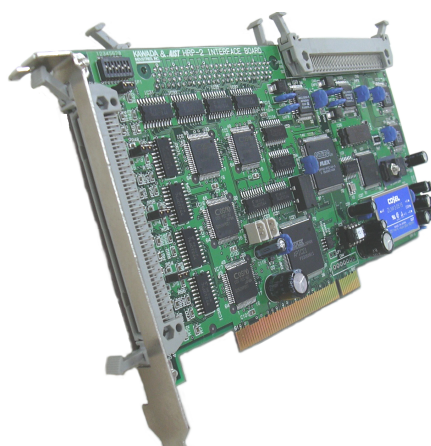


# HRP Interface Board

07-0003



リファレンスマニュアル(第1版)

2008年9月

# 目 次

<b>1. 関数 .....</b>	<b>1</b>
I7fOpen .....	1
I7fClose .....	2
I7fReset .....	2
I7fGetInfo .....	3
I7fAdRead .....	4
I7fDaWrite .....	5
I7fEncMode .....	6
I7fEncStart .....	6
I7fEncStop .....	7
I7fEncRead .....	7
I7fEncWrite .....	8
I7fEncClear .....	8
I7fPioWrite .....	9
I7fPioRead .....	9
I7fPioSet .....	10
I7fPioReset .....	10
<b>2. データ形式.....</b>	<b>11</b>
IB7_PARAINFO, PIB7_PARAINFO .....	11
IB7_PARAAD, PIB7_PARAAD .....	11
IB7_PARADA, PIB7_PARADA .....	11
IB7_PARAENC, PIB7_PARAENC .....	12
IB7_PARAPIO, PIB7_PARAPIO .....	12

**3. 設定コード.....13**

IB7_MAXADCH	13
IB7_MAXDACH	13
IB7_MAXENCCH	13
IB7_MAXPIOCH	13
ENC_MODE_PULSE	13
ENC_MODE_X1	13
ENC_MODE_X2	13
ENC_MODE_X4	13

**4. エラーコード .....13**

RTN_IBOK	13
RTN_IBOVER	13
RTN_IBNOOPEN	13
RTN_IBNCLOSE	13
RTN_IBNOPARA	13
RTN_IBCHOVER	13
RTN_ENCNOMODE	13
RTN_IBOPENED	13
RTN_IBIOCTL	13
RTN_IBTMOVER	13

**5. 改訂履歴 .....14**

## 1. 関数

各機能を使用するための関数を用意しておりますのでオリジナルのプログラム作成にご利用ください。

表1. 全関数リスト

関数	機能
I7fOpen	I/F ボードのデバイスをオープン
I7fClose	I/F ボードのデバイスをクローズ
I7fReset	D/A,PIO の出力を全チャンネル0にする。
I7fGetInfo	デバイス情報を取得
I7fAdRead	A/D のデータ読み込み
I7fDaWrite	D/A のデータ書き込み
I7fEncMode	エンコーダカウンタのモード設定
I7fEncStart	カウントスタート
I7fEncStop	カウントストップ
I7fEncRead	カウントデータの読み込み
I7fEncWrite	カウントデータの書き込み
I7fEncClear	カウントデータを0にする
I7fPioRead	全チャンネルの PIO データの読み込み
I7fPioWrite	全チャンネルの PIO データの書き込み
I7fPioSet	指定したチャンネルの出力を1にする。
I7fPioReset	指定したチャンネルの出力を0にする。

### I7fOpen

指定した I/F ボードのアクセスするドライバを開始します。オープンできる番号は 0~3 までです。

**書式** int I7fOpen( int Ib7No )

**引数** Ib7No...デバイス番号(0~3まで指定可能)

**戻り値** RTN\_IBOVER

0 ≤ 値 → 正常(オープン#)

0 > 値 → 異常

**<例>** 0番のデバイスドライバをオープン  
 if( I7fOpen( 0 ) < 0 ) printf(“open error”);

---

**I7fClose**

---

指定した I/F ボードのアクセスするドライバを終了します。クローズできる番号は 0～3 までです。

---

**書式** int I7fClose( int Ib7No )

**引数** Ib7No・・・デバイス番号(0～3まで指定可能)

**戻り値** RTN\_IBOVER  
RTN\_IBNOCLOSE  
RTN\_IBOK → 正常  
0 > 値 → 異常

**<例>** 0番のデバイスをオープン  
if( I7fClose( 0 ) < 0 ) printf(“close error”);

---

---

**I7fReset**

---

指定した I/F ボードを出力機能(D/A、PO)を全チャンネル初期状態に戻します。

(初期状態は D/A の出力が 0[V]、PO の出力が High(5[V])の状態です。)

リセットできる番号は 0～3 までです。

---

**書式** int I7fReset( int Ib7No )

**引数** Ib7No・・・デバイス番号(0～3まで指定可能)

**戻り値** RTN\_IBOVER  
0 ≤ 値 → 正常  
0 > 値 → 異常

**<例>** 0番のデバイスをリセット  
if( I7fReset( 0 ) < 0 ) printf(“reset error”);

---

---

**I7fGetInfo**

指定した I/F ボードの情報を取得します。取得した情報は引数の中に格納されます。  
複数枚の I/F ボードを使用する際に引数の中の Ib7ID を参照してボードを特定します。

---

**書式** int I7fGetInfo(PIB7\_PARAINFO)

**引数** PIB7\_PARAINFO(IB7\_PARAINFO のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOPARA  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン #)

**<例>** 0番のデバイスドライバから I/F ボードの情報を取得

```
IB7_PARAINFO Ib7ParaInfo;
Ib7ParaInfo.Ib7no=0;
if( I7fGetInfo( &Ib7ParaInfo ) < 0 ) printf("Device Info get error");
else{
    printf("Get Device Infomation¥nib7no=%i, ioBase=%08X.¥n",
        Ib7ParaInfo.DevInfo.ib7no,
        Ib7ParaInfo.DevInfo.ioBase);
    printf("IB7 version = %04X, IB7 ID=%i.¥n¥n",
        Ib7ParaInfo.DevInfo.Ib7Ver,
        Ib7ParaInfo.DevInfo.Ib7ID);
}
```

---

---

**I7fAdRead**

指定したチャンネルに入力された電圧値を 12bit の A/D データにして読み込みます。

データ[data]を電圧値[volt]に変えるには以下の式を使用してください。

±10V入力レンジ :  $\text{volt} = 20.0 * (\text{double})\text{data} / 4096.0 - 10.0$

0~5V入力レンジ :  $\text{volt} = 5.0 * (\text{double})\text{data} / 4096.0$

---

**書式** int I7fAdRead( PIB7\_PARAAD )

**引数** PIB7\_PARAAD (IB7\_PARAAD のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOOPEN  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン#)

**<例>** 0番のデバイスドライバから A/D 1ch のデータを取得

```
WORD AdDat;  
IB7_PARAAD Ib7ParaAd;  
Ib7ParaAd.Ib7no=0;  
Ib7ParaAd.ChData.ch=0;  
if( I7fAdRead( &Ib7ParaAd ) < 0 ) printf("A/D Read error");  
AdDat=Ib7ParaAd.ChData.wData[0];
```

---

---

**I7fDaWrite**

---

指定したチャンネルに 12bit の D/A データを書き込み、電圧を出力します。

電圧値[volt]をデータ[data]に変えるには以下の式を使用してください。

$$\text{data}=(\text{WORD})(\text{volt}*4096.0/20.0+2047.5)$$

---

**書式** int I7fDaWrite( PIB\_PARADA )

**引数** PIB7\_PARADA (IB7\_PARADA のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOOPEN  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン#)

**<例>** 0番のデバイスドライバから D/A 1ch のデータ出力  
WORD DaDat=2048;  
IB7\_PARADA Ib7ParaDa;  
Ib7ParaDa.Ib7no=0;  
Ib7ParaDa.ChData.ch=0;  
Ib7ParaDa.ChData.wData[0]=DaDat;  
if( I7fDaWrite( &Ib7ParaDa ) < 0 ) printf(“D/A Read error”);

---



---

### I7fEncMode

---

指定したチャンネルのカウンタのカウントモードを設定します。

カウントモードは全部で4種類用意されています。

ENC\_MODE\_PULSE : up/down カウント  
ENC\_MODE\_X1 : 1逡倍  
ENC\_MODE\_X2 : 2逡倍  
ENC\_MODE\_X4 : 4逡倍

---

**書式** int I7fEncMode(PIB7\_PARAENC)

**引数** PIB7\_PARAENC( IB7\_PARAENC のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOOPEN  
RTN\_IBCHOVER  
RTN\_ENCNOMODE  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン#)

**<例>** 0番のデバイスドライバから ENC 1ch のモードを設定。  
IB7\_PARAENC Ib7ParaEnc;  
Ib7ParaEnc.Ib7no=0;  
Ib7ParaEnc.ChData.ch=0;  
Ib7ParaEnc.ChData.mode=ENC\_MODE\_PULSE;  
if( I7fEncMode( &Ib7ParaEnc ) < 0 ) printf(“ENC Mode error”);

---

### I7fEncStart

---

指定したチャンネルのカウントをスタートさせます。

---

**書式** int I7fEncStart ( PIB7\_PARAENC )

**引数** PIB7\_PARAENC( IB7\_PARAENC のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOOPEN  
RTN\_IBCHOVER  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン#)

**<例>** 0番のデバイスドライバから ENC 1ch のカウントをスタートさせます。  
IB7\_PARAENC Ib7ParaEnc;  
Ib7ParaEnc.Ib7no=0;  
Ib7ParaEnc.ChData.ch=0;  
if( I7fEncStart( &Ib7ParaEnc ) < 0 ) printf(“ENC Start error”);

---

---

**I7fEncStop**

---

指定したチャンネルのカウントをストップさせます。

---

**書式** int I7fEncStop ( PIB7\_PARAENC )

**引数** PIB7\_PARAENC( IB7\_PARAENC のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOPEN  
RTN\_IBCHOVER  
0 ≤ 値 → 異常  
0 > 値 → 正常 ( オープン # )

**<例>** 0番のデバイスドライバから ENC 1ch のカウントをストップさせます。  
IB7\_PARAENC Ib7ParaEnc;  
Ib7ParaEnc.Ib7no=0;  
Ib7ParaEnc.ChData.ch=0;  
if( I7fEncStop( &Ib7ParaEnc ) < 0 ) printf( "ENC Stop error" );

---

---

**I7fEncRead**

---

指定したチャンネルからカウントデータを読み込みます。

---

**書式** int I7fEncRead( PIB\_PARAENC )

**引数** PIB7\_PARAENC( IB7\_PARAENC のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOPEN  
RTN\_IBCHOVER  
0 ≤ 値 → 異常  
0 > 値 → 正常 ( オープン # )

**<例>** 0番のデバイスドライバから ENC 1ch のカウント値を CntDat に移します。  
DWORD CntDat;  
IB7\_PARAENC Ib7ParaEnc;  
Ib7ParaEnc.Ib7no=0;  
Ib7ParaEnc.ChData.ch=0;  
if( I7fEncRead( &Ib7ParaEnc ) < 0 ) printf( "ENC Read error" );  
CntDat=Ib7ParaEnc.ChData.dwData[0];

---

---

**I7fEncWrite**

---

指定したチャンネルにカウントデータを書き込みます。

---

**書式** int I7fEncWrite( PIB\_PARAENC )

**引数** PIB7\_PARAENC( IB7\_PARAENC のポインタ )

**戻り値** RTN\_IBoVER  
RTN\_IBNoPEN  
RTN\_IBChOVer  
0 ≤ 値 → 異常  
0 > 値 → 正常 ( オープン # )

**<例>** 0番のデバイスドライバの ENC 1ch に 1000 を書き込む。  
DWORD CntDat=1000;  
IB7\_PARAENC Ib7ParaEnc;  
Ib7ParaEnc.Ib7no=0;  
Ib7ParaEnc.ChData.ch=0;  
Ib7ParaEnc.ChData.dwData[0]=CntDat;  
if ( I7fEncWrite( &Ib7ParaEnc ) < 0 ) printf( "ENC Write error" );

---

**I7fEncClear**

---

指定したチャンネルのカウントデータを0にします。

---

**書式** int I7fEncClear( PIB\_PARAENC )

**引数** PIB7\_PARAENC( IB7\_PARAENC のポインタ )

**戻り値** RTN\_IBoVER  
RTN\_IBNoPEN  
RTN\_IBChOVer  
0 ≤ 値 → 異常  
0 > 値 → 正常 ( オープン # )

**<例>** 0番のデバイスドライバの ENC 1ch のカウントデータを0にする  
IB7\_PARAENC Ib7ParaEnc;  
Ib7ParaEnc.Ib7no=0;  
Ib7ParaEnc.ChData.ch=0;  
if ( I7fEncClear( &Ib7ParaEnc ) < 0 ) printf( "ENC Clear error" );

---

---

### I7fPioWrite

---

指定したグループから 16bit のデータを書き込み、各PIOのチャンネルから出力させます。

グループ1は1～16チャンネルになります。

---

**書式** int I7fPioWrite(PIB7\_PARAPIO)

**引数** PIB7\_PARAPIO(PIB7\_PARAPIO のポインタ)

**戻り値** RTN\_IBOVER  
 RTN\_IBNOPEN  
 RTN\_IBCHOVER  
 0 ≤ 値 → 異常  
 0 > 値 → 正常(オープン#)

**<例>** 0番のデバイスドライバの OUT Group1 から 16bit のデータを出力する  
 WORD PioDat=0xAAAA;  
 IB7\_PARAPIO Ib7ParaPio;  
 Ib7ParaPio.Ib7no=0;  
 Ib7ParaPio.ChData.OutGroup=0;  
 Ib7ParaPio.ChData.wData= PioDat;  
 if( I7fPioWrite( &Ib7ParaPio ) < 0 ) printf("PIO Write error");

---

### I7fPioRead

---

指定したグループからPIOの各チャンネルに入力された 16bit のデータを読み込みます。

グループ1は1～16チャンネルになります。

---

**書式** int I7fPioRead(PIB7\_PARAPIO)

**引数** PIB7\_PARAPIO(PIB7\_PARAPIO のポインタ)

**戻り値** RTN\_IBOVER  
 RTN\_IBNOPEN  
 RTN\_IBCHOVER  
 0 ≤ 値 → 異常  
 0 > 値 → 正常(オープン#)

**<例>** 0番のデバイスドライバの IN Group1 から 16bit のデータを読み込む。  
 WORD PioDat;  
 IB7\_PARAPIO Ib7ParaPio;  
 Ib7ParaPio.Ib7no=0;  
 Ib7ParaPio.ChData.InGroup=0;  
 if( I7fPioRead( &Ib7ParaPio ) < 0 ) printf("PIO Read error");  
 PioDat=Ib7ParaPio.ChData.wData;

---

---

**I7fPioSet**

---

指定したチャンネルの出力を High にする。

OutGroup に必ず0を指定してください。

---

**書式** int I7fPioSet(PIB7\_PARAPIO)

**引数** PIB7\_PARAPIO(PIB7\_PARAPIO のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOPEN  
RTN\_IBCHOVER  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン#)

**<例>** 0番のデバイスドライバの PIO 1ch を High にする。  
IB7\_PARAPIO Ib7ParaPio;  
Ib7ParaPio.Ib7no=0;  
Ib7ParaPio.ChData.OutGroup=0;  
Ib7ParaPio.ChData.ch=0;  
if( I7fPioSet( &Ib7ParaPio ) < 0 ) printf("PIO Set error");

---

**I7fPioReset**

---

指定したチャンネルの出力を Low にする。

OutGroup に必ず0を指定してください。

---

**書式** int I7fPioReset(PIB7\_PARAPIO)

**引数** PIB7\_PARAPIO(PIB7\_PARAPIO のポインタ)

**戻り値** RTN\_IBOVER  
RTN\_IBNOPEN  
RTN\_IBCHOVER  
0 ≤ 値 → 異常  
0 > 値 → 正常 (オープン#)

**<例>** 0番のデバイスドライバの PIO 1ch を Low にする。  
IB7\_PARAPIO Ib7ParaPio;  
Ib7ParaPio.Ib7no=0;  
Ib7ParaPio.ChData.OutGroup=0;  
Ib7ParaPio.ChData.ch=0;  
if( I7fPioReset( &Ib7ParaPio ) < 0 ) printf("PIO Reset error");

---

## 2. データ形式

表2. 各機能のデータ形式

機能名	型名	データ形式
デバイス 情報	<b>IB7_PARAINFO, PIB7_PARAINFO</b>	<pre>typedef struct _IB7_DEVINFO {     WORD ib7no;           //デバイス番号0~3     WORD irq;             //IRQ No.     WORD Ib7Ver;         //PCI Version     WORD Ib7ID;          //シリアルNoの下4桁     DWORD ioBase;       //I/O Base Address } IB7_DEV_INFO, *PIB7_DEV_INFO;  typedef struct _IB7_PARAINFO {     WORD Ib7no;           //デバイス番号0~3     IB7_DEV_INFO DevInfo; //デバイスパラメータ } IB7_PARAINFO, *PIB7_PARAINFO;</pre>
A/D	<b>IB7_PARAAD, PIB7_PARAAD</b>	<pre>typedef struct _IB7_ADDACH {     int ch;               //チャンネル0~15     WORD wData[2];       //[0]がA/Dデータ } IB7_ADDACH, *PIB7_DADDACH;  typedef struct _IB7_PARAAD {     WORD Ib7no;           //デバイス番号0~3     IB7_ADDACH ChData;   //チャンネルデータ } IB7_PARAAD, *PIB7_PARAAD;</pre>
D/A	<b>IB7_PARADA, PIB7_PARADA</b>	<pre>typedef struct _IB7_ADDACH {     int ch;               //チャンネル0~15     WORD wData[2];       //[0]がD/Aデータ } IB7_ADDACH, *PIB7_DADDACH;  typedef struct _IB7_PARADA {     WORD Ib7no;           //デバイス番号0~3     IB7_ADDACH ChData;   //チャンネルデータ } IB7_PARADA, *PIB7_PARADA;</pre>

機能名	型名	データ形式
ENC	<b>IB7_PARAENC, PIB7_PARAENC</b>	<pre>typedef struct _IB7_ENCCH {     int ch; //チャンネル 0~15     WORD command; //コマンド     WORD mode; //カウントモード     WORD status; //ステータス     DWORD dwData[2]; //[0]がカウントデータ } IB7_ENCCH, *PIB7_ENCCH;  typedef struct _IB7_PARAENC {     WORD Ib7no; //デバイス番号 0~3     IB7_ENCCH ChData; //チャンネルデータ } IB7_PARAENC, *PIB7_PARAENC;</pre>
PIO	<b>IB7_PARAPIO, PIB7_PARAPIO</b>	<pre>typedef struct _IB7_PIOCH {     int InGroup; //IN グループ番号 0~3     int OutGroup; //OUT グループ番号 0~3     WORD ch; //チャンネル 0~15     WORD wData; //PIO データ } IB7_PIOCH, *PIB7_PIOCH;  typedef struct _IB7_PARAPIO {     WORD Ib7no; //デバイス番号 0~3     IB7_PIOCH ChData; //チャンネルデータ } IB7_PARAPIO, *PIB7_PARAPIO;</pre>

### 3. 設定コード

設定コード	値	内容
IB7_MAXADCH	16	A/D の最大チャンネル数
IB7_MAXDACH	16	D/A //
IB7_MAXENCCH	16	エンコーダカウンタ //
IB7_MAXPIOCH	16	PI と PO の最大チャンネル数(もしくは bit 数)
ENC_MODE_PULSE	16	エンコーダカウンタ UP/DOWN モード
ENC_MODE_X1	1	// 1 通倍カウントモード
ENC_MODE_X2	2	// 2 通倍カウントモード
ENC_MODE_X4	3	// 4 通倍カウントモード

### 4. エラーコード

各関数が返すエラーコードを説明します。

表3. エラーコード表

エラーコード	値	内容
RTN_IBOK	0	正常
RTN_IBOVER	-1	I/F ボードが無い(0~3以外)
RTN_IBNOPEN	-2	I/F ボードがオープンされていない
RTN_IBNCLOSE	-3	I/F ボードがクローズされない
RTN_IBNOPARA	-4	パラメータが指定されていない
RTN_IBCHOVER	-5	I/F ボードのチャンネルが間違っている
RTN_ENCNOMODE	-6	ENC モードが指定以外である
RTN_IBOPENED	-7	I/F ボードはすでにオープンされている。
RTN_IBIOCTL	-8	I/O コントロールエラー
RTN_IBTMOVER	-9	内部処理の時間制限を越えました。



## 5. 改訂履歴

版	日付	ページ	内容
1	2008/9/18	—	初版作成(ソフトウェアマニュアルから分離。) ○ソフトウェアマニュアルからの修正・変更点 設定コードの項目を追加 エラーコードの追加 I7fReset 関数の PO を Low から High に変更。

開発元

**川田工業株式会社**

製造・販売元／修理、技術的なお問い合わせは

**有限会社 図工**

TEL:0463-97-4891 / FAX:0463-97-4895

URL:<http://www.zuco.jp> / E-mail:[support@zuco.jp](mailto:support@zuco.jp)