInitialize2(0, ip_arr[0], ip_arr[1], ip_arr[2], ip_arr[3], 3000);
    if (res == 0)
    {
        MessageBox(_T("Connect Fail!"), _T("Message"), MB_OK);
        delete pDlg;
        return ;
    }
}
```

```
else if(mConnectType == CONN_HX)
{
    // 이니셜라이즈 및 연결하는 API
    // HXInitialize(IP주소1, IP주소2, IP주소3, IP주소4, PORT번호);
    long res = HXInitialize(0, ip_arr[0], ip_arr[1], ip_arr[2], ip_arr[3], 3000);

    if(res != 1 )
    {
        MessageBox(_T("Connect Fail!"), _T("Message"), MB_OK);
        delete pDlg;
        return;
    }
}
```

```
// 연결 확인
bool check = HxGetConnectState(0);
if(check)
{
    mIsConn=true;
    setTextConnStatus(true);

}
else
{
    mIsConn=false;
    setTextConnStatus(false);

}

}
```

```
// 연결 확인
long check = HXConnectState();
if(check ==1)
{
    mIsConn=true;
    setTextConnStatus(true);

}
else
{
    mIsConn=false;
    setTextConnStatus(false);

}

}
```

# 예제 코드

Hx20Api	<p><b>Bool HxInitialize2(Int32 comtype, Int32 ip1, Int32 ip2, Int32 ip3, Int32 ip4, Int32 ipport);</b></p> <ul style="list-style-type: none"> <li>- 연결 초기화 및 제거기와 연결하는 API</li> <li>- 초기화 동시에 연결까지 동작함</li> <li>- 연결 성공 시 TRUE 을 반환함</li> <li>- Com type 은 보통 0 임</li> </ul> <p><b>Bool HxGetConnectState(Int32 port);</b></p> <ul style="list-style-type: none"> <li>- HX 제어기 연결 상태를 확인하는 API</li> <li>- Port 는 보통 0 임</li> <li>- 상태값을 bool 형으로 반환함 (true : 연결 중, false : 연결 끊김)</li> <li>- 현재 한번 연결하면 HX 제어기와 통신이 끊어져도 true 값이 반환됨. 2019-12-16</li> </ul> <pre>int res = HxInitialize2(0, ip_arr[0], ip_arr[1], ip_arr[2],ip_arr[3], 3000); if (res == 0) {     MessageBox(_T("Connect Fail!"),_T("Message"), MB_OK);     delete pDlg;     return ; } bool check = HxGetConnectState(0);</pre>
HXApi	<p><b>long HXInitialize(long comtype, long ip1, long ip2, long ip3, long ip4, long port);</b></p> <ul style="list-style-type: none"> <li>- 연결 초기화 및 제거기와 연결하는 API</li> </ul>

	<ul style="list-style-type: none"> <li>- 초기화 동시에 연결까지 동작함</li> <li>- 연결 성공 시 1 을 반환함</li> <li>- comtype 은 보통 0 임</li> </ul> <p><b>long HXConnectState();</b></p> <ul style="list-style-type: none"> <li>- HX 제어기 연결 상태를 확인하는 API</li> <li>- 상태 값을 long 형으로 반환함(1 : 연결 중, 그 외 : 연결 끊김)</li> <li>- 현재 한번 연결하면 HX 제어기와 통신이 끊어져도 1 값이 반환됨. 2019-12-16</li> </ul>
	<pre> long res = HXInitialize(0, ip_arr[0], ip_arr[1], ip_arr[2],ip_arr[3],3000); if(res != 1 ) {     MessageBox(_T("Connect Fail!"),_T("Message"), MB_OK);     delete pDlg;     return; } long check = HXConnectState(); </pre>

## 5. 메시지 알람 API 사용 방법

	Setup-Motion (Motion 5 Axis)
2019-12-2 17:42:53	195000

HX API 가 연결되면 해당 부분에 메시지 코드 또는 메시지 텍스트가 뜹니다. HX API 에 따라 코드 또는 텍스트가 보여집니다.

# 프로그램 코드

```
if(mConnectType == CONN_HX_20)
{
    // 알람 코드 받아오는 변수
    int alarmCode[255] = {0,};

    // 알람 코드 가져오는 API
    // HxGetAlarmCode(PORT, INT ARRAY);
    // (보통 PORT는 0임)
    // (INT ARRAY : ALARM CODE를 담을 변수)
    HxGetAlarmCode(0, alarmCode);
    char buffer[20];
    _itoa_s(alarmCode[0],buffer, sizeof(buffer),10);
    LPCSTR p = buffer;

    SetDlgItemText(IDC_STATIC_MSG, buffer);
}
```

```
else if(mConnectType == CONN_HX)
{
    // HX_STR_ALARMMSG --> 알람 메시지
    // HX_STR_OPMSG --> OP 메시지

    // 알람 코드 받아오는 변수
    char alarmCode[255] = {0,};

    // 알람 코드 가져오는 API
    // XHXGetString(NC ID, ALARM MSG INDEX, CHAR ARRAY);
    // (보통 NC ID는 0임)
    // (ALARM MSG INDEX : 미리 정의 해놓은 ALARM MSG 값을 가져오는 MAP ADDRESS)
    // (CHAR ARRAY : ALARM TEXT를 담을 변수)
    XHXGetString(0, HX_STR_OPMSG, alarmCode);

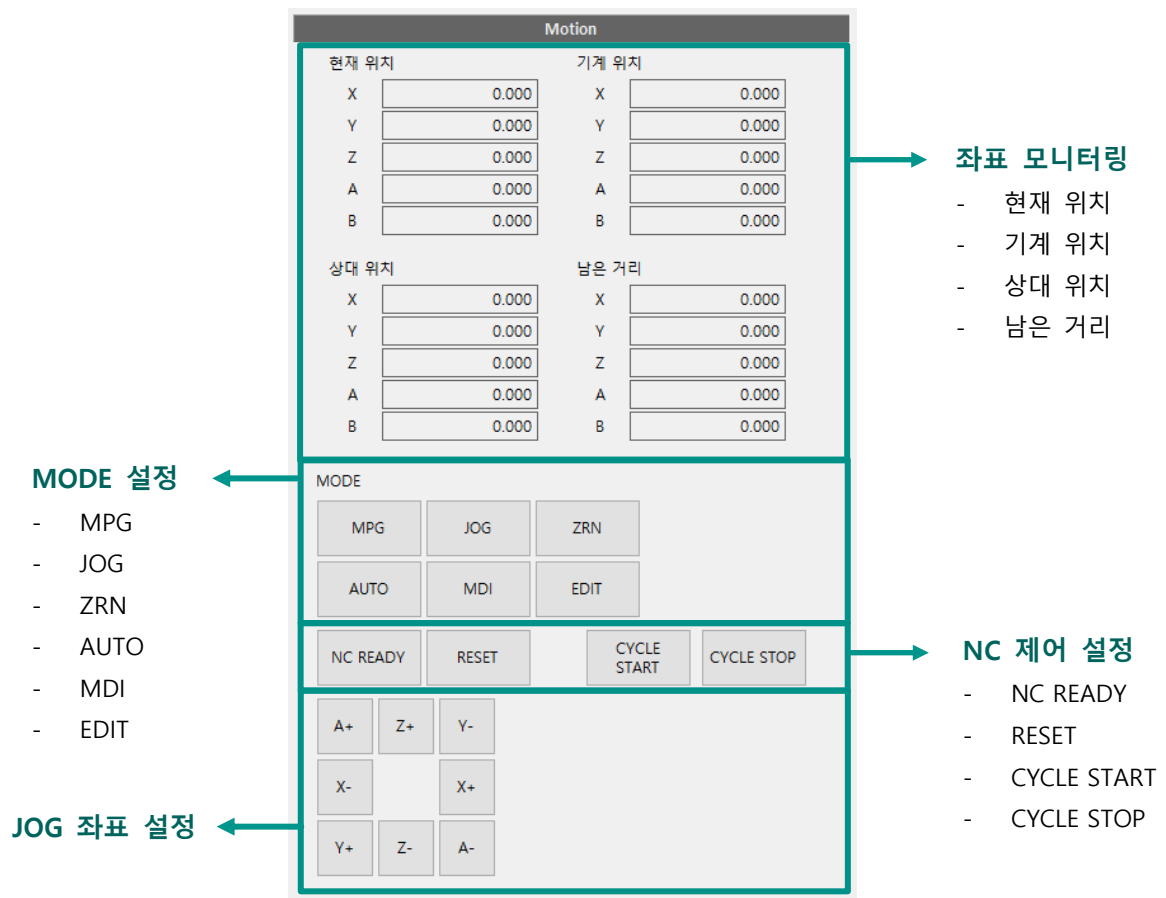
    SetDlgItemText(IDC_STATIC_MSG, alarmCode);
}
```

# 예제 코드

Hx20Api	<b>Int32 HxGetAlarmCode(Int32 port, Int32 code[]);</b> <ul style="list-style-type: none"> <li>- 알람 코드를 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- int 배열 code 에 알람 코드를 담음</li> </ul>
	int alarmCode[255] = {0,}; HxGetAlarmCode(0, alarmCode);
	<b>void XHXGetString(int ncid, int nName, char caStrVal[]);</b>

HXApi	<ul style="list-style-type: none"> <li>- 알람 텍스트를 가져오는 API</li> <li>- Ncid 는 보통 0 임</li> <li>- nName 은 HXMap.HX_STR_OPMSG 지정</li> <li>- char 배열 caStrVal 에 알람 텍스트를 담음</li> </ul>
	<pre>char alarmCode[255] = {0,}; XHXGetString(0, HX_STR_OPMSG, alarmCode);</pre>

## 6. MOTION 화면 구성 및 사용 방법



### 6.1 좌표 모니터링

100ms 주기로 업데이트 되는 좌표를 확인할 수 있습니다.



## #프로그램 코드

```
// 현재 위치
// sv 83
curpos1 = HxGetSVF(0, 83);
curpos2 = HxGetSVF(0, 84);
curpos3 = HxGetSVF(0, 85);
curpos4 = HxGetSVF(0, 86);
curpos5 = HxGetSVF(0, 87);

// 기계 위치
//sn 237
macpos1 = HxGetSNF(0, 237);
macpos2 = HxGetSNF(0, 238);
macpos3 = HxGetSNF(0, 239);
macpos4 = HxGetSNF(0, 240);
macpos5 = HxGetSNF(0, 241);

// 상대 위치
// sn 269?
relpos1 = HxGetSNF(0, 269);
relpos2 = HxGetSNF(0, 270);
relpos3 = HxGetSNF(0, 271);
relpos4 = HxGetSNF(0, 272);
relpos5 = HxGetSNF(0, 273);
|
// 남은거리
// sv 247
dispos1 = HxGetSVF(0, 247);
dispos2 = HxGetSVF(0, 248);
dispos3 = HxGetSVF(0, 249);
dispos4 = HxGetSVF(0, 250);
dispos5 = HxGetSVF(0, 251);
```

```
// sv 83
curpos1 = HXGetRegister64(HX_SV, 83);
curpos2 = HXGetRegister64(HX_SV, 84);
curpos3 = HXGetRegister64(HX_SV, 85);
curpos4 = HXGetRegister64(HX_SV, 86);
curpos5 = HXGetRegister64(HX_SV, 87);

// 기계 위치
//sn 237
macpos1 = HXGetRegister64(HX_SN, 237);
macpos2 = HXGetRegister64(HX_SN, 238);
macpos3 = HXGetRegister64(HX_SN, 239);
macpos4 = HXGetRegister64(HX_SN, 240);
macpos5 = HXGetRegister64(HX_SN, 241);

// 상대 위치
// sn 269?
relpos1 = HXGetRegister64(HX_SN, 269);
relpos2 = HXGetRegister64(HX_SN, 270);
relpos3 = HXGetRegister64(HX_SN, 271);
relpos4 = HXGetRegister64(HX_SN, 272);
relpos5 = HXGetRegister64(HX_SN, 273);|

// 남은거리
// sv 247
dispos1 = HXGetRegister64(HX_SV, 247);
dispos2 = HXGetRegister64(HX_SV, 248);
dispos3 = HXGetRegister64(HX_SV, 249);
dispos4 = HXGetRegister64(HX_SV, 250);
dispos5 = HXGetRegister64(HX_SV, 251);
```

## # 예제 코드

Hx20Api	<b>Double HxGetSVF(Int32 port, Int32 addr);</b> <ul style="list-style-type: none"> <li>- SV 맵에 있는 값을 가져오는 API</li> <li>- Port 는 보통 0 임</li> </ul>
---------	---

	<ul style="list-style-type: none"> <li>- Addr 에 원하는 주소를 기입</li> <li>- double 형으로 값을 반환함</li> </ul> <p><b>Double HxGetSNF(Int32 port, Int32 addr);</b></p> <ul style="list-style-type: none"> <li>- SN 맵에 있는 값을 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소를 기입</li> <li>- double 형으로 값을 반환함</li> </ul> <pre>// 현재 위치 // sv 83 curpos1 = HxGetSVF(0, 83); curpos2 = HxGetSVF(0, 84); curpos3 = HxGetSVF(0, 85); curpos4 = HxGetSVF(0, 86); curpos5 = HxGetSVF(0, 87);  // 기계 위치 //sn 237 macpos1 = HxGetSNF(0, 237); macpos2 = HxGetSNF(0, 238); macpos3 = HxGetSNF(0, 239); macpos4 = HxGetSNF(0, 240); macpos5 = HxGetSNF(0, 241);  // 상대위치 // sn 269? relpos1 = HxGetSNF(0, 269); relpos2 = HxGetSNF(0, 270); relpos3 = HxGetSNF(0, 271); relpos4 = HxGetSNF(0, 272); relpos5 = HxGetSNF(0, 273);  // 남은거리 // sv 247 dispos1 = HxGetSVF(0, 247); dispos2 = HxGetSVF(0, 248); dispos3 = HxGetSVF(0, 249); dispos4 = HxGetSVF(0, 250); dispos5 = HxGetSVF(0, 251);</pre>
HXApi	<p><b>double HXGetRegister64(long name, long addr);</b></p> <ul style="list-style-type: none"> <li>- HXMap 맵, 지정된 주소에서 값을 64 비트로 가져오는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소를 기입</li> <li>- double 형으로 값을 반환함</li> </ul> <pre>// sv 83 curpos1 = HXGetRegister64(HX_SV, 83); curpos2 = HXGetRegister64(HX_SV, 84); curpos3 = HXGetRegister64(HX_SV, 85); curpos4 = HXGetRegister64(HX_SV, 86); curpos5 = HXGetRegister64(HX_SV, 87);</pre>

	<pre>// 기계 위치 //sn 237 macpos1 = HXGetRegister64(HX_SN, 237); macpos2 = HXGetRegister64(HX_SN, 238); macpos3 = HXGetRegister64(HX_SN, 239); macpos4 = HXGetRegister64(HX_SN, 240); macpos5 = HXGetRegister64(HX_SN, 241);  // 상대위치 // sn 269? relpos1 = HXGetRegister64(HX_SN, 269); relpos2 = HXGetRegister64(HX_SN, 270); relpos3 = HXGetRegister64(HX_SN, 271); relpos4 = HXGetRegister64(HX_SN, 272); relpos5 = HXGetRegister64(HX_SN, 273);  // 남은거리 // sv 247 dispos1 = HXGetRegister64(HX_SV, 247); dispos2 = HXGetRegister64(HX_SV, 248); dispos3 = HXGetRegister64(HX_SV, 249); dispos4 = HXGetRegister64(HX_SV, 250); dispos5 = HXGetRegister64(HX_SV, 251);</pre>
--	--

## 6.2 MODE 설정

### 6.2.1 MPG 모드

1. MODE 에 있는 MPG 버튼을 클릭합니다.



(현재 HX API TESTER SAMPLE 에서는 해당 기능을 제공하지 않습니다.)

# 프로그램 코드

- MPG 버튼 클릭할 때

```
if(mConnType == CONN_HX_20)
{
    HxSetGB(0, 1, 5, TRUE);
    Sleep(200);
    HxSetGB(0, 1, 5, FALSE);
}
```

```

else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 1, 5, TRUE);
    Sleep(200);
    HXSetBit(HX_G, 1, 5, FALSE);
}

```

#### # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetGB(0, 1, 5, TRUE); Sleep(200); HxSetGB(0, 1, 5, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_G, 1, 5, TRUE); Sleep(200); HXSetBit(HX_G, 1, 5, FALSE);

### 6.2.2 JOG 모드

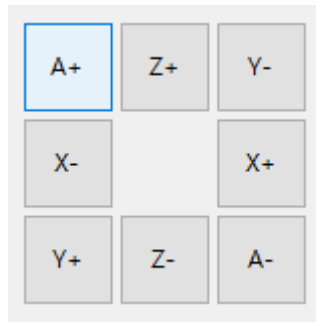
1. MODE 에 있는 JOG 버튼을 클릭합니다.



2. NC READY 버튼을 3 초 이상 눌러줍니다.



3. JOG 버튼에 있는 축 버튼들을 클릭하여 조작합니다.



(현재 4 축(A 축)은 제어기와 연결되어 있지 않습니다.)

#### # 프로그램 코드

- JOG 버튼 클릭할 때

```
if(mConnType == CONN_HX_20)
{
    HxSetGB(0, 1,4, TRUE);
    Sleep(200);
    HxSetGB(0, 1,4, FALSE);
}
```

```
else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 1,4, TRUE);
    Sleep(200);
    HXSetBit(HX_G, 1,4, FALSE);
}
```

#### # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetGB(0, 1,4, TRUE); Sleep(200); HxSetGB(0, 1,4, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_G, 1,4, TRUE); Sleep(200); HXSetBit(HX_G, 1,4, FALSE);

### 6.2.3 ZRN 모드

1. MODE 에 있는 ZRN 버튼을 클릭합니다.



2. NC READY 버튼을 3 초 이상 눌러줍니다.



3. CYCLE START 버튼을 눌러줍니다.



4. 원점 이동하는 것을 좌표 모니터링을 통해 확인할 수 있습니다.

#### # 프로그램 코드

- ZRN 버튼 클릭할 때

```
if(mConnType == CONN_HX_20)
{
    HxSetGB(0, 1,3, TRUE);
    Sleep(200);
    HxSetGB(0, 1,3, FALSE);
}
```

```
else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 1,3, TRUE);
    Sleep(200);
    HXSetBit(HX_G, 1,3, FALSE);
}
```

#### # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> </ul>
---------	---

	<ul style="list-style-type: none"> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	<pre>HxSetGB(0, 1,3, TRUE); Sleep(200); HxSetGB(0, 1,3, FALSE);</pre>
HXApi	<pre>void HXSetBit(long name, long addr, long bitno, long value);</pre> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	<pre>HXSetBit(HX_G, 1,3, TRUE); Sleep(200); HXSetBit(HX_G, 1,3, FALSE);</pre>

## 6.2.4 AUTO 모드

1. MODE 에 있는 AUTO 버튼을 클릭합니다.



2. NC READY 버튼을 3 초 이상 눌러줍니다.



3. CYCLE START 버튼을 눌러줍니다.



4. MAIN PROGRAM 에 등록된 NC 파일을 읽어 동작합니다.
5. 동작을 끝내고 싶을 시 CYCLE STOP 과 RESET 버튼을 눌러 취소합니다.

### # 프로그램 코드

- AUTO 버튼 클릭할 때

```
if(mConnType == CONN_HX_20)
{
    HxSetGB(0, 1, 0, TRUE);
    Sleep(200);
    HxSetGB(0, 1, 0, FALSE);
}
```

```

else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 1, 0, TRUE);
    Sleep(200);
    HXSetBit(HX_G, 1, 0, FALSE);
}

```

#### # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetGB(0, 1, 0, TRUE); Sleep(200); HxSetGB(0, 1, 0, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_G, 1, 0, TRUE); Sleep(200); HXSetBit(HX_G, 1, 0, FALSE);

### 6.2.5 MDI 모드

1. MODE 에 있는 MDI 버튼을 클릭합니다.



(현재 HX API TESTER SAMPLE 에서는 해당 기능을 제공하지 않습니다.)

#### # 프로그램 코드

- MDI 버튼 클릭 할 때

```

if(mConnType == CONN_HX_20)
{
    HxSetGB(0, 1,1, TRUE);
    Sleep(200);
    HxSetGB(0, 1,1, FALSE);
}

```



```

else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 1,1, TRUE);
    Sleep(200);
    HXSetBit(HX_G, 1,1, FALSE);
}

```

#### # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetGB(0, 1,1, TRUE); Sleep(200); HxSetGB(0, 1,1, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_G, 1,1, TRUE); Sleep(200); HXSetBit(HX_G, 1,1, FALSE);

### 6.2.6 EDIT 모드

1. MODE 에 있는 EDIT 버튼을 클릭합니다.



(현재 HX API TESTER SAMPLE 에서는 해당 기능을 제공하지 않습니다.)

#### # 프로그램 코드

- EDIT 버튼 클릭할 때

```

if(mConnType == CONN_HX_20)
{
    HxSetGB(0, 1,7, TRUE);
    Sleep(200);
    HxSetGB(0, 1,7, FALSE);
}

```

```

else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 1,7, TRUE);
    Sleep(200);
    HXSetBit(HX_G, 1,7, FALSE);
}

```

#### # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetGB(0, 1,7, TRUE); Sleep(200); HxSetGB(0, 1,7, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_G, 1,7, TRUE); Sleep(200); HXSetBit(HX_G, 1,7, FALSE);

## 6.3 NC 제어 설정

### 6.3.1 NC READY



마우스 DOWN 을 3 초 이상 지속해야 해당 READY 가 준비가 됩니다.

#### # 프로그램 코드

- NC READY 버튼 PRESSED 일 때

```
if (mConnType == CONN_HX_20)
{
    HxSetXB(0, 200, 0, TRUE);
}
```

```
else if (mConnType == CONN_HX)
{
    HXSetBit(HX_X, 200, 0, TRUE);
}
```

- NC READY 버튼 UP 일 때

```
if (mConnType == CONN_HX_20)
{
    HxSetXB(0, 200, 0, FALSE);
}
```

```
else if (mConnType == CONN_HX)
{
    HXSetBit(HX_X, 200, 0, FALSE);
}
```

#### # 예제 코드

Hx20Api	<b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	// 버튼 PRESSED 일 때 HxSetXB(0, 200, 0, TRUE);  // 버튼 UP 일 때 HxSetXB(0, 200, 0, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	// 버튼 PRESSED 일 때 HXSetBit(HX_X, 200, 0, TRUE);  // 버튼 UP 일 때 HXSetBit(HX_X, 200, 0, FALSE);

### 6.3.2 RESET



#### # 프로그램 코드

- RESET 버튼 클릭 할 때

```
if(mConnType == CONN_HX_20)
{
    HxSetXB(0, 200,1, TRUE);
    Sleep(200);
    HxSetXB(0, 200,1, FALSE);
}
```

```
else if(mConnType == CONN_HX)
{
    HXSetBit(HX_X, 200,1, TRUE);
    Sleep(200);
    HXSetBit(HX_X, 200,1, FALSE);
}
```

#### # 예제 코드

Hx20Api	<b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetXB(0, 200,1, TRUE); Sleep(200); HxSetXB(0, 200,1, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_X, 200,1, TRUE); Sleep(200); HXSetBit(HX_X, 200,1, FALSE);

### 6.3.3 CYCLE START



## # 프로그램 코드

- CYCLE START 버튼 클릭할 때

```
if(mConnType == CONN_HX_20)
{
    HxSetXB(0, 201,0, TRUE);
    Sleep(200);
    HxSetXB(0, 201,0, FALSE);
}
```

```
else if(mConnType == CONN_HX)
{
    HXSetBit(HX_X, 201,0, TRUE);
    Sleep(200);
    HXSetBit(HX_X, 201,0, FALSE);
}
```

## # 예제 코드

Hx20Api	<b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetXB(0, 201,0, TRUE); Sleep(200); HxSetXB(0, 201,0, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_X, 201,0, TRUE); Sleep(200); HXSetBit(HX_X, 201,0, FALSE);

## 6.3.4 CYCLE STOP



## # 프로그램 코드

- CYCLE STOP 버튼 클릭할 때

```

if(mConnType == CONN_HX_20)
{
    HxSetXB(0, 201,1, TRUE);
    Sleep(200);
    HxSetXB(0, 201,1, FALSE);
}

```

```

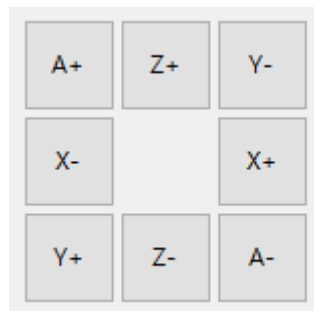
else if(mConnType == CONN_HX)
{
    HXSetBit(HX_X, 201,1, TRUE);
    Sleep(200);
    HXSetBit(HX_X, 201,1, FALSE);
}

```

#### # 예제 코드

Hx20Api	<b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetXB(0, 201,1, TRUE); Sleep(200); HxSetXB(0, 201,1, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_X, 201,1, TRUE); Sleep(200); HXSetBit(HX_X, 201,1, FALSE);

## 6.4 JOG 좌표 설정



## # 프로그램 코드

- JOG X 버튼 PRESSED 일 때

```
if (mConnType == CONN_HX_20)
{
    HxSetXB(0, 201, 7, TRUE);
}
```

```
else if (mConnType == CONN_HX)
{
    HXSetBit(HX_X, 201, 7, TRUE);
}
```

- JOG X 버튼 UP 일 때

```
if (mConnType == CONN_HX_20)
{
    HxSetXB(0, 201, 7, FALSE);
}
```

```
else if (mConnType == CONN_HX)
{
    HXSetBit(HX_X, 201, 7, FALSE);
}
```

## # 예제 코드

Hx20Api	<b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	// 버튼 PRESSED 일 때 HxSetXB(0, 201, 7, TRUE);  // 버튼 UP 일 때 HxSetXB(0, 201, 7, FALSE);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	// 버튼 PRESSED 일 때 HXSetBit(HX_X, 201, 7, TRUE);  // 버튼 UP 일 때 HXSetBit(HX_X, 201, 7, FALSE);

(기타 JOG 버튼들도 JOG X 와 동작 동일함)

## 7. INFO 화면 구성 및 사용 방법

The screenshot shows the 'Information / Setup' screen with the following components and annotations:

- Select Axis :** A dropdown menu currently showing 'AXIS 01'. An arrow points to it with the label '축 선택' (Axis Selection).
- Limit Value(한계 설정):** A section containing:
  - ☐ USE
  - Absolute Position:** Fields for 'Min : ' and 'Max : '.
- Set Value:** A section containing:
  - JOG Move:** A 'Velocity' field.
  - Target Pos. Move:** A 'Velocity' field.
  - Step Move:** A 'Velocity' field.
- Homing Move:** A section containing:
  - Fields for 'Accel : (ms)', '1st Decel Velocity :', 'Velocity :', and '2nd Decel Velocity :'.
  - A field for '금지영역 : ' followed by 'mm,de'.
  - Checkboxes for ☐ 임의 원점 and ☐ DOG 사용.
- RESET:** A button at the bottom. An arrow points to it with the label '리셋하기' (Reset).

On the right side, an arrow points to the 'Set Value' and 'Homing Move' sections with the label '파라미터 모니터링 및 저장하기' (Parameter Monitoring and Saving).

### 7.1 파라미터 설정

1. 원하는 축 선택을 한다. (AXIS 01~06)
2. TEXTBOX 에 업데이트 된 값을 모니터링한다.
3. 값을 저장할 경우, TEXTBOX 에 값을 기입한 후 ENTER 를 누른다.

#### # 프로그램 코드

- 파라미터 값을 모니터링할 때

```
int nUseOriginDog = (int)HxGetPMF(0, 1773 + nAxis);
```



```
int nUseSoftlimit = (int) HXGetRegister64(HX_PM, 3378 + nAxis);
```

- 파라미터 값을 저장할 때

```
HxSetPMF(0, 3410 + nAxis * 2, GetDataEdt(m_EdtLimitMin));
```

```
HXSetRegister64(HX_PM, 3410 + nAxis * 2, GetDataEdt(m_EdtLimitMin));
```

## # 예제 코드

Hx20Api	<p><b>Double HxGetPMF(Int32 port, Int32 addr);</b></p> <ul style="list-style-type: none"> <li>- PM 맵 지정된 주소에서 파라미터 값을 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- double 형으로 반환함</li> </ul> <p><b>void HxSetPMF(Int32 port, Int32 addr, Double real);</b></p> <ul style="list-style-type: none"> <li>- PU 맵 지정된 주소에 파라미터 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Real 에 저장할 값을 double 형으로 기입</li> </ul> <p>// 파라미터 값을 모니터링 할 때 int nUseSoftlimit = (int)HxGetPMF(0, 3378 + nAxis);</p> <p>// 파라미터 값을 저장할 때 HxSetPMF(0, 3410 + nAxis * 2, GetDataEdt(m_EdtLimitMin));</p>
HXApi	<p><b>double HXGetRegister64(long name, long addr);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에서 값을 64 비트로 가져오는 API</li> <li>- Name 에 원하는 HXMAP 에 있는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- double 형으로 반환함</li> </ul> <p><b>void HXSetRegister64(long name, long addr, double value);</b></p> <ul style="list-style-type: none"> <li>- HXMap 맵, 지정된 주소에 파라미터 값을 저장하는 API</li> <li>- Name 에 원하는 HXMAP 에 있는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Value 에 저장할 값을 double 형으로 기입</li> </ul> <p>// 파라미터 값을 모니터링 할 때 int nUseSoftlimit = (int)HXGetRegister64(HX_PM, 3378 + nAxis);</p> <p>// 파라미터 값을 저장할 때 HXSetRegister64(HX_PM, 3410 + nAxis * 2, GetDataEdt(m_EdtLimitMin));</p>

## 7.2 RESET 설정

### 1. RESET 버튼을 클릭한다

(버튼을 클릭하면 해당 비트를 TRUE 로 저장하고 일정시간 후에 다시 FALSE 로 저장함)

## # 프로그램 코드

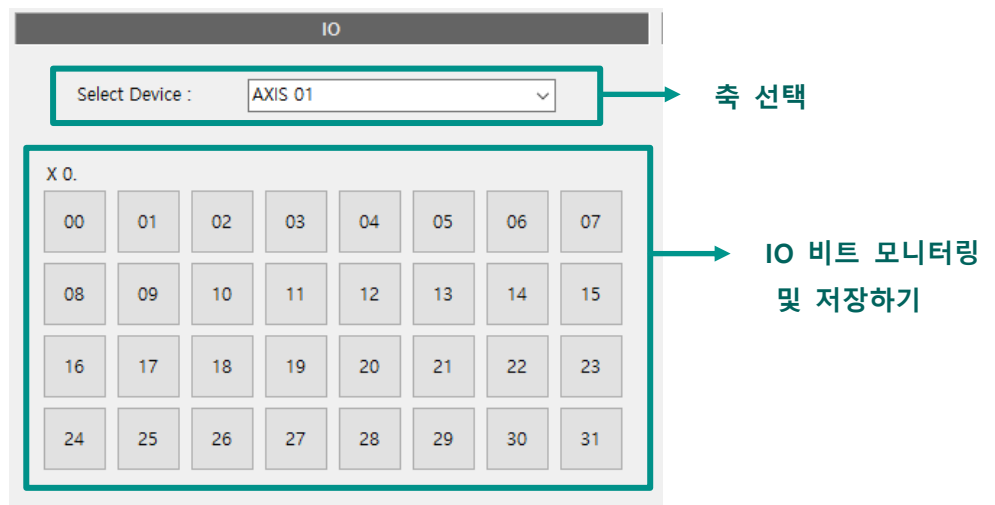
```
if(mConnType ==CONN_HX_20)
{
    HxSetGB(0, 8, 0, true);
    Sleep(200);
    HxSetGB(0, 8, 0, false);
}
```

```
}else if(mConnType == CONN_HX)
{
    HXSetBit(HX_G, 8, 0, true);
    Sleep(200);
    HXSetBit(HX_G, 8, 0, false);
}
```

## # 예제 코드

Hx20Api	<b>void HxSetGB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b> <ul style="list-style-type: none"> <li>- G 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 원하는 bool 형의 값 기입</li> </ul>
	HxSetGB(0, 8, 0, true); Sleep(200); HxSetGB(0, 8, 0, false);
HXApi	<b>void HXSetBit(long name, long addr, long bitno, long value);</b> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul>
	HXSetBit(HX_G, 8, 0, true); Sleep(200); HXSetBit(HX_G, 8, 0, false);

## 8. IO 화면 구성 및 사용 방법



1. 원하는 축 선택을 한다(AXIS 01~06)
2. IO 비트 모니터링한다
3. 원하는 비트 주소에 TRUE, FALSE 값을 저장할 경우, 해당 비트 주소에 해당하는 버튼을 클릭한다. (비트 버튼은 토글 버튼임)

### # 프로그램 코드

- IO 비트 값을 모니터링 할 때

```
if(mConnType == CONN_HX_20)
{
    for(int i =0 ; i <32; i++)
    {
        bool val = HxGetXB(0,m_curr_Index, i);

        if(val == false)
        {
            m_btnArr[i].SetCheck(0);
        }
        else if(val == true)
        {
            m_btnArr[i].SetCheck(1);
        }
    }
}
```

```

}else if(mConnType == CONN_HX)
{
    for(int i =0 ; i <32; i++)
    {
        bool val = HXGetBit(HX_X, m_curr_Index, i);

        if(val == false)
        {
            m_btnArr[i].SetCheck(0);
        }
        else if(val == true)
        {
            m_btnArr[i].SetCheck(1);
        }
    }
}

```

- IO 비트 값을 저장할 때

```

if(check == 0)
{
    if(mConnType == CONN_HX_20)
    {
        HxSetXB(0, m_curr_Index, idx, false);
    }
    else if(mConnType == CONN_HX)
    {
        HXSetBit(HX_X, m_curr_Index, idx, false);
    }
}

```

```

}else if(check ==1)
{
    if(mConnType == CONN_HX_20)
    {
        HxSetXB(0, m_curr_Index, idx, true);
    }
    else if(mConnType == CONN_HX)
    {
        HXSetBit(HX_X, m_curr_Index, idx, true);
    }
}

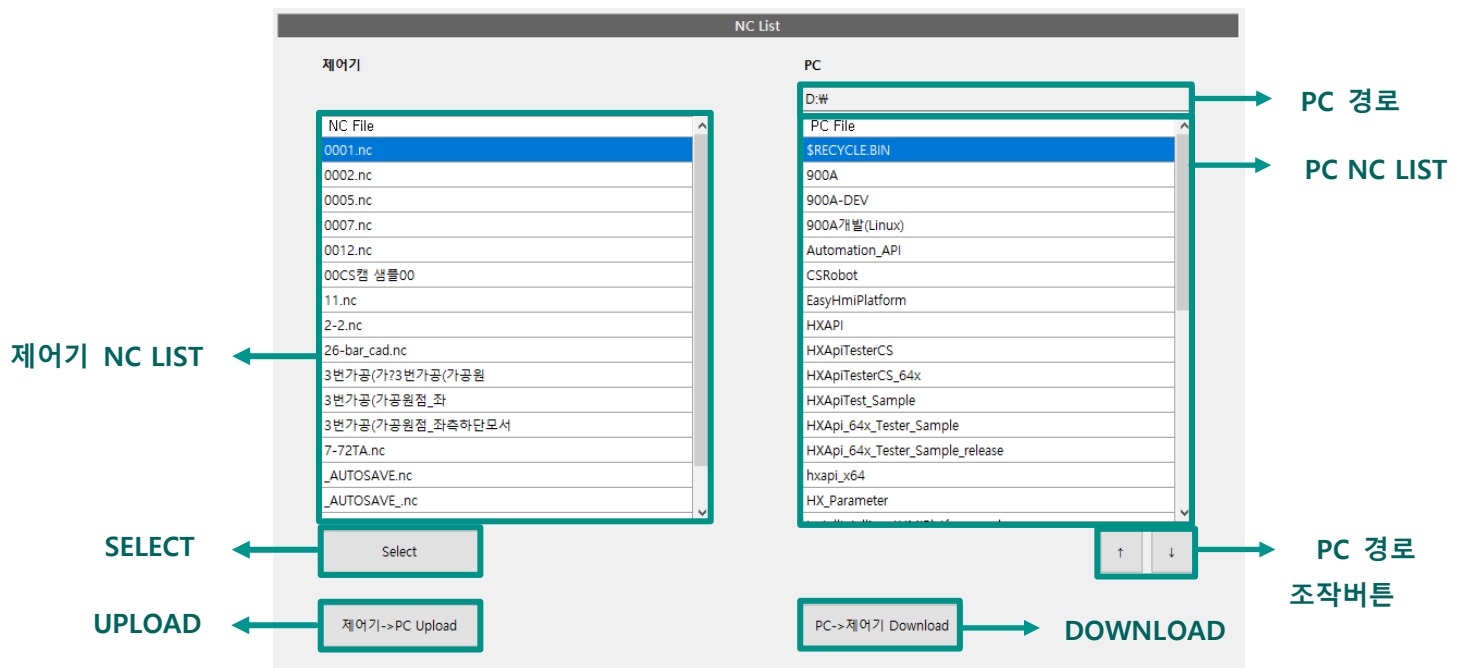
```

# 예제 코드

Hx20Api	<p><b>Bool HxGetXB(Int32 port, Int32 addr, Int32 bit);</b></p> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Bool 형으로 반환함</li> </ul> <p><b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Bool onoff);</b></p> <ul style="list-style-type: none"> <li>- X 맵 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> </ul>
---------	---

	<p>Onoff 에 원하는 bool 형의 값 기입</p>
	<pre>// 비트를 모니터링할 때 bool val = HxGetXb(0,m_curr_Index, i);  // 비트를 저장할 때 HxSetXB(0, m_curr_Index, idx, true);</pre>
HXApi	<p><b>long HXGetBit(long name, long addr, long bitno);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에서 값을 가져오는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- long 형으로 반환함</li> </ul> <p><b>void HXSetBit(long name, long addr, long bitno, long value);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul> <pre>// 비트를 모니터링할 때 bool val = HXGetBit(HX_X, m_curr_Index, i);  // 비트를 저장할 때 HXSetBit(HX_X, m_curr_Index, idx, true);</pre>

## 9. NC LIST 화면 구성 및 사용 방법



1. HX 제어기와 연결하면 NC LIST 와 PC LIST 가 구성됩니다.
  - 제어기 NC LIST 는 HX 제어기 실행파일이 있는 경로 → NC 폴더 안에 있는 NC 파일들로 구성됩니다.
  - PC NC LIST 는 지정되어 있는 기본 경로에 있는 NC 파일 및 폴더들로 구성됩니다.  
(기본 경로는 D 드라이브입니다. 기본 경로 변경 시 dlg\_NcList.cpp 에서 DEFAULT\_PATH 변수를 수정하길 바랍니다.)
2. PC 경로 이동 시 PC 경로 조작버튼을 눌러 이동합니다.
  - ↑ 화살표는 현재 경로보다 상위 폴더로 이동합니다.  
(기본 경로에서는 더 이상 이동하지 않습니다.)
  - ↓ 화살표는 현재 PC NC LIST 에서 선택된 폴더로 이동합니다.
  - 현재 나타내고 있는 경로는 PC 경로에서 확인할 수 있습니다.

### 9.1 NC LIST 불러오기

NC 파일의 개수와 파일명을 읽어 리스트를 구성합니다.

해당 NC 파일 개수는 HX 제어기를 실행할 때 NC 경로에 있는 NC 파일 개수를 읽어옵니다. HX 제어기 실행 후 NC 파일 개수의 변화가 있어도 업데이트 되지 않습니다. 업데이트를 원할 시 HX 제어기를 종료 후 다시 실행해주시길 바랍니다.

더불어 NC 파일 개수를 불러올 시 NC 경로에 NC 파일 개수가 약 65 개 초과일 경우 API 가 제대로 값을 반환하지 못하므로 개수에 주의하시길 바랍니다.  
NC FILE 명을 읽을 시 글자 수 20 자 초과 파일에 대해서는 제대로 가져오지 못하므로 파일명 길이에 주의하시길 바랍니다.

#### # 프로그램 코드

```
if(mConnType == CONN_HX_20)
{
    Int32 count = HxGetNcFileCount();
    for(int i =0; i<count; i++)
    {
        //char name[255] ="one.nc";
        char name[255] = {0,};
        HxGetNcFileStr(i, name);

        char *result = NULL;
        result= strtok(name, ".");
        m_ncList.InsertItem(i, result);
    }
}
```

```
else if(mConnType == CONN_HX)
{
    int count = HXGetNcFileCount();
    for(int i =0; i<count; i++)
    {
        char name[255] = {0,};
        HXGetNcFileStr(i, name);

        char *result = NULL;
        result= strtok(name, ".");
        m_ncList.InsertItem(i, result);
    }
}
```

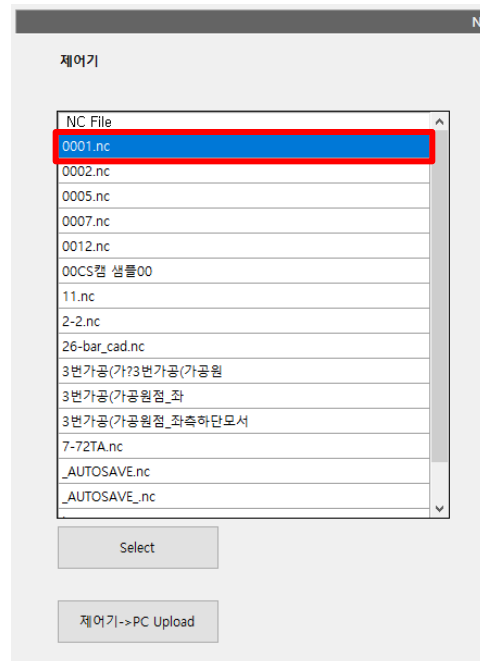
#### # 예제 코드

Hx20Api	<p><b>Int32 HxGetNcFileCount();</b></p> <ul style="list-style-type: none"> <li>- 현재 NC PATH 경로에 있는 NC 파일의 개수를 가져오는 API</li> <li>- int 형으로 개수를 반환</li> </ul> <p><b>Bool HxGetNcFileStr(Int32 index, Char* str);</b></p> <ul style="list-style-type: none"> <li>- index 에 해당되는 NC 파일명을 가져오는 API</li> <li>- index 에 원하는 넘버를 기입</li> <li>- NC 파일명을 str 변수에 담아서 반환</li> </ul> <p>// nc 파일 개수 가져올 때 Int32 count = HxGetNcFileCount();</p> <p>// nc 파일명 가져올 때 char name[255] = {0,};</p>
---------	---

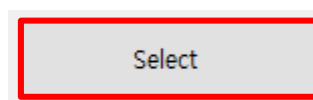
	HxGetNcFileStr(i, name);
HXApi	<b>int HXGetNcFileCount();</b> <ul style="list-style-type: none"> <li>- 현재 NC PATH 경로에 있는 NC 파일의 개수를 가져오는 API</li> <li>- int 형으로 개수를 반환</li> </ul> <b>bool HXGetNcFileStr(int index, char* str);</b> <ul style="list-style-type: none"> <li>- index 에 해당되는 NC 파일명을 가져오는 API</li> <li>- index 에 원하는 넘버를 기입</li> <li>- NC 파일명을 str 변수에 담아서 반환</li> </ul>
	<pre>// nc 파일 개수 가져올 때 int count = HXGetNcFileCount();  // nc 파일명 가져올 때 char name[255] = {0,}; HXGetNcFileStr(i, name);</pre>

## 9.2 SELECT

- 제어기 NC LIST 에서 선택하고자 하는 NC 파일을 클릭합니다.



- SELECT 버튼을 클릭합니다.



- HX 제어기에서 MAIN, SUB 프로그램명이 바뀐 것을 확인합니다.

Main	0001
Sub	0001



## # 프로그램 코드

```
if(mConnType == CONN_HX_20)
{
    HxSetMainProg(0, stName);
    HxSetSubProg(0, stName);

    char ncfile_path[255] = {0,};
    HxGetSysPath(0, ncfile_path);

    char *ncF = "NC###";
    strcat(ncfile_path, ncF) ;

    HxSetNcPath(0, ncfile_path);
}
```

```
else if(mConnType == CONN_HX)
{
    HXSetMainProgram(stName);
    HXSetSubProgram(stName);

    char ncfile_path[255] = {0,};
    HXGetSystemPath(ncfile_path);

    char *ncF = "NC###";
    strcat(ncfile_path, ncF) ;

    HXSetNcPath(ncfile_path);
}
```

## # 예제 코드

Hx20Api	<p><b>void HxSetMainProg(Int32 port, char *name);</b></p> <ul style="list-style-type: none"> <li>- 선택한 NC 파일명을 MAIN PROGRAM 으로 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- File 에 등록할 NC FILE 명을 char 형으로 기입</li> <li>- File 은 .nc 를 제외한 파일명을 기입 해야함</li> </ul> <p><b>void HxSetSubProg(Int32 port, char *name);</b></p> <ul style="list-style-type: none"> <li>- 선택한 NC 파일명을 SUB PROGRAM 으로 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- File 에 등록할 NC FILE 명을 char 형으로 기입</li> <li>- File 은 .nc 를 제외한 파일명을 기입 해야함</li> </ul> <p><b>void HxGetSysPath(Int32 port, char *path)</b></p> <ul style="list-style-type: none"> <li>- 시스템 경로를 가지고 오는 API</li> <li>- Port 는 보통 0 임</li> <li>- 시스템 경로를 char 형 path 변수에 담아서 반환함</li> </ul> <p><b>void HxSetNcPath(Int32 port, char *path);</b></p> <ul style="list-style-type: none"> <li>- NC PATH 를 저장하는 API</li> </ul>
---------	---

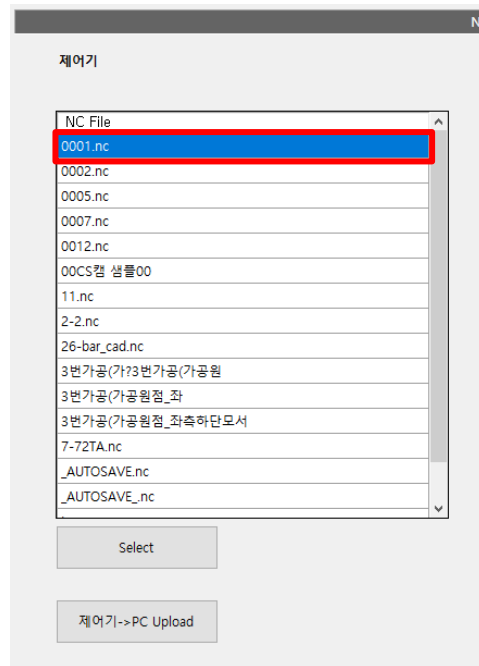
	<ul style="list-style-type: none"> <li>- Port 는 보통 0 임</li> <li>- Path 에 등록할 NC PATH 를 char 형으로 기입</li> <li>- PATH 경로 덧붙일 때 맨 마지막에 "WW" 표시가 들어가야함</li> </ul>
	<pre>// NC 파일을 메인 프로그램에 등록할 때 HxSetMainProg(0, stName);  // NC 파일을 서브 프로그램에 등록할 때 HxSetSubProg(0, stName);  // SYSTEM 경로 가져올 때 char ncfile_path[255] = {0,}; HxGetSysPath(0, ncfile_path);  // NC PATH 등록할 때 HxSetNcPath(0, ncfile_path);</pre>
HXApi	<p><b>void HXSetMainProgram(char *name);</b></p> <ul style="list-style-type: none"> <li>- 선택한 NC 파일명을 MAIN PROGRAM 으로 저장하는 API</li> <li>- Name 에 등록할 NC FILE 명을 char 형으로 기입</li> <li>- File 은 .nc 를 제외한 파일명을 기입 해야함</li> </ul> <p><b>void HXSetSubProgram(char *name)</b></p> <ul style="list-style-type: none"> <li>- 선택한 NC 파일명을 SUB PROGRAM 으로 저장하는 API</li> <li>- Name 에 등록할 NC FILE 명을 string 형으로 기입</li> <li>- File 은 .nc 를 제외한 파일명을 기입 해야함</li> </ul> <p><b>void HXGetSystemPath(char *path)</b></p> <ul style="list-style-type: none"> <li>- 시스템 경로를 가지고 오는 API</li> <li>- 시스템 경로를 char 형 path 변수에 담아서 반환함</li> </ul> <p><b>void HXSetNcPath(char *path)</b></p> <ul style="list-style-type: none"> <li>- NC PATH 를 저장하는 API</li> <li>- Path 에 등록할 NC PATH 를 char 형으로 기입</li> <li>- PATH 경로 덧붙일 때 맨 마지막에 "WW" 표시가 들어가야함</li> </ul>
	<pre>// NC 파일 메인 프로그램에 등록할 때 HXSetMainProgram(stName);  // NC 파일 서브 프로그램에 등록할 때 HXSetSubProgram(stName);  // SYSTEM 경로 가져올 때 char ncfile_path[255] = {0,}; HXGetSystemPath(ncfile_path);  // NC PATH 등록할 때 HXSetNcPath(ncfile_path);</pre>

### 9.3 UPLOAD

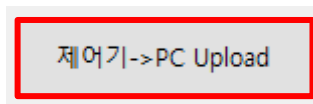
제어기 NC 폴더에서 선택한 NC 파일을 PC 로 복사하는 기능입니다.

복사되는 PC 경로는 현재 PC NC LIST 가 나타내는 경로입니다.

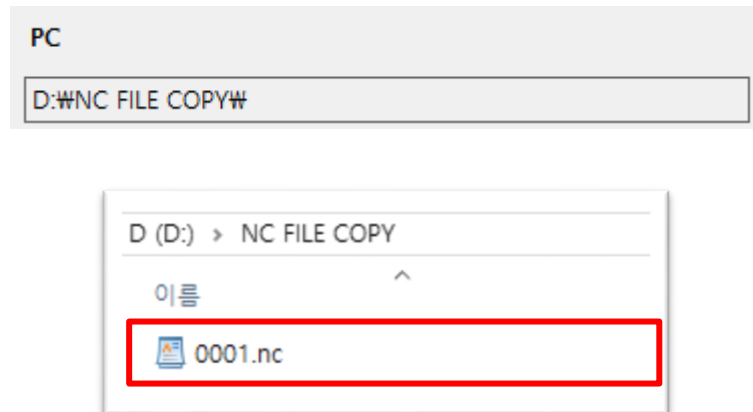
1. 제어기 NC LIST 에서 복사할 NC 파일을 선택합니다.



2. UPLOAD 버튼을 클릭합니다.



3. 지정된 PC 경로에서 복사된 NC 파일을 확인합니다.



# 프로그램 코드

```
if(mConnType == CONN_HX_20)
{
    char tmp_name[255] = {0,};
    HxGetNcFileStr(nItem, tmp_name);
    HxFtpUploadFile(0, c_dir_path, tmp_name);
}
```

```

}else if(mConnType == CONN_HX)
{
    char tmp_name[255] = {0,};
    HXGetNcFileStr(nItem, tmp_name);
    HXUploadFile(c_dir_path, tmp_name);
}

```

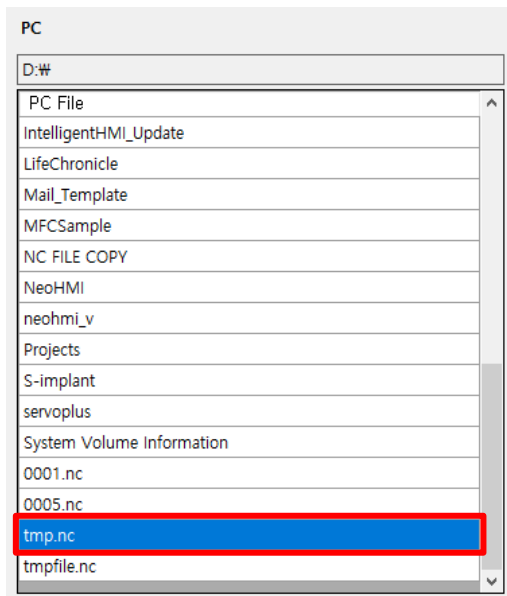
#### # 예제 코드

Hx20Api	<b>void HxFtpUploadFile(Int32 port, const char* dir, const char* name);</b> <ul style="list-style-type: none"> <li>- 선택한 NC 파일을 PC 로 복사하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Dir 에 복사할 PC 경로를 기입</li> <li>- Dir 경로 맨 마지막에 “WW” 표시가 들어가야함</li> <li>- Name 에 복사할 NC 파일명을 char 형으로 기입</li> <li>- Name 에 .nc 가 꼭 붙은 파일명으로 기입 해야함</li> </ul>
	//업로드 하기 char tmp_name[255] = {0,}; HxGetNcFileStr(nItem, tmp_name); HxFtpUploadFile(0, c_dir_path, tmp_name);
HXApi	<b>void HXUploadFile(const char* dir, const char* name);</b> <ul style="list-style-type: none"> <li>- 선택한 NC 파일을 PC 로 복사하는 API</li> <li>- Dir 에 복사할 PC 경로를 기입</li> <li>- Dir 경로 맨 마지막에 “WW” 표시가 들어가야함</li> <li>- Name 에 복사할 NC 파일명을 string 형으로 기입</li> <li>- Name 에 .nc 가 꼭 붙은 파일명으로 기입 해야함</li> </ul>
	// 업로드 하기 char tmp_name[255] = {0,}; HXGetNcFileStr(nItem, tmp_name); HXUploadFile(c_dir_path, tmp_name);

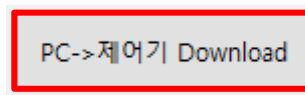
## 9.4 DOWNLOAD

PC NC LIST 에서 선택한 NC 파일을 제어기 NC 폴더로 복사하는 기능입니다.

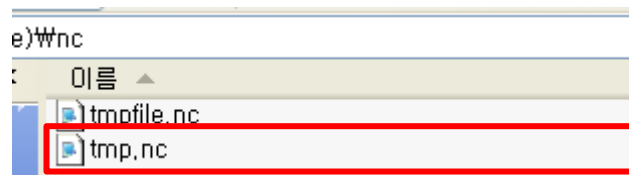
1. PC NC LIST 에서 복사할 NC 파일을 선택합니다.



2. DOWNLOAD 버튼을 클릭합니다.



3. 제어기 NC 폴더에서 복사된 NC 파일을 확인합니다.



# 프로그램 코드

```
if(mConnType == CONN_HX_20)
{
    HxFtpDownloadFile(0, c_dir_path, c_nc_file);
}
```

```
else if(mConnType == CONN_HX)
{
    HXDownloadFile(c_dir_path, c_nc_file);
}
```

# 예제 코드

Hx20Api	<b>void HxFtpDownloadFile(Int32 port, const char* dir, const char* name);</b> <ul style="list-style-type: none"> <li>- 선택한 NC 파일을 제어기로 복사하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Dir 에 NC 파일이 있는 PC 경로를 기입</li> <li>- Dir 경로 맨 마지막에 “\” 표시가 들어가야함</li> <li>- Name 에 복사할 NC 파일명을 string 형으로 기입</li> </ul>
---------	---

	- Name 에 .nc 가 꼭 붙은 파일명으로 기입 해야함
	//다운로드 하기 HxFtpDownloadFile(0, c_dir_path, c_nc_file);
HXApi	<b>void HXDownloadFile(const char* dir, const char* name);</b> <ul style="list-style-type: none"> <li>- 선택한 NC 파일을 제어기로 복사하는 API</li> <li>- Dir 에 NC 파일이 있는 PC 경로를 기입</li> <li>- Dir 경로 맨 마지막에 “\” 표시가 들어가야함</li> <li>- Name 에 복사할 NC 파일명을 string 형으로 기입</li> <li>- Name 에 .nc 가 꼭 붙은 파일명으로 기입 해야함</li> </ul>
	// 다운로드 하기 HXDownloadFile(c_dir_path, c_nc_file);

# 10. MAP TEST 화면 구성 및 사용 방법

Tester

Map Name Address Bit Axis GET

HX MAP 에서 값을 가져오는 부분

Map Name Address Bit Axis

Set Value SET

확인

HX MAP 에서 값을 저장하고 확인하는 부분

## 10.1 GET 기능

1. 확인하고자 하는 맵 이름, 주소, 비트 축 넘버를 기입합니다.  
(PS 맵인 경우 AXIS 를 기입해야 합니다.)

Map Name Address Bit Axis GET

HX\_Y 10 0

2. GET 버튼을 클릭합니다.

Map Name Address Bit Axis GET

HX\_Y 10 0

3. 가져온 값을 확인합니다.

Map Name	Address	Bit	Axis
HX_Y	10	0	
<input type="text" value="true"/>			GET

## 10.2 SET 기능

- 저장하고자 하는 맵 이름, 주소, 비트 축 넘버 및 저장하고자 하는 값을 Set Value 란에 기입합니다.  
(PS 맵인 경우 AXIS 를 기입해야 합니다.)  
(비트 값 입력 시 0 은 false 로 처리되며, 1 은 true 로 처리됩니다. 그 외의 값은 모두 false 로 처리됩니다.)

Map Name	Address	Bit	Axis
HX_Y	10	0	
Set Value			1
확인			SET

- SET 버튼을 클릭합니다.

Map Name	Address	Bit	Axis
HX_Y	10	0	
Set Value			1
확인			SET

- 값이 제대로 저장이 됐는지 확인란에서 확인합니다.

Map Name	Address	Bit	Axis
HX_Y	10	0	
Set Value			1
확인			true

### # 프로그램 코드

- GET : 비트일 경우

```
bool bval = HxGetXB(0, address, bit);
```

```
long bit = HXGetBit(l_mapname, i_address, i_bit);
```

- GET : 파라미터일 경우

```
double dval = HxGetPAF(0, address);
```

```
double reg= HXGetRegister64(l_val, address);
```



- GET : 텍스트일 경우

```
const char *val1 = HxGetString(0, HX_STR_MACHINE);
```

```
HxGetString(HX_STR_LANGUAGE, cval);
```

- SET : 비트일 경우

```
HxSetXB(0, address, bit, check);
```

```
HXSetBit(l_mapname, i_address, i_bit, check);
```

- SET : 파라미터일 경우

```
HxSetPAF(0, address, dvalue);
```

```
HXSetRegister64(lval, address, lvalue);
```

- SET : 텍스트일 경우

```
HxSetString(0, HX_STR_MACHINE, st);
```

```
HXSetString(HX_STR_AXIS, st);
```

#### # 예제 코드

Hx20Api	<p><b>Bool HxGetXB(Int32 port, Int32 addr, Int32 bit);</b></p> <ul style="list-style-type: none"> <li>- X 맵, 지정된 주소에서 비트 값을 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- bool 형으로 비트 값을 반환</li> </ul> <p><b>Double HxGetPAF(Int32 port, Int32 addr);</b></p> <ul style="list-style-type: none"> <li>- PA 맵, 지정된 주소에서 값을 64 비트로 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- Addr 에 원하는 주소 기입</li> <li>- double 형으로 값을 반환함</li> </ul> <p><b>const char* HxGetString(Int32 port, Int32 name);</b></p> <ul style="list-style-type: none"> <li>- HXMap 맵에서 값을 string 형으로 가져오는 API</li> <li>- Port 는 보통 0 임</li> <li>- Name 에 원하는 맵 기입</li> <li>- char 형으로 값을 반환함</li> </ul> <p><b>void HxSetXB(Int32 port, Int32 addr, Int32 bit, Int32 onoff);</b></p> <ul style="list-style-type: none"> <li>- X 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- addr 에 원하는 주소 기입</li> <li>- Bit 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- Onoff 에 저장할 비트 값을 int 형으로 기입</li> </ul>
---------	--

	<p><b>void HxSetPAF(Int32 port, Int32 addr, Double value);</b></p> <ul style="list-style-type: none"> <li>- PA 맵, 지정된 주소에 값을 64 비트로 저장하는 API</li> <li>- Addr 에 원하는 주소를 기입</li> <li>- Real 에 저장할 값을 double 형으로 기입</li> </ul> <p><b>void HxSetString(Int32 port, Int32 name, char* resstr);</b></p> <ul style="list-style-type: none"> <li>- 맵에 원하는 char 형 변수를 저장하는 API</li> <li>- Port 는 보통 0 임</li> <li>- Name 에 저장할 맵 기입</li> <li>- resstr 에 저장할 char 형 변수를 기입</li> </ul> <pre>// GET : 비트일 경우 bool bval = HxGetXB(0, address, bit);  // GET : 파라미터일 경우 double dval = HxGetPAF(0, address);  // GET : 텍스트일 경우 const char *val1 = HxGetString(0, HX_STR_AXIS);  // SET : 비트일 경우 HxSetXB(0, address, bit, check);  // SET : 파라미터일 경우 HxSetPAF(0, address, dvalue);  // SET : 텍스트일 경우 char* st = LPSTR(LPCTSTR(value)); HxSetString(0, HX_STR_MACHINE, st);</pre>
HXApi	<p><b>long HXGetBit(long name, long addr, long bitno);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에서 비트 값을 가져오는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- long 형으로 비트 값을 반환(0: false, 1:true)</li> </ul> <p><b>double HXGetRegister64(long name, long addr);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에서 값을 64 비트로 가져오는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소를 기입</li> <li>- double 형으로 값을 반환함</li> </ul> <p><b>void HXGetString(int nName, char caStrVal[]);</b></p> <ul style="list-style-type: none"> <li>- 맵에서 값을 string 형으로 가져오는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- char 형 배열로 caStrVal 배열에 값을 담아서 반환함</li> </ul> <p><b>void HXSetBit(long name, long addr, long bitno, long value);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 비트 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> </ul>

	<ul style="list-style-type: none"> <li>- bitno 에 원하는 주소의 비트 기입(0~31 사이 값)</li> <li>- value 에 원하는 bool 형의 값 기입</li> </ul> <p><b>void HXSetRegister64(long name, long addr, double value);</b></p> <ul style="list-style-type: none"> <li>- 맵, 지정된 주소에 파라미터 값을 저장하는 API</li> <li>- Name 에 원하는 맵 기입</li> <li>- Addr 에 원하는 주소 기입</li> <li>- Value 에 저장할 값을 double 형으로 기입</li> </ul> <p><b>void HXSetString(int nName, char caStrVal[]);</b></p> <ul style="list-style-type: none"> <li>- 맵에 원하는 string 형 변수를 저장하는 API</li> <li>- nName 에 맵 기입</li> <li>- caStrVal 에 저장할 char 형 변수를 기입</li> </ul>
	<pre>// GET : 비트일 경우 long bit = HXGetBit(l_mapname, i_address, i_bit);  // GET : 파라미터일 경우 double reg= HXGetRegister64(lval, address)  // GET : 텍스트일 경우 char cval[225] = { 0, }; HXGetString(HX_STR_OPMSG, cval);  // SET : 비트일 경우 HXSetBit(l_mapname, i_address, i_bit, check);  // SET : 파라미터일 경우 HXSetRegister64(lval, address, lvalue);  // SET : 텍스트일 경우 char* st = LPSTR(LPCTSTR(value)); HXSetString(HX_STR_AXIS, st);</pre>