

Users

Manual

NEURO HUB シリーズ

# NH - LINK (PCI)

ホロン株式会社

## 目次

1. 概要
2. ハードウェア仕様
3. ボード設定
4. ニューロハブとの接続
5. 汎用入出力との接続
6. ドライバソフトについて
7. ドライブライブラリ(DLL関数)仕様

## 1. 概要

NH - LINK (PCI)は パーソナルコンピュータ(PC-AT互換機)と、ニューロハブチップを使用した各種モジュールとのインターフェースを行うボードです。

NH - LINK (PCI)は パーソナルコンピュータのPCIバスに挿入して御利用頂けます。

NH - LINK (PCI)は 1チャンネル当たり入力 1024 点、出力 1024 点のI / O制御が可能です。

フォトカプラ / Fによる汎用入出力も、入力 4 点、出力 4 点が用意されています。

最大 10 枚までのNH - LINK (PCI)を実装することで、大規模な制御システムの構築が可能です。

WindowsNT用のドライバが付属しており、アプリケーションソフトの開発が容易に行えます。

## 2. ハードウェア仕様

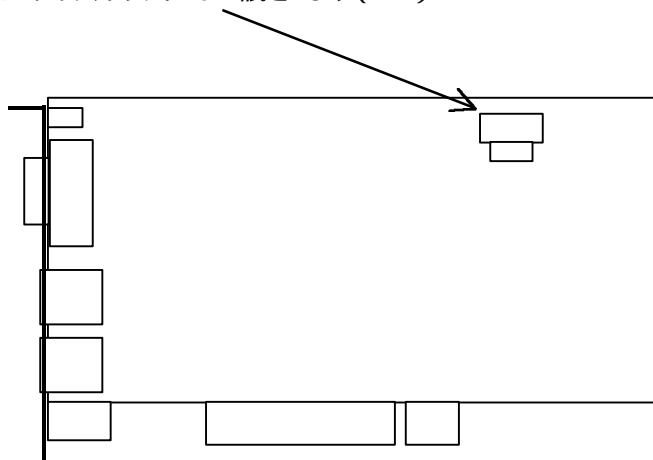
通信チャンネル数	2CH 8P-モジュラコネクタ × 2 (入力 1024 点・出力 1024 点 × 2CH)
局番割り当て	出力側: 局番 0 ~ 63 入力側: 局番 64 ~ 127 (1局 = 16 点)
通信速度	最大 1 Mbps (クロック: 32MHz)
ボードID	0 ~ 9 (ボード上のロータリスイッチにて設定)
汎用入出力	入力 4 点、出力 4 点 (フォトカプラアイソレーション) D-SUB 15Pコネクタ(メス)
使用環境	PCIバスを搭載した PC-AT 互換機

### 3. ボード設定

NH - LINK (PCI)は パーソナルコンピュータによって自動的にアドレスが割り付けられますので、アドレス等の設定は必要ありませんが、1台のパーソナルコンピュータで複数のNH - LINK (PCI)を使用するときは、識別用としてボードIDの設定が必要です。

設定は、1枚目を"0"、2枚目を"1"、... という順で、最大10枚まで同時に使用可能です。

このロータリスイッチにてID設定します(0~9)



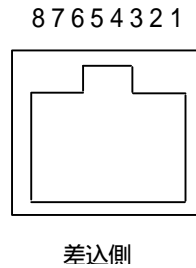
#### 4. ニューロハブとの接続

ニューロハブとのインターフェースは、下図の8P-モジュラコネクタを介して行われます。

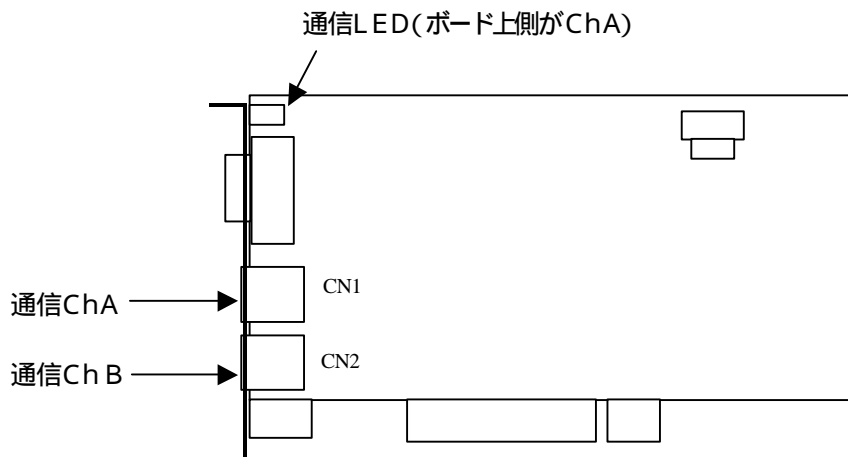
ケーブルは、市販の10BASE-T用のものがご利用になれますが、特にRS-A側とRS-B側の極性にはご注意下さい。

通信チャンネル：8P-モジュラコネクタ（2CH）

端子番号	線色*1	機能
1	白 / 橙	未使用
2	橙	未使用
3	白 / 緑	通信線RS-A側
4	青	未使用
5	白 / 青	未使用
6	緑	通信線RS-B側
7	白 / 茶	未使用
8	茶	未使用



\*1. 一般的な10BASE-Tストレートケーブルの場合。

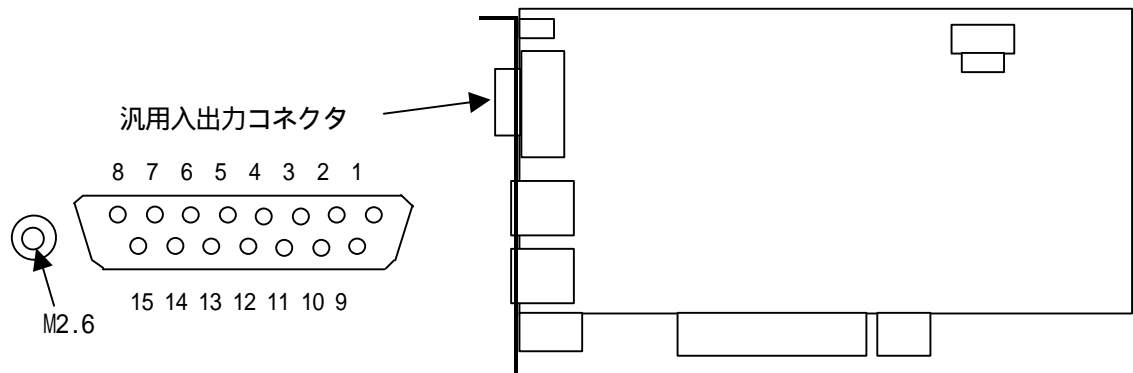


## 5. 汎用入出力との接続

汎用入出力とのインターフェースは、下図のD-SUBコネクタを介して行われます。

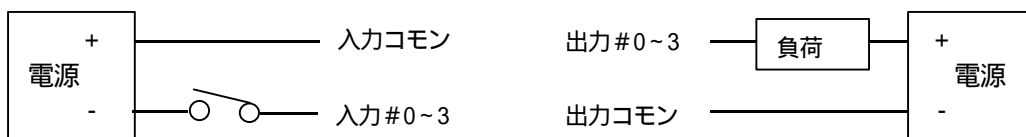
汎用入出力(フォトカプラ絶縁)：D-SUBコネクタ(15Pメス)

端子番号	機能	端子番号	機能
1	FG	9	NC
2	NC	10	出力コモン(-)
3	入力コモン(+)	11	出力コモン(-)
4	入力コモン(+)	12	出力#0
5	入力#0	13	出力#1
6	入力#1	14	出力#2
7	入力#2	15	出力#3
8	入力#3		



・最大定格(Ta=25 )

	項目	記号	定格
入力	入力端子間電圧	V <sub>IN</sub>	25 V(max)
	ON電圧	V <sub>IH</sub>	10 V(min)
	OFF電圧	V <sub>IL</sub>	5 V(max)
	入力電流	I <sub>F</sub>	6mA(max)
出力	出力端子間電圧	V <sub>CEO</sub>	25 V(max)
	出力電流	I <sub>C</sub>	100mA(max)
	出力残電圧	V <sub>O</sub>	1.2 V(max)



## 6. ドライバソフトについて

NH - LINK (PCI)には WindowsNT用のドライバが付属しており、アプリケーションソフトの開発が容易に行えます。

また、VBIによる簡単なサンプルソフトも付属していますので、開発環境でのアプリケーション評価用としてご使用下さい。

なお、ドライバのインストール方法は、添付フロッピー中の PlinkInst.txt を参照して下さい。

## 7. ドライバライブラリ(DLL)関数仕様

ライブラリ名称	: PLNKCTL.DLL
バージョン情報	: Ver 1.0a
対象OS	: Windows NT4.0/98/Me
対象言語	: Visual BASIC (6.0以降を推奨) Visual C (別途Libファイルが必要)
使用条件	: 1スレッドにつき1ハンドル (複数スレッドから同一ハンドルでDLL呼び出しを行わないで下さい。)

```
HANDLE HubOpen(long spd, long station, long* pn);
```

ニューロハブの制御を開始、装着されているニューロハブPCIボードをすべてオープンする。

引数spd - 通信速度を設定(0:標準 2:1/4)

引数station - 通信局数128を設定(固定)

戻り値: 正常終了> ハンドル値を返す(装着枚数は引数pnに格納される)  
異常終了> NULL(0)を返す

```
//C
HANDLE hHub;          //ハンドル値
long spd;             //通信速度
long station;        //通信局数
long n;               //枚数を格納する変数
hHub = HubOpen(spd,station,&n);
if(hHub == NULL){
    printf("HubOpen Error\n");
}else{
    printf("装着枚数 = %d\n",n);
}

//Basic
Declare Function HubOpen Lib "PInkctl.dll" (ByVal spd As Long,
                                           ByVal station As Long,
                                           ByRef data As Long) As Long

Dim hHub As long 'ハンドル値
Dim spd As long   '通信速度
Dim station As long '通信局数
Dim data as long  '枚数を格納する変数
hHub = HubOpen(spd,station,data)
if status = 0 Then Text1 = "HubOpen Error"
```

```
void HubClose(HANDLE hHub);
```

オープンされていたニューロハブPCIボードをすべてクローズする。

戻り値: 無し

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
HubClose(hHub);

//Basic
Declare Sub HubClose Lib "PInkctl.dll" (ByVal hHub As Long)
Dim hHub As long 'HubOpen で得たハンドル値
HubClose(hHub)
```



```
void HubAllClearDp(HANDLE hHub);
```

すべてのボードのD Pメモリー(送受信 ステータスとも)を0クリアする

戻り値: 無し

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
HubAllClearDp(hHub);

//Basic
Declare Sub HubAllClearDp Lib "Plnkctl.dll" (ByVal hHub As Long)
Dim hHub As long 'HubOpen で得たハンドル値
Call HubAllClearDp(hHub)
```

```
void HubAllStart(HANDLE hHub, long flg);
```

すべてのボードの通信を実行、停止する。

戻り値: 無し

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long flg;             //動作(1:開始 0、停止)
HubAllStart(hHub,flg);

//Basic
Declare Sub HubAllStart Lib "Plnkctl.dll" (ByVal hHub As Long, ByVal flg As Long)
Dim hHub As long 'HubOpen で得たハンドル値
Dim flg As long   '動作(1:開始 0、停止)
Call HubAllStart(hHub,flg)
```

```
long HubReadBit(HANDLE hHub, long id, long ch, long no, long bit);
```

指定されたボードの指定C Hの指定局番の指定ビットの入力状態を返す。

戻り値: 0または1 (ビットの入力状態)

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long id;              //ボードID(0~9)
long ch;              //系統Ch(A:0 ,B:1)
long no;              //局番No(0~127)
long bit;             //ビット(0~15)
long status;          //戻り値(0~1)
status = HubReadBit(hHub,id,ch,no,bit);
printf("[ID=%d,CH=%d,NO=%d,BIT=%d]の状態は[%d]です。¥n",id,ch,no,bit,status);
```

```
//Basic
Declare Function HubReadBit Lib "Pnkctl.dll" (ByVal hHub As Long,
                                              ByVal id As Long,
                                              ByVal ch As Long,
                                              ByVal no As Long,
                                              ByVal bit As Long) As Long
```

```
Dim hHub As long 'HubOpen で得たハンドル値
Dim id As long   'ボードID(0~9)
Dim ch As long   '系統Ch(A:0 ,B:1)
Dim no As long   '局番No(0~127)
Dim bit As long  'ビット(0~15)
Dim status As long '戻り値(0~1)
status = HubReadBit(id,ch,no,bit)
Text1 = status
```

```
long HubReadWord(HANDLE hHub, long id, long ch, long no);
```

指定されたボードの指定 C H の指定局番の入力状態を返す (16ビット一括)。

戻り値: 入力状態(16ビット)

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long id;              //ボードID(0~9)
long ch;              //系統Ch(A:0 ,B:1)
long no;              //局番No(0~127)
long status;          //戻り値(0~65535)
status = HubReadWord(hHub,id,ch,no);
printf("[ID=%ld,CH=%ld,NO=%ld]の状態は[%04x]です。¥n",
        id,ch,no,(unsigned short)status);

//Basic
Declare Function HubReadWord Lib "PInkctl.dll" (ByVal hHub As Long,
                                                ByVal id As Long,
                                                ByVal ch As Long,
                                                ByVal no As Long) As Long

Dim hHub As long 'HubOpen で得たハンドル値
Dim id As long   'ボードID(0~9)
Dim ch As long   '系統Ch(0,1)
Dim no As long   '局番No(0~127)
Dim status As long '戻り値(0~65535)
status = HubReadWord(hHub,id,ch,no)
Text1 = status
```

```
void HubWriteBit(HANDLE hHub, long id, long ch, long no, long bit, long val);
```

指定されたボードの指定 C H の指定局番の指定ビットの出力状態を書き込む。

戻り値: 無し

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long id;              //ボードID(0~9)
long ch;              //系統Ch(A:0 ,B:1)
long no;              //局番No(0~127)
long bit;             //ビット(0~15)
long data;            //出力する値(0~1)
HubWriteBit(hHub,id,ch,no,bit,data);

//Basic
Declare Sub HubWriteBit Lib "Pnkctl.dll" (ByVal hHub As Long,
                                           ByVal id As Long,
                                           ByVal ch As Long,
                                           ByVal no As Long,
                                           ByVal bit As Long,
                                           ByVal data As Long)

Dim hHub As long 'HubOpen で得たハンドル値
Dim id As long   'ボードID(0~9)
Dim ch As long   '系統Ch(A:0 ,B:1)
Dim no As long   '局番No(0~127)
Dim bit As long  'ビット(0~15)
Dim data As long '出力する値(0~1)
Call HubWriteBit(hHub,id,ch,no,bit,data)
```

```
void HubWriteWord(HANDLE hHub, long id, long ch, long no, long val);
```

指定されたボードの指定CHの指定局番の出力状態を書き込む(16ビット一括)。

戻り値: 無し

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long id;              //ボードID(0~9)
long ch;              //系統Ch(A:0 ,B:1)
long no;              //局番No(0~127)
long data;            //出力する値(0~65535)
HubWriteWord(Hub,id,ch,no,data);

//Basic
Declare Sub HubWriteWord Lib "Pnkctl.dll" (ByVal hHub As Long,
                                           ByVal id As Long,
                                           ByVal ch As Long,
                                           ByVal no As Long,
                                           ByVal data As Long)

Dim hHub As long 'HubOpen で得たハンドル値
Dim id As long   'ボードID(0~9)
Dim ch As long   '系統Ch(A:0 ,B:1)
Dim no As long   '局番No(0~127)
Dim data As long '出力する値(0~65535)
Call HubWriteWord(hHub,id,ch,no,data)
```

```
long HubReadlo(HANDLE hHub, long id);
```

指定されたボードの汎用入力の状態を返す(4ビット一括)。

戻り値: 読み込み情報(下位4ビット)

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long id;              //ボードID(0~9)
long status;          //戻り値(0~15)
status = HubReadlo(hHub,id);
printf("[ID=%d]の状態は[%d]です。¥n",id,status);

//Basic
Declare Function HubReadlo Lib "PInkctl.dll" (ByVal hHub As Long,
                                              ByVal id As Long) As Long

Dim hHub As long 'HubOpen で得たハンドル値
Dim id As long   'ボードID(0~9)
Dim status As long '戻り値(0~15)
status = HubReadlo(hHub,id)
Text1 = status
```

```
void HubWritelo(HANDLE hHub, long id, long val);
```

指定されたボードの汎用出力に値を書き込む(4ビット一括)。

戻り値: 無し

```
//C
HANDLE hHub;          //HubOpen で得たハンドル値
long id;              //ボードID(0~9)
long data;            //出力する値(0~15)
HubWritelo(hHub,id,data);

//Basic
Declare Sub HubWritelo Lib "PInkctl.dll" (ByVal hHub As Long,
                                           ByVal id As Long,
                                           ByVal data As Long)

Dim hHub As long 'HubOpen で得たハンドル値
Dim id As long   'ボードID(0~9)
Dim data As long '出力する値(0~15)
Call HubWritelo(hHub,id,data)
```

# ホロン株式会社

ホームページ : <http://www.holondata.co.jp>

E-Mail : [hub@mail.holondata.co.jp](mailto:hub@mail.holondata.co.jp)

本書の内容に関しては、改良のために将来予告なしに仕様などを変更することがありますので、ご了承ください。

第1版 '2001-07-12

第2版 '2001-10-03