

## 第4章．ソフトウェア

《注1》 W I N D O W S 9 8 / M E 対応： 特に断りのない限りW I N D O W S 9 5 用ソフトがそのまま使用できます。

《注2》 W I N D O W S X P 対応： 特に断りのない限りW I N D O W S 2 0 0 0 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 ( 1.44 MB ) F D またはC D に圧縮された形で格納されており、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。

#### 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 ¥ M F U ¥ M F U 5 8 3 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M F U - 5 8 3 P C I 関連プログラムが図4 - 1 に示すロケーションに展開・インストールされます。

( 2 0 0 5 年4月より以前のC D R O M を使用する場合：D O S 窓利用 )

#### 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 ¥ M F U ¥ M F U 5 8 3 【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きます。

= 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後のディレクトリ構造は図4 - 1 のようになります。

図4-1. インストール後のディレクトリ

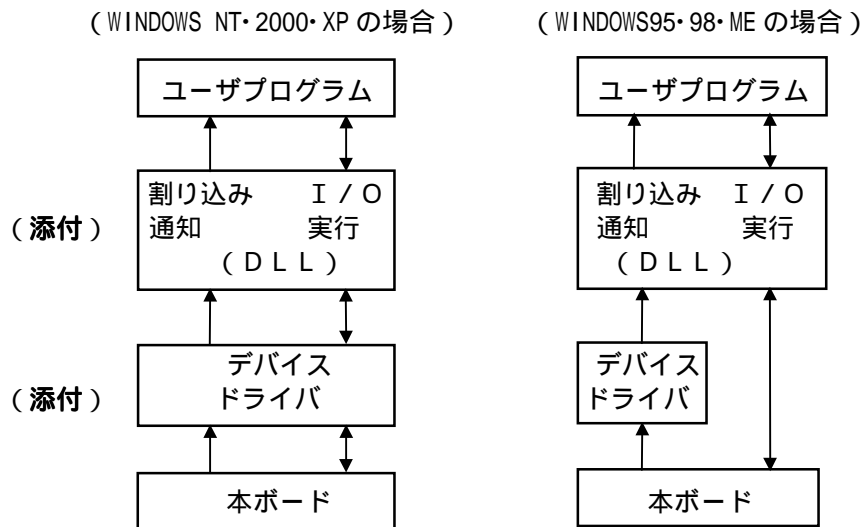
本図は原形です。 充実・改良の目的で後日、追加・変更も有り得ます。

¥			
MSCIENCE			
	UTILITY (本ボードの設定)	WIN95	CF9050DP.COM: コンフィギュレーション(95用)
		- WINNT	CF9050NT.EXE: コンフィギュレーション(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 -	- 583QB1.EXE: 本ボードの試運転・動作確認用プログラム		
	- 583QB1.COM: 英語モードに切り替えた後、EXEを実行する		
- SMP583C -	- MICROSOFT.H: MS-C用ヘッダ		
(各種Cサンプル)	- BORLAND.H: TURBO-C, BORLAND-C用ヘッダ		
	- MFU58X.H: 共通ヘッダ		
	- INT583.C: 割り込み動作例 (外部イベントに同期)		
	- POL583.C: ポーリング動作例 (外部イベントに同期)		
	- MNL583.C: マニュアル動作例		
- SMP583B -	- 583QB1.BAS: Quick-Basic(4.5)用サンプル		
(QBサンプル)	- 583QB2.BAS: 同上(外部イベントに同期)		
	SMP583VB	SMP583.VBP: プロジェクト	
(VBサンプル)	- SMP583.FRM: フォーム(メイン)		
	- DRVDLL.BAS: DLL定義		
	- MFU583.BAS: ハードウェア定義		
	Hnd_95	Mfu58x	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	Mfu58x	D58xreg: 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
			各サンプルは上記WINDOWS 9x・MEハンドラ同様
(WINDOWS 2000/XP用ハンドラ)			
	Hnd_2K	Mfu58x	各サンプルは上記WINDOWS 9x・MEハンドラ同様
	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
		- MS_PCI.DLL	: リソース取得ライブラリDLL
			(Win9x・ME・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

## 4-2. WINDOWSドライバについて

WINDOWS 9x / ME / NT / 2000 / XP 向に汎用 I / O 読み書き用 DLL が添付されています。

基本的には当 DLL (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**WINDOWS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。



**WINDOWS 3.1 :** Win31フォルダ以下に格納されており、VB (2.0) で利用できます。C, C++ の場合は当 DLL 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、DOS 同様に直接制御で対応できます。

**WINDOWS 9x :** Win9xフォルダ以下に格納されており、VB (4.0/5.0) で利用 (or ME) 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 DLL 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 DOS 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。

**WINDOWS NT :** WinNTフォルダ以下に格納されており、VB (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。

NT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16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 /

**WINDOWS 2000 :** Win2Kフォルダ以下に格納されており、VB (6.0) で利用 (or XP) できます。なおドライバ (WDM) 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 DLL 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】WDMドライバの性格から割り込みは使用できません。  
詳細は¥MSCIENCE¥WIN2K¥DOC 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 (WINDOWS2000/XP 以降のWDMでは不要)

汎用リソース情報取得関数 `MS_PCI.DLL / WINDOWS9x・ME・NT 用` について

ユーザプログラムから本PCI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章に記す本機の専用ハンドラ（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&ドライバが必要です。使用手順は、

#### 【1】PCI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

PCI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PCIボード共通)、  
 デバイスID = 102H (MFU-581/583PCI)。

#### 【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

PCI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PCI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 : PCI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BASE1 (3-2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 - 4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 VendelD,WORD DeviceID,WORD nNum,WORD Flag,WORD _far *magic)	
引数	VendelD : 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

## 【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。 )

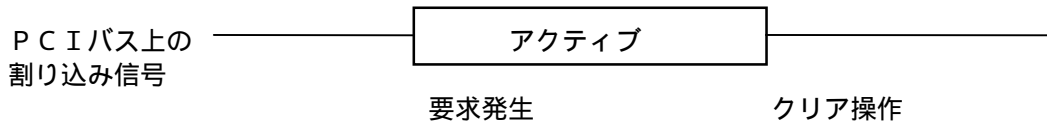
Int ReadPciByte (WORD magic , WORD reg , BYTE _far *data)	
引数	magic : 【1】GetPciDevice で得られた マジック番号、 *data : データ取得先ポインタ、 reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。
戻り値	0:成功、 -1 : 失敗

**使用方法 :** 前記W I N D O W S 版と同様です。

#### 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（WINDOWS 95 / NT用）デバイスドライバでは汎用のため、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が、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。

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。もちろん、不要なリソースを開放して他に流用することは可能です。

### 割り込み制御の手順

割り込み制御ポート【3 - 4 項】で外部入力割り込み許可ビット B 7 をセット、また、エッジ（ビット B 6）を指定します。これだけの場合、外部入力の有効な信号エッジが発生しても《PCIバス上の割り込み要求信号》はアクティブになりません。

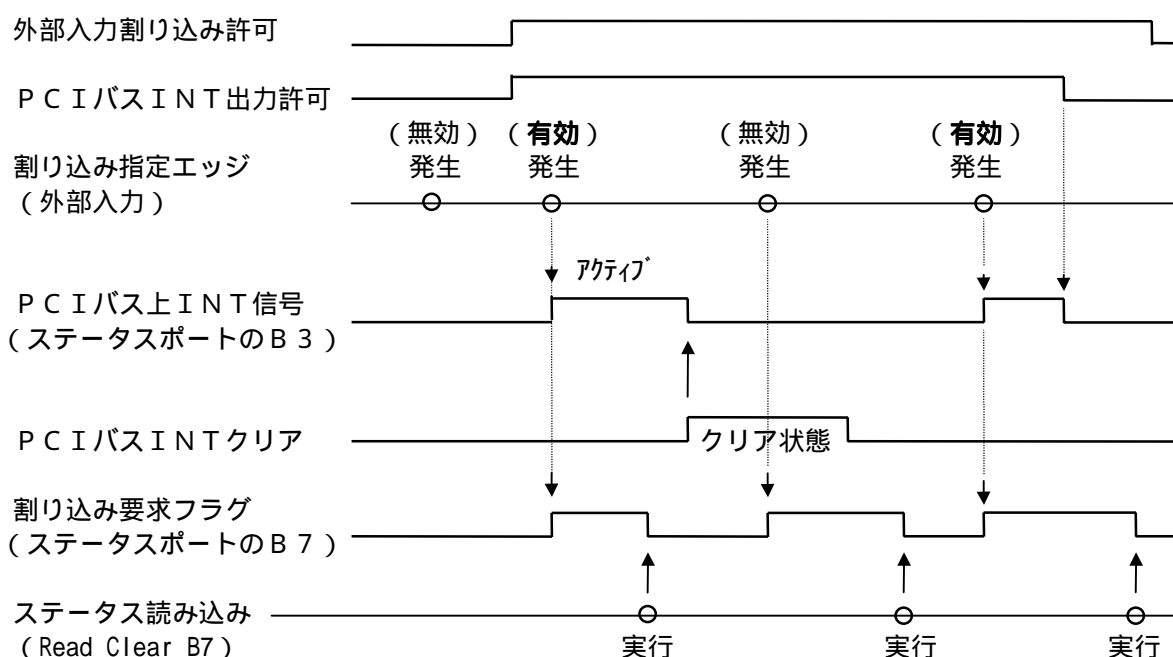
ステータスポート【3 - 8 項】の割り込み要求フラグ（B 7）がセットされるだけです。さらに PCIバスINT出力許可ビット B 0 をセットすることによって実際の割り込み要求を発信（アクティブ）可能になります。

外部入力の指定エッジが発生し、割り込み要求が発生しました。

割り込み要求が受け付けられた場合、通常はデバイスドライバ内でこれをクリアします。操作は割り込み制御ポート【3 - 4 項】の PCIバスINT信号クリアビット（B 3）をセットします。但し同ビットを再びリセットするまではクリア状態で、次の割り込み要求が発信できない状態なので通常は続けてリセットします。この様子はステータスポートの PCIバス上のINT信号ビット（B 3）に反映されますが、通常はデバイスドライバ内ですぐクリアされるためユーザプログラムからの検出は困難です。（必要もないでしょう。）

PCIバス上の割り込み要求信号は《PCIバスINT出力許可》ビットをリセットすること、またはボード・リセット【3 - 3 項】でもクリアできます。

図 4 - 4 B . 割り込み制御・ステータス変化タイミング



上記タイミング図では各制御ビットの効果を示すため外部割り込み入力の有効エッジが必要以上に発生した場合で記してあります。

### ポーリングによるイベント制御

前 で説明したように割り込み制御ポートの外部入力割り込み許可ビット B 7 をセット、エッジ（B 6）を指定しておき、PCIバスINT出力許可ビット B 0 がリセット状態なら外部入力割り込み有効エッジが発生する毎に（割り込みは発生しませんが、）ステータスポートの割り込み要求フラグがセットされます。これをポーリングして外部入力の発生を検出する方法があります。（推奨します。）



## 4-5 . Quick - Basicのサンプル

Quick-Basic (4.5) 用のサンプルプログラム “583QB1 .BAS” は基本的なBASIC文のみによる使用例です。コーディングの詳細はソースのリストを御覧ください。  
 なお本プログラムの実行形式 “583QB1 .EXE” は試運転・動作確認用にもなります。

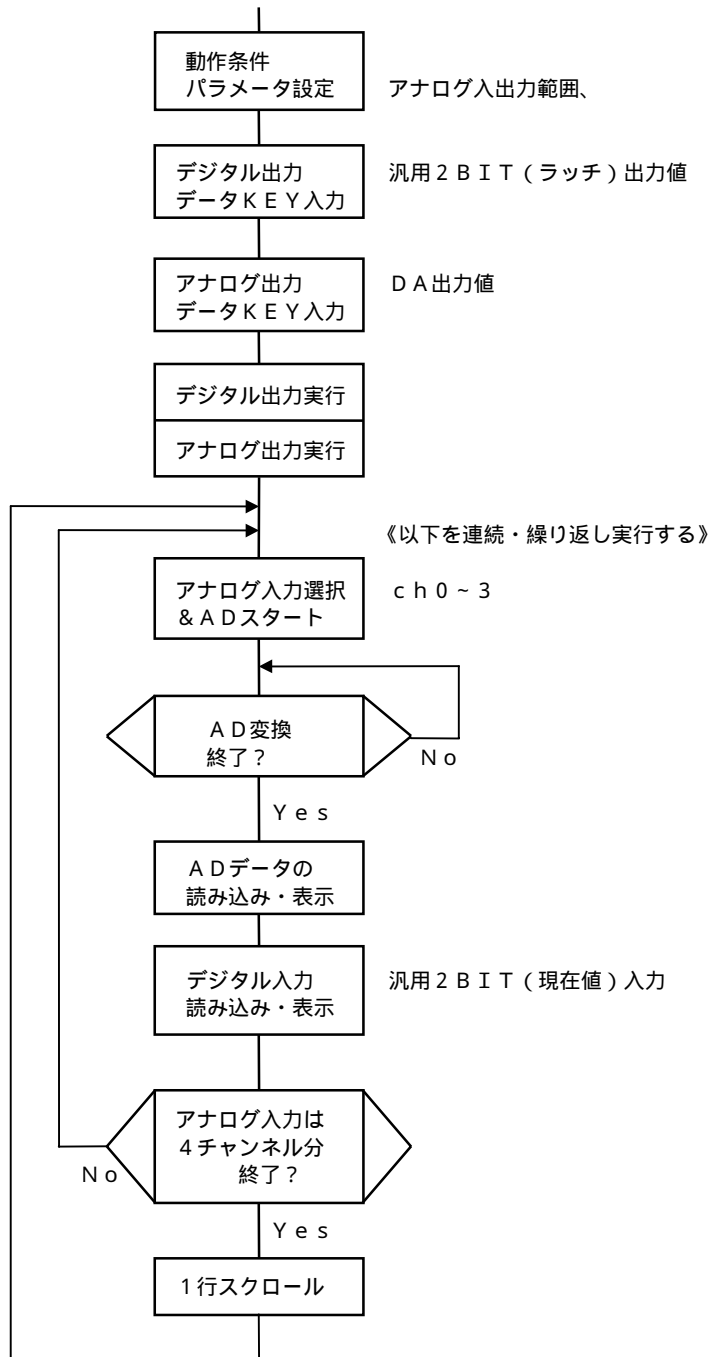


図4 - 5 . “583QB1” の  
フロー概要

また、連続・繰り返し実行部分を外部イベント（INT入力）に同期して行うサンプル  
 “583QB2 .BAS” もあります。

#### 4- 6 . C のサンプル

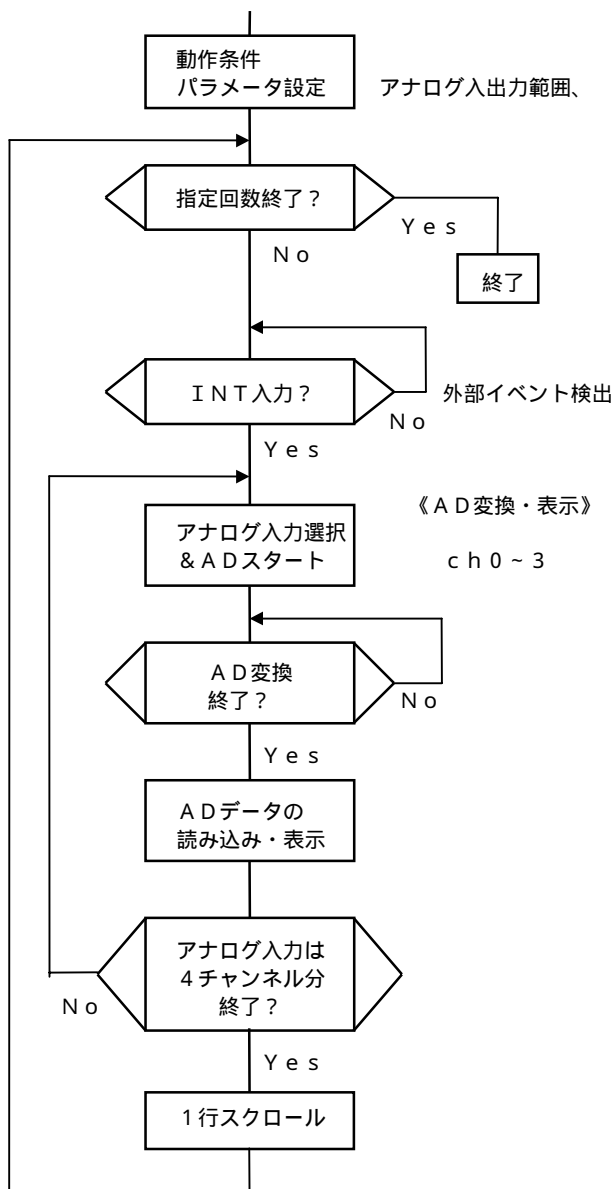
代表的な使用方法、アルゴリズムを具体化したサンプルソースがあります。TURBO - C、BORLAND - C、およびMS - Cでコンパイルすることができます。

以下、各ソースのフロー概要を記します。

P O L 5 8 3 . C

ソフト監視（ホーリング）で外部イベントを検出し、AD変換・表示します。

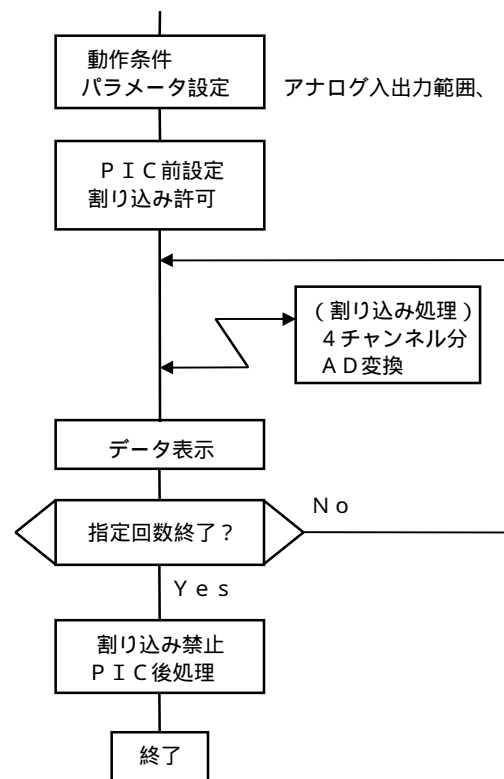
図4 - 6 A . “ P O L 5 8 3 . C ” フロ-概要



INT 5 8 3 . C

外部イベントで割り込み処理ルーチンを起動、ここでA/D変換・表示します。

図4 - 6 B . “ I N T 5 8 3 . C ” フロ-概要



MNL 583 . C

単に A/D 変換、D/A 変換、デジタル入出力を実行する例です。簡単なので特にフロー図は示しません。ソースを御参照ください。

## 4-7. Visual Basic (32BIT 版) のサンプル

WINDOWS 9x・MEまたはNT上で本ボードの機能を一通り動作させてみるものです。  
Visual Basic (4.0)で作成されており、VB (5.0/6.0)でも動作します。

当サンプルは当社提供のWINDOWS 9x・ME・NTの汎用I/Oドライバ/DLLを使用  
したものです。【1-4項参照】 第6章に記す専用ドライバ/関数DLL(推奨)にもVBサン  
プルがありますが、両者は無関係です。なお**本サンプルプログラム中の制御には割り込みを使用  
していません。** 割り込みリソースがあり、ドライバに登録した場合は外部割り込み入力要因  
に選択して(単に動作確認のため)発生回数を表示するだけです。

また**WINDOWS 9x・MEの場合で割り込みを使用しないときはドライバのインストールが  
不要です。**(9x・MEでのI/O操作はDLLが直接ハードウェアにアクセスするため)

【注1】 本ボードの(プラグアンドプレイに対する)コンフィギュレーション初期設定は  
割り込みリソースを要求しません。 割り込みを使用するためには添付のユーティ  
リティを利用して設定変更を行う必要があります。(1-4項参照)

【注2】 当サンプルで使用するWINDOWS版I/O実行DLL/デバイスドライバは  
4-1項の作業だけではインストールされません。 WINDOWS 9x・ME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。 NT4.0用はWinNTフォルダ中にあり、同フォルダ中の専用イン  
ストーラで導入してください。(1-4項参照)  
(追伸)CDROMの場合、Win9xおよびWinNTフォルダはINSTALL  
フォルダ下のDriversフォルダ下にあります。

表 4 - 7 .

ソフトウェア要素	OS	使用するモジュール/ファイル
デバイスドライバ、および インタフェースDLL	9x (ME)	pta95_0.vxd accs_95.dll
	NT (4.0)	NtPta_?sys (? : 0 ~ 15 任意の整数) port_nt.dll
Visual Basic サンプルプログラム モジュール構成	9x ME NT 共通	smp583.vbp (プロジェクト) smp583.frm (フォーム:メイン) drvdll.bas (DLL定義) def583.bas (ハードウェア定義)

本サンプルプログラム・ソースはWINDOWS 9x・ME・NTで共通ですが、使用する  
DLLが異なるのでDLL関数定義用の標準モジュール(drvdll.bas)先頭で、

```
#Const DRIVER = "Accs_95"      ' 9x・MEの場合(デフォルト)
' #Const DRIVER = "Port_Nt"    ' NT4.0の場合
```

の定義を条件コンパイルで対応、WINDOWS 9x・ME・NTでは関数名が異なるOPEN/  
CLOSEをAlias機能を使用してプログラム本文中では同一名で扱えるようにしています。

さらにOPEN操作はWINDOWS 9x・ME・NTの各DLLではパラメータが異なります  
から、frmMainのLOADイベント中で対応したパラメータをcboDrvParamに  
セット、その値を元にOPEN関数のパラメータとしています。

《操作方法》 操作手順は ドライバのオープン、 以後はボードの各機能実行の順です。  
 終了時は必ずドライバのクローズ操作を行います。  
 テキストボックスに記入する値は全てH E X表記です。  
 (例) 1 2 B I Tデータ8 0 0 Hなら3文字 “ 8 0 0 ” と入力します。

《実行画面》

### 各コントロール ( ボタン ) の動作概要

( B o a r d フ レ ム )

cmdOpen

デバイスドライバをオープンします。  
 W I N D O W S 9 x ・ M E の場合は当ボタンをクリックするだけ、  
 N T の場合は使用するデバイスドライバの枝番号 ( 0 ~ 15 ) を指定  
 してから当ボタンをクリックします。

cmdClose

デバイスドライバをクローズします。( 終了時 )

cmdBoard

プラグアンドプレイで割り当てられている I / O ベースアドレス  
 と割り込み番号が ( H E X で ) テキスト B O X に表示されます。  
 クリックするとボードがリセットされます。  
 ( 同時にボードの存在をチェックします。 )

## (Dioフレーム)

tmrGetDi

タイマのプロパティで設定した時間々隔で汎用デジタル入力と外部割り込み入力を調べ、シェイプコントロールのランプに表示します。

またデバイスドライバのGetInterCount()関数から得た割り込みの発生回数値をtxtInterCntに表示します。

cmdDioOut

txDoDataに入力された汎用デジタル出力値を更新出力します。出力値は次回操作まで保持されます。

(2ビットなので0~3と記入/3-10項参照)

## (Interruptフレーム)

cmdIntrEn

割り込みの発生回数(積算値)を示すIntrCount値をクリアし、本ボードからの割り込み信号出力を許可します。

cmdIntrDis

本ボードからの割り込み信号出力を禁止します。

## (A/Dフレーム)

cmdAdSamp

アナログ入力Ch0~3を各1回サンプリング/A/D変換を行い、(HEX)数値表示します。

Bit/Range/Mode/2'sCompのいずれかが前回から変更されている場合にはModeSetAdプロセスで各設定を更新したうえで行います。

(2'sComp:セットなら2の補数、リセットならバイナリ)

## (D/Aフレーム)

cmDaOutput

txtDaDataに入力された(Hex)値をDA変換・更新出力します。出力値は次回操作まで保持されます。

Bit/Range/Mode/2'sCompのいずれかが前回から変更されている場合にはModeSetDaプロセスで各設定を更新したうえで行います。

(2'sComp:セットなら2の補数、リセットならバイナリ)

**【注】** A/D、およびD/Aフレーム中の分解能(BIT)指定は標準の12BIT仕様ボードではデフォルト表示の12BITだけしか使用できません。オプションの16BIT仕様機に限り12/14/16BITの各指定が有効になります。



## 第5章．保守・その他

### 5-1. 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 - 5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 - 4項)  |
| (2) 割り込みレベル | リソースは取得できたか？(1 - 4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2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/図5 - 1参照） |

#### 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 - 5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、

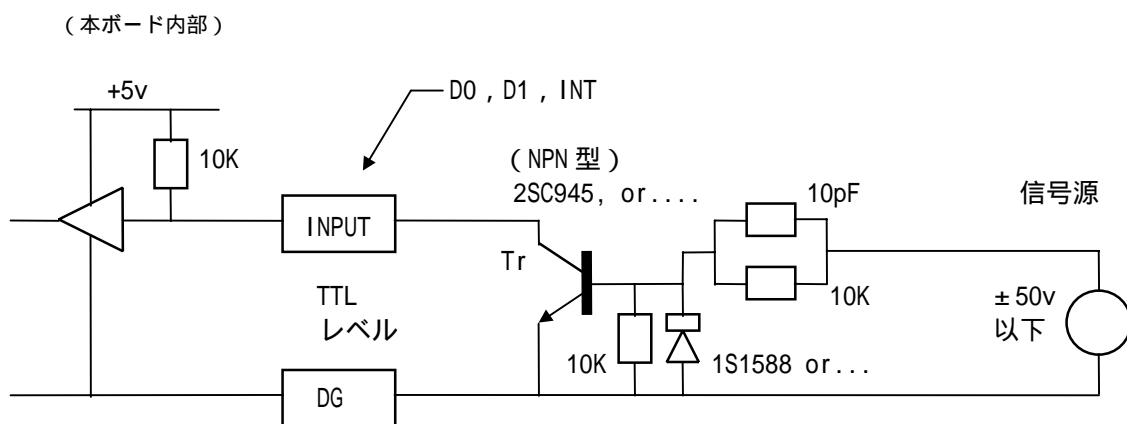
ファンクション・ジェネレータ等の交流信号出力を接続して破損させる例が多いようです。矩形波でも±に振れる信号は接続できません。特に、負側の許容レベル【-0.6V】が低いことに注意してください。

+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5-1. 【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。



## 5-2. 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。 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。

### 5-3. アナログ入出力範囲の再調整

動作テスト・確認の方法は【1 - 5 項】のとおりです。同テストから得られた値に入出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、バスイクステンダ、極細のドライバ、デジタル電圧計等を必要とし、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 スロットに装着、インストールします。

当作業の手順・詳細については1 - 4 項に記されています。

（以上、通常の使用状態）

図1 - 5のように、本ボードの任意チャンネルのAD入力を基準電圧源に接続します。またDAチャンネル出力をデジタル電圧計に接続します。デジタル入出力の接続は不要です。

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。

電源を投入、MS - DOSシステムを立上げます。再調整に使用するプログラムは試運転でも使用した“583QB1”です。> 583QB1【ENTER】でプログラムが走り始めます。

DA出力値およびデジタル出力値を指定して【Do】操作により、AD・DA各々目標値を得よう、オフセット調整とゲイン調整を交互に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出力範囲で実施してください。なお、本機の製造時はAD入力：±10v、DA出力：0～+10v範囲、各Aモードで最適調整されています。

#### AD入力調整

オフセット調整： 入力電圧が0vのとき、AD変換値（表示）が0vとなるようにTM - A0を調整する。

ゲイン調整： 入力電圧がフルスケール付近のとき、AD変換値（表示）が整合値となるようにTM - A1を調整する。

## D A (電圧) 出力調整

オフセット調整 : 指定出力値を 0 v ( 0 digit ) とし、出力電圧が 0 v となるように T M - D 0 を調整する。

ゲイン調整 : 指定出力値をフルスケール付近とし、出力電圧が指定値と一致するように T M - D 1 を調整する。

## D A (電流) 出力調整

電流出力付のモデルは 0 ~ + 1 0 v 出力をハード的に 4 ~ 2 0 m A に変換するものですから、最初に電圧出力調整を行ってから電流出力調整を実施してください。

オフセット調整 : 指定出力値を 0 v ( 0 digit ) とし、出力電流が 4 m A となるように T M - I 0 を調整する。

ゲイン調整 : 指定出力値を + 1 0 v ( 付近 ) とし、出力電流が 2 0 m A ( 整合値 ) となるように T M - I 1 を調整する。

$$\begin{aligned} \text{【注】電圧・電流出力値の換算式 : } & V = (10/16) I - (10/4) \\ & (v) \quad (mA) \qquad I = (16/10) V + 4 \end{aligned}$$

## 得られる正確度

通常、パソコン用のボードでは製造・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、ボード自体の性能を規定する相対正確度 (= 較正可能限度) 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 5 - 3 A . 調整時の精度一覧

対象	アナログ入出力範囲	非直線性	相対正確度	絶対正確度
A D	最終調整範囲 : 12BIT / $\pm 10$ v	0 . 0 1 %FS	0 . 0 6 %FS	0 . 0 9 %FS
	その他の全範囲		0 . 0 8 %FS	0 . 1 1 %FS
D A	最終調整範囲 : 12BIT / 0 ~ +10v	0 . 0 1 %FS	0 . 0 6 %FS	0 . 0 7 2 %FS
	その他の全範囲		0 . 0 8 %FS	0 . 0 9 2 %FS

## 定義

- ( a ) 非直線性 : 使用される A D / D A 変換素子に固有の性能。 ———— 【注 2】  
 ( b ) 相対正確度 : 非直線性を含む、回路全体の性能 ( 較正可能限度 ) 。 ———— 雑音を  
 ( c ) 絶対正確度 : 相対正確度に較正測定器の正確度を加算した値 ( 常温 ) 。 ———— 含まず。

【注 1】 当社では正確度 0 . 0 3 %、または 0 . 0 1 2 % 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、経年変化のデータや保証が無いことにも御注意ください。

【注 2】 当表の値には C P U を含む固有のシステム全体から発生する雑音が含まれていません。  
 当雑音は 1 2 ビット A D 機で 1 LSB ( 0 . 025 %FS ) 程度が普通です。  
 瞬時を 1 回だけ A D 変換した値には当雑音を考慮する必要があります。

## 5-4. 16ビット精度オプションによる仕様変更

(MFU-503AT/51xAT/573PCI/583PCI) 共通

### Opt16-1. アナログ仕様の変更

これらの機種は標準ロット生産時点では12ビット精度で製造・調整されています。

16ビット精度オプションの場合は標準品のAD変換素子 (and/or) DA変換素子を高精度の製品に交換、またアナログ回路中の精密抵抗をさらに高精度品に一部交換/追加実装したうえで再度調整することになります。これで(ソフト指定で)12ビット/14ビットでの各入出力範囲、および16ビットの $\pm 10\text{V}$  /  $\pm 5\text{V}$  入出力範囲が使用できるようになります。

【注1】16ビットではユニポーラ ( $0 \sim 5\text{V}$  /  $0 \sim 10\text{V}$ ) 入出力範囲が使用できません。

#### <精度>

通常、パソコン用AD/DA変換ボードでは製造・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、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) を示すことはできます。そこで当社では以下の表記方法を採用しています。

(1) 当社の製造・調整環境による常温での製造時絶対正確度を (正確度として) 記す。

(2) 非直線性、相対正確度、校正測定器の正確度は必要に応じて併記する。

#### 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 (常温)

【注3】

雑音を

含まず。

#### 16ビット精度オプション付加時の精度一覧

対応	アナログ入出力範囲	非直線性	相対正確度	絶対正確度
16ビット AD	最終調整範囲: 16BIT / $\pm 10\text{V}$	0.004 %FS	0.008 %FS	0.038 %FS
	その他の全範囲		0.028 %FS	0.058 %FS
16ビット DA	最終調整範囲: 16BIT / $\pm 10\text{V}$	0.002 %FS	0.006 %FS	0.018 %FS
	その他の全範囲		0.026 %FS	0.038 %FS

#### 12ビット調整時の精度一覧

対応	アナログ入出力範囲	非直線性	相対正確度	絶対正確度
12ビット AD	最終調整範囲: 12BIT / $\pm 10\text{V}$	0.01 %FS	0.06 %FS	0.09 %FS
	その他の全範囲		0.08 %FS	0.11 %FS
12ビット DA	最終調整範囲: 12BIT / $0 \sim +10\text{V}$	0.01 %FS	0.06 %FS	0.072 %FS
	その他の全範囲		0.08 %FS	0.092 %FS

【注2】 当社では正確度0.03%、または0.012%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。

当表の値は16ビット機では不足感もありますが、現場環境の違いを加味すればこれ以上の正確度を求めるには実機による現場環境での校正が必要かと思われます。

なお、経年変化のデータや保証が無いことにも御注意ください。

【注3】 当表の値にはCPUを含む固有のシステム全体から発生する雑音が含まれていません。当雑音は12ビットAD機で1LSB (0.025%FS)、16ビットADで4LSB (0.0062%FS) 程度が普通です。

瞬時を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。

## Opt16 - 2 . アナログ入出力範囲

本機の入出力範囲・モード・分解能 ( 1 6 / 1 4 / 1 2 B I T ) はソフトウェア選択です。  
選択方法は次 Opt16 - 3 項 を御参照ください。

本機は【 ± 1 0 v 範囲 / A モード 】で最終調整されていますが、高精度部品の使用により  
入出力範囲を切り替えても多くの用途では再調整の必要がありません。【 Opt16 - 1 項参照】  
特定の入出力範囲で最も正確度を良くしたいときは再調整を行ってください。  
御希望により当社でも ( 有償で ) 行います。

公称入出力範囲を正直に本機の各分解能 ( 1 6 / 1 4 / 1 2 B I T ) で実現すると 1 digit  
当りの電圧値が半端な割り切れない値【 B 】モードになってしまいます。そこで当社では  
入出力範囲を少し拡大して 1 digit 当りの電圧が切りのよい値となる【 A 】モードもサポート  
しています。

表 Opt16 - 2 A . 《 1 2 ビット分解能 》でのアナログ入出力範囲

公称入出力範囲	【モード】	実際の入出力範囲	分解能 [ mv/digit ]
0 ~ + 1 0 v	【 A 】	0 ~ + 1 0 . 2 3 7 5	2 . 5
0 ~ + 1 0 v	【 B 】	0 ~ + 9 . 9 9 7 6	2 . 4 4 .....
0 ~ + 5 v	【 A 】	0 ~ + 5 . 1 1 8 7 5	1 . 2 5
0 ~ + 5 v	【 B 】	0 ~ + 4 . 9 9 8 8	1 . 2 2 .....
± 1 0 v	【 A 】	- 1 0 . 2 4 0 ~ + 1 0 . 2 3 5 0	5 . 0
± 1 0 v	【 B 】	- 1 0 . 0 0 0 ~ + 9 . 9 9 5 1	4 . 8 8 .....
± 5 v	【 A 】	- 5 . 1 2 0 ~ + 5 . 1 1 7 5	2 . 5
± 5 v	【 B 】	- 5 . 0 0 0 ~ + 4 . 9 9 7 6	2 . 4 4 .....

表 Opt16 - 2 B . 《 1 4 ビット分解能 》でのアナログ入出力範囲

公称入出力範囲	【モード】	実際の入出力範囲	分解能 [ mv/digit ]
0 ~ + 1 0 v	【 A 】	0 ~ + 1 6 . 3 8 3	1 . 0
0 ~ + 1 0 v	【 B 】	0 ~ + 9 . 9 9 9 3 9	0 . 6 1 .....
0 ~ + 5 v	【 A 】	0 ~ + 8 . 1 9 1 5	0 . 5
0 ~ + 5 v	【 B 】	0 ~ + 4 . 9 9 9 6 9	0 . 3 1 .....
± 1 0 v	【 A 】	- 1 6 . 3 8 4 ~ + 1 6 . 3 8 2	2 . 0
± 1 0 v	【 B 】	- 1 0 . 0 0 0 ~ + 9 . 9 9 8 7 8	1 . 2 2 .....
± 5 v	【 A 】	- 8 . 1 9 2 ~ + 8 . 1 9 1	1 . 0
± 5 v	【 B 】	- 5 . 0 0 0 ~ + 4 . 9 9 9 3 9	0 . 6 1 .....

表 Opt16 - 2 C . 《 1 6 ビット分解能 》でのアナログ入出力範囲

公称入出力範囲	【モード】	実際の入出力範囲	分解能 [ mv/digit ]
± 1 0 v	【 A 】	- 1 3 . 1 0 7 2 0 ~ + 1 3 . 1 0 6 8 0	0 . 4
± 1 0 v	【 B 】	- 1 0 . 0 0 0 0 0 ~ + 9 . 9 9 9 6 9	0 . 3 0 5 .....
± 5 v	【 A 】	- 6 . 5 5 3 6 0 ~ + 6 . 5 5 3 4 0	0 . 2
± 5 v	【 B 】	- 5 . 0 0 0 0 0 ~ + 4 . 9 9 9 8 5	0 . 1 5 3 .....

表 Opt16 - 2 A , B , C において ± 1 0 v を超える値は理論値です。 アナログ回路に使用  
されている素子の仕様から、 ± 1 0 v を超える A D / D A 変換値の正確度は保証されません。

# 伝達関数

ソフト上で指定する分解能によって以下のとおりです。  
A/D変換の場合で示します。D/A変換の場合は“A/D”を“D/A”に、“入力”を“出力”に読み換えてください。

12BIT指定： 12ビットA/Dの分解能は“2の12乗分の1”ですから、A/Dデータとアナログ入力電圧の関係は以下ようになります。

$$\begin{aligned} \text{分解能} \quad R_{\text{es}} &= V_{\text{span}} \div 4096 \quad [\text{v} / \text{digit}] \\ \text{A/Dデータ} \quad D_{\text{ad}} &= V_{\text{in}} \div R_{\text{es}} \quad [\text{digit}] \quad / \text{ユニポーラの時} \\ D_{\text{ad}} &= (V_{\text{in}} \div R_{\text{es}}) + 2048 [\text{digit}] \quad / \text{バイポーラの時} \\ \text{入力電圧} \quad V_{\text{in}} &= D_{\text{ad}} \times R_{\text{es}} \quad [\text{v}] \quad / \text{ユニポーラの時} \\ V_{\text{in}} &= (D_{\text{ad}} - 2048) \times R_{\text{es}} [\text{v}] \quad / \text{バイポーラの時} \end{aligned}$$

【注】  $V_{\text{span}}$  はA/D入力範囲の絶対幅です。具体的には表Opt16-2Aの総範囲幅に1digit分の電圧値を加算した値です。例えばAモードの公称±10vなら20.480v（Bモードなら20v）です。

14BIT指定： 14ビットA/Dの分解能は“2の14乗分の1”ですから、A/Dデータとアナログ入力電圧の関係は以下ようになります。

$$\begin{aligned} \text{分解能} \quad R_{\text{es}} &= V_{\text{span}} \div 16384 \quad [\text{v} / \text{digit}] \\ \text{A/Dデータ} \quad D_{\text{ad}} &= V_{\text{in}} \div R_{\text{es}} \quad [\text{digit}] \quad / \text{ユニポーラの時} \\ D_{\text{ad}} &= (V_{\text{in}} \div R_{\text{es}}) + 8192 [\text{digit}] \quad / \text{バイポーラの時} \\ \text{入力電圧} \quad V_{\text{in}} &= D_{\text{ad}} \times R_{\text{es}} \quad [\text{v}] \quad / \text{ユニポーラの時} \\ V_{\text{in}} &= (D_{\text{ad}} - 8192) \times R_{\text{es}} [\text{v}] \quad / \text{バイポーラの時} \end{aligned}$$

【注】  $V_{\text{span}}$  はA/D入力範囲の絶対幅です。具体的には表Opt16-2Bの総範囲幅に1digit分の電圧値を加算した値です。例えばAモードの公称±10vなら32.768v（Bモードなら20v）です。

16BIT指定： 16ビットA/Dの分解能は“2の16乗分の1”ですから、A/Dデータとアナログ入力電圧の関係は以下ようになります。

$$\begin{aligned} \text{分解能} \quad R_{\text{es}} &= V_{\text{span}} \div 65536 \quad [\text{v} / \text{digit}] \\ \text{A/Dデータ} \quad D_{\text{ad}} &= V_{\text{in}} \div R_{\text{es}} \quad [\text{digit}] \quad / \text{ユニポーラの時} \\ D_{\text{ad}} &= (V_{\text{in}} \div R_{\text{es}}) + 32768 [\text{digit}] \quad / \text{バイポーラの時} \\ \text{入力電圧} \quad V_{\text{in}} &= D_{\text{ad}} \times R_{\text{es}} \quad [\text{v}] \quad / \text{ユニポーラの時} \\ V_{\text{in}} &= (D_{\text{ad}} - 32768) \times R_{\text{es}} [\text{v}] \quad / \text{バイポーラの時} \end{aligned}$$

【注】  $V_{\text{span}}$  はA/D入力範囲の絶対幅です。具体的には表Opt16-2Cの総範囲幅に1digit分の電圧値を加算した値です。例えばAモードの公称±10vなら26.2144v（Bモードなら20v）です。

図 2 - 2 A . バイポーラ入出力  
【Aモード】

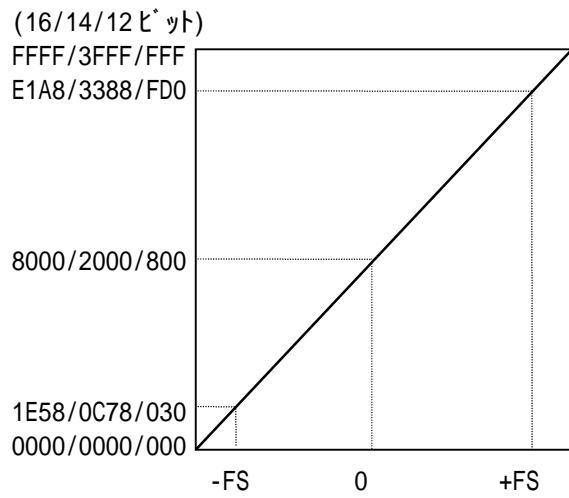


図 2 - 2 B . ユニポーラ入出力  
【Aモード】

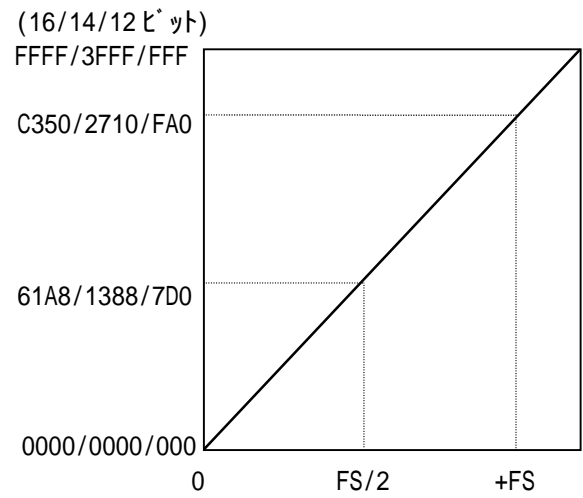


図 2 - 2 C . バイポーラ入出力  
【Bモード】

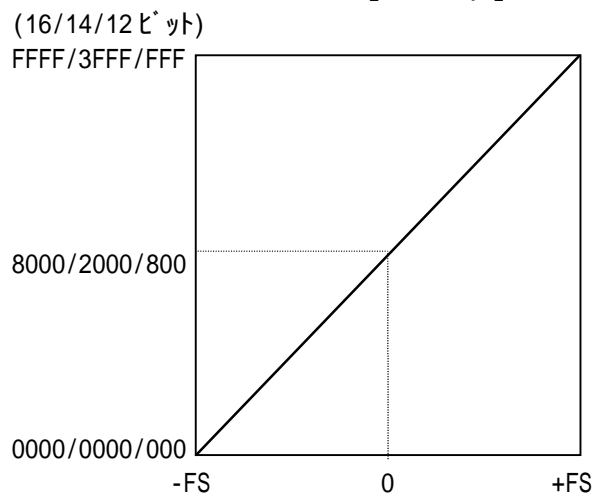
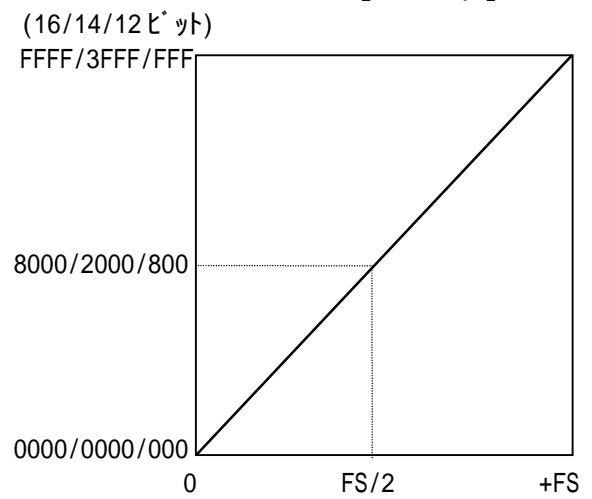


図 2 - 2 D . ユニポーラ入出力  
【Bモード】



### Opt16 - 3 . 分解能、および入出力範囲の設定操作

各ボード共、**アナログ入出力範囲の設定ポート**中、2ビット（B 3 , B 2 ）が分解能を指定することになります。 その他は変更ありません。

表 Opt16 - 3 A . アナログ入力（分解能・入力範囲）設定ポートの構成

ビット	各ビットの機能・意味	= 1 のとき	= 0 のとき	リセット時
B 7	未使用			0
B 6	未使用			0
B 5	A/Dデータコード指定	2の補数	バイナリ	0
B 4	アナログ入力範囲モード指定	Bモード	Aモード	0
B 3	アナログ入力	当値により3種類から選択。 【表 Opt16 - 3 D】		0
B 2	分解能（A/Dデータ・ビット長）指定			0
B 1	（公称）アナログ	当値により4レンジから選択。 【表 Opt16 - 3 C】		0
B 0	入力範囲指定			0

表 Opt16 - 3 B . アナログ出力（分解能・出力範囲）設定ポートの構成

ビット	各ビットの機能・意味	= 1 のとき	= 0 のとき	リセット時
B 7	未使用			0
B 6	未使用			0
B 5	D/Aデータコード指定	2の補数	バイナリ	0
B 4	アナログ出力範囲モード指定	Bモード	Aモード	0
B 3	アナログ出力	当値により3種類から選択。 【表 Opt16 - 3 D】		0
B 2	分解能（D/Aデータ・ビット長）指定			0
B 1	（公称）アナログ	当値により4レンジから選択。 【表 Opt16 - 3 C】		0
B 0	出力範囲指定			0



表 Opt16 - 3 C . 入出力範囲指定データ

B 1	B 0	アナログ入出力範囲
1	1	± 5 v
1	0	± 1 0 v
0	1	0 ~ + 5 v 【注 1】
0	0	0 ~ + 1 0 v 【注 1】

表 Opt16 - 3 D . 分解能指定データ

B 3	B 2	分解能
1	1	強制 0 v 出力
1	0	1 6 ビット 【注 1】
0	1	1 4 ビット
0	0	1 2 ビット

【注 1】 1 6 ビットではユニポーラ（0 ~ 5 v / 0 ~ 1 0 v）入出力範囲が使用できない。

【注 2】 電源投入、およびリセット操作直後はアナログ入力・出力共に 1 2 ビットの A モード / 0 ~ + 1 0 v 範囲となっている。（アナログ出力は 0 v 状態）



#### Opt16 - 4 . A D ( D A ) データ構造

A D , D A 共に下位 8 ビット、上位 8 ビットの 2 バイトに分けて読み書きします。

各ポートの I / O アドレスは変わりません。

D A 出力の場合は必ず上位・下位の順に書き込みます。( A D 入力 は 任意の順で読み込み可 )

表 Opt16 - 4 A . 上位データ読み書きポートのビット構成

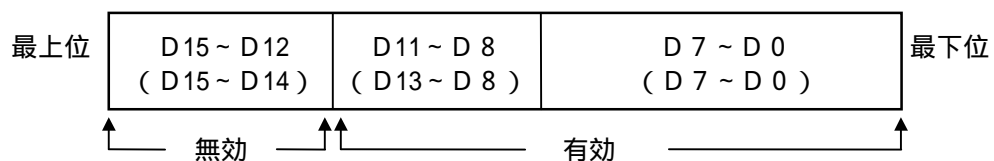
ビット	1 6 ビット指定の場合	1 4 ビット指定の場合	1 2 ビット指定の場合
B 7	D 15 ( 最上位ビット )	D 15	D 15
B 6	D 14	D 14	D 14
B 5	D 13	D 13 ( 最上位ビット )	D 13
B 4	D 12	D 12	D 12
B 3	D 11	D 11	D 11 ( 最上位ビット )
B 2	D 10	D 10	D 10
B 1	D 9	D 9	D 9
B 0	D 8	D 8	D 8

表 Opt16 - 4 B . 下位データ読み書きポートのビット構成

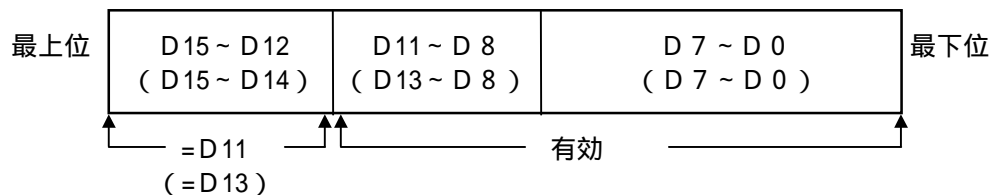
ビット	1 6 ビット指定の場合	1 4 ビット指定の場合	1 2 ビット指定の場合
B 7	D 7	D 7	D 7
B 6	D 6	D 6	D 6
B 5	D 5	D 5	D 5
B 4	D 4	D 4	D 4
B 3	D 3	D 3	D 3
B 2	D 2	D 2	D 2
B 1	D 1	D 1	D 1
B 0	D 0 ( 最下位ビット )	D 0 ( 最下位ビット )	D 0 ( 最下位ビット )

1 2 ビット ( 1 4 ビット ) 指定時の最上位 4 B I T ( 2 B I T ) について。

バイナリ指定の場合：データ 1 語中、最上位 4 B I T ( 2 B I T ) は無効です。



2 の補数指定の場合：データ 1 語中、最上位 4 B I T ( 2 B I T ) は有効なデータの最上位ビット D 11 ( D 13 ) と同一値となります。



( 1 6 ビット精度オプションによる仕様変更 ) 以上。

## 5-5. 付録 (WINDOWS 2000 / X Pについて)

### WINDOWS 2000

ボードのインストール： W I N D O W S 2 0 0 0 は N T 4.0 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 W I N D O W S 2000 対応のインストールディスク（当社製 / v r 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 D L L 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 - X P

ボードのインストールからドライバ、ハンドラ関数 D L L まで、添付の W I N D O W S 2 0 0 0 用ソフトウェアがそのまま御利用いただけます。

## 第6章. WINDOWSハンドラ

追加：1999年12月

本ボードMFU-583/581PCIをVC、VC++、VBで簡単に使用することのできるWINDOWS9x・ME・NT・2000・XP用ハンドラ（+ドライバ）です。本ボードの基本機能が関数化されており、さらにソフト的タイマによる自動入出力（波形入出力）や外部イベントに同期した入出力機能もサポートされています。ユーザは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。

## 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）、メモリ16MB以上

OS / コパイラ：WINDOWS9x・ME・NT・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)

対応ボード：MFU-583PCI、またはMFU-581PCI（DA出力機能なし）

チャンネル数：AD, DA, 各1チャンネル（ボード数1枚のみ）

内部クロック：1ms～3600s周期  
（パソコン内）

誤差：最大+40μs / 誤差は累積しない。  
WINDOWS9x、Pentium400MHzのとき。

外部クロック：最高追従速度1ms（WINDOWS9x、Pentium400MHzのとき。）

サンプリング速度：=クロック（上記）、CPU速度に多少依存。

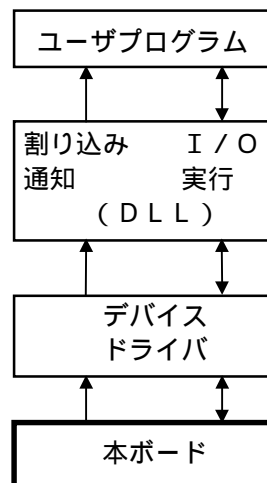
入出力データ点数：拡張メモリ空容量（1語=2バイト）

割り込み：不使用。

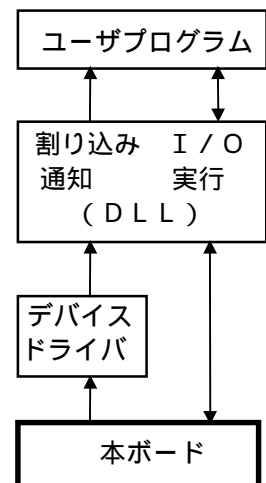
LabVIEW等で利用可能（サンプルv iソース参照）

図6-1A. ソフトウェア構造

（WINDOWS NT/2000/XP の場合）



（WINDOWS9x・ME の場合）



## マニュアル入出力動作は

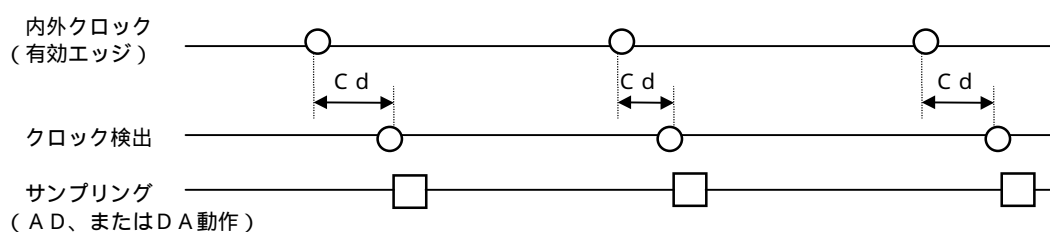
アプリケーションプログラム上任意のプロセスから即時AD入力、または即時DA更新出力する関数があります。

## 自動サンプリング動作は

内部クロック、外部クロック、共に本ハンドラはソフト的に検出します。そのため(WINDOWSシステムの性格上)特に排他的なソフトやデバイスが存在しない状態でも検出遅れ $C_d$ があり、これは(pentium/400MHzのとき)最大 $+40\mu s$ 程度です。  
なお、この遅れ誤差は累積しません。

- (1) AD, DA個別 : 内部クロックを使用して、AD、DAのいずれかを単独で自動サンプリング動作させる場合、または両方とも自動サンプリング動作させる場合でスタートを別々にする場合は各々専用のクロックがあるように動作します。

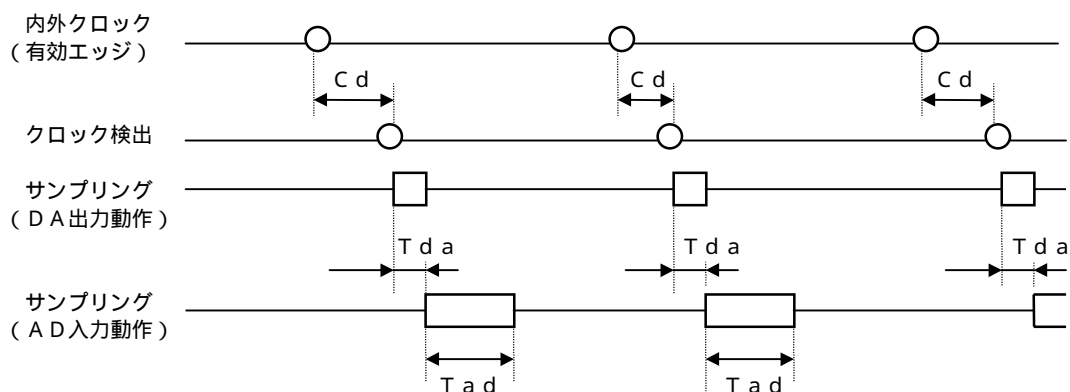
図6-2B. クロック/サンプリング(AD入力、またはDA出力動作)

【注1】  $C_d = 0 \sim 40\mu s$ 【注2】 DA出力動作時間は最大 $20\mu s$   
AD入力動作時間は最大 $50\mu s / ch$ 

at pentium/400MHz

- (2) AD, DA同時 : ADとDAの各自動サンプリングを同時にスタートさせる場合は、同一クロックでDA出力動作を先に、続けてAD入力動作を行う順になります。

図6-2B. クロック/サンプリング(AD入力&amp;DA出力動作)

【注1】  $C_d = 0 \sim 40\mu s$ 【注2】 DA出力動作時間 $T_{da}$  = 最大 $20\mu s$   
AD入力動作時間は最大 $50\mu s / ch$ 

at pentium/400MHz

## 6-2. 使用準備

まず第4章（4-1項）にしたがって本ボード関連ファイルをインストールします。  
次に本ハンドラDLL / 専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。

WINDOWS 95・98・MEの場合

- (1) ¥WINDOWS フォルダ に “ H 5 8 x \_ 9 5 . D L L ” をコピーする。
- (2) ¥WINDOWS¥SYSTEM フォルダに “ D 5 8 x \_ 9 5 . V X D ” をコピーする。

WINDOWS NT (4.0) の場合 / Administrator レベルで行う /

- (1) ¥WINDOWS¥SYSTEM32 フォルダに “ H 5 8 x \_ N T . D L L ” をコピーする。
- (2) ¥WINDOWS¥SYSTEM32¥DRIVERS フォルダに “ D 5 8 x \_ N T . S Y S ” をコピーする。
- (3) 任意のフォルダにユーティリティ “ D 5 8 x R e g . e x e ” をコピーする。

【注】ファイル名中の末尾文字 はバージョン変更時に追加（初期は無し）されます。

### 《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 5 8 x R e g）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) 本関数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¥Mfu583 以下）を使用できます。

### 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

WINDOWSハンドルの引き渡し、すなわちメッセージングの利用ができない場合は6-4項【1】初期化関数で引数Owner = 0とすればメッセージング機能が無効となります。この場合の使用方法はアプリケーション側からステータス関数をポーリングして各関数実行タイミングを得ることを前提にしています。

## 6-3. ユーザプログラム記述

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6-2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。

表 6 - 3 . 関数一覧

関数名	機能・動作	主なパラメータ等
【1】M583_Open_Sys	ボード、本ハンドラの初期化	ウインドウハンドル
【2】M583_Reset_Board	全ボードのリセット	
【3】AD_Set_SampMode	AD入力モードの設定	チャンネル数、波形入力バッファ
【4】AD_Set_Range	AD入力範囲、分解能、コードの設定	範囲、分解能、コード
【5】AD_Get_OneScan	即時1回、マニュアルAD入力	
【6】AD_Set_Clock	AD用クロック源、クロック値の設定	クロック源、周期、単位
【7】AD_Read_DLLData	DLL管理バッファからADデータ転送	波形入力データ
【8】DA_Set_SampMode	DA出力モードの設定	波形出力データ
【9】DA_Set_Range	DA出力範囲、分解能、コードの設定	範囲、分解能、コード
【10】DA_Out_Prompt	DA即時更新出力	DAデータ
【11】DA_Set_Clock	DA用クロック源、クロック値の設定	クロック源、周期、単位
【12】DA_Write_DLLBuf	DLL管理バッファへのDAデータ転送	波形出力データ
【13】M583_Start_Samp	波形入出力（サンプリング）のスタート	対象、実行回数、トリガ、
【14】M583_Stop_Samp	波形入出力（サンプリング）の強制停止	
【15】M583_Get_Status	ボード・ステータスの取得	ボード・ステータス
【16】M583_Out_Aux	汎用デジタル（ラッチ）出力	更新データ
【17】M583_Inp_Aux	汎用デジタル（現在値）入力	
【18】M583_Close_Sys	本ハンドラの終了	
【19】M583_Get_Libver	本ハンドラのバージョン情報取得	

----- AD , DA , D I O マニュアル制御手順 -----

開始 — (1) 初期化を行う。 【 M583\_Open\_Sys( ) 】

ここでは当ボードの存在やP n Pで設定されたリソースを本ハンドラが認識すると同時にボードリセット、その他、ハンドラ内の参照テーブルやデータバッファを初期化する。

DA — (2) DA出力モードを即時更新（マニュアル）に設定する。 【 DA\_Set\_SampMode( ) 】  
 (3) DA出力範囲、分解能、コードを設定する。 【 DA\_Set\_Range( ) 】  
 (4) DA即時更新（マニュアル）出力。 【 DA\_Out\_Prompt( ) 】

AD — (5) AD入力モードを即時（マニュアル）入力で設定する。 【 AD\_Set\_SampMode( ) 】  
 (6) AD入力範囲、分解能、コードを設定する。 【 AD\_Set\_Range( ) 】  
 (7) AD即時（マニュアル）入力。 【 AD\_Get\_OneScan( ) 】

DO — (8) 汎用2 B I Tデジタル（ラッチ）出力。 【 M583\_Out\_Aux( ) 】

DI — (9) 汎用2 B I Tデジタル（現在値）入力。 【 M583\_Inp\_Aux( ) 】

終了 — (10) 終了処理。 【 M583\_Close\_Sys( ) 】

----- A D and/or D A 波形入出力（自動サンプリング）制御手順 -----

- (1) 初期化を行う。 【 M583\_Open\_Sys( ) 】

ここでは当ボードの存在や P n P で設定されたリソースを本ハンドラが認識すると同時にボードリセット、その他、ハンドラ内の参照テーブルやデータバッファを初期化する。

- (2) D A 出力モードを自動サンプリング（波形出力）に設定する。 【 DA\_Set\_SampMode( ) 】  
 (3) D A 出力範囲、分解能、コードを設定する。 【 DA\_Set\_Range( ) 】  
 (4) D A 用クロック源、クロック値の設定。 【 DA\_Set\_Clock( ) 】  
 (5) 出力波形 D A データを本 D L L の管理バッファに転送する。 【 DA\_Write\_DLLBuf( ) 】

D A 準備完了。

- (6) A D 入力モードを自動サンプリング（波形出力）に設定する。 【 AD\_Set\_SampMode( ) 】  
 (7) A D 入力範囲、分解能、コードを設定する。 【 AD\_Set\_Range( ) 】  
 (8) A D 用クロック源、クロック値の設定。 【 AD\_Set\_Clock( ) 】

A D 準備完了。

- (9) スタート 【 M583\_Start\_Samp( ) 】

波形出力のみ、波形入力のみ、または両方同時スタートができる。  
 波形出力・波形入力には各専用クロックを使用できる。

波形出力・波形入力、各々指定回数の自動サンプリングが終了はメッセージで通知する。

- (10) 自動サンプリングの強制停止。（途中で中止する場合） 【 M583\_Stop\_Samp( ) 】  
 (11) 入力波形 A D データを D L L の管理バッファから転送する。 【 AD\_Read\_DLLData( ) 】  
       （ユーザプログラムの管理バッファに読み込む。）  
 (12) 終了処理。 【 M583\_Close\_Sys( ) 】

## 6-4. 関数仕様、エラーコード

初期化、動作条件設定、スタート、ストップ、ステータス取得等、各関数は波形入出力を実現する基本機能単位となっています。なお、MFU-581PCIの場合はDA部がありませんから同機能に関する部分を無視して御使用ください。

また、各関数は自身の性格から適切な実行手順があります。（前6-3項・参照）

----- 以下の【1】【2】は本ボードの機能（AD，DA，DIO）に共通の制御関数 -----

### 【1】本ハンドラのオープン、および初期化

int M583_Open_Sys ( HWND Owner )	
Owner	親のウィンドウハンドル
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6-4参照）
機能・動作	<p>プラグアンドプレイで設定された（本ボードの）I/Oアドレスを認識すると同時に、ボードリセット、その他、ハンドラ内の参照テーブルやデータバッファ等を初期化する。</p> <p>当関数実行直後のDA部は0V出力、また即時更新出力モード【8】となっており、出力範囲等を設定【9】すれば、いつでも1データ即時更新出力【10】が可能。 また波形出力する（自動サンプリングする）ときは【8】【9】【11】【12】の条件設定後、【13】でスタート。</p> <p>当関数実行直後のAD部は即時入力モード【3】となっており、入力範囲等を設定【4】すれば、いつでも1データ即時入力【5】が可能。 また波形入力する（自動サンプリングする）ときは【3】【4】【6】の条件設定後、【13】でスタート、【7】でADデータ列を取得。</p>

#### 重 要

本ハンドラDLLを（LabVIEW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】でウィンドウハンドル **HWND Owner** に **NULL** 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波形出力終了メッセージを受け取ることができません。

### 【2】本ボード（MFU-583PCI、またはMFU-581）のリセット

int M583_Reset_Board ( 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6-4参照）
機能・動作	<p>全ての制御レジスタは初期状態に戻る。（汎用デジタル出力は変化しない） アナログ出力は0Vに戻る。 なお【1】初期化実行直後には必要ない。（本関数の動作は【1】初期化の一部）</p>



----- 以下の【3】～【7】はAD部専用の制御関数 -----

### 【3】AD入力モードの設定

<code>int AD_Set_SampMode(int num_ch, int buf_dll)</code>	
<code>num_ch</code>	使用するチャンネル数 (1 ~ 4)
<code>buf_dll</code>	AD入力モード指定 / 0 : 即時入力 (1回マニュアルサンプリング) 1 : 指定クロックによる自動サンプリング (波形入力)
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	AD入力チャンネル数、およびタイミング条件を設定する。

### 【4】AD入力範囲、分解能、コードの設定 《注》分解能パラメータはバージョン 2 以降。

<code>int AD_Set_Range(int resol, int range, int range_mode, int data_code)</code>	
<code>resol</code> <code>range</code> <code>range_mode</code> <code>data_code</code>	AD分解能。 / 0 : 12ビット、 1 : 14ビット、 2 : 16ビット AD入力範囲。 / 0 : 0 ~ 10V、 1 : 0 ~ 5V、 2 : ±10V、 3 : ±5V AD入力範囲モード。 / 0 : Aモード、 1 : Bモード (2-3項参照) AD入力データコード。 / 0 : バイナリ、 1 : 2の補数
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 参照)
機能・動作	全チャンネル共通のAD入力範囲、分解能、データコードを設定します。 なお分解能は標準 12ビットです。(14ビット、16ビットはハードウェアオプション)

### 【5】即時AD入力 (1回マニュアルサンプリング) 実行

<code>int AD_Get_OneScan(WORD *buf, int bufsize)</code>	
<code>*buf</code> <code>bufsize</code>	転送先バッファ。(ユーザプログラムの管理するバッファ) 転送先バッファの容量。(バイト)
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	【3】で指定したチャンネル数分について即時各1回のAD入力 (サンプリング) を実行し、ユーザプログラムの管理するバッファ領域に転送する。

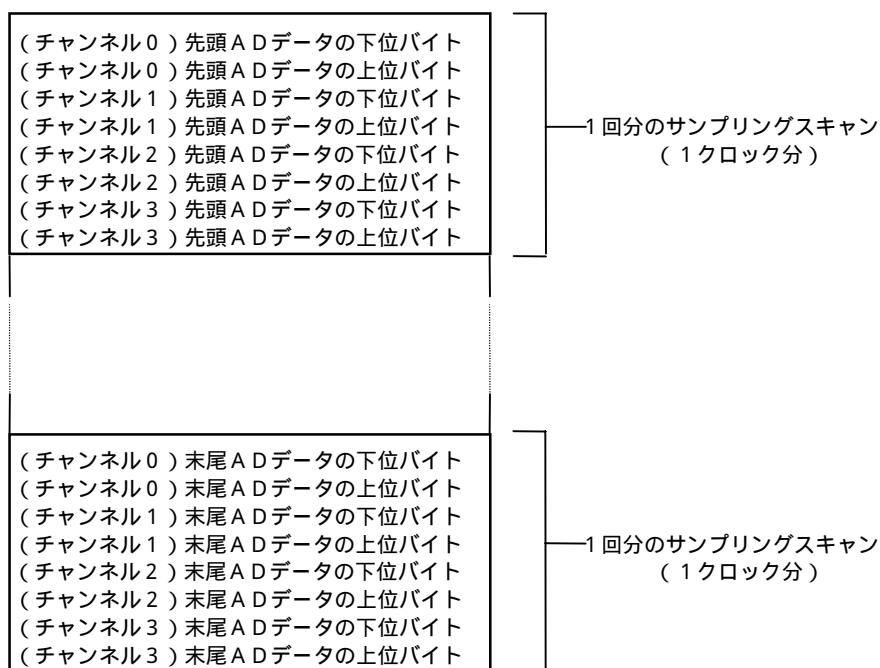
### 【6】ADサンプリング・クロックの設定 (自動サンプリング = 波形入力に使用)

<code>int AD_Set_Clock(int clk_source, int time_unit, int clk_period)</code>	
<code>clk_source</code> <code>time_unit</code> <code>clk_period</code>	クロック源。 / 0 : ソフトタイマ、 1 : 外部入力 (負エッジ)、 2 : 外部入力 (正エッジ) (クロック源 = ソフトタイマのとき) 指定クロック周期値の単位。 / 0 : s、 1 : ms (クロック源 = ソフトタイマのとき) 指定クロック周期値。 / int 整数
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 参照)
機能・動作	自動サンプリング入力のために使用するクロックを指定します。 外部 (割込) 入力をクロック源に指定したときの <code>time_unit</code> , <code>clk_period</code> は無効、また外部トリガは使用できません。 / 参照: 【13】サンプリング・スタート

## 【7】（本DLLの管理するバッファからの）ADデータ転送

<pre>int AD_Read_DLLData(int num_data,                     int data_pos, WORD *buf, int bufsize)</pre>	
num_data	転送する、1チャンネル当りのADデータ点数。（図6-4の構造）
data_pos	転送するADデータの先頭位置。（サンプリング順番号）
*buf	転送先バッファ。（ユーザプログラムの管理するバッファ）
bufsize	転送先バッファの容量。（バイト）
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	自動サンプリングの結果として本DLLの管理するバッファ領域にあるADデータを【3】で指定したチャンネル数分について、ユーザプログラムの管理するバッファ領域に転送する。

図6-4A. 波形入力データ格納バッファ内のデータ配置（4チャンネル実行の場合）



----- 以下の【8】～【12】はD A部専用の制御関数 -----

### 【8】D A出力モードの設定

int DA_Set_SampMode(int buf_dll)	
buf_dll	D A出力モード指定 / 0 : 即時更新D A出力 ( 1回マニュアルサンプリング ) 1 : 指定クロックによる自動サンプリング ( 波形出力 )
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	D A出力タイミング条件を設定する。

### 【9】D A出力範囲、分解能、コードの設定 《注》 分解能パラメータはバージョン 2 以降。

int DA_Set_Range(int resol, int range, int range_mode, int data_code)	
resol range range_mode data_code	D A分解能。 / 0 : 12ビット、 1 : 14ビット、 2 : 16ビット D A出力範囲。 / 0 : 0 ~ 10 v、 1 : 0 ~ 5 v、 2 : ± 10 v、 3 : ± 5 v D A出力範囲モード。 / 0 : Aモード、 1 : Bモード ( 2 - 3項参照 ) D A出力データコード。 / 0 : バイナリ、 1 : 2の補数
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 参照 )
機能・動作	D A出力範囲、分解能、データコードを設定します。 なお分解能は標準 12ビットです。( 14ビット、16ビットはハードウェアオプション )

### 【10】即時更新D A出力 ( 1回マニュアルサンプリング ) 実行

int DA_Out_Prompt(int upd_Prompt)	
upd_prompt	更新D A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 - 4 参照 )
機能・動作	即時更新D A出力を実行する。( 更新値は次回更新、またはリセットまで保持される。 )

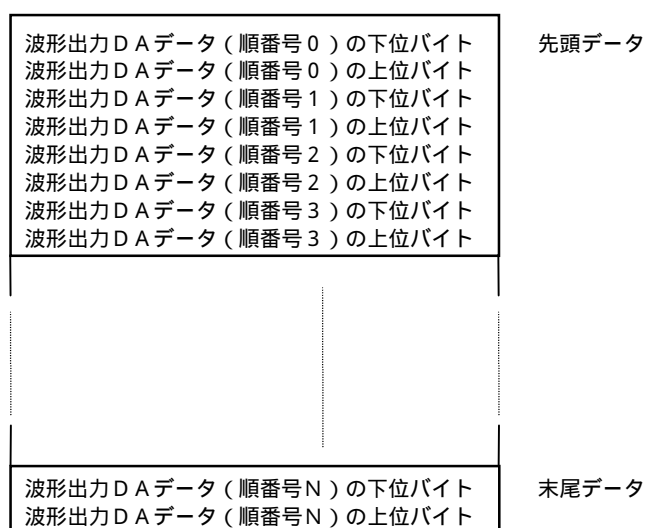
### 【11】D Aサンプリング・クロックの設定 ( 自動サンプリング = 波形出力に使用 )

int DA_Set_Clock(int clk_source, int time_unit, int clk_period)	
clk_source	クロック源。 / 0 : ソフトタイマ、 1 : 外部入力 ( 負エッジ )、 2 : 外部入力 ( 正エッジ )
time_unit	( クロック源 = ソフトタイマのとき ) 指定クロック周期値の単位。 / 0 : s、 1 : ms
clk_period	( クロック源 = ソフトタイマのとき ) 指定クロック周期値。 / int 整数
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 参照 )
機能・動作	自動サンプリング出力のときに使用するクロックを指定します。 外部 ( 割込 ) 入力をクロック源に指定したときのtime_unit, clk_periodは無効、また外部トリガは使用できない。 / 参照 : 【13】サンプリング・スタート /

## 【12】（本DLLの管理するバッファへの）DAデータ転送

<pre>int DA_Write_DLLBuf(int num_samp,                     int num_data, WORD *bufptr)</pre>	
num_samp	DAデータ総数。（図6-4の構造：本DLLの管理するバッファ）
num_data	今回転送するDAデータ数。
*bufptr	転送元バッファのポインタ。（ユーザプログラムの管理するバッファ）
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	波形出力（指定クロックによる自動サンプリング）用DAデータを一括、または分割して 本DLLの管理するバッファ領域に転送する。

図6-4B．波形出力データ格納バッファ内のデータ配置



----- 以下の【13】～【15】はA/D部，D/A部共用の制御関数 -----

### 【13】自動サンプリング（波形入力 and/or 波形出力）のスタート

<pre>int M583_Start_Samp(int start_mode, int ad_num_samp,                     int loop_count, int trig)</pre>	
start_mode	スタート対象。 / 0 : A/D部のみ、 1 : D/A部のみ、 3 : A/D and D/A同時
ad_num_samp	A/Dサンプリング点数（1チャンネル当たり）= 実行クロック数
loop_count	D/Aサンプリング・ループ回数。 / 0 : 停止操作まで無限、 1 ~ N : 指定回数
trig	トリガ。 / 0 : ソフト（即スタート）、 1 : 外部入力（ ）、 2 : 外部入力（ ）
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	自動サンプリングを開始（またはトリガ待ち開始）させる。 A/D，D/A，各独自のクロックを使用できる。 自動サンプリング動作はトリガ（スタート）時に初回のサンプリングが行われ、以後は指定クロック周期で行われる。 外部（割込）入力をクロック源に指定しているときはトリガとしての使用はできない。

### 【14】自動サンプリングの強制停止

<pre>int M583_Stop_Samp(int stop_mode)</pre>	
stop_mode	停止対象。 / 0 : A/D部のみ、 1 : D/A部のみ、 2 : A/D，D/A両方
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	自動サンプリングを停止させる。

### 【15】ボードステータスの取得

<pre>int M583_Get_Status(int *adsampled, int *adbussy,                     int *dabussy, int *status)</pre>	
*adsampled *adbussy *dabussy *status	サンプリング済みA/Dデータ点数 A/D変換フラグ。 / 0 : 変換終了、 1 : 変換中 D/A転送フラグ。 / 0 : 転送終了、 1 : 転送中（ボード内部にて） ボードステータス生データ（3-8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 参照）
機能・動作	

----- 以下の【16】【17】はD I O部専用の制御関数 -----

#### 【16】汎用デジタル出力

<code>int M583_Out_Aux(int out_data)</code>	
<code>out_data</code>	デジタル出力データ（2ビット）
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	汎用2BITデジタル出力の更新・保持。（ラッチ出力）

#### 【17】汎用デジタル入力

<code>int M583_Inp_Aux(void)</code>	
引数	なし
戻り値	正常終了時： 汎用2BIT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6-4参照）
機能・動作	汎用2BITデジタル入力。

----- 以下の【18】【19】は本ボードの機能（AD，DA，D I O）に共通の関数 -----

#### 【18】本ハンドラDLL / ドライバのクローズ

<code>int M583_Close_Sys(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	終了処理。

#### 【19】 本ハンドラのバージョン取得

<code>int M583_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>0x101</code> となります。

エラー                    本ハンドラの各関数は実行前後（または実行中）に不適当なパラメータや動作状態を検出するとエラーコードを返してきます。

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。	M583_Open_Sys
- 2	IDの不一致。	M583_Open_Sys
- 3	ドライバファイルが見つからない。	M583_Open_Sys
- 4	ドライバファイルのバージョンが違う。	M583_Open_Sys
- 5	初期化 ( M583_Open_Sys ) が未実行。	
- 6	AD自動サンプリング開始に必要な全ての設定がなされていない。	M583_Start_Samp
- 7	DA自動サンプリング開始に必要な全ての設定がなされていない。	M583_Start_Samp
- 10	使用するADチャンネル数指定エラー。	AD_Set_SampMode
- 11	AD入力モードの指定が不適当。	AD_Set_SampMode
- 12	AD分解能の指定が不適当。	AD_Set_Range
- 13	AD入力範囲の指定が不適当。	AD_Set_Range
- 14	AD入力範囲モードの指定が不適当。	AD_Set_Range
- 15	ADデータコードの指定が不適当。	AD_Set_Range
- 16	ADクロック源の指定が不適当。	AD_Set_Clock
- 17	ADクロック周期単位の指定が不適当。	AD_Set_Clock
- 18	ADクロック周期の指定が不適当。	AD_Set_Clock
- 19	ADクロック周期の値が範囲外。	AD_Set_Clock
- 20	ADサンプリング済み点数を超えて読もう（転送しよう）とした。	AD_Read_DllData
- 21	ADサンプリングデータ転送先バッファエラー	AD_Read_DllData
- 22	ADサンプリングデータ転送先バッファ容量エラー	AD_Read_DllData
- 25	DA出力モードの指定が不適当。	DA_Set_SampMode
- 26	DA分解能の指定が不適当。	DA_Set_Range
- 27	DA出力範囲の指定が不適当。	DA_Set_Range
- 28	DA出力範囲モードの指定が不適当。	DA_Set_Range
- 29	DAデータコードの指定が不適当。	DA_Set_Range
- 30	DAクロック源の指定が不適当。	DA_Set_Clock
- 31	DAクロック周期単位の指定が不適当。	DA_Set_Clock
- 32	DAクロック周期の指定が不適当。	DA_Set_Clock
- 33	DAクロック周期の値が範囲外。	DA_Set_Clock
- 34	DAデータ総数の指定が不適当。	DA_Write_DLLBuf
- 35	今回転送するDAデータ数の指定が不適当。	DA_Write_DLLBuf
- 36	DAデータ総数を超えて書き込もう（転送しよう）とした。	DA_Write_DLLBuf
- 40	スタート対象の指定が不適当。	M583_Start_Samp
- 41	ADサンプリング点数の指定が不適当。	M583_Start_Samp
- 42	DAサンプリング・ループ回数の指定が不適当。	M583_Start_Samp
- 43	トリガ条件の指定が不適当。	M583_Start_Samp
- 44	外部クロックと外部トリガの同時使用はできない。	M583_Start_Samp
- 45	自動サンプリング停止対象の指定が不適当。	M583_Stop_Samp
- 50	ADサンプリング中。	M583_Start_Samp
- 51	ADサンプリングは停止している。	M583_Stop_Samp
- 52	AD入力モードはマニュアルに設定されている。	M583_Start_Samp
- 53	ADバッファメモリ確保エラー	M583_Start_Samp
- 54	使用するADチャンネル数を指定していない。	M583_Get_OneScan
- 60	DAサンプリング中。	M583_Start_Samp
- 61	DAサンプリングは停止している。	M583_Stop_Samp
- 62	DA出力モードは自動サンプリングに設定されている。	DA_Out_Prompt
- 63	DA出力モードは即時更新（マニュアル）に設定されている。	DA_Write_DLLBuf
- 64	DAバッファメモリ確保エラー	DA_Write_DLLBuf

# 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	MFU-583 / 581 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分以内に回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		
TEL		( 所在地 )
FAX		