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

1. 関数.....1

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

2. データ形式.....11

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_IBNOPEN _____ 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 ： volt=20.0*(double)data/4096.0-10.0

0～5V入力レンジ ： volt=5.0*(double)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_IBNOPEN
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。

data=(WORD)(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_IBNOPEN
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逓倍
ENC_MODE_X2： 2逓倍
ENC_MODE_X4： 4逓倍

書式　int I7EncMode(PIB7_PARAENC)

引数　PIB7_PARAENC(IB7_PARAENC のポインタ)

戻り値　RTN_IBOVER
RTN_IBNOPEN
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_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 I7fEncCear(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 I7PioWrite(PIB7_PARAPIO)

引数 PIB7_PARAPIO(IB7_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 I7PioRead(PIB7_PARAPIO)

引数 PIB7_PARAPIO(IB7_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 I7PioSet(PIB7_PARAPIO)

引数　PIB7_PARAPIO(IB7_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 I7PioReset(PIB7_PARAPIO)

引数　PIB7_PARAPIO(IB7_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	**IB7_PARAINFO, PIB7_PARAINFO**	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 Aaddress
		} 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;

A/D	**IB7_PARAAD, PIB7_PARAAD**	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;

D/A	**IB7_PARADA, PIB7_PARADA**	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;

機能名	型名	データ形式

ENC	**IB7_PARAENC,**
	PIB7_PARAENC
		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;

PIO	**IB7_PARAPIO,**
	PIB7_PARAPIO
		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;

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 逓倍カウントモード
ENC_MODE_X2	2	〃2逓倍カウントモード
ENC_MODE_X4	3	〃4逓倍カウントモード

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	ー	初版作成(ソフトウェアマニュアルから分離。) 〇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 P0 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