

## 第4章．ソフトウェア

《注1》WINDOWS 98 / ME対応： 特に断りのない限り、WINDOWS 95用のソフトがそのまま使用できます。

《注2》WINDOWS XP対応： 特に断りのない限り、WINDOWS 2000用のソフトがそのまま使用できます。

### 4-1. ソフトウェアのインストール

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： WINDOWS 2000 / XP版 LabDAQ - AQです。  
(アプリケーション) 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： 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、  
(WIN32ライブラリ) 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】

サンプルソフト： ADボード制御プログラムを手造りするユーザ向けの操作例です。  
(DOS版/ソース) C、およびQuick-Basicで例示しています。  
サンプリング、割り込み等の使用方法を学習することができます。

その他： WINDOWS (9x/ME/NT/2000/XP) でI/O読み書き  
(for WINDOWS) 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-67X ¥16,000)

= 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 4 月版以降の ) C D R O M を使用します。

**操作手順**

インストール元 : D ドライブ ( C D R O M )

インストール先 : C ドライブ ( H D D )

の場合で例示。

( 1 ) W I N D O W S 付属のエクスプローラで、  
D : ¥ I N S T A L L ¥ P C I ¥ A D ¥ A D M 6 7 0 を開く。

( 2 ) “ S e t u p . E X E ” を実行 ( ダブルクリック ) する。

当操作以下により A D M - 6 7 0 P C I 関連プログラムが下図 4 - 1 A に示す  
ロケーションに展開・インストールされます。

< 2005 年 4 月以前の旧 C D R O M を利用する場合 >

**操作手順**

インストール元 : D ドライブ ( C D R O M )

インストール先 : C ドライブ ( H D D )

の場合で例示。

( はスペース )

C : ¥ W I N D O W S > C D ¥ 【ENTER】

C : ¥ > C D D : ¥ I N S T A L L ¥ P C I ¥ A D ¥ A D M 6 7 0 【ENTER】

C : ¥ > D : I N S T A L L D : C : 【ENTER】

各プログラムグループ ( C , B A S I C 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】 = y e s , 【N】 = n o , で答える  
だけで作業が進みます。

《注》 M S - D O S の環境変数 “ C O M S P E C ” 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 = C O M M A N D . C O M が C ドライブの ¥ にある場合、

> S E T C O M S P E C = C : ¥ C O M M A N D . C O M 【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 の  
ようになります。

【注 1】 本ボード専用の W I N D O W S 版ハンドラ D L L / デバイスドライバは当作業の後、  
6 - 3 項に従って必要なファイルを適合フォルダにコピーする必要があります。

【注 2】 ボード依存性のない汎用の W I N D O W S 版 I / O 実行 D L L / 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。 W I N D O W S 9 5 ・ 9 8 用は W i n 9 x 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。

W I N D O W S N T 用は W i n N T フォルダ中にあり、同フォルダ中の専用インス  
トラードで導入してください。( 1 - 5 項参照 )

( 追伸 ) C D R O M の場合、 W i n 9 x および W i n N T フォルダは I N S T A L L  
フォルダ下の D r i v e r s フォルダ下にあります。

図4-1. インストール後のディレクトリ

凡例 MS - C : MICROSOFT - Cの略

T - C : TURBO - Cの略

B - C : BORLAND - Cの略

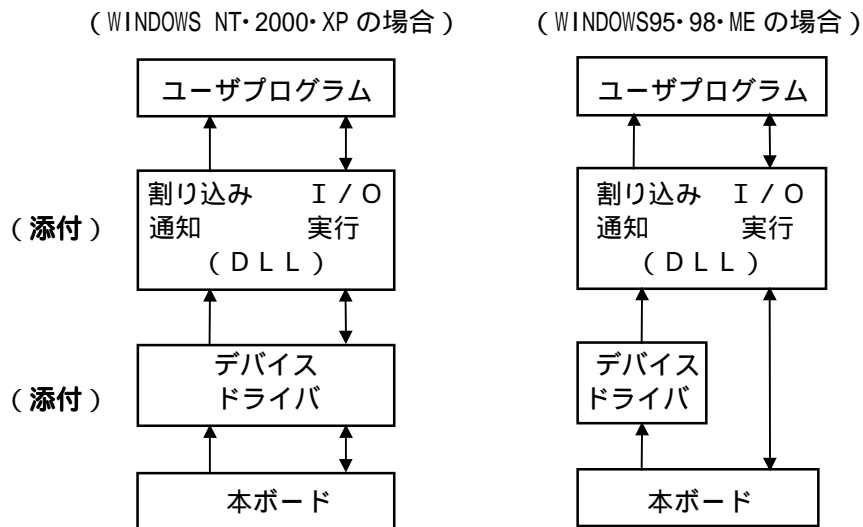
¥			
MSCIENCE			
	UTILITY (本ボードの設定)	WIN95	CF9050DP.COM : 設定変更ユーティリティ (95用)
		- WINNT	CF9050NT.EXE : 設定変更ユーティリティ (NT用)
			- CF9050NT.sys : 上記NT用デバイスドライバ
			- REGSTDV.EXE : ドライバの登録ユーティリティ
	BOARDTST	- - - 670QB2.EXE	: 本ボードの試運転・動作確認用プログラム
		- 670QB2.COM	: 英語モードに切り替えた後、EXEを実行する
	- - SMP670C - - -	MICROSOFT.H	: MS - C用ヘッダ
	(Cの各種サンプル)	- BORLAND.H	: TURBO - C, BORLAND - C用ヘッダ
		- ADM670.H	: 共通ヘッダ
		- POL670E1.C	: ポーリング動作例 (エンプティ解消フラグ使用)
		- BLK670H1.C	: ポーリング動作例 (ハーフフル・フラグ使用)
		- INT670E1.C	: 割り込み動作例 (エンプティ解消フラグ使用)
		- MSV670.C	: マスタスレーブ動作例 (ハーフフル解消フラグ使用)
		- PRE670.C	: プリトリガ動作例
		- MNL670.C	: マニュアル (1回) サンプリング動作例
		- PCI.C	: PCI関連ルーチン
	- - SMP670B - - -	670QB1.BAS	: 内部クロックによる連続サンプリング例
	(Q-Basic)	- 670QB2.BAS	: マニュアル・サンプリング例
			(試運転・動作確認用プログラムのソース)
	Hnd__95	Adm670	Dll: ハンドラDLL
	(WINDOWS 9x・ME用ハンドラ)	- Vxd	: デバイスドライバ
		- 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: Visual C++ (5.0)	: 用サンプル
		- Delphi3: Delphi (3.0)	: 用サンプル
	Hnd__NT	Adm670	D670reg: デバイスドライバ設定ユーティリティ
	(WINDOWS-NT用ハンドラ)	- Dll	: ハンドラDLL
		- Sys	: デバイスドライバ
	Hnd__2K	Adm670	Dll: ハンドラDLL
	(WINDOWS 2000/XP用ハンドラ)	-	
	GL__QB	QLB: クイックライブラリ、LIB: スタンドアロンライブラリ、	
	(ワードアクセス用/DOS)	BI: ライブラリ用インクルードファイル	
	GL__DOS	MSPCID.H	: (DOS版) リソース取得ライブラリ・ヘッダ
		- MSPCIDM.LIB	: (MS - C用) ライブラリ/全モデル対応
		- MSPCIDT.LIB	: (T - C, B - C用) ライブラリ/全モデル対応
	GL__W32	MS__PCI.H	: (Win32版) リソース取得ライブラリ・ヘッダ
	(95・98・NT用)	- MS__PCI.DLL	: リソース取得ライブラリDLL
			(Win95・98・NT兼用、要デバイスドライバ)
		- MS__PCI.LIB	: DLLインポートライブラリ
		- MS__PCI.BAS	: (VB/32bit版用) DLL関数定義モジュール

各サンプル等は上記の  
WINDOWS 9x・MEと  
同様です。

## 4-2 . W I N D O W S ドライバについて

W I N D O W S 9 x / M E / N T / 2000 / X P 向に汎用 I / O 読み書き用 D L L が添付されています。

基本的には当 D L L ( およびデバイスドライバ ) を使用して本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。 これらは御自身で ( ボードにアクセスする部分の ) ライブラリ等を製作する場合への便宜です。 添付の本ボード専用 W I N D O W S ドライバ / 関数ライブラリ ( 第 6 章 ) を利用する場合は不要です。



**W I N D O W S 3.1 :** W i n 3 1 フォルダ以下に格納されており、 V B ( 2.0 ) で利用できます。 C , C + + の場合は当 D L L を使用せずともインラインアセンブラで直接 I / O 命令を記述できます。 割り込みを使用するときは、 D O S 同様に直接制御で対応できます。

**W I N D O W S 9 x :** W i n 9 x フォルダ以下に格納されており、 V B ( 4.0/5.0 ) で利用 ( or M E ) できます。 ブロック I / O 命令もサポートされています。 C + + , C の場合は当 D L L を使用せずともインラインアセンブラで直接 I / O を記述できます。 割り込みは D O S 同様に直接制御、またはデバイスドライバ ( Pta95\_0.vxd ) で対応します。

**W I N D O W S N T :** W i n N T フォルダ以下に格納されており、 V B ( 5.0 ) で利用 ( 4.0 ) できます。 ブロック I / O 命令、割り込みもサポートされています。

N T では I / O 制御、割り込み、共に必ずデバイスドライバが必要です。

本デバイスドライバは最大 1 6 枚のボードを ( 各単独に ) 制御することができます。 / 当社製品でなくても可能 /

**W I N D O W S 2000 :** W i n 2 K フォルダ以下に格納されており、 V B ( 6.0 ) で利用 ( or X P ) できます。 なおドライバ ( W D M ) は複数種類のボードで共用利用できるもので、第 6 章で説明する本ボード専用ハンドラ関数 D L L から利用します。 ( ボードインストール作業時にインストールされます。 )

【注】 W D M ドライバの性格から割り込みは使用できません。

詳細は¥M S C I E N C E ¥ W I N 2 K ¥ D O C フォルダ内のテキスト参照。

## 4-3. ボードアクセス関連ライブラリ

汎用リソース情報取得関数 `MS_PCI_DLL` / WINDOWS 95・98・NT 版（2000・XP では不要）

ユーザプログラムから本 P C I ボードにアクセス・制御するときはボードを検出し、リソース情報（ベースアドレス値、割り込み番号等）を取得する必要があります。第 6 章に記す専用のハンドラ（専用ドライバ&関数 `DLL`）を使用する場合は内部で処理されているため、必要ありませんが、ユーザ・プログラムから本ボードを直接制御するときは本 `DLL` & ドライバが必要になります。使用手順は、

### 【1】P C I バス上のボードの検出

Int GetPciDevice (WORD VendeID, WORD DeviceID, WORD nNum, WORD Flag, WORD *magic)	
引数	VendeID : ベンダ ID、nNum : 検出対象ボード (0 ~ ) / 同ボード複数に対処・特定、DeviceID : デバイス ID、Flag : 0 (固定)、*magic :マジック番号取得先ポインタ
戻り値	0:成功、-1:失敗

P C I ボードの特定はロケーション（バス・デバイス・ファンクション）で行います。  
 本ボードのベンダ ID、デバイス ID、検出対象ボード番号（1 枚目 = 0）を指定して実行すると同ボードのロケーションが magic に得られます。  
 同一ボードを複数インストールしているときは続いて検出対象ボード番号 = 1, 2, として実行すれば各ボードごとのロケーションが得られます。

ベンダ ID = 13FDH（マイクロサイエンス社 P C I ボード共通）、  
 デバイス ID = 10DH（ADM-670PCI）。

### 【2】指定ボード・指定レジスタのダブルワード読み込み（ベースアドレス値取得に使用できる。）

Int ReadPciDword (WORD magic, WORD reg, DWORD *data)	
引数	magic : 【1】GetPciDevice で得られたマジック番号、*data : データ取得先ポインタ、reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0:成功、-1:失敗

### 【3】指定ボード・指定レジスタのワード読み込み（通常は不使用。）

Int ReadPciWord (WORD magic, WORD reg, WORD *data)	
引数	magic : 【1】GetPciDevice で得られたマジック番号、*data : データ取得先ポインタ、reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0:成功、-1:失敗

P C I リソース情報の取得は【1】で検出したロケーション（magic）とレジスタ番号を指定して行います。物理ベースアドレス値はダブルワードなので【2】の関数を使用します。  
 各レジスタは本ボード上の P C I インターフェース素子 9050（PLX 社製）内にあり、次のように割り付けられています。

レジスタ番号 10H : P C I インターフェース素子 9050 で使用。  
                   14H : 未使用  
                   18H : I / O マップレジスタのベースアドレス BASE（3-4 項）。  
                   1CH : 未使用  
                   20H : 未使用  
                   24H : 未使用

なお、

I/Oマップでは得られた data の下位 2 bit をマスクした値がベースアドレス値です。

bit31	bit 2 1 0
物理ベースアドレス	
	× 1

【4】指定ボード・指定レジスタのバイト読み込み（割り込み番号値取得に使用できる。）

Int ReadPciByte (WORD magic, WORD reg, BYTE *data)	
引数	magic: 【1】GetPciDevice で得られたマジック番号、 *data: データ取得先ポインタ、 reg: PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0: 成功、 -1: 失敗

割り込みリソース情報取得も【1】で検出したロケーション (magic) とレジスタ番号を指定して行います。 data 値はバイトなので【4】の関数を使用します。

本レジスタも本ボード上の PCI インターフェース素子 9050 (PLX 社製) 内にあり、レジスタ番号 = 0EH です。

得られる data 値 (1 ~ 15) は割り込み番号です。 / 不使用時 = 0、または 255 / プログラム内ではベクタに変換して御使用ください。

なお、  
本ボードの標準設定では (プラグアンドプレイ実行時に) 割り込みリソースを要求しません。  
割り込みを使用するときは 1 - 5 項に記す手順で設定を変更してください。

## 汎用リソース情報取得関数ライブラリ / MS-DOS 版 について

## 関数一覧

## 【1】PCIバス上のボードの検出

Int_far GetPciDevice(WORD VendeID,WORD DeviceID,WORD nNum,WORD Flag,WORD _far *magic)	
引数	VendeID : ヴェンダ ID、 nNum : 検出対象ボード ( 0 ~ ) / 同ボード複数に対処・特定、 DeviceID : デバイス ID、 Flag : 0 ( 固定 )、 *magic : マジック番号取得先ポインタ
戻り値	0:成功、 -1:失敗

## 【2】指定ボード・指定レジスタのダブルワード読み込み ( I / O アドレス値取得に使用できる。 )

Int ReadPciDword (WORD magic , WORD reg , DWORD _far *data)	
引数	magic : 【1】GetPciDevice で得られた マジック番号、 *data : データ取得先ポインタ、 reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0:成功、 -1:失敗

## 【3】指定ボード・指定レジスタのワード読み込み ( 通常は不使用。 )

Int ReadPciWord (WORD magic , WORD reg , WORD _far *data)	
引数	magic : 【1】GetPciDevice で得られた マジック番号、 *data : データ取得先ポインタ、 reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0:成功、 -1:失敗

## 【4】指定ボード・指定レジスタのバイト読み込み ( 割り込みレベル値取得に使用できる。 )

Int ReadPciByte (WORD magic , WORD reg , BYTE _far *data)	
引数	magic : 【1】GetPciDevice で得られた マジック番号、 *data : データ取得先ポインタ、 reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0:成功、 -1:失敗

**使用方法** : 前記W I N D O W S 版と同様です。

<b>Quick-Basic (4.5) でワード I/O アクセス</b>	するためのライブラリ
--	------------

```

¥
|
|      GL_QB      MSCIENCE.QLB: クイックライブラリ
|                | - MSCIENCE.LIB: スタンドアロン・ライブラリ
|                | - MSCIENCE.BI : ライブラリ用インクルード・ファイル
|

```

本ボードの各制御レジスタ (I/Oポート) は全てワード構成です。

Quick-Basic そのままではではバイト I/O アクセスしかできませんが、当ライブラリを使用してワード I/O アクセスが可能になります。クイックライブラリ (QLB) はインタープリタで使用するもので、Quick-Basic の起動時に “/LMSCIENCE” オプションを指定します。

スタンドアロン・ライブラリ (MSCIENCE.LIB) は BASIC コンパイラ (BC.EXE) でソースをコンパイルしてリンクするときに Quick-Basic 付属の “BCOM45.LIB” と共にリンクして使用します。当 LIB は本ボードの動作確認・再調整プログラム 670QB2.EXE で利用されています。

### 関数一覧

#### 【1】指定 I/Oポートへのワード・データ書き込み

OWTW (port% , data%)	
引数	port% : I/Oポートアドレス data% : 出力 (ワード) データ
戻り値	なし。

#### 【2】指定 I/Oポートからのワード・データ読み込み

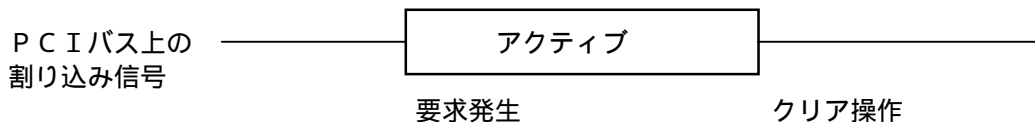
INPW% (port%)	
引数	port% : I/Oポートアドレス
戻り値	得られた (ワード) データ



## 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**（WINDOWS 9x / 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**（WINDOWS 9x / 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が、WINDOWS 2000用はWDMドライバの性格から割り込みがサポートされていません。（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。【1-4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。

## 4-5. Quick Basicサンプル

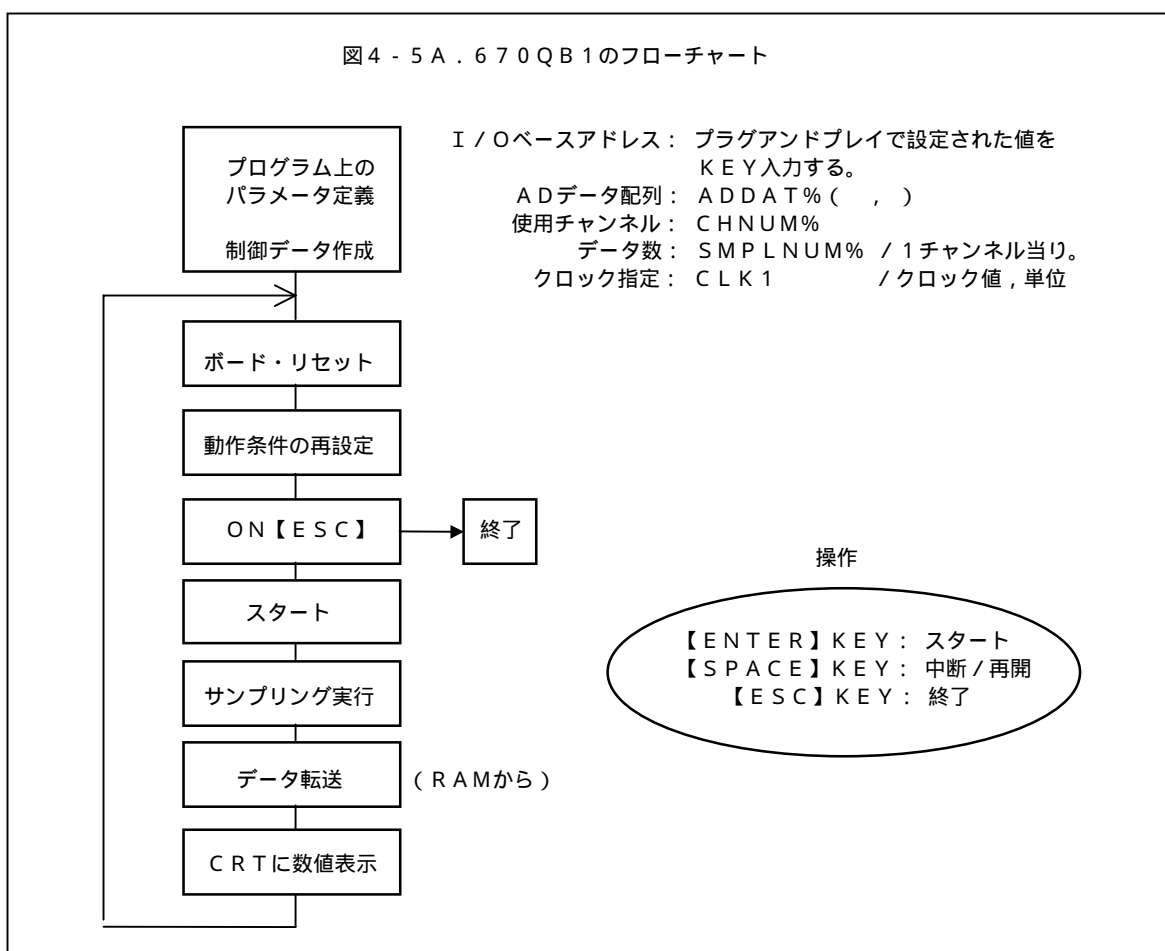
BASIC文だけで記述したサンプルソフト670QB1.BAS/670QB2.BASがあります。

670QB1.BAS

内部クロックを使用して指定点数だけ自動連続サンプリング、数値表示を繰り返し行います。  
自動連続サンプリング終了後に、本ボード上のRAMからデータを転送・表示する形ですから  
本機の最高速度が可能です。

2チャンネルまでは最高2MHzサンプリング、  
4チャンネルなら最高1MHzサンプリング

図4-5A. 670QB1のフローチャート

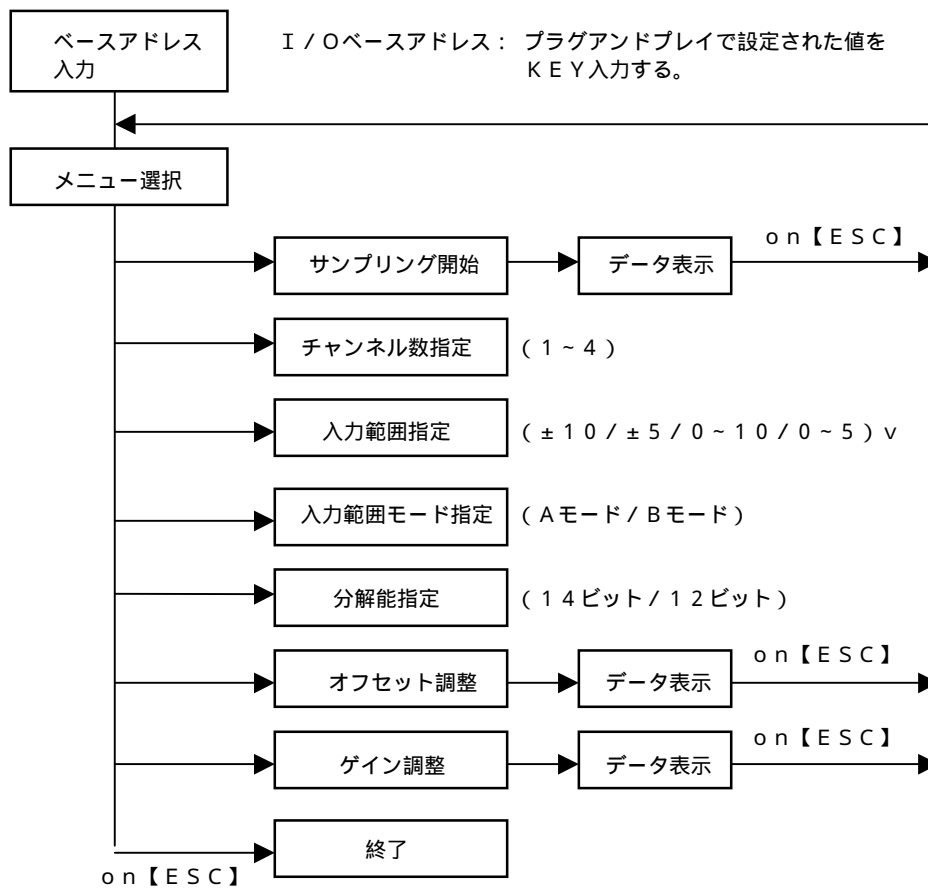


## 670QB2.BAS

本例では各チャンネルをマニュアルで10回ADサンプリングし、その平均値表示を繰り返して行います。これにより本ボード内部雑音(14ビット動作のとき約 $\pm 3$  digit)が平滑化され、直流入力なら $\pm 1$  digit内に収まります。

実行形式も用意されており、動作確認・再調整時のモニタに利用できます。(1-6項参照)

図4-5B. 670QB2のフローチャート



## 4-6. Cのサンプル

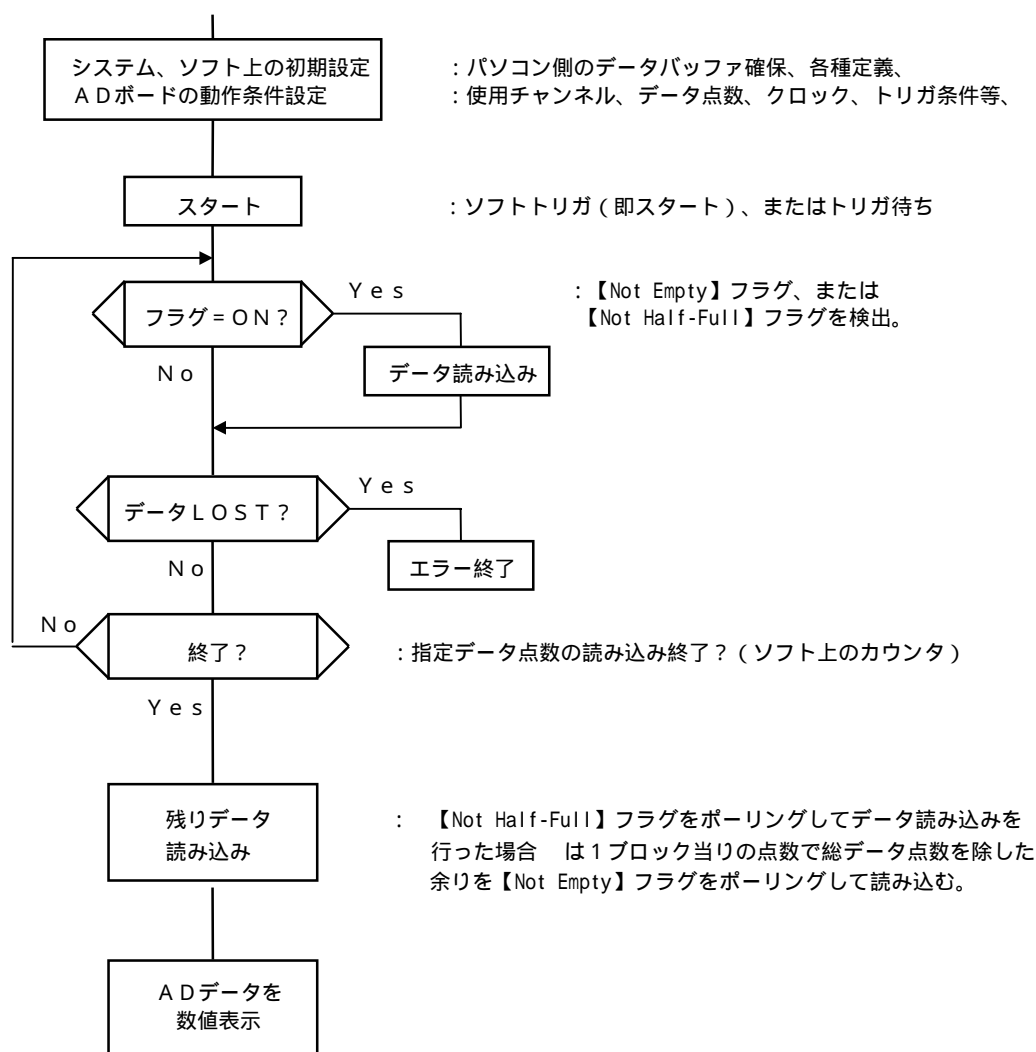
WINDOWSでは基本機能・動作が関数化されたハンドラ関数DLL/ドライバ(第6章)がありますが、MS-DOSの場合は以下に示す操作手順学習用サンプルを参考に御利用ください。

ポストトリガ動作例として以下の4プログラムがあります。

POL670E1.C  
INT670E1.C  
BLK670H1.C

は【Not Empty】フラグを監視して1語ずつ、  
は【Not Empty】状態変化による割り込みを使用して、同処理  
ルーチン内でデータを読み込む。  
は【Not Half-Full】フラグを監視してFIFOメモリ容量の  
半分(標準:2048語)単位でブロック転送命令(INSW)を  
使用してデータを読み込む。/最高速の方法です。/

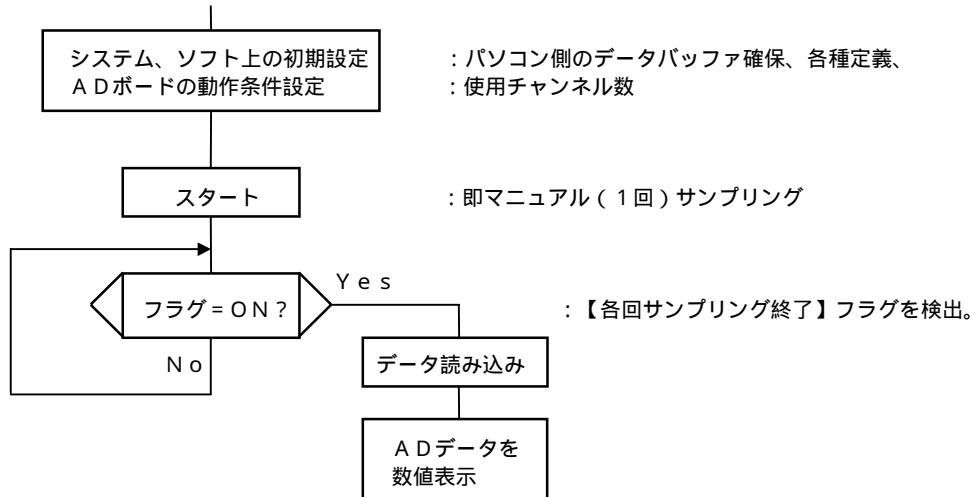
図4-6A. ポーリングによる操作手順 /



マスタスレーブ動作例 “MSV670” もあります。

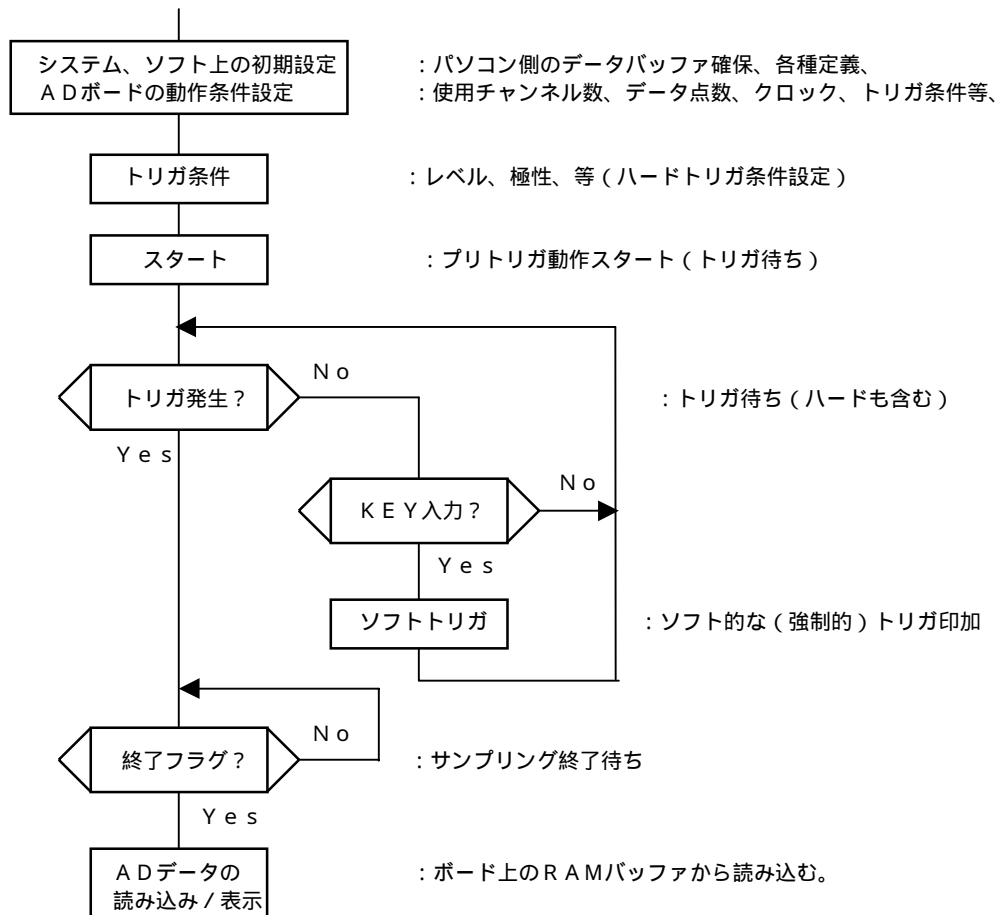
指定チャンネル数について各1回だけ即サンプリングする例 “MNL 670” です。

図4 - 6 B . マニュアルサンプリング例の操作手順 /



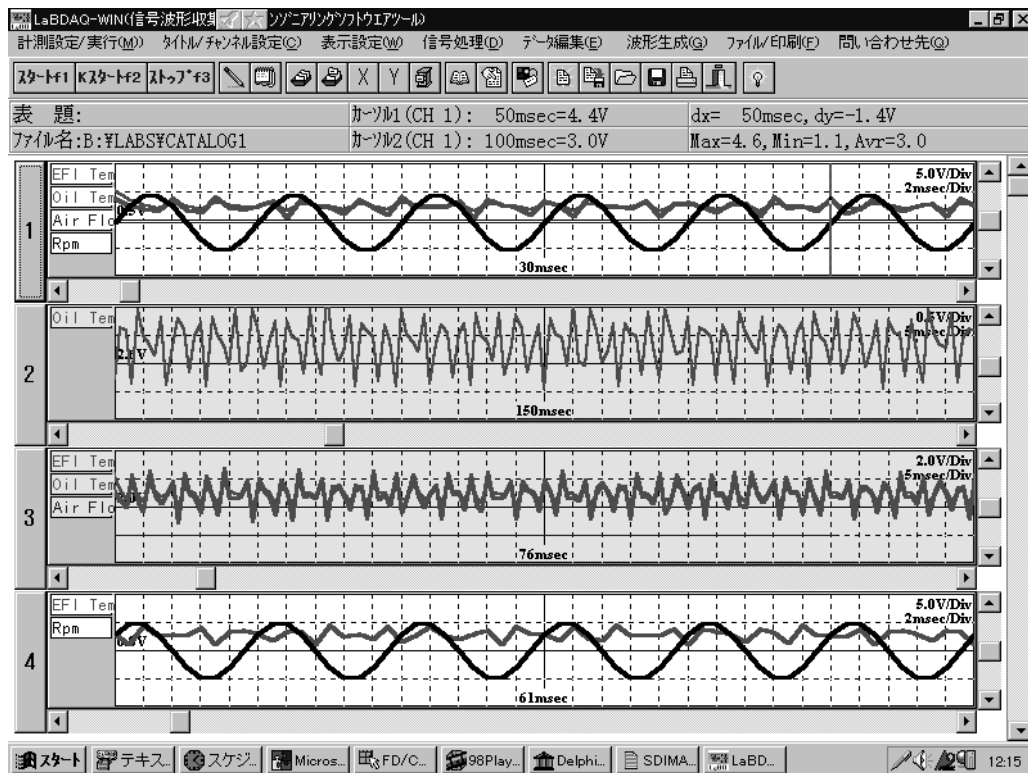
プリトリガ動作例 “PRE 670” です。

図4 - 6 C . プリトリガ動作例の操作手順 /



《信号処理なし版のLaBDAQ-AQは無償ダウンロード配布》  
 WINDOWS(2000・XP)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）

データ表示ウインドウ数：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 95・98・ME・NT・2000またはXPの動作する

IBM PC / AT互換機、98NXまたはPC9801・9821系機。

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)
- ☐ 計測中カウントのみ表示する (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)





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。FIFOメモリ付ADボードの場合、FIFOメモリ容量までは(CPU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FIFO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	Pentium(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	256KHz	(4×ch)μs		
ADM-681PCI	500KHz	(2×ch)μs		
ADM-682PCI	256KHz	(8×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。上表の値は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 LaBDAQ-AQ ¥無償配布(ADデータ収集専用、2000/XP版)  
LaBDAQ-PRO ¥78,000(ADデータ収集・解析、フル機能版)



## 第6章. WINDOWSハンドラ

当社製ADボード（IBM PC / AT互換機 = DOS / V機用）をVC、VB等で簡単に使用することのできるWINDOWS 95/98/ME/NT2000/XP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 / コパイラ : WINDOWS 9x、ME、NT(4.0)、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 : 各ADボード添付のサンプルディスク内。

対応ADボード : ADM-670PCI（最大8枚 / 32チャンネルまで制御可能）

サンプリング : RAMバッファ（動作終了後に読み出し）ではボードの最高速度が可能、FIFOバッファ利用の非同期読み出し（サンプリングと同時・平行可能）の速度は500KHz程度（WINDOWS 9x / Pentium 400MHz）。

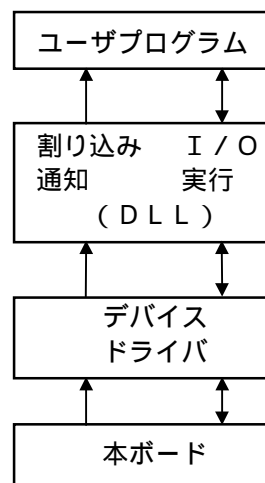
データ点数 : 拡張メモリ空容量（1語 = 2バイト）

割り込み : WINDOWS 2000・XPでは使用不可、9x・ME・NTでは**使用任意**。  
（使用する場合は要リソース取得 / 1-5項参照。）

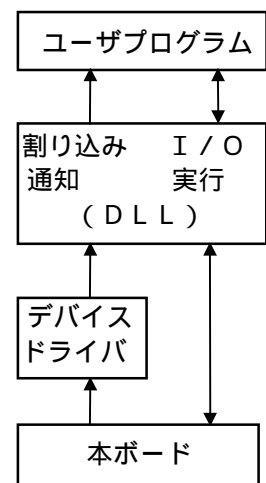
その他のボード : アドレス、割り込み番号が重複しない限り同時に使用可能。

図6-1. ソフトウェア構造

（WINDOWS2000・XP/NT4の場合）



（WINDOWS95・98・MEの場合）



## 6-2. サンプリン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/サンプリング点数/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びだけで、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形態のサンプリング動作が可能です。

- (1) マニュアル(1回)サンプリング・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、結果をバッファに得ます。
- (2) 連続サンプリン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。 ポストトリガとプリトリガ動作が選択・使用できます。 トリガはソフト(即)、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40MHz/32.768MHz、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。

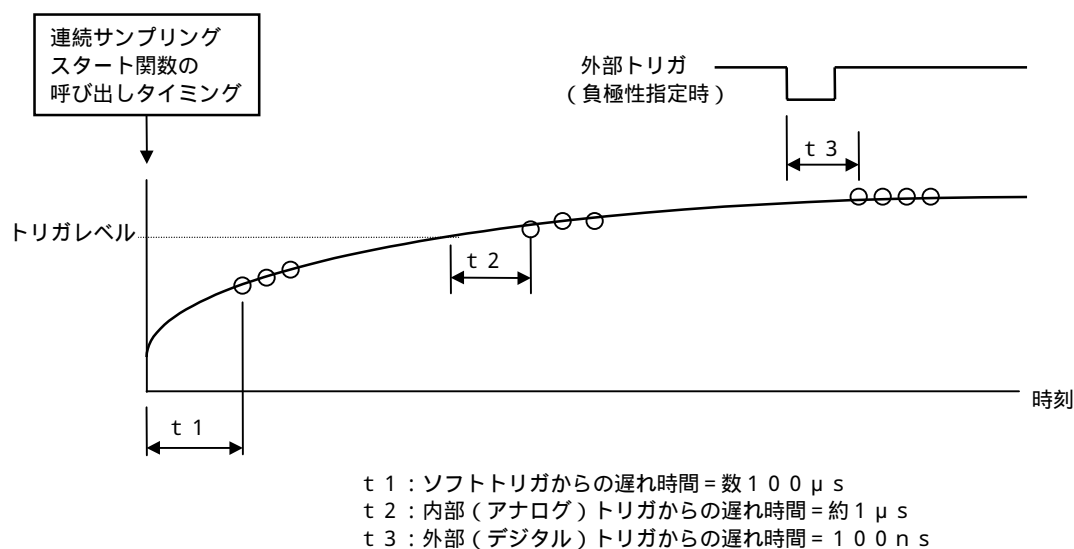
本ハンドラ関数DLLの管理するバッファに格納されるのはボード上のFIFOバッファから読み出されたADデータです。 ADデータはボード上のRAMバッファにも書き込まれています。 こちらはサンプリング終了後に専用の関数で読み出すことができます。

【注1】 指定チャンネル群: チャンネル0を先頭に指定最終チャンネルまでの一群。

【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6-2A. 連続サンプリン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t1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遅れは最大100 nsです。

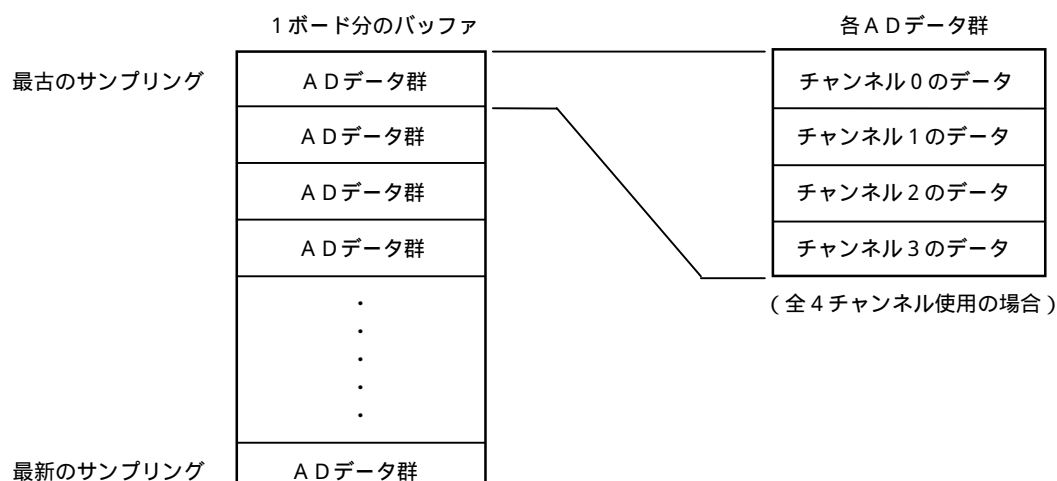
【注B】 内部(アナログ)トリガからの遅れ時間t2、および外部(デジタル)トリガからの遅れ時間t3は各スタート関数の呼び出しと同関数内の管理手続き実行してトリガ待ち状態となつてからの時間です。このトリガ待ち状態となるまでの時間は数100  $\mu$ sを要します。

### ADデータバッファと格納の様子 (本ハンドラDLLの管理するバッファ / from FIFO)

即トリガを含むポストトリガ動作で自動モードの場合、本ボード上のFIFOバッファから読み出されたADデータは(本DLLの管理する)バッファに転送されます。

この(パソコン側メモリ上の)バッファ内のデータ配置はADボード上のFIFOメモリと同一イメージで、複数の(各ボード分のADデータが各1個の)バッファに格納されます。

図6-2B. 1ボード分のADデータが1個のバッファに入る



### 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

**自動モード** : 当ハンドラDLL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データを自動転送する。ユーザプログラムは別のタスクをマルチスレッドで併行することができる。

DLL内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、数百KHzの速度を実現できる。後者は1回サンプリングごとに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。

《メッセージ》 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で行う。

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使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。

## 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】とします。

複数ボードを（マスタスレーブ接続で）使用する場合は3 - 20項の設定・接続とします。

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。

次に本機専用のハンドラ関数DLL、および汎用ドライバを所定のフォルダに移すインストール作業は以下のとおりです。

：ボード番号（=670）

WINDOWS 9x（ME）の場合

- (1) %MSCIENCE%\HND\_95\Adm      %DLL%\H      \_95.dll を %WINDOWS に、
- (2) %MSCIENCE%\HND\_95\Adm      %VXD%\D      \_95.vxd を %WINDOWS%\SYSTEM にコピーする。

WINDOWS NT（4.0）の場合 / Administrator レベルで行う /      : 0 ~ 7

- (1) %MSCIENCE%\HND\_NT\Adm      %DLL%\H      \_NT.dll を%WINDOWS%\SYSTEM32 に、
- (2) %MSCIENCE%\HND\_NT\Adm      %sys%\D      \_NT .sys を%WINDOWS%\SYSTEM32\DRIVERS に、
- (3) %MSCIENCE%\HND\_NT\Adm      %D      Reg.exe を任意のフォルダにコピーする。

《NTにおけるデバイスドライバの設定 / リソースの確認》

デバイスドライバの設定 / リソースの確認ユーティリティ（D      Reg）を起動すると、本ボードの（プラグアンドプレイで設定された）I / Oアドレス・割り込みレベル情報をレジストリに書き込み、（確認のため）表示し、さらに本専用ドライバに同情報を設定します。

このとき同時に、以後の本専用ドライバ起動方法（自動 / 手動 / 禁止）を選択設定します。

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) %MSCIENCE%\HND\_2K\Adm      %DLL%\H      \_2K .dll を %WINDOWS%\SYSTEM32 にコピーする。

L a b V I E W等の完成アプリケーションから利用する場合は

本ハンドラDLLを（L a b V I E Wなどの）市販・完成アプリケーションから呼出し実行するときはオープン&初期化関数【1】でウインドウハンドルHWN D O w n e rに N U L Lを渡します。この場合、アプリケーション側は本ハンドラからの各種メッセージを受け取ることができませんから、この条件を踏まえた利用に制限されます。

## 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次6-5項、およびサンプルソースを御覧ください。

- (1)初期化 : 【1】
- (2)サンプリング条件の設定 : 【2】～【6】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 : 【7】即スタート / トリガ待ち（内部 or 外部）
- (4)ステータス評価 : 【8】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】データ転送に割り込み使用時は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 : 【9】後処理

表 6-4 A . 制御関数一覧

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	ハンドル、使用ボード数
【2】AD_Set_SampCh	サンプリング実行チャンネル関連設定	チャンネル数、入力範囲
【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_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_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_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
【8】AD_Get_Status	ステータス取得	サンプリング進捗状況など
【10】AD_Read_DIIData	DLL内バッファからデータ読み出し	ユーザ指定バッファ
【11】AD_Read_DirectFifo	FIFOから直接にデータ読み出し	" " "
【12】AD_Read_DirectRAM	RAMから直接にデータ読み出し	" " "
【13】AD_Get_OneScan	マニュアル（1回）サンプリング	" " "
【9】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	DLL内バッファをリング状に設定	1 廻り後は上書き
【17】AD_Set_Inpmode	モード、分解能、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_Get_Pointer	ボード上RAMのポインタ取得	先頭位置、トリガ位置
【22】AD_Hit_SoftTrigger	（強制的な）ソフトトリガ操作	
【23】AD_Plus_Message	オプションメッセージ発信の設定	
【24】AD_Get_Libver	本ライブラリのバージョン情報を得る	

【注】 W I N D O W S 2 0 0 0 / X P 用に限り、関数名が異なります。  
 すなわち関数名中、文字“AD”の直後に対象ボード名（3桁番号）が含まれます。  
 例：AD670\_Open\_ADSys

以下に各関数の仕様・詳細を記します。 【注】WINDOWS 2000 / XP用に限り、  
各関数名中“AD”の直後に3桁のボード番号が  
入ります。 <例：AD670\_Open\_AD Sys>

## 【1】初期化

書式	int AD_Open_AD Sys ( HWND Owner, int num_board )
引数	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。 / num_board : 使用する本ボード数。(1~8)
戻り値	正常終了時: ボードのID (= 16H) エラー時: エラーコード / 負の値 (エラーコード表)
機能・動作	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。 【注1】: 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。 / 3-5項参照。

【注2】メガFIFOモジュール搭載のときは同モジュール上のDIPスイッチで<ハーフフル値>を  
設定しますが、本ハンドラのサポートする範囲は0(512語)~5(16K語)に限られます。  
これは市販のFIFO素子で定義された元々の語意(総容量の半分)を離れ、メモリの充満度を  
示すインジケータとなっています。

## 【2】サンプリング実行チャンネル関連の設定

書式	int AD_Set_SampCh ( int board_num, int no_ch, int scan_order[], int range[] )
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。 no_ch : サンプリングを実行するチャンネル数(1~4) / 必ず全ボード同一値。 scan_order[] : 各チャンネルのスキャン順。(本ボードでは固定) range[] : 各チャンネルの入力レンジ番号。 / 0: ±10V、 1: ±5V 2: 0~+10V、 3: 0~+5V
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)
機能・動作	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。(バイポーラ・ユニポーラの混在不可) 初期の分解能は12ビット/Aモードとなっており、補助関数【17】により14ビットでの利用も可能 だが、14ビットのときはバイポーラ(±10V、または±5V)入力範囲のみ。 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。

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】FIFOからのサンプリングADデータ転送モード、および割り込みイベントの設定

書式	int AD_Set_SampMode ( int trs_trig, int buf_area, int intr_sw )
引数	trs_trig : ADデータ転送の参照フラグ / 0: EMPTY解消、 1: HALF-FULL buf_area : ADデータ転送先 / 0: DLL内バッファ、 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) intr_sw : 割り込みイベント発生要因指定 / 0: 割り込み不使用、 4: トリガ発生 1: 連続サンプリング・クロック、 5: 1回サンプリングスキャン終了 2: 外部割り込み入力信号( ), 6: Not-Empty 3: 外部割り込み入力信号( ), 7: Half-Full
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)
機能・動作	buf_areaでDLL内バッファとした場合は当ハンドラ自体がDLLバッファ内にデータを 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AD_Read_DLLDataで読み出す。 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 FIFO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。(プリトリガ動作のときはマニュアルモードのみ。) 【注1】: いずれのモードでも参照するフラグをtrs_trigで指定する。 【注2】: いずれのモードでもADデータ転送自体には割り込みは使用されない。 また、intr_swは独立したイベント発生要因指定。



#### 【４】トリガ関連の設定A（レンジトリガ以外の場合）

書式	int AD_Set_Trigger(int trg_mode, int trg_source, int trg_pol, int trg_level)								
引数	<tr> <td>trg_mode</td> <td>: トリガ動作モード / 0 : 即トリガ、1 : ポストトリガ、2 : プリトリガ</td> </tr> <tr> <td>trg_source</td> <td>: トリガ源 / 0 : ソフト、1 : 内部 (アナログ)、2 : 外部デジタル</td> </tr> <tr> <td>trg_pol</td> <td>: 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</td> </tr> <tr> <td>trg_level</td> <td>: 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。</td> </tr>	trg_mode	: トリガ動作モード / 0 : 即トリガ、1 : ポストトリガ、2 : プリトリガ	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。
trg_mode	: トリガ動作モード / 0 : 即トリガ、1 : ポストトリガ、2 : プリトリガ								
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	<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)</p>								
機能・動作	<p>トリガ動作モード、源、極性、レベル等の設定。</p> <p>なお、トリガ動作モードをポストアトリガ、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 )</p> <p>アナログトリガ検出対象はチャンネル 0。 ( 複数ボード使用のときはマスタボード )</p>								

【4】'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 AD_Set_RangeTrigger(int trg_mode, int trg_sel, int trg_level_hi, int trg_level_lo)
引数	trg_mode : トリガ動作モード / 0 : 即トリガ、1 : ポストトリガ、2 : プリトリガ 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)
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】を使用) アナログトリガ検出対象はチャンネル0。(複数ボード使用のときはマスタボード)

## 【5】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 AD_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)
機能・動作	オプション、または外部クロック源を使用するとき、その周波数値を設定する。 後記【6】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には必要ない。

## 【6】サンプリング・クロック値の設定

書式	int AD_Set_Clock(int clk_source, int set_mode, int *time_unit, int *clk_period)
引数	<p>clock__source : クロック源 / 0 : 内部クロック源0 (40.00MHz)  1 : 内部クロック源1 (32.768MHz)  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 = )</p> <p>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 : <math>\mu</math>s、 3 : ns  clk__period : クロック周期の値、または分周比</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)</p>
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【5】ad__set__exclckでクロック源周波数値を定義しておく。

## 【7】サンプリング・スタート

書式	int AD_Start_Samp ( int post_samp, int dcn )
引数	<p>post__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)  dcn : FIFOバッファへの間引き書き込み指定データ、  (dcn + 1) 回ごとに書き込む、 = 0 のときは間引きなし。【3-11項】</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)</p>
機能・動作	<p>サンプリングを実行する。  前記【4】トリガ設定でトリガ動作モードを《即トリガ》または《プリトリガ》としてあるときは即サンプリングスタート、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。  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、DLL内に確保されるデータバッファサイズ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。(1データ = 2バイト)</p>

## 【8】ステータス取得

書式	int AD_Get_Status (int *sampled, int status[])
引数	<p>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 / (3-16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)</p>
戻り値	<p>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)</p>
機能・動作	<p>《即トリガ》または《ポストトリガ》動作で自動モード (【3】参照) のときは当該時点でのトリガ後サンプリング済みデータ点数 (DLL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。</p> <p>《プリトリガ》動作のときは自動モード / マニュアルモード、いずれでもボード上のトリガ後サンプリング数カウンタ値、およびマスタボードのステータス生データを得る。 / 3-16項参照。 /</p>

## 【9】本ハンドラの終了

書式	int AD_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)
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。

## 【10】DLL内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

書式	int AD_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内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内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)
機能・動作	サンプリングされたADデータがDLL内の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。 【注】当方法はサンプリング動作モード設定【3】で自動モードを指定したときだけ有効。

## 【11】FIFOバッファ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

書式	int AD_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)
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。

## 【12】ボード上のRAMRAM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

書式	int AD_Read_DirectRAM (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 : RAM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 : RAMバッファから読み出すデータの先頭位置。 / 関数【21】で得た値が指標 / 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)
機能・動作	サンプリングされたADデータがボード上のRAM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。

## 【13】マニュアル (1回) サンプリング・スキャン

書式	int AD_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)
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。 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, 3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

書式	int AD_Out_Aux (int board_no, int out_data)
引数	board_no : 対象ボード番号 out_data : 汎用デジタル (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)
機能・動作	各ボードごとの汎用4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

書式	int AD_Inp_Aux (int board_no)
引数	board_no : 対象ボード番号
戻り値	正常終了時 : = 汎用デジタル4ビット (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)
機能・動作	各ボードごとの汎用デジタル (現在値) 入力データ読み込み。

## 【16】DLL内データバッファをリング状 / エンドレスに設定

書式	int AD_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)
機能・動作	DLL内のADデータバッファをリング状・エンドレスで使用するかどうかの設定。(トグル) リング状・エンドレスに設定した場合、自動モードでDLL内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。 《即トリガ》または《ポストトリガ》動作の場合は本設定により当ハンドラは無限サンプリング・ モードとなる。【7】サンプリングスタート関数で指定する<トリガ後のサンプリング点数>と 【2】で指定する<サンプリング実行チャンネル数>の積がDLL内ADデータバッファのサイズと なるが、<トリガ後のサンプリング点数>は指定値ではなく、【18】強制停止まで無限に実行する。 長時間の監視システム等に利用できる。 《プリトリガ》動作のときは元々、DLL内ADデータバッファを用意しないので無意味。

## 【17】アナログ入力モードの設定

書式	int AD_Set_Inpmode (int inp_mode, int ad_code, int ad_reso, int ad_rangemode)
引数	inp_mode : アナログ入力信号形式 (本ハンドラではダミー) ad_code : ADデータコード指定 / 0: バイナリ、1: 2の補数 ad_reso : AD変換分解能 / 0: 12、1: 14 ad_rangemode : アナログ入力範囲モード / 0: Aモード、1: Bモード (3-6項)
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)
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 AD_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)
機能・動作	サンプリングを強制的に中止する。 (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。) 本ハンドラではADデータ転送には割り込みを使用していないが、ユーザプログラム中で割り込みイベントを利用している場合は連続サンプリング終了後、必ず当関数を実行すること。

## 【19】残りADデータの読み込み

書式	int AD_Read_RestData (void)
引数	なし
戻り値	正常終了時 : = 読み込んだ (残り) データ数 エラー時 : エラーコード / 負の値 (エラーコード表)
機能・動作	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。 なお、前【18】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。

## 【20】(ADボード) 各フラグのクリア

書式	int AD_Clear_Flags (int c_data)
引数	c_data : クリアビット指定データ。 / 3-16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)
機能・動作	ADボード上の各フラグをクリアする。

## 【21】ボード上RAMバッファのポインタを得る。

書式	int Ad_Get_Pointer (int *top_pos, int *trig_pos)
引数	top_pos : (RAM内の) 先頭データ位置 / 最も古い有効データ位置 / trig_pos : (RAM内の) トリガデータ位置 / ポストトリガ動作時は先頭データ位置と同値 /
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)
機能・動作	【12】RAM内データ読み込み関数を実行するとき、当関数で得たポインタを使用する。

## 【22】（強制的な）ソフトトリガ操作

<code>int AD_Hit_SoftTrigger(void)</code>	
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 参照）
機能・動作	ソフト上からトリガを発生させる。（トリガ待ち状態のとき、強制的に）

## 【23】（オプション的な）メッセージ発信の設定

<code>int AD_Plus_Message(int submit)</code>	
引 数	<code>submit</code> ：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 参照）
機能・動作	（マニュアルモードの場合） 関数【3】の <code>trs__trig</code> 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 <code>trs__trig</code> 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

<code>int AD_Get_Libver(int ver)</code>	
<code>ver</code>	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 6 - 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 <code>ver = 0</code> として実行すると 戻り値は <code>0 x 1 0 1</code> となります。

表6-4. 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?)	AD_Open_AD Sys ( )
- 2	ボードIDが違う。	AD_Open_AD Sys ( )
- 3	FIFOバッファにアクセスできない。	AD_Open_AD Sys ( )
- 4	ドライバファイルを検出できない。	AD_Open_AD Sys ( )
- 5	ドライバファイルのバージョンが不適切。	AD_Open_AD Sys ( )
- 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 ( )
- 8	ボード番号(スイッチSW-BN)の設定値が不適当。	AD_Open_AD Sys ( )
- 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
- 13	サンプリング・スキャン順 指定パラメータの不適当	scan_order[ ]
- 14	アナログ入力レンジ 指定パラメータの不適当	range[ ]
- 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	
- 30	FIFOバッファに書き込む間引き 指定パラメータの不適当	dcn
- 31	アナログ入力信号形式 指定パラメータの不適当	inp_mode
- 32	ADデータ・コード 指定パラメータの不適当	data_code
- 33	AD変換分解能 指定パラメータの不適当	ad_reso
- 34	アナログ入力範囲モード 指定パラメータの不適当	ad_rangemode
- 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、チャンネル数設定の未実行	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 ( )
- 48	ブリトリガ動作で(トリガ後)無限サンプリングを指定した。	AD_Start_Samp ( )
- 49	メモリ確保エラー	
- 50	ユーザプログラム内のAD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AD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 ( )
- 54	現在、トリガ待ち状態ではない。	AD_Hit_SoftTrigger ( )
- 60	ブリトリガ動作で無限サンプリングを指定した。	AD_Start_Samp ( )
- 61	ブリトリガ動作でDLLバッファ使用を指定した。	AD_Start_Samp ( )

## 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【10】【11】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が、いずれの場合もサンプリング動作終了後、ボード上のRAMに有効データが残ります。なお、＜例1＞から＜例3＞は即トリガを含むポストトリガ動作の場合です。

## &lt; 例 1 &gt;

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 <b>有限サンプリング</b> 。	Sample1.vbp	Visual Basic (5,6)
	Sample1.c	Visual C (5,6)
	Sample1.c	Borland C (5)
	Sample.dsp	Visual C++ (5,6)
	Sample1.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>		

## &lt; 例 2 &gt;

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 <b>無限サンプリング</b> 。	Sample2.vbp	Visual Basic (5,6)
	Sample2.c	Visual C (5,6)
	Sample2.c	Borland C (5)
<p>当ハンドラDLLの管理するバッファを“<b>エンドレス・リング状</b>”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【16】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や、低速のプリトリガ動作などに利用できる。</p>		



## &lt; 例 3 &gt;

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。	S a m p l e 3 . v b p	Visual Basic (5,6 )
	S a m p l e 3 . c	Visual C (5,6 )
	S a m p l e 3 . 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、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、【11】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、は1ボード当り16M語以内であるが、【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>		

## &lt; 例 4 &gt;

利用形態	プログラムファイル名	対応言語 (version)
ブリトリガ動作	S a m p l e 4 . v b p	Visual Basic (5,6 )
	S a m p l e 4 . c	Visual C (5,6 )
	S a m p l e 4 . c	Borland C (5 )
	S a m p l e . d s p	Visual C++ (5,6 )
	S a m p l e 2 . d p r	Delphi (3,4,5)
<p>ブリトリガ動作のときは“DLL内バッファ”が無いので普通はサンプリング終了を待ってボード上のRAMからADデータを読み出します。サンプリング中もデータをモニタしたいときは&lt;例3&gt;と同様に“マニュアルモード”として、FIFOからリアルタイムに読み出すことができます。(高速サンプリングのときは間引きも可能)</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”	
“AD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内のバッファ”	ユーザプログラム側へは【10】AD_Read_DllData( )で転送、容量は【2】指定チャンネル数×【7】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”	【8】ステータスから得る
“エンドレス・リング状”	【16】DLLバッファをリング状・エンドレスに設定関数
“ストップ操作”	【18】連続サンプリング動作の強制停止関数

## 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充分な確認が必要です。

- (2) 本ハンドラは【AD\_Open\_AD Sys ( )】で使用開始、  
【AD\_Close\_AD Sys ( )】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) 本ハンドラの管理する割り込みは1系統のみです。  
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と、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。すなわち、ボード上のRAMに残る最終データのほかに、FIFOメモリ容量までは本ボード自体の最高速サンプリング実行と平行してのデータ読み出しが可能です。

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Pentium 400MHz / WINDOWS 9xシステムで約800KHz程度のリアルタイム読み込み（バス転送速度）が可能です。

- (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？(通常は不要/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。<br>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、

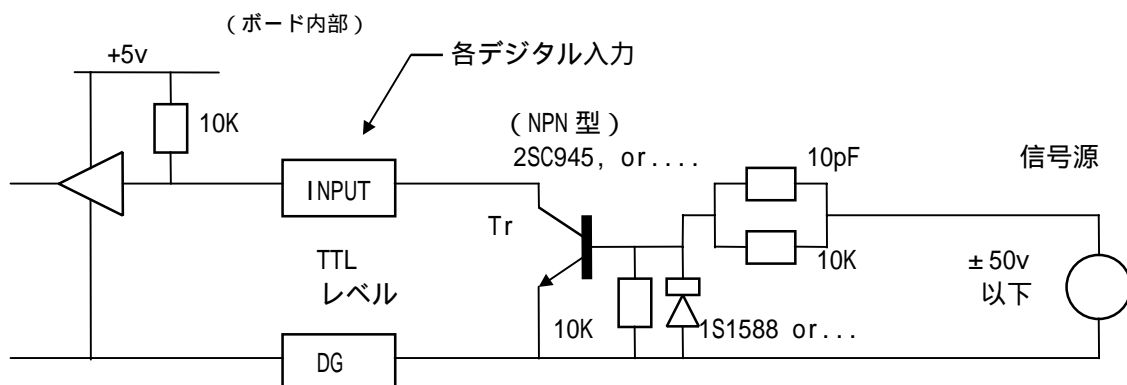
ファンクション・ジェネレータ等の交流信号出力を接続して破損させる例が多いようです。矩形波でも±に振れる信号は接続できません。特に、負側の許容レベル【-0.6V】が低いことに注意してください。

+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 : 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。 2002 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。

### 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 ==

本ボード上の諸設定は出荷時の状態（1-2項）とします。

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）

図1-6のように本ボードの全アナログ入力を基準電圧源に接続します。

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 ==

テストシステムの電源を投入し、MS-DOSを立上げます。

試運転・動作確認用プログラム“670QB2”を読み込み、実行します。  
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値を入力します。  
スタートすると、  
全チャンネルのアナログ入力をサンプリングしては数値表示する動作を繰り返します。

【0】Sampling Start ではサンプリングした生データを表示しますが、  
【5】【6】の調整ではバラツキ（内部雑音）除去のため、10回サンプリングした値の平均を表示しています。/これにより、ほぼ±1digit 程度のバラツキに収まります。/

操作は、

0 . Sampling Start	【Space】キー： 中断 / 再開
1 . Select Channels 【4】	【ESC】キー： 終了
2 . Select Range 【+ - 10V】	
3 . Select Resolution 【14BIT】	
4 . A.mode / B.mode 【A.mode】	→ 入力範囲モード：2-2項参照。
5 . Offset adjust	→ オフセット調整
6 . Gain adjust	→ ゲイン調整

なお、本プログラムではADデータを10進数で表示します。

## 調整手順

メニュー【5】【6】の順に作業を行います。

【5】Input Offset adjust : 信号源を0Vに設定（または各入力をアナロググランドに接続）、表示値（TARGET）が0V相当となるようにTM00～TM30を調整します。

調整対象チャンネル名	CH0	CH1	CH2	CH3
オフセット調整トリマ	TM00	TM10	TM20	TM30

【6】Gain adjust : 信号源をフルスケール値付近に設定し、表示値が適正值±1となるようにTM01～TM31を調整します。

調整対象チャンネル名	CH0	CH1	CH2	CH3
ゲイン調整トリマ	TM01	TM11	TM21	TM31

必要により【5】【6】を繰り返します。 【ESC】でプログラムを終了します。

## 得られる正確度

通常、パソコン用のADボードでは製造・調整環境と実機を含む現場環境が一致しませんから、現場での絶対正確度を保証することはできませんが、ADボード自体の性能を規定する相対正確度（＝校正可能限度）と製造・調整環境で使用する測定器で決まる絶対正確度（製造時・常温）は下表のとおりです。

表7-3A. 正確度

アナログ入力範囲	非直線性%FS	相対正確度 %FS	絶対正確度%FS
最終調整範囲 (14bit / ±10V / AE-ド)	0.013	0.033	0.063
その他の入力範囲		0.053	0.083

## 定義

- (a) 非直線性 : 使用されるAD変換素子に固有の性能。
- (b) 相対正確度 : 非直線性を含む、回路全体の性能。（＝校正可能限度）
- (c) 絶対正確度 : 相対正確度に校正測定器の正確度を加算した値。（製造時・常温）

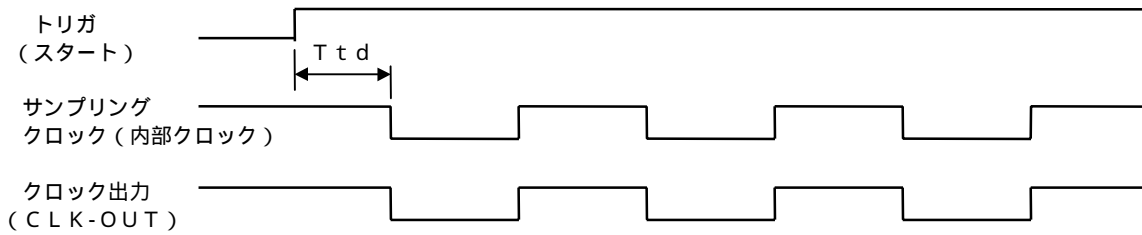
【注1】 当製品は正確度0.03%の測定器を使用し、常温で最終調整を行っています。  
当表に表示した相対正確度と絶対正確度の差はこれによるものです。  
なお、周囲温度の変化が大きいときは温度ドリフト（typ. 25ppm/℃）による誤差も加算されます。 また、経年変化のデータや保証はありません。

【注2】 当表の値にはCPUを含むシステム全体から発生する雑音が含まれていません。  
この雑音は安定した直流電圧を連続的にAD変換した値のバラツキとして認識され、  
12ビットのとき1digit、14ビットのときは4digit程度が普通です。  
瞬時値を1回だけAD変換した値には当雑音を考慮する必要があります。

## 7-4. 制御信号・タイミング等

## クロック入出力

図7-4A. 内部クロック源を使用する場合

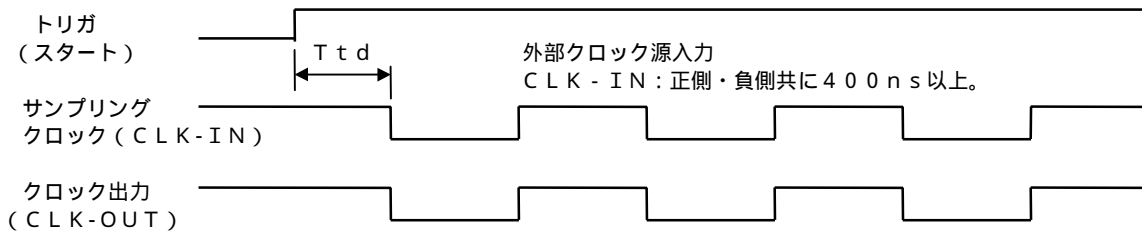


: クロック出力の有効エッジは常に立下り。

$T_{td}$ : 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  $100\text{ ns}$ )

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  $50\text{ ns}$  以内に開始。

図7-4B. 外部クロック源を非分周 (1/1) で使用する場合

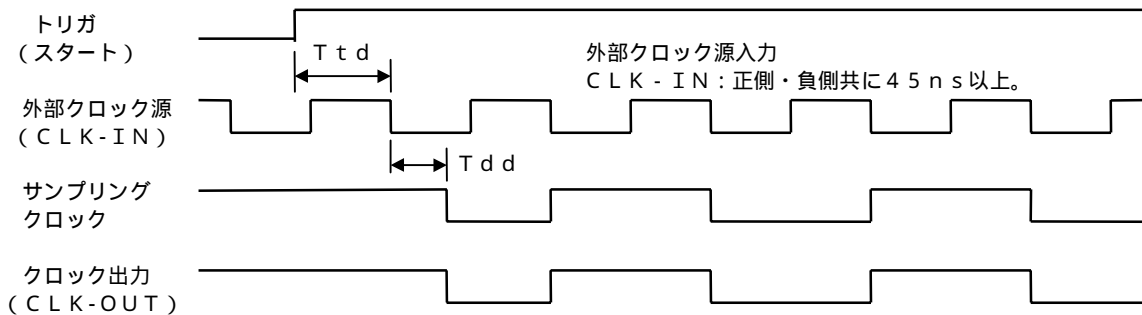


: クロック出力の有効エッジは常に立下り。

$T_{td}$ : 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  $100\text{ ns}$  以内に開始。

図7-4C. 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)

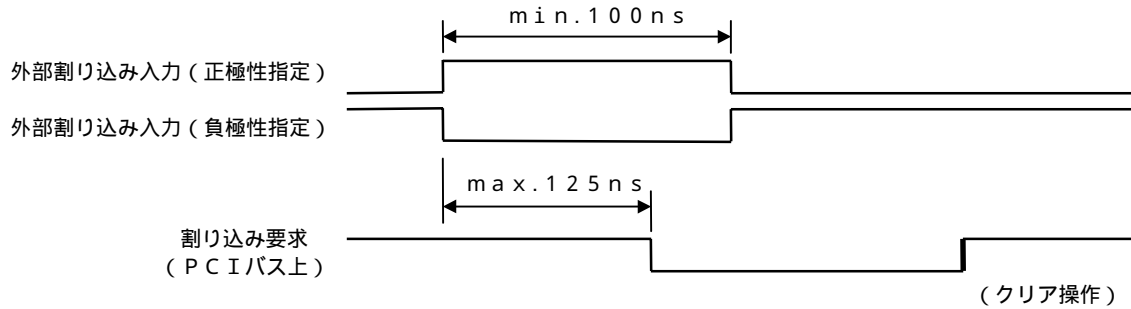
実際のサンプリング実行はクロックの各有効エッジから  $100\text{ ns}$  以内に開始。

$T_{dd}$ : 分周開始の遅れ時間 (最大  $100\text{ ns}$ )



### 外部割り込み入力 (許可されている場合 / 3-14項)

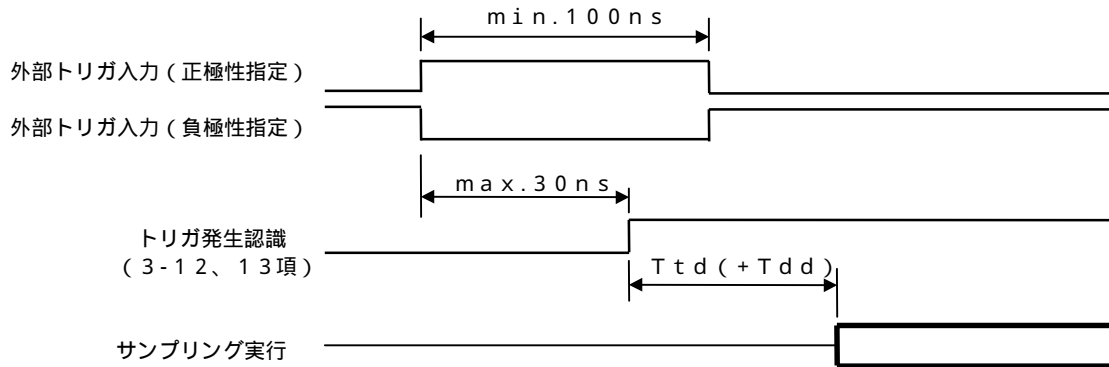
図7-4D. 外部割り込み入力 ~ 割り込み受け付け



: 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 / 3-13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。  
 $T_{td}$ : 図7-4A, B, C参照。  
 $T_{dd}$ : 図7-4C参照。

## 7-5. 付録 (WINDOWS 2000・XP対応について)

### WINDOWS 2000

ボードのインストール： WINDOWS 2000はNT4.0の上位バージョンですが、プラグアンドプレイ機能を持つため、本ボード装着直後のインストール作業時にWINDOWS 2000 対応のインストールディスク（当社製 / ver 2.00 以上）が必要です。作業手順は本書 1 - 5 項、または本ボードに同梱の作業説明書に従ってください。

ソフトウェアサポート： 汎用のI/Oドライバ、および本ボード専用の関数DLLが追加されています。 前者については4 - 2 項、後者については第6章をごらんください。

### WINDOWS-XP

ボードのインストールからドライバ、ハンドラ関数DLLまで、添付のWINDOWS 2000用ソフトウェアがそのまま御利用いただけます。



FAX : 03 ( 3301 ) 5593

## Q & A フォーム

発信：      年      月      日 /      時      分

製品名	ADM-670PCI		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)8377

御使用者		( 所属部・課 )
団体名		
T E L		( 所在地 )
F A X		