

高速、大容量入力ボード (LVDS インターフェース)

MS-PCI ソフト(上位DLL)説明書

第2版

株式会社 ファード

〒183-0006 府中市緑町 3-8-2 新東邦ビル 4 F
TEL 042-362-5072 FAX 042-369-8026
fird@coral.ocn.ne.jp www2.ocn.ne.jp/~fird

【改版履歴】

年月日	改 版 内 容
2002/7/22	初 版
2003/11/8	第 2 版 MSP_SetStoreCndt()変更、MSP_SetINPUT_Mode()を追加 1
2005/9/30	第 3 版 2000/XP のインストール説明追加

目 次

1 . 添付ファイル.....	3
1 - 1 . ドライバ.....	3
1 - 2 . 関数 (DLL で提供し、WindowsNT/2000/XP は、同じファイルを使用します。)	3
1 - 3 . サンプルプログラム.....	3
2 . ドライバインストール.....	4
2 - 1 . WindowsNT の場合	4
2 - 2 . Windows2000/XP の場合	4
3 . テストプロ実行.....	5
4 . 標準関数.....	6
MSP_Open ()	7
MSP_Close ()	8
MSP_GetStatus ()	9
MSP_SetControl ()	10
MSP_SetCMask ()	11
MSP_SetIMask ()	12
MSP_ClearIMask ()	13
MSP_ClearINT ()	14
MSP_GetMEMSize ()	15
MSP_SetCommand ()	16
MSP_SetStoreCndt ()	17
MSP_SetINPUT_Mode ()	18
MSP_BlockRead ()	19
MSP_BlockWrite ()	20

1 . 添付ファイル

1 - 1 . ドライバ

(1) WindowsNT

¥NT_Driver¥ MSPCI.SYS (ドライバー本体)
MSPCI.INI (インストール情報)
ENABLE.BAT (インストール実行バッチ)
REGINI.EXE (Windows側のレジストリ登録用)

(2) Windows2000/XP

¥2k_Driver¥ ms_pci.sys (ドライバー本体)
MSpci.inf (インストール情報)

1 - 2 . 関数 (DLL で提供し、WindowsNT/2000/XP は、同じファイルを使用します。)

(1) 下位 DLL

MSPCIDLL.LIB (コンパイル時の参照ファイル)
MSPCIDLL.DLL (関数本体)
MSPCIDLL.H (関数定義)

(2) 上位 DLL

MSPCIAPI.LIB (コンパイル時の参照ファイル)
MSPCIAPI.DLL (関数本体)
MSPCIAPI.H (関数定義)

・以上のファイルは、サンプルプログラムのソースファイル (¥MsTestSamle¥) に含まれています

1 - 3 . サンプルプログラム

¥MsTestSample¥ MSPCITEST.EXE (実行ファイル)
およびその project file 一式

2 . ドライバインストール

2 - 1 . WindowsNT の場合

- (1) Window NT(Ver 4.0)で、上のドライバ部分を HardDisk のディレクトリに Copy し、ENABLE.BAT をマウスのダブルクリックで実行します。(または DOS 窓で実行します)。所定のところにドライバファイルがコピーされます。
- (2) この後、PC を再立ち上げて下さい。
- (3) 立ち上がったら、コントロールパネルの「デバイス」を立ち上げ、「MS_PCI」を探し、それを「開始」させて下さい。「開始」できれば、ドライバはインストール完了です。

2 - 2 , Windows2000/XP の場合

- (1) パソコンの電源を切って、本ボードを P C I バスに挿入します。
- (2) 立ち上げの途中で、P C ボードのドライバーのインストールを要求してきますので、MSpci . inf を指定してください。後は、自動的にインストールをしてくれます。
- (3) インストールが、終わりましたら <コントロールパネル> <システム> <ハードウェア> <デバイスマネージャ> に以下の表示があれば終了です。

```
F i r d M S - P C I
  |
  └─ M S - P C I
```

3 . テストプロ実行

- 1) 上の添付ファイルの、下位 DLL、上位 DLL、サンプルプロを HardDisk の 1 つのディレクトリにコピーして下さい。
- 2) MSPCITEST.EXE をダブルクリックで実行して下さい。
- 3) このテストプロで、ファイルデータの書き込みテスト、読み出し、データのコンペアができ、さらに、個別レジスタの設定もできます。
- 4) 「Input Start」で外部からの入力可能にしておけば、入力条件が成立すれば、その時の Data を、個別レジスタで指定されたアドレスに指定サイズ分取り込みます。
- 5) 「Read Start」で、指定されたファイルに指定サイズ、DIMM メモリ Data を読み出し書き込みます。

詳細は「MS-PCI テストプロ説明書」をご参照下さい。開発手順

- 1) 下位 DLL、上位 DLL を開発するプロジェクトのディレクトリにコピーしておきます。
- 2) Visual C++の「プロジェクト」 > 「設定」で「リンク」のタグの「オブジェクト / ライブラリ」の中に、MSPCIDLL.LIB と MSPCIAPI.LIB を書き込み指定します。
- 3) これで以下の標準関数を使用したプログラムの開発ができます。

4 . 標準関数

本基板 MS - PCI は、製品として外部からの入力動作を行うための機能がすでに組み込まれています。その動作を行わせるための添付のソフトウェアの説明を行います。

以下の関数を DLL にて用意しております。

- | | |
|----------------------------|------------------------------------|
| 1) MSP_Open () | : ドライバのオープン |
| 2) MSP_Close () | : ドライバのクローズ |
| 3) MSP_GetStatus () | : 外部信号 (STATUS3 ~ 0) と内部状態信号読み出す |
| 4) MSP_SetControl () | : 外部信号 (CONT3 ~ 0) への出力 |
| 5) MSP_SetCMask () | : 受信制御線の有効・無効を設定する |
| 6) MSP_SetIMask () | : 割り込み関係の Mask の設定を行う |
| 7) MSP_ClearIMask () | : 割り込み関係の Mask のクリアを行う |
| 8) MSP_ClearINT () | : 割り込み要因のクリアを行う |
| 9) MSP_GetMEMSize () | : 搭載メモリのサイズを得る |
| 10) MSP_SetCommand () | : 収録の許可、禁止などを命令する |
| 11) MSP_SetStoreCndt () | : 外部データを蓄積するアドレス、サイズなどを設定する |
| 12) MSP_SetINPUT_Mode () | : 入力時のビットサイズを設定する |
| 13) MSP_BlockRead () | : メモリの指定アドレスから指定サイズのデータを読み出す |
| 14) MSP_BlockWrite () | : メモリの指定アドレスから指定サイズのデータを書き込む |

MSP_Open ()

関数名	MSP_Open
プロトタイプ	HANDLE MSP_Open(void)
機能	ドライバのオープン オープンできたら、MS-PCI 基板が使用可能となる。
引数	なし
戻り値	HANDLE :オープンしたボードのハンドル番号 この HANDLE で、以後の関数で使用する。
備考	

MSP_Close ()

関数名	MSP_Close
プロトタイプ	void MSP_Close(HANDLE hHND)
機能	ドライバのクローズ
引数	HANDLE hHND : オープンしたボードのハンドル値
戻り値	なし
備考	

MSP_GetStatus ()

関数名	MSP_GetStatus
プロトタイプ	WORD MSP_GetStatus(HANDLE hHND)
機能	外部信号 (EXT_IN3 ~ 0) とその他入力信号の状態を読み出す
引数	HANDLE hHND : オープンしたボードのハンドル値
戻り値	<p>WORD</p> <p>D15 INPUT 外部からの入力許可なら 1、禁止状態なら 0</p> <p>D14 ~ D12 空き</p> <p>D11 RESERVE 入力信号 RESERVE_b の状態、1 なら High、0 なら Low</p> <p>D10 READY 入力信号 READY_b の状態、1 なら High、0 なら Low</p> <p>D9 VSYNC 入力信号 VSYNC_b の状態、1 なら High、0 なら Low</p> <p>D8 HSYNC 入力信号 READY_b の状態、1 なら High、0 なら Low</p> <p>D7 RESERVE 入力信号 RESERVE_a の状態、1 なら High、0 なら Low</p> <p>D6 READY 入力信号 READY_a の状態、1 なら High、0 なら Low</p> <p>D5 VSYNC 入力信号 VSYNC_a の状態、1 なら High、0 なら Low</p> <p>D4 HSYNC 入力信号 READY_a の状態、1 なら High、0 なら Low</p> <p>D3 EXT_IN3 外部からの入力信号 (LVTTTL) 状態、1 なら High、0 なら Low</p> <p>D2 EXT_IN2 外部からの入力信号 (LVTTTL) 状態、1 なら High、0 なら Low</p> <p>D1 EXT_IN1 外部からの入力信号 (LVTTTL) 状態、1 なら High、0 なら Low</p> <p>D0 EXT_IN0 外部からの入力信号 (LVTTTL) 状態、1 なら High、0 なら Low</p>
備考	

MSP_SetControl ()

関数名	MSP_SetControl
プロトタイプ	void MSP_SetControl(HANDLE hHND、WORD cont)
機能	外部信号 (CONT3 ~ 0) への出力
引数	HANDLE hHND : オープンしたボードのハンドル値 WORD cont : D15 ~ D4 空き D3 EXT_OUT3 外部への出力信号 (LVTTTL) 1 で High、0 で Low D2 EXT_OUT2 外部への出力信号 (LVTTTL) 1 で High、0 で Low D1 EXT_OUT1 外部への出力信号 (LVTTTL) 1 で High、0 で Low D0 EXT_OUT0 外部への出力信号 (LVTTTL) 1 で High、0 で Low
戻り値	なし
備考	

MSP_SetCMask ()

関数名	MSP_SetCMask
プロトタイプ	void MSP_SetCMask(HANDLE hHND、WORD Cmask)
機能	受信制御線の有効・無効を設定する マスクされた信号は、入力 Data の収録条件には入らなくなる (無視される)。 マスクされていない信号の Active の AND で入力条件成立とする。
引数	HANDLE hHND : オープンしたボードのハンドル値 WORD Cmask : D15 ~ D3 空き D2 R_MASK 1 : READY 信号を Mask する、0 : Mask しない D1 V_MASK 1 : VSYNC 信号を Mask する、0 : Mask しない D0 H_MASK 1 : HSYNC 信号を Mask する、0 : Mask しない
戻り値	なし
備考	

MSP_SetIMask ()

関数名	MSP_SetIMask アイ
プロトタイプ	void MSP_SetIMask(HANDLE hHND、WORD Imask)
機能	割り込み関係の Mask の設定を行う
引数	HANDLE hHND : オープンしたボードのハンドル値 WORD Imask : D15 ~ D9 空き D10 M_INT2 1 : INT2 割り込みをマスクする。 D9 M_INT1 1 : INT1 割り込みをマスクする。 D8 M_ENT 1 : 転送終了割り込みをマスクする。 D7 ~ D0 空き
戻り値	なし
備考	

MSP_ClearIMask ()

関数名	MSP_ClearIMask アイ
プロトタイプ	void MSP_ClearIMask(HANDLE hHND、WORD Imask)
機能	割り込み関係の Mask のクリアを行う
引数	HANDLE hHND : オープンしたボードのハンドル値 WORD Imask : D15 ~ D3 空き D2 M_INT2 1 : 転送終了割り込みマスクをクリアする D1 M_INT1 1 : 転送終了割り込みマスクをクリアする D0 M_END 1 : 転送終了割り込みマスクをクリアする
戻り値	なし
備考	

MSP_ClearINT ()

関数名	MSP_ClearINT
プロトタイプ	void MSP_ClearINT(HANDLE hHND、WORD Imask)
機能	割り込み要因のクリアを行う
引数	HANDLE hHND : オープンしたボードのハンドル値 WORD Imask : D15 ~ D3 空き D2 M_INT2 1 : 転送終了割り込みをクリアする D1 M_INT1 1 : 転送終了割り込みをクリアする D0 M_END 1 : 転送終了割り込みをクリアする
戻り値	なし
備考	

MSP_GetMEMSize ()

関数名	MSP_GetMEMSize
プロトタイプ	DWORD MSP_GetMEMSize(HANDLE hHND)
機能	基板搭載メモリの全サイズを得る
引数	HANDLE hHND : オープンしたボードのハンドル値
戻り値	DWORD D31 ~ D0 : メガバイトを単位とする値 【例】1GBytes の場合は、0x00000400 8GBytes の場合は、0x00002000
備考	

MSP_SetCommand ()

関数名	MSP_SetCommand
プロトタイプ	void MSP_SetCommand(HANDLE hHND、WORD Cmnd)
機能	収録の許可、禁止などを命令する
引数	HANDLE hHND : オープンしたボードのハンドル値 WORD Cmnd : D15 ~ D8 空き D7 ~ D0 CMND 0x00 : 収録禁止 0x80 : 収録許可
戻り値	なし
備考	注意) DIMM データを読み出す場合は、「収録禁止」状態で行うこと

MSP_SetStoreCndt ()

関数名	MSP_SetStoreCndt
プロトタイプ	void MSP_SetStoreCndt(HANDLE hHND、 _int64 Adrs、 _int64 Size)
機能	収録時のアドレス、収録サイズを設定する
引数	HANDLE hHND : オープンしたボードのハンドル値 _int64 Adrs : 収録する SDRAM の先頭アドレス D63 ~ D0 8Gbytes 以内で設定する _int64 Size : 収録するデータサイズ D63 ~ D0 8Gbytes 以内で設定する int Mode : 外部入力データ幅を設定する。 D7 ~ D0 8、16、32 (8 は予備)
戻り値	なし
備考	注意) DIMM データを読み出す場合は、「収録禁止」状態で行うこと

MSP_SetINPUT_Mode ()

関数名	MSP_SetINPUT_Mode
プロトタイプ	void MSP_SetINPUT_Mode (HANDLE hHND、 int Mode)
機能	収録時のモードを設定する
引数	HANDLE hHND : オープンしたボードのハンドル値 int Mode : 外部入力データ幅を設定する。 D7~D0 (8:8ビット、未対応)、16:16ビット、32:32ビット
戻り値	なし
備考	

MSP_BlockRead ()

関数名	MSP_BlockRead
プロトタイプ	int MSP_BlockRead (HANDLE hHND、 DWORD *Buff、 DWORD Adrs、 DWORD Size)
機能	メモリの指定アドレスから指定サイズのデータを読み出す
引数	HANDLE hHND : オープンしたボードのハンドル値 DWORD *Buff : 読み出した Data の格納アドレスポインタ DWORD Adrs : D31 ~ D0 32 バイトアドレスを単位とする、DIMM メモリ上の 転送開始アドレス 【例】1Gbytes 目を指定する場合 0x02000000 4Gbytes 目を指定する場合 0x08000000 DWORD Size : D31 ~ D0 バイトを単位とする、転送サイズ
戻り値	正常終了の場合 0、異常終了の場合 2
備考	注意) この関数は、「収録禁止」状態で実行すること

MSP_BlockWrite ()

関数名	MSP_BlockWrite
プロトタイプ	int MSP_BlockWrite (HANDLE hHND、 DWORD *Buff、 DWORD Adrs、 DWORD Size)
機能	メモリの指定アドレスから指定サイズのデータを DIMM に書き込む
引数	HANDLE hHND : オープンしたボードのハンドル値 DWORD *Buff : 書き込むべき Data の格納されているアドレスポインタ DWORD Adrs : D31 ~ D0 32 バイトアドレスを単位とする、DIMM メモリ上の 転送開始アドレス 【例】1Gbytes 目を指定する場合 0x02000000 4Gbytes 目を指定する場合 0x08000000 DWORD Size : D31 ~ D0 バイトを単位とする、転送サイズ
戻り値	正常終了の場合 0、異常終了の場合 2
備考	注意) この関数は、「収録禁止」状態で実行すること