

第4章. ソフトウェア

4-1. (ソフトウェアの)インストール

製品添付のソフトウェアは3.5インチ(1.44MB)フロッピーまたはCDに圧縮された形で格納されており、インストーラ“Setup.EXE”の実行により展開されます。

なお内容については充実・改良の目的で後日、追加・変更も有り得ます。

また重要な変更については同メディア内のドキュメントファイルに記すこととします。

操作手順

インストール元：Dドライブ

インストール先：Cドライブ(HDD) の場合で例示。

(1) WINDOWS付属のエクスプローラで、
D:¥INSTALL¥PCI¥DIO¥MDO282 を開く。

(2) “Setup.EXE”を実行(ダブルクリック)する。

当操作によりMDO-282PCI関連プログラムが次ページ：図4-1Aに示すロケーションに展開・インストールされます。

図4-1A. インストール後のディレクトリ

本図は原形です。

充実・改良の目的で後日、追加・変更も有り得ます。

パソコンCドライブ

```

¥
|
MSCIENCE
|
|   Hnd__2K
|

```

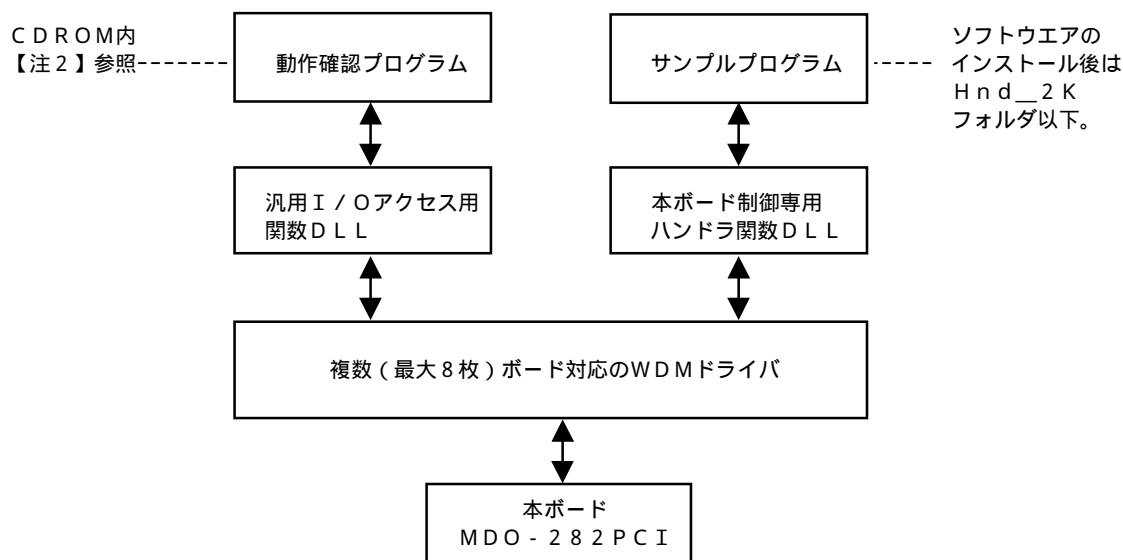
本ボード専用ハンドラ関数（第5章参照）：WINDOWS2000/XP/98/ME用
Mdo282

```

| - Dll : ハンドラ関数DLL
| - Vb5 : Visual-Basic (5.0)用サンプル
| - Vc5 : Visual-C (5.0)用サンプル
| - Bc5 : Borland-C (5.0)用サンプル
| - Vc5_cpp : VC++ (5.0)用サンプル
| - Delphi3 : Delphi (3.0)用サンプル等

```

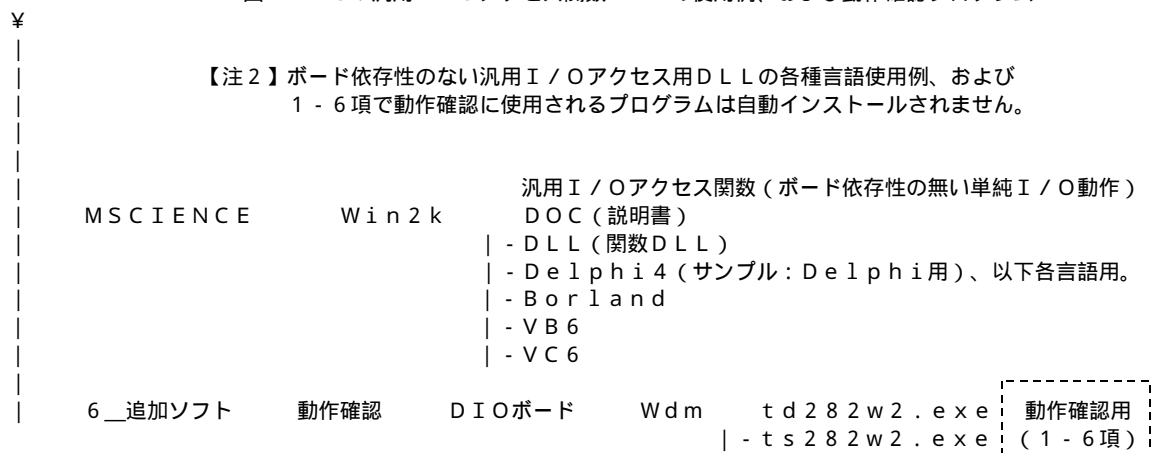
図4-1B. ソフトウェア構成



【注1】上図中の両DLLとWDMドライバはボードインストールのとき同時に自動インストールされる。（SYSTEM32フォルダ）

CDROM

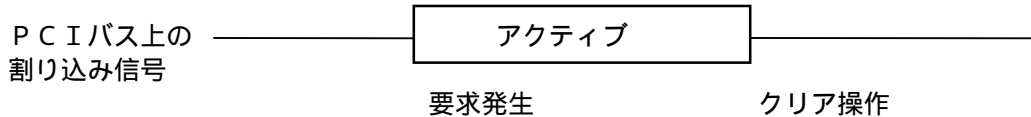
図4-1C. 汎用I/Oアクセス関数DLLの使用例、および動作確認プログラム



4-2．割り込みについて

PCIバス上の割り込み信号は、これを検知したソフトウェアからクリア操作を行うまでアクティブ状態を（要求元側が）維持する“**レベル動作**”です。この仕組みでは複数のデバイスが1本の割り込みリソースを共有することもできます。

図4-2．



要注意

当社製を始め、多くのISAバスボードの割り込みは要求元がパルス状の単発信号を発信する“**エッジ動作**”ですから割り込み要求のアクティブ状態は自動解消されるのですが、PCIバス上の“**レベル動作**”ではプログラム開発中などの事情で適切なクリア操作が行われなかった場合のハングアップ等、非常事態解消のためのハードウェアリセット（電源OFF）を余儀なくされることも考えられます。このような場合はハードディスクのクラッシュ等の大きな損害が発生する恐れもあります。

そこで本ボード上のROMに書き込まれているデフォルト（初期）のコンフィギュレーション情報では（プラグアンドプレイの動作時に）割り込みリソースを要求しません。もし要求したときに空きが無く拒否されるとI/Oアドレスの割り当ても受けられず認識不能状態になる恐れがあるからです。割り込みを利用するときはリソースに空きがあることを確認してから添付のコンフィギュレーション・ユーティリティで（割り込みリソースを要求するように）修正してください。【1-5項参照】

但し添付の専用ハンドラ関数DLL、および汎用I/Oアクセス関数DLLは本ボードのインストールと同時に自動インストールされるWDMドライバを使用しており、割り込みは使用できません。が、専用ハンドラ関数【第5章】は本ボードの自動サンプリングによる取得データをパソコン側のバッファに自動高速転送するスレッドを生成・実行する構造ですから、割り込みを使用せずともバックグラウンド運転が可能です。

（追伸） 一部のパソコンは標準状態で割り込みリソースに空きが無いものがあります。

第5章. WINDOWS ハンドラ

本ボード：MDI-281PCI (IBM PC / AT 互換機用) を VC、VB 等、メジャーな言語で簡単に使用することのできる WINDOWS 2000・XP、98、ME 用ハンドラ関数 DLL (+ ドライバ) です。

本ボードの基本機能が関数化されており、ユーザは御自身の記述するメインルーチンの中から呼び出して使用することができます。すなわち必要なパラメータ (動作条件) を変数に代入して本ハンドラを呼び出すだけで自動サンプリングからパソコン本体メモリへのデータ転送まで、ポーリング・ブロック転送等により高速実行されます。

5-1. システム構成・ソフトウェア構成

パソコン本体 : IBM PC / AT 互換機 (含む 98 NX 機)

拡張メモリ量 : 16MB 以上

OS / コパイラ : WINDOWS 98、ME、または 2000、XP / 32 ビット専用。

添付サンプル : Visual-C, C++ (5.0)

Visual-Basic (5.0)

Borland-C (5.0), Delphi (3.0), C++Builder

供給メディア : 本ボード添付のサンプルディスク内。

対応ボード : MDO-282PCI (最大 8 枚 / マスタスレーブ動作)

搭載 FIFO メモリ容量は全ボード同一であること。

サンプリング(1) : 非サイクルモードのとき、

FIFO バッファメモリ容量までは本ボードの最高速度で出力が可能、

なおボード側への実用的に期待できる最高データ書き込み速度は、

900 K 語 / 秒 (Pentium4/1GHz)、200 K 語 / 秒 (Pentium2/400MHz)。

上記速度は当ハンドラ DLL が本ボードへの補給データ書き込みに専念したときに期待できる PCI バス上の実用的な最高データ流速です。したがって平行動作するプログラムがあると書き込みがリアルタイムに追いつく最高サンプリング速度は低下します。また動作ボード数にも比例して低下します。

但し書き込み速度が低下しても、各ボードの搭載する FIFO メモリ容量まではボード自体の最高サンプリング速度でデータを有効に更新出力することができます。【注】速度保証はできませんので余裕をもって御利用ください。

サンプリング(2) : サイクルモードのときの最高出力速度はビット幅に比例し、

16 ビット幅のとき 4 MHz、

32 ビット幅のとき 2 MHz、

48 ビット幅のとき 1.33 MHz

64 ビット幅のとき 1.00 MHz

出力データ点数 : 非サイクルモードのときは拡張メモリ空容量、

サイクルモードのときは FIFO メモリ搭載容量。

割り込み : 不使用。

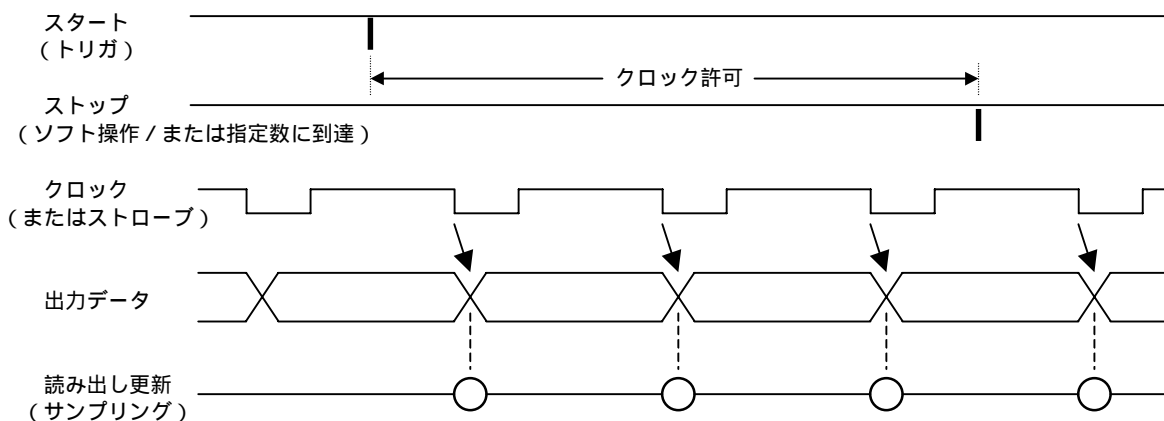
その他のボード : I/O アドレスが重複しない限り同時に使用可能。

5 - 2 . サンプリグの様子とデータバッファ構造

ハンドラの使用方法はとても簡単です。 具体的には5 - 4 項以下で詳述しますが、要は出力しようとするパラレルデータ長 / サンプリグ点数 / クロック値 / トリガ (スタート) 条件等のパラメータをセットして各関数を呼ぶだけで、出力データは本ハンドラの管理するバッファから自動出力されます。 本ハンドラでは大別して以下に記す2 形態のサンプリグ動作が可能です。

- (1) 直接入力動作はメインルーチン上任意のプロセスで現在入力値を取得します。
- (2) 連続サンプリグ動作は指定パラレル出力に対し指定時間々隔 (クロック周期) で指定回数だけ書き込み更新します。
動作開始となるトリガはソフト (即スタート) 、または外部 (デジタルTTLエッジ) から選択できます。 クロック源も内蔵の20MHz、または外部TTL入力源から選択できます。 いずれのクロック源も (本ボードの実行可能な速度以内で) 任意の整数値で分周して利用します。 外部クロックを分周比 = 1 / 1 で使用すると、外部のイベントに同期した出力動作となります。
FIFOバッファメモリ容量の範囲内で1周期分だけのデータを繰り返し出力するサイクルモードと、FIFOバッファメモリ容量を超えるデータをボードに逐次補給しながら出力する非サイクルモードがあります。
- (3) 複数ボードを同期運転する場合、本ハンドラの管理する出力データバッファは各ボードごとに複数用意されます。

図5 - 2 A . 連続サンプリグFIFOバッファ出力動作例



スタート： ソフトトリガ (即スタート) 、または外部トリガ入力の指定エッジにより指定クロック (例えば外部TTLクロック入力の指定エッジ) が有効となります。

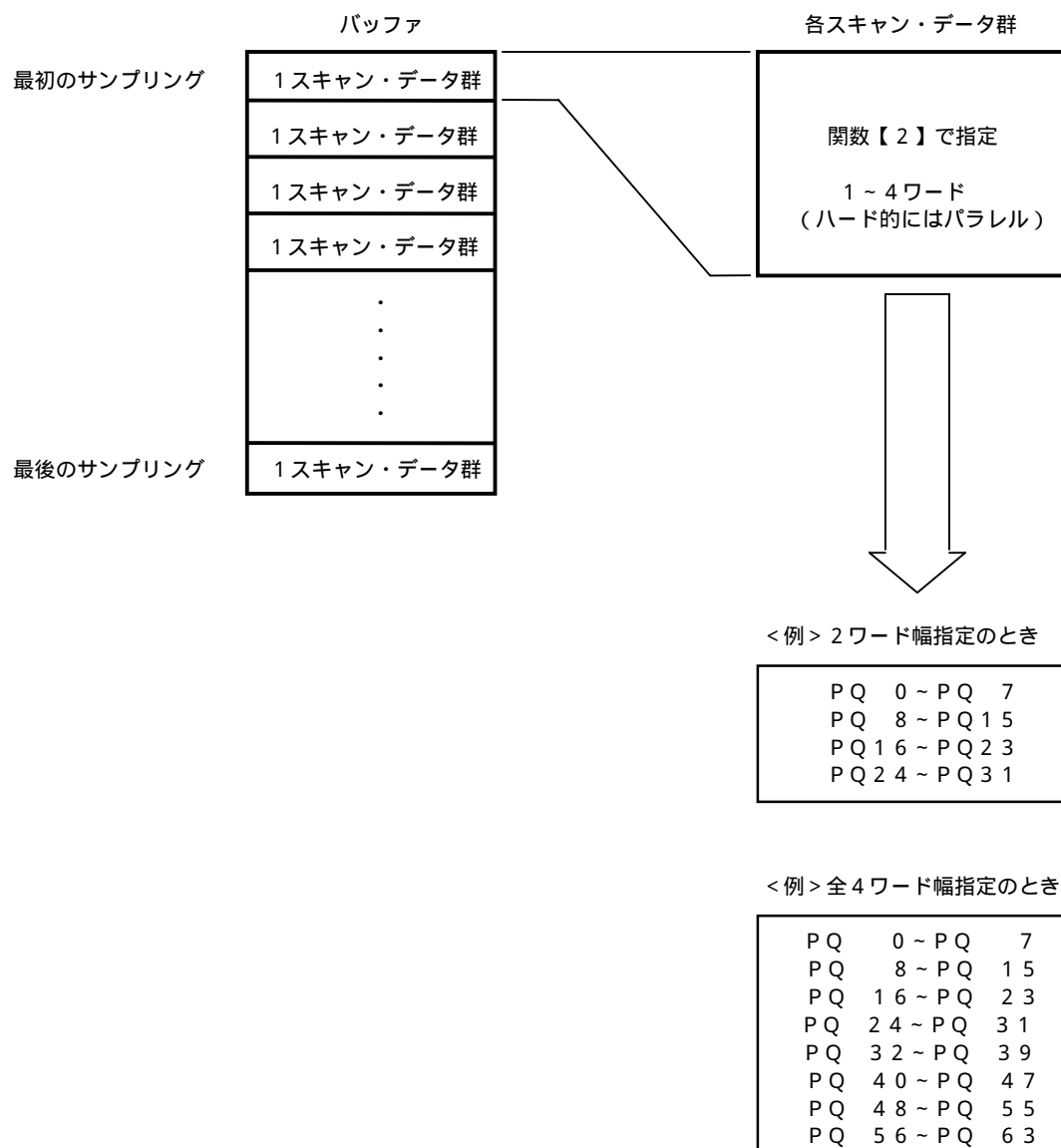
読み出し： 指定ビット幅 (TTL出力) データは、指定クロックの指定エッジに同期してFIFOバッファメモリから読み出し更新されます。 この動作は指定回数に達するか、ソフト上の停止操作まで続きます。

動作タイミングの詳細は3 - 1 項を参照。

出力データバッファと格納の様子

出力データは本ハンドラの管理するバッファからボード上のFIFOバッファに転送されます。
 この（パソコン側メモリ上にある）バッファ内のデータ配置は（局所的には）本ボード上の
 FIFOメモリ内と同一イメージで、全データが1個のバッファに格納されます。
 また複数ボードの同期運転の場合は、同一構造のバッファが各ボードごとに用意されます。

図5-2B. 全データが1個のバッファに入る



5-3. 使用準備

ボード上の設定・入力接続は標準設定【1-2項】【3-14項】とします。

このとき、ボード番号設定スイッチSW-BNはマスタを【0】に、以下、スレーブボードを各【1】～【7】に順次設定します。

続いて第4章（4-1項）に従い、本ボード関連ソフトをインストールします。

本機のハンドラ関数DLLとWDMドライバはボードインストールのとき同時にインストール済みなので、以後は各言語用サンプル（¥MSCIENCE¥Hnd_2K¥Mdio282 フォルダ）を使用できます。

WINDOWS 2000・XPの場合 / Administrator レベルで行う /

- (1) 当社製PCIボード（複数可能）に共通使用できるWINDOWS 2000用のWDMドライバ“DMS_PCI.SYS”はボードインストール時に（ボードインストールディスクから）自動的にインストールされます。

インストール先：¥WINDOWS¥SYSTEM32¥DRIVERS フォルダ

このWDMドライバは当社製の全PCIボード（複数可）から共通に使用できる汎用品です。すなわち各個別PCIボード専用の関数DLLを用意すれば、当WDMドライバ1本で当社製の全PCIボードを動作させることができます。

- (2) 本関数DLL（H282_2K.dll）もボードインストール時に所定の¥WINDOWS¥SYSTEM32にコピーされているので、即サンプル（¥MSCIENCE¥Hnd_2K¥Mdo282 以下）を使用できます。

WINDOWS 98、MEの場合

当社提供のボードインストール環境は通常、WINDOWS 95との互換性をとるため、単一ボード用ハンドラをVXD型ドライバで使用する前提で設定されていますが、

本ハンドラDLLはWINDOWS 2000と同一のWDM型ドライバを使用するため、WINDOWS 2000と同一のインストール作業【1-5項】が必要です。もし既にWINDOWS 95・98の方法でインストール済みの場合は以下の要領で一旦削除してから、あらためてWINDOWS 2000と同一の方法でインストールしてください。

- (1) WINDOWSのデバイスマネージャで表示される本ボードの関連付けを削除する。
- (2) ¥Windows¥Inf¥Others フォルダに以下の定義ファイルがあれば、これらを削除する
Micro Science ms_pci.inf、Micro Science Co.,Ltd.DMS_PCI.INF

なお ¥Windows¥Inf フォルダは隠しフォルダとなっているので、エクスプローラの
<表示> <フォルダオプション> <表示>にある“詳細設定”内の
“ファイルとフォルダの設定”で“全てのファイルを表示する”に変更して作業する。

- (3) WINDOWS 2000 / XPと同一の方法・手順で本ボードをインストールする。
- (4) 念のため、デバイスマネージャで本ボードの登録を確認します。
- (5) 本関数DLL（H282_2K.dll）もボードインストール時に所定の¥WINDOWS¥SYSTEM32にコピーされているので、即サンプル（¥MSCIENCE¥Hnd_2K¥Mdo282 以下）を使用できます。

5-4 . 関数仕様・エラーコード

御自身の記述したメインプログラムから本ハンドラDLL（+ドライバ）を使用します。

テストには付属のサンプルプログラムを御利用ください。 前5-3項に従ってインストールしておきます。 本ボードの操作は通常以下の手順となります。 具体的なコーディングについてはサンプル・ソースを御覧ください。

- (1) 初期化 : 【1】
- (2) サンプルング条件の設定 : 【2】～【6】 出力データ幅 / トリガ / クロック / 動作モード
- (3) 出力データを用意 : 【7】 or 【8】
- (4) スタートまたはトリガ待ち : 【9】 即スタート / 外部トリガ入力待ち
- (5) ステータス評価 : 【10】 連続サンプルング進捗状況
- (6) 以下・途中は任意
- (7) サンプルング中止 : 【15】
- (8) ソフトのリセット : 【16】 このあと、再度（2） or （3）に戻るときは必要。
- (9) ハンドラ終了 : 【12】 後処理

表 5-4 A . 制御関数一覧 （注）正確には各関数名中、M d o の後にボード番号が入る。

関数名	機能・内容	引数（パラメータ）等
【1】 Mdo_Open_MdoSys	ボード、および本ハンドラの初期化	
【2】 Mdo_Set_SampMode	動作モード設定	
【3】 Mdo_Set_TransMode	データ転送モード、イベント設定	データ転送元、方法
【4】 Mdo_Set_Trigger	トリガ関連設定	トリガ源、レベル、モード
【5】 Mdo_Set_ExclK	オプション、外部クロック源の設定	クロック源の周波数値
【6】 Mdo_Set_Clock	サンプルング・クロックの設定	クロック源、周期値、単位
【7】 Mdo_Write_DllData	DLL内バッファにデータ書き込み	サンプルング点数
【8】 Mdo_Write_DirectFifo	本ボードに直接データ書き込み	（マニュアルモード）
【9】 Mdo_Start_Samp	サンプルング開始（トリガ待 or 即）	繰り返し回数
【10】 Mdo_Get_Status	ステータス取得	サンプルング進捗状況など
【11】 Mdo_Immediate_Update	データ更新（1回スキャン）	
【12】 Mdo_Close_MdoSys	本ハンドラの終了	
	《以上が基本、以下は補助的》	
【13】 Mdo_Out_Aux	補助デジタル（ラッチ）出力	出力データ
【14】 Mdo_Inp_Aux	補助デジタル（現在値）入力	入力データ
【15】 Mdo_Stop_Samp	サンプルング動作の（強制）中止	
【16】 Mdo_Reset	ソフトのリセット	補助出力は現状を保持
【17】 Mdo_Clear_Flags	ボードのフラグクリア	ビット指定
【18】 Mdo_Plus_Message	オプション的なメッセージ発信設定	任意
【19】 Mdo_Get_Libver	バージョン情報取得	

以下に各関数の仕様・詳細を記します。

【1】初期化

書式	int Mdo282_Open_MdoSys (HWND Owner, int num_board, int *fifo_size)
引数	<p>Owner : ウィンドウハンドル / 割り込み発生、サンプリング終了、 / / データロス発生等のメッセージングに使用。 /</p> <p>num_board : 使用する本ボード数。 (1 ~ 8)</p> <p>fifo_size : 検出した (マスタボード上の) F I F O メモリ容量</p> <p>【注】 メガ F I F O 搭載の場合はスイッチで設定された < HALF-FULL ポインタ > の 2 倍値を示す。 なお、スイッチ設定値は 0 (5 1 2 語) ~ 5 (1 6 K 語) の 範囲に限る。</p>
戻り値	<p>正常終了時 : ボードの I D / 2 C H (for M D O - 2 8 2 P C I)</p> <p>エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)</p>
機能・動作	<p>プラグアンドプレイで設定した I / O アドレス、割り込み番号を本ハンドラが自動認識すると共に、 本ボードのリセット、ハンドラ内部の参照テーブルやデータバッファを初期化する。</p> <p>【注】 : ここで得るボード I D は P C I バスで定義される Subsystem I D とは別物。 / 3 - 4 項参照。</p>

【2】動作モードの設定

書式	int Mdo282_Set_SampMode (int board_num, int mode1, int mode2, int width)
引数	<p>mode1 : / 0 : マニュアル操作、1 : 非サイクル連続サンプリング、2 : サイクル連続サンプリング</p> <p>mode2 : 連続サンプリングのモード / 0 : 有限サンプリング、1 : 無限サンプリング、</p> <p>width : サンプリングを実行する出力ビット幅を 1 6 ビット単位のポート数で指定する。 最大 4 ポート (64bit) 。</p>
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	<p>mode1 でマニュアル操作を選択のときは関数【11】マニュアル 1 回サンプリングスキャンのみ可能。 有限サンプリングはデータ転送関数【7】で設定する回数を実行したら自動停止、 サイクル連続サンプリングの無限サンプリングはスタートすると強制停止関数【15】まで無限に実行する。 非サイクル連続サンプリングの無限サンプリングは次項【7】で設定するマニュアルモードのみ有効。</p>

ボード番号 : 1 枚だけのボードを使用する場合はボード上の番号設定スイッチ S W - B N を 0 に設定すること。
複数ボードを使用する場合はマスタ機の番号設定スイッチ S W - B N を 0 とし、以下、スレーブ機の
同スイッチを順次 1 ~ 7 に設定する。

【3】F I F O へのデータ転送モードの設定

書式	int Mdo282_Set_TransMode (int trs_trig, int buf_area, int intr_sw)
引数	<p>trs_trig : データ転送の参照フラグ / 0 : F U L L 解消、 1 : H A L F - F U L L、</p> <p>buf_area : データ転送元 / 0 : D L L の管理するバッファ、1 : ユーザプログラム内バッファ (自動モード) (マニュアルモード)</p> <p>intr_sw : 割り込みイベント発生要因指定 / (当ハンドラではダミー)</p>
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	<p>trs_trig で < 0 : F U L L 解消 > を指定するとデータを 1 個々連続的に転送する動作、 < 1 : H A L F - F U L L > は F I F O 容量の半分単位で間欠的に転送する動作、</p> <p>buf_area で D L L の管理するバッファとした場合は当ハンドラが管理する D L L バッファから ボード上の F I F O バッファに出力データを書き込む自動モード、ユーザプログラム側からはステータ ス取得関数でサンプリング済みのデータ点数を認識できる。</p> <p>ユーザプログラム内バッファとした場合はユーザプログラム自体がステータス取得関数でボード上の F I F O メモリ状態を直接監視して、専用の関数 M d o 2 8 2 _ W r i t e _ D i r e c t F i f o で ユーザプログラム側バッファからボード上の F I F O バッファに出力データを書き込むマニュアル モード。</p> <p>【注 1】 : いずれのモードでもデータ転送時に参照するフラグを trs_trig で指定する。 【注 2】 : 当ハンドラは割り込みをサポートしていないので intr_sw はダミー。</p>

【4】トリガ関連の設定

書式	int Mdo282_Set_Trigger(int trg_mode, int trg_pol)
引数	trg_mode : トリガ動作モード / 0 : 即トリガ (=ソフトトリガ=即スタート)、 1 : ポストトリガ (外部デジタル入力待ちスタート) trg_pol : トリガ極性 / 0 : 負エッジ、1 : 正エッジ、2 : 負レベル、3 : 正レベル
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	トリガ動作モード (源)、極性の設定。 なおトリガ動作モードをポストトリガ (外部デジタル入力)、極性をレベルに指定したときは帯域サンプリングとなる。 / (3 - 1 項参照) 複数ボード使用時の外部デジタルトリガ検出対象はマスタボード。

【5】外部クロック源周波数値の設定

書式	int Mdo282_Set_Exclk (int exclk_freq)
引数	exclk_freq : オプション、または外部クロック源の周波数値 (Hz 単位)
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	外部クロック源 (CLK - IN) を使用するとき、その周波数値を設定する。 後記の関数【6】で分周比によるサンプリングクロック指定、またはボードに標準搭載の内部クロック源を使用するときは設定の必要がない。

【6】サンプリング・クロック値の設定

書式	int Mdo282_Set_Clock(int clk_source, int set_mode, int *time_unit, int *clk_period)
引数	clk_source : サンプリング・クロック源選択。 / 0 : 内部クロック源 0 (20MHz)、1 : 内部クロック源 1 (未使用) 2 : 内部クロック源 2 (未使用)、3 : 内部クロック源 3 (未使用) 4 : 外部クロック源 (有効極性 =)、5 : 外部クロック源 (有効極性 =) set_mode : クロック値の指定方法 / 0 : クロック周期の値、1 : 分周比 time_unit : クロック周期の単位 / 0 : s、1 : ms、2 : μ s、3 : ns clk_period : クロック周期の値、または分周比
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	サンプリングクロック値を設定する。 なお設定できないクロック周期の値を指定すると設定可能な長い方の近似値が設定される。 また、外部クロック源を指定し、クロック周期の値で設定するときは前記の関数【5】でクロック源周波数値を定義しておく。

クロックによる連続サンプリング・データ読み込み方法

本ハンドラは関数【2】～【6】でサンプリング条件の設定後、関数【7】以降で実際のサンプリング動作を実行します。

このサンプリング動作実行時に、当ハンドラが管理するバッファからボード上のFIFOバッファメモリにデータを転送する自動モード、およびユーザプログラムが自己の管理するバッファから直接ボード上のFIFOバッファメモリにデータを書き込むマニュアルモードがあり、関数【3】“FIFOへの出力データ転送モード”の第2引数で指定します。

自動モード

: 当ハンドラDLL側でバッファを確保、ここからボード側のFIFOバッファに出力すべきデータを自動転送する。ユーザプログラムは別タスクをマルチスレッドで併行することができる。

DLLの管理するバッファからボード側FIFOバッファへのデータ転送はスタート時に当ハンドラの生成するスレッドがFIFOメモリの充満状態を示す【Half-Full】または【FULL】フラグを監視して

<FIFOメモリ容量の半分>または<1スキャン分>単位で行う。

前者はブロック転送命令を使用するので速く、

900 K 語 / 秒 (Pentium4/1GHz)、200 K 語 / 秒 (Pentium2/400MHz) の速度が期待できる。後者は1回サンプリングごとに評価・実行すべき処理があるような場合向きで、やや低速。

《メッセージ》

本ボードのサンプリング速度にプログラム側が追いつかなくなるか、本ボードが実行不可能な速いクロックを与えた場合に発生するサンプリングエラー（データロスト）、または指定数サンプリング終了がある。

マニュアルモード

: ユーザプログラム側にバッファを確保して、ここから出力データをボード側のFIFOバッファに向けて【連続サンプリングクロック】 / 【1回サンプリングスキャン終了】 / 【Half-Full】 / 【Not-Empty】 / 【トリガ発生】 / 【外部割り込み】 / 等の割り込みイベントをポーリングで検出し、<FIFOメモリ容量の半分>または<1スキャン分>単位で行う。

《メッセージ》

前記の各割り込みイベントによる。サンプリングエラー、指定サンプリング終了などはユーザプログラム中でステータスをチェックする。

《注意》

実際には割り込みは使用されず、スレッドからメッセージを発する。

重 要

本ハンドラを構成する関数は一定手順、または特定の組み合わせでのみ有効に動作します。 手順については本項先頭に記しましたが、組み合わせの点で注意すべきことは【3】で設定するサンプリングモードと【9】【10】のデータ転送関数です。 以下に連続サンプリングの典型的な例を示します。

< 例 1 > “自動モード”で“スタート操作”を実行すると自動的に“当ハンドラDLLの管理するバッファ”から本ボード側のFIFOバッファにデータを書き込む。

このとき、“データ転送の参照フラグ”を《HALF-FULL》に設定してある場合は“FIFOメモリ容量”の半分単位でブロック転送、また《FULL 解消》に設定の場合は“1スキャン分”単位で転送する。 以上はバックグラウンドで自動実行され、指定データ点数に達すると自動停止する。

ユーザプログラムはスタートに先立ち関数【9】により“出力すべきデータ”を“当ハンドラDLLの管理するバッファ”に書き込んでおく。

< 例 2 > 当ハンドラDLLの管理するバッファを“エンドレス・リング状”に設定し、例1と同様に操作する。 この場合“当ハンドラDLLの管理するバッファ”の末尾と先頭が連結された構造となり、1廻り以後は再転送される動作となる。

ユーザプログラム側からは“サンプリング済みデータ点数”の値を当バッファのポインタ（先頭 = 0）として利用する。 連続サンプリングが“ストップ操作”まで無限に実行されるので、際限なく新出力データが発生するような場合に当バッファを逐次書き変えるような使い方ができる。

< 例 3 > “マニュアルモード”で“スタート操作”を実行すると、本ボードが連続サンプリングを開始しても当ハンドラは何もしない。 ユーザプログラムは自力でステータス関数をポーリングして“ボードの生ステータス”から《HALF-FULL》、または《FULL 解消》を検出してデータの書き込みタイミングを知り、関数【10】でFIFOに直接書き込む。

このとき、“データ転送の参照フラグ”を《HALF-FULL》に設定してある場合は“FIFOメモリ容量”の半分単位でブロック転送、また《FULL 解消》に設定の場合は“1スキャン分”単位で書き込む。

補足説明

キーワード	説明、関連関数 / 【 】内の数字は関連する関数項目番号
“自動モード”	【3】連続サンプリング動作モード、 および割り込みイベントの設定関数で選択・指定する
“マニュアルモード”	
“データ転送の参照フラグ”	
“スタート操作”	【7】連続サンプリング・スタート関数
“当ハンドラDLLの管理するバッファ”	ユーザプログラムからは【9】Mdo282_Write_DllDataで書く、容量は【2】指定ビット幅（ワード）×【7】指定データ点数
“FIFOメモリ容量”	【1】初期化のとき自動検出
“1スキャン分”	【2】指定ビット幅（ワード数）を1回サンプリング分
“サンプリング済みデータ点数”	【8】ステータスから得る
“エンドレス・リング状”	【2】バッファ末尾の次が先頭に戻る。
“ストップ操作”	【15】連続サンプリング動作の強制停止関数の実行

【7】ユーザプログラム側からDLL管理バッファにデータを転送 / (ボードごと)

書式	int Mdo282_Write_DLLData(int board_no, int no_samp, int no_data, WORD *bufptr, int bufsize)
引数	board_no : 対象ボード番号。 / 0 ~ 7 (複数ボード使用のときはボードごとに行う) no_samp : トリガ後のサンプリング点数 (サイクル動作のときはサイクル数) 本パラメータと関数【2】で指定されるビット幅ワード数の積がDLLの管理する バッファサイズとなる。 no_data : DLL管理バッファに書き込むデータ点数 bufptr : 転送元データバッファのポインタ bufsize : 転送元データバッファの大きさ (バイト指定)
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	出力すべき全データ、または初期データ (後で追加する場合) を当ハンドラ側に渡す。 出力すべきデータがユーザプログラム側のデータバッファに格納されているとき、 任意の部分をDLLの管理するバッファに転送する。 【注】 当方法は関数【3】データ転送モード設定で自動モードを指定したときだけ有効。

【8】ユーザプログラム側バッファからFIFOに直接、データを書き込む / (ボードごと)

書式	int Mdo282_Write_DirectFifo (int board_no, WORD *bufptr, int bufsize)
引数	board_no : 対象ボード番号。 / 0 ~ 7 (複数ボード使用のときはボードごとに行う) bufptr : 転送元データバッファのポインタ bufsize : 転送元データバッファの大きさ (バイト指定)
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	ユーザプログラム側のデータバッファから直接、本ボード上のFIFOメモリにデータを転送する。 一般的に当関数は割り込みイベント (関数【3】参照)、またはFIFOメモリ状態をポーリングして 実行タイミングを得る。 サンプリング動作モード設定【3】でデータ転送の参照フラグに FULL 解消を指定したときは1回サンプリングスキャン分のデータを転送する。 また、 Half-Full を指定したときはFIFOメモリ容量の半分を転送する。 【注】 当方法はデータ転送モード設定【3】でマニュアルモードを指定したときだけ有効。

【9】サンプリング・スタート

書式	int Mdo282_Start_Samp (int no_cycle)
引数	no_cycle : サイクル動作のときの繰り返し回数 (非サイクル指定のときは無視される)
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	関数【1】 ~ 【8】で設定した条件で連続・自動サンプリングを実行する。 関数【4】トリガ設定でトリガ動作モードを《即トリガ》としてあるときは即スタート、また 《ポストトリガ》が選択されているときはトリガ待ちスタートとなる。

<注> サンプリング点数はタイミング (クロック) 到来ごとに更新出力する1スキャン分のデータブロックを数えるものです。 例えば関数【2】で出力ビット幅を3ポート (48ビット) に指定した場合の1点は3ワード (48ビット) になります。 / 図5 - 2 B 参照 /

【10】ステータス取得 / (マスタボード)

書式	int Mdo282_Get_Status (int *sampled, int status[])
引数	sampled : サンプルング済み点数 (本DLLの管理するバッファ内の有効点数) status[0] : ボードのステータス1 = 基本ステータス (3-12項) status[1] : ボードのステータス2 = 追加ステータス (3-12項)
戻り値	正常終了時 : = 0 : 連続サンプルング実行中、 = 1 : 停止中。 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	当該時点でのサンプルング済み点数 (DLL管理バッファ内 / ソフト認識)、および マスタボードのステータス生データ (FIFOメモリ状態等) を得る。 マニュアルモードのときはステータス生データのみ有効。

【11】マニュアル1回サンプルングスキャン = 即出力更新 / (全ボードについて)

書式	int Mdo282_Immediate_Update (WORD *bufptr, int bufsize)
引数	bufptr : データバッファのポインタ bufsize : データバッファの大きさ (バイト指定)
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	全ボードに対して各1回だけサンプルングを実行する。 出力データは全て、ここで指定した1個の専用バッファに用意してから実行する。 複数ボード使用時のデータ並び順は先頭がボード番号0 (マスタ機) で、 以下、ボード番号1 ~ 7 (スレーブ各機) のデータが順に続く。 【注】本操作は関数【2】のmode1でマニュアル操作を選択しているときだけ有効。

【12】本ハンドラの終了

書式	int Mdo282_Close_MdoSys (void)
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	本ハンドラの終了、確保したメモリ領域の開放等を行う。

【13】補助デジタル (ラッチ) 出力Q0 / (ボードごとに実行)

書式	int Mdo282_Out_Aux (int board_no, int out_data)
引数	board_no : 対象ボード番号 out_data : 補助1ビット・デジタル (ラッチ) 出力Q0データ。
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値 (エラーコード表 5 - 4 B 参照)
機能・動作	各ボードごとの補助1ビット (ラッチ) 出力Q0データの更新。 / 3 - 15項。 当出力ポートは電源投入 (ハードウェアリセット) 時 : 0H となるが、ボードの制御部リセット (ソフトリセット) では変化しない。

【14】補助デジタル（現在値）入力 / （ボードごとに実行）

書式	int Mdo282_Inp_Aux (int board_no, int *gpin_data)
引数	board_no : 対象ボード番号 gpin_data : 補助3ビット・デジタル（現在値）入力データ
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値（エラーコード表5 - 4 B 参照）
機能・動作	各ボードごとの補助3ビット・デジタル（現在値）入力データ読み込み。 / 3 - 15 項参照。

【15】サンプリング動作の強制停止

書式	int Mdo282_Stop_Samp (void)
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値（エラーコード表5 - 4 参照）
機能・動作	サンプリングを強制的に中止する。

【16】ソフト的リセット操作 / （全ボードについて共通に実行する）

書式	int Mdo282_Reset (void)
引数	なし
戻り値	正常終了時 : = 0 エラー時 : エラーコード / 負の値（エラーコード表5 - 4 B 参照）
機能・動作	サンプリング動作が終了（含強制停止）したときの出力は当時の値を保持しているので、これを = 0 に戻すと同時に F I F O ほかの各フラグをクリアする。（なお補助出力は現状を保持する。） 本関数は次記の関数【17】の機能を含む。

【17】各フラグのクリア / （全ボードについて共通に実行する）

書式	int Mdo282_Clear_Flags (int c__data1, int c__data2)
引数	c__data1 : クリアビット指定データ。（基本ステータス用） / 3 - 12 項参照 / c__data2 : クリアビット指定データ。（追加ステータス用） / 3 - 12 項参照 /
戻り値	正常終了時 : = 0、 エラー時 : エラーコード / 負の値（エラーコード表5 - 4 B 参照）
機能・動作	本ボード上の各フラグをクリアする。

【18】（オプション的な）メッセージ発信の設定

書式	int Mdo282_Plus_Message (int submit)
引数	submit : オプション・メッセージ発信の有無指定。 / 0 : 発信しない、 1 : 発信する
戻り値	正常終了時 : = 0 エラー時 : エラーコード (負の値 / エラーコード表 5 - 4 B 参照)
機能・動作	(マニュアルモードの場合) 関数【3】の trs_trig で指定する (本ボードの F I F O にデータを転送するための) 参照フラグを検出した時点でメッセージを発信する。 (自動モードの場合) 関数【3】の trs_trig で指定する (本ボードの F I F O にデータを転送のための) 参照フラグを検出し、所定の書き込み動作を行った直後にメッセージを発信する。

【19】本ハンドラのバージョン取得

書式	int Mdo282_Get_Libver (int ver)
	ver : 0 = 戻り値は (メジャー・バージョン番号) + (マイナー・バージョン番号) 1 : 戻り値は (メジャー・バージョン番号) 2 : 戻り値は (マイナー・バージョン番号)
戻り値	正常終了時 : 本ハンドラのバージョン番号 エラー時 : エラーコード (負の値 / エラーコード表 5 - 4 参照)
機能・動作	本ハンドラのバージョン情報を得る 例えばバージョンが 1.01 の場合、本関数を ver = 0 として実行すると 戻り値は 0 x 1 0 1 となります。

表 5 - 4 B . エラーコード一覧

戻り値	不具合の内容、または因果情報		適用関数、引数、等
- 1	ボードを検出できない。(未装着?)		Mdo282_Open_MdoSys ()
- 2	ボードIDが違う。		Mdo282_Open_MdoSys ()
- 3	FIFOバッファにアクセスできない。		Mdo282_Open_MdoSys ()
- 4	ドライバファイルを検出できない。		Mdo282_Open_MdoSys ()
- 5	ドライバファイルのバージョンが不適切。		Mdo282_Open_MdoSys ()
- 6	初期化が未来実行		
- 7	連続サンプリング開始前の設定不足。		Mdo282_Start_Samp ()
- 10	指定ボード枚数 (1 ~ 8 が適切)	指定パラメータの不適当	num_board
- 11	指定ボード番号 (0 ~ 7 が適切)	指定パラメータの不適当	board_no
- 12	サンプリング出力データ幅	指定パラメータの不適当	width
- 15	データ転送時の参照フラグ	指定パラメータの不適当	trs_trig
- 16	データ転送元	指定パラメータの不適当	buf_area
- 17	割り込みイベント発生要因	指定パラメータの不適当	intr_sw、
- 18	トリガ動作モード	指定パラメータの不適当	trg_mode
- 19	トリガ源	指定パラメータの不適当	trg_source
- 20	トリガ極性	指定パラメータの不適当	trg_pol
- 21	トリガ源選択	指定パラメータの不適当	trg_sel
- 25	クロック源	指定パラメータの不適当	clk_source
- 26	サンプリング・クロックの指定方法	パラメータの不適当	set_mode
- 27	クロック周期の単位	指定パラメータの不適当	time_unit
- 28	クロック周期の値、または分周比	指定パラメータの不適当	clk_period
- 29	トリガ後のサンプリング点数	指定パラメータの不適当	
- 30	動作モード	指定パラメータの不適当	mode1
- 31	連続サンプリングモード	指定パラメータの不適当	mode2
- 32	入力データ幅	指定パラメータの不適当	width
- 35	メモリ確保エラー		Mdo282_Start_Samp
- 41	外部クロック源周波数を設定していない。		Mdo282_Set_Clock
- 42	データバッファ・サイズが小さすぎる。		bufsize
- 43	マニュアルサンプリング開始前の設定不足		Mdo282_immediate_Update
- 44	転送先データバッファ・エラー		bufptr
- 45	FIFO空領域のデータ数を超過して書こうとした。		Mdo282_Write_Data
- 50	ユーザプログラム内のデータバッファが指定されている。		Mdo282_Write_DllData
- 51	本ハンドラDLL管理のデータバッファが指定されている。		Mdo282_Write_DirectFifo
- 52	連続サンプリング中なのに各種条件設定等をしようとした。		

5-5. サンプルプログラム、使用上の注意

重 要

本ハンドラを構成する関数は一定手順、または特定の組み合わせでのみ有効に動作します。 手順については前項先頭にも記しましたが、組み合わせの点で注意すべきことは前項の関数【3】で設定するサンプリングモードと【9】【10】のデータ転送関数です。 以下に連続サンプリングの典型例を示します。

< 例 1 >

利用形態	プログラム・ファイル名	対応言語 (version)
即トリガ、またはポストトリガ動作で、 当ハンドラDLLの管理するバッファ から 有限サンプリング 。	Sample1m.vbp	Visual Basic (5,6)
	Sample1m.c	Visual C (5,6)
	Sample1m.c	Borland C (5)
	Sample1m.dsp	Visual C++ (5,6)
	Sample1m.dpr	Delphi (3,4,5)
<p>“自動モード”で“スタート操作”を実行すると、 当ハンドラ自体が自動的に“当ハンドラDLLの管理するバッファ”から本ボード上のFIFOメモリにデータを転送する。 全データがFIFOに入りきれないときは、サンプリングの進行に合わせて補給転送する。 このとき、“データ転送の参照フラグ”を《Not-HALF-FULL》に設定してある場合は、“FIFOメモリ容量”の半分単位でブロック転送、また、《FULL 解消》に設定の場合は、“1 スキャン分”単位で転送する。 以上はスレッドによりバックグラウンドで自動実行され、指定トリガ後サンプリング点数に達すると自動停止する。</p>		

< 例 2 >

利用形態	プログラムファイル名	対応言語 (version)
即トリガ、またはポストトリガ動作で、 当ハンドラDLLの管理するバッファ から 無限サンプリング 。	Sample2m.vbp	Visual Basic (5,6)
	Sample2m.c	Visual C (5,6)
	Sample2m.c	Borland C (5)
<p>当ハンドラDLLの管理するバッファを“エンドレス・リング状”に設定し、前<例1>と同様に操作する。 この場合、“当ハンドラDLLの管理するバッファ”の末尾と先頭が連結された構造となり、1 廻り以後は再度先頭から転送される動作となる。 ユーザプログラム側からは、“サンプリング済みデータ点数”の値を当バッファのポインタ(関数【8】ステータスから得る)として認識し、追加データを補給転送する。 連続サンプリングがストップ操作まで無限に実行されるので、出力データがリアルタイムに発生する動作などに利用できる。</p>		

< 例 3 >

利用形態	プログラムファイル名	対応言語 (version)
即トリガ、またはポストトリガ動作で、ユーザプログラムの管理するバッファから有限 / 無限サンプリング。	S a m p l e 3 m . v b p	Visual Basic (5,6)
	S a m p l e 3 m . c	Visual C (5,6)
	S a m p l e 3 m . c	Borland C (5)
<p>“ マニュアルモード ” で “ スタート操作 ” を実行すると、本ボードが連続サンプリングを開始しても当ハンドラは何もしない。 ユーザプログラムは自力でステータス関数をポーリングして “ ボードの生ステータス ” から《Not-HALF-FULL》または《FULL 解消》を検出してデータの追加転送タイミングを知り、関数【10】Mdo282_Write_DirectFifo()で書き込む。</p> <p>このとき、 “ データ転送の参照フラグ ” を《Not-HALF-FULL》に設定してある場合は、 “ F I F Oメモリ容量 ” の半分単位でブロック転送、また、《FULL 解消》に設定の場合は、 “ 1 スキャン分 ” 単位で書き込む。 当モードで指定できるトリガ後サンプリング点数は、【7】スタート時に指定する有限値であるが、【2】非サイクル・無限サンプリング動作に設定すると “ ストップ操作 ” まで無限にサンプリング動作を続けることもできる。</p>		

補足説明

キーワード	説明、関連関数 / 【 】内の数字は関連する関数項目番号
“ 自動モード ”	【3】連続サンプリング動作モード、および割り込みイベントの設定関数で選択・指定する。
“ マニュアルモード ”	
“ データ転送の参照フラグ ”	
“ スタート操作 ”	【7】連続サンプリング・スタート関数
“ 当ハンドラDLLの管理するバッファ ”	ユーザプログラム側から【9】Mdo282_Write_DllDataで転送、容量は【2】指定ビット幅(ワード数) × 【7】指定データ点数
“ F I F Oメモリ容量 ”	【1】初期化のとき自動検出。
“ 1 スキャン分 ”	【2】指定ビット幅(ワード数)を各1回サンプリング分。
“ サンプリング済みデータ点数 ”	【8】ステータスから得る。
“ エンドレス・リング状 ”	【2】バッファ末尾の次が先頭に戻る。
“ ストップ操作 ”	【15】連続サンプリング動作の強制停止関数の実行。

Cハンドラ使用上の注意・制限

- (1) 本ハンドラは不適切な使い方をするとエラーを返してきますから、原因が除去されるようにデバッグしてください。特に問題を起こしやすい点は、

ハード（本ボード）とソフト（ハンドラ）設定の不整合。
本ボードと他のハード（非PnP対応ボード）の設定重複。：I/Oアドレス
割り込み番号

特にハード同志の重複設定による不具合は本ハンドラで検出できませんから、システムの構築時に十分な確認が必要です。

- (2) 本ハンドラは【Mdo282_Open_MdoSys（ ）】で使用開始、
【Mdo282_Close_MdoSys（ ）】で使用終了とします。

特に終了手続きは本ハンドラ内で確保したメモリの開放等を含むので重要です。

- (3) サンプリング実行時の《パソコン側データバッファ》～《ボード上のFIFOバッファ》間データ転送の実行タイミング検出に割り込みは使用しておらず、単独の指定イベントによるメッセージ通知機能だけです。

- (4) データロスト・エラー

本ボードの自動サンプリング動作でFIFOバッファメモリからデータが流出する速度よりパソコン側から書き込む速度が遅く、FIFOバッファメモリが空になったところで次のデータが読み出されようとするとき実現できず、出力動作が止まってしまいます。

これがデータロスト・エラーで、構築されたシステム全体の処理速度を超えたクロック値を指定して実行したときに起ります。なおデータロスト・エラーの発生を本ハンドラが検出すると連続サンプリングを強制停止させます。

システムパフォーマンスはCPUを含むパソコン本体の実行速度や周辺機器の状態に左右されますが、（ユーザが記述する）本ハンドラの応用プログラムがサンプリング実行&データ書き込み（ブロック転送）に専念した場合、本ボードがデータロストエラーを起こすことなく実現できる（本ハンドラでの）期待最高サンプリング速度はパソコン側CPU能力に依存し、900K語/秒（Pentium4/1GHz）、200K語/秒（Pentium2/400MHz）程度です。

- (5) 直接I/O操作を行うとき

本ハンドラを介さずボードに直接I/O操作（OUT/INP）を行うと、本ハンドラの管理が行き届きませんから不本意な動作となることがあります。

第6章．保守・その他

6-1. 故障・トラブル等の原因と対処

本機は全数検査のうえ出荷されています。

動作に不具合等が見られるときは以下の諸点を再点検してください。

それでも不明なときは巻末の【Q & A フォーム】にシステム構成（特に外部機器の接続回路）等の動作条件を御記入のうえ、技術部宛 FAX してください。

迅速に応答する体制となっています。なお TEL いただく場合も、客観情報の整理・評価は問題解決のスピードアップにつながりますから、事前に【Q & A フォーム】を FAX してください。

再点検・確認ポイント

- | | |
|--------------|--|
| (1) I/O アドレス | 他の特に非 P n P デバイスと重複・競合はないか？ (1 - 5 項) |
| (2) 割り込みレベル | 他のデバイスと重複・競合はないか？ (3 - 10 項) |
| (3) ボード番号 | 他の同ボードと重複はないか？ (3 - 14 項 / マスタスレーブ動作) |
| (4) デジタル入出力 | 本ボードの TTL 入力（割り込み、トリガ、クロック、汎用 bit）に接続できる信号源は TTL（LS、CMOS 等の 5V 電源動作素子）に限ります。現場などで不適切な信号源を接続したために本ボード内の TTL 入力素子を破損する事故が頻発していますので御注意ください。 |

動作確認方法

当社では原則として、ユーザ作成のソフトウェアについては評価しません。

動作確認は無償配布の当社製プログラム実行結果について推測・適否・判定を行います。

QA リクエスト時には当プログラムの実行結果をレポートしてください。

6-2. 修理のときは

入手経路の如何にかかわらず当社宛に直接お申しつけください。 商社等を経由されますと時間がかかるだけでなく、情報交換の不便、費用の面でも不利になります。 なお当社では修理依頼を受けた製品が検査の結果、良品と判定された場合は（保証期間内でも）手数料を申し受けます。

特に最初からの不具合には誤解や情報不足によることが多いので、事前に御相談ください。

【Q & A フォーム】が便利です。

無償修理

納入後1年以内の自然故障、および当社製造上の問題に起因した故障に対しては無償修理を行います。 但し、故障・不具合の原因や無償修理の対象となるか否かは（過去の経験等に照らして）当社側で判定させていただきます。

なお当社では保証書を発行していませんが、社内では製造番号と出荷年月日の記録を基に判定しています。

有償修理

落雷等の自然現象、漏電・過電圧印加・機械的破損・その他、ユーザ側の責に帰する故障品、または納入後1年間を経過した製品の自然故障に対しては実費・有償にて修理をお請けします。 性格上、事前見積もりは不可能ですが、制限額を事前通知いただければ、作業過程で制限を超えそうな見通しがたった時点で連絡・相談させていただきます。

受け渡し : 通常の授受は宅配便で行います。

修理期間 : 全んどの場合、当社内で24時間以内に完了・返送しています。時間を要するような場合は御連絡いたします。

費用の目安 : 修理費用は事務管理手数料、技術者の所要時間（1時間単位）手数料、および交換部品代の合計です。 2003年9月現在（時勢により変動します）では、

事務管理手数料（1件当り、返送運賃含）：＝ ¥ 4,000

修理時間手数料：＝（時間単価 ¥ 6,000）× 所要時間

交換部品代 : ＝ ¥ 実費

故障経緯、システム客観情報の添付は時間の節約・コストダウンに有効です。典型的な事例では費用合計が ¥ 20,000 を超えることは希れです。

【注2】 当社製品に対してユーザが改造を行った場合は、当社サポートの対象外になります。 改造とは製品に新たな部品を追加実装、または実装部品を削除したり、回路パターン・接続に変更を加えることです。 なお、当社がオプションとして供給、または指定した部品の追加実装・交換はこの限りではありません。

FAX : 03 (3301) 5593

Q & A フォーム

発信： 年 月 日 / 時 分

(動作状況)

《60分以内に応答のないときはお叱りください。》 TEL: 03(3396)8377

(所属部・課)

(所在地)