

HLSB-PCI Rev 1.01

ユーザーズマニュアル

2004, 7, 8

はじめに

この度は、弊社 HLSB-PCI 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 MKY33 に関する詳細につきましては、「MKY33 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

HLSB-PCI ユーザーズマニュアル

1 HLSB-PCI のハードウェア

1.1 特徴

HLSB-PCI は、HLS (Hi-speed Link System) のセンター IC である MKY33 (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 MKY33 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、MKY33 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。

1.2 仕様

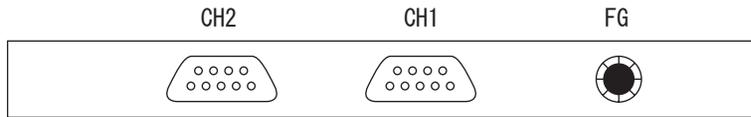
HLSB-PCI の各種仕様を表 . 1-1 に示します。

表 . 1-1 HLSB-PCI 仕様

名 称	HLSB-PCI
搭載チップ	MKY33 × 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	9pin D-sub × 2
電 源	DC +5V
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	Windows98/Me/2000用ドライバ・標準ライブラリ

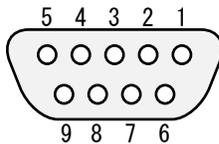
1.3 インターフェイス

パネル面



CH1	MKY-33 channel-1のインターフェイス
CH2	MKY-33 channel-2のインターフェイス
FG	必ずアースへ接続して下さい。

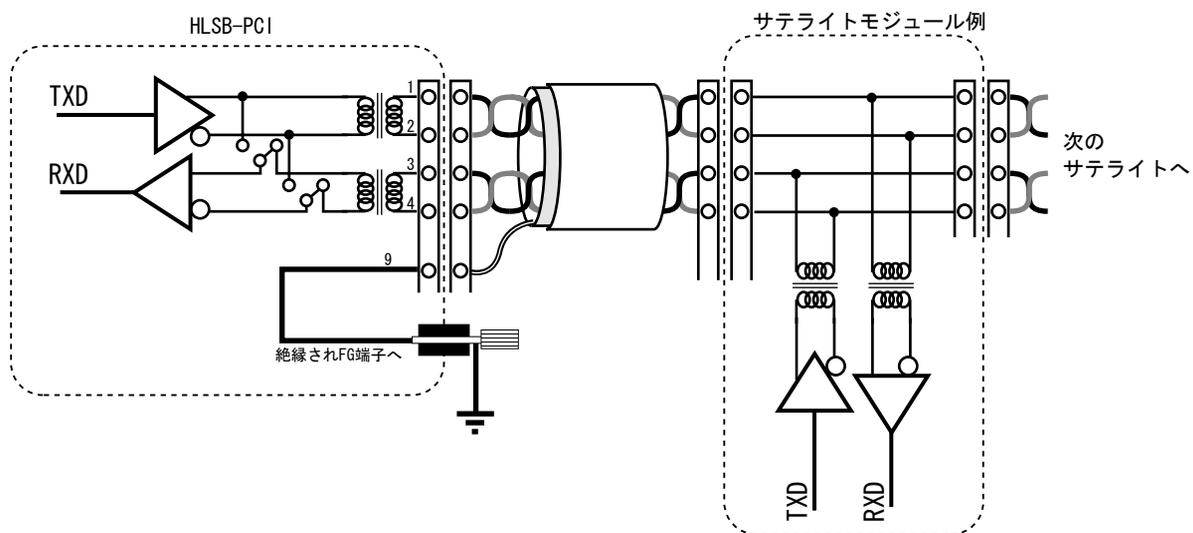
パネル面から見たピン配置（上下の向きに注意）



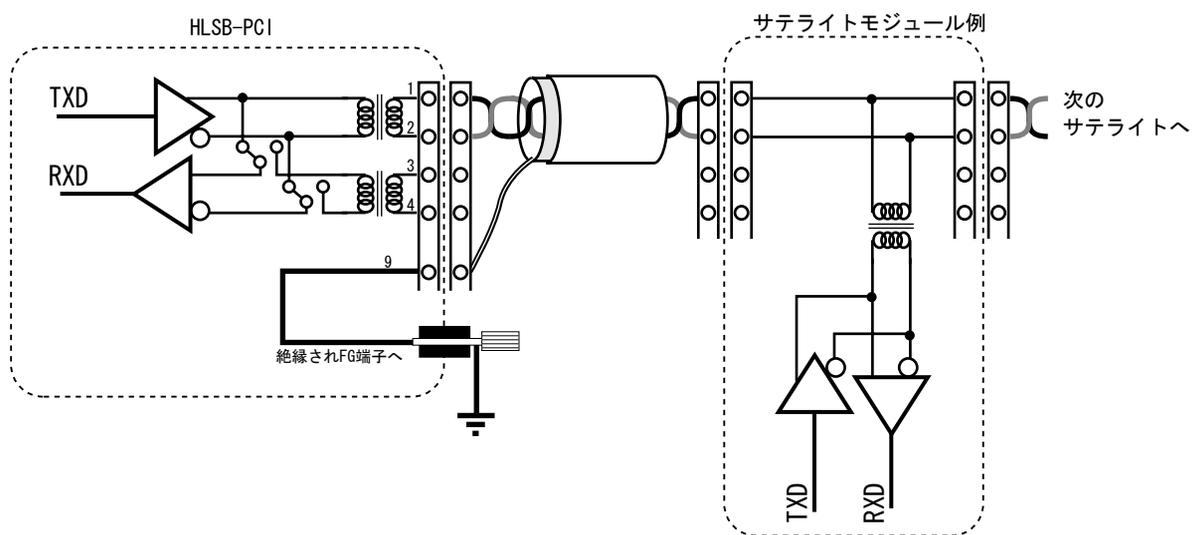
Pin	名称	備考
1	CT1+	センター送信 +
2	CT1-	センター送信 -
3	ST1+	サテライト送信 +
4	ST1-	サテライト送信 -
5	CT2+	センター送信 +
6	CT2-	センター送信 -
7	ST2+	サテライト送信 +
8	ST2-	サテライト送信 -
9	F. G.	伝送ケーブルのシールド線を接続します。パネル面のFG端子に接続されているのでFG端子をGND処理しない場合シールドが浮いてしまいます。

1.4 サテライトとの接続モデル

フルデュプレックス

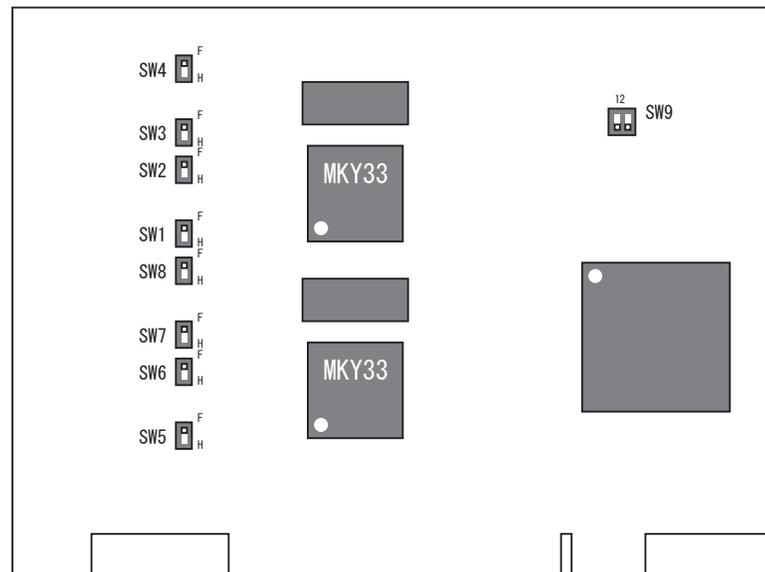


ハーフデュプレックス



1.5 SW 設定

HLSB-PCI デザイン（出荷時設定）



上図のとおり、HLSB-PCI 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 MKY33 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 MKY33 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。

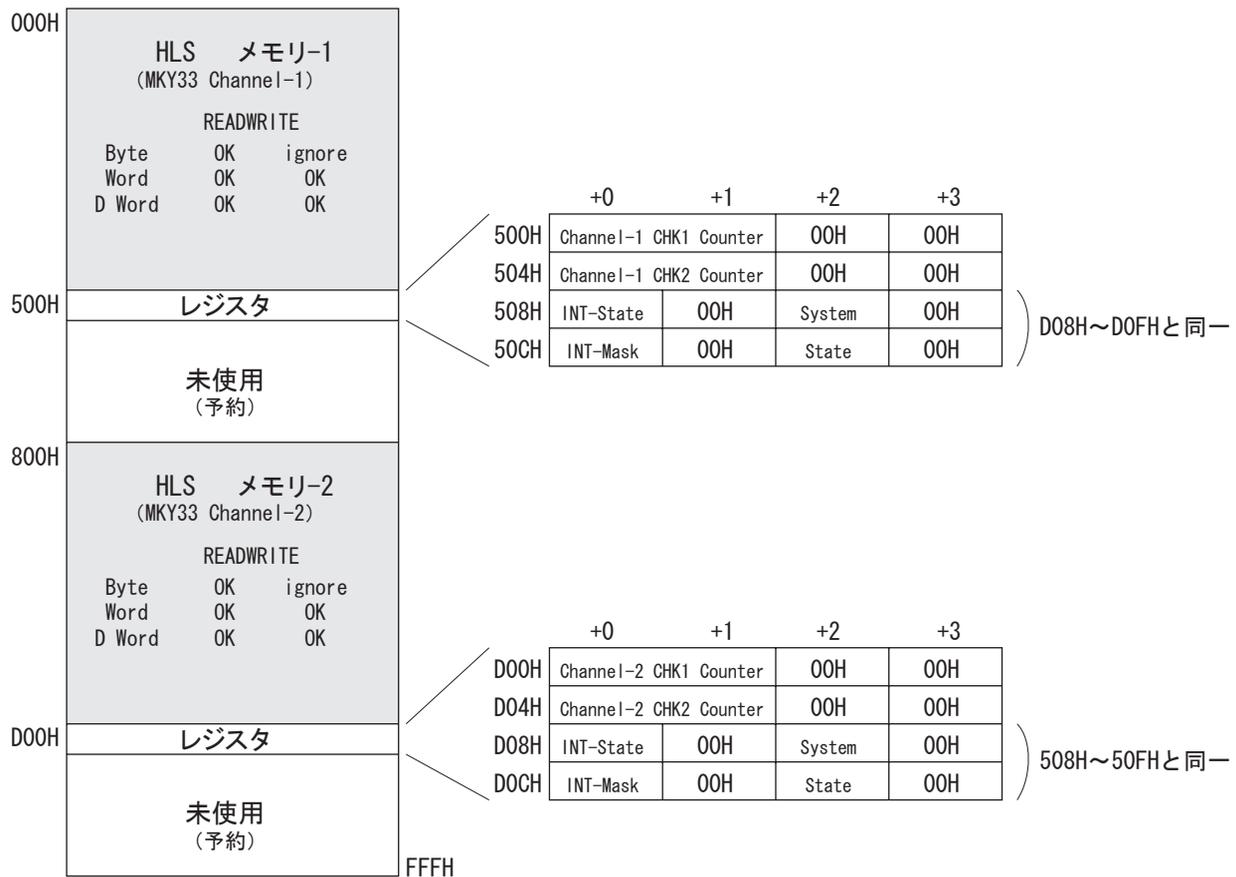
MKY33 チップへのデュプレックス設定は、ソフトウェアで行います。詳しくは、ステータスレジスタの詳細をご覧ください。

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-PCI 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（“0”）で構いません。

1.6 メモリマップ

HLSB-PCI ユーザーズマニュアル



上図は、HLSB-PCI 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-PCI はバイト / ワード / ダブルワードの各リード / ライトに対応していますが、HLS メモリエリア (000H ~ 4FFH と 800H ~ CFFH) へのバイトライトは無視される仕様になっています。HLS メモリエリアは、MKY33 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、「Hi-speed Link System 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 FFH 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-PCI は、計 8 個のレジスタを持っています。各 MKY33 のフル / ハーフ設定、通信レート

設定や個別のリセットなどは、このレジスタを通してソフトウェアで設定を変更します。また、CHK1, CHK2 のパルスカウンタの数値もレジスタから取得します。508H ~ 50FH と、D08H ~ D0FH は常に同一の内容です。リード/ライトともどちらを通して行っても構いません。

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、チップリセットもかかります。

チャンネル 1 CHK1 カウンタ (500H ~ 501H)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R		R/W							

チャンネル 1 の MKY33 から出力される CHK1 パルスをカウントするバイナリアップカウンタです。

C31 が MSB、C0 が LSB で、ボードリセット、または 500H へのバイト / ワード / ダブルワードへのダミーデータ書き込み動作によって 0 にクリアされます。フルカウント後は、FFFFH のままロールアップはしません。

チャンネル 1 CHK2 カウンタ (504H ~ 505H)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R		R/W							

チャンネル 1 の MKY33 から出力される CHK2 パルスをカウントするバイナリアップカウンタです。

C31 が MSB、C0 が LSB で、ボードリセット、または 504H へのバイト / ワード / ダブルワードへのダミーデータ書き込み動作によって 0 にクリアされます。フルカウント後は、FFFFH のままロールアップはしません。

チャンネル 2 CHK1 カウンタ (508H ~ 509H)

HLSB-PCI ユーザーズマニュアル

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W							

チャンネル2のMKY33から出力されるCHK1パルスをカウントするバイナリアップカウンタです。C31がMSB、C0がLSBで、ボードリセット、または508Hへのバイト/ワード/ダブルワードへのダミーデータ書き込み動作によって0にクリアされます。フルカウント後は、FFFFHのままロールアップはしません。

チャンネル2 CHK2 カウンタ (50CH ~ 50DH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W							

チャンネル2のMKY33から出力されるCHK2パルスをカウントするバイナリアップカウンタです。C31がMSB、C0がLSBで、ボードリセット、または50CHへのバイト/ワード/ダブルワードへのダミーデータ書き込み動作によって0にクリアされます。フルカウント後は、FFFFHのままロールアップはしません。

システムレジスタ (50AH)

Bit:	7	6	5	4	3	2	1	0
	ID2	ID1	RUN2	RUN1	-	-	RST2	RST1
Initial:	*	*	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

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 の通信を停止する全ての要因で 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 の通信を停止する全ての要因で 0 になります。

RST2

当ビットへの 1 書き込みにより、チャンネル 2 MKY33 にチップリセットがかかります。

リセット終了と同時に 0 に戻ります。

RST1

当ビットへの 1 書き込みにより、チャンネル 1 MKY33 にチップリセットがかかります。

リセット終了と同時に 0 に戻ります。

HLSB-PCI ユーザーズマニュアル

ステータスレジスタ (50EH)

Bit:	7	6	5	4	3	2	1	0	
	C2FH	C2BPS1	C2BSP0	C1FH	C1BPS1	C1BPS0	C2DAE	C1DAE	
Initial:	1	1	0	1	1	0	0	0	:D8h
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

※このレジスタへの書き込み時、意図しないビットを変更しない様ご注意ください。

C2FH

チャンネル2のフルデュプレックス/ハーフデュプレックスを設定します。1がフルデュプレックスで、0がハーフデュプレックスになります。変更後は、同チャンネルにチップリセットをかけてください。

ディップスイッチ SW5 ~ SW8 もこのビットの運用時の設定に合わせて設定します。

C2BPS1, C2BSP0

チャンネル2の通信レートを設定します。各ビット“11”が12Mbps、“10”が6Mbps、“01”が3Mbpsの設定です。“00”の設定では通信を行うことはできません。変更後は、同チャンネルにチップリセットをかけてください。

C1FH

チャンネル1のフルデュプレックス/ハーフデュプレックスを設定します。1がフルデュプレックスで、0がハーフデュプレックスになります。変更後は、同チャンネルにチップリセットをかけてください。

ディップスイッチ SW1 ~ SW4 もこのビットの運用時の設定に合わせて設定します。

C1BPS1, C1BSP0

チャンネル1の通信レートを設定します。各ビット“11”が12Mbps、“10”が6Mbps、“01”が3Mbpsの設定です。“00”の設定では通信を行うことはできません。変更後は、同チャンネルにチップリセットをかけてください。

C2DAE

チャンネル2のDAE端子への入力レベルを設定します。1でHiレベル、0でLoレベルがDAE端子に印加されます。MKY33のDAEに関して十分な理解がない場合、初期値のままご使用ください。

C1DAE

チャンネル1のDAE端子への入力レベルを設定します。1でHiレベル、0でLoレベルがDAE端子に印加されます。MKY33のDAEに関して十分な理解がない場合、初期値のままご使用ください。

割り込みマスクレジスタ (50CH)

Bit:	7	6	5	4	3	2	1	0
	C2DR	C2CHK	C2SCR	C2SCW	C1DR	C1CHK	C1SCR	C1SCW
Initial:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HLSB-PCI からどの要因でハードウェア割り込みを発生させるか、各要因ごとにオン / オフを設定します。全ビットが0のときハードウェア割り込みは発生しません。

割り込みステータスレジスタの解説と併せて、十分ご理解の上ご使用ください。

C2DR (Channel-2 DREQ)

当ビットが1のとき、チャンネル2のMKY33のDREQ信号がHi出力になったときに割り込みを発生します。

C2CHK (Channel-2 CHK2)

当ビットが1のとき、チャンネル2のMKY33のCHK2パルス発生により割り込みを発生します。

C2SCR (Channel-2 SCANR STOP)

当ビットが1のとき、チャンネル2のMKY33に対する“SCANRストップ”が有効になります。

SCANRストップとは、MKY33のSCANR信号がHi出力になったときに同チップのDAE端子にHiを印加し、割り込みを発生させる機能です。

この割り込みが発生したとき、必ずステータスレジスタのC2DAEビットは1になります。

C2SCW (Channel-2 SCANW STOP)

当ビットが1のとき、チャンネル2のMKY33に対する“SCANWストップ”が有効になります。

SCANWストップとは、MKY33のSCANR信号がHi出力になったときに同チップのDAE端子にHiを印加し、割り込みを発生させる機能です。

この割り込みが発生したとき、必ずステータスレジスタのC2DAEビットは1になります。

C1DR (Channel-1 DREQ)

当ビットが1のとき、チャンネル1のMKY33のDREQ信号がHi出力になったときに割り込みを発生します。

C1CHK (Channel-1 CHK2)

当ビットが1のとき、チャンネル1のMKY33のCHK2パルス発生により割り込みを発生します。

C1SCR (Channel-1 SCANR STOP)

当ビットが1のとき、チャンネル1のMKY33に対する“SCANRストップ”が有効になります。

この割り込みが発生したとき、必ずステータスレジスタのC1DAEビットは1になります。

C1SCW (Channel-1 SCANW STOP)

当ビットが1のとき、チャンネル1のMKY33に対する“SCANWストップ”が有効になります。

この割り込みが発生したとき、必ずステータスレジスタのC1DAEビットは1になります。

HLSB-PCI ユーザーズマニュアル

割り込みステータスレジスタ (508H)

Bit:	7	6	5	4	3	2	1	0
	C2DR	C2CHK	C2SCR	C2SCW	C1DR	C1CHK	C1SCR	C1SCW
Initial:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

ハードウェア割り込みを発生させることのできる各要因の成立状況を示すレジスタです。

G2DR (Channel-2 DREQ)

チャンネル 2 の MKY33 の DREQ 信号が Hi 出力のときに 1 になります。同信号が Lo 出力のときに 0 になります。

当ビットと、割り込みマスクレジスタの同名ビットとの AND 条件で割り込みが発生します。

G2CHK (Channel-2 CHK2)

チャンネル 2 の MKY33 の CHK2 パルス発生により 1 になります。同 CHK2 パルスカウンタへのリードアクセスにより 0 になります。

当ビットと、割り込みマスクレジスタの同名ビットとの AND 条件で割り込みが発生します。

G2SCR (Channel-2 SCANR STOP)

割り込みマスクレジスタの同名ビットが 1 のとき、チャンネル 2 の MKY33 の SCANR 信号が Hi 出力になったときに同チップの DAE 端子に Hi を印加し、1 になります。

ステータスレジスタの G2DAE ビットを 0 にすることで、当ビットも 0 になります。

G2SCW (Channel-2 SCANW STOP)

割り込みマスクレジスタの同名ビットが 1 のとき、チャンネル 2 の MKY33 の SCANW 信号が Hi 出力になったときに同チップの DAE 端子に Hi を印加し、1 になります。

ステータスレジスタの G2DAE ビットを 0 にすることで、当ビットも 0 になります。

C1DR (Channel-1 DREQ)

チャンネル 1 の MKY33 の DREQ 信号が Hi 出力のときに 1 になります。同信号が Lo 出力のときに 0 になります。

当ビットと、割り込みマスクレジスタの同名ビットとの AND 条件で割り込みが発生します。

C1CHK (Channel-1 CHK2)

チャンネル 1 の MKY33 の CHK2 パルス発生により 1 になります。同 CHK2 パルスカウンタへのリードアクセスにより 0 になります。

当ビットと、割り込みマスクレジスタの同名ビットとの AND 条件で割り込みが発生します。

C1SCR (Channel-1 SCANR STOP)

割り込みマスクレジスタの同名ビットが 1 のとき、チャンネル 1 の MKY33 の SCANR 信号が Hi 出力になったときに同チップの DAE 端子に Hi を印加し、1 になります。

ステータスレジスタの G2DAE ビットを 0 にすることで、当ビットも 0 になります。

C1SCW (Channel-1 SCANW STOP)

割り込みマスクレジスタの同名ビットが 1 のとき、チャンネル 1 の MKY33 の SCANW 信号が Hi 出力になったときに同チップの DAE 端子に Hi を印加し、1 になります。

ステータスレジスタの G2DAE ビットを 0 にすることで、当ビットも 0 になります。

1.8 SCANR/SCANW ストップ機能

SCANR/SCANW ストップ機能は、割り込みを利用した応用機能であり、HLS の通信サイクル単位で何らかの処理を行いたいときに利用します。センターから各サテライトへ送る出力データを処理したい場合は SCANW ストップ、逆に各サテライトからセンターに送られてくる入力データを処理したい場合は SCANR ストップの機能を使います。

この機能は、SCANR/SCANW の各タイミング通知パルスの出力をトリガーに、ハードウェアで MKY33 の DAE 端子を Hi にすることで DAE 機能を強制的にオフにし、同時に割り込みを発生させてユーザに通知する機能です。DAE 機能がオフの間、MKY33 のバッファ RAM の状態は書き換えられる事があるため、その間に必要な処理を行い、終了時に DAE 機能をオンにすることで、サイクル単位で行う必要のある処理を実現することができます。

長時間 DAE 機能がオフのままになっていると、やがて通信が一時的に停止してしまいます。通信を停止させたくない場合は、DAE 機能のオフ時間が、下表の時間を超えないように処理を行わなくてはなりません。

	フルデュプレックス	ハーフデュプレックス
3Mbps	60.7 μ s	118 μ s
6Mbps	30.4 μ s	59 μ s
12Mbps	15.2 μ s	29.5 μ s

2. 付属ソフトウェアについて

HLSB-PCI ユーザーズマニュアル

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、Windows98, Windows Me, Windows 2000 Pro, WindowsXP です。

初めて HLSB-PCI 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 CD 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 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 CD 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-ISA 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。

今後も対応 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 CD 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-PCI 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。

2.3 ファイルの種類

付属 CD 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。

HlsbPci.dll

【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

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。

HlsbPci.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。

HlsbPci.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かかる制限

現バージョンの Windows ドライバでは、割り込みをサポートしていません。そのため、割り込みを発生させる HlsSetIM() 関数は、割り込みマスクレジスタへの実際の書き込み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 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-ISA DLL Ver2.00 を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-PCI 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、新しいダイレクト API を利用しなくてはなりません。

3. 割り込み

HLSB-PCI では、割り込みマスクレジスタの設定が、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いませんので、HlsSetIM() 関数は、割り込みマスクレジスタへの実際の書き込みを行いません。ただし、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_NORESOURCE	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、利用しやすい、おすすめの 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。また、その件に関して一切のサポートいたしません。

4.2 ハードウェアの設定

基本 API で利用する HLSB-PCI 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-PCI 上の MKY33 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。

HlsResetBoard

書式

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

説明

カレントの HLSB-ISA ボードをハードリセットします。

戻り値

なし

HlsClrBoard

書式

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

HLSB-PCI ユーザーズマニュアル

説明

カレントボード上の両 MKY33 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 MKY33 の全メモリエリアを 0 クリアします。
1 の時、両 MKY33 の全メモリエリアを 0 ~ 0x27F 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-ISA を複数枚利用するときに HlsOpen() した後に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。

HlsGetStatus

書式

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

説明

カレントボードのステータス情報を 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ステータス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。

また、fh や bps を変更した後は、HlsResetBoard() を行ってください。

HLSB-PCI ユーザーズマニュアル

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。

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。

パラメータ

ch アクセスする MKY33 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 ~ 0x27F (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。

パラメータ

HlsbStat

HlsbStat 型は、“HlsbIsa.h” 内で宣言されています。

この型の全メンバは MKY33 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
} status33;

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

ch アクセスする MKY33 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 ~ 0x27F (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

DAE 信号の元の状態にかかわらず指定チャンネルの DAE 信号を Hi レベルにします。

パラメータ

ch DAE を Hi にする MKY33 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。

HlsSetDaeLo

書式

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

説明

DAE 信号の元の状態にかかわらず指定チャンネルの DAE 信号を Lo レベルにします。

パラメータ

ch DAE を Lo にする MKY33 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。

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 MKY33 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-PCI ユーザーズマニュアル

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 MKY33 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。

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 MKY33 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 MKY33 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

カレントボードの現在の割り込みステータスレジスタの値を取得します。

戻り値

現在の割り込みステータスレジスタの値を返します。

戻り値が有効かどうか、直後で HlsGetLastError() をコールしてください。

HlsGetIM

書式

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

説明

カレントボードの現在の割り込みマスクレジスタの値を取得します。

戻り値

現在の割り込みマスクレジスタの値を返します。

戻り値が有効かどうか、直後で HlsGetLastError() をコールしてください。

HlsSetIM

書式

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

説明

カレントボードの割り込みマスクレジスタに新しい値を設定します。

パラメータ

im 新しい割り込みマスク。

戻り値

成功したら TRUE を、エラーの時は FALSE を返します。

HlsGetCNo

書式

HLSB-PCI ユーザーズマニュアル

```
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-PCI 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 MKY33 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 MKY33 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-PCI 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 MKY33 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 MKY33 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。

HlsDialog

書式

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

説明

カレントの HLSB-ISA 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が運用数設定後にコールしてください。

パラメータ

hwnd 親ウィンドウのウィンドウハンドル。

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, 0, 10 ); // 運用数に 10 を設定。
    HlsDialog( hwnd ); // 下の Satellite Control Dialog が表示される。
```

5 ダイレクト API

5.1 概要

HLSB-PCI 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-PCI 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-PCI 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、割り込みマスクレジスタへ 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

HLSB-PCI ユーザーズマニュアル

HlsCountDevice

書式

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

説明

搭載されている HLSB-PCI の枚数を取得します。

戻り値

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

HlsOpenHandle

書式

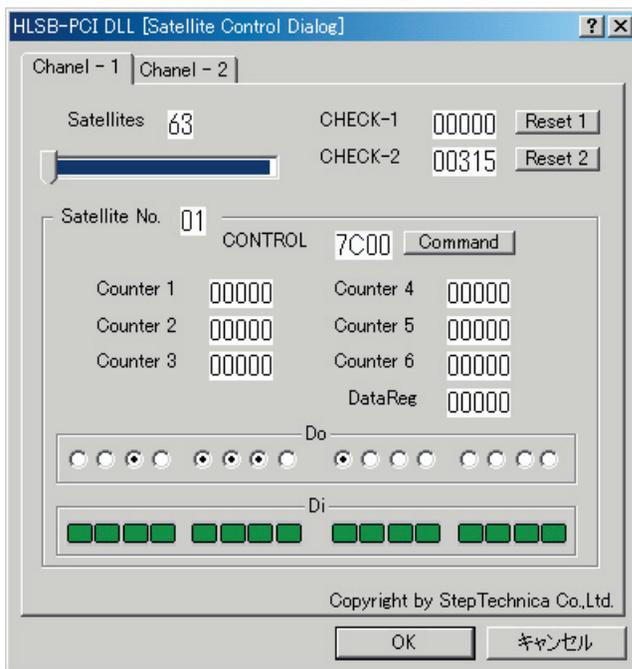
```
HANDLE HLSBAPI HlsOpenHandle( int Inst )
```

説明

HLSB-PCI へのハンドルを取得します。

パラメータ

Inst HLSB-PCI のインスタンスを番号で指定します。



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。



インスタンスは、存在する HLSB-PCI 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-PCI 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-PCI 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-PCI 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。

パラメータ

Hdl 対象 HLSB-PCI 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

HLSB-PCI ユーザーズマニュアル

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-PCI 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。

パラメータ

Hdl 対象 HLSB-PCI 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。

株式会社ステップテクニカ

〒358-0011 埼玉県入間市下藤沢 757-3

TEL 042-964-8804 FAX 042-964-7653