

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

製品添付のソフトウェアは3.5インチ(1.44MB)フロッピーまたはCDに圧縮された形で格納されており、インストーラ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(CDROM)
インストール先：Cドライブ(HDD) 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:¥INSTALL¥PCI¥DIO¥DIO275 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DIO-275PCI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。

(2005年4月より以前のCDROMを使用する場合：DOS窓利用)

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¥DIO¥DIO275【ENTER】
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きます。

= 設定例 = COMMAND.COMがCドライブの¥にある場合、
>SET COMSPEC=C:¥COMMAND.COM【ENTER】

【注】 W I N D O W S 9 x ・ M E ・ N T 用の汎用I/O実行DLLとデバイスドライバ(本機専用ではない!)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 x ・ M E 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。 またNT用はWinNTフォルダ中にあるので、同フォルダ中の専用インストーラで導入してください。(1-5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Driver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

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。

T - C : TURBO - C、B - C : BORLAND - C

```

¥
|
MSCIENCE
|
|   UTILITY      WIN95      CF9050DP.COM : コンフィギュレーション (95用)
|   (本ボードの設定) |
|   | - 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 : デバイスドライバ登録・削除用
|
| - - BOARDTST - - - 275QB1.EXE : 本ボードの試運転・動作確認用プログラム
|   |               | - 275QB1.COM : 英語モードに切り替えた後、EXEを実行する
|
| - - SMP275C - - - - MICROSOFT.H : MS-C用ヘッダファイル
|   (各種Cサンプル) | - - BORLAND.H : TURBO-C, BORLAND用ヘッダファイル
|   |               | - - INT275.C : ストローブ入力割り込み時に入出力動作
|   |               | - - POL275.C : ストローブ入力フラグをポーリングして入出力動作
|   |               | - - PRW275.C : 単純入出力動作
|
|   SMP275B      275QB1.BAS : Quick-Basic (4.5) 用サンプル
|
|   Hnd__2K      Dio275      Dll : ハンドラDLL
|   (WINDOWS2000/XP/98用ハンドラ) | - sys : 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 : VC++ (5.0) 用サンプル
|   |               | - Delphi3 : Delphi (3.0) 用サンプル等
|
|   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

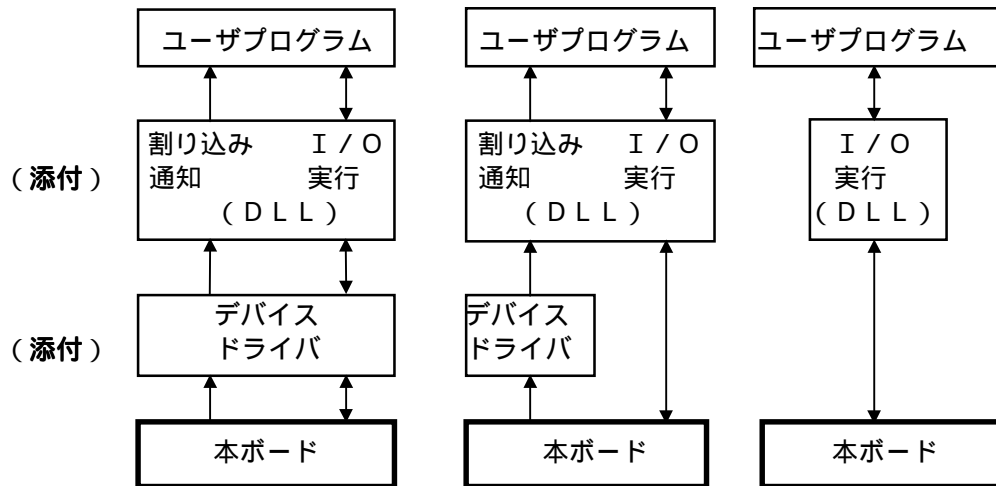
```

4-2 . WINDOWSドライバについて

WINDOWS 9 x / ME / NT / 2000 / XP の場合はI / O読み書き用の汎用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。WINDOWS 2000 / XP / 98用の本ボード専用ドライバ／関数ライブラリ（第6章）を利用する場合は不要です。

（WINDOWS NT・2000/XP の場合）（WINDOWS9x・ME の場合）（WINDOWS 3.1 の場合）



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（orME）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（orXP）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からも利用します。（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 = 010FH (DIO-275PCI)。

【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 (3-1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 | x 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figuration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 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。)

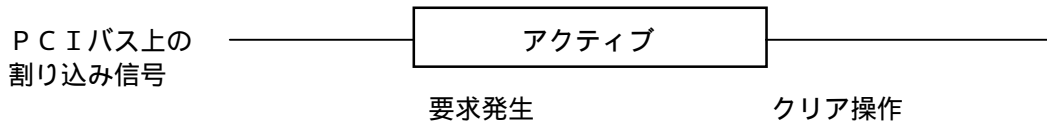
| | |
|---|---|
| Int ReadPciByte (WORD magic , WORD reg , BYTE _far *data) | |
| 引数 | magic : 【1】GetPciDevice で得られた マジック番号、 *data : データ取得先ポインタ、 reg : PCI コンフィギュレーション空間ヘッダ領域のレジスタ番号。 |
| 戻り値 | 0:成功、 -1 : 失敗 |

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

P C I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I S A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、P C I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O F F）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W I N D O W S 9 5 ・ 9 8 ・ N T用デバイスドライバは汎用のため、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D L L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R O M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。

割り込み制御の手順（& デバイスドライバ内での操作）

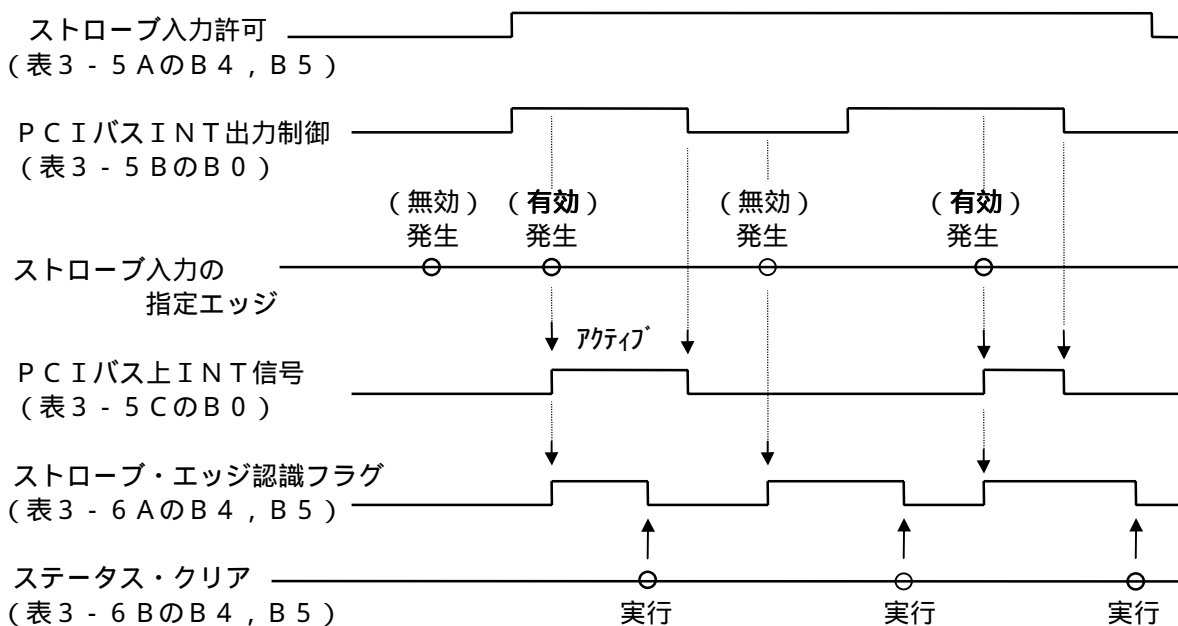
ストローブ入力制御ポート【3-5項】でストローブ入力許可ビット（B4 or B5）をセットし、同有効極性を（B0 or B1）で指定します。これだけの場合は、有効な信号エッジが発生しても《PCIバス上への割り込み要求信号出力》はアクティブになりません。ステータス【3-6項】のエッジ認識フラグ（B4 or B5）がセットされるだけです。さらに、表3-5Bに記すPCIバスINT出力制御ビットB0をセットしておくことによって実際の割り込み要求信号が出力（アクティブ）可能になります。

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【表3-5B】のPCIバスINT出力制御ビットB0をクリアします。但し同ビットを再び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再セットします。この様子は【表3-5C】のPCIバス上INT信号ビットB0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INT信号出力はボード・リセット【3-2項】でもクリアできます。

図4-4B. 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、ストローブ入力の指定エッジが必要以上に発生した場合で記してあります。

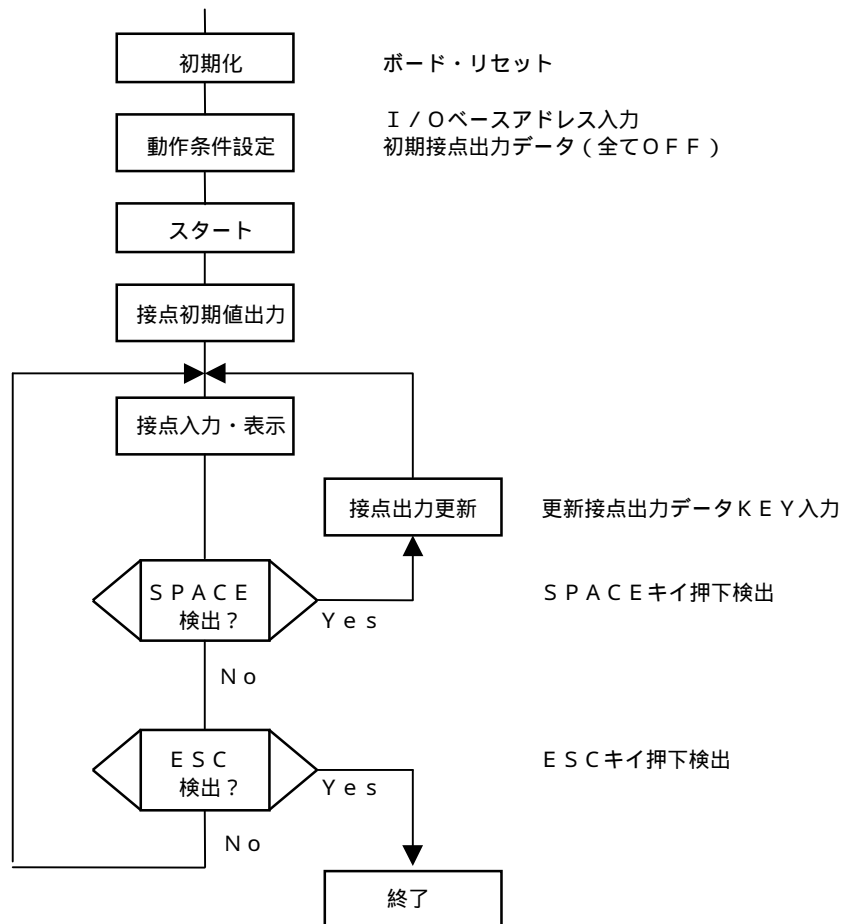
ポーリングによるイベント制御（割り込み不使用）

前 で説明したようにストローブ入力制御ポートのストローブ入力許可ビットをセットしておき、PCIバス上のINT出力制御ビットB0がクリア状態なら、ストローブ入力の有効な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エッジ認識フラグがセットされます。これをポーリングしてイベント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“275QB1.BAS”は基本的なBASIC文のみによる使用例です。なお本プログラムの実行形式“275QB1.EXE”は試運転・動作確認用にもなります。コーディングの詳細は同ソースのリストを御覧ください。

図4-5. “275QB1.BAS”のフロ-概要



4-6 . Cのサンプル (16bitコード専用)

ストローブ指定エッジ入力のポーリングによる接点入出力、 同割り込みによる入出力例です。 以下にアルゴリズムの概要を記します。 具体的にはソースを御参照ください。

図4-6A . ストローブ指定エッジ入力のポーリングによる入出力【POL275 . C】

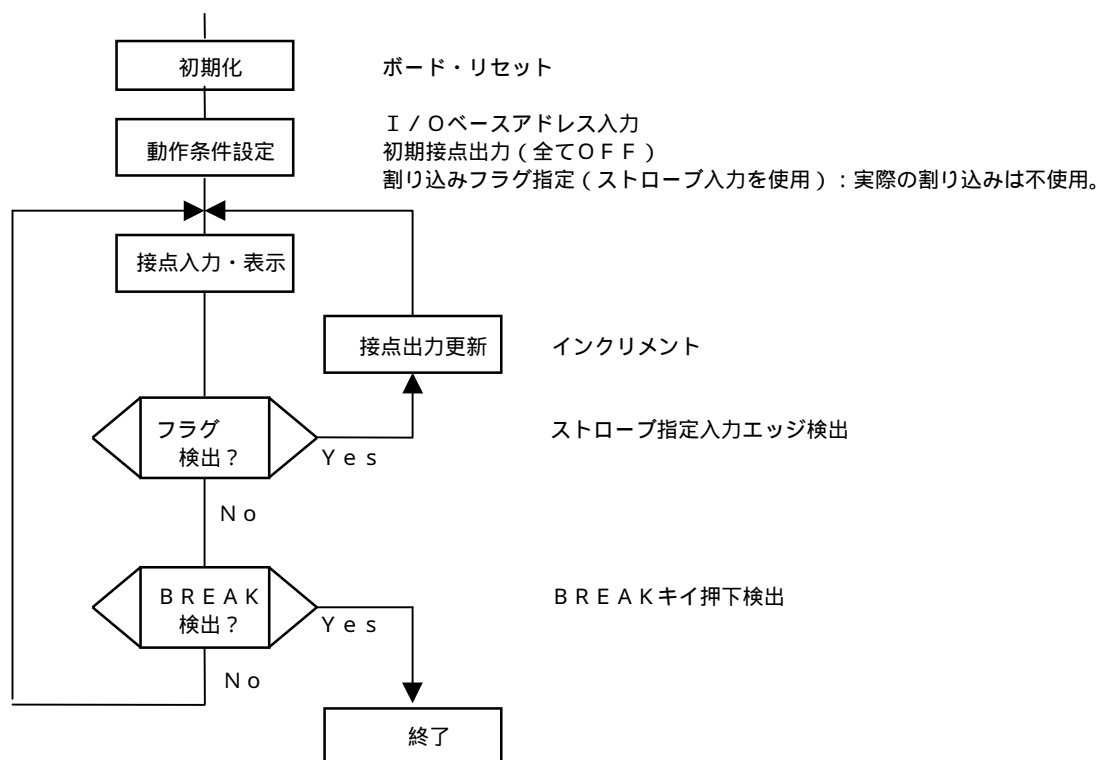
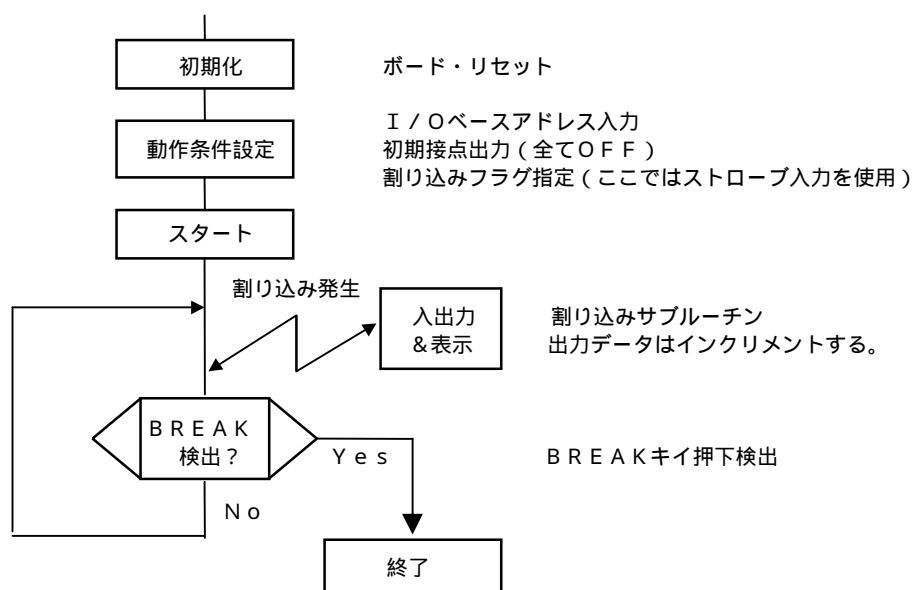


図4-6B . ストローブ指定エッジ入力の割り込みによる入出力【INT275 . C】



第6章. WINDOWSハンドラ

DIO-275PCIをVC、VB等で簡単に使用することのできるWINDOWS2000、XP、98・ME用のハンドラDLL(+ドライバ)です。本ボードの基本機能が関数化されており、ユーザは御自身の記述するメインルーチンの中から呼び出して使用することができます。

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 : WINDOWS2000・XP、または98・ME / 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)
供給メディア : 本ボード添付のサンプルディスク内。
対応ボード : DIO-275PCI
最大8枚まで対応 (同期動作なし)
【実現機能】
8ビット×2ポート現在値入力
入力状態の監視機能 : 専用スレッドで監視、変化検出時にメッセージ発信。
ストローブ更新入力 : 専用スレッドで監視、更新 " " "
8ビット×2ポート (ラッチ) 更新出力
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。
市販ソフト対応 : LabVIEW等からも利用可能。

6-2. 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。

複数の当ボードを使用するときは、各ボード上のスイッチSW-BNの設定値が当ハンドラでのボード番号になります。“0”から順に連続した値を設定しておきます。

ソフトウェアは第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 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/XP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(H275_2K.dll)もボードインストール時に所定の ¥WINDOWS¥SYSTEM32 にコピーされているので、即サンプル(¥MSCIENCE¥HND_2K¥Dio275 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、VXD型ドライバで使用する前提で設定されていますが、

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

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

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

- (3) WINDOWS 2000と同一の方法・手順で本ボードをインストールする。
- (4) 念のため、デバイスマネージャで本ボードの登録を確認します。
- (5) 本関数DLL(H275_2K.dll)もボードインストール時に所定の ¥WINDOWS¥SYSTEM32 にコピーされているので、即サンプル(¥MSCIENCE¥HND_2K¥Dio275 以下)を使用できます。

LabVIEW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。（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 . 制御関数一覧

| 関数名 | 機能・内容 | 備考 |
|---------------------------|-----------------------|-------------|
| 【1】Dio275_Open_Diosys | ボード、および本ハンドラの初期化 | |
| 【2】Dio275_Inp_16bit | 8 接点 × 2 ポート入力 | 入力データ |
| 【3】Dio275_Out_16bit | 8 接点 × 2 ポート（ラッチ）更新出力 | 出力データ |
| 【4】Dio275_Set_Mode | 入力モード設定 | 現在値 / ストロープ |
| 【5】Dio275_In0x_PortWatch | 入力ポート 0 の監視ビット指定 | |
| 【6】Dio275_In1x_PortWatch | 入力ポート 1 の監視ビット指定 | |
| 【7】Dio275_Start_PortWatch | （監視動作、ストロープ動作）の開始 | 監視スレッド動作開始 |
| 【8】Dio275_Stop_PortWatch | （監視動作、ストロープ動作）の停止 | 監視スレッド動作停止 |
| 【9】Dio275_Inp_Status | ステータス取得 | |
| 【10】Dio275_Close_Diosys | 本ハンドラの終了 | |
| 【11】Dio275_Get_Libver | 本ハンドラのバージョン情報取得 | |

便利な入力・監視機能

本ハンドラには《入力監視スレッド》が用意されており、
指定入力ビットの変化を検出したときにメッセージを発信する動作、
各ポート専用のストロープ入力を検出したときにメッセージを発信する動作が可能です。
但しメッセージ発信・受信のプロセスはWINDOWS管理下にあり、数ms～数10msを要することに御注意ください。

入力変化監視動作：通常《現在値入力モード》では入力関数【2】Dio275_Inp_16bitで現在の各ポート・各ビットの入力状態を得ますが、監視ビット指定関数【5】Dio275_In0x_PortWatch 【6】Dio275_In1x_PortWatch を実行すると、その時の指定監視ビット入力が読み込み記憶され、監視スタート関数【7】Dio275_Start_PortWatch 以後に指定ビットの変化を検出するとメッセージを発信します。但し当動作は1回限りです。再度の動作には監視ビット指定関数【5】【6】の再実行が必要です。
片方ポートのみ監視を中止するには、監視ビット指定関数の指定データ = 0 として実行します。

ストロープ入力動作：《ストロープ入力モード》では各ポート別の専用ストロープ信号の指定エッジ（関数【4】Dio275_Set_Mode で指定）が入力される毎に該当ポート（8bit）入力が本ボード内部にラッチ・保存され、入力関数【2】Dio275_Inp_16bit では最後に更新・保存されたデータを得ます。
当モードで監視スタート関数【7】Dio275_Start_PortWatch を実行すると、本ボード内部ラッチの更新・保存動作（各ポート毎）が発生するたびにメッセージが発信されます。

《 動作の中止》 監視ストップ関数【8】Dio275_Stop_PortWatch を実行します。

6-4. 関数仕様・エラーコード

以下に各関数の仕様・詳細を記します。

【1】初期化

| | |
|-------|--|
| 書式 | int Dio275_Open_DioSys (HWND Owner, int num_board) |
| 引数 | Owner : ウィンドウハンドル (メッセージングに使用。) num_board : 使用ボード数 (1 ~ 8) |
| 戻り値 | 正常終了時 : ボードのID / DIO-275PCI : H 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。 なお、Owner = 0 とすればメッセージング機能が無効となり、 LabVIEW等からも利用可能になる。 |

【2】汎用デジタル (接点) 入力の読み込み

| | |
|-------|--|
| 書式 | int Dio275_Inp_16bit(int board_no, int &in_1x, int &in_0x) |
| 引数 | board_no : 対象ボード番号 (0 ~ 7) in_1x : 1 xポート入力データ (8ビット現在値) in_0x : 0 xポート入力データ (8ビット現在値) |
| 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 | 汎用デジタル入力ポートから16ビット (8ビット×2) の値を読み込む。 現在値入力モード (初期状態) に設定されている入力ポートは現在値を読み込む。 またストローブ入力モードに設定されている入力ポートは、 最新のストローブ (指定) 有効エッジによってラッチ保存された値を読み込む。 |

【3】汎用デジタル (接点) ラッチ出力の更新書き込み

| | |
|-------|---|
| 書式 | int Dio275_Out_16bit(int board_no, int out_1x, int out_0x) |
| 引数 | board_no : 対象ボード番号 (0 ~ 7) out_1x : 1 xポート (ラッチ) 出力更新データ out_0x : 0 xポート (ラッチ) 出力更新データ |
| 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 | 汎用デジタル出力ポート16ビット (8ビット×2) に更新データを書き込む。 更新された出力データは次の更新、またはパワーオンリセットまで保持される。 なお、ソフト的なリセット (制御部リセット操作) ではクリアされない。 |

【4】入力モード設定

| | |
|-------|--|
| 書式 | int Dio275_Set_Mode (int board_no, int inp0_mode, int inp1_mode) |
| 引数 | board_no : 対象ボード番号 (0 ~ 7) inp0_mode : 0xポートの入力モード / 0 : 現在値入力、1 : ストローブ入力・立下 ()、 0 : ストローブ入力・立上 () inp1_mode : 0xポートの入力モード / 0 : 現在値入力、1 : ストローブ入力・立下 ()、 0 : ストローブ入力・立上 () |
| 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 | 入力ポートの動作モード、および (ストローブ入力モード時の) ストローブ動作極性を指定する。 |

【5】入力ポート0の監視ビット指定、【6】入力ポート1の監視ビット指定

| | |
|-------|---|
| 書式 | 【5】int Dio275_In0x_PortWatch (int board_no, int Watch_Bit) 【6】int Dio275_In1x_PortWatch (int board_no, int Watch_Bit) |
| 引数 | board_no : 対象ボード番号 (0 ~ 7) Watch_Bit : 監視ビット指定データ |
| 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 | 《現在値入力モード》で入力変化監視動作を行うときの監視ビットを指定する。 指定方法は監視対象入力ビット = 1 とした 8 ビットパターン値、 例えば入力ビット 7、3、2、1、0 ならパターンは (1 0 0 0 1 1 1 1) = 8 F H とする。 |

【7】監視・検出&メッセージ発信動作のスタート

| | |
|-------|---|
| 書式 | int Dio275_Start_PortWatch (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 | 《現在値入力モード》で入力変化&メッセージ発信動作、または 《ストローブ入力モード》でストローブ入力 (ラッチ・更新) 検出&メッセージ発信動作の開始。 |

【8】監視・検出&メッセージ発信動作のストップ

| | |
|-------|---|
| 書式 | int Dio275_Stop_PortWatch (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 | 《現在値入力モード》で入力変化検出&メッセージ発信動作、または 《ストローブ入力モード》でストローブ入力 (ラッチ・更新) 検出&メッセージ発信動作の停止。 |

【9】ステータス取得

| | |
|-------|--|
| 書式 | int Dio275_Get_Status (int board_no, int &status) |
| 引数 | board_no : 対象ボード番号 (0 ~ 7) status : 本ボードのステータス生データ (3 - 8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 | 当時点でのストロブ (割り込み) 検出フラグ、および本ボードのステータス生データを得る。 (当ハンドラでは割り込みを使用していない。) |

【10】本ハンドラの終了

| | |
|-------|---|
| 書式 | int Dio275_Close_DioSys (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。本ボードのリセットを行う。 但し、汎用デジタル出力はクリアされず、現状を維持する。 |

【11】本ハンドラのバージョン取得

| | |
|-------|--|
| 書式 | Int Dio275_Get_Libver (int ver) |
| 引数 | ver : バージョン情報 / 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) |
| 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 - 4 B . エラーコード一覧

| 戻り値 | 不具合の内容、または因果情報 | 適用関数、引数、等 |
|------|--------------------------|-------------------------------|
| - 1 | ボードを検出できない。 | Dio275_Open_DioSys |
| - 2 | IDが違う。 | Dio275_Open_DioSys |
| - 3 | ドライバファイルを検出できない。 | Dio275_Open_DioSys |
| - 4 | ドライバファイルのバージョンが違う。 | Dio275_Open_DioSys |
| - 5 | ハンドラ初期化の未実行。 | |
| - 6 | | |
| - 7 | | |
| - 8 | | |
| - 9 | | |
| - 10 | 使用ボード枚数 パラメータの不適當 | num_board |
| - 11 | ボード番号 パラメータの不適當 | board_no |
| - 12 | 入力モード パラメータの不適當 | inp0_mode、 inp1_mode |
| - 13 | | |
| - 14 | | |
| - 15 | 現在値入力モードではない。 | In0x_PortWatch、In1x_PortWatch |
| | | |
| - 20 | 実行中。 | |

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) デジタル接点入力 | 本ボードの入力は電流駆動ですから適切な電源を使用して、接点のON/OFF時に適切な電流がON/OFFされるよう接続されていなければなりません。(2-1項) |
| (4) デジタル接点出力 | 本ボードの出力は無電圧接点出力です。実体はFETですから、ON/OFFは抵抗値の変化となります。電流を流すときは定格(100mA)を超えないようにしてください。 (2-2項) |

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。

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。 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】 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. 付録 (WINDOWS 2000・XP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 /vr 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 | D I O - 2 7 5 P C I | | 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 B O X | | |
| | 本体メモリ | | | | | |
| | O S | D O S () W I N () | | | | |
| 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 | |