

第4章. ソフトウェア

《注1》 以下説明のソフトウェアはWINDOWS 9x/2000/XP/VISTAに対応したもので、旧WINDOWS 95/NT、およびMSDOSには対応していません。これら旧OSでの御利用は旧機ADM-68xPCI用マニュアル (adm68x-ac.pdf) に紹介されています。ボードのインストールから違いますので御注意ください。

《注2》 マルチサンプリング、

平均処理機能対応： 現在、当機能に対応しているのは第6章に掲載の
WINDOWS 9x/2000/XP/VISTA用ハンドラ
関数、およびデータ収集ソフトLABDAQのみです。
(2007年3月)

4-1. ソフトウェアのインストール (ボードのインストール後に行います。)

本製品用のソフトウェアは3.5インチ(1.44MB)FDまたはCDに圧縮された形で格納されており、同メディア内のインストーラ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。

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。

- ◆ADデータ収集： (アプリケーション) WINDOWS 2000/XP/VISTA版LABDAQ-AQです。
ADサンプリング/リアルタイム波形表示/データファイル保存等、
すぐ実用になる基本機能を備えています。(信号処理なし版)
当社ホームページ (www.microscience.co.jp) から無償ダウンロード。
- ◆市販ソフト対応： 表計算 (EXCEL等) 向けのCSV形式、
および波形解析ソフトDADISP向けの専用形式ファイルを上記のAD
データ収集ソフトで作成・保存できます。
- ◆ハンドラDLL： (WIN32ライブラリ) ADボード制御の基本機能を関数化、サンプリング条件設定から実行、
配列にデータを得るまで、ユーザ記述のメインプログラムから呼び出す
だけです。VB, C, C++, Delphi, C++Builder の使用例を添付。
【専用マニュアル：第6章】
- ◆その他： (for WINDOWS) WINDOWS 9x/2000/XP/VISTAでI/O読み書きを
実行するための汎用ドライバ、およびDLLが添付されています。
基本的には当DLLを使用してボード上の各レジスタを読み書きする
ことでプログラミングが可能です。
(当部分は圧縮されておらず、生のまま格納されています。)
- ◆LabVIEW： VIサンプルソースを添付ディスク内、または当社ホームページから
ダウンロードすることができます。
- ◆ActiveX： オプション (松山アドバンス社製：AXADM-68X ¥16,000)
- ◆Linux： オプション (ナリタ社製：Kuziraシリーズ) 各¥36,000

= ソフトウェアのインストール作業 = (1-5項: ボード自体のインストール後に行う。)

添付の(2005年2月版以降の)CDROMを使用します。

操作手順

◆インストール元: Dドライブ(CDROM)

◆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¥AD¥ADM686z (ADM687z) を開く。

(2) “Setup. EXE”を実行(ダブルクリック)する。

当操作以下によりADM-686z/687zPCI関連プログラムが
次頁: 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。

<2005年2月版以前の旧CDROMを利用する場合>

操作手順

◆インストール元: Dドライブ(CDROM)

◆インストール先: Cドライブ(HDD) の場合で例示。

(□はスペース)

C:¥WINDOWS>CD¥【ENTER】

C:¥>CD□D:¥INSTALL¥PCI¥AD¥ADM686Z【ENTER】
(¥ADM687Z)

C:¥>D:INSTALL□D:□C:【ENTER】

各プログラムグループ(C, BASIC等)ごとにインストール実行の有無を問うてきますから、【Y】=yes, 【N】=no, で答えるだけで作業が進みます。

《注》 MS-DOSの環境変数“COMSPEC”が設定されていないか、または正常に設定されていないと本インストール・プログラムの作業が途中で停止してしまいます。 実行前に確認、または設定しておいてください。

= 設定例 = COMMAND.COMがCドライブの¥にある場合、

>SET□COMSPEC=C:¥COMMAND.COM【ENTER】

全ファイルをインストールした場合のディレクトリ構造は図4-1に示します。

【注】 ボード依存性のない汎用のWINDOWS版I/O実行DLL/デバイスドライバは当作業ではインストールされません。 WINDOWS95・98用はWin9xフォルダにありますので各ファイルを適合フォルダにコピーする必要があります。

WINDOWS NT用はWinNTフォルダ、またWINDOWS2000/XP/VISTA用はWin2Kフォルダにあります。

(追伸) CDROMの場合、Win9x、WinNT、Win2KフォルダはINSTALLフォルダ下のDriversフォルダ下にあります。

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。

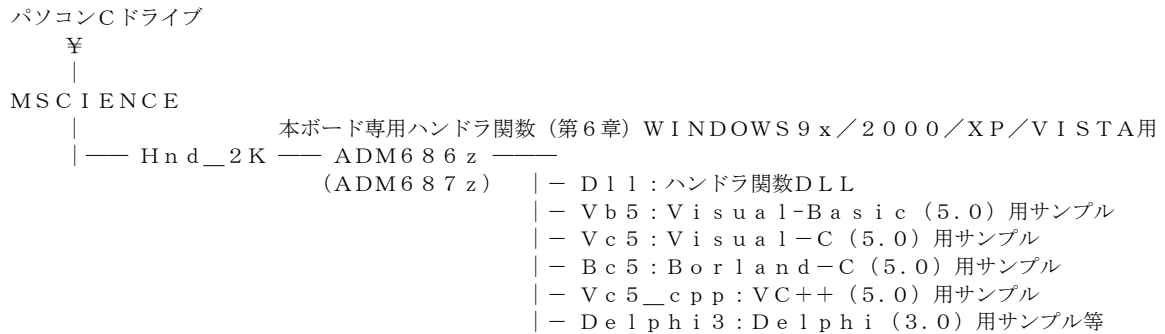
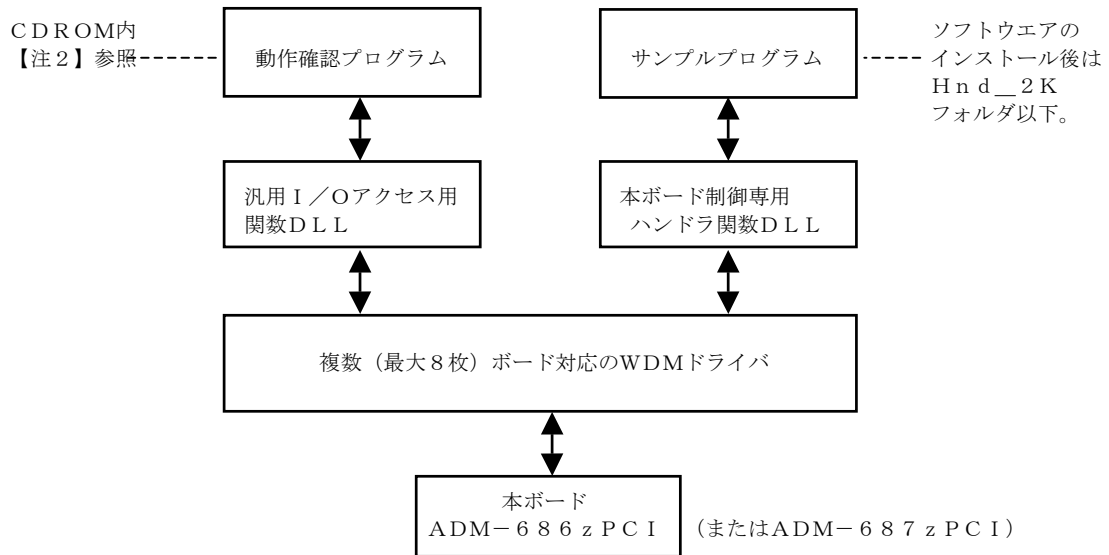


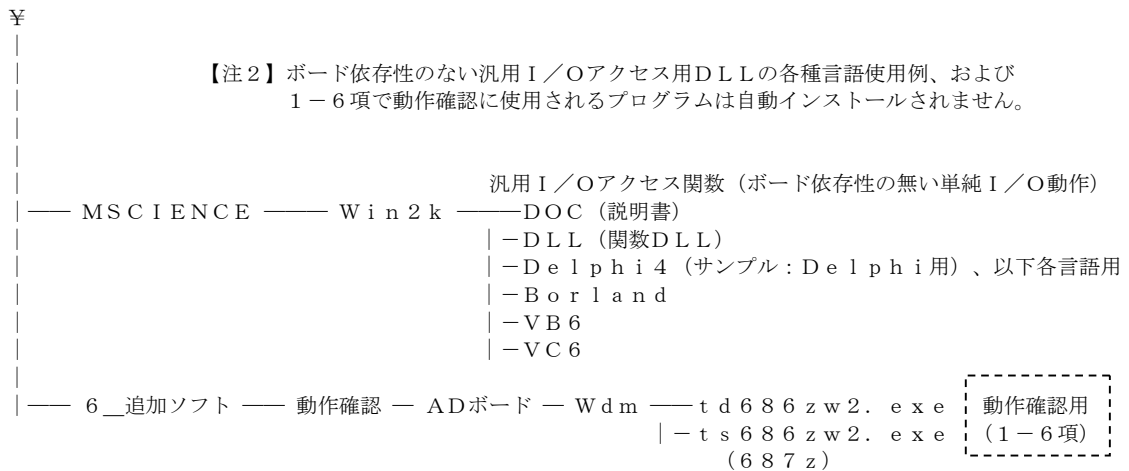
図4-1B. ソフトウェア構成



【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

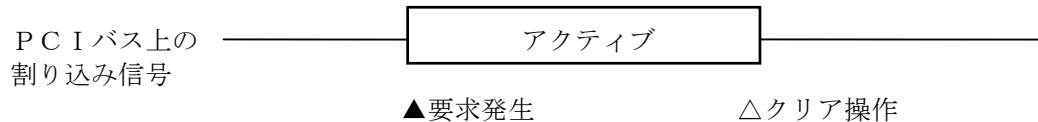


【注2】ボード依存性のない汎用I/Oアクセス用DLLの各種言語使用例、および1-6項で動作確認に使用されるプログラムは自動インストールされません。

4-4. 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4-4.



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があります。

添付の汎用デバイスドライバ（WINDOWS9x/ME/NT用）では、アプリケーション側からクリア操作に必要なパラメータを（あらかじめ）受け取っておき、割り込みが発生したらクリア操作を実行します。またアプリケーション側からは割り込み発生（回数：Read Clear）を読み取る関数DLLをポーリングする形をサポートしています。

このようなアルゴリズムは（割り込みを使用せず）ボードのステータスをポーリングする方法と等価ですから、無用なトラブルを回避するためにも後者をお勧めします。

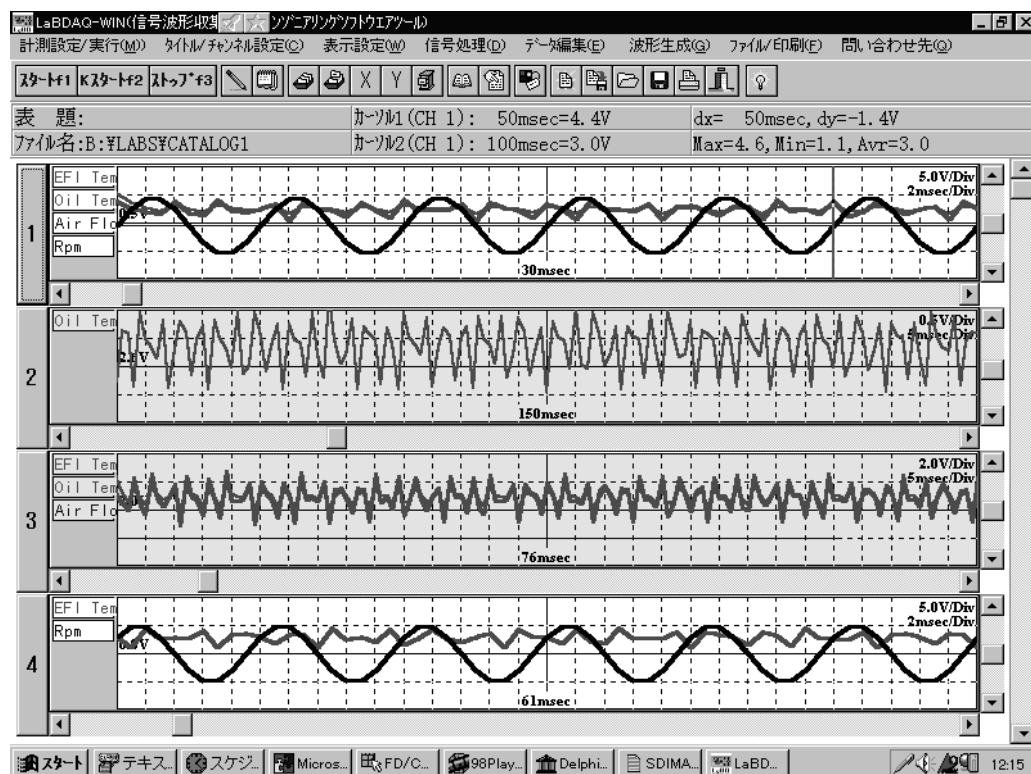
同じく添付の本ボード専用ハンドラ／デバイスドライバ（WINDOWS9x/ME/NT用）では割り込みによるADデータ転送、または外部イベント対応処理が標準でサポートされています。（割り込み不使用でもマルチスレッドでポーリング／ADデータ転送できます。）

なお、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】

（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。

《信号処理なし版のLaBDAQ-AQは無償ダウンロード配布》
 WINDOWS (2000・XP・VISTA) データ収集・解析ソフトウェア

LaBDAQ-AQ/PRO



LaBDAQ-PROはADボード、データファイルから読み込んだ時系列データ群のグラフ表示（リアルタイム可能）、ハードコピー、レポート印刷、市販ソフトにも読み込めるテキストファイル保存、さらにFFT・相関・移動平均・ピーク検出等のデータ処理まで行うことのできるパッケージソフトです。（機能限定デモ版は添付CDROMの松山アドバンス社フォルダ内）

- ◆データ格納チャンネル数：最大512（ADデータ入力用、演算処理結果出力用の合計）
- ◆データ表示ウィンドウ数：最大32（同時表示は横4列×縦8列）
- ◆縦軸は任意の物理単位・スケールで表示可能、横軸は時間またはデータ順番号。
- ◆カーソルによる座標認識（時刻・値）、任意区間のズームング。
- ◆ADサンプリング実行中でもリアルタイム表示可能。
- ◆繰り返しADサンプリング実行機能。
- ◆FFT・相関・移動平均・ピーク検出等の汎用デジタル信号処理機能。
- ◆市販の表計算やデータ処理ソフトにインポート可能（CSVテキストファイル保存機能）
- ◆多様なADボードに対応（マイクロサイエンス社製ADボード各機／別表）

動作環境

- ◇WINDOWS 9x・NT・2000・XP、またはVISTAの動作するIBM PC/AT互換。
- ◇本体メモリ：16MB以上。
- ◇ハードディスク：空容量10MB以上
- ◇ディスプレイ：800×600以上。

ADサンプリング・デジタル信号処理仕様

- ◆AD入力チャンネル数 : 最大 512 (使用ボードのチャンネル数: 各 1 枚のみ対応/別表)
- ◆データ格納チャンネル数: 最大 512 (ADデータ入力用、演算処理結果出力用の合計)
- ◆最高サンプリング速度 : 各ADボードの最高値 (チャンネル数、データ数に依る/別表)
- ◆最大 (管理) データ量 : 搭載空RAM、および仮想メモリの範囲内。
- ◆デジタル信号処理機能 : チャンネル間の四則演算、直流成分の抽出または除去、
【PRO版のみ】単純移動平均、2次多項式適合平均、ピーク検出、
FFTパワースペクトル、位相スペクトル、振幅スペクトル、
自己相関、相互相関、伝達関数、AMDF、ヒストグラム、
IIRフィルタ (最大 30 次)、FIRフィルタ (最大 125 次)
(演算精度: IEEE規格32BIT浮動小数点/ANSI準拠)

データファイルの保存/読み込み

- ◆ADデータ保存: LaBDAQ専用バイナリ形式による高速保存/読み込み、または汎用CSVテキストファイル (表計算等、市販ソフトに読み込み可能)、波形解析ソフトDADISP専用形式。
- ◆動作条件保存 : ADサンプリング動作条件の保存/読み込み

サンプリング条件 (トリガ・クロック・モード) の設定

計測実行条件の設定

スタートモード

☒ 即時スタート
☐ トリガスタート

実行モード

☒ 1回のみ
☐ ストップまで連続

実行中状況表示

☐ 計測中状況表示しない(高速)
☒ 計測中カウンタのみ表示する(中速)
☐ 計測中波形表示する(低速)

取込サンプリングクロック

1K: 1mS

クロック基準

☒ 1MHz(通常)
☐ 1.024MHz(FFT解析用)

トリガ条件設定

トリガ発生源
☒ 内部チャンネル0
☐ 外部コネクタトリガ端子

トリガデータ/電圧

128 0.0000 Volts

トリガモード

☒ レベル
☐ エッジ

トリガ極性

☒ - ↓
☐ + ↑

実行 中止(A)

最大512のデータ格納チャンネル（ADデータ入力用、演算処理結果出力用の合計）

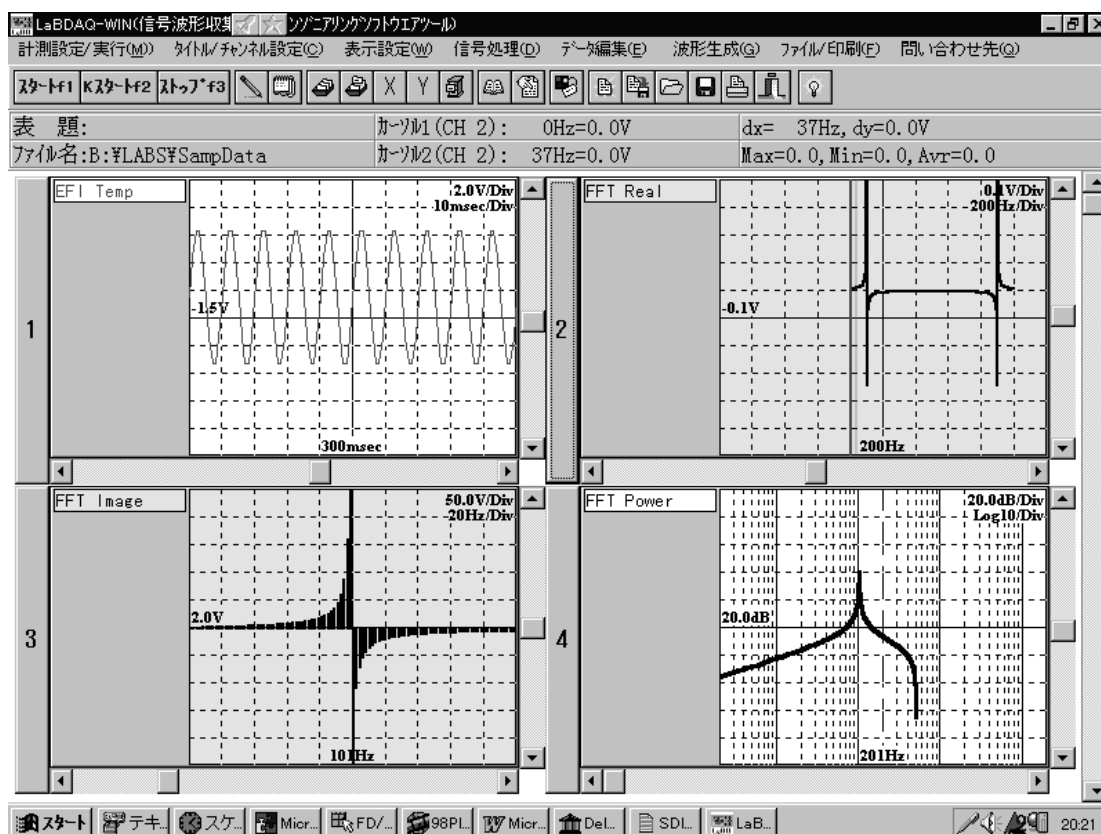
チャンネル設定

サンプリングチャンネル数: 4 サンプリングデータ点数: 512 (チャンネル×データ点数32M以下)
 演算チャンネル数: 4 演算データ点数: 512 (1チャンネル当たり1M以下)
 (両チャンネル計で32以下)
 データ間隔時間(サンプリングクロック): 1.0000 ☐ usec ☒ msec ☐ sec

チャンネルNo	信号名称	表示形式	表示色	変換w	変換係数a,b (Y=aX+b)	データ数
1 サンプリング用		実線2		<input type="checkbox"/>	1.000 0.000	512
2 サンプリング用		実線1		<input type="checkbox"/>	1.000 0.000	512
3 サンプリング用		実線3		<input type="checkbox"/>	1.000 0.000	512
4 サンプリング用		実線2		<input type="checkbox"/>	1.000 0.000	512

OK

入力データ（波形）に対するFFT等のデジタル信号処理



実行速度

下表は代表的な条件での最高サンプリング速度です。実際にはシステム構成や総データ点数等によって異なります。F I F Oメモリ付ADボードの場合、F I F Oメモリ容量までは（C P Uを含む）システム能力にかかわらず常にADボード自体の最高速度が得られます。

ADボードのF I F Oメモリ容量は現在、最大32M語まで増設可能です。
（ソケット実装により後日の増設も可能）

各ADボード/最高サンプリング速度：1Mデータの場合（複数チャンネルの場合は周期で示す）

ADボード名	P e n t i u m (400MHz)		486 (66MHz)	
	単チャンネル	複数チャンネル	単チャンネル	複数チャンネル
ADM-640AT	1MHz	(1×ch) μs	1MHz	(1×ch) μs
ADM-652AT	100KHz	(10×ch) μs	80KHz	4+(12×ch) μs
ADM-656AT	40KHz	(25×ch) μs	40KHz	(25×ch) μs
ADM-670PCI	2MHz	1MHz		
ADM-676PCI	200KHz	(5×ch) μs		
ADM-677PCI	100KHz	(10×ch) μs		
ADM-680xPCI	500KHz	(2×ch) μs		
ADM-681PCI	500KHz	(2×ch) μs		
ADM-682zPCI	256KHz	(4×ch) μs		
ADM-686zPCI	200KHz	(5×ch) μs		
ADM-687zPCI	200KHz	(5×ch) μs		
ADM-688PCI	500KHz	(2×ch) μs		
ADS-0128K	200KHz	(10×ch) μs		
ADM-5298BPC	100KHz	(10×ch) μs	35KHz	(32×ch) μs
ADM-5598BPC	100KHz	(10×ch) μs	35KHz	(32×ch) μs
ADM-5698BPC	40KHz	(25×ch) μs	40KHz	(25×ch) μs
ADM-5898BPC	83KHz	4+(8×ch) μs	35KHz	(32×ch) μs
ADM-8298BPC	200KHz	(5×ch) μs	100KHz	4+(11×ch) μs
ADM-8398BPC	200KHz	(5×ch) μs	100KHz	4+(11×ch) μs
ADM-8498BPC	200KHz	(5×ch) μs	100KHz	4+(11×ch) μs
ADM-8698BPC	166KHz	(6×ch) μs	100KHz	4+(11×ch) μs

【注】ISAボード：WINDOWS 9x・NTのみ対応、Cバスボード：WINDOWS 9xのみ対応。

- ◆合計データ点数：1データは2バイトですから、パソコンの空メモリ（RAM）に設定仮想メモリ量を加算した値を2で除したものになります。
但し仮想メモリ領域はディスクですから、サンプリングデータ数が空RAM領域を超えると急に遅くなります。上表値はRAM領域に書き込む場合の速度です。
- ◆リアルタイム表示：サンプリングを実行しながら同時に経過をグラフ表示することもできます。実行・描画速度はパソコン側の能力にもよりますが、数ms／点程度です。
- ◆供給メディア：3.5インチFD（1.44MB）／IBM互換機・98系機共用。
- ◆価格
L a B D A Q - A Q ¥無償（ADデータ収集、2000/XP/VISTA版）
L a B D A Q - P R O ¥78,000（ADデータ収集&解析、フル機能版）

第5章. DOSハンドラ

本機（ADM-686z／687zPCI）用のMS-DOSハンドラ関数ライブラリはありませんが。旧ADM-686／687（a）PCI用があり、流用できます。
但し、マルチサンプリング機能には対応していません。

第6章. WINDOWSハンドラ

当社製の高速ADボードADM-686z/687zPCI (IBM PC/AT互換機用) をVC、VB等、メジャー言語で簡単に使用することのできるマルチスキャン・複数ボード対応のWINDOWS 9x、2000・XP、VISTA用ハンドラ関数DLL (+ドライバ) です。

AD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AD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。

6-1. システム構成・ソフトウェア構造

パソコン本体 : IBM PC/AT互換機 (含む98NX機)

拡張メモリ量 : 16MB以上

OS/コンパイラ : WINDOWS 9x、2000、XP、VISTA/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 : 各ADボード添付のサンプルディスク内。

対応ADボード : ADM-686z/687zPCI (最大8枚まで制御可能、混在不可)

サンプリング : ボード側からの最高データ転送 (PCIバス上の) 速度は、
約800KHz (Pentium4/2GHz)、250KHz (Pentium2/400MHz) 程度です。

上記速度は当ハンドラDLLがボードからのADデータ読み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。したがって使用ボード枚数が増えると比例して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。

但し読み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5μs/chで取得したADデータを (遅れながらも) 有効に読み込むことができます。

■ FIFOメモリは最大32M語まで増設可能です。■

マルチ

サンプリング : 連続・自動サンプリングのとき1クロックごとのサンプリング点数を複数回 (最大2の32乗回) とすることができます。この複数回サンプリング実行間隔も別のクロック (マルチサンプリングクロック) で指定できるので測定サイクルを間断なく繰り返し実行するような利用方法があります。。

さらに、(最大255回ですが) これを平均処理した1データとして取り込むこともでき、雑音除去に有効です。この演算はボード上でハード的に行われるので、ソフトの負担はありません。

データ点数 : 拡張メモリ空容量 (1語 = 2バイト)

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。

6-2. 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6-4項以下で詳述しますが、要は使用するチャンネル数／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、ADデータはバッファに格納されて戻ってきます。

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。

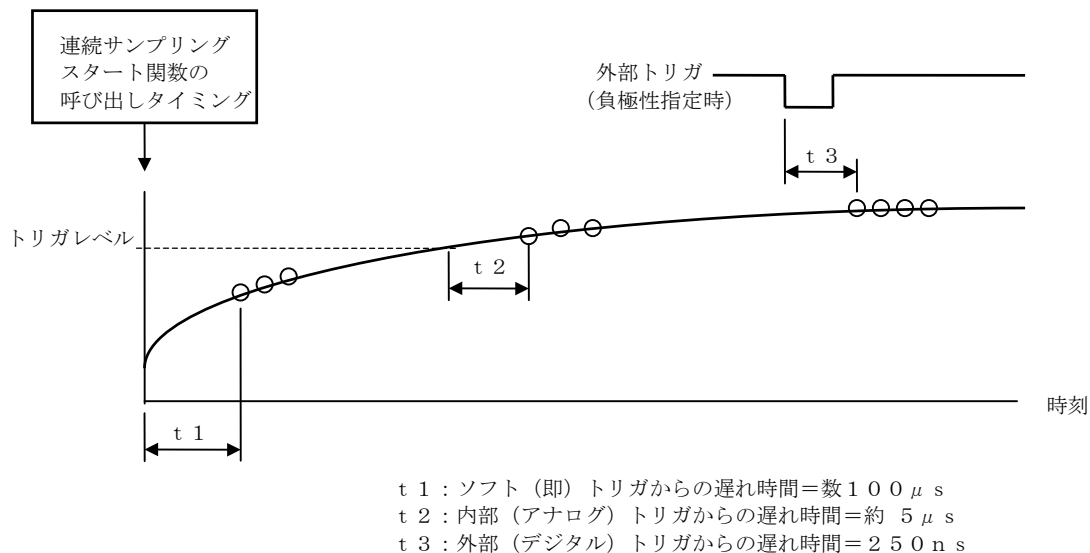
- (1) マニュアル（1回）サンプリグ・スキャン動作はメインルーチン上任意のプロセスで、指定チャンネル群入力を各1回だけAD変換し、結果をADデータバッファに得ます。
- (2) 連続サンプリグ動作は指定チャンネル群に対し指定時間々隔（クロック周期）で指定回数だけAD変換を行い、結果をバッファに得ます。
動作開始となるトリガはソフト（即スタート）、内部（アナログ）、外部（デジタル）から選択できます。クロック源も内蔵の10MHz / 8.192MHz、外部TTL入力から選択できます。いずれのクロック源も（ADボードの実行可能な速度以内で）任意の整数値で分周して利用します。外部クロックを分周比＝1 / 1で使用すると、外部のイベントに同期したAD変換動作となります。
- (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ADデータバッファは各ボードごとに複数用意されます。

【注1】 指定チャンネル群： チャンネル0を先頭とし、連続したチャンネル群。

【注2】 内部（アナログ）トリガ： 指定レベル／極性とチャンネル0入力を比較して検出する。
外部（デジタル）トリガ： 専用TTL入力（TRG-IN）の指定極性エッジ。

【注3】 外部クロック源入力： 専用TTL入力（CLK-IN）の指定極性エッジ。／最高10MHz

図6-2 A. 連続サンプリグの開始タイミング例



【注A】 ソフトトリガからの遅れ時間 t 1 はスタート関数の呼び出し、および同関数内の管理手続き実行時間を含むものです。ADボード自体のソフトトリガ・ビットをセットしてからの遅れは最大250 ns です。

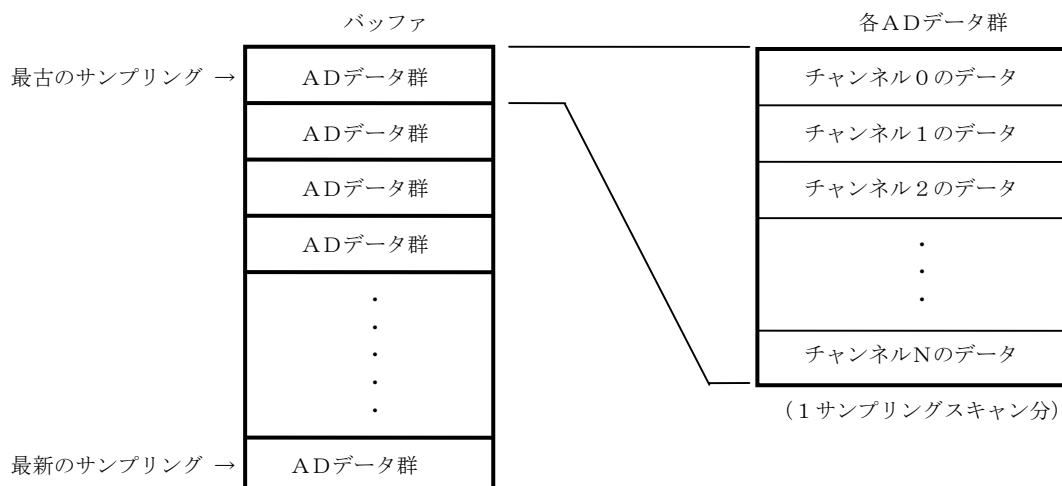
【注B】 内部（アナログ）トリガからの遅れ時間 t 2、および外部（デジタル）トリガからの遅れ時間 t 3 は各スタート関数の呼び出しと同関数内の管理手続き実行してトリガ待ち状態となつてからの時間です。このトリガ待ち状態となるまでの時間は数100 μ s を要します。

ADデータバッファと格納の様子

サンプリングされたADデータは指定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）ADボード上のFIFOメモリと同一イメージで全チャンネルのAD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6-2B. 全チャンネルのADデータが1個のバッファに入る

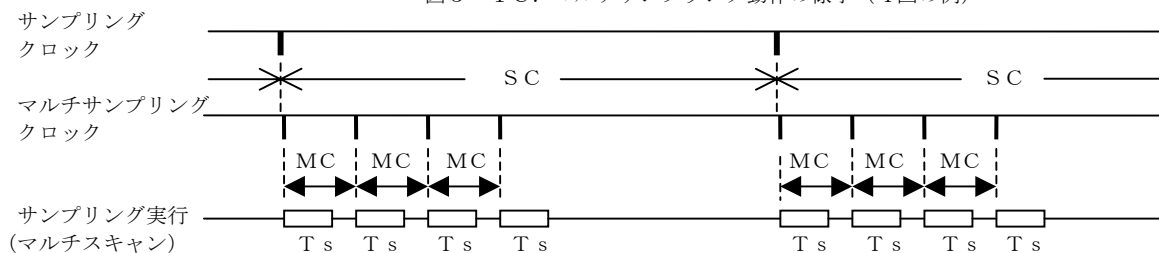


■マルチサンプリング機能を使用する場合は、

ADボード内で**平均化処理する条件**でマルチサンプリング機能を使用するときは同機能を使用しない場合（図6-2B）と同一です。

ADボード内で**平均化処理をしない条件**でマルチサンプリング機能を使用するときは、各回のサンプリング・クロックによる“ADデータ群”が指定マルチサンプリング回数倍になります。

図3-1C. マルチサンプリング動作の様子（4回の例）



SC : 通常のサンプリングクロック、MC : マルチサンプリングクロック【3-21項参照】

Ts : 1 スキャン時間 = (スキャン速度) × (使用チャンネル数)

スキャン速度 = 10 μs / 低速モード、または 5 μs / 高速モード【3-5項参照】

【注1】マルチサンプリングクロック周期MCの最小値 = 1 スキャン時間 = (使用チャンネル数) × (スキャン速度)
(最高速動作時) 関数【2】 関数【21】

【注2】サンプリングクロック周期SCの最小値 = (使用チャンネル数) × (スキャン速度) × (マルチスキャン回数)
(最高速動作時) 関数【2】 関数【21】 関数【5】

6-3. 使用準備

ボード上の設定・入力接続は標準設定【1-2項】【3-16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¥Adm68■zフォルダ）を使用できます。

◆WINDOWS 2000・XP・VISTA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（H68■z_2K.d11）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¥Adm68■z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WDM型ドライバを使用するため、WINDOWS 2000／XP／VISTA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／VISTAと同一の方法・手順でインストールする。
- (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（H68■z_2K.d11）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¥Adm68■z以下）を使用できます。

6-4. 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6-3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) サンプル条件の設定 : 【2】～【7】 チャンネル／トリガ／クロック／動作モード
 (3) スタートまたはトリガ待ち : 【8】 即スタート／トリガ待ち（内部 or 外部）
 (4) ステータス評価 : 【9】 連続サンプリング進捗状況
 (5) 以下・途中は任意
 (6) サンプリング中止 : 【18】 サンプリング終了後に必須。
 (7) ボードのフラグクリア : 【20】 このあと、再度（2） or （3）に戻るときは必要。
 (8) ハンドラ終了 : 【10】 後処理

表 6-4 A. 制御関数一覧 （注）正確には各関数名中、AD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】 AD_Open_ADSys	ボード、および本ハンドラの初期化	
【2】 AD_Set_SampCh	サンプリング実行チャンネル関連設定	実行チャンネル数
【3】 AD_Set_SampMode	サンプリングモード、イベントの設定	ADデータ転送先、方法
【4】 AD_Set_Trigger	トリガ関連設定（レンジトリガ以外）	トリガ源、レベル、モード
【4】 AD_Set_RangeTrigger	トリガ関連設定（レンジトリガ）	
【5】 AD_Set_MultiSamp	マルチサンプリング動作モード設定	
【6】 AD_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】 AD_Set_Mexclk	同上マルチサンプリングクロック源用	クロック源の周波数値
【7】 AD_Set_Clock	サンプリング・クロックの設定	クロック源、周期値、単位
【7】 AD_Set_MsClock	同上マルチサンプリング・クロック用	クロック源、周期値、単位
【8】 AD_Start_Samp	サンプリング開始（トリガ待ち or 即）	サンプリング点数
【9】 AD_Get_Status	ステータス取得	サンプリング進捗状況など
【11】 AD_Read_DllData	DLL管理バッファからデータ読出し	ADデータ格納バッファ
【12】 AD_Read_DirectFifo	ADボードから直接にデータ読出し	ADデータ格納バッファ
【13】 AD_Get_OneScan	マニュアル（1回）サンプリング	
【10】 AD_Close_ADSys	本ハンドラの終了	
	《以上が基本、以下は補助的》	
【14】 AD_Out_Aux	汎用デジタル（ラッチ）出力	出力データ
【15】 AD_Inp_Aux	汎用デジタル（現在値）入力	入力データ
【16】 AD_Set_SampLoop	データバッファをリング状に設定	1 廻り後は上書き
【17】 AD_Set_Inpmode	データコード指定	
【18】 AD_Stop_Samp	サンプリング動作の（強制）中止	
【19】 AD_Read_RestData	ADボードの残りデータ読み込み	エラー停止の後
【20】 AD_Clear_Flags	ADボードのフラグクリア	ビット指定
【21】 AD_Set_ScanSpeed	サンプリングスキャン速度指定	
【22】 AD_Set_Shc	外付サンプルホールド制御	オプション
【23】 AD_Plus_Message	オプション的なメッセージ発信の設定	任意
【24】 AD_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。 ADM-686zPCI用：■＝6z

ADM-687zPCI用：■＝7z

【1】初期化

書式	int AD68■_Open_ADSys (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 : F I F Oメモリ容量 (WORD単位のデータ容量) = (ハーフフル点数) × 2 メガF I F Oモジュール使用時の (ハーフフル) は半分ではなく、同モジュール 上のスイッチ設定値0 (512B=256点) ～6 (32KB=16K点)。</p>
戻り値	<p>正常終了時 : ボードのID／ FH (ADM-686zPCI)、11H (ADM-687zPCI)</p> <p>エラー時 : エラーコード／負の値 (エラーコード表6-4B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IDはPCIバスで定義されるSubsystem IDとは別物。／3-4項参照。</p>

【2】サンプリング実行チャンネル関連の設定／ (ボードごとに実行)

書式	int AD68■_Set_SampCh (int board_num, int no_ch, int scan_order[], int range[])
引数	<p>board_no : 対象ボード番号。 (0～7) ／本関数は各ボードごとに行う。</p> <p>no_ch : サンプリングを実行するチャンネル数 (1～16) ／必ず全ボード同一値。</p> <p>scan_order [] : 各チャンネルのスキャン順。 (未使用、本ボードでは固定)</p> <p>range [] : 各チャンネルの入力レンジ番号。 (未使用、本ボードでは固定)</p>
戻り値	<p>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6-4B参照)</p>
機能動作	<p>サンプリングを実行するチャンネル数、入力レンジを設定する。</p> <p>スキャン順番 (バッファメモリ中のADデータ並び) は若いチャンネル番号からの順に固定されている。</p> <p>入力レンジはボード上のスイッチ設定。</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SW-BN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SW-BN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からのサンプリングADデータ転送モードの設定

書式	int AD68■_Set_SampMode (int trs_trig, int buf_area, int intr_sw)
引数	<p>trs_trig : ADデータ転送の参照フラグ／0 : EMPTY解消、 1 : HALF-FULL、</p> <p>buf_area : ADデータ転送先 / 0 : DLL管理バッファ、 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	<p>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6-4B参照)</p>
機能動作	<p>trs_trigで<0 : EMPTY解消>を指定するとADデータを1個々連続的に転送する動作、 <1 : HALF-FULL>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DLL管理バッファとした場合は当ハンドラが管理するバッファにデータを 読み込む自動モード、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データ点数を 認識して専用の関数AD68■_Read_DLLDataで読み出す。</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身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、専用関数AD_Read_DirectFifoでユーザプログラ ム側バッファに読み込むマニュアルモード。</p> <p>【注1】 : デュアルコアCPU (マルチコアCPU) のパソコンではマニュアルモードの使用不可。 【注2】 : いずれのモードでもADデータ転送時に参照するフラグをtrs_trigで指定する。 【注3】 : メガF I F Oモジュール使用時の (HALF-FULL) は半分ではなく、スイッチ設定値。 【注4】 : 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>

【4-1】トリガ関連の設定A（レンジトリガ以外の場合）

書式	int AD68■_Set_Trigger(int trg_mode, int trg_source, int trg_pol, int trg_level)
引数	trg_mode : トリガ動作モード／0 : 即トリガ、1 : ポストトリガ trg_source : トリガ源 / 0 : ソフト、1 : 内部（アナログ）、2 : 外部デジタル 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 trg_level : トリガレベル（アナログ）／対応するADデータコード上位8ビットで指定する。
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6-4B参照）
機能動作	トリガ動作モード、源、極性、レベル等の設定。なおトリガ動作モードをポストトリガ、トリガ源を外部（デジタル）で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） トリガ検出対象は（複数ボード使用のときはマスタボードの）チャンネル0。

【4-2】トリガ関連の設定B（アナログ・レンジトリガ使用の場合）

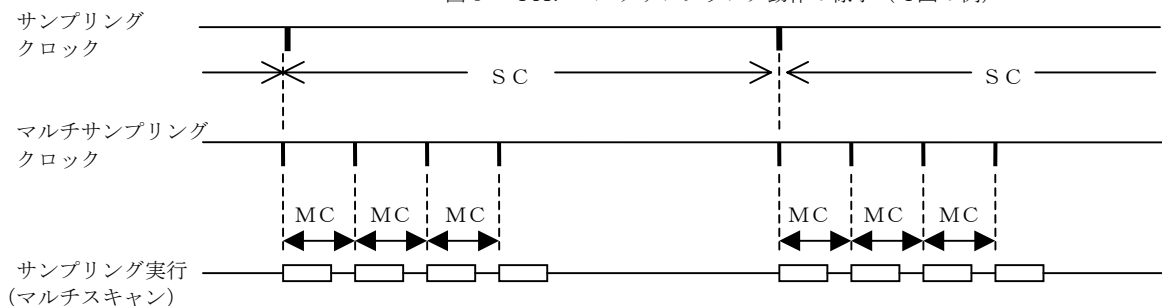
書式	int AD68■_Set_RangeTrigger(int trg_sel, int trg_level_hi, int trg_level_lo)
引数	trg_sel : アナログトリガ・モード／0 : アウトレンジ、1 : インレンジ 2 : デュアルスロープ（+）、3 : デュアルスロープ（-） trg_level_hi : トリガレベル上限値 trg_level_lo : トリガレベル下限値
戻り値	正常終了時：=0、エラー時：エラーコード／負の値（エラーコード表6-4B参照）
機能動作	レンジトリガの極性を含む動作モード、レベルの設定。（レンジトリガ以外は前記【4-1】を使用） アナログトリガ検出対象は（複数ボード使用のときはマスタボードの）チャンネル0。

【5】マルチサンプリング動作モード設定

★同機能を使用しないときは不要。

書式	int AD68■_Set_MultiScan(WORD no_msamp, int do_average)
引数	no_msamp : マルチスキャン回数 / 1～（2の32乗）【注】 do_average : 平均化処理実行の有無 / 0 : しない、1 : する 【注】平均化処理をする場合、マルチサンプリング回数の最大値は255。
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6-4B参照）
機能動作	平均化処理を実行しないときの最終結果データは＜サンプリング実行チャンネル数／関数【2】＞、 ＜サンプリングデータ点数／関数【8】＞、および＜マルチスキャン回数＞の積となる。 平均化処理をハードで実行する場合に指定できるマルチスキャン回数の最大値は255。

図6-4A. マルチサンプリング動作の様子（4回の例）



《注1》マルチサンプリングクロック周期MCの最小値 = (使用チャンネル数) × (スキャン速度)
(最高速動作時) 関数【2】 関数【21】

《注2》サンプリングクロック周期SC最小値 = (使用チャンネル数) × (スキャン速度) × (マルチスキャン回数)
(最高速動作時) 関数【2】 関数【21】 関数【5】

【6-1】通常のサンプリングクロック用オプション、または外部クロック源周波数値の設定

書式	int AD68■_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6-4B参照)
機能動作	オプション、または外部クロック源 (CLK-IN) を使用するとき、その周波数値を設定する。 後記【7-1】ad_set_clock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。

【6-2】マルチサンプリング用・外部クロック源周波数値の設定 ★不使用時は設定不要。

書式	int AD68■_Set_Mexclk (int mexclk_freq)
引数	mexclk_freq : マルチサンプリング用・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6-4B参照)
機能動作	マルチサンプリング実行タイミングに外部クロック源 (汎用デジタル入力D0) を使用するとき、その周波数値を設定する。 後記【7-2】ad_set_msclock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。

【7-1】サンプリング・クロック値の設定

書式	int AD68■_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 (10MHz)、1 : 内部クロック源1 (8.192MHz) 2 : 内部クロック源2 (オプション)、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 : μs、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6-4B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、クロック源に2, 4, 5を指定し、クロック周期の値で設定するときは前記【6-1】ad_set_exclkでクロック源周波数値を定義しておく。

【7-2】マルチサンプリング・クロック値の設定 ★同機能を使用しないときは不要

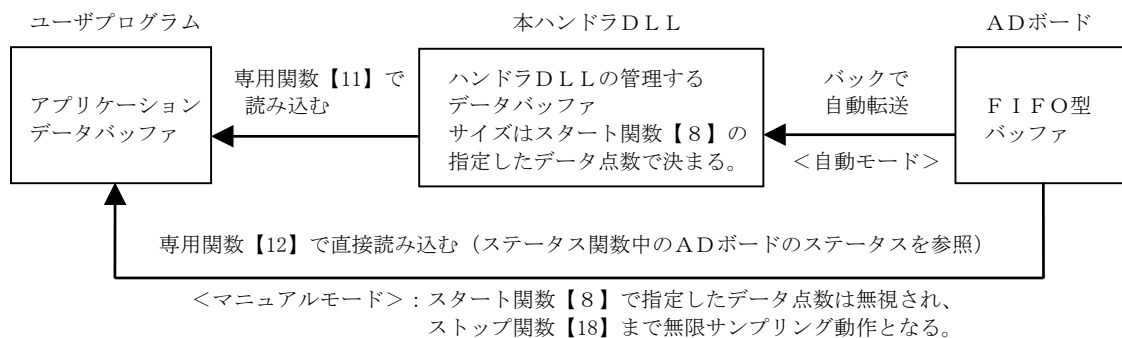
書式	int AD68■_Set_MsClock(int clk_source, int set_mode, int *time_unit, int *clk_period)
引数	clk_source : マルチサンプリング・クロック源選択。 ／0 : 内部クロック源0 (10MHz)、1 : 内部クロック源1 (使用不可) 2 : 内部クロック源2 (使用不可)、3 : 内部クロック源3 (使用不可) 4 : 外部クロック源 (有効極性=↓)、5 : 外部クロック源 (有効極性=↑) set_mode : クロック値の指定方法／0 : クロック周期の値、1 : 分周比 time_unit : クロック周期の単位 / 0 : s、1 : ms、2 : μs、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6-4B参照)
機能動作	マルチサンプリング・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。またクロック源に4, 5を指定し、クロック周期の値で設定するときは前記【6-2】ad_set_mexclk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・ADデータ読み込み方法

本ハンドラは関数【2】～【7】でサンプリング条件の設定後、関数【8】以降で実際のADサンプリング動作を実行します。

このADサンプリング動作実行時にボード上のFIFOバッファメモリから当ハンドラが自己の管理するバッファ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直接ボード上のFIFOバッファメモリから自己の管理するバッファにデータを読み込むマニュアルモードがあり、関数【3】“FIFOからのADデータ転送モード”の第一引数で指定します。

図6-4 B. 自動モードで使用するか、マニュアルモードで使用するか



- 自動モード**：
- 当ハンドラDLL自体がバッファを確保、これにADボード内で自動連続サンプリングされたADデータを専用スレッドによりバックで自動転送する。ユーザプログラムは表で別タスクを併行することができる。
 - 当バッファへのデータ読み込みは当ハンドラ自体がFIFOメモリの充満状態を示す【Half-Full】または【Not-Empty】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、本ボードの最高速度を実現できる。後者は1回サンプリングごとに刻々のADデータを評価して実行すべき処理があるような場合向きで、やや低速。
 - いずれの場合もユーザプログラム側からはステータス関数中のサンプリング済み点数を参照し、専用のデータ転送関数【11】で行う。
- 《メッセージ》→ AD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、AD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）、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**：
- ユーザプログラム内にバッファを確保し、これにADボード内で自動連続サンプリングされたADデータを【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】／等による割り込みイベントまたはポーリングで<FIFOメモリ容量の半分>または<1スキャン分>単位で行う。サンプリング停止関数実行まで無限サンプリング動作となり、割り込みイベントを利用した場合はプログラム側の処理が追いつかないと例外発生によりユーザプログラムが停止する危険がある。
- 《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1》→ 実際に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。
- 《注意2》→ 本モードはデュアル（マルチ）コアCPUパソコンでは使用不可。

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【11】【12】のデータ転送関数です。 以下に連続サンプリング（波形取り込み）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、当ハンドラ自体が自動的にADボードのF I F Oメモリからデータを“当ハンドラD L L管理バッファ”に読み込む。このとき、“ADデータ転送の参照フラグ”を《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《Not-Empty》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。
ユーザプログラムでは“サンプリング済みデータ点数”を参照して有効なデータを“当ハンドラD L L管理バッファ”から読み込む。
- <例2> 当ハンドラD L L管理バッファを“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D L L管理バッファ”の末尾と先頭が連結された構造となり、1 廻り以後は上書きされる動作となる。
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長時間の監視システムや、低速のプリトリガ動作などに利用できる。
- <例3> “マニュアルモード”で“スタート操作”を実行すると、AD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《Not-Empty》を検出してデータ読み込みタイミングを知り、【12】AD_Read_DirectFifo()で読み込む。このとき、“ADデータ転送の参照フラグ”を《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《Not-Empty》に設定の場合は“1 スキャン分”単位で読み込む。
当モードでは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例2と同様なアプリケーションに適用できる。

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”	
“ADデータ転送の参照フラグ”	
“スタート操作”	【8】連続サンプリング・スタート関数
“当ハンドラ D L L管理バッファ”	ユーザプログラム側へは【11】AD_Read_DllData()で転送、 容量は【2】指定チャンネル数×【8】指定データ点数/ch
“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】指定チャンネル数を各1回サンプリング分
“サンプリング済みデータ点数”	【9】ステータスから得る
“エンドレス・リング状”	【16】D L L管理バッファをリング状・エンドレスに設定
“ストップ操作”	【18】連続サンプリング動作の強制停止関数の実行

《追伸》 ユーザプログラム上でADボード上のF I F Oメモリ容量をソフト認識したいときは<例3>を“マニュアルモード” “ADデータ転送の参照フラグ”を《HALF-FULL》とし、ユーザプログラム上のバッファ容量を変化させて実行したとき、同容量がF I F Oメモリの《HALF-FULL》サイズ未満なら【11】AD_Read_DirectFifo()実行がエラーとなることから知ることができる

【8】サンプリング・スタート

書式	int AD68■_Start_Samp (int no_samp)
引数	no_samp : トリガ後のサンプリング・データ点数 (1チャンネル当りの点数)
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 6-4 B 参照)
機能動作	<p>関数【1】～【7】で設定した条件で連続・自動サンプリングを実行する。</p> <p>前記【4-1】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。</p> <p>前記【4-2】で設定するレンジトリガのときは全てトリガ待ちスタートとなる。</p> <p>なお前記【3】でサンプリングモードを自動に設定した場合、本DLLで確保されるデータバッファサイズ (1ボード分) は当関数で指定する1チャンネル当りサンプリング・データ点数と前記【2】で指定されるサンプリング実行チャンネル数の積となる。 なおマルチスキャン機能を使用する場合は、さらに前記【5】で指定されるマルチスキャン回数を乗じた値となる。</p>

【9】ステータス取得／ (マスタボード)

書式	int AD68■_Get_Status (int *sampled, int status[])
引数	sampled : サンプリング済みデータ点数 (1チャンネル当り) / DLLバッファ status [0] : ADボードのステータス1 / (3-12項) status [1] : ADボードのステータス2 / (未使用: 本ボードでは無い)
戻り値	正常終了時 : = 0 : 連続サンプリング実行中、 = 1 : 停止中。 エラー時 : エラーコード／負の値 (エラーコード表 6-4 B 参照)
機能動作	<p>当該時点でのサンプリング済みデータ点数 (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。</p> <p>マニュアルモードのときはステータス生データのみの有効。</p>

【10】本ハンドラの終了

書式	int AD68■_Close_ADsys (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 6-4 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。

【11】DLL管理バッファからユーザプログラム内バッファにADデータを転送 / (ボードごと)

書式	int AD68■_Read_DLLData(int board_no, int no_data, int data_pos, 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_data : DLL管理バッファから読み出すデータ点数 (1チャンネル当り) data_pos : DLL管理バッファから読み出すデータの先頭位置 (サンプリング順番号) bufptr : 転送先ADデータバッファのポインタ bufsize : 転送先ADデータバッファの大きさ (バイト指定=総データ数の2倍)
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 6-4 B 参照)
機能動作	<p>サンプリングされたADデータがDLL管理バッファに格納されているとき、任意の部分をユーザプログラム内のデータバッファに転送する。 読み出すデータの末尾はステータス関数で得る<サンプリング済みデータ点数>の値以内でなくてはならない。</p> <p>【注】当方法は関数【3】サンプリング動作モード設定で自動モードを指定したときだけ有効。</p>

【12】FIFOからユーザプログラム内バッファに直接、ADデータを読み込む／（ボードごと）

書式	int AD68■_Read_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 : 転送先ADデータバッファのポインタ bufsize : 転送先ADデータバッファの大きさ（バイト指定＝総データ数の2倍）
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6-4B参照）
機能動作	ADボード上のFIFOメモリから直接、ユーザプログラム内のデータバッファに転送する。 一般的に当関数は割り込みイベント（【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ADデータ転送の参照フラグにNot-Emptyを指定したときは1回サンプリングスキャン分のADデータを転送する。 また、Half-Fullを指定したときはADボード上のFIFOメモリ容量の半分を転送する。 【注】当方法はサンプリング動作モード設定【3】でマニュアルモードを指定したときだけ有効。

【13】マニュアル（1回）サンプリング・スキャン／（全ボードについて実行）

書式	int AD68■_Get_OneScan (WORD *bufptr, int bufsize)
引数	bufptr : ADデータバッファのポインタ bufsize : ADデータバッファの大きさ（バイト指定＝総データ数の2倍）
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6-4B参照）
機能動作	前記【2】で指定した全ボード、全チャンネル群に対して各1回だけサンプリングを実行する。 得られたADデータは全て、ここで指定した1個の専用バッファに格納される。 関数【5】で設定するマルチサンプリング&平均処理機能も有効であるが、関数【6-2】および【7-2】で指定するマルチサンプリングクロック値は無効で、間断ない繰り返しスキャン動作となる。 複数ボード使用時の並び順は先頭がボード番号0（マスタ機）のチャンネル0, 1, 2, ～15で、以下、ボード番号1～7（スレーブ各機）のADデータが同チャンネル順に続く。

【14】汎用デジタル（ラッチ）出力／（ボードごとに実行）

書式	int AD68■_Out_Aux (int board_no, int out_data)
引数	board_no : 対象ボード番号 out_data : 汎用1ビット・デジタル（ラッチ）出力データ。
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6-4B参照）
機能動作	各ボードごとの汎用1ビット（ラッチ）出力データの更新。 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（ソフトのリセット）では変化しない。

【15】汎用デジタル（現在値）入力／（ボードごとに実行）

書式	int AD68■_Inp_Aux (int board_no)
引数	board_no : 対象ボード番号
戻り値	正常終了時 : =汎用1ビット・デジタル（現在値）入力データ。 エラー時 : エラーコード／負の値（エラーコード表6-4B参照）
機能動作	各ボードごとの汎用1ビット・デジタル（現在値）入力データ読み込み。

【16】DLL管理バッファをリング状/エンドレスに設定

書式	int AD68■_Set_SampLoop (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 6-4 B 参照)
機能動作	<p>DLL管理ADデータバッファをリング状・エンドレスで使用するか否かの設定。(トグル)</p> <p>リング状・エンドレスに設定した場合、自動モードでDLL管理データバッファが満杯になると次は同バッファ先頭を上書きする形となる。この時、ステータス取得関数【8】で得るサンプリング済みデータ点数の値も先頭値(ゼロ)に戻る。</p> <p>本設定により当ハンドラは無限サンプリングモードになる。</p> <p>長時間の監視システム等に利用できる。</p>

【17】アナログ入力モードの設定 / (全ボード共通)

書式	int AD68■_Set_Inpmode (int inp_mode, int ad_code, int ad_reso, int ad_rangemode)
引数	<p>inp_mode : アナログ入力信号形式 (本ハンドラではダミー) / ボード上のスイッチ設定</p> <p>ad_code : ADデータコード指定 / 0:バイナリ、1:2の補数</p> <p>ad_reso : AD変換分解能 (本ハンドラではダミー) / ボード上のスイッチ設定</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 6-4 B 参照)
機能動作	アナログ入力信号形式、変換出力データ・コード(符号化の型)、およびAD変換分解能を指定する。

【18】サンプリング動作の強制停止

書式	int AD68■_Stop_Samp (void)
引数	なし
戻り値	<p>正常終了時 : = 0</p> <p>エラー時 : エラーコード/負の値 (エラーコード表 8-4 参照)</p>
機能動作	<p>サンプリングを強制的に中止する。</p> <p>(ADボード上のFIFOバッファメモリに残りデータがあれば、それらは無効となる。)</p>

【19】残りADデータの読み込み / (全ボードについて共通に実行する)

書式	int AD68■_Read_RestData (void)
引数	なし
戻り値	<p>正常終了時 : =読み込んだ(残り)データ数</p> <p>エラー時: エラーコード/負の値 (エラーコード表 6-4 B 参照)</p>
機能動作	<p>データロス・エラー発生(当ハンドラが検出するとサンプリングを中止させる)したとき、ADボード上のFIFOバッファメモリに残りデータがあれば、これを当関数で読み込むことができる。</p> <p>なお、前【17】サンプリングの強制停止を実行すると残りデータは読み出しできず、無効となる。</p>

【20】(ADボード) 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 AD68■_Clear_Flags (int c__data)
引数	c__data : クリアビット指定データ。/ 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 6-4 B 参照)
機能動作	ADボード上の各フラグをクリアする。

【21】 サンプリングスキャン速度指定（初期は低速モード）

書式	int AD68■_Set_ScanSpeed (int scan_speed)
引数	scan_speed : サンプリングスキャン速度指定／ 0 : 低速 (10 μ s / ch) 1 : 高速 (5 μ s / ch)
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 6-4 B 参照)
機能動作	動作の詳細については (3-1 項) を参照。

【22】 外付・同時サンプルホールド制御／（各ボードごとに指定）

書式	int AD68■_Set_Shc (int board_no, int shc)
引数	board_no : 対象ボード番号。(0～7)／本関数は各ボードごとに行う。 shc : 外付・同時サンプルホールド制御／ 0 : しない、 1 : 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 6-4 B 参照)
機能動作	複数チャンネル同時サンプルホールド回路を外付・接続する場合の各サンプリングスキャン開始タイミング制御。 shc = 1 のときは 1 μ s 遅らせることで先頭チャンネル (ch 0) にかかるホールドステップ誤差を最少化するが、スキャン時間が 1 μ s 増加するため最高サンプリング速度が遅くなる。 shc = 0 のときは最高サンプリング速度で動作するが、スキャン先頭チャンネル (ch 0) に数 mv のホールドステップ誤差が発生する。(3-1 8 項参照)

【23】（オプション的な）メッセージ発信の設定

書式	int AD68■_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 6-4 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 (AD ボードからデータを読み込むための) 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 (AD ボードからデータを読み込むための) 参照フラグを検出し、所定の読み込み動作を行った直後にメッセージを発信する。

【24】 本ハンドラのバージョン取得

書式	int AD68■_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 8-4 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6-4B. エラーコード一覧

(■■■■=ボード番号)

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?)		AD■■■■_Open_ADSys()
- 2	ボードIDが違う。		AD■■■■_Open_ADSys()
- 3	FIFOバッファにアクセスできない。		AD■■■■_Open_ADSys()
- 4	ドライバファイルを検出できない。		AD■■■■_Open_ADSys()
- 5	ドライバファイルのバージョンが不適切。		AD■■■■_Open_ADSys()
- 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。		AD■■■■_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	サンプリング実行チャンネル数	指定パラメータの不適当	no_ch
- 15	AD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	ADデータ転送先	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
- 22	トリガレベル (レンジトリガ以外)	指定パラメータの不適当	trg_level
- 23	トリガレベル (レンジトリガ)	指定パラメータの不適当	trg_level_hi/_lo
- 24	トリガレベル上限値・下限値の大小関係不適当		trg_level
- 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	
- 32	ADデータ・コード	指定パラメータの不適当	data_code
- 35	外付・同時サンプルホールド	指定パラメータの不適当	shc
- 36	アナログ入力スキャン速度	指定パラメータの不適当	speed
- 41	外部クロック源周波数を設定していない。		AD■■■■_Set_Clock
- 42	ADデータバッファ・サイズが小さすぎる。		bufsize
- 43	1回サンプリングスキャン実行時、チャンネル数設定未実行		AD■■■■_Get_OneScan
- 44	bufptrエラー		bufptr
- 45	サンプリングされたADデータ数を超えて読もうとした。		AD■■■■_Read_Data
- 46	割り込みリソースが割り当てられていない。		AD■■■■_Set_SampMode
- 47	割り込み設定エラー (重複など)		AD■■■■_Set_SampMode
- 49	メモリ確保エラー		
- 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。		AD■■■■_Read_DllData
- 51	本ハンドラDLL内のデータバッファが指定されている。		AD■■■■_Read_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。		
- 53	現在、サンプリング中である。		AD■■■■_Read_RestData

6-5. 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【11】【12】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に 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ADボードのFIFOメモリからデータを“当ハンドラDLLの管理するバッファ”に読み込む。</p> <p>このとき、“AD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、《Not-Empty》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、指定トリガ後サンプリング点数に達すると自動停止する。</p> <p>ユーザプログラムでは“サンプリング済みデータ点数”を参照して有効なデータを当ハンドラDLLの管理するバッファから読み込む。</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に 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内のバッファ”の末尾と先頭が連結された構造となり、1 廻り以後は上書きされる動作となる。</p> <p>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【9】ステータスから得る)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長時間の監視システム、低速のプリトリガ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に 有限／無限サンプリング 。	Sample3m.vbp	Visual Basic (5,6)
	Sample3m.c	Visual C (5,6)
	Sample3m.c	Borland C (5)
<p>“マニユアルモード”で“スタート操作”を実行すると、AD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《Not-Empty》を検出してデータ読み込みタイミングを知り、【12】AD_Read_DirectFifo()で読み込む。</p> <p>このとき、“AD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、《Not-Empty》に設定の場合は、“1スキャン分”単位で読み込む。当モードで指定できるトリガ後サンプリング点数は、【8】スタート時に指定する有限値であるが、【16】無限サンプリングモード設定操作を行うと、“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
“マニユアルモード”	
“ADデータ転送の参照フラグ”	
“スタート操作”	【8】連続サンプリング・スタート関数
“当ハンドラDLL管理 バッファ”	ユーザプログラム側へは【11】AD_Read_DllData()で転送、容量は【2】指定チャンネル数×【8】指定データ点数/ch
“FIFOメモリ容量”	【1】初期化のとき自動検出
“1スキャン分”	【2】指定チャンネル数を各1回サンプリング分
“サンプリング済みデータ点数”	【9】ステータスから得る
“エンドレス・リング状”	【16】DLL管理バッファをリング状・エンドレスに設定
“ストップ操作”	【18】連続サンプリング動作の強制停止関数

□■ 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（ADボード）とソフト（ハンドラ）設定の不整合。
- ◆AD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【AD68■__Open__ADSys（）】で使用開始、
【AD68■__Close__ADSys（）】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《ボード上のFIFOバッファ》～《パソコン側データバッファ》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

ADボードの自動サンプリング動作でFIFOバッファメモリにADデータが流入する速度よりパソコン側から読み出す速度が遅く、FIFOバッファメモリが満杯になったうえに次のデータが書き込まれようとする、そのデータはこぼれ（失われ）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が、それ以前のデータ（FIFOバッファメモリ中にある）は残りデータとして専用関数【19】で有効に読み出すことができます。

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&データ読み込み（ブロック転送）に専念した場合は本ADボード自体の最高サンプリング速度が実現可能です。

- (5) 直接I/O操作を行うとき

本ハンドラを介さずAD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7章. 保守・その他

7-1. 故障・トラブル等の原因と対処

本機は【DOS/V系パソコン】＋【拡張ボックス】のシステム構成で全数検査のうえ出荷されています。お手元での動作確認方法は1－6項に記されています。動作に不具合があ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 | ◆ ボードのインストール／認識は成功したか？（1－5項） |
| (2) 割り込みレベル | ◆ リソースは取得できたか？（1－5項） |
| (3) デジタル入出力 | ◆ 本ボードのTTL入力（外部割り込み、および汎用ビット）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。（次ページ参照） |
| (4) アナログ入力 | ◆ 過電圧入力保護：±3.5V以内。
◆ 複数チャンネル使用時は各信号源のGND間電位差に注意。 |

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。動作確認は本製品添付の当社製プログラム（1－6項）の実行結果について推測・適否・判定を行います。

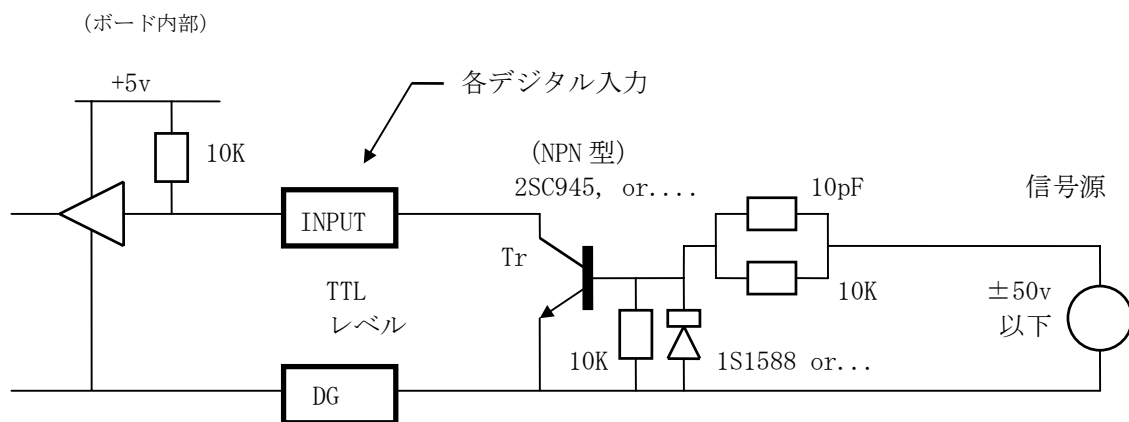
QAリクエスト時には当プログラムの実行結果をレポートしてください。

ボード内TTL入力素子破損の主な原因

TTL入力素子の絶対最大定格は【負側：-0.6V】【正側：+7V】です。このレベルを一瞬でも超えると入力素子破壊の原因になります。主な危険要素は、

- ◆ ファンクション・ジェネレータ等の交流信号出力を接続して破損させる例が多いようです。矩形波でも±に振れる信号は接続できません。特に、負側の許容レベル【-0.6V】が低いことに注意してください。
- ◆ +5V以上に振れるロジック信号も接続できません。12V～24V電源を使用する機器からのデジタル信号は不可、信号レベルが不明なときは信号源の電源電圧が目安になります。
- ◆ アナログ信号源は±15V電源によるオペアンプ出力が多く危険です。なお、TTL入力にアナログ信号を接続しても立上り／立下り特性等が仕様を満足せず、正常な動作は期待できないでしょう。
- ◆ 信号源と本ボードのグラウンド・レベルに差があるときも危険です。（テストで測定可能）

図7-1. 【高レベル信号】→【TTLレベル】変換回路例



《注》本回路はインバータ（極性反転）です。

7-2. 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・

有償にて修理をお願いします。 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 : 宅配便によるセンドバックで行います。

◆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。 2007年3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。

7-3. アナログ入力範囲の再調整

動作テスト・確認の方法は【1-6項】のとおりです。同テストから得られた値に入力範囲の変化やオフセットが認められるときは再調整が必要です。アナログ回路は経年環境変化に対する保守を定期的に行うことが望ましく、夏冬の使用環境・周囲温度に差がある場合は季節単位、通年安定した使用環境の場合は1～2年に1度は校正することが理想的です。

再調整の方法・手順を以下に記しますが、極細のドライバ、デジタル電圧計を必要とし、手順もやや複雑ですから御希望により当社でも（実費で）お請けします。

== 準備 ==

- ① 本ボード上の諸設定は出荷時の状態（1-2項）とします。
- ② 本ボードをパソコン本体または拡張I/Oボックスに装着・インストールします。
当作業の詳細については1-5項に記されています。
（以上、通常の使用状態）
- ③ 図1-6のように、本ボード全チャンネルのアナログ入力を基準電圧源に接続します。
デジタル入出力の接続は不要です。
- ④ 以上で準備完了です。電源投入順序は全機器同時、または外部機器を先にパソコン本体を最後に行います。電源切断は逆順序です。

== 操作 ==

- ⑤ 電源を投入、MS-DOSシステムを立ち上げます。再調整に使用するプログラムは試運転でも使用した“68■QB2”です。>68■QB2【ENTER】でプログラムが走り始めます。最初にI/Oベースアドレスの入力を要求されますから1-5項で調査した4桁のHex値を入力すると以下、操作メニューとなります。

【注】 ファイル名： ADM-686zPCI用は■=6
ADM-687zPCI用は■=7

- ⑥ 外部基準電圧源を操作し、本ADボード設定入力範囲の下限付近でオフセット調整を、上限付近でゲイン調整を、目標値を得るよう交互に2～3回繰り返して最適位置を求めます。
調整はユーザシステムに都合のよい入力範囲で実施してください。なお、本機の標準出荷時はAモード±10V入力範囲で最適調整されています。

以後の操作は、

【SPACE】キー： 中断／再開

【ESC】キー： 終了

本プログラムではADデータを16進数（Hex）で表示します。

調整はアナログ入力範囲の《最大値付近》と《0v》の2点間で行います。本機のアナログ回路／AD変換器は $\pm 0.004\%FS$ ($\pm 1.5LSB$) 以内の非直線性を持っており、2点間調整だけで設定入力範囲の全域に渡って $\pm 0.008\%FS$ の精度 (=相対正確度) を実現する能力を持っています。／誤差=0の信号源で校正したときに可能な正確度です。／

この相対正確度 (=校正可能限度) に校正に使用した測定器の正確度を積算した値が絶対正確度です。当社内での調整は $0.015\%FS$ のデジタル電圧計を使用して常温で行っています。

また高精度部品の使用により調整を実施した入力範囲以外に設定を変更してもレンジ間誤差は $\pm 0.02\%FS$ 以内です。

したがって製造時点での絶対正確度は、

最終調整 $\pm 10v$ 範囲／Aモードで $0.023\%FS$ 、
その他の入力範囲で $0.043\%FS$
です。

なお経年変化のデータ／保証は無く、システム内部雑音・温度ドリフトにも御注意ください。

= 調整手順 =

- ①オフセット調整 : 入力電圧が0vのとき、AD変換値(表示)が8000HとなるようTM0を調整します。
(バイポーラ入力範囲のとき)
- ①オフセット調整 : 入力電圧が0vのとき、AD変換値(表示)が0000HとなるようTM1を調整します。
(ユニポーラ入力範囲のとき)
- ②ゲイン調整 : 入力電圧が最大値付近のとき、AD変換値(表示)が整合した値となるようにTM2を調整します。

[①オフセット調整 ②ゲイン調整] を2～3回繰り返して所定の精度に追い込みます。

【注】電源ONから調整実施までに**10分以上**のウォームアップ時間をとって下さい。

表7-3C. 標準Aモードの調整対象・目標

設定入力範囲 →→		$\pm 10v$	$\pm 5v$	$\pm 2.5v$	$0 \sim +5v$	$0 \sim +10v$
オフセット調整	基準入力	0v	0v	0v	0v	0v
	設定目標	8000H	8000H	8000H	0000H	0000H
	調整トリマ	TM0	TM0	TM0	TM1	TM1
ゲイン調整 (スケール)	基準入力	+10v	+5v	+2.5v	+5v	+10v
	設定目標	E1A8H	E1A8H	E1A8H	C350H	C350H
	調整トリマ	TM2	TM2	TM2	TM2	TM2

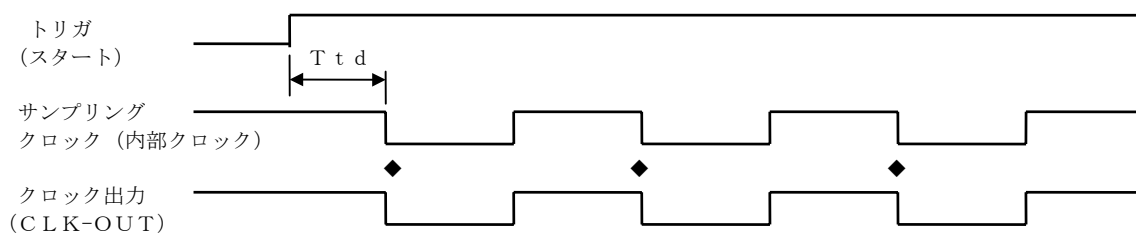
表7-3D. 非標準Bモードの調整対象・目標

設定入力範囲 →→		$\pm 10v$	$\pm 5v$	$\pm 2.5v$	$0 \sim +5v$	$0 \sim +10v$
オフセット調整	基準入力	0v	0v	0v	0v	0v
	設定目標	8000H	8000H	8000H	0000H	0000H
	調整トリマ	TM0	TM0	TM0	TM1	TM1
ゲイン調整 (スケール)	基準入力	+9.99969v	+4.99985v	+2.49992v	+4.99992v	+9.99985v
	設定目標	FFFFFH	FFFFFH	FFFFFH	FFFFFH	FFFFFH
	調整トリマ	TM2	TM2	TM2	TM2	TM2

7-4. 制御信号・タイミング等

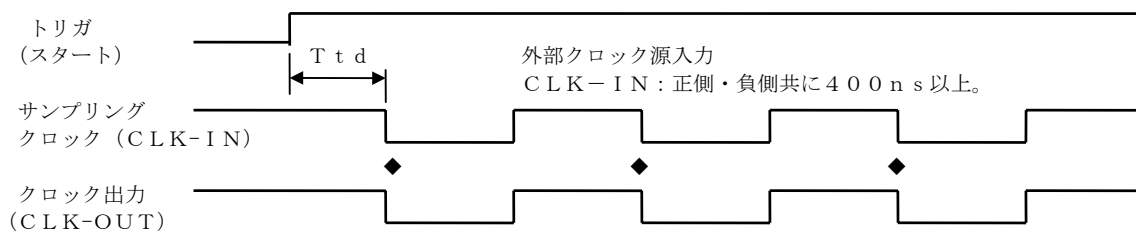
クロック入出力

図7-4 A. 内部クロック源を使用する場合



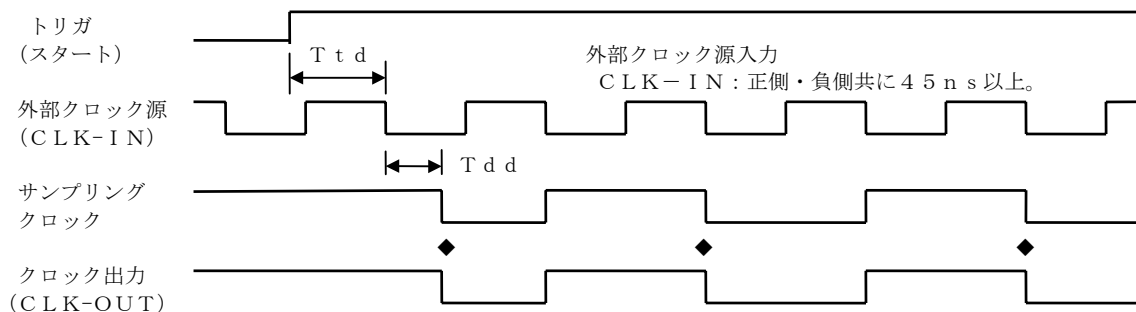
◆ : クロック出力の有効エッジは常に立下り。
 T_{td} : トリガ認識から最初のクロック有効エッジまでの最大遅れ時間 (最大 250 ns)
 実際のサンプリング実行はクロックの各有効エッジから 125 ns 以内に開始。

図7-4 B. 外部クロック源を非分周 (1 / 1) で使用する場合



◆ : クロック出力の有効エッジは常に立下り。
 T_{td} : トリガ認識から最初のクロック有効エッジまでの最大遅れ時間 (外部クロック源 1 周期)
 実際のサンプリング実行はクロックの各有効エッジから 125 ns 以内に開始。

図7-4 C. 外部クロック源を (任意に) 分周して使用する場合



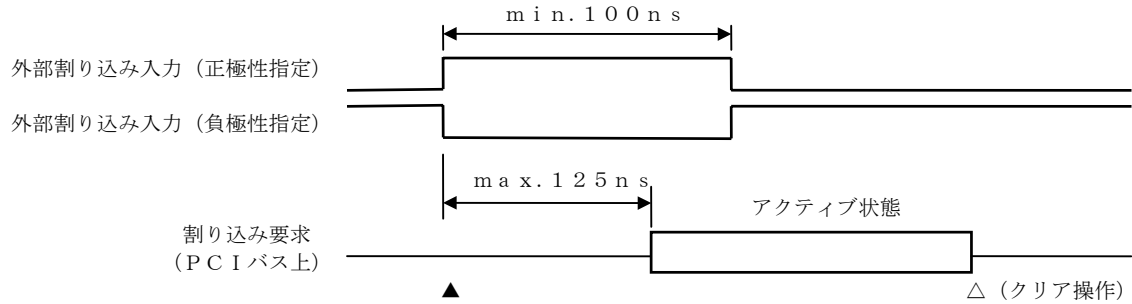
◆ : クロック出力の有効エッジは常に立下り。
 T_{td} : トリガ認識から最初のクロック有効エッジまでの最大遅れ時間 (外部クロック源 1 周期)
 実際のサンプリング実行はクロックの各有効エッジから 125 ns 以内に開始。

T_{dd} : 分周開始の遅れ時間 (最大 250 ns)

外部割り込み入力

 (許可されている場合／3-14項)

図7-4D. 外部割り込み入力 ～ 割り込み受け付け



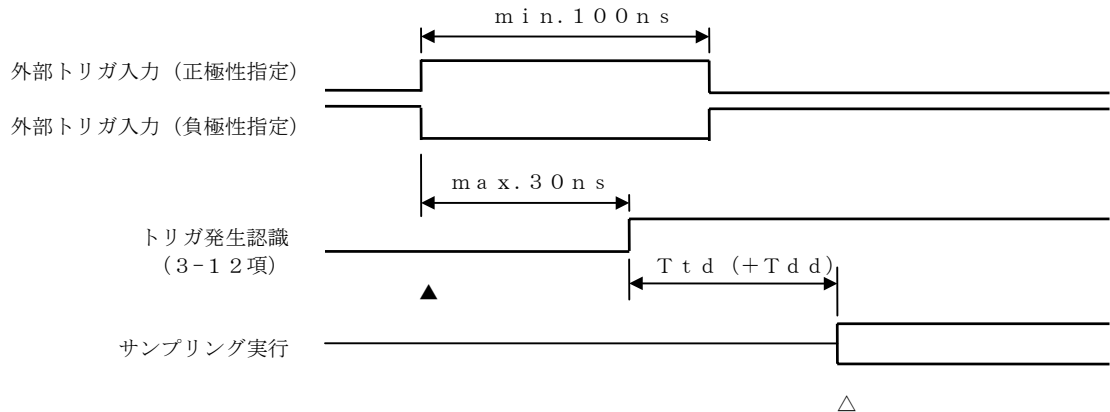
▲ : 外部割り込み信号入力 (INT-IN) の有効エッジ。

△ : 割り込みクリア操作タイミング。(通常、デバイスドライバ内で実行する)

外部トリガ入力

 (許可されている場合／3-10項)

図7-4E. 外部トリガ ～ 連続サンプリング開始



▲ : 外部トリガ信号入力 (TRG-IN) の有効エッジ。

△ : 連続サンプリングの開始タイミング。

Ttd : 図7-4A, B, C参照。

Tdd : 図7-4C参照。

LS型TTL素子の基本規格

 : 本機のデジタル入出力素子
入力 : $I(IL) = -0.4\text{ mA}$ …… Lレベル (+0.4V ~ 0V) 印加時の流れ出し電流 $I(IH) = 20\text{ }\mu\text{ A}$ …… Hレベル (+2.7V ~ +5V) 印加時の流れ込み電流出力 : $I(OL) = 8\text{ mA}$ …… Lレベル (+0.4V ~ 0V) 出力時の流れ込み可能電流 $I(OH) = -0.4\text{ mA}$ …… Hレベル (+2.7V ~ +5V) 出力時の流れ出し可能電流

マイクロサイエンス（株）行

FAX：03（3301）5593

Q & A フォーム

発信： 年 月 日／ 時 分

製品名	ADM- PCI	購入時期	年 月	
ボード上の 設定、 使用状況				
その他				
I/O、 周辺状況	同時使用の 他ボード		I/Oアドレス 割り込み、等	
本体 システム	パソコン本体		拡張BOX	
	本体メモリ			
	OS	DOS () WIN ()		
ソフト	言語		コンパイラ	(v r)
	プログラム名			
(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）8362

御使用者		(所属部・課)
団体名		
TEL		(所在地)
FAX		