





2002年3月

第7.0版



| 2 |  | 1_ 1 | • |
|---|--|------|---|
|   | 第1単はUDIC   | 1-1  | • |
| ۲ |  | 1-7  |   |
| • |  | 1-2  | • |
| ٠ | = (1-3) / (-1) | 1-3  | • |
| ٠ | (1-3-1) コネッシェンテジョン   | 1-4  |   |
| ۲ | (1-3-2) //ゴノロロロはな  | 1-5  | • |
| • | (1-5-0) アンスノビ係<br>第2音 Windows05/08/Ma 解説   | 2-1  |   |
| • | 第2章 Windowsso, so, mo 解説<br>(2-1) インストレーション  | 2-1  | • |
|   | (2-1) FC カード設定内容の確認  | 2-6  |   |
|   | (2-2) アンインストール   | 2-8  | • |
|   | (2-0) アンドンストール<br>(2-4) C 言語 API ライプラ川 解説  | 2-9  | • |
|   | (2 4) 0 日間 / 1 / 7 / 7 / 7 / 7 / 7 / 7 / 7 / 7 / 7   | 2-9  |   |
| • | (2-4-2) Visual C サンプルプログラム   | 2-17 |   |
| • | (2-5) BASIC 言語 API ライプラリ解説   | 2-22 |   |
| • | (2-0) Dicici Calif, パーンテンデーが<br>(2-5-1) Visual BASIC によるアプリケーション開発   | 2-22 |   |
| • | (2-5-2) Visual BASIC サンプルプログラム   | 2-28 |   |
| • | 第3章 Windows2000/XP 解説  | 3-1  |   |
|   | (3-1) インストレーション  | 3- 1 |   |
| • | (3-2) PC カード設定内容の確認  | 3-5  | • |
| ٠ | (3-3) アンインストール   | 3-6  | • |
| ٠ | (3-4) C 言語 API ライプラリ解説   | 3-8  | • |
| ٠ | (3-4-1) Visual C によるアプリケーション開発   | 3-8  | • |
| • | (3-4-2) Visual C サンプルプログラム   | 3-13 | • |
| • | (3-5) BASIC 言語 API ライプラリ解説   | 3-19 | • |
|   | (3-5-1) Visual BASIC によるアプリケーション開発   | 3-19 | • |
|   | (3-5-2) カスタムコントロール   | 3-24 | • |
|   | (3-5-3) Visual BASIC サンプルプログラム   | 3-28 | • |
|   | 第4章 MS-DOS/Windows3.1 解説   | 4- 1 | • |
| • | (4-1) MS-DOS/Windows3.1 でのインストレーション  | 4- 1 |   |
| • | (4-1-1) イネープラのインストール   | 4- 1 |   |
|   | (4-1-2) DOS/V 版カードサービス対応イネーブラを使用する場合   | 4- 2 |   |
| • | (4-1-3) PC-9800 シリーズ版カードサービス対応イネーブラを使用する場合   | 4-5  |   |
|   | (4-1-4) DOS/V 版ポイントイネーブラを使用する場合  | 4- 7 |   |
| • | (4-2) MS-DOS ライブラリ解説   | 4-9  | • |
| • | (4-2-1) MS-DOS ライプラリ   | 4-9  | • |
|   | (4-2-2) MS-DOS サンプルプログラム   | 4-10 | • |
| • | (4-3) Windows3.1 16Bit ライプラリ解説   | 4-20 | • |
| • | (4-3-1) Windows3.1 アプリケーション  | 4-20 | • |
| • | (4-3-2) 16Bit DLL 関数仕様   | 4-23 | • |
|   | 第5章 WindowsNT 解説   | 5- 1 | • |
|   | (5-1) インストレーション  | 5-1  | • |
| • | (5-2) Visual C によるアプリケーション開発   | 5-3  | • |
| • | (5-2-1) DLL 関数仕様   | 5-4  | • |
| • | (5-2-2) Visual C サンプルプログラム   | 5-8  | • |
|   | (5-3) Visual BASIC によるアプリケーション開発   | 5-11 | • |
|   | (5-3-1) Visual BASIC サンフルフログラム   | 5-12 |   |
|   | 第6章 オフションアイソレーションユニット  | 6-1  |   |
|   | (6-1) REX-IPI16 製品微要   | 6-1  |   |
|   | (6-2) REX-IPO16 製品機要   | 6-2  |   |
|   | (6-3) RCL-TRM48 製品概要   | 6-3  |   |

第1章 はじめに

# (1-1) 製品概要

REX-5055 は DOS/V・PC-9800 シリーズ対応の TTL レベル 16 ビットパラレル入 出力 PC カードです。PC Card Standard (Release 2.1 クラスの 16Bit PC Card) に準拠した仕様になっており、PC/AT 互換機及び NEC PC-9800 シリーズで使用す ることができます。本製品の主なハードウェア仕様及びソフトウェア仕様を下 表に示します。

### *□□ ハードウェア仕様*

| 項目      | 内容                             |  |  |  |
|---------|--------------------------------|--|--|--|
| 入出力ビット数 | 16 ビット(8 ビット単位で入出力方向設定)        |  |  |  |
|         | 割り込み入力:入力ポートの1ポートを割り当て可        |  |  |  |
| 入出力仕様   | TTL レベル(74HCT245 相当)           |  |  |  |
|         |                                |  |  |  |
|         | 最大入出力電流 入力:±20mA 出力:±35mA      |  |  |  |
|         | 標準入出力電圧 0V/5V                  |  |  |  |
|         | (注)上記定格は1ポート当たりの仕様になります。       |  |  |  |
| 入出力コネクタ | セントロニクス 24 ピンコネクタ付き            |  |  |  |
|         | ケーブル長∶約 50 cm                  |  |  |  |
|         | (参考)                           |  |  |  |
|         | PC カード側コネクタ:NX-25P ヒロセ電機製相当品使用 |  |  |  |

### *□ ソフトウェア仕様*

| 項 目                   | 内 容                                |
|-----------------------|------------------------------------|
| カードサービス版イネーブラ         | DOS/V 版・PC-9800 シリーズ版              |
| ポイントイネーブラ             | DOS/V版                             |
| MS-DOS 対応ソフト          | Microsoft C 対応ライブラリ                |
|                       | サンプルソフト                            |
| Windows3.1 対応ソフト      | Visual C/Visual BASIC 対応 16Bit DLL |
|                       | Visual C/Visual BASIC サンプルソフト      |
| Windows95/98/Me 対応ソフト | セットアップ用カード情報ファイル                   |
|                       | Visual C/Visual BASIC 対応 32Bit DLL |
|                       | Visual C/Visual BASIC サンプルソフト      |
| WindowsNT 対応ソフト       | セットアッププログラム                        |
|                       | Visual C/Visual BASIC 対応 32Bit DLL |
|                       | Visual C/Visual BASIC サンプルソフト      |
| Windows2000 /XP 対応ソフト | セットアップ用カード情報ファイル                   |
|                       | Visual C/Visual BASIC 対応 32Bit DLL |
|                       | Visual C/Visual BASIC サンプルソフト      |

# (1-2) 添付品

製品には PC カードと下記添付品が添付されています。ご使用前にご確認願います。 図セントロニクス 24 ピンコネクタ付きケーブル

図セントロニクス 24 ピンオス側コネクタ

☑セットアップ・ライブラリディスク 1.44MB 3.5" (3 枚)

☑ユーザーズマニュアル

図ご愛用者登録はがき·保証書

### 

ご愛用者カードは保証書を切り離した後、必要事項を記入の上、必ずご返送く ださい。ご返送頂けない場合、弊社からのバージョンアップ等のサポートサービ スは受けられなくなりますのでご注意ください。



NX-25P

(1-3) ハードウェア仕様

(1-3-1) コネクタピンアサイン



|          | 信               | 弓 名             |
|----------|-----------------|-----------------|
| コネクタピン番号 | カードバックコネクタ側     | セントロニクスコネクタ側    |
| 1        | GND             | V <sub>cc</sub> |
| 2        | PIO 0           | NC              |
| 3        | PIO 1           | PIO 15          |
| 4        | PIO 2           | PIO 13          |
| 5        | PIO 3           | PIO 11          |
| 6        | GND             | PIO 9           |
| 7        | GND             | V <sub>cc</sub> |
| 8        | PIO 4           | NC              |
| 9        | PIO 5           | PIO 7           |
| 10       | PIO 6           | PIO 5           |
| 11       | PIO 7           | PIO 3           |
| 12       | GND             | PIO 1           |
| 13       | GND             | GND             |
| 14       | PIO 8           | NC              |
| 15       | PIO 9           | PIO 14          |
| 16       | PIO 10          | PIO 12          |
| 17       | PIO 11          | PIO 10          |
| 18       | GND             | PIO 8           |
| 19       | GND             | GND             |
| 20       | PIO 12          | NC              |
| 21       | PIO 13          | PIO 6           |
| 22       | PIO 14          | PIO 4           |
| 23       | PIO 15          | PIO 2           |
| 24       | GND             | PIO 0           |
| 25       | V <sub>cc</sub> |                 |

## (1-3-2) 入出力回路仕樣

PIO0-PIO15 入力回路(TTL レベル負論理)



PIO0-PIO15 出力回路(TTL レベル負論理)



## (1-3-3) レジスタ仕様

REX-5055 DIO PC カードは、ベースアドレスから連続した4バイトのレジスタがマッ ピングされます。コントロールレジスタ及び割込マスクレジスタは、必ずバイトアクセス してください。データレジスタは、バイトアクセスまたはワードアクセスのどちらでも可 能です。

| OFFSET | REX-5055 レジスタ仕様 |                  |           |                      |       |     |            |     |    |
|--------|-----------------|------------------|-----------|----------------------|-------|-----|------------|-----|----|
| Base+0 | コントローノ          | ルレジスタ[WR         | ITE]      |                      |       |     |            |     |    |
|        | B7              | B6 B5            | B4        | B3                   | B2    |     | B1         | В   | 0  |
|        | IMD             | IR3 IR2          | IR1       | IR0                  | -     | C   | DIRH       | DI  | RL |
|        |                 |                  |           |                      |       |     |            |     |    |
|        | ビット             |                  | ۲.        | J                    | 容     |     |            |     |    |
|        | DIRL            | PIO0-PIO7        | のデータ入け    | 出力方向                 | を設定   | しま  | <b>ታ</b> . |     |    |
|        |                 | 0:入力 1:出         | 力         |                      |       |     |            |     |    |
|        | DIRH            | PIO8-PIO15       | のデータ入     | 出力方向                 | 句を設え  | 定しま | ₹す。        |     |    |
|        |                 | 0:入力 1:出         | 力         |                      |       |     |            |     |    |
|        | IR0-IR3         | 割り込みソー           | ・スビットを    | 指定します                | す。    |     |            |     |    |
|        |                 | IR0-IR3 で指       | ;定された F   | NO ポー                | トにM   | Dで  | 指定す        | された | :信 |
|        |                 | 号の立ち下れ           | がり・立ち上    | がりがあ                 | った場   | 合、  | 割込         | ステ- | ータ |
|        |                 | スレジスタの           | IntFlag ビ | ットがセッ                | トされま  | ます。 |            |     |    |
|        |                 |                  | 室山に入っい    | リックに日                |       | 100 |            |     | 1  |
|        |                 |                  | 割り込み      |                      | , IR3 |     |            |     | -  |
|        |                 |                  | Pl        | 01                   | 0     | 0   | 0          | 1   | -  |
|        |                 |                  | Pl        | 0 2                  | 0     | 0   | 1          | 0   |    |
|        |                 |                  | Pl        | 03                   | 0     | 0   | 1          | 1   |    |
|        |                 |                  | Pl        | 04                   | 0     | 1   | 0          | 0   |    |
|        |                 |                  | Pl        | O 5                  | 0     | 1   | 0          | 1   |    |
|        |                 |                  | Pl        | O 6                  | 0     | 1   | 1          | 0   |    |
|        |                 |                  | Pl        | 07                   | 0     | 1   | 1          | 1   |    |
|        |                 |                  | Pl        | 08                   | 1     | 0   | 0          | 0   |    |
|        |                 |                  | Pl        | 09                   | 1     | 0   | 0          | 1   |    |
|        |                 |                  | PIC       | 0 10                 | 1     | 0   | 1          | 0   |    |
|        |                 |                  | PIC       | <u>) 11</u>          | 1     | 0   | 1          | 1   |    |
|        |                 |                  | PIC       | ) 12                 | 1     | 1   | 0          | 0   | _  |
|        |                 |                  | PIC       | ) 13                 | 1     | 1   | 0          | 1   | -  |
|        |                 |                  |           | ) 15                 | 1     | 1   | 1          | 1   | -  |
|        |                 |                  |           | 715                  |       | I   |            |     | 1  |
|        | IMD             | <b>実</b> liλトリガー | モードを記     | 定します                 |       |     |            |     |    |
|        |                 |                  |           | んこしみ 9<br>- 11 T ぃミジ | •     |     |            |     |    |
|        |                 |                  |           |                      |       |     |            |     |    |
|        |                 |                  |           |                      |       |     |            |     |    |

| OFFSET     | REX-5055 レジスタ仕様        |                                       |                                       |  |                            |                        |                |              |
|------------|------------------------|---------------------------------------|---------------------------------------|--|----------------------------|------------------------|----------------|--------------|
| Base+1     | 刮り込みw                  | (スクレ                                  | シンスタ[Wi                               | RIF  |                            |                        |                |              |
|            | B7                     | <b>B</b> 6                            | B5                                    | B4   | B3                         | B2                     | B1             | B0           |
|            | lrEbl                  | -                                     | -                                     | -  | ICLRW3                     | ICLRW2                 | ICLRR3         | ICLRR2       |
|            | ビット                    |                                       |                                       |  | 内                          | 容                      |                |              |
|            | B0:ICLRR2<br>B1:ICLRR3 |                                       | 割込ステ・<br>設定します<br>1 データ               | ータスレシ<br>ナ。<br>レジスタ(   | ジスタの I<br>の下位ハ             | IntFlag ビ<br>、イトをリ・    | ットクリア<br>ードした時 | 7条件を<br>持クリア |
|            |                        |                                       | 割込ステ-<br>設定します<br>1 データ               | ータスレシ<br>ナ。<br>レジスタ(   | ジスタの I<br>の上位ハ             | IntFlag ビ<br>、イトをリ・    | ットクリア<br>ードした時 | 7条件を<br>持クリア |
|            | B2:ICLF                | RW2                                   | 割込ステ-<br>設定します<br>1 データ               | ータスレシ<br>す。<br>レジスタ(   | ジスタの I<br>の下位ハ             | IntFlag ビ<br>ベイトにラ・    | ットクリア<br>イトした眠 | ∇条件を<br>持クリア |
|            | B3:ICLRW3              |                                       | 割込ステ-<br>設定します<br>1 データ               | ータスレシ<br>ナ。<br>レジスタ(   | ジスタの I<br>の上位ハ             | IntFlag ビ<br>ベイトをラ・    | ットクリア<br>イトした時 | ア条件を<br>テクリア |
| B7 : IrEbl |                        | 割込モー<br>0 ディセ<br>1 コントロ<br>ポート<br>生しま | ドを設定(<br>ーブル()<br>コールレジ<br>に入力だ<br>す。 | します。<br>リセット時<br>ジスタの I<br>があった印   | うのデフォ<br>R0-IR3 て<br>寺、CPU | ・ルト値)<br>で指定され<br>に割込要 | れた PIO<br>要求を発 |              |
| Base+1     | 割り込みス                  | くテーダ                                  | タスレジスタ                                | [READ]   | ]                          |                        |                |              |
|            | B7                     | B6                                    | B5                                    | B4   | B3                         | B2                     | B1             | B0           |
|            | IntFlag                | -                                     | -                                     | -  | ICLRW3                     | ICLRW2                 | ICLRR3         | ICLRR2       |
|            |                        |                                       |                                       |  | _                          | -                      |                |              |
|            |                        |                                       | = /     +                             | はおっつ   | <b>N</b>                   | <b>谷</b>               | * * * + +      |              |
|            | B3-B0                  |                                       | 71 FU/2'                              | 直かその   | コンコー                       | トハックる                  | されより。          |              |
|            | B7                     |                                       | IntFlag:割<br>0 割込要<br>1 割込要           | 込要求の<br> <br> | D有無を<br>)ません。<br>生している     | 示します。<br>ます。           | )              |              |
|            |                        |                                       |                                       |  |                            |                        |                |              |

| OFFSET |                     |          | REX   | K-5055 l  | <b>ノジスタ</b> (   | L様   |  |  |
|--------|---------------------|----------|---|---|---|--|--|--|
| Base+2 | データレシ               | バスタ下     | 位バイト[   | READ/W  | RITE]   |  |  |  |
|        | B7                  | B6       | B5  | B4  | B3  | B2   | B1   | B0   |
|        | PIO 7               | PIO 6    | PIO 5   | PIO 4   | PIO 3   | PIO 2  | PIO 1  | PIO 0  |
|        | ピッ                  | <b>۲</b> |   |   | 内   | 容  |  |  |
|        | B7-B0               |          | PIO7-PIO  | 0:外部と   | のデータ  | の入出プ   | りを行い   | ます。  |
|        |                     |          | コントロ<br>場合 !  | ールレジ<br>七カモー  | スタの D<br>ドになり   | IRL ビット<br>≢す  | を1にセ   | ットした   |
|        |                     |          | 0 ⇔外部   | ゴッう<br>怒ラインに  | t High (5   | 、  | イブされ   | ます。  |
|        |                     |          | 1 ✑外音   | 部ラインは   | Low(0   | V)にドラ·   | イブされる  | ます。  |
|        |                     |          | コントロ  | ールレジ  | スタの   | DIRL ビッ  | トを0に   | セットし   |
|        |                     |          | た場合   | 、入力モ  | ードにな  | ります。   |  |  |
|        |                     |          | 0 \$\$ 外音   | 部ラインた   | Nら High   | (5V)が入<br>(のいが)  | 、力されて  | こいます   |
|        |                     |          | 1 -> ୬ト គ   | おフイ ンズ  | いら LOW  | עע( vu)<br>אימ   | JIAIC  | いより  |
| Base+3 | データレシ               | バスタ上     | 位バイト[   | READ/W  | RITE]   |  |  |  |
|        | B7                  | B6       | B5  | B4  | B3  | B2   | B1   | B0   |
|        | PIO 15              | PIO 14   | PIO 13  | PIO 12  | PIO 11  | PIO 10   | PIO 9  | PIO8   |
|        |                     |          |   |   |   |  |  |  |
|        | ビッ                  | <u>۲</u> |   |   | 内   | 容  |  |  |
|        | <b>ビッ</b><br>B15-B8 | <b>۲</b> | PIO15-PI  | <br>28∶外部   | <b>内</b><br>とのデー  | <b>容</b><br>タの入出   | 力を行い   | 1ます。   |
|        | ピッ<br>B15-B8        | ۲<br>۱   | PIO15-PI(<br>コントロ   | O8∶外部<br>ールレジ   | <b>内</b><br>とのデー<br>スタの D   | <b>容</b><br>タの入出<br>IRH ビット  | 力を行い<br>、を1にセ                                  | 1ます。<br>ミットした                                |
|        | <b>ビッ</b><br>B15-B8 | <u>۲</u> | PIO15-PIO<br>コントロ<br>場合、1   | O8∶外部<br>ールレジ<br>出力モー                                     | 内<br>とのデー<br>スタのD<br>ドになり:  | <b>容</b><br>タの入出<br>IRH ビット<br>ます。                                   | 力を行し<br>`を1にセ                                  | \ます。<br>∶ットした                                |
|        | <b>ビッ</b><br>B15-B8 | <u>۲</u> | PIO15-PI(<br>コントロ<br>場合、 <br>0 ≎外音                                  | O8∶外部<br>ールレジ<br>出力モー<br>₿ラインに                            | <b>内</b><br>とのデー<br>スタのD<br>ドになり:<br>は High (5                                  | <b>容</b><br>タの入出<br>IRH ビット<br>ます。<br>V) にドラ                         | 力を行し<br>を1にセ<br>イブされ:                          | ヽます。<br>ミットした<br>ます。                         |
|        | <b>ビッ</b><br>B15-B8 | F        | PIO15-PI(<br>コントロ<br>場合、<br>0 ⇔外語<br>1 ⇔外語                          | O8:外部<br>ールレジ<br>出力モー<br>ぷラインに<br>ぷラインに                   | 内<br>とのデー<br>スタの D<br>ドになり<br>ま High (5<br>ま Low (0                             | 容<br>タの入出<br>IRH ビット<br>ます。<br>V) にドラ<br>V) にドラ・                     | 力を行し<br>を1にセ<br>イブされ:<br>イブされ:                 | \ます。<br>☆ットした<br>ます。<br>ます。                  |
|        | <b>ビッ</b><br>B15-B8 | F        | PIO15-PI(<br>コントロ<br>場合、<br>1 ⇔外語<br>コントロ                           | 08:外部<br>ールレジ<br>出力モー<br>ポラインに<br>ポラインに<br>ールレジ           | 内<br>とのデー<br>スタの D<br>ドになり<br>は High (5<br>は Low (0<br>スタの [                    | 容<br>タの入出<br>IRH ビット<br>ます。<br>V) にドラ・<br>V) にドラ・<br>DIRH ビッ         | 力を行い<br>を1にセ<br>イブされ:<br>イブされ:<br>トを 0 に       | 1ます。<br>☆ットした<br>ます。<br>ます。<br>・セットし         |
|        | <b>ビッ</b><br>B15-B8 | F        | PIO15-PIC<br>コントロ<br>場合、日<br>0 ⇔外部<br>1 ⇔外部<br>コントロ<br>た場合          | O8:外部<br>ールレジ<br>出力モー<br>ポラインに<br>ポラインに<br>ポラインに<br>、入力に  | 内<br>とのデー<br>スタの D<br>ドになり<br>は High (5<br>は Low (0<br>スタの [<br>ードにな            | 容<br>タの入出<br>IRH ビット<br>ます。<br>V) にドラ・<br>V) にドラ・<br>DIRH ビッ<br>ります。 | 力を行し<br>を1にセ<br>イブされ<br>イブされ<br>トを 0 に<br>カさわっ | 1ます。<br>ミットした<br>ます。<br>ます。<br>セットし          |
|        | <b>ビッ</b><br>B15-B8 | F<br>3   | PIO15-PIC<br>コントロ<br>場合、<br>1 ⇔外語<br>コントロ<br>た場外語<br>0 ⇔外語<br>1 ⇔外語 | D8:外部<br>ールレジ<br>出力モー<br>ポラインは<br>ポラインは<br>、入力ンだ<br>ポラインズ | 内<br>とのデー<br>スタの D<br>ドになり<br>は High (5<br>は Low (0<br>スタの E<br>ードにな<br>いら High | 容<br>タの入出<br>IRH ビット<br>ます。<br>V) にドラ・<br>DIRH ビッ<br>ります。<br>(5V) が入 | 力を行し<br>を1にセ<br>イブされ<br>イブされ<br>トを 0 に<br>力されて | 1ます。<br>シットした<br>ます。<br>ます。<br>たマットし<br>こいます |

(空白ページ)



本章では Windows95、Windows98(SecondEdition を含む)および WindowsMe での REX-5055の使用方法について解説しております。

# (2-1) インストレーション

Windows95 OSR-2<sup>(注1)</sup>のリリースにより現在 Windows95 のバージョンには、Windows95 OSR-2と OSR-2以前のバージョンがあります。「マイコンピュータ」を右クリックし「プロパティ」情報を表示することによりどちらのバージョンがインストールされているか調べることができます。システム情報が「Microsoft Windows95 4.00.950 a」の場合はOSR-2以前のバージョンになり、OSR-2の場合は「Microsoft Windows95 4.00.950 B」となります。ご利用の Windows95 が OSR-2 かそれ以前のバージョンかによりインストールの方法が異なりますので注意してください。

(注1) OSR-2(OEM Service Release 2)では FAT32、CardBus 等の新しい機能がサポ ートされています。

m Windows95 OSR-2 でのインストール方法

【1】PC カードの挿入

PC カードをスロットに 挿入すると、ハードウェ アウィザードが起動し右 のデバイスドライバーウ ィザードのインストール が表示されます。ここで は、次へを選択します。



### 【2】ドライバーファイル場所の指定

次にドライバーファイル(INF ファイル)の場所を指定します。PC/AT にイン ストールされる場合は、

| A: ¥PCATINF  | 2440の参照   | <b>¥</b> 8 |
|--|---|------------|
| PC-9800 シリーズにインストールされる   | このデルイス用のドライル、情報が含まれているフォルケを運捩、<br>ください。   | π          |
| C:¥PC98INF   | = = 7'x7h-y7'   | -          |
| を設定し次に進んでください。   | = = 35(// FD (A)  |            |
| (使用するトライムがあるフォルダ名を入力してくどさい。(参照)を約ックするとフォルダの/<br>ます。<br>場所し<br>(AvPcaint) | 本部ができ<br>○ 0 σ 2 (0)<br>○ 0 |            |
|  | OK \$v/th   |            |
| OK 1   | Fe/dal  |            |

【3】インストールの完了

インストールが正常に完 了した場合は、「このデバ イス用に更新されたドライ バーが見つかりました。」 のメッセージが表示されま すので確認してください。

以上でインストールは完 了です。

| NACE PERCENTION OF THE |  |
|------------------------|--|
|                        | このデバイス用の更新されたどうかが見つかりました。  |
|                        | REX5055 DIO PC CARD For PC/AT  |
|                        | このドライバを使用する場合は、[宮子]を押してくだみい。別<br>のドライバを検索したい場合は、[場所の悟定]を押してくださ<br>い。 |
| 33                     | Postinf  |
|                        | 構成の指定の   |
|                        |  |
|                        | (頁名(日) 頁了 \$+>世紀   |

🎟 Windows95 (OSR-2 以前のバージョン) でのインストール方法

【1】PC カードの挿入 PC カードをスロットに 挿入すると、右のハードウ ェアウィザードが起動しま す。ここでは「ハードウェ アの製造元が提供する「ド ライバ」選択し次に進みま す。

| 新しい・トウェア                        |                              | ? × |
|---------------------------------|------------------------------|-----|
| RATOC Sy                        | stem, IncREX5055 DIO PC Card |     |
| 新しいトウェア用につ                      | ストールするトライがを選択してください。         |     |
| C Windows (ERD))                |                              |     |
| ◎ 万一トウェアの製造元                    | 「が提供するドライバー」                 |     |
| <ul> <li>C 一覧から選び(s)</li> </ul> |                              |     |
| ○ ドライバをインストール                   | Ltat KD)                     |     |
|                                 | ок <del>1+/12</del> % чК     | 7°H |

### 【2】配布ファイルコピー元の指定

次にドライバーファイル (INF ファイル)の場所を指定します。PC/AT 互換機 にインストールされる場合は、



### m Windows98 でのインストール方法

【1】PC カードの挿入

PC カードをスロットに 挿入すると、ハードウェ アウィザードが起動し右 のデバイスドライバーウ ィザードのインストール 画面が表示されますの で、「次へ」ボタンを押 します。



ドライバの検索方法 を選択するため「使用中 のデバイスに最適なド ライバを検索する(推 奨)」をチェックし、「次 へ」ボタンを押します。



### 【2】配布ファイルコピー元の指定

製品添付FDをドライブ に挿入し、「検索場所の 指定」にチェックします。 PC/AT 互換機にインスト ールされる場合は、 □ A:¥PCATINF PC-9800 シリーズにイン ストールされる場合は、 □ C:¥PC98INF を設定し「次へ」ボタン を押します。



索され、右画面のように ドライバのある場所が表 示されますので「次へ」 ボタンを押します。



【3】インストールの完了

インストールが正常に 行われるとビープ音で完 了が通知され、右画面が 表示されますので「完了」 ボタンを押します。

以上でインストールは 完了です。



WindowsMe でのインストール方法

【1】PC カードの挿入

PC カードをスロットに挿入す ると、右の新しいハードウェア の追加ウィザードが表示されま すので、製品添付の Win95/98/Me 用フロッピーディスクを FD ドラ イブへ挿入してください。

次に、「適切なドライバを自 動的に検索する (推奨)(<u>A</u>)」 を選択し「次へ」ボタンを押し ます。



【2】ドライバーファイル場所の指定

右のようにセットアップ情報 ファイル (inf ファイル)が、フ ロッピーディスク上から自動的 に検索されますので、 「REX-5055 DIO PC CARD For PC/ATを選択し、「OK」ボタンを押 します。

| PSYTODRA<br>INC. 2 IN MEMORY AND PROVIDE<br>EXERT OD PC CAPO For PC-98 | APPORTNENPOCARDATINE<br>AMPORTNENPOCARDATINE<br>AMPORTNENPOCARDISIDE | 1-8-1007<br>1-8-1007 |
|--|--|----------------------|
|  |  |                      |
|  |  |                      |

右の画面が表示されましたら、「完了」ボタンを押します。



以上で、REX-5055 インストールは終了です。

(2-2) PC カード設定内容の確認

### システムプロパティの起動

コントロールパネルのシステムを起動し、デバイスマネージャを選択します。 カードの設定が正常に行われていれば、コンピュータのレジストリツリー 「Otherdevices」の下に「REX5055 DIO PC CARD For PC/AT(または PC98)」が登 録されます。

プロパティのリソースタグを選択して1/0ポートアドレスおよびIRQの割り当 てで競合していないことを確認してください。

| 294057711/1797  | 2 ×                                   |
|---|---------------------------------------|
| 情報目 デバイスマネージや ハートウェア環   | 度 ハウォーマノス                             |
| ④ 種類別に表示(T) ()  | (抽読時に表示(の)                            |
| <ul> <li>Dt's→9</li> <li>OD-ROM</li> <li>Otherdevices</li> <li>PEDSIO DIO EA BOA</li> <li>PCMCIA )794</li> <li>PCMCIA )794</li> <li>PCMCIA )794</li> <li>PCMCIA )794</li> <li>PCMCIA )794</li> <li>PCMCIA )794</li> <li>PCMCIA )797</li> <li>PCMCIA )</li></ul> | R R R R R R R R R R R R R R R R R R R |
|   |                                       |
|   | ×                                     |
|   | OK キャンセル                              |

### リソースの変更

リソースの変更は、自動設定を選択しないようにして、設定の登録名を別の 設定に変更します。手動設定を行うと、「変更不可の環境設定の作成」ダイア ログが表示されますが、続行「はい」を選択してください。

手動設定したリソースが他のデバイスと競合していなければ、「ピッポッ」 というビープ音とともにシステムプロパティ画面に戻ります。もう一度、レジ ストリ「REX5055 DIO PC CARD For PC/AT」をダブルクリックし、手動設定した リソースが他のデバイスと競合していないことを確認してください。

| 変更不可の環境             | 定の作成  |   |   | 88                      |        |
|---------------------|---|---|---|-------------------------|--------|
| 1 15-20             | 生産がいくつか手動で調   | 「整きれています。                                     |   |                         |        |
| · 統行する<br>変更する      | と、これらの該宅は予約<br>ことはできなくなります。                           | 短れ、この先ヘードウェアをインストー                            | *762802101  | 自動的に                    |        |
| たとたは<br>による目<br>自動的 | 、フラゲ アット・フレイ機器を<br>動的なセットアップができた<br>2] チェックホリックスを遵べける | インストールする場合、設定を変更<br>になる可能性が張ります。その<br>元に限ります。 | できないために V<br>D 場合は、[V)-ス]                         | Andows<br>りつてE          |        |
| 続行しま                | Tto ?   | REX5055 DIO                                   | PC CARD For                                       | PC/AT070/Fr             | ? ×    |
|                     | 134   | () 情報 トライパ                                    | 、 リソース  |                         |        |
|                     |   | サワースの 語言<br>リワースの 語言<br>リワースの 種<br>IRG        | Disoss Dio i<br>空(B)<br>(別 銀気<br>- サンレス 030<br>07 | C CARD For PO/AT        |        |
|                     |   | 設定の重制<br>動定の                                  | 1名(四) 基本<br>(変更( <u>0</u> )                       | :設定 0008<br>  「自動設定(()) | -      |
|                     |   | 戦合すらた   | <u>^4</u> 2:                                      |                         | ۲<br>۲ |
|                     |   |   |   | ОК                      | ++>25  |

## (2-3) アンインストール

カードが正しくインストールされなかった場合は以下の手順でカード情報の 削除と INF ファイルの削除を行い、再度、(2-1)の方法でインストールを行っ てください。

【1】カード情報の削除

PC カードを挿入した状態 で、コントロールパネルからシ ステムを起動します。

「デバイスマネージャ」のタブ を選択し、Otherdevicesの下に ある「REX-5055 DIO PC CARD For PC/AT(または PC-98)」を選択 して「削除」ボタンを押します。 デバイス削除の確認が表示され ますので、「OK」を押してく ださい。



### 【2】INF ファイルの削除

「エクスプローラ」を起動し、¥Windows¥Inf¥Other フォルダにある「RATOC System, Inc.PCCARDAT.INF」ファイルを削除してください。

| 19270-3 - Other                  | The second second second second | INPROVE COMPANY   |    |
|----------------------------------|---------------------------------|-------------------|----|
| 274(14日) 編集句 表示创                 | (1466) 表例に入り(各)                 | 5-14 AA769        | 17 |
| 義・言・皇                            |                                 | 10 HIRT HAR 70/0+ | I. |
| THERE (II) CHARLEOWEREN          | OTHER                           |                   |    |
| 7615                             | RATOO Systematic                | POCARDAT          |    |
| E Envortes                       | -                               |                   |    |
| Hill Con Help<br>El (30) History | 4                               |                   |    |
| Catalos                          | _                               |                   |    |
| Cthar                            |                                 |                   |    |
| instalier<br>B instalier         |                                 |                   |    |
| 🗄 🫄 Mecia                        | -                               |                   |    |
| t Ginefrishisäte                 | heren                           | P 2/ 74%          |    |

### (2-4) C 言語 API ライブラリ解説

### (2-4-1) Visual C によるアプリケーション開発

Visual C 4.0 以上を使って REX-5055 DIO PC カードを制御するアプリケーションを開発する場合は、本製品に添付されている 32Bit 版 DLL の関数をコールする必要があります。

Visual C で作成したアプリケーションプログラムから 32Bit 版 DLL を呼び出 すためには、

- 1. アプリケーションプログラムに DIOLIB32.H ファイルをインクルードする。
- 2. アプリケーションプログラムのプロジェクトファイルに DIOLIB32.LIB を追加する。

 アプリケーションの実行ディレクトリまたは WINDOWS¥SYSTEM に、 DIOLIB32.DLL ファイルと VR5055D.VXD ドライバーをコピーする。 必要があります。



# DLL 関数仕様

|   | GetMyC | ardResource  |  | リソース情報の取得  |
|---|--------|--|--|--|
| 書 | 式      | BOOL GetMyCardResc<br>SlotNo, LPWORD IOE   | <b>burce</b> ( HWND <b>hWnd</b> , LPSTI<br>Base, LPWORD <b>IrqNo</b> )   | R <b>MyCardName</b> , WORD <b>NameLen</b> , WORD   |
| 機 | 能      | 指定スロットに挿入<br>ます。一致した場合<br>情報を返します。   | 、されているカードが指定<br>は、割り当てられている  | 定のカード名と一致するか否か調べ<br>ぅ I /0 ベースアドレス・IRQ リソース  |
| 引 | 数      | HWNDhWndLPSTRMyCardNameWORDNameLenWORDSlotNoLPWORDIOBaseLPWORDIrqNo  | <ul> <li>&gt; ウィンドウハンドル</li> <li>&gt; カード識別名を示す</li> <li>&gt; 文字列バッファの長</li> <li>&gt; カード挿入スロット</li> <li>&gt; (出力) I/0 リソース</li> <li>&gt; (出力) IRQ リソース</li> </ul> | ,<br>文字列へのポインタ<br>さ<br>番号<br>、情報を格納する変数アドレス<br>、情報を格納する変数アドレス  |
| 戻 | 値      | 0 : 正常終了<br>-1 : ドライバ呼びと<br>-2 : カードサービス<br>-3 : GET_CARD_SERV<br>-4 : GET_FIRST_TUP<br>-5 : GET_TUPLE_DAT<br>-6 : GET_CONFIG_IN<br>-7 : メモリーアログ<br>-8 : スロットに力-<br>-9 : スロットに挿入                                     | 出しエラー<br>スドライバーバージョン<br>/ICES_INFO ファンクショ<br>PLE ファンクションコー<br>FA ファンクションコー川<br>FO ファンクションコー<br>ケーションエラー<br>ードが挿入されていない<br>入されているカードは自                              | エラー<br>ョンコールサービスエラー<br>ルサービスエラー<br>ルサービスエラー<br>ルサービスエラー  |
| Ψ | ンプル    | <pre>#define MyCardName WORD SlotNo WORD MylOBase; WORD MylrqNo; BOOL Function( void {     /*スロット0または、     for ( SlotNo = 0;         Status = GetMyC         &amp;MylOBase, &amp;Mylr         if ( Status == 0</pre> | "REX5055 DIO PC Card"<br>)<br>スロット1に挿入されている<br>SlotNo < 2; SlotNo++ ){<br>CardResource( hDlg, MyCar<br>rqNo );<br>) ){<br>得正常終了時の処理 */                             | <pre>// DIOLIB32.Hで定義されています // サーチするスロット番号 // 取得したベースアドレス // 取得した割り込み番号 p自分のカードのリソース情報を取得 */ rdName, sizeof(MyCardName), SlotNo,</pre> |

### ShowCardUtil

# リソース表示ユーティリティ起動

- 書式 void ShowCardUtil(HWND hWnd)
- 機能 カードリソース情報表示ユーティリティを起動します。
- 引数 HWND hWnd ▶ ウィンドウハンドル
- 戻値 なし

```
画面
```

| п |  |  |
|---|--|--|
| _ |  |  |
| _ |  |  |
|   |  |  |

| All Rights Reserved\$ | csパージョン <u>210</u>     |
|-----------------------|------------------------|
| 79升数 👂                |                        |
| 79710                 | ሃታታ 1                  |
| 产加名                   | 火力名                    |
| RATOC System.Inc.     | RATOC System, Inc.     |
| 製品名                   | 製品名                    |
| REX5055 DIO PC Card   | REX5056 RS2320 PC Card |
| IRQ///-7 10           | IROUY-7 255            |
|                       |                        |

### StartHWIntPostMessage ユーザ定義メッセージ割り込みの開始

- 書式 BOOL StartHWIntPostMessage(HWND hWnd, WORD MyIOBase, WORD MyIrqNo, BYTE DirDataRegLo, BYTE DirDataRegHi, BYTE IrqPin, BYTE IrqMode, BYTE IntCIrMode, WORD StopCount)
- 機能 ユーザ定義メッセージ版の割り込み処理を開始します。割り込みが発生すると 割り込みハンドラからアプリケーションにメッセージを通知します。 wParamの 上位パイトに PI07-PI00の読み込み値が、下位バイトに PI015-PI08の読み込み値 がセットされています。また、IParam には割り込み発生の累計カウント数がセ ットされています。
- 引数 HWND hWnd ▶ ユーザアプリケーションのウィンドウハンドル WORD MylOBase ▶ カードに割り当てられている 1/0 ベースアドレス WORD MylrqNo ▶ カードに割り当てられている IRQ 番号 BYTE DirDataRegLo ▶ データレジスタ下位バイトの入出力方向 0:入力方向 1:出力方向 BYTE DirDataRegHi ▶ データレジスタ上位バイトの入出力方向 0:入力方向 1:出力方向 BYTE IrqPin ▶ 割り込みソース入力ポート番号 0:PI00 1:PI01 2:PI02 ··· 15:PI015 BYTE IrqMode ▶ 割り込みトリガーモード 0:立下りエッジ 1:立上りエッジ BYTE IntClrMode ▶ 割り込みクリア条件 0:下位バイトリード 1:上位バイトリード 2:下位バイトライト 3:上位バイトライト WORD StopCount > 割り込み終了回数
- 戻値 0:正常終了
  - -1:割り込み登録エラー
  - -2 : DeviceloControl ステータスエラー
- サンプル Visual C サンプルプログラム解説参照

| E | EndHWIr | ntPostMessage                             | ユーザ定義メッセージ割り込みの終了  |
|---|---------|---|--|
| 書 | 式       | BOOL EndHWIntPostMessage( void            | (  |
| 機 | 能       | ユーザ定義メッセージ版の割り込<br>による割り込み処理終了時は、必ず<br>い。 | み処理を終了します。 ユーザ定義メッセーミ<br>ずEndHWIntPostMessage()を呼び出してくだる |

- 引数 なし
- 戻値 常に0を返す
- サンプル Visual Cサンプルプログラム解説参照

| RegistA | syncProcCall  | 非同期プロシジャーコール割り込みの開始  |
|---------|---|--|
| 書式      | BOOL <b>RegistAsyncProcCal</b><br>WORD <b>MyIrqNo</b> , BYTE <b>Dir</b><br>IrqMode, BYTE IntCIrMoo  | II(HWND <b>hWnd</b> , PVOID <b>AsyncCallAdrs</b> , WORD <b>MyIOBase</b> ,<br>DataRegLo, BYTE DirDataRegHi, BYTE IrqPin, BYTE<br>de, WORD StopCount )   |
| 機 能     | 非同期プロシジャーコーノ<br>理を開始します。割り込ん<br>ジャーを呼び出します。   | ル(APC:Asynchronous Procedure Call)版の割り込み処<br>みが発生すると割り込みハンドラが登録されたプロシ  |
| 引 数     | HWND hWnd<br>PVOID AsyncCallAdrs<br>WORD MylOBase<br>WORD MylrqNo<br>BYTE DirDataRegLo<br>BYTE DirDataRegHi<br>BYTE IrqPin<br>BYTE IrqMode<br>BYTE IntCIrMode<br>WORD StopCount | <ul> <li>ユーザアプリケーションのウィンドハンドル</li> <li>呼び出すプロシージャのアドレス</li> <li>カードに割り当てられている I/0 ベースアドレス</li> <li>カードに割り当てられている IRQ 番号</li> <li>データレジスタ下位バイトの入出力方向</li> <li>0:入力方向 1:出力方向</li> <li>ジータレジスタ上位バイトの入出力方向</li> <li>0:入力方向 1:出力方向</li> <li>割り込みソース入力ポート番号</li> <li>0:PI00 1:PI01 2:PI02 ・・・ 15:PI015</li> <li>割り込みトリガーモード</li> <li>0:立下りエッジ 1:立上りエッジ</li> <li>割り込みクリア条件</li> <li>0:下位バイトリード 1:上位バイトリード</li> <li>2:下位バイトライト 3:上位バイトライト</li> </ul> |
| 戻 値     | 0 : 正常終了<br>-1 : 割り込み登録エラー  | ▶ 割り込み終了回致   |

-2 : DeviceloControl ステータスエラー

サンプル Visual Cサンプルプログラム解説参照

| Release | eAsyncProcCall   | 非同期プロシジャーコール割り込みの終了                              |
|---------|--|--|
| 書式      | BOOL ReleaseAsyncProcCal   | I( void )  |
| 機 能     | 非同期プロシジャーコール<br>RegistAsyncProcCall()に、<br>ReleaseAsyncProcCall()を | 版の割り込み処理を終了します。<br>よる割り込み処理終了時は、必ず<br>呼び出してください。 |
| 引 数     | なし   |  |
| 戻 値     | 常に 0 を返す   |  |
| サンプル    | Visual C サンプルプログラ  | ム解説参照  |

StartHWIntMyVxD

### 高速割り込み処理の開始

- 書式 BOOL StartHWIntMyVxD(HWND hWnd,WORD MyIOBase,WORD MyIrqNo,BYTE InputDataReg,BYTE OutputDataReg,BYTE IrqPin,BYTE IrqMode,BYTE IntCIrMode,WORD StopCount)
- 機能割り込みハンドラ内部で全ての処理を行う高速版の割り込み処理を開始します。割り込みハンドラ内部では、入力方向に設定されたデータレジスタの値を リードし、出力方向に設定されたデータレジスタにリードした値を出力します。 指定回数の割り込み処理が終了すると、呼び出し側プログラムに終了メッセー ジをポストします。IParamには割り込み発生の累計カウント数がセットされて います。
- 引数 HWND hWnd ▶ ユーザアプリケーションのウィンドウハンドル WORD MyloBase ▶ カードに割り当てられている 1/0 ベースアドレス ▶ カードに割り当てられている IRQ 番号 WORD MylrqNo BYTE InputDataReg ▶ 入力データレジスタ 0:下位バイト(PI07-PI00) 1:上位バイト(PI015-PI08) BYTE OutputDataReg ▶ 出力データレジスタ 0:下位(PI07-PI00) 1:上位(PI015-PI08) BYTE IrgPin ▶ 割り込みソース入力ポート番号 0:PI00 1:PI01 2:PI02 ··· 15:PI015 BYTE IrqMode ▶ 割り込みトリガーモード 0:立下りエッジ 1:立上りエッジ BYTE IntClrMode ▶ 割り込みクリア条件 0:下位バイトリード 1:上位バイトリード 2:下位バイトライト 3:上位バイトライト WORD StopCount ▶ 割り込み終了回数
- **戻 値** 0:正常終了
  - -1 : 割り込み登録エラー
  - -2: DeviceIoControl ステータスエラー

### EndHWIntMyVxD

### 高速割り込み処理の終了

- 書式 BOOL EndHWIntMyVxD(void)
- 機能 高速版の割り込み処理を終了します。 StartHWIntMyVxD ()による割り込み処理
   終了時は、必ず EndHWIntMyVxD ()を呼び出してください。
- 引数 なし
- 戻値 常に0を返す

OutPort

1バイトをポート出力

- 書式 WORD OutPort(WORD IOAddr, WORD OutVal)
- 機能 1バイトをポートに出力
- 引数 WORD IOAddr > ポート番号 WORD OutVal > バイト出力値(上位バイトは無視)
- 戻値 バイト出力値をそのまま返し、エラー値はなし

wOutPort

1ワードをポート出力

- 書式 WORD wOutPort(WORD IOAddr, WORD OutVal)
- 機能 1 ワードをポートに出力
- 引数 WORD IOAddr > ポート番号 WORD OutVal > ワード出力値
- 戻値 ワード出力値をそのまま返し、エラー値はなし

### 1バイトをポート入力

- 書式 WORD InPort(WORD IOAddr)
- 機能 ポートから1バイト読み込む
- 引数 WORD IOAddr >ポート番号
- 戻値 ポートから読み込んだバイトデータを返します(上位バイトは無視して ください)

#### wInPort

InPort

1ワードをポート入力

- 書式 WORD wInPort(WORD IOAddr)
- 機能 ポートから1ワード読み込む
- 引数 WORD IOAddr > ポート番号
- 戻値 ポートから読み込んだワードデータを返します

# GetDIIVersion

DLL バージョンダイアログの表示

- 書式 void GetDIIVersion(HWND hWnd)
- 機能 DIOライブラリのバージョン表示ダイアログを呼び出す
- 引数 HWND hWnd > ウィンドウのハンドル
- 戻値 なし

(2-4-2) Visual C サンプルプログラム

アイソレーションユニット PHO-8 または PHI-8 を使った、サンプルプログラム が添付しています。

PHO-8 は右図に示す LED 点灯回路と アイソレーションを行って外部へ出 力する回路を 8 ブロック実装してお り、CN1-4~CN1-11 に本カードの PIOO から PIO15 を接続することによりオン オフ制御のシミュレーションを行う ことができます。PHI-8 は PHO-8 とは 反対に外部オンオフ信号をアイソレ ーションして入力するためのボード です。



サンプルプログラムは、6 つのモジュールで構成されています。 PHO-8 LED オンオフ制御プログラムと PHI-8 入力監視プログラムは、割り込みを使わない で単純に入出力を行うサンプルプログラムです。割り込みを使ったサンプルプロ グラムとして、 ユーザ定義メッセージによる割り込み制御と、 非同期プロシ ジャーコールを使った割り込み制御、及び 割り込みハンドラ内部で全ての入 出力制御を行う高速割り込み制御プログラムがあります。その他に カードリ ソース表示ユーティリティがあります。 , は下図に示す画面になります。次 ページ以降では、 , のサンプルプログラムについて解説を行います。

| PHO-8 LEDオンオフ制御フロウ<br>出力ボート選択 PHO15 | '74<br>-PIO8 ▼                           | PC Card リソー<br>レク ベースアド<br>IRD番号 | ×<br>2債報<br>レス 120 h         |                      |
|-------------------------------------|--|----------------------------------|------------------------------|----------------------|
| LED 7 LED 6                         | LED 5 LED 4 LI                           | ED 3 LED 2                       | LED 1 LED 0                  |                      |
| 全点灯                                 | PHI-8 入力監視がログラム<br>又イッチ入力株出間隔<br>入力ボート選択 | 5000 mase (1)<br>PI015-PI08      | PC Card)<br>1/0 ペース<br>IRQ番号 | マントス情報<br>アドルス 120 h |
|                                     | - 入力¥=9-<br>                             | SW 5 SW 4                        | SW3 SW2<br>件止                | SW1 SW0              |

ユーザ定義メッセージによる割り込みプログラム

割り込みソース入力ポートを指定して StartHWIntPostMessage()を実行する ことにより、割り込み発生に同期したユーザ定義メッセージ WM\_VXDEVENT が割 り込みハンドラから送られてきます。 このとき、追加情報 wParam の上位バイ トには PI07-PI00 の読み込み値が、下位バイトには PI015-PI08 の読み込み値が セットされています。また、IParam には割り込み発生の累計カウント数がセッ トされています。

```
#include "diolib32.h"
* 機能 ユーザ定義メッセージによる通知を使った割り込み入出力制御
*/
BOOL CALLBACK DIgProcPostMessage( HWND hDIg,
                                       // ウィンドウハンドル
                         UINT message,
                                      // メッセージのタイプ
                         WPARAM wParam, // 追加情報
                         LPARAM IParam // 追加情報
                        )
{
 switch( message )
 {
     /* 割り込みハンドラからユーザ定義メッセージ受信 */
     case WM_VXDEVENT:
        /*
         * wParam -> 上位バイト:PI07-PI00 下位バイト:PI015-PI08
         */
        // PI07-PI00の値 = (BYTE)wParam;
        // PI015-PI08の値 = (BYTE)(wParam>>8);
        /*
         * IParam -> 割り込みカウント数
        */
        // 割り込みカウント数 = (WORD) IParam;
        /* 割り込み終了回数に達した時の処理 */
        if(割り込み終了回数に達した時)
           EndHWIntPostMessage();
        return TRUE:
     case WM_COMMAND:
        switch( wParam ){
        case IDOK:
           StartHWIntPostMessage(hDIg, MyIOBase, MyIrqNo, DataRegLoDir, DataRegHiDir,
           IntPIONo, TrgMode, IntClrMode, MaxCount );
           return TRUE;
        case IDCANCEL:
           EndHWIntPostMessage();
           EndDialog(hDlg, TRUE); // ダイアログボックスの終了
           return TRUE;
        }
        break ;
     }
   return FALSE ;
}
```

### 非同期プロシジャーコールを使った割り込み制御

割り込みハンドラから呼び出すプロシジャーのアドレスと割り込みソース入 カポートを指定して非同期プロシジャーコール(APC:Asynchronous Procedure Call)版の割り込み処理 RegistAsyncProcCall()を実行します。この後、 SleepEx()を使って呼び出し側のプログラムを呼び出し可能な待ち状態に設定 します。割り込みが発生し割り込みハンドラがプロシジャーを呼び出しを行う と、呼び出し側のプログラムは呼び出し可能な待ち状態から復帰します。 割り込みハンドラから呼び出されるプロシジャーは、ASYNCPROC 型構造体へのポ インターが渡されます。構造体のメンバーCallbackProcとStopCountは RegistAsyncProcCall()で設定した値ですので、無視します。 ReadValの上位バ イトが PI07 から PI00 を読み出し値で、下位バイトが PI015 から PI08 を読み出 した値になります。 IntCount は、現在までの割り込み発生累積カウント値にな ります。

```
#include"diolib32.h"
       ExitSleepFlag = 0; // 非同期プロシージャコール SleepEx() 脱出用フラグ
BYTE
BOOL CALLBACK DIgProcAsyncProcCall(HWND hDig, UINT message, WPARAM wParam, LPARAM IParam)
{
   switch(message)
   {
   case WM COMMAND:
       switch( wParam )
       {
       case IDOK:
         /* コールバックルーチン登録等のパラメータ設定と実行 */
         RegistAsyncProcCall(hDlg, ApcCallback, MyIOBase, MyIrqNo,
         DataRegLoDir, DataRegHiDir, IntPIONo, TrgMode, IntClrMode, MaxCount);
         ExitSleepFlag = 0;
         while(ExitSleepFlag == 0)
           SleepEx( (DWORD)(-1), TRUE ); // 呼び出し可能な状態にする
         ReleaseAsyncProcCall();
         return TRUE;
       case IDCANCEL:
         EndDialog( hDlg,TRUE );
         return TRUE;
       }
       break ;
   }
   return FALSE;
}
/* 次ページに続く */
```

/\* \* 書 式 DWORD WINAPI ApcCallback( PVOID pApcPrm ) \*機能 非同期コールプロシージャ \* 引 数 PVOID pApcPrm : ASYNCPROC 型構造体へのポインター \* typedef struct AsyncProcCall\_t \* { \* DWORD CallbackProc; \* WORD StopCount; \* WORD ReadVal; 上位バイト (PI07-0)下位バイト(PI015-8) リード値 \* WORD IntCount; 割り込み回数積算値 \* } ASYNCPROC, \*pASYNCPROC; \* 戻 値 常に 0 を返す \*/ DWORD WINAPI ApcCallback( PVOID pApcPrm ) { ApcCount; // VxD から渡される現在までの割り込み回数 WORD ApcRead; // VxD から渡されるポート 0-15 のリードデータ WORD ApcCount = ((pASYNCPROC)pApcPrm)->IntCount; ApcRead = ((pASYNCPROC)pApcPrm)->ReadVal; /\* \* 呼び出された時の処理を記述します \*/ ExitSleepFlag = 1; // この例では1回の呼び出しで終了します return 0; }



| PHI-8 割り込み入力(非同期) | ロシージャコールバージョン | )                               |           | ×    |
|-------------------|---------------|---------------------------------|-----------|------|
| 割儿込み終了回数          | 10 回で終了       | PC Card 9                       | ソース情報     |      |
| デーリンジスタ下位方向設定     | 入力            | ■ 1/0 ペーズ)<br>■ 1RQ番号           | 7147 J120 | h    |
| データレジスタ上位方向設定     | λ <u>η</u>    | -                               | -         | _    |
| 割り込みソースピット        | PIOD          | -                               |           |      |
| 割り込みりがーモード        | 立下りエッジ        | •                               |           |      |
| 割引込み例7条件          | 下位パイリード       | • 8                             | 則込みフログレ   | 7    |
| モニター表示ホート         | PI07-PI00     | <ul> <li>AsynoProcCa</li> </ul> | - 受信回数    |      |
| - 入力モニター          |               |                                 |           |      |
| SW 7 SW 6         | SW 5 SW 4     | SW 3 SW 2                       | SW 1      | SW 0 |
|                   |               |                                 | 実行        | 終了   |

高速割り込み制御プログラム

割り込みハンドラ内で全ての入出力制御を行うことにより高速割り込み処理 を行うことができます。但し、割り込みハンドラ内で全ての入出力制御を行う 場合は、ユーザ毎の仕様に基づいて仮想デバイスドライバ内の割り込みハンド ラをカスタマイズする必要があります。弊社では、有償によるカスタマイズ作 業のご相談も承っております。

サンプルプログラムの割り込みハンドラは、入力方向に設定されたデータレジスタの値をリードし、出力方向に設定されたデータレジスタにリードした値を出力します。指定回数の割り込み処理が終了すると、呼び出し側プログラムにユーザ定義メッセージをポストし、追加情報 IParam には割り込み発生の累計カウント数がセットされています。

入出力データレジスタと割り込み終了回数を指定して、StartHWIntMyVxD()を 実行することにより、割り込みハンドラ内部で全ての処理を行う高速版の割り 込み処理を開始します。

| <u> </u>      | <br>_ | _   |
|---------------|-------|-----|
| - <del></del> | <br>  |     |
|               |       | 101 |
| ~             |       | шu  |
| _             |       |     |

| 高速 V×D 内部割り込みぬ | 処理パージョン   | ×                 |
|----------------|-----------|-------------------|
| 割り込み終了回数       | 10 回で終了   | - PC Card リゾース情報  |
| データ入力ホート選択     | PI07-PI00 | I/O ベースアドレス 120 h |
| データ出力ポート選択     | PI07-PI00 | IRQ # 5 ja        |
| 割り込みソースビット     | PIOO      |                   |
| 割り込みりがーモード     | 立下りエッジ 🔽  | ······            |
| 割り込み切ア条件       |           |                   |

## (2-5) BASIC 言語 API ライブラリ解説

### (2-5-1) Visual BASIC によるアプリケーション開発

Visual BASIC 4.0 を使って REX-5055 DIO PC カードを制御するアプリケーションを開発する場合は、本製品に添付されている 32Bit 版 DLL の関数をコール する必要があります。また、割り込み制御を行う場合は OLE カスタムコントロ ール(OCX)を使用します。

Visual BASIC で作成したアプリケーションプログラムから 32Bit 版 DLL を呼び出すためには、

1. モジュールファイルで DLL 関数の参照宣言を行う。

2. アプリケーションの実行ディレクトリまたは WINDOWS¥SYSTEM に、 DIOLIB32.DLL ファイルと VR5055D.VXD ドライバーをコピーする。

### 必要があります。

また、割り込み制御を行う場合は割り込みハンドラから送られてくるユーザ 定義メッセージを Visual BASIC 側のアプリケーションで受け取るために、本製 品に添付されている OLE カスタムコントロール MBOX を使用します。MBOX の使用 方法については、サンプルプログラム解説を参照してください。



# DLL 関数仕様

|   | GetMyCardResourceリソース情報の取得 |   |  |  |  |
|---|----------------------------|---|--|--|--|
| 書 | 式                          | Declare Function <b>Ge</b><br>ByVal <b>MyCardName</b> A<br>Integer, <b>IOBase</b> As  | <b>tMyCardResource</b> Lib"diolib32.d<br>s String, ByVal <b>NameLen</b> As In<br>Integer, <b>IrqNo</b> As Integer) As                              | II" (ByVal <b>hWnd</b> As Long,<br>teger, ByVal <b>SlotNo</b> As<br>Long |  |
| 機 | 能                          | 指定スロットに挿入<br>ます。一致した場合I<br>情報を返します。   | されているカードが指定のカード<br>は、割り当てられている I/0 ベーン   | 名と一致するか否か調べ<br>スアドレス・IRQ リソース  |  |
| 引 | 数                          | hWnd<br>MyCardName<br>NameLen<br>SlotNo<br>IOBase<br>IrqNo  | <ul> <li>ウィンドウハンドル</li> <li>カード識別名を示す文字列への</li> <li>文字列バッファの長さ</li> <li>カード挿入スロット番号</li> <li>1/0 リソース情報を格納する姿</li> <li>IRQ リソース情報を格納する姿</li> </ul> | Dメモリアドレス<br>S数アドレス<br>S数アドレス   |  |
| 戻 | 値                          | 0 : 正常終了<br>-1 : ドライバ呼び出<br>-2 : カードサービス<br>-3 : GET_CARD_SERV<br>-4 : GET_FIRST_TUPI<br>-5 : GET_TUPLE_DAT/<br>-6 : GET_CONFIG_INI<br>-7 : メモリーアロク<br>-8 : スロットにカー<br>-9 : スロットに挿入 | {しエラー<br>、ドライバーバージョンエラー<br>ICES_INFO ファンクションコール<br>E ファンクションコールサービス<br>A ファンクションコールサービス<br>Tーションエラー<br>・ドが挿入されていない<br>、されているカードは自分のカード             | サービスエラー<br>(エラー<br>エラー<br>(エラー   |  |

| ShowCardUtil |   |                                       | IJ                           | ソース表示ユ                     | ーティリティ <b>起動</b>         |
|--------------|---|---------------------------------------|------------------------------|----------------------------|--------------------------|
| 書            | 式 | Declare Function <b>SH</b><br>As Long | <b>nowCardUtil</b> Lib "diol | ib32.dll" (By <sup>\</sup> | /al <b>hWnd</b> As Long) |
| 機            | 能 | カードリソース情報                             | 表示ユーティリティを起                  | 己動します。                     |                          |
| 引            | 数 | hWnd                                  | ▶ ウィンドウハンドル                  | ,                          |                          |
| 戻            | 値 | なし                                    |                              |                            |                          |

| Start | HWIntPostMessag   | e ユーザ定義メッセージ割り込みの開始  |
|-------|---|--|
| 書式    | Declare Function<br>Long, ByVal <b>My<br/>DirDataRegLo</b> As<br>Integer, ByVal<br><b>StopCount</b> As In | n <b>StartHWIntPostMessage</b> Lib "diolib32.dll" (ByVal <b>hWnd</b> As<br>rIOBase As Integer, ByVal <b>MyIrqNo</b> As Integer, ByVal<br>Integer, ByVal <b>DirDataRegHi</b> As Integer, ByVal <b>IrqPin</b> As<br><b>IrqMode</b> As Integer, ByVal <b>IntCIrMode</b> As Integer, ByVal<br>teger) As Long |
| 機 能   | ユーザ定義メッt<br>割り込みハンドラ<br>上位バイトに PI07-<br>がセットされてい<br>ットされています  | zージ版の割り込み処理を開始します。割り込みが発生すると<br>ラからアプリケーションにメッセージを通知します。 wParam の<br>PI00 の読み込み値が、下位バイトに PI015-PI08 の読み込み値<br>\ます。また、 IParam には割り込み発生の累計カウント数がセ<br>「。  |
| 引 数   | hWnd<br>MylOBase<br>MylrqNo<br>DirDataRegLo<br>DirDataRegHi   | <ul> <li>ユーザアプリケーションのウィンドウハンドル</li> <li>カードに割り当てられている 1/0 ベースアドレス</li> <li>カードに割り当てられている IRQ 番号</li> <li>データレジスタ下位パイトの入出力方向</li> <li>0:入力方向 1:出力方向</li> <li>0:入力方向 1:出力方向</li> </ul>   |
|       | lrqPin<br>IrqMode   | <ul> <li>&gt; 割り込みソース入力ポート番号</li> <li>0:PI00 1:PI01 2:PI02 ・・・ 15:PI015</li> <li>&gt; 割り込みトリガーモード</li> <li>0:立下りエッジ 1:立上りエッジ</li> </ul>  |
|       | IntClrMode<br>StopCount   | <ul> <li>▶ 割り込みクリア条件</li> <li>0:下位バイトリード 1:上位バイトリード</li> <li>2:下位バイトライト 3:上位バイトライト</li> <li>▶ 割り込み終了回数</li> </ul>  |
| 戻 値   | 0 : 正常終了  |  |

-1 : 割り込み登録エラー

-2 : Device loControl ステータスエラー

## EndHWIntPostMessage ユーザ定義メッセージ割り込みの終了

- 書式 Declare Function EndHWIntPostMessage Lib "diolib32.dll" () As Long
- 機能 ユーザ定義メッセージ版の割り込み処理を終了します。 ユーザ定義メッセージ版による割り込み処理終了時は、必ず EndHWIntPostMessage()を呼び出して ください。
- 引数 なし
- 戻値 常に0を返す

| StartHWIntMyVxD           |   | 高速割り込み処理の開始  |
|---------------------------|---|--|
| 書式                        | Declare Function<br>ByVal <b>MyIOBase</b> A<br>As Integer, ByVa<br><b>IrqMode</b> As Inte<br>Integer) As Long | StartHWIntMyVxD Lib "diolib32.dll" (ByVal hWnd As Long,<br>s Integer, ByVal MyIrqNo As Integer, ByVal InputDataReg<br>OutputDataReg As Integer, ByVal IrqPin As Integer, ByVal<br>ger, ByVal IntCIrMode As Integer, ByVal StopCount As   |
| 機能                        | 割り込みハンドラ<br>す。割り込みハン<br>リードし、出力方<br>指定回数の割り込<br>ジをポストします<br>います。  | や内部で全ての処理を行う高速版の割り込み処理を開始しま<br>ドラ内部では、入力方向に設定されたデータレジスタの値を<br>向に設定されたデータレジスタにリードした値を出力します。<br>み処理が終了すると、呼び出し側プログラムに終了メッセー<br>。 IParam には割り込み発生の累計カウント数がセットされて  |
| 引 数                       | hWnd<br>MyIOBase<br>MyIrqNo<br>InputDataReg<br>OutputDataReg<br>IrqPin<br>IrqMode<br>IntCIrMode<br>StopCount  | <ul> <li>ユーザアプリケーションのウィンドウハンドル</li> <li>カードに割り当てられている 1/0 ベースアドレス</li> <li>カードに割り当てられている IRQ 番号</li> <li>入力データレジスタ <ul> <li>0:下位バイト(PI07-PI00) 1:上位バイト(PI015-PI08)</li> </ul> </li> <li>出力データレジスタ <ul> <li>0:下位(PI07-PI00) 1:上位(PI015-PI08)</li> </ul> </li> <li>割り込みソース入力ポート番号 <ul> <li>0:PI00 1:PI01 2:PI02 ・・・ 15:PI015</li> </ul> </li> <li>割り込みトリガーモード <ul> <li>0:立下りエッジ 1:立上りエッジ</li> </ul> </li> <li>割り込みクリア条件 <ul> <li>0:下位バイトリード 1:上位バイトリード</li> <li>2:下位バイトライト 3:上位バイトライト</li> </ul> </li> </ul> |
| 戻 値                       | 0 : 正常終了<br>-1 : VPICD 割り込<br>-2 : DIO コントロ<br>-3 : パラメータ言<br>-4 : ポストメッ1<br>-5 : 割り込みス?                      | ▲み登録エラー<br>1ールステータスエラー<br>&定エラー<br>zージ用メモリーアロケーションエラー<br>タートエラー  |
| EndHWIntMyVxD 高速割り込み処理の終了 |   |  |

- 書式 Declare Function EndHWIntMyVxD Lib "diolib32.dll" () As Long
- 機能 高速版の割り込み処理を終了します。 StartHWIntMyVxD()による割り込み処理
   終了時は、必ず EndHWIntMyVxD()を呼び出してください。
- 引数 なし
- 戻値 常に0を返す

|   | DutPort |   | 1パイトをポート出力   |
|---|---------|---|--|
| 書 | 式       | Declare Function <b>Ou</b><br><b>OutVal</b> As Integer) | t <b>Port</b> Lib "diolib32.dll" (ByVal <b>IOAddr</b> As Integer, ByVal<br>As Integer    |
| 機 | 能       | 1バイトをポートに   | 出力   |
| 引 | 数       | IOAddr<br>Val   | ➢ ポート番号<br>➢ バイト出力値(上位バイトは無視)  |
| 戻 | 値       | バイト出力値をその?  | まま返し、エラー値はなし   |
| V | vOutPor | 't  | 1ワードをポート出力   |
| 書 | 式       | Declare Function w<br>ByVal <b>OutVal</b> As In         | <b>DutPort</b> Lib "diolib32.dll" (ByVal <b>IOAddr</b> As Integer,<br>nteger) As Integer |
| 機 | 能       | 1ワードをポートに   | 出力   |
| 引 | 数       | IOAddr<br>Val   | <ul> <li>&gt; ポート番号</li> <li>&gt; ワード出力値</li> </ul>                                      |
| 戻 | 値       | ワード出力値をその   | まま返し、エラー値はなし   |
|   | nPort   |   | 1パイトをポート入力   |
| 書 | 式       | Declare Function <b>I</b><br>Integer                    | n <b>Port</b> Lib "diolib32.dll" (ByVal <b>IOAddr</b> As Integer) As                     |
| 機 | 能       | ポートから 1 バイト   | 売み込む   |
| 引 | 数       | IOAddr  | ▶ ポート番号  |
| 戻 | 値       | ポートから読み込ん;<br>ください)                                     | ビバイトデータを返します(上位バイトは無視して  |
| V | vinPort |   | 1ワードをポート入力   |
| 書 | 式       | Declare Function <b>w</b><br>As Integer                 | I <b>nPort</b> Lib "diolib32.dll" (ByVal <b>IOAddr</b> As Integer)                       |
| 機 | 能       | ポートから 1 ワード   | 売み込む   |
| 引 | 数       | IOAddr  | ▶ ポート番号  |
| 戻 | 値       | ポートから読み込ん   | ビワードデータを返します   |
| GetDIIVersion | DLL バージョンダイアログの表示 |
|---------------|-------------------|
|---------------|-------------------|

- 書式 Declare Sub GetDIIVersion Lib "diolib32.dll" (ByVal hWnd As Long)
- 機能 DIOライブラリのバージョン表示ダイアログを呼び出す
- 引数 hWnd > ウィンドウのハンドル
- 戻値 なし

(2-5-2) Visual BASIC サンプルプログラム

| 割り込みを使わない<br>で単純に入出力を行<br>うプログラム例 | PHO-8 LED オンオフ制御プログラム<br>PHI-8 入力プログラム                    |
|-----------------------------------|---|
| 割り込みを使ったプ<br>ログラム例                | ユーザ定義メッセージによる割り込み制御<br>割り込みハンドラ内部で全ての入出力制御を行<br>う高速割り込み制御 |

動作環境等については、Visual C のサンプルプログラム解説を参照してくだ さい。 , 実行画面は下記のようになります。ここでは、 ユーザ定義メッセ ージによる割り込み制御について解説し、 , , については省略致します。

ユーザ定義メッセージによる割り込み制御サンプルプログラムの解説 下記画面は、VB4.0 でのデザイン完成時の画面です。割り込み発生に同期した ユーザ定義メッセージを VB で作成したプログラムで受け取るためには、本製品 添付の OLE カスタムコントロール(OCX)MBOX を使用します。以下、作成の手順に ついて説明します。

| HORDerror - Horoard Vasat Base ひざひ     アドルロ デルロ パンルロ パンルロ     オテロ 挿入の 実行(の) シールロ パン(ス) へんての     オテロ     オー      オー |  |
|--|--|
|  | 1 H25.1241 1 2965×6290   |
|  | HDaSarp 2  |
| ●          | Ledora.tras PornProLED     PostMag.fras PornProLED     PostMag.fras PornNersi     Sepres.tras PornNersi     Version.fras PornVersion     eff.Has1995.has REIS000.basters   |
|  | TEXTO 2017 - ForniPostNag s<br>ForniPostNag form s<br>Appearance 1 - 3D s<br>AutoRichary Palae   |
| SW7 SW6 SW3 SW2 SW1 SW0  | Darde Style 2 - 可定<br>Darde Style 2 - 可定<br>Darde Style 3 - 「文法トケー(有利日本)<br>CipControls True<br>ControlBox True<br>DrawNools 13 - Copy Pen<br>DrawNools 0 - 東線   |
|  | Fill         Fill           Fill         True           Fill         SH0000000           Fill         SH00000000           Fill         SH00000000           Fill         SH00000000           Fill         SH000000000           Fill         MS PCE/cyc27           PantTinunpsent         True           File         AH00000012A |
| 本製品に添付されている OLE カスタ  | ムコントロール"MBOX"<br>module」を追加  |

Step.1 本製品添付ソフトのコピー

OLE カスタムコントロール: MBOX.OCX と 32 ビット版 DLL: DIOLIB32.DLL 及び 仮想デバイスドライバー: VR5055D.VXD を添付のフロッピーディスクからコピ ーします。VB プログラム作成ディレクトリ名を"MyProg"とします。

>COPY "コピー元ドライブ名":¥Win95¥DI132¥Diolib32.dll "コピー先ドライブ名":¥Windows¥System >COPY "コピー元ドライブ名":¥Win95¥DI132¥Vr5055d.vxd "コピー先ドライブ名":¥Windows¥System >COPY "コピー元ドライブ名":¥Win95¥DI132¥Mbox.ocx "コピー先ドライブ名":¥MyProg

Step.2 0CX のレジストリー登録 (割り込みサービス使用時必須)

本製品添付の OCX "MBOX.OCX"を VB で使用するためには、VB の CD-ROM に添付 されているツール"REGSVR32.EXE"を使って OCX のレジストリ登録を行います。 "REGSVR32.EXE"は 32 ビットコンソールアプリケーションですので、Windows の DOS BOX から実行します。尚、"REGSVR32.EXE"は VB の CD-ROM の"TOOLS¥PSS"に 添付されています。

OCX をレジストリー登録するときは、下記構文で実行します。

>REGSVR32 "ドライブ名":¥MyProg¥Mbox.ocx

OCX をレジストリー登録から削除するときは、"/U"を付けて下記構文で実行します。

>REGSVR32 /U "ドライブ名":¥MyProg¥Mbox.ocx

| RegSvr32 | ×                                     | RegSvr32 | ×                                       |
|----------|---------------------------------------|----------|---|
| ٩        | mbox.ocx の DIIRegisterServer に成功しました。 | ٩        | mbox.ocx の DIIUnregisterServer に成功しました。 |
|          | OK                                    |          | Ŭ, OK                                   |
|          | 登録成功メッセージ                             |          | 登録削除成功メッセージ                             |

A abl

Step.3 DIOLIB32 関数の Declare 宣言

次に、VB プログラムの作成に入ります。VB デザインメニューから新規プロジェクトを作成し、「ファイル」の「挿入」から標準モジュールを追加します。 追加した標準モジュールファイルで DLL 関数の参照宣言を行います。宣言部分 は、サンプルプログラム"REX5055.BAS"の下記部分をコピーしてください。

Declare Function OutPort Lib "diolib32.dll" (ByVal IOAddr As Integer, ByVal OutVal As Integer) As Integer Declare Function wOutPort Lib "diolib32.dll" (ByVal IOAddr As Integer, ByVal OutVal As Integer) As Integer Declare Function InPort Lib "diolib32.dll" (ByVal IOAddr As Integer) As Integer Declare Function winPort Lib "diolib32.dll" (ByVal IOAddr As Integer) As Integer Declare Function StartHWIntPostMessage Lib "diolib32.dll" (ByVal hWnd As Long, ByVal MyIOBase As Integer, ByVal MyIrqNo As Integer, ByVal DirDataRegLo As Integer, ByVal DirDataRegHi As Integer, ByVal IrqPin As Integer, ByVal IrqMode As Integer, ByVal IntCIrMode As Integer, ByVal StopCount As Integer) As Long Declare Function EndHWIntPostMessage Lib "diolib32.dll" () As Long Declare Function StartHWIntMyVxD Lib "diolib32.dll" (ByVal hWnd As Long, ByVal MyIOBase As Integer, ByVal MyIrqNo As Integer, ByVal InputDataReg As Integer, ByVal OutputDataReg As Integer, ByVal IrqPin As Integer, ByVal IrqMode As Integer, ByVal IntCIrMode As Integer, ByVal StopCount As Integer) As Long Declare Function EndHWIntMyVxD Lib "diolib32.dll" () As Long Declare Function GetMyCardResource Lib "diolib32.dll" (ByVal hWnd As Long, ByVal MyCardName As String, ByVal NameLen As Integer, ByVal SlotNo As Integer, IOBase As Integer, IrqNo As Integer) As Long Declare Function ShowCardUtil Lib "diolib32.dll" (ByVal hWnd As Long) As Long Declare Sub GetDIIVersion Lib "diolib32.dll" (ByVal hWnd As Long)

Step.4 MBOX OLE Control Module の追加 (割り込みサービス使用時必須)

VB のカスタムコントロールに MBOX(OCX)を追加します。 VB4.0 の場合、VB デザインメニューの「ツール」の「カスタムコン トロール」を起動し、利用可能なコントロールから「MBOX OLE Control module」をチェックします。VB ツールバーに MBOX が追加されます。

➢ VB4.0の場合

| 1794 TMD=4.   | <u> 1</u> 1 | 친  |
|---|-------------|----|
| 利用可能なコントロール(A): OK  | Ø           |    |
| Adobe Table 25 Apex Data Bound Grid   | 8           | ~  |
| CoreIDRAW 60 Exchange Graphic CoreIDRAW 60 がう747  CoreIMEMO 60 Object   |             | 삗  |
| Crystal Report Control     Desaware Animated Button Control     Employment Mittain Control     Employment Mittain Control |             |    |
| USAtk OLE Control module<br>Lead Std OLE Custom Control module<br>に ad Std OLE Custom Control module<br>ロ スカルターを(の)       | <br>        | 24 |
| □ Marquee Control Library<br>図 MEOX OLE Control module<br>□ Media Architects Video Play Control                           | <u> </u>    |    |
| MBOX OLE Control module   | 駏           | Ø  |
| THE CHUARDEDCOT WINKOW DOLEVOLATINDOAT RELEASE MIDOXOOX   | 20<br>88    | 57 |
|   |             | 1  |

VB5.0/6.0 の場合、VB デザインメニューの「プロジェクト」から「コ ンポーネント」を開き、「コントロール」タブを選びます。利用可能 なコントロールから「MBOX OLE Control module」をチェックします。 VB ツールバーに MBOX が追加されます。



A 35

" 🗖

최지 집

~

æ 📴

-

0 🗆

Ю

0.3

臣口

20 t.

超圖

×

Step.5 フォームに MBOX(OCX)を貼り付ける (割り込みサービス使用時必須)

フォームを作成し、割り込みハンドラが割り込み起動元プログラムに送るユ ーザ定義メッセージを受け取るための MBOX(0CX)を貼り付けます。これにより、 割り込みが発生すると MBOX がサービスするプロシージャ

MBOX1\_OnMsgPost(ByVal wParam As Integer, ByVal IParam As Long) が呼び出されます。この中で、割り込み通知に同期した処理を記述します。

| ■ 2-サ定義かセージ同期語刊込みま     | 807125'94                            |                       |                           |          |
|------------------------|--------------------------------------|-----------------------|---------------------------|----------|
| 割り込み終了回数               | рс <i>ђ</i>                          | ートリソース情報              |                           |          |
| 入力术一片選択                | • V                                  | 0 ላ'-27ት'レ2 🚺 h       |                           |          |
| 出力末一ト選択                |                                      | xi 善号                 |                           |          |
| 割り込みソースピット             | ×                                    |                       |                           |          |
| 割り込み付がーモード             | ¥                                    |                       |                           |          |
| 割り込み切げ条件               | <b>I</b>                             |                       |                           |          |
| モニターホート選択              | ▼ 割削減                                | SAJEJUZ               |                           |          |
| 入力ポートモニター              |                                      |                       |                           |          |
|                        |                                      |                       | -                         |          |
|                        |                                      |                       |                           |          |
| SW 7 SW 6 SW 5         | SW 4 SW 3                            | SW2 SW1 SW0           |                           |          |
|                        |                                      | 1                     | 1                         |          |
| H9050x X17             | 停止                                   | 87 B                  |                           |          |
| <u> </u>               |                                      |                       |                           |          |
| - D. F                 |                                      |                       |                           | le v     |
| ATTEND IN              | C                                    | The state state       | a Harabara                |          |
| ₹J*9*±9№               | I BOX1                               | T 7.09~9*+            | . Oneserost               | <u> </u> |
| Private S<br>Dim ReadD | ub MBOX1_OnMsgPost<br>ata As Integer | (ByVal wParam As Inte | eer. ByVal IParam As Lone | °        |
| Dim IntNs              | gCount As Long                       |                       |                           |          |
| · 割년:<br>Latio         | 込みハンドラから渡さ<br>Count = LParce         | れた割り込み発生回数            |                           | - 11     |
| · 횖니)<br>ReadD         | 込みハンドラから渡さ<br>stall Person           | れたデータレジスタ読み           | 込み値                       | - 11     |
| End Sub                | 159 - m 9/88                         |                       |                           |          |
|                        |                                      |                       |                           | -        |
|                        |                                      |                       |                           | - 100    |

Step.6 カードに割り当てられているリソース情報取得

DIO カードを制御するためには、プログラム実行時に自分のカードに割り当 てられている I/O ベースアドレスと割り込み番号のリソース情報を取得する必 要があります。通常は、割り込みを実行するフォームがロードされた時にリソ ースを取得します。

| Global Const MyName = "REX5055 DIO PC Card"<br>Global MyIOBase As Integer<br>Global MyIrqNo As Integer | ' 製品情報タプルの製品名の定義<br>' カード I /0 ベースアドレス<br>' カード IRQ 番号 |
|--|--|
| Private Sub Form_Load()  |  |
| Dim SlotNo, Stat As Integer  |  |
| 'スロット0または1に挿入されている自分の  | )カードのリソース情報取得  |
| For SlotNo = 0 To 1  |  |
| Stat = GetMyCardResource(hWnd, MyName  | e, Len(MyName), SlotNo, MyIOBase, MyIrqNo)             |
| If Stat = 0 Then   |  |
| TextAdrs.Text = Hex(MyIOBase)  |  |
| TextInt.Text = Str(MyIrqNo)  |  |
| Exit For   |  |
| End If   |  |
| Next SlotNo  |  |
| End Sub  |  |
|  |  |

### Step.7 ユーザ定義メッセージによる割り込み処理の起動

フォームの OK ボタンが押されたらユーザ定義メッセージによる割り込み制 御 StartHWIntPostMessage()を実行します。 StartHWIntPostMessage()へは MBOX1.GetMboxWnd()により取得した OCX のウィンドウハンドルを引き渡してく ださい。尚、 StartHWIntPostMessage()を実行した際は、終了時には必ず EndHWIntPostMessage()を実行し割り込み登録の解除を行ってください。また、 VB 上の全てのバグをフィックスした後に記述してください。

```
Private Sub CommandOk_Click()
Global My10Base, MyIrqNo As Integer
Dim OleHandle As Long
Dim RegVal, dummy As Integer
Dim DataRegLoDir, DataRegHiDir, IntSrcBit, IntTrgMode, IntClrMode As Integer
'OLE のウィンドウハンドル取得
OleHandle = MBOX1.GetMboxWnd()
If (OleHandle = 0) Then
MsgBox "OLE のハンドルが取得できません。", vbOKOnly + vbCritical, "エラー"
Exit Sub
End If
'ユーザ定義メッセージを受け取る割り込み処理の起動
dummy = StartHWIntPostMessage(OleHandle, My10Base, My1rqNo, DataRegLoDir,DataRegHiDir,
IntSrcBit, IntTrgMode, IntClrMode, IsrStopCount)
End Sub
```

(空白ページ)



本章では Windows2000 および WindowsXP での REX-5055 の使用方法について解 説しております。

(3-1) インストレーション

m Windows2000 でのインストール方法

【1】PC カードの挿入

PC カードを挿入すると「ハードウェアウィザード」が起動し(右下画面)、インストー ルが開始されます。「RATOC\_System\_Inc. REX5055\_DIO\_PC\_Card」と表示されている かを確認し、以下の手順でインストールを行ってください。

| 新しんソート | ウェアが見つかりました                           |
|--------|---------------------------------------|
| *      | RATOC_System_Inc. REX5055_DIO_PC_Card |

「新しいハードウェアの検索 ウィザードの開始」で「次へ (<u>N</u>)>」ボタンを押します。



Page.3-2

「ハードウェアデバイスドライ バのインストール」では「デ バイスに最適なドライバを検 索する(推奨)(<u>S</u>)」にチェック を入れて「次へ(<u>N</u>)>」ボタン を押します。

| 特していードウェアの構成ウィザード  |                               |
|--|-------------------------------|
| n-P017 1147 F34/804321-6<br>5/47 F34/81 J1-F017 17428+44-74055<br>934017 1055674 | anteriatorialeto 💱            |
| はのサリビスモインストームします<br>・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・                         | Sampaga Balanggana<br>Sampaga |
| (8   | 4848) (28598) - #40458        |

「ドライバファイルの特定」で 「場所を指定(<u>S</u>)」にチェック を入れて「次へ(<u>N</u>)>」ボタン を押します。



## 【2】infファイル場所の指定

製品添付の FD をフロッピー ディスクドライブに挿入しま す。 製造元のファイルのコピ ー元(<u>C</u>)で inf ファイルの場所 を指定し、「OK」 ボタンを押 します。



「ドライバファイルの検索」で は、製品添付の FD より右画 面のように inf ファイルが検 索されますので「次へ(<u>N</u>)>」 ボタンを押します。



【3】インストールの完了





以上で、REX-5055 のインストールは終了です。

Page.3-4

## WindowsXP でのインストール方法

### 【1】 PC カードの挿入

PC カードをスロットに挿入す ると、右図の「新しいハードウェ アの検出ウィザード」が表示され ますので、製品添付のWin2000/XP 用フロッピーディスクを FD ドラ イブへ挿入してください。

次に、「ソフトウェアを自動的 にインストールする(推奨)(<u>1</u>)」 を選択し「次へ」ボタンを押しま す。

セットアップ情報ファイル (inf ファイル)が、フロッピ ーディスク上から検索され、自 動的にインストールが行われ ます。





右の画面が表示されました ら、「完了」ボタンを押します。

「完了」ボタンを押した後、
 右下の画面が出ますので、
 「はい(Y)」を押してパソコン
 を再起動してください。

以上で、REX-5055 インストー ルは終了です。



| 174 | 24058   |
|-----|---|
| 2)  | 時によびパイスかしストームが完了しました。モジイスキャポートモジンフトウェアにおうエージェージを再加加すると第(550)<br>ます。時によび言葉をかにするには、エンビューが再加加すると思くかけます。<br>キッジ・再加加しますが |
|     | 11/0 (xxkb)   |

# (3-2) PC カード設定内容の確認

コントロールパネルのシステムを起動します。「システムのプロパティ」のハードウェ アのタグから「デバイスマネージャ(D)」ボタンを押します。「OtherDevices」をクリックし て新しく REX5055.SYS for REX-5055 DIO PC CARD が追加されているのを確認して ください。

また、「プロパティ」でリソースが正しく割当てられているかを確認してください。デバ イスの競合が発生した場合は「自動設定(U)」のチェックを外し、競合が起こらない値 に設定を変更してください。

| 특가/i/スマネージャ 💷 🗙  |          |
|--|----------|
| 特作④ 表示②   ← →   □ 図 🖆 😫 🛃 🗸  |          |
| □       ■       COMPAQ         ■       DVD/CD-ROM F9-(7)         ■       □       DIDE ATA/ATAPI 12-1-0-5         ■       ○       Otherdevices         ■       ■       ■         ■       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ■         ●       ●       ●         ●       ●       ■         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ●       ●       ●         ● <td< td=""><td></td></td<> |          |
| ● ● ネットワーク パタフタ<br>単合 フロッピー ディスク コントローラ<br>REX5055 SYS for REX5055 DID PO CARD/01プロパラィ   | ?   X    |
| ◎ == 20% ディスクドライブ<br>田- 夏オート 00MとしびD   |          |
| ● う マウスとその頃かのホインティング デ/  | 1        |
| REX5055.SYS for REX5055 DID PC CARD  |          |
| リソースの設定(FD:  |          |
| リソースの種類 設定   |          |
| IRQ 07   |          |
| ( 1/○ \$600 - 0304   |          |
| 設定の登録名(臣): 現在の構成   | <u>×</u> |
|  | 主の東更心。   |
| 競合するデバイス:  |          |
|  | ×        |
| OK   | キャンセル    |

## (3-3) アンインストール

m Windows2000 でのアンインストール方法

インストールした内容を削除する方法について説明します。

削除は、

(1)デバイスの削除

(2)INF ファイルの削除の手順で行います。

【1】 デバイスの削除

PC カードを挿入した状態で、コントロールパネルのシステムを起動します。「システムのプロパティ」のハードウェアのタグから「デバイスマネージャ(D)」ボタンを押します。「Otherdevices」をクリックして REX-5055.SYS for DIO PC CARD を表示させクリックします。

メニューバーのー「操作(A)」-「削除(U)」をクリックするか、キーボード上の 「Delete」キーを押します。デバイスの削除の確認で「OK」ボタンを押し削除してください。

| 」操作④ 表示公 ↓ ← → ⑧ 回 ☎ 2 3 3 8 🗶  |  |
|---|--|
| COMPAG     DVD/CD-ROM ドライブ     DVD/CD-ROM ドライブ     DE ATA/ATAPI コントローラ     Otherdevices     Otherdevi |  |
|   |  |

【2】INF ファイルの削除

エクスプローラからフォルダ「C:¥WINNT¥inf」を開き、oemX.infファイル(X=数字)を検 索し、例えば <u>oem0.inf が1つだけの場合は、oem0.infと拡張子のみ異なる oem0.PNF</u> <u>を削除して〈ださい。</u>oemX.infが複数ある場合(oem0.inf, oem1.inf・・・)は、メモ帳など でそれぞれの inf ファイルを開いて、その内容の[Manufacturer] セクション が %REX5055, Manufacturer%=REX5055 となっているファイルと拡張子のみ異なる PNF ファイルを削除して〈ださい。

| 11 M   |                           |  |  |   | A. 18 M  |
|--|---------------------------|--|--|---|--|
| 2+(AU) 編集会 単元() お物に入り() 3-4  | PED VESRI                 |  |  |   | 12   |
| +ms + + - 1 Dire 3mm   | : Para Xa                 | 101  |  |   | -  |
| Larrydo 1 1 14   |                           |  |  | - <u></u>   | • (~1948   |
| Detail A server works  | +Py29466<br>994-001-10900 | a nota lant<br>nota land Phil<br>nota acc Phil<br>nota acc Phil<br>nota acc Phil<br>nota rung<br>nota ru | <ul> <li>Instructure (Instructure)</li> <li>Instructure (Instructure)</li> <li>Instructure (Instructure)</li> <li>Instructure (Instructure)</li> <li>Instructure)</li> <li>Instructure (Instructure)</li> <li>Instructure)</li> <li>Instructure)</li></ul> | Nimi<br>militati<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>internet<br>interne | A Transformer<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approximation<br>approx |
| [Version]<br>Bignature="Windows: NT\$"<br>Class=Otherdevices<br>Class=Qtherdevices<br>Class=Qtherdevices<br>Class=Qtherdevices   |                           | ±<br>F   | netscpq<br>PretscpqPt#<br>PretscpqPt#<br>Pretscon<br>Pretscon<br>Ptraces<br>Ptraces  | a) parti / o<br>a)  | al strok<br>al strok<br>al strok<br>al strok<br>al strok<br>al sources<br>al sources   |
| Provider=\$PE/5055_Provider%   |                           |  | a otar en  | 3 IDPAPMF   | (in the second s   |
| [Manufacturer]<br>\$PE05055.Manufacturer\$=RE6055  |                           |  | Porteon Parts  | Alereo  |  |
| L0 Iana Iront al 132.)<br>Addiag (PE/SIES, AddC1 anoffing  |                           |  |  | PF 44 1941  | 1-8  |
| (FED6055)<br>BFED5055.DeviceDescH-REX5055.INSTALL, FCBCLANRATOC_DISTEM_INCFED5055.   | 100_PC_040-01F            | 8  |  |   |  |
| <ul> <li>1000: Add an optional Tag File to ensure that the end user<br/>has specified the correct path to the diskette. (D-HOM.<br/>or mateorii drive. See the INF File Reference in the<br/>IDE documentation for acre information on SourceDisketween<br/>sections.</li> <li>[SourceDisketTema]</li> <li>1:3805055. (PriverDiskNameT.".1</li> </ul>  | この画i<br><u>oem3.ir</u>    | 面の場合<br><u>nf</u>  | に削除する  | るファイル   | lt,  |
| [SourceDistoFiles]<br>HEX095.cos1<br>DIULEX.eti1=1<br>Mod6055.cos1   | oem3.P                    | <u>NF</u>  | となり  | )ます。  |  |
| a manufacture and a second sec |                           | 2  |  |   |  |

以上の操作でアンインストール完了です。

カードスロットより、REX-5055を抜きパソコンを再起動してください。



## (3-4) C 言語 API ライブラリ解説

### (3-4-1) Visual C によるアプリケーション開発

Visual C 5.0 以上を使って REX-5055 DIO PC カードを制御するアプリケーションを開発する場合は、本製品添付フロッピーディスクの¥Win2000¥DI132 フォルダに格納されている 32Bit 版 DLL の関数をコールする必要があります。

Visual C で作成したアプリケーションプログラムから Windows2000/XP 用 32Bit 版 DLL(**DIOLIB2K.DLL**)を呼び出すためには、

- 1. アプリケーションプログラムに DIOLIB2K.H ファイルをインクルードする。
- 2. アプリケーションプログラムのプロジェクトファイルに DIOLIB2K.LIB を追加する。

必要があります。



# DLL 関数仕様

- 書式 BOOL GetMyCardResource(HWND hwnd, LPSTR MyCardName, WORD NameLen, WORD SlotNo, LPWORD IOBase, LPWORD IrqNo)
- 機能 指定スロットに挿入されているカードが指定のカード名と一致するか否か調べます。一致した場合は、割り当てられている I/O ベースアドレス・IRQ リソース 情報を返します。

| 引 数 | HWND<br>LPSTR<br>WORD<br>WORD<br>LPWORD<br>LPWORD | hwnd<br>MyCardName<br>NameLen<br>SlotNo<br>IOBase<br>IrqNo | <ul> <li>ウィンドウハンドル</li> <li>カード識別名を示す文字列へのポインタ</li> <li>文字列バッファの長さ         <ul> <li>(Windows95/98 互換用)(0を指定してください)</li> <li>(出力) I/0 リソース情報を格納する変数アドレス</li> <li>(出力) IRQ リソース情報を格納する変数アドレス</li> </ul> </li> </ul> |
|-----|---|--|---|
|-----|---|--|---|

戻値 正常終了時0を返し、リソースが正常に取得できなかった場合は-1を返します。

### StartHWIntPostMessage ユーザ定義メッセージ割り込みの開始

- 書式 BOOL StartHWIntPostMessage(HWND hWnd, WORD MyIOBase, WORD MyIrqNo, WORD DirDataRegLo, WORD DirDataRegHi, WORD IrqPin, WORD IrqMode, WORD IntCIrMode, WORD StopCount)
- 機能 ユーザ定義メッセージ版の割り込み処理を開始します。割り込みが発生すると 割り込みハンドラからアプリケーションにメッセージを通知します。 wParamの 上位バイトに PI07-PI00 の読み込み値が、下位バイトに PI015-PI08 の読み込み 値がセットされています。また、 IParam には割り込み発生の累計カウント数が セットされています。

| 引 | 数 | HWND | hWnd         | ト ユ        | ーザアプリケーションのウィンドウハンドル            |
|---|---|------|--------------|------------|---------------------------------|
|   |   | WORD | MyIOBase     | > カ        | ードに割り当てられている 1/0 ベースアドレス        |
|   |   | WORD | MylrqNo      | > カ        | ードに割り当てられている IRQ 番号             |
|   |   | WORD | DirDataRegLo | <b>≻</b> デ | ータレジスタ下位バイトの入出力方向               |
|   |   |      |              | 0:         | 入力方向 1:出力方向                     |
|   |   | WORD | DirDataRegHi | <b>≻</b> デ | ータレジスタ上位バイトの入出力方向               |
|   |   |      |              | 0:         | 入力方向 1:出力方向                     |
|   |   | WORD | IrqPin       | ≻ 割        | り込みソース入力ポート番号                   |
|   |   |      |              | 0:         | PIO0 1:PIO1 2:PIO2 ··· 15:PIO15 |
|   |   | WORD | l rqMode     | ≻ 割        | り込みトリガーモード                      |
|   |   |      |              | 0:         | 立下りエッジ 1:立上りエッジ                 |
|   |   | WORD | IntClrMode   | ▶ 割        | り込みクリア条件                        |
|   |   |      |              | 0:         | 下位バイトリード 1:上位バイトリード             |
|   |   |      |              | 2:         | 下位バイトライト 3:上位バイトライト             |
|   |   | WORD | StopCount    | ▶ 割        | り込み終了回数                         |
|   |   |      |              |            |                                 |

戻値 正常終了0を返します。I/0ベースアドレス設定エラー等は-1を返し、その他のマイナスエラーはDirDataRegLo・DirDataRegHi・IrqMode・IntCIrModeの設定エラーです。

## EndHWIntPostMessage ユーザ定義メッセージ割り込みの終了

- 書式 BOOL EndHWIntPostMessage(void)
- 機能 ユーザ定義メッセージ版の割り込み処理を終了します。 ユーザ定義メッセージ による割り込み処理終了時は、必ず EndHWIntPostMessage()を呼び出してくださ い。
- 引数 なし
- 戻値 常に0を返す

| 3 | StartHW | IntMy                        | /VxdBuf   |  | 高速割り込み処理の開始  |
|---|---------|------------------------------|---|--|--|
| 書 | 式       | BOOL<br>Inpu<br>char         | StartHWIntMyVxo<br>tDataReg, WORD I<br>*buf )   | d <b>Buf</b> ( HWND <b>hWnd</b> , WOF<br>rqPin, WORD IrqMode,  | RD <b>MylOBase</b> , WORD <b>MylrqNo</b> , WORD<br>WORD IntClrMode, WORD StopCount ,           |
| 機 | 能       | IrqP<br>ポー<br>に格約<br>び出      | in で指定した上位<br>ト(上位ポート)の<br>納します。指定し<br>し側に返します。 | 立ポート(下位ポート<br>Dデータレジスタを読る<br>た Count 数だけ割り   | )に割り込み信号が発生すると下位<br>み込み、指定したバッファ char*buf<br>込みが発生すると本関数は処理を呼                                  |
| 引 | 数       | HWND<br>WORD<br>WORD<br>WORD | hWnd<br>MylOBase<br>MylrqNo<br>InputDataReg     | <ul> <li>&gt; ユーザアプリケー</li> <li>&gt; カードに割り当て</li> <li>&gt; カードに割り当て</li> <li>&gt; 入力データレジス<br/>0:下位バイト(PIO</li> <li>&gt; 割1133 みソーフ入</li> </ul> | ションのウィンドウハンドル<br>られている I/O ベースアドレス<br>られている IRQ 番号<br>タ<br>7-PIOO)1:上位バイト(PIO15-PIO8<br>カポート番号 |
|   |         | WORD                         | l rqMode  | 0:PI00 1:PI01 2:<br>> 割り込みトリガー<br>0:立下りエッジ 1   | PIO2 ・・・ 15:PIO15<br>モード<br>· 立 トリエッジ  |
|   |         | WORD                         | IntClrMode                                      | <ul> <li>3.3.4</li> <li>&gt; 割り込みクリア条</li> <li>0:下位バイトリー</li> <li>2:下位バイトライ</li> </ul>   | ・ユー・ユー・<br>件<br>ド 1:上位バイトリード<br>ト 3:上位バイトライト   |
|   |         | WORD<br>char                 | StopCount<br>*buf                               | > 割り込み終了回数> データを格納する   | バッファアドレス   |

戻値 正常終了時0を返します。引数設定エラー等は-1、メモリアロケーションエラ ーの場合は-2を返します。

# EndHWIntMyVxdBuf 高速割り込み処理の終了

- 書式 BOOL EndHWIntMyVxBuf(void)
- 機能 高速版の割り込み処理を終了します。 StartHWIntMyVxdBuf()による割り込み処
   理終了時は、必ず EndHWIntMyVxdBuf()を呼び出してください。
- 引数 なし
- 戻値 常に0を返す

OutPort

1バイトをポート出力

- 書式 WORD OutPort(WORD IOAdrs, WORD bOutVal)
- 機能 1バイトをポートに出力
- 引数 WORD IOAdrs ▷ ポート番号 WORD bOutVal ▷ バイト出力値(上位バイトは無視)
- 戻値 バイト出力値をそのまま返し、ドライバ呼び出しエラーは-1を返します。

wOutPort

1ワードをポート出力

- 書式 WORD wOutPort(WORD IOAdrs, WORD wOutVal)
- 機能 1 ワードをポートに出力
- 引数 WORD IOAdrs > ポート番号 WORD wOutVal > ワード出力値
- 戻値 ワード出力値をそのまま返し、ドライバ呼び出しエラーは-1を返します。

### InPort

1バイトをポート入力

- 書式 WORD InPort(WORD IOAdrs)
- 機能 ポートから1バイト読み込む
- 引数 WORD IOAdrs > ポート番号
- 戻値 ポートから読み込んだバイトデータを返します。ドライバ呼び出しエラーは-1 を返します。(上位バイトは無視してください)

#### wInPort

1ワードをポート入力

- 書式 WORD wInPort(WORD IOAdrs)
- 機 能 ポートから1ワード読み込む
- 引数 WORD IOAdrs >ポート番号
- 戻値 ポートから読み込んだワードデータを返します。ドライバ呼び出しエラーは-1 を返します。

(3-4-2) Visual C サンプルプログラム

本製品には、以下3つのサンプルプログラムを添付しております。

基本デジタル入出力プログラム (割り込みを使わないで単純に入出力を行うサンプルプログラム)

外部イベント監視プログラム (割り込みを使ったサンプルプログラム)

データ受信プログラム (高速割り込みを使ったサンプルプログラム)

次頁より、各サンプルプログラムについて解説いたします。 (詳細については各サンプルプログラムのソースをご参照ください。)

Windows95/98/Me で作成したアプリケーションを Windows2000/XP で使用する場合

基本的には Windows2000/XP 用ヘッダファイル DIOLIB2K.H とライブラリファイル DIOLIB2K.LIB を新規プロジェクトに追加し、Windows95/98/Me で作成したソースフ ァイルにインクルード後、コンパイルすることによって使用可能になります。 但し、以下の関数を使用されている場合は、Windows2000/XP の DIOLIB2K.DLL ではサポートしておりませんのでご注意ください。

ShowCardUtil() ResistAsyncProcCall() ReleaseAsyncProcCall() StartHWIntMyVxD()<sup>1</sup> EndHWIntMyVxD()<sup>2</sup> GetDIlversion()

1.Windows2000/XP では、StartHWIntMyVxdBuf()をご使用ください。

2.Windows2000/XP では、EndHWIntMyVxdBuf()をご使用ください。

基本デジタル入出力プログラム

本サンプルプログラムは 16 ポート(8 ポート単位)を使って外部機器とオンオフ入出 力を行う場合のサンプルプログラムです。

| 🚰基本デョ           | ジタル入出力      |                                   |      |    | ×   |  |  |  |
|-----------------|-------------|-----------------------------------|------|----|-----|--|--|--|
|                 | 入出力方向選択     | と <sup>、</sup> ットマッフ <sup>®</sup> | 入力   | 出力 |     |  |  |  |
| P15-P 8         | ⊙ IN ⊙ OUT  |                                   | 00 H | Н  | 入出力 |  |  |  |
| P 7-P 0         | ○ IN ⓒ OUT  |                                   | 00 H | Н  | 入出力 |  |  |  |
| 全柿°-小入力 全柿°-小出力 |             |                                   |      |    |     |  |  |  |
| REX-5055        | 正常検出 I/0\`、 | -ኢ:0×120 IRQ₫                     | 号:10 |    |     |  |  |  |

DIO カードを制御するためには、プログラム実行時にカードに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のリソース情報を取得する必要があります。また、 Outport(), Inport()により方向設定を行ってください。

リソースの取得、方向設定の記述例

```
BOOL DIg_OnInitDialog (HWND hwnd, HWND hwndFocus, LPARAM IParam)
{
   WORD
              SlotNo;
                                  //Windows95/98 互換用
   WORD
              Status;
   /*
    *
        スロットに挿入されている自分のカードのリソース情報を取得する
    */
   Status = GetMyCardResource( hwnd, MyCardName, sizeof(MyCardName), 0, &MyIOBase, &MyIrqNo );
   if (Status == 0)
   {
       /*
        *
                 リソース情報を表示する
        */
       sprintf(MsgBuf, "REX-5055 正常検出 I/O ペース:0x%x IRQ 番号:%d", MyIOBase, MyIrqNo);
       SetDIgItemText( hwnd, IDS_STATUS, MsgBuf );
       // 出力方向設定 (PI00-PI07)
       Status = Outport( 0x120, 0x1 );
       // 下位バイト (PI00-PI07) すべて出力
       Status = Outport( 0x122, 0xFF );
       Return = TRUE ;
   }
   return TRUE ;
```

外部イベント監視プログラム

本サンプルプログラムは割り込みソースビットで指定したポートに外部機器からの 割り込み信号を入力し、オンオフの状態変化を検出するサンプルプログラムです。

割り込みソースビットを指定して StartHWIntPostMessage()を実行することにより、割 り込み発生に同期したユーザ定義メッセージ WM\_VXDEVENT が割り込みハンドラか ら送られてきます。このとき、追加情報 wParam の上位バイトには PIO7-PIO0 の読み 込み値が、下位バイトには PIO15-PIO8 の読み込み値がセットされています。また、 IParam には割り込み発生の累計カウント数がセットされています。

リソース取得、方向設定方法については、 基本デジタル入出力プログラムをご 参照ください。

| 🚰 外部イヘシト監視フログラム 🛛 🔀                   |      |          |            |  |  |  |
|---------------------------------------|------|----------|------------|--|--|--|
| 各種設定                                  |      |          |            |  |  |  |
| 終了回数                                  | 100  | ንትአቲስቃኑ  | PI000 💌    |  |  |  |
| 下位方向設定                                | 入力 💽 | ŀリカ°モ∽Ւ° | 立下りエッジ     |  |  |  |
| 上位方向設定                                | 出力 💽 | 別ア条件     | 下位バイトリード 💌 |  |  |  |
| - 外部八* 가検出                            |      |          |            |  |  |  |
| リート 値 12345678 H 発生回数 0 イベント検出 イベント停止 |      |          |            |  |  |  |
| REX-5055 正常検出 I/Oベース:0×120 IRQ番号:10   |      |          |            |  |  |  |

```
void Cmd_OnStartEvent ( HWND hwnd )
{
   WORD
          Status;
                       // データレジスタ下位バイトの入出力方向
   WORD
          DataRegLo;
   WORD
         DataRegHi;
                       // データレジスタ上位バイトの入出力方向
   WORD
          IntPIONo;
                       // 割り込みソース入力ポート番号
                       // 割り込みトリガーモード
   WORD
         TrgMode:
   WORD
         ClrMode;
                       // 割り込みクリア条件
   WORD
         MaxCount;
                       // 割り込み終了回数
   Status = StartHWIntPostMessage( hwnd, MyIOBase, MyIrqNo, DataRegLo, DataRegHi, IntPIONo,
   TrgMode, ClrMode, MaxCount );
   if (Status != 0)
   {
      sprintf( MsgBuf, "StartHWIntPostMessage() I7-:%d", Status );
      SetDIgItemText( hwnd, IDS_STATUS, MsgBuf );
   }
```

```
BOOL CALLBACK DIg_Proc (HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM IParam)
{
   switch ( uMsg )
    {
   case WM_INITDIALOG:
       DIg_OnInitDialog( hwnd, (HWND)wParam, IParam );
        return TRUE;
   case WM_COMMAND:
       DIg_OnCommand( hwnd, (int)(LOWORD(wParam)), (HWND)(IParam), (UINT)HIWORD(wParam));
        return TRUE:
   case WM_VXDEVENT:
        DIg_OnUserDefineMessage( hwnd, uMsg, wParam, IParam );
        return TRUE;
   case WM_DESTROY:
        DIg_OnDestroy( hwnd );
        return TRUE;
    }
    return FALSE;
}
```

```
void DIg_OnUserDefineMessage (HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM IParam)
{
                         // 割り込み終了回数
   WORD MaxCount;
   EventCount++;
   sprintf( MsgBuf, "%d", EventCount );
   SetDIgItemText( hwnd, IDS_COUNTER, MsgBuf );
   sprintf( MsgBuf, "0x%x", wParam );
   SetDIgItemText( hwnd, IDS_EVENTVAL, MsgBuf );
   /* 終了回数 */
   MaxCount = (WORD)GetDigitemint( hwnd, IDE_COUNT, NULL, FALSE );
   if( MaxCount == EventCount )
   {
       sprintf(MsgBuf, "指定個数終了");
       SetDIgItemText( hwnd, IDS_STATUS, MsgBuf );
   }
}
```

### データ受信プログラム

本サンプルプログラムは割り込みハンドラ内ですべての入出力を行うことにより高 速割り込み処理を行うことが可能なサンプルプログラムです。

サンプルプログラムの割り込みハンドラは、入力方向に設定されたデータレジスタ の値をリードし、指定バッファに格納します。指定回数の割り込み処理が終了すると、 呼び出し側プログラムにユーザ定義メッセージをポストし、追加情報 IParam には割り 込み発生の累計カウント数がセットされています。

入出力データレジスタと割り込み終了回数を指定して、StartHWIntMyVxdBuf()を実行することにより、割り込みハンドラ内部で全ての処理を行う高速版の割り込み処理 を開始します。

リソース取得、方向設定方法については、 基本デジタル入出力プログラムをご参照ください。

| 😚 データ受信プログラム                        |        |        |        | ×   |  |
|-------------------------------------|--------|--------|--------|-----|--|
| 各種設定<br>終了回数 100                    |        | ソースビット | PIO00  | •   |  |
| 入力データレジスタ                           | •      | りがモード  | 立下りエッシ | •   |  |
|                                     |        | 切了条件   | 下位バイリー | * • |  |
| C:¥WINDOWS¥デスクトップ¥505               | 55.dat |        | 77     | 们名  |  |
|                                     |        | イベント検  | 出 イベント | 停止  |  |
| REX-5055 正常検出 I/Oベース:0x120 IRQ番号:10 |        |        |        |     |  |

| voi<br>{  | d Cmd_O | nStartEvent ( | HWND hwnd )       |  |  |
|---|---------|---------------|-------------------|--|--|
| ľ   | WORD    | Status:       |                   |  |  |
|   | WORD    | DataReg;      | // 入力データレジスタ      |  |  |
|   | WORD    | IntPIONo;     | // 割り込みソース入力ポート番号 |  |  |
|   | WORD    | TrgMode;      | // 割り込みトリガーモード    |  |  |
|   | WORD    | CIrMode;      | // 割り込みクリア条件      |  |  |
|   | WORD    | MaxCount;     | // 割り込み終了回数       |  |  |
| <pre>Status = StartHWIntMyVxdBuf( hwnd, MyIOBase, MyIrqNo, DataReg, IntPIONo, TrgMode, CIrMode,<br/>MaxCount, pRcvData );<br/>if ( Status != 0 )<br/>{<br/>sprintf( MsgBuf, "StartHWIntMyVxdBuf()Iラ-:%d", Status );<br/>SetDIgItemText( hwnd, IDS_STATUS, MsgBuf );<br/>return:</pre> |         |               |                   |  |  |
|   | 3       | ,             |                   |  |  |
| }   | J       |               |                   |  |  |

```
void DIg_OnUserDefineMessage (HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM IParam)
{
   WORD
           Status:
   WORD
           EventCount;
           hFile; // ファイルハンドル
nBytesRead; // リードしたバイト数
szFile[MAX_PATH]; // ファイル名
   HANDLE hFile;
   DWORD nBytesRead;
   char
   MessageBeep(0xFFFFFFF);
   /* サイズ取得 */
   EventCount = GetDigitemint( hwnd, IDE_COUNT, NULL, FALSE );
   /* 選択された送信ファイル名取得 */
   GetDlgltemText( hwnd, IDS_RCVFILE, szFile, sizeof(szFile) );
   /* ファイルオープン */
   hFile = CreateFile( szFile,
                                                      /* アクセスモート゛*/
                 GENERIC_WRITE,
                 0, /* 共有モート* */
(LPSECURITY_ATTRIBUTES)NULL, /* ポインタ */
                 0,
                 CREATE_ALWAYS,
                                                      /* 創作方法 */
                                                 /* ファイルの限定 */
/* テープレートファイルのハンドル */
                 FILE_ATTRIBUTE_NORMAL,
                  ( HANDLE )NULL);
   Status = WriteFile( hFile, pRcvData, EventCount, &nBytesRead, NULL );
   if ( (Status != TRUE) || (nBytesRead != EventCount) )
   {
       CloseHandle( hFile );
       LocalFree( pRcvData );
       sprintf( MsgBuf, "WriteFile error" );
       SetDIgItemText( hwnd, IDS_STATUS, MsgBuf );
        return ;
    }
   CloseHandle( hFile );
   LocalFree( pRcvData );
   sprintf(MsgBuf, "指定個数終了");
   SetDIgItemText( hwnd, IDS_STATUS, MsgBuf );
```

## (3-5) BASIC 言語 API ライブラリ解説

### (3-5-1) Visual BASIC によるアプリケーション開発

Visual BASIC 5.0 以上を使って REX-5055 DIO PC カードを制御するアプリケーショ ンを開発する場合は、本製品に添付されている 32Bit 版 DLL の関数をコールする必 要があります。また、割り込み制御を行う場合は OLE カスタムコントロール(OCX)を使 用します。

Visual BASIC で作成したアプリケーションプログラムから 32Bit 版 DLL を呼び出すためには、

モジュールファイルで DLL 関数の参照宣言を行う。

必要があります。

また、割り込み制御を行う場合は割り込みハンドラから送られてくるユーザ定義メッ セージを Visual BASIC 側のアプリケーションで受け取るために、本製品に添付されて いる OLE カスタムコントロール MBOX を使用します。 MBOX の使用方法については、 (3-5-2)カスタムコントロールを参照してください。

| PCカードに割り当てられてるリソースを取得する<br>Windows2000/XP では、Windows95/98/Me と同様に、GetMyCardResource()をサポートし<br>ています。アプリケーションの初期化手続き部分で GetMyCardResource()により PC カードに<br>割り当てられている I/O ベースアドレス・IRQ 番号を取得してください。<br>PC カードへの入出力<br>Windows95 と同様、Visual Cでは、I/O 入出力関数がサポートされていません。従って、PC<br>カードへのI/O 入出力はDLLでサポートされているOutPort()・InPort()・wOutPort()・wInPort()<br>を使って行います。<br>割り込み制御<br>DLL では割り込みサービスを提供しています。割り込みサービスには、 |
|---|
| (1)割り込みハンドラからポストメッセージにより割り込み通知を受け取る方法<br>(2)割り込みハンドラ内で全ての処理を行う方法<br>があります。高速な処理が要求されるような場合は、(2)の方法で行う必要があります。詳細   |

# DLL 関数仕様

|   | GetMyCa | ardResource   |  | リソース情報の取得                          |  |  |
|---|---------|---|--|------------------------------------|--|--|
| 書 | 式       | Declare Function <b>GetMyCardResource</b> Lib "diolib2K.dll" (ByVal <b>hWnd</b> As Long<br>ByVal <b>MyCardName</b> As String, ByVal <b>NameLen</b> As Integer, ByVal <b>SlotNo</b> A<br>Integer, <b>IOBase</b> As Integer, <b>IrqNo</b> As Integer) As Long |  |                                    |  |  |
| 機 | 能       | 指定スロットに挑<br>ます。一致した場<br>情報を返します。  | 挿入されているカードが指定のフ<br>合は、割り当てられている I/0  | カード名と一致するか否か調べ<br>ベースアドレス・IRQ リソース |  |  |
| 引 | 数       | hWnd<br>MyCardName<br>NameLen<br>SlotNo<br>IOBase<br>IrqNo  | <ul> <li>&gt; ウィンドウハンドル</li> <li>&gt; カード識別名を示す文字</li> <li>&gt; 文字列バッファの長さ<br/>(Windows95/98 互換用)</li> <li>&gt; I/0 リソース情報を格納</li> <li>&gt; IRQ リソース情報を格納</li> </ul> | 列へのメモリアドレス<br>する変数アドレス<br>する変数アドレス |  |  |

戻値 正常終了時0を返し、リソースが正常に取得できなかった場合は-1を返します。

### StartHWIntPostMessage ユーザ定義メッセージ割り込みの開始

- 書式 Declare Function **StartHWIntPostMessage** Lib "diolib2K.dll" (ByVal **hWnd** As Long, ByVal **MyIOBase** As Integer, ByVal **MyIrqNo** As Integer, ByVal **DirDataRegLo** As Integer, ByVal **DirDataRegHi** As Integer, ByVal **IrqPin** As Integer, ByVal **IrqMode** As Integer, ByVal **IntCIrMode** As Integer, ByVal **StopCount** As Integer) As Long
- 機 能 ユーザ定義メッセージ版の割り込み処理を開始します。割り込みが発生すると 割り込みハンドラからアプリケーションにメッセージを通知します。 wParam の 上位バイトに PI07-PI00 の読み込み値が、下位バイトに PI015-PI08 の読み込み 値がセットされています。また、 IParam には割り込み発生の累計カウント数が セットされています。
- 引数 hWnd ▶ ユーザアプリケーションのウィンドウハンドル MyIOBase ▶ カードに割り当てられている 1/0 ベースアドレス MylrqNo ▶ カードに割り当てられている IRQ 番号 DirDataRegLo ▶ データレジスタ下位// 小の入出力方向 0:入力方向 1:出力方向 DirDataRegHi ▶ データレジスタ上位バイトの入出力方向 0:入力方向 1:出力方向 IrqPin ▶ 割り込みソース入力ポート番号 0:PI00 1:PI01 2:PI02 ··· 15:PI015 I rqMode ▶ 割り込みトリガーモード 0:立下りエッジ 1:立上りエッジ IntClrMode ▶ 割り込みクリア条件 0:下位バイトリード 1:上位バイトリード 2:下位バイトライト 3:上位バイトライト StopCount ▶ 割り込み終了回数
- 戻値 正常終了0を返します。I/0ベースアドレス設定エラー等は-1を返し、その他のマイナスエラーはDirDataRegLo・DirDataRegHi・IrqMode・IntCIrModeの設定エラーです。

| E | EndHWIr | ntPostMessage                               | ユーザ定義メッセージ割り込みの終了   |
|---|---------|---|---|
| 書 | 式       | Declare Function EndHWIntPost               | <b>Message</b> Lib "diolib2K.dll" () As Long              |
| 機 | 能       | ユーザ定義メッセージ版の割り込<br>ジ版による割り込み処理終了時に<br>ください。 | 込み処理を終了します。 ユーザ定義メッセー<br>は、必ず EndHWIntPostMessage()を呼び出して |

- 引数 なし
- 戻値 常に0を返す

| StartHWIn           | tMyVxdBuf   | 高速割り込み処理の開始   |
|---------------------|---|---|
| 書式 D<br>L<br>B<br>L | eclare Function <b>S</b><br>ong, ByVal <b>MyIOB</b><br><b>nputDataReg</b> As Inte<br>yVal <b>IntCIrMode</b> As<br>ong | <b>tartHWIntMyVxdBuf</b> Lib "diolib2K.dll" (ByVal <b>hWnd</b> As<br><b>ase</b> As Integer, ByVal <b>MyIrqNo</b> As Integer, ByVal<br>eger, ByVal <b>IrqPin</b> As Integer, ByVal <b>IrqMode</b> As Integer,<br>s Integer, ByVal <b>StopCount</b> As Integer, <b>Buf</b> As Any) As |
| 機能 I<br>-<br>約<br>て | rqPin で指定した上<br>- ト(上位ポート)(<br>内します。指定した<br>び出し側に返します。  | 位ポート(下位ポート)に割り込み信号が発生すると下位ポ<br>Dデータレジスタを読み込み、指定したバッファ Buf に格<br>StopCount 数だけ割り込みが発生すると本関数は処理を呼   |
| 引数h<br>M<br>M<br>I  | Wnd<br>ylOBase<br>ylrqNo<br>nputDataReg<br>rqPin  | <ul> <li>&gt; ユーザアプリケーションのウィンドウハンドル</li> <li>&gt; カードに割り当てられている I/0 ベースアドレス</li> <li>&gt; カードに割り当てられている IRQ 番号</li> <li>&gt; 入力データレジスタ         <ul> <li>0:下位バイト(PI07-PI00) 1:上位バイト(PI015-PI08)</li> <li>&gt; 割り込みソース入力ポート番号</li> </ul> </li> </ul>                                  |
| 1                   | rqMode  | 0:PI00 1:PI01 2:PI02 ・・・ 15:PI015<br>> 割り込みトリガーモード<br>0:立下りエッジ 1:立上りエッジ   |
| I<br>S<br>B         | ntCIrMode<br>topCount<br>uf   | <ul> <li>&gt; 割り込みクリア条件</li> <li>0:下位バイトリード 1:上位バイトリード</li> <li>2:下位バイトライト 3:上位バイトライト</li> <li>&gt; 割り込み終了回数</li> <li>&gt; データを格納するバッファアドレス</li> </ul>  |
|                     |   |   |

戻値 正常終了時0を返します。引数設定エラー等は-1、メモリアロケーションエラ ーの場合は-2を返します。

| EndHWIntMyVxdBuf |   | ntMyVxdBuf                                      | 高速割り込み処理の終了                                  |  |  |
|------------------|---|---|--|--|--|
| 書                | 式 | Declare Function EndHWIntMyVxdBuf               | Lib "diolib2K.dll" () As Long                |  |  |
| 機                | 能 | 高速版の割り込み処理を終了します。<br>理終了時は、必ずEndHWIntMyVxdBuf() | StartHWIntMyVxdBuf()による割り込み処<br>)を呼び出してください。 |  |  |
| 引                | 数 | なし  |  |  |  |
| 戻                | 値 | 常に 0 を返す  |  |  |  |

| OutPort | 1パイトをポート出力   |
|---------|--|
| 書 式     | Declare Function <b>OutPort</b> Lib "diolib2K.dll" (ByVal <b>IOAddr</b> As Integer, ByVal <b>OutVal</b> As Integer) As Integer     |
| 機 能     | 1 バイトをポートに出力   |
| 引 数     | IOAddr > ポート番号<br>OutVal > バイト出力値(上位バイトは無視)  |
| 戻 値     | バイト出力値をそのまま返し、ドライバ呼び出しエラーは-1 を返します。  |
| wOutPo  | rt 1ワードをポート出力  |
| 書式      | Declare Function <b>wOutPort</b> Lib "diolib2K.dll" (ByVal <b>IOAddr</b> As Integer,<br>ByVal <b>OutVal</b> As Integer) As Integer |
| 機 能     | 1 ワードをポートに出力   |
| 引 数     | IOAddr > ポート番号<br>OutVal > ワード出力値  |
| 戻 値     | ワード出力値をそのまま返し、ドライバ呼び出しエラーは-1 を返します。  |
| InPort  | 1パイトをポート入力   |
| 書式      | Declare Function <b>InPort</b> Lib "diolib2K.dll" (ByVal <b>IOAddr</b> As Integer) As<br>Integer                                   |
| 機 能     | ポートから1バイト読み込む  |
| 引 数     | IOAddr > ポート番号   |
| 戻 値     | ポートから読み込んだバイトデータを返します。ドライバ呼び出しエラーは-1<br>を返します。(上位バイトは無視してください)   |
| winPort | 1ワードをポート入力   |
| 書 式     | Declare Function <b>wInPort</b> Lib "diolib2K.dll" (ByVal <b>IOAddr</b> As Integer)<br>As Integer                                  |
| 機 能     | ポートから1ワード読み込む  |
| 引 数     | IOAddr > ポート番号   |
| 戻 値     | ポートから読み込んだワードデータを返します。ドライバ呼び出しエラーは-1<br>を返します。   |

## (3-5-2) カスタムコントロール

下記画面は、VB5.0 でのデザイン完成時の画面です。割り込み発生に同期した ユーザ定義メッセージを VB で作成したプログラムで受け取るために、本製品添 付の OLE カスタムコントロール(OCX)MBOX を使用します。次頁より作成の手順に ついて説明します。



#### Step.1 OCX のレジストリ登録 (割り込みサービス使用時必須)

本製品添付の OCX "MBOX5055.OCX"を VB で使用するためには、VB の CD-ROM に添付されているツール"REGSVR32.EXE"を使って OCX のレジストリ登録 を行います。"REGSVR32.EXE"は 32 ビットコンソールアプリケーションですので、 Windows の DOS BOX から実行します。尚、"REGSVR32.EXE"は VB の CD-ROM に 添付されています。

OCX をレジストリー登録するときは、下記構文で実行します。

>REGSVR32 "ドライブ名":¥WINNT¥SYSTEM¥Mbox5055.ocx

OCX をレジストリー登録から削除するときは、"/U"を付けて下記構文で実行します。

>REGSVR32 /U "ドライブ名":¥WINNT¥SYSTEM¥Mbox5055.ocx

| RegSvr32   | RepSel2   |
|--|---|
| DIFAGINTESENER IN CAMENITARYSTEMANBOUGGEGOCK succeeded | DILImegisterServer in C#WINNTHSYSTEMNHE0/5025.0CX succeeded |
|  |   |

登録成功メッセージ

登録削除成功メッセージ

Step.2 DIOLIB2K 関数の Declare 宣言

次に、VB プログラムの作成に入ります。VB デザインメニューから新規プロジェクト を作成し、「プロジェクト」の「標準モジュールの追加」から標準モジュールを追加しま す。追加した標準モジュールファイルで DLL 関数の参照宣言を行います。宣言部分 は、サンプルプログラム"REX5055.BAS"の下記部分をコピーしてください。

| 'DLL 外部関数の参照宣言  |  |  |  |  |
|---|--|--|--|--|
| Declare Function <b>OutPort</b> Lib "diolib2K.dll" (ByVal IOAddr As Integer, ByVal OutVal As Integer) As Integer    |  |  |  |  |
| Declare Function wOutPort Lib "diolib2K.dll" (ByVal IOAddr As Integer, ByVal OutVal As Integer) As Integer          |  |  |  |  |
| Declare Function InPort Lib "diolib2K.dll" (ByVal IOAddr As Integer) As Integer                                     |  |  |  |  |
| Declare Function winPort Lib "diolib2K.dll" (ByVal IOAddr As Integer) As Integer                                    |  |  |  |  |
| Declare Function StartHWIntPostMessage Lib "diolib2K.dll" (ByVal hWnd As Long, ByVal MyIOBase As Integer,           |  |  |  |  |
| ByVal MyIrqNo As Integer, ByVal DirDataRegLo As Integer, ByVal DirDataRegHi As Integer, ByVal IrqPin As             |  |  |  |  |
| Integer, ByVal IrqMode As Integer, ByVal IntCIrMode As Integer, ByVal StopCount As Integer) As Long                 |  |  |  |  |
| Declare Function EndHWIntPostMessage Lib "diolib2K.dll" () As Long  |  |  |  |  |
| Declare Function GetMyCardResource Lib "diolib2K.dll" (ByVal hWnd As Long, ByVal MyCardName As String, ByVal        |  |  |  |  |
| NameLen As Integer, ByVal SlotNo As Integer, IOBase As Integer, IrqNo As Integer) As Long                           |  |  |  |  |
| Declare Function <b>StartHWIntMyVxdBuf</b> Lib "diolib2K.dll" (ByVal hWnd As Long, ByVal MyIOBase As Integer, ByVal |  |  |  |  |
| MyIrqNo As Integer, ByVal InputDataReg As Integer, ByVal IrqPin As Integer, ByVal IrqMode As Integer, ByVal         |  |  |  |  |
| IntClrMode As Integer, ByVal StopCount As Integer, Buf As Any) As Long  |  |  |  |  |
| Declare Function EndHWIntMyVxdBuf Lib "diolib2K.dll" () As Long   |  |  |  |  |
|   |  |  |  |  |

×

A abi

ø

間

03 03

Step.3 MBOX OLE Control Module の追加 (割り込みサービス使用時必須)

VB5.0/6.0 の場合、VB デザインメニューの「プロジェクト」の「コンポーネ ント」を起動し、利用可能なコントロールから「MBOX OLE Control module」 をチェックします。VB ツールバーに MBOX が追加されます。

➤ VB5.0/6.0 の場合

Page.3-26

| VD3.0/0.000场口   |  |
|---|--|
| a*  | ×  |
| 12月1日 〒9月1 挿入可能なオフジェ外   |  |
| <ul> <li>Kodak イメージ スキャン コントロール</li> <li>Kodak イメージ管理コントロール</li> <li>Kodak イメージ管理コントロール</li> <li>Kodak イメージ2編集コントロール</li> <li>Kodak イメージ2編集コントロール</li> <li>Un Library</li> <li>MBOX OLE Control module</li> <li>MCDWndX Control</li> <li>McDWndX Control</li> <li>McDWndX Control</li> <li>Mcrosoft Access BarCode Control 90</li> <li>Microsoft ActiveMovie Control</li> </ul> | ▲<br>一<br>一<br>一<br>一<br>一<br>一<br>一<br>一<br>一<br>一<br>一<br>一<br>一 |
| HERE COMMINITYSYSTEMAMPOVERED COV   |  |
| AND CONTRACTOR DO CONTRACTOR  |  |
| ОК  | キャンセル 通用(A)  |
|   |  |
|   |  |

フォームに MBOX(OCX)を貼り付ける (割り込みサービス使用時必須) Step.4

フォームを作成し、割り込みハンドラが割り込み起動元プログラムに送るユーザ定 義メッセージを受け取るための MBOX(OCX)を貼り付けます。これにより、割り込みが 発生すると MBOX がサービスするプロシージャ

MBOX5055\_OnMsgPost(ByVal wParam As Integer, ByVal IParam As Long) が呼び出されます。この中で、割り込み通知に同期した処理を記述します。

| 🛋 外部イベント監視プログラム   |              |                      |  |  |  |  |
|---|--------------|----------------------|--|--|--|--|
| ▲ 「各種設定————————————————————————————————————   |              |                      |  |  |  |  |
| ·<br>終了回数 / ソ   | ースビット Combo1 | •                    |  |  |  |  |
| ·<br>· 下位方向設定 Combo1 ▼ M  | リガモート Combol | •                    |  |  |  |  |
| ·<br>上位方向設定 Combo1 マク   | リア条件 Combol  | •                    |  |  |  |  |
|   | 050X         | (+ 5.1 + A.U. 1      |  |  |  |  |
| ・ リートが値 「12345 発生回数   |              | 14/21/横出<br>14/21/停止 |  |  |  |  |
|   |              | 終了                   |  |  |  |  |
|   |              |                      |  |  |  |  |
|   |              |                      |  |  |  |  |
|   |              |                      |  |  |  |  |
| UtbDioSano - Form1 (0=k2)   |              |                      |  |  |  |  |
| MBOX5055  | 💌 On         | MsePost              |  |  |  |  |
| Private Sub WBDX5055_OnWsgPost(ByYal =Param As Integer, ByYal  Param As Long)                         |              |                      |  |  |  |  |
| EventCount = EventCount + 1<br>IDS_COUNTER.Caption = EventCount<br>IDS_EVENTYAL.Caption = Hex(wParam) |              |                      |  |  |  |  |
| '終了回数<br>MaxCount = Val(IDE_COUNT.Text)<br>If MaxCount = EventCount Then                              |              |                      |  |  |  |  |
| IDS_STATUS.Caption = "指定個就終了"<br>End If<br>End Sub  |              |                      |  |  |  |  |
| -3.   |              |                      |  |  |  |  |

(3-5-3) Visual BASIC サンプルプログラム

本製品には、以下の3つのサンプルプログラムを添付しております。

基本デジタル入出力プログラム (割り込みを使わないで単純に入出力を行うサンプルプログラム)

外部イベント監視プログラム (割り込みを使ったサンプルプログラム)

データ受信プログラム (高速割り込みを使ったサンプルプログラム)

次頁より、各サンプルプログラムについて解説いたします。 (詳細については各サンプルプログラムのソースをご参照〈ださい。)

Windows95/98/Me で作成したアプリケーションを Windows2000/XP で使用する場合

基本的にはモジュールファイルで DLL 関数の参照宣言を行い、コンパイルすることによって使用可能になります。

但し、以下の関数を使用されている場合は、Windows2000/XP の DIOLIB2K.DLL ではサポートしておりませんのでご注意ください。

ShowCardUtil() ResistAsyncProcCall() ReleaseAsyncProcCall() StartHWIntMyVxD()<sup>1</sup> EndHWIntMyVxD()<sup>2</sup> GetDIlversion()

1. Windows2000/XP では、StartHWIntMyVxdBuf()をご使用ください。

2. Windows2000/XP では、EndHWIntMyVxD()をご使用ください。
基本デジタル入出力プログラム

本サンプルプログラムは 16 ポート(8 ポート単位)を使って外部機器とオンオフ入出 力を行う場合のサンプルプログラムです。

| 🛢 基本形列从入出力                      |                    |                            |                            |                   |
|---------------------------------|--------------------|----------------------------|----------------------------|-------------------|
| 入出力方向選択                         | 沢 ビットマッフ。          | 入力                         | 出力                         |                   |
| Р15-Р8 СІМ €ОU<br>Р7-Р0 СІМ €ОU |                    | <u>ро</u> н<br><u>ро</u> н | <u>оо</u> н<br><u>оо</u> н | 入出力           入出力 |
| REX-5055 正常検出 1/0^3             | -ㅈ:0x120 IRQ番号: 10 | 全ポート)                      |                            | 全ポート出力<br>終了      |

DIO カードを制御するためには、プログラム実行時にカードに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のリソース情報を取得する必要があります。また、 Outport(), Inport()により方向設定を行ってください。

リソースの取得、方向設定の記述例

```
Global Const MyName = "REX5055 DIO PC Card"
                                           '製品情報タプルの製品名の定義
                                           'カード I/0 ベースアドレス
Global MyIOBase As Integer
                                           ' カード IRQ 番号
Global MylrqNo As Integer
Private Sub Form_Load()
   Dim RetCode As Integer
   Dim STATUS As Integer
   RetCode = GetMyCardResource(hWnd, MyCardName, Len(MyCardName), 0, MyIOBase, MyIrqNo)
   If RetCode = 0 Then
       IDS_STATUS.Caption = "REX-5055 正常検出 I/0 ペース:0x" + Hex(MyIOBase) + " IRQ 番号:" +
       Str(MylrqNo)
       '出力方向設定(PI00-PI07)
       STATUS = Outport( &H120, &H1 )
       '下位バイト (PI00-PI07) すべて出力
       STATUS = Outport( &H122, &HFF )
       Exit Sub
   End If
   IDS_STATUS.Caption = "DIO カート' 検出エラー"
End Sub
```

外部イベント監視プログラム

本サンプルプログラムは割り込みソースビットで指定したポートに外部機器からの 割り込み信号を入力し、オンオフの状態変化を検出するサンプルプログラムです。

割り込みソースビットを指定して StartHWIntPostMessage()を実行することにより、割 り込み発生に同期したユーザ定義メッセージが割り込みハンドラから送られてきま す。このとき、追加情報 wParam の上位バイトには PIO7-PIO0 の読み込み値が、下位 バイトには PIO15-PIO8 の読み込み値がセットされています。また、IParam には割り 込み発生の累計カウント数がセットされています。

リソース取得、方向設定方法については、 基本デジタル入出力プログラムをご 参照ください。

| Ē | ■ 外部イヘント監視プロ  | コグラム            |           |            |
|---|---------------|-----------------|-----------|------------|
|   | -各種設定         |                 |           |            |
|   | 終了回数          | 100             | ソースビット    | PIO00      |
|   | 下位方向設定        | 入力 💽            | りがモード     | 立ち下がりエッジ・💌 |
|   | 上位方向設定        | 出力 💽            | 切了条件      | 下位バイリード・   |
|   | -外部イベント検出     |                 |           |            |
|   | リートኘ直  12345  | 発生              | 回数  0     |            |
|   | REX-5055 正常検出 | ן I∕O^°−ス:0x120 | IRQ番号: 10 |            |

| <pre>Private Sub IDB_START_EVENT_Click()</pre>  |   |
|---|---|
| Dim STATUS As Integer   |   |
| Dim DataRegLo As Integer  | ' データレジスタ下位バイトの入出力方向  |
| Dim DataRegHi As Integer  | ' データレジスタ上位バイトの入出力方向  |
| Dim IntPIONo As Integer   | '割り込みソース入力ポート番号   |
| Dim TrgMode As Integer  | ' 割り込みトリガーモード   |
| Dim ClrMode As Integer  | ' 割り込みクリア条件   |
| Dim MaxCount As Integer   | '割り込み終了回数   |
| Dim OleHandle As Long   | ' MB0X5055.0CX ハント・ル  |
| OleHandle = <b>MBOX5055.GetMboxWnd</b> ()<br>If (OleHandle = 0) Then<br>MsgBox "OLE のハンドルが取得でき<br>Exit Sub<br>End If<br>STATUS = <b>StartHWIntPostMessage</b> (O<br>IntPIONo, TrgMode, CIrMode, MaxCou<br>If STATUS <> 0 Then<br>IDS_STATUS.Caption = "StartHWI<br>Exit Sub | きません。", vbOKOnly + vbCritical, "エラー"<br>HeHandle, MyIOBase, MyIrqNo, DataRegLo, DataRegHi,<br>nt)<br>ntPostMessage()Iラ-:" + Str(STATUS) |
| End If  |   |
| End Sub   |   |

```
Private Sub MBOX5055_OnMsgPost(ByVal wParam As Integer, ByVal IParam As Long)
Dim MaxCount As Integer '割り込み終了回数
EventCount = EventCount + 1
IDS_COUNTER.Caption = EventCount
IDS_EVENTVAL.Caption = Hex(wParam)
'終了回数
MaxCount = Val(IDE_COUNT.Text)
If MaxCount = EventCount Then
IDS_STATUS.Caption = "指定個数終了"
End If
End Sub
```

データ受信プログラム

本サンプルプログラムは割り込みハンドラ内ですべての入出力を行うことにより高 速割り込み処理を行うことが可能なサンプルプログラムです。

サンプルプログラムの割り込みハンドラは、入力方向に設定されたデータレジスタ の値をリードし、指定バッファに格納します。指定回数の割り込み処理が終了すると、 呼び出し側プログラムにユーザ定義メッセージをポストし、追加情報 IParam には割り 込み発生の累計カウント数がセットされています。

入出力データレジスタと割り込み終了回数を指定して、StartHWIntMyVxdBuf()を実行することにより、割り込みハンドラ内部で全ての処理を行う高速版の割り込み処理 を開始します。

リソース取得、方向設定方法については、 基本デジタル入出力プログラムをご参照ください。

| 🚔 外部イヘント監視ブ1                            | コグラム                         |          |         | <u>- 0 ×</u> |
|---|------------------------------|----------|---------|--------------|
| 各種設定                                    |                              |          |         |              |
| 終了回数                                    | 100                          | ソースビット   | PIO00   | •            |
| 入力データレジスタ                               | 下位バイト 💌                      | トリガモート   | 立ち下がりエッ | 197 <b>-</b> |
|   |                              | 夘ア条件     | 下位パイヤート |              |
| C:¥WINDOWS¥7*                           | スクトップ <sup>°</sup> ¥5055.dat |          | 771     | 1/名          |
|   |                              | <u>_</u> | 検出 イベント | 停止           |
| REX-5055 正常検出 I/Oベース:0x120 IRQ番号: 10 終了 |                              |          |         |              |

```
Private Sub IDB_START_EVENT_Click()
   Dim STATUS As Integer
                               ' 入力データレジスタ
   Dim DataReg As Integer
                               ' 割り込みソース入力ポート番号
   Dim IntPIONo As Integer
                               ' 割り込みトリガーモード
   Dim TrgMode As Integer
                               ' 割り込みクリア条件
   Dim ClrMode As Integer
                             |割り込み終了回数
| MB0X5055.0CX ルンドル
   Dim MaxCount As Integer
   Dim OleHandle As Long
   ReDim pRcvData(MaxCount - 1) As Byte
   'OLE のウィンドウハンドル取得
   OleHandle = MBOX5055.GetMboxWnd()
   If (0leHandle = 0) Then
       MsgBox "OLE のハンドルが取得できません。", vbOKOnly + vbCritical, "エラー"
       Exit Sub
   End If
   STATUS = StartHWIntMyVxdBuf(OleHandle, MyIOBase, MyIrqNo, DataReg, IntPIONo, TrgMode,
   ClrMode, MaxCount, pRcvData(0))
   If STATUS <> 0 Then
       IDS_STATUS.Caption = "StartHWIntMyVxdBuf()I7-: " + Str(STATUS)
       Exit Sub
   End If
End Sub
```

```
Private Sub MBOX5055_OnMsgPost(ByVal wParam As Integer, ByVal IParam As Long)
Dim EventCount As Integer '割り込み終了回数
Beep
'サイス 取得
EventCount = IDE_COUNT.Text
'選択された送信ファイル名取得
RcvFileName = IDS_RCVFILE.Caption
'ファイルオープン
Open RcvFileName For Binary As #1
Put #1, , pRcvData
' ファイルを閉じる
Close #1
IDS_STATUS.Caption = "指定個数終了"
```

(空白ページ)

第4章 MS-DOS/Windows3 1解説

(4-1) MS-DOS/Windows3.1 でのインストレーション (4-1-1) イネープラのインストール

PC カードのイネーブルを行うために、本製品添付のイネーブラのインストールを行う必要があります。PC-AT または互換機でお使いの場合は、DOS/V 版カードサービス対応イネーブラとポイントイネーブラを用意しています。どちらを使用するか選択してください。 PC-9800 シリーズの場合は、PC-9800 シリーズ版カードサービス対応イネーブラを使用します。

#### □ DOS/V版カードサービス対応イネーブラのインストール

添付のフロッピーからハードディスクにカードサービス対応イネーブラをコピーして ください。

C: ¥>COPY A: ¥ENABLER¥DOSV¥REXDIO.EXE C: ¥CARD

REXDIO.EXE はデバイスドライバですので、CONFIG.SYS に登録して使います。

#### □ DOS/V版ポイントイネーブラのインストール

添付のフロッピーからハードディスクにポイントイネーブラをコピーしてください。

C: ¥>COPY A: ¥ENABLER¥DOSV¥D10365.EXE C: ¥CARD

DIO365.EXE は、カード挿入状態で DOS プロンプトから実行します。

#### 日 PC-9800 シリーズ版カードサービス対応イネーブラのインストール

添付フロッピーディスクからハードディスクに PC-9800 シリーズ用カードサービス 対応イネーブラをコピーしてください。

A: ¥>COPY B: ¥ENABLER ¥PC98 ¥REXDI098. EXE A: ¥CARD

REXDIO98.EXE はデバイスドライバですので、CONFIG.SYS に登録して使います。

#### カードイネーブラとは…

パソコンのスロットに挿入した直後はメモリーカードとして認識されており、1/Oカ ードとしての動作はしていません。このメモリーカードの中には、PCカードを1/Oカ ードにコンフィグレーションするために必要な情報(カード属性情報)が書き込まれ ています。

PC カードを I/O カードとして機能させるためには、コンフィグレーションソフト「イネーブラ」が必要となります。イネーブラは、PC カードのカード属性情報を読み込んだ後、その情報に基づいて PC カードを所定の I/O カードにコンフィグレーションします。イネーブラによるコンフィグレーションが正常に行なわれて、はじめて PC カードは I/O カードとして使える状態になります。

(4-1-2) DOS/V 版カードサービス対応イネーブラを使用する場合

最初に、カードサービスのインストールが完了しているか確認してください。次に、 本製品添付のカードサービス対応イネーブラをカードサービスドライバの後に追加し ます。カードサービス対応イネーブラには、下記オプション仕様に従って必要なオプシ ョン情報を記載します。

次頁以降にCONFIG.SYSの登録例を示します。CONFIG.SYSの内容はお使いのパ ソコンにより多少異なることがあります。登録内容については、ご利用されているパソ コン添付のカードサービスマニュアル記載内容に従ってください。

## オプション仕様

DEVICE=C:¥CARD¥REXDIO.EXE [/<オプション>] [] ・・・ []

| オプション        |   |
|--------------|---|
| /SLOTnBASE=x | スロット番号 <n>と I/O ベースアドレス<x>を指定します.</x></n> |
|              | (1)スロット番号は1から4を指定します。パソコン側の PCMCIA ス      |
|              | ロットの順番に対応していますので、予めスロット番号を調べ              |
|              | てお〈必要があります。                               |
|              | 1枚のカードしか使用しないときは、0を指定することもできま             |
|              | す。0が指定された場合はスロットを順に調べて最初に見つか              |
|              | った DIO カードのみをイネーブルします。                    |
|              | (2)スロット n に挿入するカードに割り当てる I/O ベースアドレスを     |
|              | 16進表記で指定します。何も指定しない場合は、300hにアドレ           |
|              | スを割り当ます。                                  |
| /SLOTnIRQ=x  | スロット番号 <n>と割り込み番号<x>を10進表記で指定します。</x></n> |
|              | (1) 何も指定しない場合、割り込みは使用しません。                |
|              | (2) 指正可能は割り込み留ちは、5,7,10,11,12,15 じ9。      |

■ 複数枚のカードを使用する場合の留意点

パソコンの PC カードスロットには、スロット 1・スロット 2 というように スロット番号が付いてます。上記イネーブラオプションで指定する"/SLOTn" は、パソコンで決められているスロット番号に対応しています。複数枚のカー ドの抜き差しを行った場合にカードに割り当てられる I/O アドレス・IRQ 番号 は、オプションで指定されたスロットに対応する I/O アドレス・IRQ 番号にな ります。

# CONFIG.SYS 記述例1 IBM カードサービス PlayAtWill の場合

| DEVICE=C:¥WINDOWS¥EMM386.EXE RAM X=  | C800-CFFF  | (1)  |
|--|--|--|
| DEVICEHIGH=C:¥EZPLAY¥SSDPCIC1.SYS<br>DEVICEHIGH=C:¥EZPLAY¥IBMDOSCS.SYS<br>DEVICEHIGH=C:¥EZPLAY¥RMUDOSAT.SYS /  | SH=1 /NS=1 /MA=C800-CFFF   | (2)<br>(3)<br>(4)  |
| DEVICEHIGH=C:¥EZPLAY¥AUTODRV.SYS   |  | (5)  |
| DEVICE=C:¥CARD¥REXDIO.EXE /Slot1Bas<br>/Slot2Bas   | e=300 /Slot1lrq=5<br>e=304 /Slot2lrq=7   | (6)  |
| [ <b>解説</b> ]<br>(1) 拡張メモリマネージャが[C800~CFFF<br>に指定しています。<br>(2) ソケットサービスを起動しています。ソ<br>択したマシーンにより異ります。<br>(3) カードサービスを起動しています。<br>(4) リソースマップユーティリティに対しか・<br>ウセグメントを使用するように指定して<br>(5) カードサービス標準イネーブラを起動し<br>(6) 本製品添付のカードサービス版イネー<br>い)。スロット1 に挿入されるカードに I<br>ト2 に挿入されるカードに I/O ベースフ<br>定しています。 | F]のメモリウィンドウセグメントを<br>ケットサービスファイル名はイン<br>ードサービスが[C800~CFFF]の<br>います。<br>しています。<br>ーブラを起動しています(1 行で<br>/O ベースアドレス 300h および<br>7ドレス 304h および IRQ7 を割り | 使用しないよう<br>ストール時に選<br>のメモリウィンド<br>記述してくださ<br>IRQ5 を、スロッ<br>当てるように指 |

■ アプリケーションからカード情報を取得する 実際にカードに割り当てられた I/O アドレスと割り込み番号等のコンフィグレーション情報は、カードサービスコール GetConfigurationInfo(ファンクション:04h)を使うことにより取得することができます。DIO ライブラリの 関数 GetConfigurationInfo()を使うことにより、アプリケーションの中から、 この I/O アドレス及び割り込み番号を取得することができます。

# CONFIG.SYS 記述例 2 COMPAQ SystemSoft CardSoft の場合

| DEVICE=C:¥DOS¥EMM386.EXE 1024 X=D000-DFFF<br>DEVICE=C:¥CPQDOS¥SSVLSI.EXE<br>DEVICE=C:¥CPQDOS¥CS.EXE<br>DEVICE=C:¥CPQDOS¥CSALLOC.EXE<br>INSTALL=C:¥CPQDOS¥CARDID.EXE C:¥CPQDOS¥CARDID.INI   | (1)<br>(2)<br>(3)<br>(4)<br>(5) |
|--|---------------------------------|
| DEVICE=C:¥CARD¥REXDIO.EXE /SIot0Base=300   | (6)                             |
| <ul> <li>[解説]</li> <li>(1) 拡張メモリマネージャが[D000~DFFF]のメモリウィンドウセグメントを使用したに指定しています。</li> <li>(2) ソケットサービスを起動しています。</li> <li>(3) カードサービスを起動しています。</li> <li>(4) リソースマネージャを起動しています。</li> <li>(5) カードサービス添付の標準イネーブラを起動しています。</li> <li>(6) 本製品添付のカードサービス版イネーブラを起動しています。</li> <li>1枚のカードしか使用しない時の例になります。挿入スロットを指定しないで、挿えたカードに I/O ベースアドレスを 300h に割り当て、割り込みは使用しません</li> </ul> | ないよう                            |

## CONFIG.SYS 記述例 3 TOSHIBA Phoenix PCM Plus の場合

| DEVICE=C:¥D0S¥EMM386.EXE RAM P0=D000 P1=D400 P2=D800 P3=DC00 I=B000-B7FF X=C800-C8FF  | (1)                 |
|---|---------------------|
| DEVICE=C:¥PCMPLUS3¥CNFIGMAN.EXE /DEFAULT  |                     |
| DEVICE=C: ¥PCMPLUS3¥PCMSS.EXE   | (2)                 |
| DEVICE=C: ¥PCMPLUS3¥PCMCS.EXE   | (3)                 |
| DEVICE=C: ¥PCMPLUS3¥PCMRMAN.SYS   |                     |
| DEVICE=C: ¥PCMPLUS3¥PCMSCD. EXE   | (4)                 |
| •••••   |                     |
| DEVICE=C:¥CARD¥REXDIO.EXE /SIot1Base=300 /SIot1Irq=5  |                     |
| /Slot2Base=304 /Slot2Irq=7  |                     |
| /Slot3Base=308 /Slot3lrq=10   |                     |
| /Slot4Base=30C /Slot4lrq=11   | (5)                 |
| <ul> <li>(1)拡張メモリマネージャーが[C800~C8FF]のメモリウィンドウセグメントを使用しうに指定しています。</li> <li>(2)ソケットサービスを起動しています。</li> <li>(3)カードサービスを起動しています。</li> <li>(4)カードサービス添付の標準イネーブラを起動しています。</li> <li>(5)本製品添付のカードサービス版イネーブラを起動しています(1 行で記述してい)。スロット1から4に挿入されるカードに、それぞれI/Oベースアドレス300h308h·30Chを、割り込みIRQ5·IRQ7·IRQ10·IRQ11を割り当てます。</li> </ul> | ないよ<br>くださ<br>304h・ |

(4-1-3) PC-9800 シリーズ版カードサービス対応イネーブラを使用する場合

最初に、カードサービスのインストールが完了しているか確認してください。次に、 本製品添付のカードサービス対応イネーブラをカードサービスドライバの後に追加し ます。カードサービス対応イネーブラには、下記オプション仕様に従って必要なオプシ ョン情報を記載します。

次頁に CONFIG.SYS の登録例を示します。CONFIG.SYS の内容はお使いのパソコンにより多少異なることがあります。登録内容については、ご利用されているパソコン添付のマニュアル記載内容に従ってください。

#### オプション仕様

DEVICE=A:¥CARD¥REXDI098.EXE [/<オプション>] [] ・・・ []

| オプション        | 解   |
|--------------|---|
| /SLOTnBASE=x | スロット番号 <n>と I/O ベースアドレス<x>を指定します.</x></n> |
|              | (1)スロット番号は1から4を指定します。パソコン側の PCMCIA ス      |
|              | ロットの順番に対応していますので、予めスロット番号を調べ              |
|              | てお〈必要があります。                               |
|              | 1枚のカードしか使用しないときは、0を指定することもできま             |
|              | す。0が指定された場合はスロットを順に調べて最初に見つか              |
|              | った DIO カードのみをイネーブルします。                    |
|              | (2)スロット n に挿入するカードに割り当てる I/O ベースアドレスを     |
|              | 16 進表記で指定します。何も指定しない場合は、ODOh にアド          |
|              | レスを割り当ます。                                 |
| /SLOTnIRQ=x  | スロット番号 <n>と割り込み番号<x>を10進表記で指定します。</x></n> |
|              | (1)何も指定しない場合、割り込みは使用しません。                 |
|              | (2)指定可能な割り込み番号は、3,5,6,10,12,13 です。        |

■ PC-9800 シリーズ版対応カードサービスについて...

PC-9800 シリーズで初期の機種では注 1)のソケットサービスしか提供されておらず、本製 品添付のイネーブラを使ってカードをイネーブルすることはできません。別売版カードサー ビスを入手してください。PC-9800 シリーズ対応カードサービスと搭載機種は下表の通りで す。

| CS バージョン識別名                                       | SS,CS ドライバー名 | 搭載パソコン機種                 |  |  |
|---|--------------|--------------------------|--|--|
| 別売版カードサービス  | SSMECIA.SYS, | PC-9821 Ne               |  |  |
| SystemSoftCardSSoft2.10                           | CS.EXE       | PC-9801 NX/C,P,NS/A,NL/R |  |  |
| Version2.06                                       |              |                          |  |  |
| バンドル版カードサービス                                      | SSDRV.EXE,   | PC-9821 Np,Ns,Ne2,Nd,Ld  |  |  |
| SystemSoftCardSSoft2.10                           | CS.EXE       | Nf,Nm,Lt,Ne3,Nd2         |  |  |
| Version2.06                                       |              | PC-9801 NL/A その他新機種      |  |  |
| 注 1) ソケットサービス NEC SocketService 2.00 Version 1.00 |              |                          |  |  |

# CONFIG.SYS 記述例 4 NEC バンドル版カードサービス SystemSoft CardSoft

PC-9821 Np,Ns,Ne22,Nd,Ld,Nf,Nm,Lt,Ne3,Nd2,La10/7,Na13/12/9,Nb7 等新機種 PC-9801 NL/A

| DEVICE=A: ¥DOS¥EMM386.EXE /UMB /E=DCOO-DFFF   | (1)                                  |
|---|--------------------------------------|
| DEVICE=A: ¥DOS¥SSDRV. SYS   | (2)                                  |
| DEVICE=A: ¥DOS¥CS.EXE   | (3)                                  |
| DEVICE=A: ¥DOS¥CSALLOC. EXE A: ¥DOS¥CSALLOC. INI  | (4)                                  |
| INSTALL=A: ¥DOS¥CARDID.EXE A: ¥DOS¥CARDID.INI   | (5)                                  |
| •••••   |                                      |
| DEVICE=A:¥CARD¥REXDI098.EXE /Slot1Base=D0 /Slot1lrq=3   |                                      |
| /Slot2Base=D4 /Slot2Irq=5   | (6)                                  |
| <ul> <li>(1)拡張メモリマネージャが[DC00~DFFF]のメモリウィンドウセグメントを使用<br/>に指定しています。</li> <li>(2)ソケットサービスを起動しています。</li> <li>(3)カードサービスを起動しています。</li> <li>(4)リソースマネージャを CSALLOC.INI を参照するようにして起動しています。</li> <li>(5)カードサービス添付の標準イネーブラを起動しています。</li> <li>(6)本製品添付の PC-9800 シリーズ版カードサービス対応イネーブラを起動して<br/>行で記述してください)。スロット1 に挿入されるカードに I/O ベースアドレス D4h および IR<br/>当てるように指定しています。</li> </ul> | しないよう<br>こいます(1<br>00h および<br>Q5 を割り |

#### NEC 別売り版カードサービス SystemSoft CardSoft

PC-9821 Ne PC-9801 NX/C,P,NS/A,NL/R

| DEVICE=A:¥DOS¥EMM386.EXE /UMB /E=DA00-DBFF     | (1) |
|--|-----|
| •••••  |     |
| DEVICE=A: ¥DOS¥SSMECIA. SYS                    | (2) |
| DEVICE=A: ¥DOS¥CS.EXE                          | (3) |
| DEVICE=A: ¥DOS¥CSALLOC.EXE A: ¥DOS¥CSALLOC.INI | (4) |
| INSATLL=A: ¥DOS¥CARDID.EXE A: ¥DOS¥CARDID.INI  | (5) |
| •••••  |     |
| DEVICE=A:¥CARD¥REXD1098.EXE /Slot0Base=D0      | (6) |
| [解説]<br>(1)~(5)上記解説参照                          |     |

(6)本製品添付のカードサービス版イネーブラを起動しています。1 枚のカードしか使用し ない時の例になります。挿入スロットを指定しないで、挿入されたカードに I/O ベースア ドレスを D0h に割り当て、割り込みは使用しません。

#### (4-1-4) DOS/V 版ポイントイネーブラを使用する場合

DOS/V でカードサービスが提供されていない機種をご利用されている場合、本製 品に添付されているポイントイネーブラにより DIO カードをイネーブルすることができ ます。また、カードサービス等のドライバをメモリーに常駐させるとコンベンショナルメ モリの空き領域が不足して不都合が生じることがあります。このような場合、カードサ ービスを CONFIG.SYS に登録しないでポイントイネーブラを使ってカードのイネーブル を行うことができます。

ポイントイネーブラは、パソコン本体のメモリウィンドウを通してカードの情報を読み 出します。EMM386.EXE が CONFIG.SYS に組み込まれている場合には、4K バイトのメ モリウィンドウを EMM386.EXE が使用しないように<X=>オプションを追加してください。 下記例は、ポイントイネーブラが使用するメモリウィンドウのセグメントアドレス [DF00-DFFF]を設定する場合の例になります。DOS プロンプトから起動します。

DEVICE=C:¥EMM386.EXE 512 X=DF00-DFFF

## オプション仕様

C:¥>DI0365.EXE [/<オプション>] [] ・・・ []

| オプション        | 解認問題                                      |
|--------------|---|
| /SLOTnBASE=x | スロット番号 <n>と I/O ベースアドレス<x>を指定します.</x></n> |
|              | (1)スロット番号は1から4を指定します。パソコン側の PCMCIA ス      |
|              | ロットの順番に対応していますので、予めスロット番号を調べ              |
|              | てお〈必要があります。                               |
|              | 1枚のカードしか使用しないときは、0 を指定することもできま            |
|              | す。0 が指定された場合はスロットを順に調べて最初に見つか             |
|              | った DIO カードのみをイネーブルします。                    |
|              | (2)スロット n に挿入するカードに割り当てる I/O ベースアドレスを     |
|              | 16 進表記で指定します。 何も指定しない場合は 300h にアドレ        |
|              | スを割り当ます。                                  |
| /SLOTnIRQ=x  | スロット番号 <n>と割り込み番号<x>を10進表記で指定します。</x></n> |
|              | (1)何も指定しない場合、割り込みは使用しません。                 |
|              | (2)指定可能な割り込み番号は 5,7,10,11,12,15 です。       |
| /MEM=x       | イネーブラが使用するメモリーウィンドウセグメントアドレスを指定           |
|              | します。 指定しないときはセグメント DF00h から 4K バイトを使いま    |
|              | ୍ <b>୨</b> .                              |

#### ●<sup>™</sup>注意... ●<sup>™</sup>

PCMCIAコントローラがインテル82365Lまたは互換チップ以外は動作しません。

# カードサービス対応イネーブラとポイントイネーブラ

カードサービス(CS)対応イネーブラは起動された時点で、CSのファンクションセッ トである GetCardServiceInfo により、CS が常駐しているかチェックします。CS が常 駐していれば、イネーブラは CSのファンクションセット RegisterClient により、カード が抜き差しされた時 CS がイネーブラを呼び出すために必要なコールバック情報を 登録しメモリに常駐します。PC カードが挿入または抜き取られると、CS は登録され たコールバック情報をもとに全てのイネーブラに抜き差しの通知を行います。CS は、複数の PC カードが使用する I/O アドレス・IRQ のリソースをリソースマネージメ ントテーブルで管理します。同時に、上記のカード抜き差しの監視を行います。図 で示すようにカードが挿入されるとそれを検出してイネーブラに通知します。イネー ブラは CS からの通知を受けて自分のカードかどうか調べます。自分のカードの時 は、CS に対し必要なI/O アドレスおよび IRQ を割り当ててくれるようにリソースの要 求とイネーブルの要求を発行します。この要求を受けてCS は要求されたリソース が他で使われていなければ、ソケットサービス(SS)と呼ばれる低レベルのファン クションセットを呼び出してリソースを確保しカードのイネーブルを行います。

ポイントイネーブラは、PC Card Interface Controller(PCIC)を直接制御してカード をイネーブルします。カードの抜き差しの管理は行いません。



## (4-2) MS-DOS ライブラリ解説

(4-2-1) MS-DOS ライプラリ

カードサービスを使って REX-5055 DIO PC カードをイネーブルした場合、実際にカ ードに割り当てられている I/O ベースアドレスおよび IRQ 情報をカードサービスに問い 合わせて自動取得するために関数が提供されています。 I/O ベースアドレスおよび IRQ 情報を作成したアプリケーションで取得する必要がある場合は本ライブラリをリン クしてください。

I/O アドレスおよび IRQ 等の資源情報はイネーブラ起動時のオプションで指示して いる内容になります。従って、特にアプリケーション内で自動取得する必要がない場 合は、MS-DOS ライブラリをリンクする必要はありません。

ライブラリモデル

| モデルサイズ  | ライブラリ名      |
|---------|-------------|
| スモールモデル | SLIBDIO.LIB |
| ラージモデル  | LLIBDIO.LIB |
| ヒュージモデル | HLIBDIO.LIB |

インクルードファイル

DIOLIB.H

作成したアプリケーションにインクルードします。

#### CheckCSRegistration

カードサービス常駐確認

- 書式 BOOL CheckCSRegistration (void)
- 機 能 カードサービスが常駐しているか調べます
- 引数 なし

戻 値 カードサービス常駐していれば 0以外の値を返します。

#### GetCSConfigInfo

#### リソース情報の取得

- 書 式 BOOL GetCSConfigInfo (WORD Slot, WORD \*plOAdrs, WORD \*plRQNo)
- 機 能 カードサービスをコールしてカードに割り当てられている I/O アドレス・割り込み番 号を取得します
- 引数 WORD Slot :カードが挿入されているスロットの番号 WORD \*plOAdrs :I/O アドレス格納先を示すポインタ WORD \*plRQNo :割り込み番号格納先を示すポインタ
- 戻値 正常に取得できた場合 0を返します。その他はエラー。

(4-2-2) MS-DOS サンプルプログラム

#### RYO-8との接続

リードリレー出力ユニット(RYO-8)をREX-5055と接続し、順次1つずつリレーを動作 させます。

| REX5055 DIO Card<br><b>外部入出力コネクタ信号名</b> | 機器接続図 | RYO-8<br>外部入出力コネクタ信号名 |
|---|-------|-----------------------|
| + 5V ( 1)                               | ← →   | + 5V (1)              |
| G N D (19)                              | ← →   | GND(12)               |
| PIO7(9)                                 |       | DI-8(11)              |
| PIO6(21)                                |       | DI-7(10)              |
| PIO5(10)                                |       | DI-6(9)               |
| PIO4(22)                                |       | DI-5(8)               |
| PIO3(11)                                |       | DI-4(7)               |
| PIO2(23)                                |       | DI-3(6)               |
| PIO1(12)                                |       | DI-2(5)               |
| PIO0(24)                                |       | DI-1(4)               |

## ⊞ *C 言語プログラム例*

添付ディスク中の[RY08.C]がソースプログラムです。 このプログラムは、各種出力ユニット(PH0-8,SSR-81,SSR-83)等でも同様です。

```
void main( void )
{
   CardInit( IOBase );
   while( !kbhit() ){
       for( BitData = 0; BitData <= 15; BitData++ ){</pre>
           printf( "%3d", BitData );
           Relay( IOBase, BitData, 1 );
           delay(40000);
           Relay( IOBase, BitData, 0 );
      }
   }
void CardInit( WORD adrs )
{
   }
void Relay( WORD adrs, WORD BitData, WORD flg )
{
   BOOL
          d;
   d = inpw( adrs + DRL );
   outpw( adrs + DRL , ( d & ( 1 << BitData ) ) | ( flg << BitData ) );
}
```

#### ▷ N88BASIC 言語プログラム例

添付ディスク中の"RYO-8.BAS" がソースプログラムです。

```
10 'SAVE"ryo-8.bas",A
20 'REX-5055 DIO PC Card & RYO-8
30 '-----
                                     ' Default Port
120 REXPORT=&HD0
                                      ' Controll register (Write Only)
130 \text{ CTRG} = 0
150 DRL = 1
160 DPL
                                    ' INT Enable register ( Write Only )
                                   ' Data register low (Read / Write )
                                     ' Data register high ( Read / Write )
160 \text{ DRH} = 3
170 ADRS$ = ""
180 IOADRS = 0
190 SPEED=2000
200 CLS
210 PRINT"I/Oアドレス値の入力", HEX$( REXPORT )
220 INPUT"アドレス値 ( xxxxh ): "; ADRS$
230 IF LEN( ADRS$ ) > 4 THEN 200
240 IOADRS=VAL( "&h"+ADRS$ )
250 IF IOADRS <> 0 THEN REX5055 = IOADRS ELSE REX5055 = REXPORT
260 OUT REX5055 + CTRG ,&H3 ' mode = low & high output mode
270 OUT REX5055 + INTEBL , 0 ' INT Disable
280 WHILE(INKEY$="")
290 RANP=1
300 FOR BIT = 0 TO 7
       PRINT USING"###";BIT;
310
320
             OUT REX5055+DRL, RANP
330
340
350 NEXT
             FOR I=1 TO SPEED:NEXT
             RANP=RANP+RANP
360 OUT REX5055+DRL,0
370 RANP=1
      FOR BIT = 8 TO 15
380
390
             PRINT USING"###";BIT;
400
              OUT REX5055+DRH, RANP
410
              FOR I=1 TO SPEED:NEXT
420
              RANP=RANP+RANP
430 NEXT
440 OUT REX5055+DRH,0
450 WEND
460 END
```

#### PHI-8との接続

フォトカプラ入力ユニット(PHI-8)をREX-5055と接続し、ON-OFFの状態を読み込み ます。

| REX5055 DIO Card<br><b>外部入出力コネクタ信号名</b> | 機器接続図        | PHI-8<br>外部入出力コネクタ信号名 |
|---|--------------|-----------------------|
| + 5V( 1)                                | $\leftarrow$ | +5V                   |
| GND (19)                                | ← →          | GND                   |
| PIO7(9)                                 | ←───         | DO-1                  |
| PIO6(21)                                | ←───         | DO-2                  |
| PIO5(10)                                | <b>←</b>     | DO-3                  |
| PIO4(22)                                |              | DO-4                  |
| PIO3(11)                                |              | DO-5                  |
| PIO2(23)                                | ·            | DO-6                  |
| PIO1(12)                                |              | DO-7                  |
| PIO0(24)                                |              | DO-8                  |

#### 🖽 C 言語プログラム例

添付ディスク中の[PHI8.C]がソースプログラムです。 このプログラムは、各種入力ユニット(PHI-8,SSR-82,SSR-84)等でも同様です。

```
void main( void )
{
/* --- 途中省略 --- */
MAIN_PROG:
   CardInit( IOBase );
   while( !kbhit() ) {
       for( port = 0; port <= 1; port++ ) {</pre>
          data = (WORD)~_inp( IOBase + port + 2 );
                                                             /* Data read */
          printf( " Address %04X -- Data %02X ",IOBase + port + 2, data );
          delay( 40000 );
                                                              /* delay */
       printf( "¥n" );
   }
}
void CardInit( WORD Adrs )
{
   outp(Adrs + CTRG, 0); /* コントロールレジスタ Base + 0 */
                              /*
                                  IMD:0 BSEL3:0 BSEL2:0 BSEL1:0 */
                              /* BSEL0:0
                                                 DIRH :0 DIRL :0 */
   outp( Adrs + INTEBL , 0 ); /* 割り込みディセーブル */
}
```

#### マトリクス入力回路

フォトカプラ入力ユニット(PHI-8)とフォトカプラ出力ユニット(PHO-16)で、マトリクスを 構成します。マトリクス上の、合計 128 点の接点入力の状態をスキャンして調べます。 機器の稼働状態のモニタや、ドアの開閉状態のモニタ等に、幅広く用いることがで きます。



| REX5055 DIO Card<br><b>外部入出力コネクタ信号名</b> | 機器接続図 | PHO-16<br><b>外部入出力コネクタ信号名</b> |
|---|-------|-------------------------------|
| + 5V (7)                                | ← →   | + 5V (1)                      |
| G N D (19)                              | ← →   | GND(12)                       |
| PIO7(9)                                 |       | NC (4)                        |
| PIO6(21)                                |       | NC (5)                        |
| PIO5(10)                                |       | NC (6)                        |
| PIO4(22)                                |       | NC (7)                        |
| PIO3(11)                                |       | D ( 8)                        |
| PIO2(23)                                |       | C ( 9)                        |
| PIO1(12)                                |       | B (10)                        |
| PIO0(24)                                |       | A (11)                        |

| REX5055 DIO Card<br><b>外部入出力コネクタ信号名</b> | 機器接続図        | PHI-8<br>外部入出力コネクタ信号名 |
|---|--------------|-----------------------|
| + 5 V ( 1 )                             | ← →          | + 5 V ( 1 )           |
| G N D (13)                              | $\leftarrow$ | G N D (12)            |
| PIO15(3)                                |              | DO - 8(11)            |
| PIO14(15)                               |              | DO - 7(10)            |
| PIO13(4)                                |              | DO - 6(9)             |
| PIO12(16)                               |              | DO - 5(8)             |
| PIO11(5)                                |              | DO - 4 ( 7)           |
| PIO10(17)                               |              | DO - 3(6)             |
| PIO9 (6)                                |              | DO - 2(5)             |
| PIO8 (18)                               |              | DO - 1 ( 4)           |

(注)PHI-8/PHO-16 のネジターミナルから、接点までのケーブルは約 1000m までの 延長が可能です。アイソレート電源の GND は、パソコン本体の GND と接続しないでく ださい。

## 🖽 C 言語プログラム例

添付ディスク中の[MATRIX.C]がソースプログラムです。

```
void main( void )
{
   CardInit( IOBase );
   printf( "¥x0c" );
   while( !kbhit() ) {
       printf( "¥x0b" );
           for( rowad = 0; rowad <= 15; rowad++ ){</pre>
           for( no = 0; no <= 7; no++ ) {</pre>
               printf( "%3d-", rowad * 8 + no + 1 );
               if( swdt & ( 1 << no ) )
                   printf( "OFF " );
               else
                   printf( "ON " );
           }
       }
   }
}
void CardInit( WORD Adrs )
{
   outp ( Adrs + CTRG, 1 ); /* low byte = output high byte = input mode */
outp ( Adrs + INTEBL , 0 ); /* INT Disable */
}
```

#### アスキーキャラクタの読み込み

外部からのデータストローブ入力を PIO7 で受け、PIO8 からアクノリッジを出力し、 ポート下位7ビットをデータ入力として使用します。

| REX5055 DIO Card<br>外部入出力コネクタ信号名 | 機器接続図                                 | 外部機器   |
|----------------------------------|---------------------------------------|--------|
| G N D (19)                       | ← →                                   | G N D  |
| PIO8(18)                         | · · · · · · · · · · · · · · · · · · · | アクノリッジ |
| PIO7(9)                          | ◀                                     | ストローブ  |
| PIO6(21)                         | L                                     | b6 データ |
| PIO5(10)                         | ·                                     | b5 データ |
| PIO4(22)                         | ·                                     | b4 データ |
| PIO3(11)                         | le                                    | b3 データ |
| PIO2(23)                         | l ————                                | b2 データ |
| PIO1(12)                         | <b></b>                               | b1 データ |
| PIO0(24)                         | ·                                     | b0 データ |

#### ■ C 言語プログラム例

添付ディスク中の[ASCIN.C]がソースプログラムです。

```
void main( void )
{
   CardInit( IOBase );
   while( !kbhit() ){
      if( inp( IOBase + DRL ) >> 7 == 1 ) /* strobe in ? */
          continue;
      c = ~(inp( IOBase + DRL ));
outp ( IOBase + DRH , 0x01 );
                                          /* data read */
                                          /* ack out */
      putchar( c & 0x7f );
                                           /* すぐに続けるとデータを */
      for( i=0; i<1000000; i++ )
                                           /* とり続けてしまうため */
         ;
                                           /* Strobe off を待つ時間*/
   }
}
void CardInit( WORD Adrs)
{
   outp( Adrs + CTRG , 0x02 ); /* Low Byte input mode / High Byte output mode */
   }
```

# アスキーキャラクタの出力

PIO7 を外部へのストローブ出力、PIO8 をアクノリッジ入力、ポート B の下位7ビット をデータ出力として使用します。

| REX5055 DIO Card<br><b>外部入出力コネクタ信号名</b> | 機器接続図   | 外部機器     |
|---|---------|----------|
| G N D (19)                              | ← →     | G N D    |
| PIO8(18)                                | <b></b> | - アクノリッジ |
| PIO7(9)                                 |         | ストローブ    |
| PIO6(21)                                |         | b6 データ   |
| PIO5(10)                                |         | b5 データ   |
| PIO4(22)                                |         | b4 データ   |
| PIO3(11)                                |         | b3 データ   |
| PIO2(23)                                |         | b2 データ   |
| PIO1(12)                                |         | b1 データ   |
| PIO0(24)                                |         | b0 データ   |

#### 🖽 C 言語プログラム例

添付ディスク中の[ASCOUT.C]がソースプログラムです。

```
void main( void )
{
   CardInit( IOBase );
   while( !kbhit() ){
       s = string;
       while( *s ){
          outp(IOBase + DRL, *s); /* データ出力 */
outp(IOBase + DRL, *s | 0x80); /* ストローブ出力 */
           for( i=0L; i<100000L; i++ )</pre>
          putchar( *s++ );
          outp( IOBase + DRL, 0 );
                                             /* ストローブをOff */
       }
   }
}
void CardInit( WORD Adrs)
{
   outp (Adrs + CTRG, 0x01); /* Low Byte output mode / High Byte input mode */
   outp ( Adrs + INTEBL , 0 ); /* INT Disable */
}
```

## 割り込み処理

PIO7 を割り込み入力端子に設定し、PIO7 に方形波のパルス信号を入力します。サンプルプログラムは、PIO7 のパルス立ち上がりエッジで割り込みをトリガし、割り込み回数をカウント表示します。

| REX5055 DIO Card<br>外部入出力コネクタ信号名 | 機器接続図 | 外部機器   |
|----------------------------------|-------|--------|
| G N D (19)                       | ← →   | G N D  |
| PIO8(18)                         |       | NC     |
| PIO7(9)                          | ←     | パルス方形波 |
| PIO6(21)                         |       | NC     |
| PIO5(10)                         |       | NC     |
| PIO4(22)                         |       | NC     |
| PIO3(11)                         |       | NC     |
| PIO2(23)                         |       | NC     |
| PIO1(12)                         |       | NC     |
| PIO0(24)                         |       | NC     |

# 🖽 C 言語プログラム例

添付ディスク中の[INT.C]がソースプログラムです。割り込み処理をコーディ ングする場合、PC-ATとPC-9800シリーズでは割り込みコントローラのアドレス 及び割り込みベクタテーブルの内容が異なります。添付のサンプルプログラム はどちらの機種にも対応していますが、プログラムの先頭で MACHINE\_DOSV/MACHINE\_PC98のどちらを有効にするか定義してください。サンプ ルプログラムはPC-ATが有効になっています。

```
/*
*
       機種定義
              MACHINE PC98
#define
                              1
*/
#define
             MACHINE_DOSV
                           1
void main( void )
{
                                             /* I/0 アドレス 300h 使用 */
   IOBase = 0x300;
   wIRQNo = 7;
                                             /* 割込み番号 INT7 使用 */
   if ( SetIntNo( wIRQNo ) != 0 ) {
       printf("IRQ Number Error¥n");
       exit(0);
   }
  /* --- 次ページに続く --- */
```

```
/* 割り込みによるデータ入力を開始する */
   StartIrq();
   IrqStatus = IRQMODE_RUN;
   printf("Start Interrupt Test¥n");
   /* 入力完了まで待つ */
   switch( IrqLoopProc() )
   {
              printf("ESC Key Stop¥n");
   case 1:
                                          break;
   case 0:
              printf("Normally Stop¥n");
                                          break;
   }
}
void StartIrq( void )
{
   /* コントロールレジスタの設定 */
   /* PIO 0-7 ->入力; PIO 8-15 ->出力; 割込みソース->PIO7 ;立上りエッジで割込み */
   SetControlReg(0, 1, 7, 1);
   /* 割り込みハンドラの登録 */
   IrqSetup();
   /* 割り込み許可ビットをONにする */
   IrqEnable( CLEAR_R_LOW );
}
BOOL IrqSetup( void )
{
   WORD VallMR, MaskBit;
   /* 旧割り込み処理ハンドラのアドレスを待避 */
   OldFunc = dos getvect( drv irg vect table[tbl int no] );
   /* 新割り込み処理ハンドラのアドレスを設定 */
   _dos_setvect( drv_irq_vect_table[tbl_int_no], HandlerFunc );
   /* PICにコマンドを書き込む */
                                            /*INT9 以上ならスレーブ PIC */
   if ( tbl_int_no > 1 ) {
       ValIMR = _inp( SlvIMRAddr );
                                            /* Read IMR of slave PIC */
       MaskBit = ( WORD )(0xff^_drv_irq_mask_table[ tbl_int_no ]);
       ValIMR = ValIMR & MaskBit;
       _outp( SIvIMRAddr, (BYTE)ValIMR );
                                          /* Write IMR of slave PIC */
                                           /* マスタ PIC */
   } else {
                                           /* Read IMR of master PIC */
       ValIMR = inp( MstIMRAddr );
       MaskBit = (WORD)(0xff^_drv_irq_mask_table[ tbl_int_no ]);
       ValIMR = ValIMR & MaskBit;
       _outp( MstIMRAddr, (BYTE)ValIMR ); /* Write IMR of master PIC */
   }
   return 0;
}
  /* --- 次ページに続く --- */
```

```
void _interrupt _far HandlerFunc( void )
{
   _disable();
   /* データの転送 */
   ReadVal = _inp(10Base + 0x02);
   IrqStat = _inp( IOBase + 0x01 );
   wCount++;
   /* EOIを発行 */
   SendEOI();
   /* 割り込み処理中に次の割り込みがある場合、ここでは故意に中止します。 */
   if ( ( _inp( IOBase + 0x01 ) & 0x80 ) == 0x80 ){
       IrqStatus = IRQMODE_STOP;
   }
   _enable();
}
BOOL IrqLoopProc( void )
{
   /* 割込処理ループ */
   for( ;; ){
       if( KeyIn() == KY_ESC ) {
               Stoplrq();
               return 1;
       }
       /* ステータスチェック
                               */
       switch( IrqStatus )
       {
       case IRQMODE_STOP:
       /* 割り込み処理中止 */
               StopIrq();
               return 0;
       case IRQMODE_RUN:
               if( wCount != wCountOld ) {
                       wCountOld = wCount;
                       printf("%lu[%x:%x] ¥t",wCount, ReadVal, IrqStat);
               }
       break;
       }
   }
}
```

(4-3) Windows3.1 16Bit ライブラリ解説

(4-3-1) Windows3.1 アプリケーション

#### Visual C による作成

Windows3.1 環境で Microsoft Visual C でアプリケーションを作成する場合、 Visual C では、DIO カードへの入出力を行うために必要な\_inp()/\_outp()等のポート入出力関数はサポートされていません。本製品添付の DLL は FD:¥LIBRARY¥WIN31¥DLL16 に収録されており、 DIO カードに入出力を行うための関数とカードサービスに関する関数を提供しています。

Visual C/C++でアプリケーションを作成する場合は、DLL からイクスポートされた 関数を.DEF ファイルでインポート宣言すると同時に、下記のように作成したプログラム のヘッダー部でインポート関数の宣言を行ってください。

DLL は、プログラム実行時のカレントディレクトリにコピーしてください。カレントディ レクトリから DLL のロードができない場合は、WINDOWS¥SYSTEM ディレクトリにコピ ーしてください。

DLL ライブラリ名 : DIOLIB.DLL(16Bit Version)

| DE DE | Fファ | イルの | 作成例 | (Visual | C) |
|-------|-----|-----|-----|---------|----|
|-------|-----|-----|-----|---------|----|

| NAME<br>DESCRIPTION<br>EXETYPE<br>STUB<br>CODE<br>DATA<br>HEAPSIZE<br>IMPORTS | REX5055<br>'REX5055 DIO PC Card Sample Program (C)RATOC System, Inc.'<br>WINDOWS<br>'WINSTUB.EXE'<br>PRELOAD MOVEABLE DISCARDABLE<br>PRELOAD MOVEABLE MULTIPLE<br>1024<br>DIOLIB.OutPort<br>DIOLIB.WOUPOrt<br>DIOLIB.InPort |
|---|---|
|   | DIOLIB. InPort  |
|   | DIOLIB.CheckCSRegistration  |
|   | DIOLIB.GetCSConfigInf   |

## アプリケーションでのインポート宣言 (Visual C)

| WORD $\_export$ | WINAPI | OutPort( WORD, WORD );                              |
|-----------------|--------|---|
| WORD $\_export$ | WINAPI | wOutPort( WORD, WORD );                             |
| WORD $\_export$ | WINAPI | InPort( WORD );                                     |
| WORD $\_export$ | WINAPI | wInPort( WORD );                                    |
| BOOL _export    | WINAPI | CheckCSRegistration( void );                        |
| BOOL $\_export$ | WINAPI | <pre>GetCSConfigInfo( WORD, WORD *, WORD * );</pre> |

# ## サンプルプログラム実行例

フォトカプラ出力ユニットPHO-8と接続し、LEDをオンオフ制御を行うサンプルプロ グラムになります。実行時の画面は下記のようになります。



#### Windows3.1 での割り込み処理 添付ディスクの FD:¥LIBRARY¥WIN31¥HANDLER の中に、MS-DOS の割り込みサ ンプルプログラムを Windows3.1 上に移植したサンプルプログラム WINIRQ.C が収録 されています。WINIRQ.EXE は、FD:¥LIBRARY¥WIN31¥HANDLER¥DLL の中の IRQLIB.DLL をロードします。割り込みハンドラ内の処理は、DLL ライブラリの中に記 述してあります。必要な割り込み処理を記述して再構築してください。 DIOLIB.DLLとWINIRQ.DLLとは、下記の内容が相違しています。 DEF ファイル PRELOAD FIXED - コードの移動・破棄は不可を指定 CODE PRELOAD SINGLE DATA HEAPSIZE 512 EXPORTS WEP SetCardResource Stoplrq 割り込み用関数をイクスポート StartIrg SetISRWindow WinGetVect/WinSetVect 関数をサポート WinGetVect(dos getvect 相当の関数) 書式 void (\_cdecl \_interrupt \_far \* \_cdecl WinGetVect( BYTE VectNo ))() 機能 プロテクトモード割込みベクタの取得 引数 VectNo: 割込みベクタ番号 戻 値 割込みベクタ VectNo の far ポインタ WinSetVect(dos setvect 相当の関数) 書式 BOOL cdecl WinSetVect( BYTE VectNo, void ( cdecl interrupt far \*IsrAdrs)() ) 機能 プロテクトモードの割込ベクタ設定 引 数 : ベクター番号 BYTE VectNo void (\_cdecl \_interrupt \_far \*lsrAdrs)(): 割込ハンドラのエントリポイント 戻 値 0:成功 -1:失敗

## Visual BASIC による作成

Visual BASIC では、DIO カードへの入出力を行うためのポート入出力命令 IN/OUT はサポートされていません。Visual BASIC でプログラムを作成する場合も Visual C と同じように DLL ライブラリでサポートしているポート入出力命令を呼び出します。

Visual BASIC から DLL ライブラリの API を呼び出すためには、コードモジュールファ イルの中で DLL 外部関数の参照宣言を行う必要があります。DLL 関数の参照宣言例 を下記に示します。詳細は、添付ディスクの FD:¥LIBRARY¥WIN31¥VB4 に収録されて います。尚、Visual BASIC は Version4.0 16Bit Version を使用しています。

# *細モジュール定義ファイル* Declare 宣言例 (Visual BASIC)

'DLL 外部関数の参照宣言 Declare Function OutPort Lib "diolib.dll" (ByVal IOAddr As Integer, ByVal OutVal As Integer) As Integer Declare Function wOutPort Lib "diolib.dll" (ByVal IOAddr As Integer, ByVal OutVal As Integer) As Integer Declare Function InPort Lib "diolib.dll" (ByVal IOAddr As Integer) As Integer Declare Function wInPort Lib "diolib.dll" (ByVal IOAddr As Integer) As Integer Declare Function wInPort Lib "diolib.dll" (ByVal IOAddr As Integer) As Integer Declare Function GetCSConfigInfo Lib "diolib.dll" (ByVal Socket As Integer, IpCsvIOAddr As Any, IpCsvIRQNo As Any) As Integer Declare Function CheckCSRegistration Lib "diolib.dll" () As Integer

# 細 サンプルプログラム実行例

フォトカプラ出力ユニット PHO-8 と接続し、LED をオンオフ制御を行うサンプルプログラムになります。実行時の画面は下記のようになります。



# (4-3-2) 16Bit DLL **関数仕様**

| OutPort |        |  | ポートに1パイトを出力         |  |
|---------|--------|--|---------------------|--|
| 書       | 式      | WORD _export WINAPI OutPort( WORD IOAddr, WOR                                  | RD OutVal)          |  |
| 機       | 能      | 1バイトをポートに出力  |                     |  |
| 引       | 数      | WORD <b>IOAddr</b> : 出力する I/O ポートアドレス<br>WORD <b>OutVal</b> : バイト出力値(上位バイトは無視) | されます)               |  |
| 戻       | 値      | バイト出力値をそのまま返し、エラー値はなし  |                     |  |
| W       | /OutPo | rt   | ポートに1ワードを出力         |  |
| 書       | 式      | WORD _export WINAPI wOutPort( WORD IOAddr, WC                                  | ORD <b>OutVal</b> ) |  |
| 機       | 能      | 1ワードをポートに出力  |                     |  |
| 引       | 数      | WORD <b>IOAddr</b> : 出力する I/O ポートアドレス<br>WORD <b>OutVal</b> : ワード出力値           |                     |  |
| 戻       | 値      | ワード出力値をそのまま返し、エラー値はなし  |                     |  |
| I       | nPort  |  | ポートから1バイト入力         |  |
| 書       | 式      | WORD _export WINAPI InPort(WORD IOAddr)  |                     |  |
| 機       | 能      | ポートから1バイト読み込む  |                     |  |
| 引       | 数      | WORD <b>IOAddr</b> : 出力する I/O ポートアドレス  |                     |  |
| 戻       | 値      | ポートから読み込んだバイトデータ   |                     |  |
| wInPort |        |  | ポートから1ワード入力         |  |
| 書       | 式      | WORD _export WINAPI wInPort ( WORD IOAddr )                                    |                     |  |
| 機       | 能      | ポートから1ワード読み込む  |                     |  |
| 引       | 数      | WORD <b>IOAddr</b> : 出力する I/O ポートアドレス  |                     |  |

戻 値 ポートから読み込んだワードデータ

CheckCSRegistration

カードサービス常駐チェック

- 書式 BOOL \_export WINAPI CheckCSRegistration (void)
- 機能 カードサービスが常駐しているか調べます
- 引数 なし
- 戻値 カードサービス常駐していれば 0 以外の値を返します。

 GetCSConfigInfo
 リソース情報の取得

- 書 式 BOOL \_export WINAPI GetCSConfigInfo (WORD Slot, WORD \*pAdrs, WORD \*pIRQ)
- 機 能 カードサービスをコールしてカードに割り当てられている I/O アドレス・割り込み番 号を取得する
- 引数 WORD Slot : カードが挿入されているスロットの番号
   WORD \*pAdrs : I/O アドレス格納先を示すポインター
   WORD \*pIRQ : 割り込み番号格納先を示すポインター
- 戻 値 正常に取得できた場合 0を返します。その他はエラー。

#### ●<sup>™</sup>*注意...* ●<sup>™</sup>

アプリケーション実行時、DLL ファイルは WINDOWS¥SYSTEM ディレクトリに置いてください。通常アプリケーション実行ディレクトリに置いてもロードされますが、ディレクトリの階層が深いとロードされないことがあります。

2.Windows3.1 ではハードウェアを直接操作する I/O は基本的には禁止されています。従っ て、Microsoft Visual C/Visual BASIC についても Windows 上では、\_inp()/\_outp()等の API をサーポートされていません(実際はインプリメントされている)。しかしながら、 Windows システム動作に関係しない外部機器との通信機器との通信制御の用途に限っ てはハードウェアを直接操作しても問題無いと考え、DLL ライブラリの中で\_inp()/\_outp() 相当の API をサポートしました。





# (5-1) インストレーション

WindowsNT 上で REX5055 を使用した計測システムを接続する場合は、本製品添付のセットアッププログラム (INISETUP.EXE)を実行します。

# セットアッププログラムの実行

本製品に含まれるセットアッププログラム(INISETUP.EXE)を実行すると、下記のような画面が表れます。使用されるされるカードを選択し[OK]を押してください。

| REX PCCARD セットアップ 🗙                |  |  |  |  |  |
|------------------------------------|--|--|--|--|--|
| PCカードを Windows NTで使用する為には          |  |  |  |  |  |
| カードの情報を レジストリに登録する必要があります。         |  |  |  |  |  |
| [ PCカード名:] の所を、ご購入されたPCカード名をご選択頂き、 |  |  |  |  |  |
| 最後に[OK]を押して下さい。                    |  |  |  |  |  |
| PC力ード名: REX5055 💌                  |  |  |  |  |  |
| OK 終了                              |  |  |  |  |  |

実行画面

このセットアッププログラムにより指定されたカードのドライバの登録と、ドライ バ、DLL(ダイナミックライブラリ)のコピーが行われます。 ドライバは ¥WinNT¥System32¥Drivers DLL は¥WinNT¥System32 ヘコピーされます。

以上で設定完了です。WindowsNTを再起動し、REX5055が使用可能な状態になったかどうかを確認します。

#### 再起動後の確認

再起動後リソースの取得が正常に行われていればセットアップは正常に行われて います。(下図参照)

| PC カート (PCMCIA) デバイス                       |   | <u>? ×</u>            |  |  |
|--|---|-----------------------|--|--|
| ソケットの状態 コントローラ                             |   |                       |  |  |
| PC カードおよびバクット                              | ま次のとおりです。   |                       |  |  |
| RATOC System, Inc. REX3                    | 056 RS232C PC Card - 77ット0<br>055 DIO PC Card - 75ット1 |                       |  |  |
|  | RATOC System,Inc. REX5055                             | DIO PC Card プロパティ 🏾 🏋 |  |  |
| 7℃11、7~11、10000000000000000000000000000000 | カート・情報 トライバ リソース                                      | - ここをクリックしてください       |  |  |
| 注 PC かードの挿入または取り                           | RATOC System, Inc. REX                                | 5055 DIO PC Card      |  |  |
|  | デバイスの種類 (不明   |                       |  |  |
| ここをクリックしてください                              | 製造元 RATOC Syste                                       | em,Inc.               |  |  |
|  | デバイス マッブ 利用不可   |                       |  |  |
|  | 「 デバイスの状態―――  |                       |  |  |
|  | デバイスは正常に動作しています。                                      | ,                     |  |  |
| DATOC Swater Inc. DEVE                     |   |                       |  |  |
| h-k/Hall k-/ W-z                           |   |                       |  |  |
|  |   |                       |  |  |
| RATOC System, Inc. F                       | RATOC System, Inc. REX5055 DIO PC Card                |                       |  |  |
|  |   |                       |  |  |
| リアーズの設定                                    | = 1   |                       |  |  |
| 1975人の種類                                   | <u>該定</u><br>11                                       |                       |  |  |
| ₩1/0 範囲                                    | 300-304   |                       |  |  |
|  |   |                       |  |  |
|  |   |                       |  |  |
|  |   |                       |  |  |
|  | ОК <b>+</b> +Уセル                                      |                       |  |  |

これらの画面はコントロールパネルの[PC カード]を選択することにより画面に表示されます。

リソースが取得できていれば、セットアップ完了です。また、コントロールパネルの [デバイス]を起動することにより、Windows NT で現在使用できるドライバの一覧 が表示されます。そこに REX-5055 のドライバも再起動により表示されます。 ドライバのロード,アンロード等の設定を行う場合はここで設定を行います。 (特に指定のない場合は何も行う必要はありません)

# (5-2) Visual C によるアプリケーション開発

Visual C 4.0 以上を使って REX-5055 カードを制御するアプリケーションを開発する 場合は、本製品に添付されている 32Bit 版 DLL の関数をコールする必要があります。

Visual C で作成したアプリケーションプログラムから 32Bit 版 DLL を呼び出すため には、下記 3 点の作業を行ってください。

- セットアッププログラムを実行しリソースのレジストリ登録及びドライバのインストー ルを行う(「(5-1)インストレーション参照」)
- 2. アプリケーションプログラムに REX5055.H ファイルをインクルードする。
- 3. アプリケーションプログラムのプロジェクトファイルに 5055LIB.LIB を追加する。

## アプリケーション作成上のアドバイス

I/O ポートへの入出力

WindowsNT の保護機能によりアプリケーションレベルでは I/O への入出力は 行えま

せん。従って、PC カードへの I/O 入出力は DLL でサポートされている OutPort()・wOutPort()・InPort()・wInPort()を使って行います。

割り込み制御

DLL では割り込みサービスを提供しています。 割り込みサービスには、

(1)割り込みハンドラからユーザ定義メッセージにより割り込み通知を受け取る 方法

(2)割り込みハンドラ内で全ての処理を行う方法

があります。高速な処理が要求されるような場合は、(2)の方法で行う必要があります。

サンプルプログラムは高速割り込み処理の例になりますが、割り込み処理内で 要求される内容はユーザー毎で変わります。サポートセンターでは高速割り込み 処理のカスタマイズに関するご相談も承っています。

#### (5-2-1) DLL 関数仕様

| Fastl | ntStart   |                                 | ドライバ内部処理による高速割り込み開始   |  |  |  |
|-------|---|---------------------------------|---|--|--|--|
| 書 式   | void <i>FastIntStart</i> (HWND hWnd, UCHAR InPortReg, UCHAR OutPortReg, ULONG Count, USHORT Flg)  |                                 |   |  |  |  |
| 機能    | 割り込みハンドラ内部で全ての処理を行う高速版の割り込み処理を開始します。<br>割り込みハンドラ内部では、入力方向に設定されたデータレジスタの値をリードし、<br>出力方向に設定されたデータレジスタにリードした値を出力します。割り込み終了回<br>数Countで指定した回数の割り込み処理が終了すると、呼び出し側プログラムに<br>終了メッセージをポストします。 |                                 |   |  |  |  |
| 引 数   | HWND<br>UCHAR<br>UCHAR  | hWnd<br>InPortReg<br>OutPortReg | <ul> <li>ユーザアプリケーションのウィンドウハンドル</li> <li>入力データレジスタ(オフセット)</li> <li>2:PA7-PA0 3:PA15-PA8</li> <li>出力データレジスタ(オフセット)</li> <li>2:PA7-PA0 3:PA15-PA8</li> </ul> |  |  |  |
|       | ULONG<br>USHORT   | Count<br>Flg                    | <ul> <li>▶ 割り込み終了回数</li> <li>▶ 割り込み設定</li> <li>0: InPortReg読み込みのみ (割り込みフラグクリアのみ)</li> <li>1: InPortRegをOutPortRegに出力</li> </ul>                           |  |  |  |
| 戻 値   | なし  |                                 | 1. In orthoge out orthogic II/J   |  |  |  |

MsgIntStart

ユーザ定義メッセージ同期割り込み開始

- 書式 void *MsgIntStart* (HWND hWnd, WORD Count, WORD Flg)
- 機能 ユーザー定義メッセージ版の割り込み処理を開始します。割り込み発生に同期し たユーザー定義メッセージをユーザー側アプリケーションに毎回ポストします。これ によりユーザーアプリケーション側で割り込み発生に同期した処理を行うことが出 来ます。

| 引 数 | HWND | hWnd  | ▶ ユーザアプリケーションのウィンドウハンドル        |
|-----|------|-------|--------------------------------|
|     | WORD | Count | ▶ 割り込み終了回数                     |
|     | WORD | Flg   | ▶ 割り込み設定                       |
|     |      |       | 1以外: 割り込みハンドラ内でメッセージ通知のみを      |
|     |      |       | 行う                             |
|     |      |       | 1: 割り込みハンドラ内でデータポート(16ビット)をリード |
|     |      |       | しメッセージの第一引数(wParam)に出力         |

戻値 Flg が1の時のみ wParam にデータレジスタの内容を返します。

# アプリケーション作成上のアドバイス

WindowsNT ではシステムのオーバーヘッド等により *MsgIntStart* を使用した場合、50Hz, *FastIntStart* を使用した場合、100Hz よりも低速の割込み信号を想定しております。 (PentiumPro200MHz 32MBの場合) 但し、割り込み処理は、処理内容,機種等に依存します。 IntSet

コントロールレジスタ、割込みマスクレジスタの設定

書式 void IntSet (UCHAR CtrlReg, UCHAR IntMask)

機能 コントロールレジスタ、IrEblビットを除く割込みマスクレジスタの設定を行います。
 ここで設定される内容はドライバ内で保持され割り込み処理ルーチン内で使用されます。
 ( コントロールレジスタの制御でOutPort関数を使用しないでください)

- 引数 UCHAR CtrlReg > コントロールレジスタ値 UCHAR IntMask > 割込みマスクレジスタ値
- 戻値 なし

*IODrvResource* 

1/0 ベースアドレスの取得

- 書式 long IODrvResource (void)
- 機能レジストリ内に登録されているI/Oベースアドレスの取得を行います。 (但し、DLLでドライバーのオープンに失敗した場合正しい取得を行えません。)
- 引数 なし
- 戻値 I/O ベースアドレスを返します。

#### **IRQDrvResource**

IRQ 番号の取得

- 書式 long *IRQDrvResource* (void)
- 機能 レジストリ内に登録されているIRQ番号の取得を行います。 (但し、DLLでドライバーのオープンに失敗した場合正しい取得を行えません)
- 引数 なし
- 戻 値 IRQ 番号を返します。

#### **OutPort**

1バイトをポート出力

- 書式 WORD OutPort(WORD IOAddr, WORD OutVal)
- 機能 1バイトをポートに出力
- 引数 WORD IOAddr > ポート番号 WORD OutVal > バイト出力値(上位バイトは無視)
- 戻値 バイト出力値をそのまま返し、エラー値はなし。

wOutPort

1ワードをポート出力

- 書式 WORD wOutPort(WORD IOAddr, WORD OutVal)
- 機能 1ワードをポートに出力
- 引数 WORD IOAddr > ポート番号 WORD OutVal > ワード出力値
- 戻 値 ワード出力値をそのまま返し、エラー値はなし。

## InPort

1バイトをポート入力

- 書式 WORD InPort(WORD IOAddr)
- 機能 ポートから1バイト読み込む
- 引数 WORD IOAddr > ポート番号
- 戻値 ポートから読み込んだバイトデータを返します。
wInPort

1ワードをポート入力

- 書式 WORD winPort(WORD IOAddr)
- 機能 ポートから1ワード読み込む
- 引数 WORD IOAddr > ポート番号
- 戻値 ポートから読み込んだワードデータを返します。

#### TerminateInterrupt

メッセージ割り込み処理の中断

- 書式 void *TerminateInterrupt*(void)
- 機能 割り込み処理の中断
- 引数 なし
- 戻値 なし

#### アプリケーション作成上のアドバイス

*MsgInterrupt* は中断処理を実行することができますが、*FastIntStart* は入力した回数の 割込みが起るまで処理をユーザー側には返すことができません。 *FastIntStart*を使用される際にはご注意〈ださい。 (5-2-2) Visual C サンプルプログラム

アイソレーションユニット PHO-8 または PHI-8 を使った、サンプルプログラム について説明します。

PHO-8 は右図に示す LED 点灯回路と アイソレーションを行って外部へ出 力する回路を 8 ブロック実装してお り、下表のように接続することにより オンオフ制御のシミュレーションを 行うことができます。 PHI-8 は PHO-8 とは反対に外部オンオフ信号をアイ ソレーションして入力するためのボ ードです。



| REX5055 DIO Card<br>外部入出力コネクタ信号名 | 機器接続図      | PHI-8<br>外部入出力コネクタ信号名 |
|----------------------------------|------------|-----------------------|
| +5V                              | <b>←</b> → | +5V                   |
| GND                              | ← →        | GND                   |
| PA7                              |            | DO-1                  |
| PA6                              |            | DO-2                  |
| PA5                              |            | DO-3                  |
| PA4                              |            | DO-4                  |
| PA3                              |            | DO-5                  |
| PA2                              |            | DO-6                  |
| PA1                              |            | DO-7                  |
| PA0                              |            | DO-8                  |

サンプルプログラムは、下記のモジュールで構成されています。

PHO-8 LED オンオフ制御プログラム

PHI-8 入力プログラム(割り込みを使用しない単純に入力を行うサンプルプログラム) ユーザ定義メッセージによる割り込み制御

割り込みハンドラ内部で全ての入出力制御を行う高速割り込み制御プログラム

次ページ以降では、 , のサンプルプログラムについて解説を行います。

```
ユーザ定義メッセージによる割り込みプログラム
```

割り込みソース入力ポートを指定して MsgIntStart()を実行することにより、 割り込み発生に同期したユーザ定義メッセージ"WM\_USER + 5055"が割り込みハン ドラから送られてきます。 このとき、 MsgIntStart の引数である Flg に 1 を指 定していれば、wParam には両方のデータポートの内容がセットされています。

| LRESULT CALLBACK IntProc( HWND hDlg, UINT m<br>IParam ){ | essage, WPARAM wParam, LPARAM |
|--|-------------------------------|
| switch( message ) {                                      |                               |
| case WM_USER + 5510:                                     | // 割り込み通知メッセージ                |
| data = (BYTE)InPort(2);                                  | // 割り込みフラグクリア                 |
| break;   |                               |
| case WM_COMMAND:   |                               |
| switch( wParam ){  |                               |
| case ID_INT_START:                                       | //割り込み開始要因                    |
| IntSet( 0x00, 0x01 );                                    | //割り込みソース入力ポート設定              |
| MsgIntStart(hDlg, 100, 0);                               | //割り込み開始                      |
| }  |                               |
| break;   |                               |
| case IDEND:  |                               |
| case IDCANCEL:   |                               |
| return TRUE;   |                               |
| default:   |                               |
| break;   |                               |
| }  |                               |
| }  |                               |



高速割り込み制御プログラム

割り込みハンドラ内で全ての入出力制御を行うことにより高速割り込み処理 を行うことができます。但し、割り込みハンドラ内で全ての入出力制御を行う 場合は、ユーザ毎の仕様に基づいてデバイスドライバ内の割り込みハンドラを カスタマイズする必要があります。弊社では、有償によるカスタマイズ作業の ご相談も承っております。

サンプルプログラムの割り込みハンドラは、入力方向に設定されたデータレジスタの値をリードし、出力方向に設定されたデータレジスタにリードした値 を出力します。指定回数の割り込み処理が終了すると、呼び出し側プログラム にユーザ定義メッセージをポストします。

入出力データレジスタと割り込み終了回数を指定して、FastIntStart()を実行 することにより、割り込みハンドラ内部で全ての処理を行う高速版の割り込み 処理を開始します。

| 高速割り込みテスト(PHI-8 PHO-8)  | Ľ   |
|---|---|
| <ul> <li>割り込み設定</li> <li>割り込み終了回数:</li> <li>割り込みソースビット:</li> <li>割り込みフラグ:</li> <li>ICLRR2</li> <li>割り込みトリガ:</li> <li>立上りエッジ・</li> </ul> |   |
| ボート入出力設定<br>ボート上位: 入力 ・ 出力ボート: ボート上位 ・<br>ボート下位: 入力 ・ 入力ボート: ボート上位 ・<br>開始  |   |
|   | 高速割の込みテスト(PHI-8 PHO-8)         割り込み設定         割り込みがつれかい         割り込みパースピット:         割り込みフラグ:         ロLRR2         割り込みトリガ:         立上りエッジ・         パート入出力設定         ボート入出力設定         ボート上位:         ボート下位:         スカ         開始 |

## (5-3) Visual BASIC によるアプリケーション開発

Visual BASIC 4.0 以上を使って REX-5055を制御するアプリケーションを開発する場合は、本製品に添付されている 32Bit 版 DLL の関数をコールする必要があります。 Visual Basic で作成したアプリケーションプログラムから 32Bit 版 DLL を呼び出すためには、

1. セットアッププログラムを実行しリソース設定、ドライバのインストールを行う。

2. モジュールファイルで DLL 関数の参照宣言を行う。

(DLL の関数仕様については「(5-2-1)DLL 関数仕様」を参照してください) を行う必要があります。

また、割り込み制御を行う場合は割り込みハンドラから送られてくるユーザ定義メッセ ージを Visual BASIC 側のアプリケーションで受け取るために、本製品に添付されてい る OLE カスタムコントロール MBOX を使用します。 MBOX の使用方法については、サ ンプルプログラム解説を参照してください。

#### アプリケーション作成上のアドバイス

PC カードへの入出力

WindowsNT の保護機能によりアプリケーションレベルでは I/O への入出力は行えま せん。従って、PC カードへの I/O 入出力は DLL でサポートされている OutPort()・ wOutPort()・InPort()・wInPort()を使って行います。

割り込み制御

DLL では割り込みサービスを提供しています。 割り込みサービスには、

(1)割り込みハンドラからユーザ定義メッセージにより割り込み通知を受け取る方法 (2)割り込みハンドラ内で全ての処理を行う方法

があります。高速な処理が要求されるような場合は、(2)の方法で行う必要があります。 詳細は、各サンプルプログラムを参照してください。

### (5-3-1) Visual BASIC サンプルプログラム

| 割り込みを使わない<br>で単純に入出力を行 | PHO-8 LED オンオフ制御プログラム<br>PHI-8 入力プログラム |
|------------------------|--|
| うプログラム例                |  |
| 割り込みを使ったプ              | ユーザ定義メッセージによる割り込み制御                    |
| ログラム例                  | 割り込みハンドラ内部で全ての入出力制御を                   |
|                        | 行う高速割り込み制御                             |

動作環境等については、Visual C のサンプルプログラム解説を参照してくだ さい。ここでは、 ユーザ定義メッセージによる割り込み制御について解説 し、 , , については省略致します。

ユーザ定義メッセージによる割り込み制御サンプルプログラムの解説

下記画面は、VB32 でのデザイン作成時の画面です。割り込み発生に同期させ ユーザ定義メッセージを VB で作成したプログラムで受け取るためには、本製品 添付の OLE カスタムコントロール(OCX)MBOX を使用しなければ VB ではユーザー 定義メッセージを受け取ることが出来ません。以下、カスタムコントロールの インストール手順と MBOX サンプルプログラムの内容について説明します。



└カスタムコントロール「MBOX OLE Control module」を追加

Step.1 本製品添付ソフトのコピー

OLE カスタムコントロール: MBOX.OCX を添付のフロッピーディスクからコピーします。WindowsNT システムディレクトリ名を"WinNT¥System32"とします。

>COPY "コピー元ドライブ名" MBOX.OCX "コピー先ドライブ名":¥WinNT¥System32

**Step.2** 0CX のレジストリー登録 (割り込みサービス使用時必須)

本製品添付の OCX "MBOX.OCX"を VB で使用するためには、VB の CD-ROM に 添付されているツール"REGSVR32.EXE"を使って OCX のレジストリ登録を行います。 "REGSVR32.EXE"は 32 ビットコンソールアプリケーションですので、WindowsNT の DOS BOX から実行します。尚、"REGSVR32.EXE"は VB の CD-ROM の "TOOLS¥PSS"に添付されています。

OCX をレジストリー登録するときは、下記構文で実行します。

>REGSVR32 "ドライブ名":¥WinNT¥System32¥Mbox.ocx

OCX をレジストリー登録から削除するときは、"/U"を付けて下記構文で実行します。

>REGSVR32 /U "ドライブ名":¥ WinNT¥System32¥Mbox.ocx



登録成功メッセージ



登録削除成功メッセージ

Step.3 50551 ib 関数の Declare 宣言

次に、VBプログラムの作成に入ります。VBデザインメニューから新規プロジェクトを 作成し、「ファイル」の「挿入」から標準モジュールを追加します。追加した標準モジュ ールファイルで DLL 関数の参照宣言を行います。宣言部分は、サンプルプログラム "REX5055M.BAS"の下記部分をコピーしてください。

Declare Function IODrvResource Lib "5055lib" () As Long Declare Function IRQDrvResource Lib "5055lib" () As Long Declare Function OutPort Lib "5055lib" (ByVal Adrs As Integer, ByVal WriteData As Integer) As Integer Declare Function wOutPort Lib "5055lib" (ByVal Adrs As Integer, ByVal WriteData As Integer) As Integer Declare Function InPort Lib "5055lib" (ByVal Adrs As Integer) As Integer Declare Function InPort Lib "5055lib" (ByVal Adrs As Integer) As Integer Declare Function wInPort Lib "5055lib" (ByVal Adrs As Integer) As Integer

Declare Function IntSet Lib "50551ib" (ByVal CtrlReg As Byte, ByVal IntMask As Byte) As Integer Declare Sub MsgIntStart Lib "50551ib" (ByVal hWnd As Long, ByVal Count As Integer, ByVal Flg As Integer) Declare Sub FastIntStart Lib "50551ib" (ByVal hWnd As Long, ByVal InPortReg As Byte, ByVal OutPortReg As Byte, ByVal Count As Long, ByVal Flg As Integer) Declare Sub TerminateInterrupt Lib "50551ib" ()

Step.4 MBOX OLE Control Module の追加 (割り込みサービス使用時必須)

VB のカスタムコントロールに MBOX(OCX)を追加します。 VB デザインメニューの 「ツール」の「カスタムコントロール」を起動し、利用可能なコントロールから「MBOX OLE Control module」をチェックします。VB ツールバーに MBOX が追加されます。



Step.5 フォームに MBOX(OCX)を貼り付ける (割り込みサービス使用時必須)

フォームを作成し、割り込みハンドラが割り込み起動元プログラムに送るユーザ定 義メッセージを受け取るための MBOX(OCX)を貼り付けます。これにより、割り込みが 発生すると MBOX がサービスするプロシージャ

MBOX1\_OnMsgPost(ByVal wParam As Integer, ByVal IParam As Long) が呼び出されます。この中で、割り込み通知に同期した処理を記述します。

| 業割り込み処理  | <u>ም</u> አኑ         |                                       |                                       |            |
|--|---------------------|---------------------------------------|---------------------------------------|------------|
| ┌割り込み設定―   |                     | ー <sub>「</sub> カードリソー!                | z                                     |            |
| 割り込み終了回  | 回数                  | ^~~גדי                                |                                       |            |
| 割り込みクリア事   | 要因 ICLRR2 ▼         | ] 割り込み者                               | 時月 📃 🔤                                |            |
| 割り込みビ  | ット PIO 0 🔽          |                                       |                                       |            |
| モニタリングボ・   | -ト  ボート上位 👱         | 」                                     | []                                    | :          |
|  |                     |                                       |                                       | - 1        |
|  |                     |                                       |                                       |            |
|  |                     | · · · · · · · · · · · · · · · · · · · | · · · · · · · · · · · · · · · · · · · |            |
|  |                     | 開始                                    | 終了                                    |            |
|  |                     |                                       |                                       |            |
|  |                     |                                       |                                       |            |
|  |                     |                                       |                                       |            |
|  |                     |                                       |                                       |            |
|  |                     |                                       |                                       |            |
| 🍽 IntTest 🗸                                      |                     |                                       |                                       | <u>-0×</u> |
| オフ <sup>®</sup> ୬ <sup>®</sup> ェクト: <b>■BOX1</b> | ·                   | 7°በ୬-୬°ቀ: 0                           | n∎sgPost                              | <b>•</b>   |
| Private Sub MBOX1_                               | OnMsgPost(ByVal wPa | aram As Integer,                      | ByVal IParam .                        | As Long)   |
| End Sub  |                     |                                       |                                       | -          |
|  |                     |                                       |                                       |            |



この章では、REX-5055のオプションについて説明します。

## (6-1) REX-IPI16 製品概要

REX-IPI16 は、PHI-8の後継型であり外部機器からの無電圧接点入力をフォト カプラによって電気的にアイソレートして TTL レベル信号に変換するアイソレ ーション入力ユニットです。リードスイッチや近接センサーからの信号入力イ ンターフェイスとして使用できます。各ポートの信号状態を LED で確認できま す。

入力電圧を 5V,12V,24V に変更可能です。パソコンの電源電圧(5V)以上の 外部機器と接続される場合には、使用する必要があります。

PHI-8 との違いは、入力ポート数 16、ワンタッチスクリューの螺子端子を使用し、また外形も BOX タイプとなっており非常に使いやすくなっています。



REX-5055 と接続するには、別売オプションケーブル RCL-5055T が必要です。

## サンプルプログラムでのご使用について

MS-DOS/Windows3.1/Windows95/98/WindowsNT に添付している PHI-8 を使用 したサンプルプログラムで、そのままご使用いただけます。

## (6-2) REX-IPO16 製品概要

REX-IP016 は、PHO-8 の後継型でありパソコン側からの 16 ポートの TTL レベ ル信号をフォトカプラによって電気的にアイソレートして外部出力を行うアイ ソレーション出力ユニットです。

出力は、最大 300V、165mA の駆動能力があり、電磁弁やパワーリレーの駆動 制御、スイッチの ON/OFF 制御のインターフェイスとして使用できます。各ポー トの信号状態を LED で確認できます。L レベル信号出力時に LED が点灯します。

PHO-8 との違いは、出力ポート数 16、ワンタッチスクリューの螺子端子を使用し、また外形も BOX タイプとなっており非常に使いやすくなっています。



REX-5055 と接続するには、別売オプションケーブル RCL-5055T が必要です。

# サンプルプログラムでのご使用について

MS-DOS/Windows3.1/Windows95/98/WindowsNT に添付している PHO-8 を使用 したサンプルプログラムで、そのままご使用いただけます。

## (6-3) RCL-TRM48 製品概要

製品添付のオス側コネクタと外部機器を結線し、そのコネクタを本体メス側 コネクタに挿入して信号の入出力を行うワンタッチスクリューターミナルユニ ットです。REX-5055 の信号を全てワンタッチスクリューの螺子端子に出すため のユニットです。

面倒な外部機器からの配線接続作業をドライバ1本で行うことが可能です。 また、外形も BOX タイプとなっており非常に使いやすくなっています。



REX-5055 と接続するには、別売オプションケーブル RCL-5055T が必要です。

REX-5055 DIO PC Card User's Manual

発行 ラトックシステム株式会社2002 年 03 月 22 日 第 7.0 版 第 1 刷発行

## 🖇 製品に対するお問い合わせ

REX-5055 の技術的なご質問やご相談の窓口を用意していますのでご利用<ださい。



