

HLSB-36PC I Rev 1.00

ユーザーズマニュアル

2004, 6, 1

はじめに

この度は、弊社 HLSB-36PCI をお買い上げ頂き誠にありがとうございます。本製品の御利用にあたり本ユーザーズマニュアルを良くお読みいただき、十分な御理解の上、完全に使いこなして頂ければ弊社メーカーといたしましても大変喜ばしい限りでございます。

尚、本ユーザーズマニュアルの PDF 書類及び弊社製品の最新情報がインターネットのホームページから入手可能です。是非、常に最新の情報をご確認ください。

HomePage URL <http://www.steptecnica.com/>

【ご注意】

- 本製品の使用、保管の際は静電気および衝撃などに十分に注意してお取り扱い願います。
- パソコンの電源（POWER）スイッチを“ON”にしたままで本製品およびケーブルなどの脱着を行わないでください。
- 本製品本体の拡張バスコネクタの金メッキ部分には絶対に手を触れないでください。動作不良の原因になります。
- 本製品上の MKY36 に関する詳細につきましては、「MKY36 ユーザーズマニュアル」（ステップテクニカ）などをご参照ください。
- 本製品の改造およびその使用にともなう弊害につきましては、当社は一切責任を負いかねますのでご了承願います。
- 本マニュアルの内容の一部または全部を当社に無断で複製・再配布することを禁じます。
- 本マニュアルに掲載される製品名は、一般に開発メーカーの商標または登録商標です。

株式会社ステップテクニカ

1 HLSB-36PCI のハードウェア

1.1 特徴

HLSB-36PCI は、HLS (Hi-speed Link System) のセンター IC である MKY36 (ステップテクニカ) を 2 チャンネル搭載した PCI バス拡張ボードです。

4KB のメモリエリアと、1 本の割り込みリソースを使用し、ホストマシンのメモリマップ上に MKY36 チップの HLS メモリを構成します。

HLS の評価が容易に行えるよう、MKY36 でサポートする 3/6/12Mbps でのフルデュプレックス / ハーフデュプレックス通信に対応し、各種の HLS 端末装置との親和性を高めています。

本製品は、PCI Ver. 2.1 に準拠した 5V 系 PCI カードです。電源に +5V と +3.3V を使用します。+3.3V に対応していない PCI コネクタには使用することができません。使用前に確認下さい。

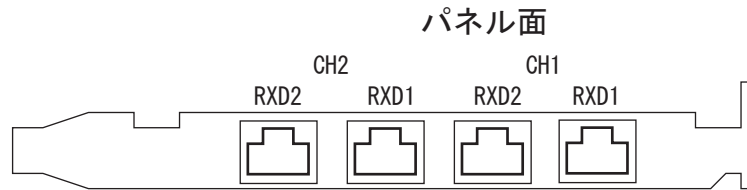
1.2 仕様

HLSB-36PCI の各種仕様を表 . 1-1 に示します。

表 . 1-1 HLSB-36PCI 仕様

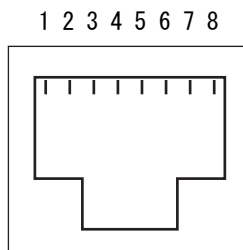
名 称	HLSB-36PCI
搭載チップ	MKY36 × 2 基
通信方式	HLS通信方式 Full/Half
転送レート	3M/6M/12M bps
対応バス	PCI Ver2.1に準拠した、32ビット・33MHz拡張バス
占有リソース	4KBの連続したメモリエリア (PnPにて自動割当)
割り込み	1ライン使用 (PnPにて自動割当)
コネクタ	RJ-45 × 4
電 源	DC +5V +3.3V
消費電流	500mA以下
使用条件	温度0～50°C 湿度20～90% (非結露)
サイズ	122mm (W) × 107mm (H) ※パネル部含まず
付属ソフト	Windows98SE/Me/2000/XP用ドライバ・ライブラリ

1.3 インターフェイス



CH1	MKY36 chanel-1のインターフェイス
CH2	MKY36 chanel-2のインターフェイス
FG	コネクタフレーム、8番端子は共通。 必ずアースへ接続して下さい。

パネル面から見たピン配置（上下の向きに注意）

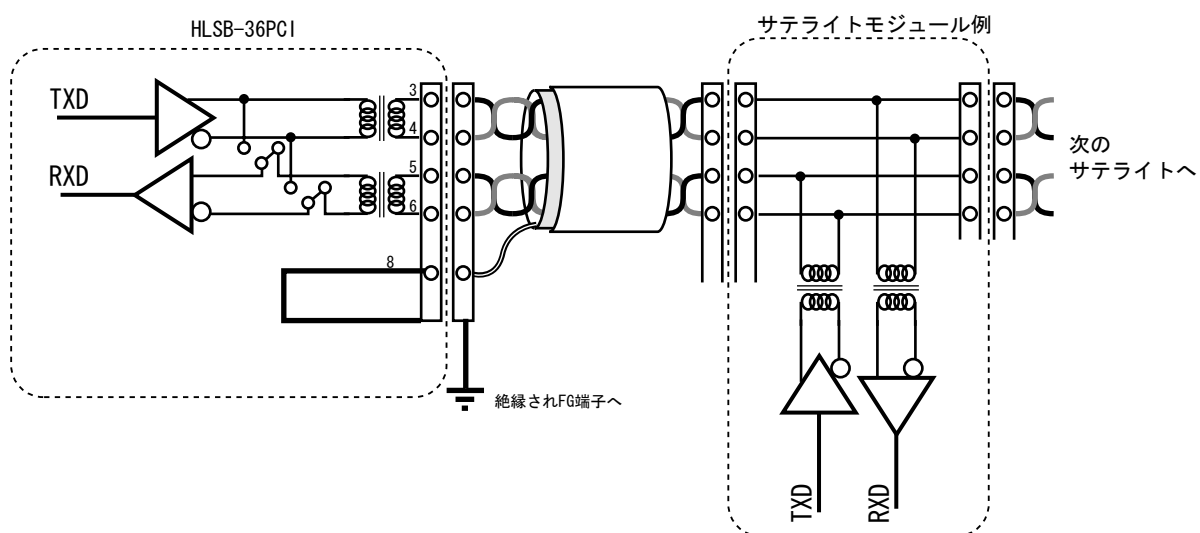


Pin	名称	備 考
1		
2		
3	CT1+	センター送信 +
4	CT1-	センター送信 -
5	ST1+	サテライト送信 +
6	ST1-	サテライト送信 -
7		
8	F. G.	伝送ケーブルのシールド線を接続 します。コネクタフレームに接続 されています。

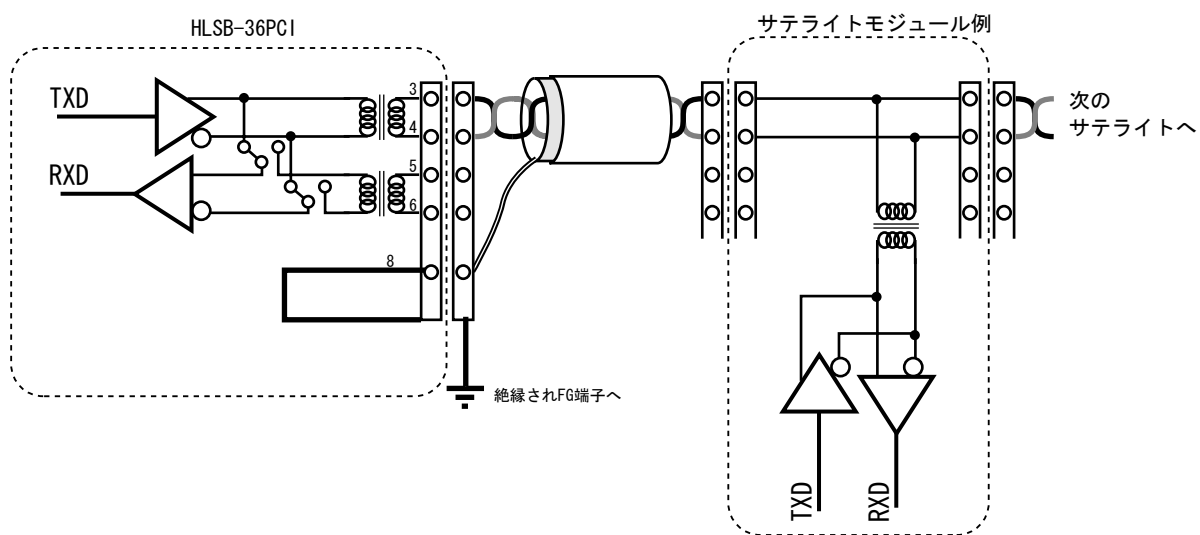
HLSB-36PCI ユーザーズマニュアル

1.4 サテライトとの接続モデル

フルデュプレックス

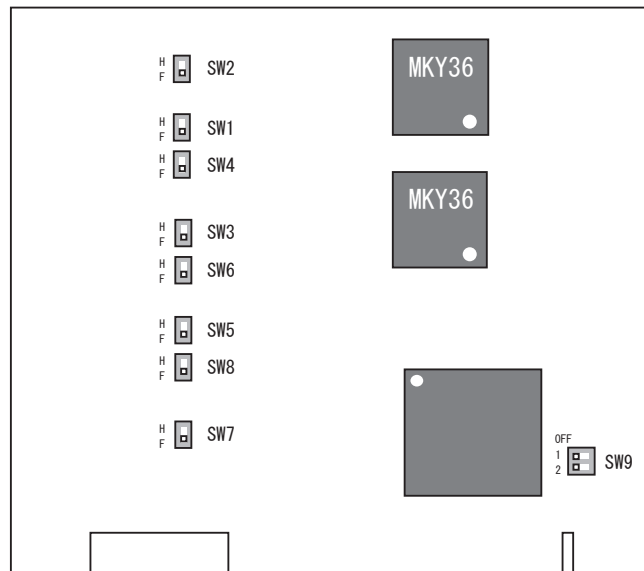


ハーフデュプレックス



1.5 SW 設定

HLSB-36PCI デザイン（出荷時設定）



上図のとおり、HLSB-36PCI には SW1 ～ SW8 までの回線切り替えスイッチと、SW9 のボード ID 設定スイッチがあります。

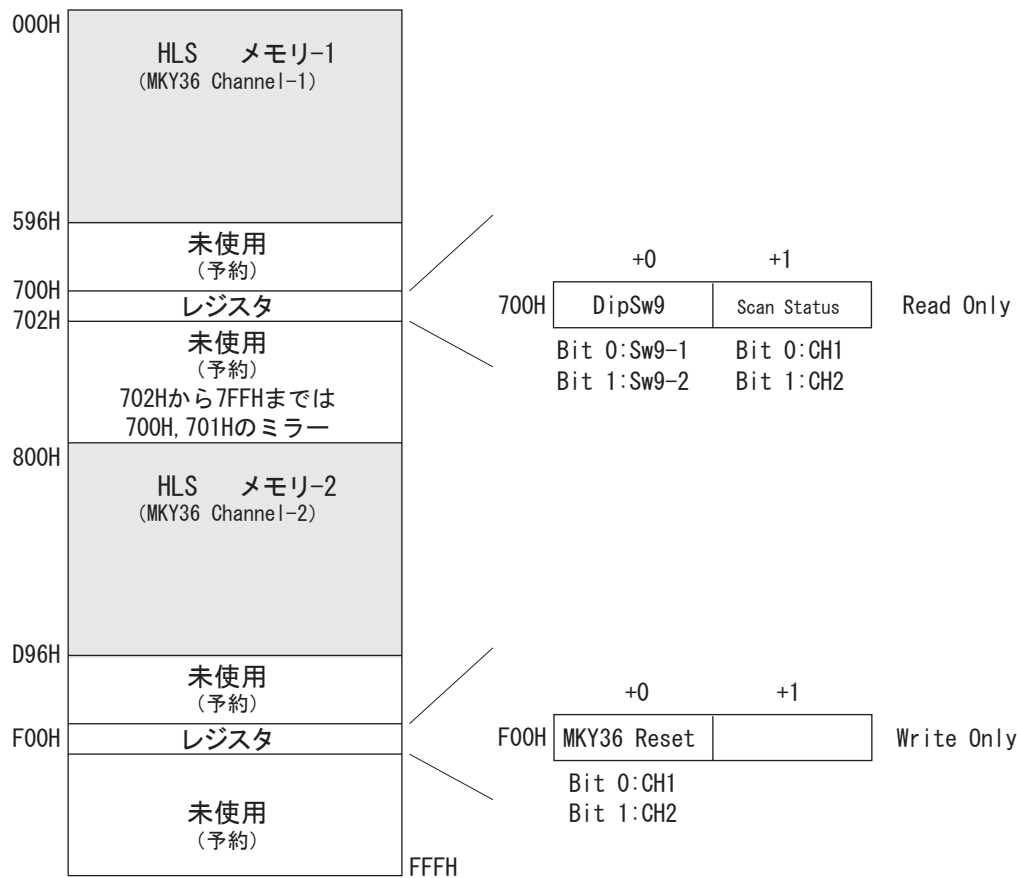
SW1 ～ SW4 がチャンネル 1、SW5 ～ SW8 がチャンネル 2 の MKY36 に対応した通信回線のフルデュプレックス / ハーフデュプレックス設定ですので、各 MKY36 の運用設定と食い違わないよう、フルデュプレックス時は“F”側に、ハーフデュプレックス時は“H”側にスイッチを合わせて下さい。MKY36 チップへのデュプレックス設定は、ソフトウェアで行います。詳しくは、MKY36 ユーザーズマニュアル BCR レジスタの詳細、または基本 API HlssetStatus() をご覧下さい。

SW9 は 2 ビットのボード ID 設定スイッチです。このスイッチの各ビットは、システムレジスタ内に反映されるので、HLSB-36PCI の 4 枚までの同時使用において各ボードに個別の ID を設定することで、ソフトウェアから各ボードを特定することができます。ボード ID を数値表現する場合、ビット 2 が上位ビット、ビット 1 が下位ビットとして表現します。

1 枚のみの使用ではこのスイッチに意味はありませんので初期値 (“OFF”) で構いません。

HLSB-36PCI ユーザーズマニュアル

1.6 メモリマップ



上図は、HLSB-36PCI のメモリマップです。実際のアドレスは、上記の数値にボードの先頭アドレスが加算されたものになることに注意してください。

HLSB-36PCI はワードリード / ワードライトに対応しています。バイトリード、ダブルワードリード、ダブルワードライトは、付属の DLL にてワードリード / ワードライトを使用して可能となっております。HLS メモリエリアは、MKY36 のメモリエリアをそのままホストマシンのメモリ上に展開したものですので、このエリアに関する詳細は、「MKY36 ユーザーズマニュアル」を参照してください。

未使用エリアは、リードでは全て 00H が読め、ライトは無視されますが、将来の予約のために読み込みは不定値を前提とし、書き込みは行わないで下さい。

1.7 レジスタ

HLSB-36PCI は、計 2 個のレジスタを持っています。ボードの ID の取得や MKY36 のリセットなどは、このレジスタを通してソフトウェアで行なえます。各 MKY36 のフル / ハーフ設定、通信レート設定は、MKY36 のレジスタに直接設定するか、基本 API HlsSetStatus() 関数を利用して下さい。尚、解説中のチップリセットとは、各チップのハードリセット端子を直接駆動してチップ単体にかけるリセットを指し、ボードリセットとは、PCI バスのリセット信号によるボード全体のリセットを指します。各レジスタの “Initial” の値はボードリセット直後の値です。また、ボードリセット中は、MKY36 のチップリセットもかかります。

システムレジスタ 1 (700H)

Bit:	7	6	5	4	3	2	1	0
	-	-	RUN2	RUN1	-	-	ID2	ID1
Initial:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

ID2, ID1

SW9 で設定したボード ID が反映されます。

RUN2

チャンネル 2 が通信を行っている状態のとき、“1” になります。

チャンネル 2 の通信が停止時は “0” になります。

RUN1

チャンネル 1 が通信を行っている状態のとき、“1” になります。

チャンネル 1 の通信が停止時は “0” になります。

システムレジスタ 2 (F00H)

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RST2	RST1
Initial:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	W	W

RST2

当ビットへの “1” 書き込みにより、チャンネル 2 の MKY36 にチップリセットがかかります。

RST1

当ビットへの “1” 書き込みにより、チャンネル 1 の MKY36 にチップリセットがかかります。

2. 付属ソフトウェアについて

2.1 概要

本製品には、お客様による HLS の評価・学習を一層行いやすい物にするために、Microsoft Windows 用の WDM ドライバ、及び HLSB 標準ライブラリ WDM 版が付属しています。現在、動作確認が取れている OS は、Windows98SE, Windows Me, Windows 2000 Pro, WindowsXP です。

初めて HLSB-36PCI を搭載して PC を立ち上げる時、Windows のハードウェア追加ウィザードに従い、付属のフロッピーディスクより WDM ドライバを PC にインストールしてください。尚、Windows 2000, WindowsXP では、デバイスドライバのインストールにはアドミニストレータ権限が必要になりますので、ログインの際にご注意ください。

付属フロッピーディスクの“¥D11”フォルダには、HLSB 標準ライブラリ WDM 版が収められています。HLSB 標準ライブラリを使用することで、Microsoft Visual Basic をはじめ、各種のプログラミングツールを使って容易に HLS をコントロールすることが可能です。

また、従来より HLSB-PCI で HLSB 標準ライブラリを利用していたユーザソフトウェアは、ほとんどの場合、わずかな修正と再コンパイル程度の手間でそのまま利用することが可能です。

今後に対応 OS の拡大や、アップバージョンのリリース等も予定しておりますので、随時、弊社 web ページ (<http://www.steptecnica.com/>) にてご確認ください。

2.2 著作権・免責

本製品付属フロッピーディスクに収められた、全てのドキュメント・プログラム・プログラムソースの著作権は、株式会社ステップテクニカが所有しています。株式会社ステップテクニカは、以下の注意事項を了承された個人・法人、または、その他の団体が弊社製品 HLSB-36PCI を利用する場合に限り、これら著作物の複製・利用をする権利をライセンスするものであり、株式会社ステップテクニカに無断でこれら著作物の一部または全部を改訂・再配布したり、上記以外の目的のために複製・利用することはできません。

【注意事項】

- 本製品付属フロッピーディスク内のソフトウェア及び、弊社 web ページより入手した全てのソフトウェアの使用による、いかなる結果に対しても弊社は一切責任を負いません。
- ライブラリの説明は、よく読み、正しくお使いください。
- 仕様・内容は、将来予告無く変更になる場合があります。弊社は、将来への互換性について、一切保証いたしません。
- 弊社製品以外の OS や開発環境等に関するお問い合わせはサポートいたしかねます。
- バグ・不具合などを発見された方は、弊社技術部までご連絡ください。

技術部 e-mail: info@steptecnica.com

2.3 ファイルの種類

付属フロッピーディスクの“¥D11”フォルダに収められたファイルは以下のとおりです。

Hlsb36Pci.dll

DLL 本体です。Windows のシステムフォルダか、本 DLL を使用するユーザプログラムと同じディレクトリにコピーしてお使いください。

Hlsb36Pci.lib

Microsoft Visual C++ 用のインポートライブラリです。
同バージョン 5.0 で作成した物です。

Hlsb36Pci.h

DLL のヘッダファイルです。ご使用の際は、Windows.h より後ろにインクルードしてください。

2.4 現バージョンのドライバによる制限

現バージョンの Windows ドライバは、割り込みをサポートしておりません。そのため、割り込みを発生させる MKY36 への設定は、反映はされますが割り込み発生ステータスは、すべてドライバーで破棄されます。DLL 内の割り込み関係の関数は、実際の処理を行わずに成功時の戻り値を戻すことに注意してください。

2.5 HLSB-ISA 及び HLSB-PCI DLL との互換性

基本的に全く異なるハードウェアを持ちますが、ライブラリレベルではかなり高いレベルで上位互換性を保っています。HLSB-PCI DLL 利用していたソフトウェアの移植をされる場合、以下の点にご注意ください。

1. ファイル名

DLL 本体のファイル名が違います。ソースの修正や再コンパイルが必要です。

2. 同時利用可能枚数

互換性のある基本 API を利用される場合、4 枚までの HLSB-36PCI が同時利用可能です。4 枚を越えた同時利用をされる場合、ダイレクト API を利用しなくてはなりません。

3. 割り込み

HLSB-36PCI では、割り込みは MKY36 のレジスタの設定が、即、割り込みの発生に繋がります。

現バージョンでは割り込みに対応しておりません。HlsSetIM() 等の関数には、互換性の為存在しますが実際の処理は行いません。戻り値は失敗を示しません。

4. オーバーヘッド

互換性を保つために、新しいダイレクト API に比べ、僅かのオーバーヘッドがあります。

3 共通 API

3.1 概要

共通 API は、HLSB-ISA DLL と互換性を持ち、ダイレクト API 利用時にも呼び出し可能な API です。これらは、HlsOpen() の前や、HlsClose() の後に呼び出すことも可能です。

3.2 仕様

HlsGetLastError

書式

```
UINT HLSBAPI HlsGetLastError( void )
```

説明

プロセスが最後にコールした HLSB API のエラーコードが得られます。

戻り値

HlsbPci.h で以下のように定義されています。

文字定数	値	
HLS_SUCCESS	0	/* 正常終了 */
HLS_ERR_ALREADYOPENED	1	/* すでにオープンされている */
HLS_ERR_CLOSED	2	/* HlsOpen() が一度もコールされていない */
HLS_ERR_INVALIDPARAM	3	/* 無効なパラメータでコールされた */
HLS_ERR_FAILED	4	/* 原因不明により処理が遂行されなかった */
HLS_ERR_NORESOUCE	5	/* 実行に必要なリソースが足りない */
HLS_ERR_DEVICENOTEXIST	7	/* デバイスが存在しない */
HLS_NOTCALLYET	99	/* まだ1度も HLSBAPI がコールされていない */

HlsGetVersion

書式

```
UINT HLSBAPI HlsGetVersion( void )
```

説明

HLSB API のバージョンが取得できます。

戻り値

戻り値が 0x210 の場合、バージョン 2、リビジョン 1、リリース 0 を表わします。プログラム作成時に想定したバージョンと異なるバージョン番号が得られた場合、互換性が保証されませんので、プログラムの先頭でバージョンチェックすることをおすすめします。リビジョン番号・リリース番号の変更では、互換性は保たれます。

4 基本 API

4.1 概要

基本 API は、HLSB-ISA DLL Ver2.00 と互換性のある API です。また、動作の意味ごとに各 API が用意されているため、ダイレクト API に比べ、動作を理解しやすいソースを記述することができます。

ほとんどの利用形態において、利用しやすく、ご推奨する API です。

注意：基本 API とダイレクト API の同時利用はできません。

4.2 ハードウェアの設定

基本 API で利用する HLSB-36PCI は、1～4 枚までに 0 からの昇順のボード ID をディップスイッチ (SW9) で設定しておかなくてはなりません。

HlsOpen() は、必ずボード ID.0 のボードを割り当て、HlsAddBoard() は昇順で次のボード ID のボードから追加していきます。ボード ID に空きや重複があると正しく動作しません。

4.3 仕様

HlsOpen

書式

```
BOOL HLSBAPI HlsOpen( WORD dmy )
```

説明

ボード ID.0 の HLSB-36PCI 上の MKY36 を基本 API 用にチャンネル 1,2 として初期化します。

ダイレクト API と共通 API を除く、全ての HLSB API コールの前に必ず呼び出し、呼び出したプロセスが終了する前に必ず HlsClose() でクローズしなくてはなりません。

パラメータ

dmy 互換性のためのパラメータです。値は何でも構いません。

戻り値

成功すれば TRUE を、既にオープンされている場合 FALSE を返します。

HLSB-36PCI ユーザーズマニュアル

HlsResetBoard

書式

```
void HLSBAPI HlsResetBoard( void )
```

説明

カレントの HLSB-36PCI ボードをハードリセットします。
※ MKY36 が初期化されます。

戻り値

なし

HlsClrBoard

書式

```
BOOL HLSBAPI HlsClrBoard( int clr_mode )
```

説明

カレントボード上の両 MKY36 のメモリエリアをクリア、もしくはフィルします。

パラメータ

clr_mode 0 の時、両 MKY36 の全メモリエリアを 0 クリアします。
1 の時、両 MKY36 の全メモリエリアを 0 ~ 0x2BF の値でフィルします。(テスト用)

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsClose

書式

```
BOOL HLSBAPI HlsClose( void )
```

説明

基本 API 使用后、必要な終了処理をします。

パラメータ

なし

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。
通常、HLS_ERR_CLOSED 以外のエラーはありません。

HlsAddBoard

書式

```
BOOL HLSBAPI HlsAddBoard( WORD dmy )
```

説明

HLSB-36PCI を複数枚利用するときに HlsOpen() した後で更にボードを追加オープンします。

パラメータ

dmy 互換性のためのパラメータです。値は何でも構いません。

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsSetCurrentBoard

書式

```
BOOL HLSBAPI HlsSetCurrentBoard( UINT b_no )
```

説明

以下の API の対象となるカレントボードをボード番号で設定します。

HlsGetStatus(), HlsResetBoard(), HlsClrBoard(), HlsSetIM(), HlsGetIM(),
HlsGetII(), HlsDialog()

パラメータ

b_no ボード番号（オープンした順に 1 から付けられる。ボード ID+1）。

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsNumberOfChanel

書式

```
UINT HLSBAPI HlsNumberOfChanel( void )
```

説明

現在利用可能なチャンネル数を返します。

戻り値

現在利用可能なチャンネル数を返します。2～8の偶数になります。

HLSB-36PCI ユーザーズマニュアル

HlsGetStatus

書式

```
BOOL HLSBAPI HlsGetStatus( HlsbStat *hs )
```

説明

カレントボードの MKY36 ベーシックレジスタ情報を HlsbStat 構造体にセットします。

パラメータ

hs ステータス情報を格納する HlsbStat 構造体へのポインタ

戻り値

成功すれば TRUE を、失敗した場合 FALSE を返します。

HlsSetStatus

書式

```
BOOL HLSBAPI HlsSetStatus( HlsbStat *hs )
```

説明

HlsbStat 構造体のおりにカレントボードの MKY36 ベーシックレジスタを設定します。

HlsGetStatus() で現在値を取得後、必要なパラメータだけ書き換えてコールするのが安全です。

※ HlsResetBoard() 関数の実行で MKY36 は初期化され fh=1, bps=12 となります。

※ HlsSetStatus() は、変更するチャンネルの通信が停止中に行なってください。

※ HlsSetStatus() 後に HlsResetBoard() を行なわないで下さい。(HLSB-PCI と異なります)

パラメータ

hs 設定した HlsbStat 構造体へのポインタ

戻り値

成功すれば TRUE を、失敗した場合 FALSE を返します。

HlsbStat

HlsbStat 型は、“HLSB36PCI.h” 内で宣言されています。

この型の全メンバは MKY36 の現在の設定を参照する為のものです。

```
typedef struct {
    int    fh;        //FULL=1 / HALF=0
    int    bps;       //3Mbps=3 / 6Mbps=6 / 12Mbps=12 / EXT=0
    int    dae;       //High=1 / Lo=0 未使用
} status33;

typedef struct {
    status33    ch1,
               ch2;
} HlsbStat;
```

HlsGetData

書式

```
WORD HLSBAPI HlsGetData( UINT ch, UINT adr )
```

説明

HLS メモリの指定アドレスから 1 ワードのデータを読み込みます。
adr=0 の時は HLSB-PCI 互換の為 DLL 内で、adr=0x2C0 (SCR) に置換されます。

パラメータ

ch アクセスする MKY36 のチャンネルを指定。1 ~ 8
adr 読み込む HLS メモリのワードアドレス。0 ~ 0x2CA (ワード配列アクセス)

戻り値

読み込んだデータを返します。
戻り値が有効かどうか、直後で HlsGetLastError() をコールしてください。

HlsSetData

書式

```
BOOL HLSBAPI HlsSetData( UINT ch, UINT adr, WORD dat )
```

説明

HLS メモリの指定アドレスへ 1 ワードのデータを書き込みます。
adr=0 の時は HLSB-PCI 互換の為 DLL 内で、adr=0x2C0 (SCR) に置換されます。

パラメータ

ch アクセスする MKY36 のチャンネルを指定。1 ~ 8
adr 書き込む HLS メモリのワードアドレス。0 ~ 0x2CA (ワード配列アクセス)
dat 書き込むワードデータ

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsSetDaeHi

書式

```
BOOL HLSBAPI HlsSetDaeHi( UINT ch )
```

説明

互換性のために存在します。何も処理を行いません。

パラメータ

戻り値

TRUE

HLSB-36PCI ユーザーズマニュアル

HlsSetDaeLo

書式

```
BOOL HLSBAPI HlsSetDaeLo( UINT ch )
```

説明

互換性のために存在します。何も処理を行いません。

パラメータ

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsGetChk1

書式

```
WORD HLSBAPI HlsGetChk1( UINT ch )
```

説明

指定チャンネルの CHK1 パルスカウンターのカウント値を読み込みます。

パラメータ

ch カウント値を読み込む MKY36 のチャンネルを指定。1～8

戻り値

読み込んだデータを返します。HLSB-36PCI では 0～255 の数値が戻ります。
戻り値が有効かどうか、直後で HlsGetLastError() をコールしてください。

HlsGetChk2

書式

```
WORD HLSBAPI HlsGetChk2( UINT ch )
```

説明

指定チャンネルの CHK2 パルスカウンターのカウント値を読み込みます。

パラメータ

ch カウント値を読み込む MKY36 のチャンネルを指定。1～8

戻り値

読み込んだデータを返します。HLSB-36PCI では 0～255 の数値が戻ります。
戻り値が有効かどうか、直後で HlsGetLastError() をコールしてください。

HlsResetChk1

書式

```
BOOL HLSBAPI HlsResetChk1( UINT ch )
```

説明

指定チャンネルの CHK1 パルスカウンターのカウント値をリセットします。

パラメータ

ch カウント値をリセットする MKY36 のチャンネルを指定。1～8

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsResetChk2

書式

```
BOOL HLSBAPI HlsResetChk2( UINT ch )
```

説明

指定チャンネルの CHK2 パルスカウンターのカウント値をリセットします。

パラメータ

ch カウント値をリセットする MKY36 のチャンネルを指定。1～8

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsGetII

書式

```
BYTE HLSBAPI HlsGetII( void )
```

説明

互換性のために存在します。何も処理を行いません。

戻り値

TRUE

HlsGetIM

書式

```
BYTE HLSBAPI HlsGetIM( void )
```

説明

互換性のために存在します。何も処理を行いません。

戻り値

TRUE

HlsSetIM

書式

```
BOOL HLSBAPI HlsSetIM( BYTE im )
```

説明

互換性のために存在します。何も処理を行いません。

パラメータ

戻り値

TRUE

HlsGetCNo

書式

```
BYTE HLSBAPI HlsGetCNo( UINT ch )
```

説明

互換性のために存在します。HlsSetCNo() で最後に設定した値が読めます。
HLSB-36PCI には、HLSB-ISA の C No. は存在しません。

パラメータ

ch アクセスする MKY36 のチャンネルを指定。1～8

戻り値

HlsSetCNo() で最後に設定した値を返します。C No. を設定していない場合、0 を返します。

HlsSetCNo

書式

```
BOOL HLSBAPI HlsSetCNo( UINT ch, BYTE dat )
```

説明

互換性のために存在します。

パラメータ

ch アクセスする MKY36 のチャンネルを指定。1 ~ 8
dat 書き込むデータ。

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsGetSTNo

書式

```
BYTE HLSBAPI HlsGetSTNo( UINT ch )
```

説明

互換性のために存在します。HlsSetSTNo() で最後に設定した値が読めます。
HLSB-36PCI には、HLSB-ISA の ST No. は存在しません。

パラメータ

ch アクセスする MKY36 のチャンネルを指定。1 ~ 8

戻り値

HlsSetCNo() で最後に設定した値を返します。ST No. を設定していない場合、0 を返します。

HlsSetSTNo

書式

```
BOOL HLSBAPI HlsSetSTNo( UINT ch, BYTE dat )
```

説明

互換性のために存在します。

パラメータ

ch アクセスする MKY36 のチャンネルを指定。1 ~ 8
dat 書き込むデータ。

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HLSB-36PCI ユーザーズマニュアル

HlsDialog

書式

```
BOOL HLSBAPI HlsDialog( HWND hwnd )
```

説明

カレントのHLSB-36PCIを対象に、Satellite Control Dialogを表示させます。このダイアログからは運用数設定は行えませんので運用数設定後にコールしてください。

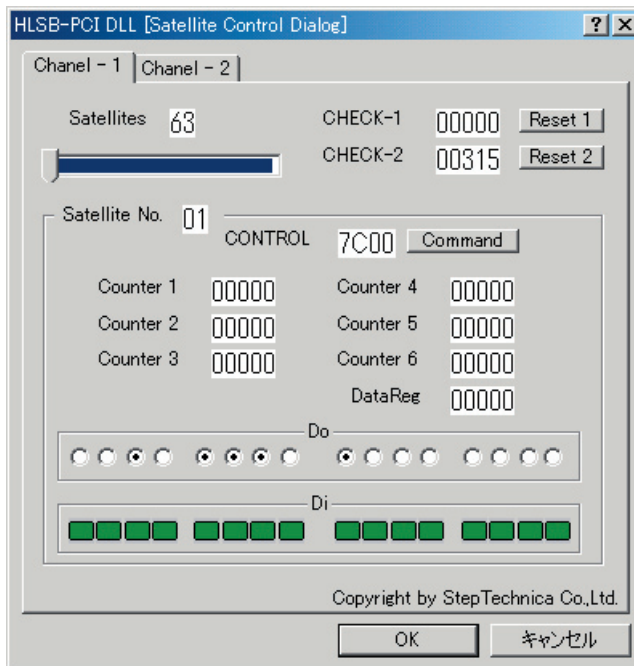
パラメータ

hwnd 親ウインドウのウインドウハンドル。

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

```
例 HlsOpen( 0xXXX ); // HLSB API の初期化処理
    HlsResetBoard(); // ボードのハードリセット
    HlsClrBoard( 0 ); // ボード上のメモリの0クリア
    HlsSetData( 1, 0x2C7, 0x12 ); // FULL, 6MBPS (BCR)
    HlsSetData( 1, 0, 10 ); // 運用数に10を設定 (SCR)
    HlsDialog( hwnd ); // 下の Satellite Control Dialog が表示される
```



Satellite Control Dialog

HlsDialog() で表示されるダイアログを利用して簡単なサテライトテストを行えます。

タブによるチャンネル切り替えや、トラックバーによる表示サテライトの変更も簡単に行え、Do 出力もラジオボタンをクリックすることで行えます。トラックバー中の青で表示されている部分が現在通信を行っているサテライトです。

Command ボタンをクリックすれば Command Set Dialog が現われますので、ここからコマンドをセットすることも可能です。

終了は“OK”ボタンでも“キャンセル”ボタンでも違いはありません。



5 ダイレクト API

5.1 概要

HLSB-36PCI を直接アクセスする API を他の API と分ける為に“ダイレクト API”と呼びます。API 数は少ないですが、単なるメモリリード/ライト命令が基本なので、ハードウェアの解説を参照しながら利用することで、HLSB-36PCI の全ての機能を柔軟に利用することが可能です。基本 API も内部でこれらのダイレクト API を呼び出すラッパーになっています。基本 API では、ライブラリ内部でハンドルの管理等を行っているため、基本 API とダイレクト API を同時に使用することはできません。

また、現バージョンの HLSB-36PCI デバイスドライバは、割り込みをサポートしていませんので、MKY36 の割り込みレジスタへ 0 以外の数値を書き込まない様に、ご注意ください。

5.2 仕様

HlsCountDevice

書式

```
INT HLSBAPI HlsCountDevice( void )
```

説明

搭載されている HLSB-36PCI の枚数を取得します。

戻り値

搭載されている HLSB-36PCI の枚数を返します。0 であれば、HLSB-36PCI が存在しないことを意味します。

HLSB-36PCI ユーザーズマニュアル

HlsOpenHandle

書式

```
HANDLE HLSBAPI HlsOpenHandle( int Inst )
```

説明

HLSB-36PCI へのハンドルを取得します。

パラメータ

Inst HLSB-36PCI のインスタンスを番号で指定します。
インスタンスは、存在する HLSB-36PCI に Windows が順に割り当てた番号です。どのインスタンスがどのボードなのかは特定できません。取得したハンドルからボード ID を調べることで、実際のボードを特定することができます。
4 枚以上のボードの区別の方法は、お客様で用意してください。

戻り値

指定したインスタンスの HLSB-36PCI へのハンドルを返します。

HlsCloseHandle

書式

```
BOOL HLSBAPI HlsCloseHandle( HANDLE Hdl )
```

説明

全ての操作を終了する時にハンドルを閉じます。

パラメータ

Hdl 対象 HLSB-36PCI へのハンドル。

戻り値

成功したら TRUE を、失敗したら FALSE を返します。
HlsGetLastError() で原因を探れます。

HlsReadByte, HlsReadWord, HlsReadDWord

書式

```
BOOL HLSBAPI HlsReadByte( HANDLE Hdl, ULONG Adr, BYTE* Dat )
BOOL HLSBAPI HlsReadWord( HANDLE Hdl, ULONG Adr, WORD* Dat )
BOOL HLSBAPI HlsReadDWord( HANDLE Hdl, ULONG Adr, DWORD* Dat )
```

説明

対象 HLSB-36PCI の指定アドレスから、バイト / ワード / ダブルワードのデータをリードします。
※ HlsReadByte, HlsReadDWord は、ハードの仕様により HlsReadWord を DLL 内部で使用しています。

パラメータ

Hdl 対象 HLSB-36PCI へのハンドル。
Adr リードしたいデータのあるアドレス。ワードアクセスでは 2 の倍数、ダブルワードアクセスでは 4 の倍数でなくてはなりません。
Dat リードしたデータを格納する場所へのポインタ。

戻り値

成功したら TRUE を、失敗したら FALSE を返します。
HlsGetLastError() で原因を探れます。

HlsWriteByte, HlsWriteWord, HlsWriteDWord

書式

```
BOOL HLSBAPI HlsWriteByte( HANDLE Hdl, ULONG Adr, BYTE Dat )
BOOL HLSBAPI HlsWriteWord( HANDLE Hdl, ULONG Adr, WORD Dat )
BOOL HLSBAPI HlsWriteDWord( HANDLE Hdl, ULONG Adr, DWORD Dat )
```

説明

対象 HLSB-36PCI の指定アドレスへ、バイト / ワード / ダブルワードのデータをライトします。
※ HLS メモリエリアへのバイトライトはハード仕様により無視されます。
※ HlsWriteDWord は、ハードの仕様により HlsWriteWord を DLL 内部で使用しています。

パラメータ

Hdl 対象 HLSB-36PCI へのハンドル。
Adr ライトしたいアドレス。ワードアクセスでは 2 の倍数、ダブルワードアクセスでは 4 の倍数でなくてはなりません。
Dat ライトするデータ。

戻り値

成功したら TRUE を、失敗したら FALSE を返します。
HlsGetLastError() で原因を探れます。

■開発・製造

株式会社ステップテカ

〒358-0011 埼玉県入間市下藤沢 757-3

TEL: 04-2964-8804

FAX: 04-2964-7653

<http://www.steptechnica.com/>

info@steptechnica.com

ハイスピードリンクシステム

HLSB-36PCI ユーザーズマニュアル

ドキュメント No. : STD-HLSB-36PCI-V1.0J

発行年月日 : 2004年6月 第1版