

モーション制御用省配線システム
motionCAT series

PCI Bus Master

HPCI-MCAT520M

CompactPCI Bus Master

HCPCI-MNT720M

ソフトウェアマニュアル

〈DOS 版〉



<http://www.hivertec.co.jp/>

本マニュアル及びプログラムの全部又は一部の無断転載、コピーを禁止します。
本製品の内容に関しましては、改良等により将来予告なしに変更することがあります。
本製品の内容についてお気づきの点がございましたら、お手数ながら当社までご連絡ください。

Microsoft は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows, Windows 98, Windows NT 4.0, Windows 2000, Windows XP, Windows Vista, Windows 7 は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Visual Studio, Visual Basic, Visual C#, Visual C++ は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

.NET Framework は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他、記載されている会社名、製品名は、各社の商標又は登録商標です。

株式会社 ハイバーテック
東京都江東区新大橋 1-8-11
三井生命新大橋ビル
TEL 03-3846-3801
FAX 03-3846-3773
sales@hivertec.co.jp

1.00 版 2010 年 6 月 4 日発行
不許複製・転載

本製品をご使用される前に



motionCAT シリーズユーザーズマニュアル<導入編>「1. はじめに」
を必ずご一読の上ご利用をお願い致します。



本書は、motionCAT シリーズユーザーズマニュアル<導入編>の
1章～3章, 7, 8章をご理解いただいた前提で書かれています。

目 次

1.	はじめに	1
1.1	保証範囲	2
1.2	免責事項	2
1.3	安全にお使い頂くために	2
1.3.1	試運転・調整	3
1.4	マニュアル構成	3
1.4.1	motionCAT シリーズユーザーズマニュアル<導入編>	3
1.4.2	motionCAT シリーズユーザーズマニュアル<運用編>	3
1.4.3	各種ソフトウェアマニュアル	3
1.5	呼称	3
1.6	Windows 版ソフトウェアとの違い	4
1.7	デバイスドライバ	4
1.8	ソフトウェアの構成	4
2.	サンプルプログラム	5
2.1	サンプルプログラム概略	6
2.2	サンプルプログラムの構成	6
2.3	プログラムの実行可能環境	6
2.4	起動画面	7
2.5	motionCAT メインメニュー	7
2.5.1	タイトル・認識ボード枚数とソフトのバージョン番号	8
2.5.2	ボード情報	8
2.5.3	ライン接続モジュール情報	9
2.5.4	マスターボード状態	9
2.5.5	操作メインメニュー	10
2.6	サイクリック通信開始/停止	10
2.7	DIO モジュール	11
2.7.1	DIO モジュール選択画面	11
2.7.2	出力操作画面	11
2.8	モーションモジュール	12
2.8.1	モーションモジュール選択画面	12
2.8.2	操作メイン画面	12
2.8.3	動作状態の表示	13
2.8.4	モジュールの初期化	14
2.8.5	動作コマンド	14
3.	ソフトウェアの準備	18
3.1	マスターボードを複数枚使用する場合	19
3.1.1	ボードのデバイス番号	19

3.1.2	ボード ID の使用.....	19
3.2	ボードアクセス方法.....	19
3.2.1	データ型とデバイス情報構造体.....	19
3.3	ボードアクセスの準備手順と終了処理.....	20
3.3.1	準備手順.....	20
3.3.2	終了処理.....	20
3.4	関数の戻り値.....	21
4.	ドライバ関数.....	22
4.1	ドライバ関数仕様.....	23
4.2	ドライバ関数を割り込み環境で使用する際の注意事項.....	28
5.	ポート資料.....	29
5.1	ポート表.....	30
5.2	オプションポート.....	31
5.2.1	マスターボード汎用入力ポート(HCPCI-MNT720M のみ).....	31
5.2.2	マスターボード汎用出力設定ポート(HCPCI-MNT720M のみ).....	31
5.2.3	G9001A ステータス.....	31
5.2.4	G9001A 通信速度設定状態.....	32
5.2.5	PCI 割込許可.....	32
5.2.6	PCI 割込ステータスポート.....	32
5.2.7	ボード ID 確認ポート.....	32
5.2.8	ボードコード確認ポート.....	33
5.3	PCI コンフィギュレーションレジスタ.....	33
5.4	割り込み機構.....	34
5.5	割り込み使用手順.....	35
5.5.1	初期化.....	35
5.5.2	運用.....	36

1. はじめに

1.1 保証範囲

- 1 本製品の保証期間は、お買い上げ頂いた日より 3 年間です。保証期間中に弊社の判断により欠陥が判明した場合には、本製品を弊社に引き取り、修理または交換を行います。
- 2 保証期間内外に関わらず、弊社製品の使用、供給(納期)または故障に起因する、お客様及び第三者が被った、直接、間接、二次的な損害あるいは、遺失利益の損害に付いて、弊社は本製品の販売価格以上の責任を負わないものとしますので、予めご了承ください。

1.2 免責事項

- 1 本書に記載された内容に沿わない、製品の取付、接続、設定、運用により生じた損害に対しましては、一切の責任を負いかねますので、予めご了承ください。
- 2 本製品は、一般電子機器用(工作機械・計測機器・FA/OA 機器・通信機器等)に製造された半導体製品を使用していますので、その誤作動や故障が直接、生命を脅かしたり、身体・財産等に危害を及ぼしたりする恐れのある装置(医療機器・交通機器・燃焼機器・安全装置等)に適用できるような設計、意図、または、承認、保証もされていません。
ゆえに本製品の安全性、品質および性能に関しては、本マニュアル(またはカタログ)に記載してあること以外は明示的にも黙示的にも一切保証するものではありませんので、予めご了承ください。
- 3 保証期間内外に関わらず、お客様が行った弊社の承認しない製品の改造または、修理が原因で生じた損害に対しましては、一切の責任を負いかねますので、予めご了承ください。
- 4 本書に記載された内容について、弊社もしくは、第三者の特許権、著作権、商標権、その他の知的所有権の権利に対する保証または実施権の許諾を行うものではありません。
また本マニュアルに記載された情報を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合、弊社は、その責任を負いかねますので、予めご了承ください。


1.3 安全にお使い頂くために


この度は、弊社 NC ボードシリーズをご採用頂きまして、誠に有り難う御座います。本マニュアルは、本製品をご使用して頂く場合の取扱い、留意点に付いて記入してありますので、必ずご一読の上ご利用をお願い致します。

尚、本マニュアルは、本マニュアルが添付された NC ボード常設箇所付近の分かりやすい場所に常時保管し、必要に応じて適宜参照・確認頂きますよう、お願い致します。




安全上の注意

本製品のご使用前に、必ずこのユーザーズマニュアル及び付属書類を全て熟読し、内容を理解してから正しくご使用ください。本製品の知識、安全の情報及び注意事項の全てに付いて習熟してからご使用ください。本ユーザーズマニュアルでは、安全注意事項のランクを「警告」、「注意」として区分してあります。

 警告	この表示を無視して、誤った取扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示しています。
---	--

 注意	この表示を無視して、誤った取扱いをすると、人が傷害を負う可能性または物的損害が想定される内容を示しています。
---	--

1.3.1 試運転・調整

 警告	
	本シリーズ製品を使用し装置を動作させる時は、プログラムのデバッグを充分行ってから動作させてください。プログラムに間違いがあると、思わぬ動きをすることがあります。
	本シリーズ製品に添付してあるサンプルプログラムを使用し装置を動作させる時、最初は速度の低いところで、また機械系に合った設定を行って動作を確認してください。機械系に合わない設定で動作を行うと思わぬ動きをすることがあります。

1.4 マニュアル構成

motionCAT シリーズの製品には次のマニュアルが添付されています。

- motionCAT シリーズユーザーズマニュアル<導入編>
- motionCAT シリーズユーザーズマニュアル<運用編>
- 各種マスターソフトウェアマニュアル

1.4.1 motionCAT シリーズユーザーズマニュアル<導入編>

このマニュアルには以下の事項が記述されています。

- (1) 使用上の注意
- (2) 用語の説明
- (3) motionCAT の導入
- (4) マスターボード
- (5) スレーブ
- (6) アクセサリ
- (7) 接続例
- (8) インストール
- (9) 試運転

(1)～(3),(7),(8)の内容は motionCAT システムを使用する方全ての人を対象としていますので、必ずご一読の上ご利用をお願いします。(4)～(6),(9)は主として設置・接続・配線をする開発者を対象としています。

1.4.2 motionCAT シリーズユーザーズマニュアル<運用編>

このマニュアルには motionCAT シリーズ製品に共通するソフトウェア構築のために必要な事項が記述されています。

- (1) 各モジュールの運用
- (2) デバイス資料
- (3) 応用例

1.4.3 各種ソフトウェアマニュアル

各種マスタに添付されるソフトウェアの使用方法について記述されています。

- (1) サンプルプログラム
- (2) ソフトウェアの準備
- (3) ライブラリ関数
- (4) ドライバ関数
- (5) ポート資料

1.5 呼称

本書では motionCAT シリーズ PCI Bus マスタ HPCI-MCAT520M, CompactPCI Bus マスタ HCPCI-MNT720M を MCAT と呼称します。

1.6 Windows 版ソフトウェアとの違い

DOS 版ソフトウェアを使用する場合、Windows 版とは以下に示す点で異なります。

- ソフトウェア構成
- デバイスドライバのインストール
- 割り込みの使用
- ドライバ関数の仕様
- ライブラリ関数の非サポート
- サンプルプログラム使用方法

なお、標準添付CDにはDOS版ソフトウェアは含まれておりません。別途ご請求下さい。

ドライバ関数を使用して作られたライブラリ関数(Windows版でのhpx_xxxx関数)のDOS版はご提供させて頂いておりません。Windows版ライブラリ関数をご参考に作成して下さい。

尚、Windows版ライブラリ関数の大部分は、DOS版で転用が可能ですが、十分検証を行ったうえでご使用下さい。

1.7 デバイスドライバ

DOS版ソフトウェアでは、デバイスドライバ部分を非常駐型に置き換え、その機能をドライバ関数内に含ませたため、デバイスドライバのインストールは不要です。

1.8 ソフトウェアの構成

弊社の提供するDOS版ソフトウェアは、アプリケーションとMCATをつなぐ入出力関数をドライバ関数としてライブラリファイルの形式で提供させて頂いており、メモリーモデル毎にスモール、コンパクト、ミディアム、ラージモデルの4種類用意しております。

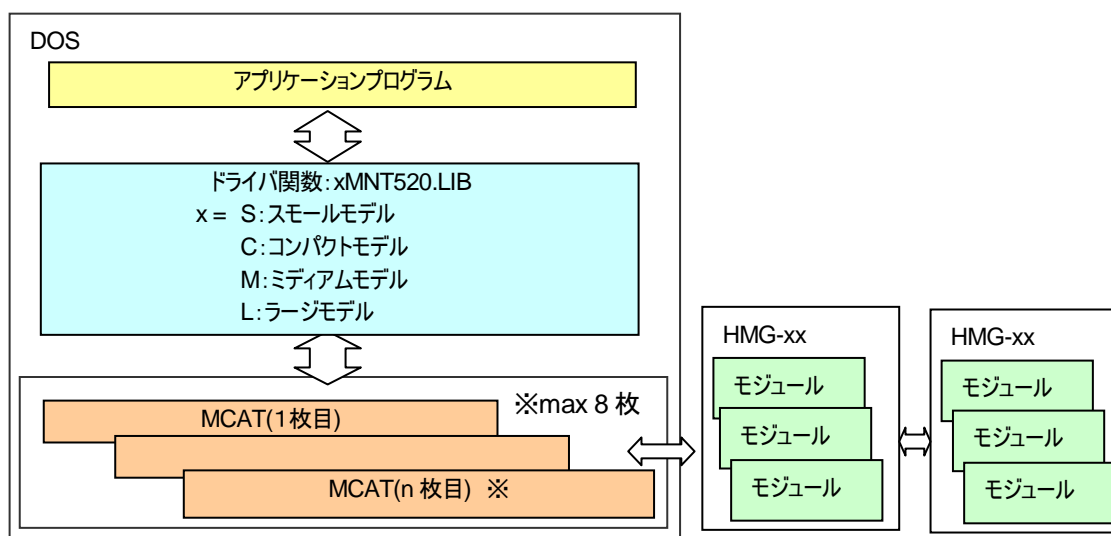


図 1.8-1 ソフトウェアの関連図

2. サンプルプログラム

2.1 サンプルプログラム概略

DOS 版ソフトウェアには「C 言語」で作成されたサンプルプログラムが同梱されています。
このサンプルプログラムは、次の目的で使用して下さい。

(1) 装着ボードと接続モジュールの確認

マスターボードの装着とモジュールの接続を行った後、サンプルプログラムの実行ファイル起動を行いますと、装着ボードの「デバイス情報」表示とボードおよび接続されたモジュールの状態表示と動作確認を行う事ができます。

(2) ドライバ関数の使用例

アプリケーションプログラムは「ドライバ関数←→メモリアクセス関数」経由でボードへの各種操作を行う必要があります。
この操作の一例をサンプルプログラムで記述しておりますので、アプリケーション開発時のご参考用としてご使用頂けます。

2.2 サンプルプログラムの構成

サンプルプログラム・ソースファイルは次の構成となっています。

(1) HPCI-MCAT520M の場合

└─SMCT520.EXE	.. サンプル実行ファイル
└─CLK.BAT	.. 実行ファイル作成用バッチファイル(MS-C V60)
└─SMCT520.C	.. サンプルプログラム・メインソース
└─SMPPXX.C	.. Px サンプルプログラムソース
└─SMPDIO.C	.. Dx サンプルプログラムソース
└─HC520LIB.H	.. ライブラリ参照用ヘッダーファイル
└─COMMON.H	.. サンプルプログラム用ヘッダーファイル
└─LMNT520.LIB	.. リンク用ライブラリ(ラージモデル用)

(2) HCPCI-MNT720M の場合

CLK.BAT	.. 実行ファイル作成用バッチファイル(MS-C V60)
└─SMNT720.EXE	.. サンプル実行ファイル
└─SMNT720.C	.. サンプルプログラム・メインソース
└─SMPPXX.C	.. Px サンプルプログラムソース
└─SMPDIO.C	.. Dx サンプルプログラムソース
└─HC520LIB.H	.. ドライバ関数用ヘッダーファイル
	“SMNT720.C, SMPPXX.C, SMPDIO.C で”#include”
└─COMMON.H	.. サンプルプログラム用ヘッダーファイル
	“SMNT720.C, SMPPXX.C, SMPDIO.C で”#include”
└─LMNT520.LIB	.. リンク用ドライバ関数(ラージモデル用)

2.3 プログラムの実行可能環境

本サンプルプログラム(ドライバー関数を含む)は DOS 上でのみご使用頂けます。Windows や Windows の DOS プロンプト、コマンドプロンプト上では動作しません。また EMM386 等のエクスパンドメモリーマネージャーが組み込まれた環境では動作しません。本サンプルプログラムおよびドライバ関数をご利用されたアプリケーションの実行時には、EMM386.exe を Config.sys から削除してご使用下さい。

2.4 起動画面

実行ファイル "SMCT520.EXE" または "SMNT720.EXE" を起動した時、下記の画面が表示されます。

(1) HPCI-MCAT520M の場合

```
*** HPCI-MNT520M : DOS(16) Device Driver Sample ***
—— Lmmnt520.lib Ver1.2 & Lhmnt520.lib Ver1.0 ——
Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=_
```

(2) HPCPI-MNT720M の場合

```
*** HPCI-MNT720M : DOS(16) Device Driver Sample ***
—— Lmmnt520.lib Ver1.2 & Lhmnt520.lib Ver1.0 ——
Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=_
```

この画面で操作対象のボードの選択を行います。

- 割込機能付きで選択 .. 'I'または'I', 割込機能なしで選択.. 'Y'または'y'キーを押下
- 選択しない .. その他キー入力で次に検出されたボードを表示します。装着されている最後のボードを表示している場合はそのボードが割り込み無しで選択されます。

なお、起動時ボードが装着されていない場合、あるいは認識できない場合は下記メッセージが画面に表示されます。

このメッセージが表示された場合はボードの確認をして下さい。また前述した実行可能な環境を満たさない場合、

<ILL CPU MODE>が表示され処理が中断されます。

```
*** End of Sample Program *** => <NOT FOUND>
```

2.5 motionCAT メインメニュー

正常に起動した後以下画面が表示されます。この画面はサンプルプログラムのメインメニューとなります。

(1) HPCI-MCAT520M の場合

```
*** HPCI-MCAT520M : DOS(16) Device Driver Sample ***
—— Lmmnt520.lib Ver1.2 & Lhmnt520.lib Ver1.0 ——
ID=0000 Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=I
Slave Module Line1: 4 [ DI032: 1 / P100: 3 ] Connect ISTS=00*
Line2:— [ DI032:— / P100:— ] Disconnect ISTS=00*

[ Master Board ]

                Line1                Line2
Main Status  0001x000x0000000 ( 0x1000 )  0000x000x0000000 ( 0x0000 )
Int. Status  00000000x0000000 ( 0x0000 )  00000000x0000000 ( 0x0000 )
EMG Status (1:active) : 0

[ 0:End 1:CONNECT 2:DIO 3:MOTION ] = _
```

(2) HCPCI-MNT720M の場合

```

*** HCPCI-MNT720M : DOS(16) Device Driver Sample ***
—— Lmmt520.lib Ver1.2 & Lhmt520.lib Ver1.0 ——
ID=0000 Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=I
  Slave Module Line1: 4 [ DI032: 1 / P100: 3 ] Connect  ISTS=00*
                    Line2:— [ DI032:— / P100:— ] Disconnect ISTS=00*

[ Master Board ]

Main Status          Line1                Line2
0001x000x00000000 ( 0x1000 )  0000x000x00000000 ( 0x0000 )
Int. Status          00000000x00000000 ( 0x0000 )  00000000x00000000 ( 0x0000 )
General DIO
  in 8-1             xxxxxxxx ( 0x00 )
  out 8-1            xxxxxxxx ( 0x00 )

[0:End 1:CONNECT 2:MASTER DIO 3:DIO 4:MOTION ] = _

```

2.5.1 タイトル・認識ボード枚数とソフトのバージョン番号

ドライバ関数のバージョン番号 デバイスドライバのバージョン番号

```

—— Lmmt520.lib Ver1.2 & Lhmt520.lib Ver1.0 ——
Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=_

```

2.5.2 ボード情報

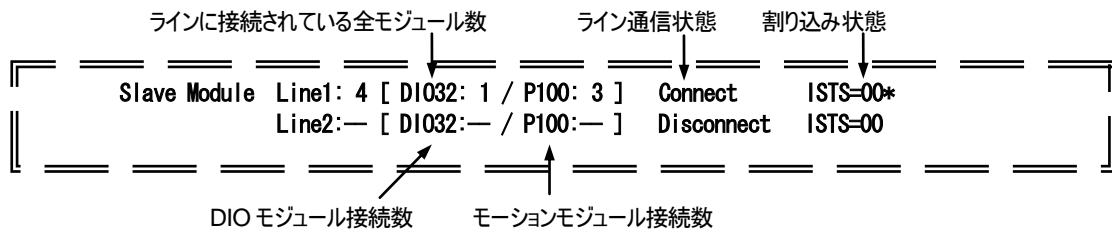
```

ID=0000 Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=I

```

- デバイスハンドルの表示
ID: デバイスオープン時の「デバイスハンドル」値です。(16進数表記)
ボードIDを使用した「デバイスオープン処理」により、デバイスIDの上位ワードにボードID(BID)が付加されています。
- デバイス情報の表示
以下の表示項目は、選択したボードから取得したデバイス情報です。
Bus: バス番号 Dev: デバイス番号(PCIスロット番号)
B2 : ベースアドレス2 B3 : ベースアドレス3
IRQ: IRQ番号 BID: ボードID(ディップスイッチ:0-15)

2.5.3 ライン接続モジュール情報



- モジュール数
各ラインに接続されるモジュール数が表示されます。通信状態が Disconnect の場合は“-”で表示されます。ラインにモジュールが1つも接続されていない場合も同様の表示になります。
- 割り込み状態
“数字”はセンターメインステータス CMSTS の下位7ビットの状態をラッチした値です。(16進数表記)
“*”は割り込みが選択され、割り込み処理が実行される毎にブリンク表示します。

2.5.4 マスターボード状態

(1) HPCI-MCAT520M の場合

	Line1	Line2
Main Status	0001x000x0000000 (0x1000)	0000x000x0000000 (0x0000)
Int. Status	00000000x0000000 (0x0000)	00000000x0000000 (0x0000)
EMG Status (1:active) :	0	

(2) HCPCI-MNT720M の場合

	Line1	Line2
Main Status	0001x000x0000000 (0x1000)	0000x000x0000000 (0x0000)
Int. Status	00000000x0000000 (0x0000)	00000000x0000000 (0x0000)
General DIO		
in 8-1	xxxx0000 (0x00)	
out 8-1	xxxx0000 (0x00)	

- Main Status: センターメインステータス表示
- Int. Status: センター割込ステータス表示
- EMG Status: 非常停止入力(HCPCI-MCAT520M のみ)
- General DIO: マスターボード汎用入出力 入力=4ビット, 出力=4ビット表示(HCPCI-MCAT720M のみ)

ステータスは各ラインについて状態を次の表現で表示します。表示は英数字または記号1文字で表されます。

なお、ステータス右隣に表示される()内は、ステータスの状態を16進数値で表記した値です。

Main	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名称	---	CAER	ERAE	EDTE	EIOE	IOPC	BRKF	CEND
	1表示	?	1	1	1	1	1	1	1
	0表示	X	o	o	o	o	o	o	o
	ビット値	b15	b14	b13	b12	b11	b10	b9	b8
	名称	BBSY	DBSY	RBSY	SBSY	---	RDBB	TDBB	REF
	1表示	1	1	1	1	?	1	1	1
	0表示	o	o	o	o	X	o	o	o
Int.	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名称	LNRV	---	EDN5	EDN4	EDN3	EDN2	EDN1	EDN0
	1表示		?	1	1	1	1	1	1
	0表示		X	o	o	o	o	o	o
	ビット値	b15	b14	b13	b12	b11	b10	b9	b8
	名称	CAE3	CAE2	CAE1	CAE0	ERA3	ERA2	ERA1	ERA0
	1表示	1	1	1	1	1	1	1	
	0表示	o	o	o	o	o	o	o	

DIO 入出力状態は次の表記で表されます。(HCPCI-MCAT720Mのみ)

DIO	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名称	---	---	---	---	in/out4	in/out3	in/out2	in/out1
	1表示	?	?	?	?	1	1	1	1
	0表示	X	X	X	X	o	o	o	o

2.5.5 操作メインメニュー

```
[ 0:End 1:CONNECT 2:DIO 3:MOTION ] = _
```

キーボードからメニューの番号を入力します。

- "0"キー: プログラム終了です。
- "1"キー: 指定ラインのサイクリック通信開始/停止を行います。
- "2"キー: 指定DIOモジュールのDIO入出力状態表示と出力操作を行います。
- "3"キー: 指定モーションモジュールの各種操作を行います。

2.6 サイクリック通信開始/停止

ライン毎にサイクリック通信の開始/停止を操作します。

操作時に通信がすでに開始されているラインでは停止、停止しているラインでは開始が行えます。

```
[ 0:End 1:CONNECT 2:MASTER DIO 3:DIO 4:MOTION ] = 1
Select Line(1,2) = 1    Disconnect?(Y,N) = _
```

- Select Line : 操作ライン番号(1 or 2)を入力します。
- Connect/Disconnect: 通信開始/停止の確認(Y or N)を行います。

操作が成功すると画面上部の通信状態の表示が変化します。操作後はメインメニューへ戻ります。

なお、モジュールが1台も接続されていないラインは通信を開始できません。

2.7 DIO モジュール

DIO モジュールの入出力状態の表示, 出力状態の変更を行います。

2.7.1 DIO モジュール選択画面

```

*** HPCI-MCAT520M : DOS(16) Device Driver Sample ***
—— Lmmnt520.lib Ver1.0 & Lhmt520.lib Ver1.0 ——
ID=0000 Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=I
  Slave Module Line1: 4 [ D1032: 1 / P100: 3 ]   Connect   ISTS=00*
                    Line2:— [ D1032:— / P100:— ]   Disconnect ISTS=00*

[ DIO Module ]

      Port  I/O      Status

Valid Module MID = 03

[ 0:End 1:CONNECT 2:DIO 3:MOTION ] = 3
Select Line( 1,2 ) = 1      Select MID( HEX ) = 3
Int. SelPort( HEX ) = f

```

- Select Line : 操作ライン番号(1 or 2)を入力します。
- Select MID : 操作モジュール番号(1~63)を入力して Enter を押します。
入力可能な有効モジュール ID は“Valid Module ID”に表示されます。
- Int. SelPort : 入力ポート変化フラグを有効にします。有効にするビットを 1 にした 1 桁の 16 進数(0~F)で入力します。

2.7.2 出力操作画面

```

┌─── [ DIO Module ] Line:1 MID:03 IntP:0xf ───┐
│                                             │
│      Port  I/O      Status                │
│      0   in  8- 1   11000000( 0xc0 )     │
│      1   in 16- 9   00000000( 0x00 )     │
│      2   out 8- 1   00000100( 0x04 )     │
│      3   out16- 9   00100001( 0x21 )     │
│                                             │
│      Output Operation( Select:←,→,Num Fin.:ESC Invert:Others )
└───┘

```

DO 出力操作を以下のキーを使って行います。

- 矢印キー(←, →, ↑, ↓) : 上下左右へカーソルを移動します。
- 数字キー(0-7) : 指定したビットへカーソルを移動します。
- ESC キー : 操作を終了して操作メインメニューへ戻ります。
- その他キー : カーソル位置の出力ビットを反転します。

2.8 モーションモジュール

モーションモジュールの操作および設定を行います。

2.8.1 モーションモジュール選択画面

```

*** HPCI-MCAT520M : DOS(16) Device Driver Sample ***
—— Lmmt520.lib Ver1.0 & Lhmt520.lib Ver1.0 ——
ID=0000 Bus=02 Dev=04 B2=f8122000 B3=f8120000 IRQ=11 BID=00 Int/Use/Next?=I
  Slave Module Line1: 4 [ DI032: 1 / P100: 3 ] Connect ISTS=00*
                    Line2:— [ DI032:— / P100:— ] Disconnect ISTS=00*

[ Motion Module ]

Valid Module MID = 01 02 03

[ 0:End 1:CONNECT 2:DIO 3:MOTION ] = 4
Select Line( 1,2 ) = 1      Select MID( HEX ) = 0
Int. SelPort( HEX ) = f

```

- Select Line : 操作ライン番号(1 or 2)を入力します。
- Select MID : 操作モジュール番号(1~63)を入力して Enter を押します。
入力可能な有効モジュール ID は“Valid Module ID”に表示されます。
- Int. SelPort : 入力ポート変化フラグを有効にします。有効にするビットを 1 にした 1 桁の 16 進数(0-F)で入力します。

2.8.2 操作メイン画面

```

┌───────────────────────────────────────────────────────────────────────────────────┐
│ [Motion Module] Line:1 MID:03 IntP:0xf                                         │
│                                                                                   │
│ CTR1          647                                                                 │
│ CTR2           0      **** 2.Execute Command List ****                       │
│ rSPD           0      0:CNST 1:Get OLS 2:OLS+Z 3:Get_ELS                       │
│ MssL  xxxxxxxx( 0x00 ) 4:Post 5:'+' Dir 6:'-' Dir                               │
│ MssH  x111xxxx( 0x70 ) S:SrvOn R:SrvRst T:TrqOn G:Gain                       │
│ ExsLL  ooooooooo( 0x00 )                                                                 │
│ ExsLH  oooooo111( 0x07 )  ─── 3.Register Disp & set ───                       │
│ ExsHL  ooooooooo( 0x00 )  0:BasS 1:MovS 2:Acc 3:Mult 4:Mode                   │
│ ExsHH  xxxxxxxx( 0x00 )  5:AuxS 6:RENV1 7:RENV2 8:RENV3 I:INT                 │
│ IOPB   ooooooooo( 0x00 )  D:DLS 0:OLS E:ELS A:ALM                             │
│ RIST   0x000000                                                                 │
│ REST   0x000000                                                                 │
│                                                                                   │
│ Operation [0: End 1: BoardInit 2: Cmd Exec. 3: Register] = _                 │
└───────────────────────────────────────────────────────────────────────────────────┘

```

キーボードからメニューの番号を入力します。

- "0"キー : モーションモジュール操作を終了し、操作メインメニューへ戻ります。
- "1"キー : モーションモジュールの初期化をします。(起動時にも初期化が実行されます)
- "2"キー : 動作指令を送ります。
- "3"キー : レジスタの設定を行います
- "その他" : 全軸への減速停止指令を行います。

2.8.3 動作状態の表示

次の項目がレジスタより読み出され表示されます。

- CTR1 : 指令パルス数
- CTR2 : フィードバックパルス数
- rSPD : 動作中の速度(RFL/RFHと同一単位のステップ値)
- MssL : メインステータス下位8ビット
- MssH : 上位8ビット
- ExsLL : 拡張ステータスビット7~0
- ExsLH : 拡張ステータスビット15~8
- ExsHL : 拡張ステータスビット23~16
- ExsHH : 拡張ステータスビット31~24
- IOPB : 汎用出力ポートの状態
- RIST : イベントステータス
- REST : エラーステータス

ステータスは各ラインについて状態を次の表現で表示します。表示は英数字または記号1文字で表されます。

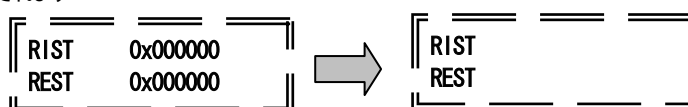
なお、ステータス右隣に表示される()内は、ステータスの状態を16進数値で表記した値です。

MssL	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名称	---	---	---	---	SEVT	SERR	SECD	SINT.
	1表示	?	?	?	?	1	1	1	1
	0表示	x	x	x	x	o	o	o	o
MssH	ビット値	b15	b14	b13	b12	b11	b10	b9	b8
	名称	---	GRP2	GRP1	GRP0	---	---	---	SBSY
	1表示	?	1	1	1	?	?	?	1
	0表示	x	o	o	o	x	x	x	o
ExsL L	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名称	SMEL	SPEL	SALM	SDIR	CND3	CND2	CND1	CND0
	1表示	1	1	1	1	1	1	1	1
	0表示	o	o	o	o	o	o	o	o
ExsL H	ビット値	b15	b14	b13	b12	b11	b10	b9	b8
	名称	SERC	SPCS	SEMG	SSTP	SSTA	SDIN	SSD	SORG
	1表示	1	1	1	1	1	1	1	1
	0表示	o	o	o	o	o	o	o	o
ExsH L	ビット値	b23	b22	b21	b20	b19	b18	b17	b16
	名称	SPLS	SCP3	SCP2	SCP1	SINP	SLTC	SCLR	SEZ
	1表示	1	1	1	1	1	1	1	1
	0表示	o	o	o	o	o	o	o	o
ExsH H	ビット値	b31	b30	b29	b28	b27	b26	b25	b24
	名称	---	---	---	---	SPH4	SPH3	SPH2	SPH1
	1表示	?	?	?	?	1	1	1	1
	0表示	x	x	x	x	o	o	o	o
IOPB	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名称	ELL	GRP2	GRP1	GRP0	SVGAIN	SVTL	SVRST	SVON
	1表示	1	1	1	1	1	1	1	1
	0表示	o	o	o	o	o	o	o	o

- 割り込み使用時と不使用時の表示の違い

RISTおよびRESTの表示は、割り込み有無により表示方法が変わります。

割り込み不使用時は、下記の表示が常時表示されます。割り込み使用時はRISTまたはREST要因発生時に一定時間だけ表示されます。



2.8.4 モジュールの初期化

モジュールの初期化では以下の動作を行います。

- ソフトウェアリセット(オプションポートの状態はリセットされません)
- レジスタへ初期値書き込み

2.8.5 動作コマンド

動作指令の種類は右図に示す11種類ありますが、これは次の3種類に分類されます。

- 動作速度の変更 ……O: 定速/高速切替
- サーボ信号出力 ……S: サーボ ON
……R: サーボリセット
……T: トルク制御切替
……G: サーボゲイン切替
- 動作指令 ……上記以外

```

=====
**** 2. Execute Command List ****
O:CNST 1:Get OLS 2:OLS+Z 3:Get_ELS
4:Post 5:'+ Dir 6:'- Dir
S:SrvOn R:SrvRst T:TrqOn G:Gain
=====

```

(1) 動作速度変更

軸への動作指令を行う場合の速度を切り替えます。操作は SelCmd=0 と指定します。

- ・「CNST」表示……FL定速スタート
- ・「HIGH」表示……高速スタート

```

=====
**** 2. Execute Command List ****
O:CNST 1:Get OLS 2:OLS+Z 3:Get_ELS
4:Post 5:'+ Dir 6:'- Dir
S:SrvOn R:SrvRst T:TrqOn G:Gain
=====
-----
Operation [ 0:End 1:BoardInit 2:Cmd Exec. 3:Register ] = 2
SelCmd( 0-6, S, R, T, G ) = 0
HIGH SPEED ==> Normal End
=====

```

コマンド0の表示が“CNST”から“HIGH”に変化します。

```

=====
**** 2. Execute Command List ****
O:HIGH 1:Get OLS 2:OLS+Z 3:Get_ELS
4:Post 5:'+ Dir 6:'- Dir
S:SrvOn R:SrvRst T:TrqOn G:Gain
=====

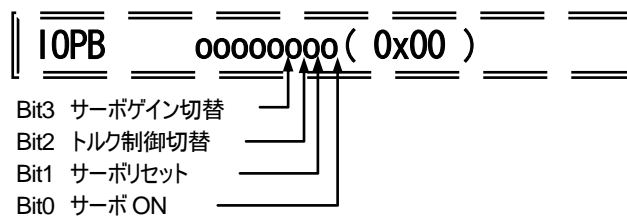
```

(2) サーボ信号出力

サーボオン, サーボリセット, トルク制御, ゲイン切替の各サーボ制御信号を操作します。
サーボアンプに接続されていない信号は単なる DO 信号として動作します。

- サーボ ON
サーボ ON 信号を ON/OFF します。ON/OFF はトグル動作となります。
操作は SelCmd=S と指定します。
- サーボリセット
サーボリセット信号を ON/OFF します。ON/OFF はトグル動作となります。
自動的に OFF しませんので手動で OFF する必要があります。
操作は SelCmd=R と指定します。
- トルク制御切替
トルク制御切替機能があるサーボアンプでトルク制御を ON/OFF します。
操作は SelCmd=T と指定します。切替はトグル動作となります。
- サーボゲイン切替
ゲイン切替機能があるサーボアンプでゲイン切替をします。
操作は SelCmd=G と指定します。切替はトグル動作となります。

ON/OFF の結果は“IOPB”ステータスの表示で確認できます。(o:OFF 1:ON)

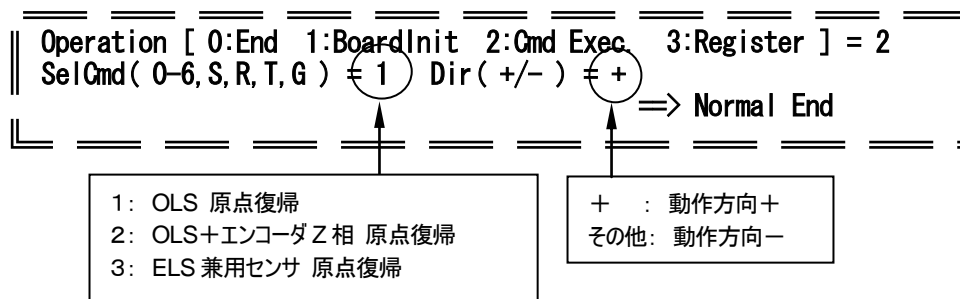


(3) 原点復帰動作

原点復帰は次の3通り用意されています。

原点復帰中に手動停止(キー押下), 異常検出, 移動方向のリミットを検出すると復帰動作を中断して停止します。

- OLS 原点復帰(OLS 検出後拔出し再突入完了)
OLS 検出で(減速)停止し, 反転拔出し再度 OLS 検出で終了します。
操作手順は, SelCmd=1 とし, 動作方向方向('+'キー以外は全て'-')を指定します。
- OLS+エンコーダZ相
OLS 検出後(減速開始し), Z相検出で終了します。
操作手順は, SelCmd=2 とし, 動作方向('+'キー以外は全て'-')を指定します。
- ELS 兼用センサ原点
ELSon 検出で(減速)停止し, 反転低速拔出し, ELSoff で終了します。
操作手順は, SelCmd=3 とし, 動作方向('+'キー以外は全て'-')を指定します。



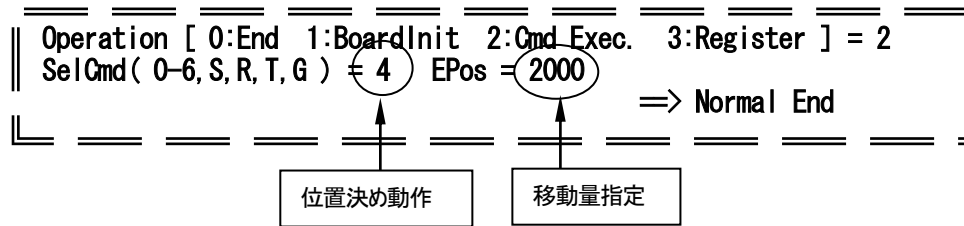
※ “Normal End”の表示はモーションモジュールに正常にコマンドを送れた事を示すものであり, 完了を示すものではありません。
(以降同様)

(4) 位置決め動作

位置決め動作を行います。

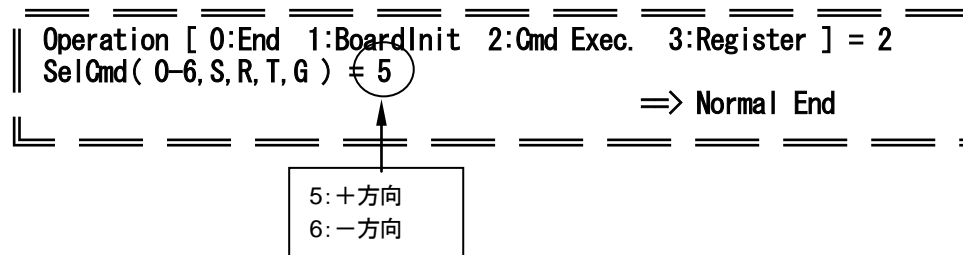
操作手順は、SelCmd=4とし、移動量を指定します。(移動量はマイナス指定も可能)

位置決め動作中に手動停止(キー押下)、異常検出、移動方向のリミットを検出すると動作を中断して停止します。

**(5) 連続運転**

連続運転を行います。

- 十方向連続
十方向の連続運転を行います。手動停止(キー押下)、異常停止、+リミット検出で停止します。操作は SelCmd=5と指定します。
- 一方向連続
一方向の連続運転を行います。手動停止(キー押下)、異常停止、-リミット検出で停止します。操作は SelCmd=6と指定します。



■ レジスタの設定

以下のレジスタ値はサンプルプログラム上で変更することができます。
キー入力値による変更で書込みが行われます。

選択表示	キー	変更内容	選択表示	キー入力	変更内容
0:Base	0	ベース速度	I:INT.	I	イベントマスク
1: MovS	1	動作速度	D:DLS	D	DLS入力極性反転
2: Acc	2	加速レート	O:OLS	O	OLS入力極性反転
3: Mult	3	速度倍率	E:ELS	E	ELS入力極性反転
4: Mode	4	動作モード	A:ALM	A	ALM入力極性反転
5: AuxS	5	移動量補正速度			
6: RENV1	6	環境1			
7: RENV2	7	環境2			
8: RENV3	8	環境3			

本サンプルプログラムで設定が変更可能なレジスタは上表の14種類あります。メニューから”3(Register)”を選択し、表示/変更したい項目の数値/英文字1字をキー入力して選択します。

```

===== 3. Register Disp & set =====
0: BasS 1: MovS 2: Acc 3: Mult 4: Mode
5: AuxS 6: RENV1 7: RENV2 8: RENV3 I: INT
D: DLS O: OLS E: ELS A: ALM
  
```

```

=====
Operation [ 0:End 1:BoardInit 2:Cmd Exec. 3:Register ] = 3
SelReg( 0-8, I, D, O, E, A ) = 1 Move Speed[2000: 1-65, 535] = _
=====
                                     ↑
                               現在の設定値
                                     ↓
                               設定値入力可能範囲
  
```

ここで変更したい値を入力すると、適正な値であれば入力値が設定されます。
不正な値や数値を入力せずにEnterキーを押した場合は変更を行わずに終了します。

4種類の入力センサを選択した場合は入力極性の反転を行った後直ちに終了します。入力極性反転操作の結果は表示画面に反映されます。
(ここでは拡張レジスタのビット6,7)

```

=====
ExsLL  11000000 ( 0xc0 )
ExsLH  00000111 ( 0x07 )
ExsHL  00000000 ( 0x00 )
ExsHH  xxxxxxxx ( 0x00 )
=====
  
```

```

=====
Operation [ 0:End 1:BoardInit 2:Cmd Exec. 3:Register ] = 3
SelReg( 0-8, I, D, O, E, A ) = E
=====
=> ELS N. C
  
```

ELS 検出から検出なしの状態に変化

```

=====
ExsLL  00000000 ( 0x00 )
ExsLH  00000111 ( 0x07 )
ExsHL  00000000 ( 0x00 )
ExsHH  xxxxxxxx ( 0x00 )
=====
  
```

3. ソフトウェアの準備

3.1 マスターボードを複数枚使用する場合

MCAT を 1 台のコンピュータに複数枚装着し、それぞれのボードと外部の接続を 1 対 1 に対応させる場合について説明します。

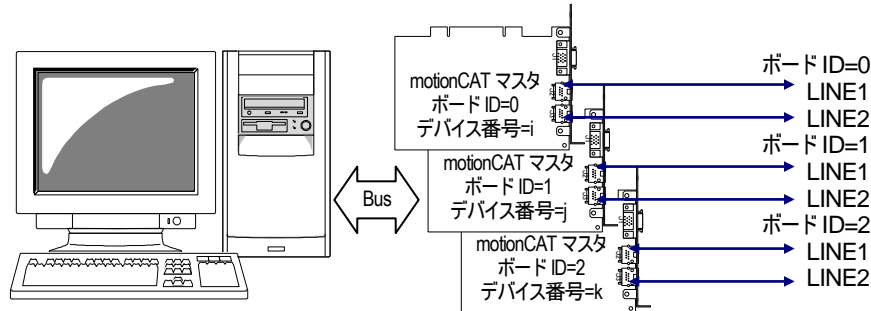


図 3.1-1 ボードを複数枚使用

3.1.1 ボードのデバイス番号

PCI では BIOS がボードのアドレス管理をしています。

ボードが装着されるスロットにより BIOS 側で決めたデバイス番号が割振られます。これをデバイス番号と呼びます。

3.1.2 ボード ID の使用

デバイス番号は BIOS によって割振られるため、ボードとスロットの関係が外部から認識しにくい場合があります。

個々のボードとソフトウェアを対応しやすくするために MCAT には「ボード ID 設定ロータリ SW」が設けられています。

ボード ID は 0h~Fh(0~15)が設定出来ますので MCAT を 16 枚まで扱えます。

ボード ID の設定については「motionCAT シリーズユーザーズマニュアル<導入編>」を参照して下さい。

3.2 ボードアクセス方法

ドライバ関数・ライブラリ関数では複数の MCAT を制御することができます。

制御したい MCAT にアクセスするためには、まずこの MCAT をオープンして、デバイスハンドルを取得します。

MCAT をオープンするために必要なハードウェアリソース(※)をデバイス情報と呼びます。

※. ハードウェアリソース…ボード ID, ベースアドレス, 割込番号等

3.2.1 データ型とデバイス情報構造体

(1) データ型

データ型	符号付		符号なし	
	8bit	char	U_CHAR	
	16bit	Short	U_SHORT	
32bit	long	U_LONG		

(2) ボード(デバイス)認識用構造体

ボード認識のために次に示す HMNT520INFO 型構造体を、ボード枚数最大 8 枚として、使用枚数分用意します。

```

Typedef struct {
    Short    BusNo;      /* 0:+ 0: バス番号 (0 to 255) */
    Short    DevNo;     /* 1:+ 2: デバイス番号 (0 to 31) */
    U_LONG   Badr2;     /* 2:+ 4: ベースアドレス(2) (32bit) */
    U_LONG   Badr3;     /* 3:+ 8: ベースアドレス(3) (32bit) */
    U_LONG   Badr4;     /* 4:+12: ベースアドレス(4) (32bit) */
    Short    IrqNo;     /* 5:+16: IRQ 番号 (0 to 15) */
    Short    BrdID;     /* 6:+18: ボード ID (0 to 15) */
    Short    BdSel;     /* 7:+20: 32bit FS セクタ : Badr2,3 */
} HMNT520_INFO;

```


3.3 ボードアクセスの準備手順と終了処理

3.3.1 準備手順

(1) ボード毎にデバイスオープン

まず MCAT の枚数を取得し、ボード認識用のデータ構造体を枚数分用意します。

次に MCAT のデバイス情報を取得し、制御したい MCAT のデバイス情報をデバイスオープン関数に渡します。この結果その MCAT がオープンされます。

また、デバイスオープン関数はこの MCAT にアクセスする為のデバイスハンドル値を返してきます。

MCAT 枚数が 2 枚以上の場合には、個々の MCAT 毎にこの処理を行います。

- ◆mnt520__GetMstBrdCount() ...MCAT の枚数取得
- ◆mnt520__GetMstBrdInfo() ...MCAT の PCI デバイス情報取得
- ◆mnt520__OpenMstBrd() ...MCAT のオープン処理

(2) 初期設定

MCAT のオープンに成功した後は、システム通信を開始し、ローカルデバイス情報を自動設定します。

ポートデータの初期値を設定し、サイクリック通信を開始します。サイクリック通信開始後、データ通信も可能になりますので、ローカルデバイスの初期化を行います。

- ◆mnt520__rSysLclInfo() ...ローカルデバイス(DIO, モーション)の情報取得
- ◆mnt520__wCenCmd() ...センターデバイスへのコマンド書込み
- ◆mnt520__rCenMsts() ...センターデバイスからのセンターメインステータス読出し
- ◆mnt520__wIoPortB() ...DIO デバイスのポートへの書込み
- ◆mnt520__wIoPortW() ...DIO デバイスのポートへの書込み
- ◆mnt520__wLclSetInt() ...ローカルデバイスのポート入力変化割込設定
- ◆mnt520__wPclPortB() ...モーションデバイスのポートへの書込み
- ◆mnt520__wPclSetInt() ...モーションデバイスのポート入力変化割込設定
- ◆mnt520__wPclReg() ...モーションデバイスのレジスタへの書込み

(3) 運用

初期設定終了後はデータ通信、サイクリック通信によりローカルデバイスを制御します。

- ◆mnt520__rIoPortB() ...DIO デバイスのポートからの読出し
- ◆mnt520__wIoPortB() ...DIO デバイスのポートへの書込み
- ◆mnt520__rIoPortW() ...DIO デバイスのポートからの読出し
- ◆mnt520__wIoPortW() ...DIO デバイスのポートへの書込み
- ◆mnt520__rLclInt() ...ローカルデバイスのポート入力変化割込読出し
- ◆mnt520__wLclInt() ...ローカルデバイスのポート入力変化割込リセット
- ◆mnt520__rPclPort() ...モーションデバイスのポートからの読出し
- ◆mnt520__wPclPort() ...モーションデバイスのポートへの書込み
- ◆mnt520__rPclMsts() ...モーションデバイスのモーションメインステータス読出し
- ◆mnt520__wPclCmd() ...モーションデバイスへのコマンド書込み
- ◆mnt520__rPclReg() ...モーションデバイスのレジスタからの読出し
- ◆mnt520__wPclReg() ...モーションデバイスのレジスタへの書込み

3.3.2 終了処理

(4) オープンしたデバイスの「クローズ処理」

全ての処理が終了してアプリケーションを終了する場合には、システム上の終了処理を終えてから、オープンしたデバイスの「クローズ処理」を行います。

MCAT 枚数が 2 枚以上の場合には、個々の MCAT 毎にこの処理を行います。

- ◆mnt520__CloseMstBrd() ...MNT520 のクローズ処理

3.4関数の戻り値

Windows 用ドライバ関数の戻り値と一部異なります。関数の戻り値が異常値('0'以外)であった場合には、異常内容に対応した処理をアプリケーション側で行って下さい。

No	戻り値		異常内容と確認項目
	記号表記	Hex	
1	NO_ERROR	00000000	正常 ◎異常は発生していません
2	NOT_FOUND	00000001	デバイスが存在しない ◎ボードが挿入されていない ◎違う種類のボードが入っている ◎ボード上 DSW の設定とアクセス方法が異なっている
3	ALREADY_OPENED	00000002	オープン済デバイスをオープン ◎既にオープンしているボードとアドレスが重複している ◎9 枚以上のボードをオープンしようとした
4	INVALID_HANDLE	00000008	無効なデバイスハンドル ◎指定されたデバイスハンドルが不正 ◎オープンされていないデバイスハンドルを使用
5	ILL_PARAM	00000100	関数の引数の値が不正 ◎パラメータに指定範囲外の値を設定した ◎システムで設定不可の設定または組み合わせの設定を行った
6	OTHER_BOARD	00000200	ボード種類が違う ◎MCAT 以外のボードをオープンしようとした
7	ILL_CPU_MODE	00002000	CPU モード不正 ◎Windows または Windows 上の DOS 窓で実行した ◎EMM386.EXE を使用している
8	ACCESS_BUSY	00004000	デバイスにアクセス中 ◎システムの不整合が考えられます。弊社までご連絡ください。
9	NO_LOCAL	00008000	モジュールが存在しない ◎ライン上に指定したモジュールが見つからない ◎ケーブルが接続されていない、または外れている
10	CYC_COM_ERROR	00010000	サイクリック通信エラー ◎スレーブの電源が入っていない ◎通信ケーブルが断線している ◎通信ケーブルにノイズが入っている
11	CYC_COM_STOP	00020000	サイクリック通信停止中 ◎サイクリック通信がスタートされていない状態でデータ通信をしようとした
12	DATA_COM_ERROR	00040000	データ通信エラー
13	COM_OTHER_ERROR	00080000	その他の通信エラー ◎上記以外の通信エラー検出 ◎「センター割込ステータス」で詳細を確認して下さい
14	MODULE_OVERCOUNT	00100000	1ラインに接続されているモジュールの数が 33 個以上 ◎1ラインに接続できるモジュールの数は 32 個以下です。
15	ILL_ACCESS_COM	00400000	通信中異常検出 ◎不正なコマンドで通信を行った ◎システム通信中に通信を行った
16	SEND_DATA_NONUSE	00800000	未使用デバイスにデータ送信 ◎未使用のモジュールに対して通信を行った
17	OTHER_ERROR	08000000	その他のエラー ◎システムの不整合が考えられます。弊社までご連絡ください。

表 3.4-1 関数の戻り値

4. ドライバ関数

4.1 ドライバ関数仕様

DOS 用ドライバ関数の関数仕様は以下のとおりです。

Windows 用のドライバ関数とは、戻り値、引数のデータ型が異なりますが、関数名や関数の機能、戻り値、引数の内容については同じです。

(1) mnt520_GetMstBrdCount () MCAT 枚数の取得		
機能	現在システムに装着されている MCAT 枚数を取得します。	
DOS	書式	U_LONG mnt520_GetMstBrdCount(short * count);
	引数	short * count .. 取得した MCAT 枚数の格納アドレス
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(2) mnt520_GetMstBrdInfo () MCAT の PCI デバイス情報取得		
機能	現在システムに装着されている MCAT の PCI デバイス情報を取得します。	
DOS	書式	U_LONG mnt520_GetMstBrdInfo(short * count, HMCAT520_INFO * hInfo);
	引数	short * count .. 取得した MCAT 枚数の格納アドレス HMCAT520_INFO * hInfo 取得した MCAT の PCI デバイス情報の先頭アドレス
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(3) mnt520_OpenMstBrd () MCAT のオープン		
機能	MCAT をオープンし、MCAT のデバイスハンドルを取得します。	
DOS	書式	U_LONG mnt520_OpenMstBrd(short * hDev, HMCAT520_INFO * hInfo);
	引数	short * hDev .. 取得したデバイスハンドルの格納アドレス HMCAT520_INFO * hInfo .. オープンする MCAT の PCI デバイス情報の先頭アドレス
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(4) mnt520_CloseMstBrd () MCAT のクローズ		
機能	MCAT をクローズします。	
DOS	書式	U_LONG mnt520_CloseMstBrd(short hDev);
	引数	short hDev .. デバイスハンドル
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(5) mnt520_rCenMsts () センターメインステータス読出し		
機能	センターメインステータスを読出します。	
DOS	書式	U_LONG mnt520_rCenMsts(short hDev, short wLine, U_SHORT * wSts);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン1, 1:ライン2) U_SHORT * wSts .. センターメインステータス
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(6) mnt520_wCenCmd () センターデバイスへのコマンド書込み		
機能	センターデバイスへコマンドを書込みます。	
DOS	書式	U_LONG mnt520_wCenCmd(short hDev, short wLine, U_SHORT wCmd);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン1, 1:ライン2) U_SHORT wCmd .. センターデバイスコマンドコード
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(7) mnt520_rCenIsts () センター割込ステータス読出し		
機能	センター割込ステータスを読出します。	
DOS	書式	U_LONG mnt520_rCenIsts(short hDev, short wLine, U_SHORT * wSts);
	引数	short hDev .. デバイスハンドル
		short wLine .. ライン番号(0:ライン1, 1:ライン2) U_SHORT * wSts .. センター割込ステータス
戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照	

(8) mnt520_rCenBuf () センターデバイスの入出力バッファ読出し		
(9) mnt520_wCenBuf () センターデバイスの入出力バッファ書込み		
機能	センターデバイスの入出力バッファを読出し, または書込みます。	
DOS	書式	U_LONG mnt520_rCenBuf(short hDev, short wLine, U_SHORT * wInBuf); U_LONG mnt520_wCenBuf(short hDev, short wLine, U_SHORT wOutBuf);
	引数	short hDev .. デバイスハンドル
		short wLine .. ライン番号(0:ライン1, 1:ライン2)
		U_SHORT * wInBuf .. センターデバイス入出力バッファ読出しデータ U_SHORT wOutBuf .. センターデバイス入出力バッファ書込みデータ
戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照	

(10) mnt520_rCenRFifo () センターデバイスの受信用 FIFO 読出し		
(11) mnt520_wCenSFifo () センターデバイスの送信用 FIFO 書込み		
機能	センターデバイスの受信用 FIFO から読出します。 センターデバイスの送信用 FIFO へ書込みます。	
DOS	書式	U_LONG mnt520_rCenRFifo(short hDev, short wLine, U_SHORT * wReceive); U_LONG mnt520_wCenSFifo(short hDev, short wLine, U_SHORT wSend);
	引数	short hDev .. デバイスハンドル
		short wLine .. ライン番号(0:ライン1, 1:ライン2)
		U_SHORT * wReceive .. センターデバイス受信用 FIFO 読出しデータ U_SHORT wSend .. センターデバイス送信用 FIFO 書込みデータ
戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照	

(12) mnt520_rLclCycErr () サイクリック通信エラーフラグ読出し		
(13) mnt520_wLclCycErr () サイクリック通信エラーフラグリセット		
機能	サイクリック通信エラーフラグを読出します。 サイクリック通信エラーフラグをリセットします。	
DOS	書式	U_LONG mnt520_rLclCycErr(short hDev, short wLine, short wMidPm, U_SHORT * wFlag); U_LONG mnt520_wLclCycErr(short hDev, short wLine, short wMidPm, U_SHORT wFlag);
	引数	short hDev .. デバイスハンドル
		short wLine .. ライン番号(0:ライン1, 1:ライン2)
		short wMidPm .. モジュール ID (MID) 指定パラメータ (0: MID= 0-15, 1: MID=16-31, 2: MID=32-47, 3: MID=49-64)
戻り値	U_SHORT * wFlag .. エラーフラグ U_SHORT wFlag .. エラーフラグ '0'で正常, '0'以外は「3.4 関数の戻り値」参照	

(14)mnt520_rLclInfo () ローカルデバイス情報読出し		
(15)mnt520_wLclInfo () ローカルデバイス情報書込み		
機能	MCAT に接続されているローカルデバイス情報を読出します。 MCAT に接続されているローカルデバイス情報を書込みます。	
DOS	書式	U_LONG mnt520_rLclInfo(short hDev, short wLine, short wMid, U_CHAR * byData); U_LONG mnt520_wLclInfo(short hDev, short wLine, short wMid, U_CHAR byData);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン 1, 1:ライン 2) short wMid .. モジュール ID(0:MID=0, ..., 63:MID=63) U_CHAR * byData .. ローカルデバイス情報 U_CHAR byData .. ローカルデバイス情報
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(16)mnt520_rLclSetInt () ローカルデバイス入力ポート変化割込設定状態読出し		
(17)mnt520_wLclSetInt () ローカルデバイス入力ポート変化割込設定書込み		
機能	指定したローカルデバイスのポートの入力変化割込設定状態を読み出します。 指定したローカルデバイスのポートの入力変化割込を設定します。	
DOS	書式	U_LONG mnt520_rLclSetInt(short hDev, short wLine, short wMidPrm, U_SHORT * wData); U_LONG mnt520_wLclSetInt(short hDev, short wLine, WORD wMidPrm, U_SHORT wData);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン 1, 1:ライン 2) short wMidPrm .. モジュール ID(MID) 指定パラメータ (0:MID= 0~ 3, 1:MID= 4~ 7, 2:MID= 8~11, 3:MID=12~15, 4:MID=16~19, 5:MID=20~23,6:MID=24~27, 7:MID=28~31, 8:MID=32~35, 9:MID=36~39,10:MID=40~43,11:MID=44~47, 12:MID=48~51,13:MID=52~55,14:MID=56~59,15:MID=60~63) U_SHORT * wData .. ローカルデバイス 4 個分(16 ポート)入力変化割込設定状態 U_SHORT wData .. ローカルデバイス 4 個分(16 ポート)入力変化割込設定データ
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(18)mnt520_rLclInt () ローカルデバイス入力ポート変化フラグ読出し		
(19)mnt520_wLclInt () ローカルデバイス入力ポート変化フラグリセット		
機能	指定したローカルデバイスのポートの入力ポート変化フラグを読み出します。 指定したローカルデバイスのポートの入力ポート変化フラグをリセットします。	
1111DOS	書式	U_LONG mnt520_rLclInt(short hDev, short wLine, short wMidPrm, U_SHORT wFlag); U_LONG mnt520_wLclInt(short hDev, short wLine, short wMidPrm, U_SHORT wFlag);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン 1, 1:ライン 2) short wMidPrm .. モジュール ID 指定パラメータ (0:MID= 0~ 3, 1:MID= 4~ 7, 2:MID= 8~11, 3:MID=12~15, 4:MID=16~19, 5:MID=20~23,6:MID=24~27, 7:MID=28~31, 8:MID=32~35, 9:MID=36~39,10:MID=40~43,11:MID=44~47, 12:MID=48~51,13:MID=52~55,14:MID=56~59,15:MID=60~63) short * wFlag .. ローカルデバイス4個分(16ポート)入力ポート変化フラグ U_SHORT wFlag .. ローカルデバイス4個分(16ポート)入力ポート変化フラグ
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(20)mnt520_rloPortB ()		DIO デバイスポート1バイト読出し
(21)mnt520_rloPortW ()		DIO デバイスポート2バイト読出し
(22)mnt520_wloPortB ()		DIO デバイスポート1バイト書込み
(23)mnt520_wloPortW ()		DIO デバイスポート2バイト書込み
機能		指定した DIO デバイスのポートから読出します。 指定した DIO デバイスのポートへ書込みます。
DOS	書式	U_LONG mnt520_rloPortB(short hDev, short wLine, short wMid, short wPrtNo, U_CHAR * byData); U_LONG mnt520_rloPortW(short hDev, short wLine, short wMid, short wPrtNo, U_SHORT * wData); U_LONG mnt520_wloPortB(short hDev, short wLine, short wMid, short wPrtNo, U_CHAR byData); U_LONG mnt520_wloPortW(short hDev, short wLine, short wMid, short wPrtNo, U_SHORT wData);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン1, 1:ライン2) short wMid .. モジュール ID(0:MID=0, ..., 63:MID=63) short wPrtNo.. ポート番号(0:ポート0, ..., 3:ポート3) U_CHAR * byData.. 読出されたポートデータ(1バイト)の格納アドレス U_CHAR byData.. 書込むポートデータ(1バイト) U_SHORT * wData .. 読出されたポートデータ(2バイト)の格納アドレス U_SHORT wData .. 書込むポートデータ(2バイト)
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(24)mnt520_rPclPort ()		モーションデバイス入出力ポートからの1バイト読出し
(25)mnt520_wPclPort ()		モーションデバイス出力ポートへ1バイト書込み
機能		指定したモーションデバイスの汎用入出力ポートから読出します。 指定したモーションデバイスの汎用出力ポートへ書込みます。
DOS	書式	U_LONG mnt520_rPclPort(short hDev, short wLine, short wMid, U_CHAR * byData); U_LONG mnt520_wPclPort(short hDev, short wLine, short wMid, U_CHAR * byData);
	引数	short hDev .. デバイスハンドル short wLine .. ライン番号(0:ライン1, 1:ライン2) short wMid .. モジュール ID(0:MID=0, ..., 63:MID=63) U_CHAR * byData.. 入出力ポートから読出された1バイトデータの格納アドレス U_CHAR byData.. 出力ポートへ書込むデータ(1バイト)
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(26)mnt520_rPclMsts ()		モーションメインステータスの読出し
機能		指定したモーションデバイスのモーションメインステータスを読出します。
DOS	書式	U_LONG mnt520_rPclMsts(short hDev, short wLine, short wMid, U_SHORT * wSts);
	引数	short hDev .. デバイスハンドル short wLine.. ライン番号(0:ライン1, 1:ライン2) short wMid .. モジュール ID(0:MID=0, ..., 63:MID=63) U_SHORT * wSts .. モーションメインステータスの格納アドレス
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
(27)mnt520_wPclCmd ()		モーションデバイス制御コマンド書込み
機能		指定したモーションデバイスへ制御コマンドを書込みます。
DOS	書式	U_LONG mnt520_wPclCmd(short hDev, short wLine, short wMid, U_SHORT wCmd);
	引数	short hDev .. デバイスハンドル short wLine.. ライン番号(0:ライン1, 1:ライン2) short wMid .. モジュール ID(0:MID=0, ..., 63:MID=63) U_SHORT wCmd .. 制御コマンドコード
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(28)mnt520_rPclReg () モーションデバイスレジスタからの読出し		
(29)mnt520_wPclReg () モーションデバイスレジスタへの書き込み		
機能	指定したモーションデバイスのレジスタから読出します。 指定したモーションデバイスのレジスタへ書き込みます。	
DOS	書式	U_LONG mnt520_rPclReg(short hDev, short wLine, short wMid, U_SHORT wCmd, U_LONG * dwReg); U_LONG mnt520_wPclReg(short hDev, short wLine, short wMid, U_SHORT wCmd, U_LONG dwReg);
	引数	short hDev .. デバイスハンドル short wLine.. ライン番号(0:ライン1, 1:ライン2) short wMid .. モジュールID(0:MID=0, ..., 63:MID=63) U_SHORT wCmd .. レジスタ制御コマンドコード U_LONG * dwReg.. レジスタから読み出されたレジスタデータの格納アドレス U_LONG dwReg.. レジスタへ書き込むレジスタデータ
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(30)mnt520_rOptPortW () オプションポート2バイト読出し		
(31)mnt520_wOptPortW () オプションポート2バイト書き込み		
機能	指定したマスターボードのオプションポートから読出します。 指定したマスターボードのオプションポートへ書き込みます。	
DOS	書式	U_LONG mnt520_rOptPortW(short hDev, U_SHORT wAdrs, U_SHORT * wData); U_LONG mnt520_wOptPortW(short hDev, U_SHORT wAdrs, U_SHORT wData);
	引数	short hDev .. デバイスハンドル U_SHORT wAdrs .. オプションポート部ボード内アドレス U_CHAR * byData.. 読出しデータ(1バイト) U_SHORT * wData .. 読出しデータ(2バイト) U_CHAR byData.. 書き込みデータ(1バイト) U_SHORT wData .. 書き込みデータ(2バイト)
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(32)mnt520_rCenPortW () センターデバイス指定ポート2バイト読出し		
(33)mnt520_wCenPortW () センターデバイス指定ポート2バイト書き込み		
機能	センターデバイスの指定ポートから読出します。 センターデバイスの指定ポートへ書き込みます。	
DOS	書式	U_LONG mnt520_rCenPortW(short hDev, U_SHORT wAdrs, U_SHORT * wData); U_LONG mnt520_wCenPortW(short hDev, U_SHORT wAdrs, U_SHORT wData);
	引数	short hDev .. デバイスハンドル U_SHORT wAdrs.. センターデバイス部ボード内アドレス U_SHORT * wData.. 読出しデータ(2バイト) U_SHORT wData.. 書き込みデータ(2バイト)
	戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照

(34)mnt520_SetIntCall() 割り込み処理関数の登録・削除	
機能	デバイス ID で指定された MCAT ボードからの割り込みが発生した場合の処理関数を登録または削除します。
書式	U_LONG mnt520_SetIntCall(short hDev, PINT.PROC padr);
引数	short hDev .. デバイスハンドル PINT.PROC fnIntCall..登録: 割り込みが発生した時の割り込み処理モジュールのアドレス 削除: NULL(=0)
戻り値	'0'で正常, '0'以外は「3.4 関数の戻り値」参照
呼出例	short ret; //関数の戻り値 PINT.PROC int_module; //割り込み処理モジュールのアドレス //割り込み処理関数登録 ret=mnt520_SetIntCall(ret, int_module);
DOSのみ	《ご注意》...割り込みを使用する場合には、次の点に留意して下さい。 (1)初期化時の指令関数の順序 ① センターデバイスの初期化を実行する。(ソフトウェアリセットコマンド0x100の発行) ② センターデバイスに接続するすべてのローカルデバイスの初期設定を行う。 ③ 不要な割り込みのマスクを RENV0.bit0~6で行う。デフォルトではこれらビットはすべて割り込み有効(マスクなし=0)になっています ④ 入力ポート変化フラグの設定を行う。 ⑤ mnt520_SetIntCall()関数で割り込み処理モジュールをドライバに通知します。 (オプションポート INT.EPCI 割り込み許可ポート)の操作は関数内で行っています
備考	(2)割り込み処理モジュール(アプリケーションプログラム) ① ボード単位で処理を行います。(全搭載軸の一括処理) ② モジュール内では”CPU 割り込み禁止”を保持します。(モジュールは割り込み禁止で起動。) ③ モジュール本体およびモジュール内で起動するモジュールでは”スタックチェック”禁止です。(関数用スタックエリア=1KB) (禁止としない時,DOS から”スタックオーバーフロー”が報告される事があります。) ④ ドライバ関数実行中に割り込み処理モジュール内でドライバ関数を起動する時、異常終了「戻り値=0x40(ILLEGAL_ACCESS)」となる場合があります。 (I/Oポートのリエントラント処理ができない為) ⑤ センターメインステータスの通信終了およびエラービット(bit0~6)で割り込みが発生する設定の場合、これらステータスの読み込みは割り込み処理内で行い、その他の処理でステータスを使用する場合はフラグ渡しにしてご使用下さい。 ◎ ステータスのリセットが自動リセット(RENV0 bit9=0)に設定されている場合、一度読み出すとクリアされてしまうため ◎ 手動リセットの場合、割り込み処理内で割り込みフラグをクリアしないと割り込みが発生し続け、割り込みから抜けなくなるため。これは自動リセットの場合でも自動的にリセットされないエラー項目については同様のことが起きます。 ⑥ 作成方法はサンプルプログラムを参照して下さい。 (3)終了時に忘れてはならないこと。 ①入力ポート変化フラグのクリア。 ② mnt520_SetIntCall()関数で割り込み処理モジュールをドライバから消去します

4.2 ドライバ関数を割り込み環境で使用する際の注意事項

関数 mnt520_wPclCmd(No.17), mnt520_rPclReg, mct110_wPclReg(No.18)は、前述したドライバ関数のいくつかを組み合わせて作られており、通信完了待ち、エラー監視などの処理も含んでいます。そのため、実行時にボードからの割り込みを発生させないよう、関数内でオプションポート「INTE」をマスク、ボードからの割り込みを一時的に停止する処理を行っています。(実行終了時に復帰させています)

5. ポート資料

5.1ポート表

HPCI-MCAT520MとHCPCI-MNT720Mのアドレスマップは同じです。ポートはすべてメモリマップです。

BAR	区分	Address (Hex)	読み込み(INP)		書き込み(OUT)	
			呼称	内容	呼称	内容
BAR2	#1 ライン1	0000	CMSTS	センターメインステータス(15-0)	CCMD	コマンド(15-0)
		0002	CISTS	センター割込ステータス(15-0)	-	Reserved
		0004	CBUF	入出力バッファIN(15-0)	CBUF	入出力バッファ OUT(15-0)
		0006	RFIFO	データ受信FIFO(15-0)	SFIFO	データ送信FIFO(15-0)
		0008 ~ 0077	-	Reserved	-	Reserved
		0078	DINFO	ローカルデバイス情報(78-B7)	DINFO	ローカルデバイス情報(78-B7)
		00B8	IOERR	サイクリック通信エラーフラグ(B8-BF)	IOERR	サイクリック通信エラーフラグ(B8-BF)
		00C0	IINT.	入力ポート変化フラグ設定状態(C0-DF)	IINT.	入力ポート変化フラグ設定(C0-DF)
		00E0	IRES	入力ポート変化フラグ(E0-FF)	IRES	入力ポート変化フラグリセット(E0-FF)
		0100	DATA	ポートデータ IN(100-1FF)	DATA	ポートデータ OUT(100-1FF)
	#2 ライン2	0000	CMSTS	センターメインステータス(15-0)	CCMD	コマンド(15-0)
		0002	CISTS	センター割込ステータス(15-0)	-	Reserved
		0004	CBUF	入出力バッファIN(15-0)	CBUF	入出力バッファ OUT(15-0)
		0006	RFIFO	データ受信FIFO(15-0)	SFIFO	データ送信FIFO(15-0)
		0008 ~ 0077	-	Reserved	-	Reserved
		0078	DINFO	ローカルデバイス情報(78-B7)	DINFO	ローカルデバイス情報(78-B7)
		00B8	IOERR	サイクリック通信エラーフラグ(B8-BF)	IOERR	サイクリック通信エラーフラグ(B8-BF)
		00C0	IINT.	入力ポート変化フラグ設定状態(C0-DF)	IINT.	入力ポート変化フラグ設定(C0-DF)
		00E0	IRES	入力ポート変化フラグ(E0-FF)	IRES	入力ポート変化フラグリセット(E0-FF)
		0100	DATA	ポートデータ IN(100-1FF)	DATA	ポートデータ OUT(100-1FF)
BAR3	オプション ポート	0000	DIN	マスターボード汎用入力状態(注) (J1: MDR14)	-	Reserved
		0002	DOUT	マスターボード汎用出力状態(注)	DOUT	マスターボード汎用出力設定(注) (J1: MDR14)
		0004	STS	G9001Aステータス	-	Reserved
		0006	SPD	通信速度設定状態	-	Reserved
		0008	INT.E	PCI割込許可状態	INT.E	PCI割込許可
		000A	INT.S	PCI Bus割込ステータス	-	Reserved
		0010	BID	ボードID設定状態	-	Reserved
		0012	BCOD	Board Code 'MNT520_ _'	-	Reserved

BAR:ベースアドレス

注: HPCI-MCAT520Mには DIN, DOUT はありません。

表 5.1-1 ボードアドレス

5.2 オプションポート

オプションポートはマスターボード全体に1組おかれています。

オプションポートからデータの読み出しは

mnt520_rOptPortB() ... 1バイトデータ読み出し

mnt520_rOptPortW() ... 2バイトデータ読み出し

データの書き込みは

mnt520_wOptPortB() ... 1バイトデータ書き込み

mnt520_wOptPortW() ... 2バイトデータ書き込み

を使用します。

5.2.1 マスターボード汎用入力ポート(HCPCI-MNT720Mのみ)

(BAR3+00H:DIN)(Read only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	IN4	IN3	IN2	IN1

図 5.2-1 マスターボード汎用入力ポートのビット構成

bit	記号	機能	備考
3-0	IN4-1	汎用入力状態 ONで"1"	
15-4	(未定義)	常に"0"	

表 5.2-1 マスターボード汎用入力ポートの内容

5.2.2 マスターボード汎用出力設定ポート(HCPCI-MNT720Mのみ)

BAR3+02H: DOUT (Read/Write)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	OUT4	OUT3	OUT2	OUT1

図 5.2-2 マスターボード汎用出力設定及び出力状態確認ポートのビット構成

bit	記号	機能	備考
3-0	OUT4-1	汎用出力設定"1"でON, 汎用出力状態 ONで"1"	
15-4	(未定義)		書き込みした値が読めます

表 5.2-2 マスターボード汎用出力設定及び出力状態確認ポートの内容

5.2.3 G9001A ステータス

BAR3+04H: STS (Read only)

15-12	11	10	9	8	7~4	3	2	1	0
0	2MSYN	2MERF	2MERR	2MCRY	0	1MSYN	1MERF	1MERR	1MCRY

図 5.2-3 G9001A ステータス確認ポートのビット構成

bit	記号	機能	備考
8,0	xMCRY	1: 信号ライン信号検出で一定時間	
9,1	xMERR	1: 異常フレーム受信時と、無応答時に一定時間	
10,2	xMERF	1: エラー応答フレーム受信時一定時間Lレベル	
11,3	xMSYN	サイクリック周期毎に反転	

表 5.2-3 G9001A ステータス確認の内容

5.2.4 G9001A 通信速度設定状態

BAR3+06H: SPD(Read Only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SPD0	SPD1	DSW2	DSW1

図 5.2-4 G9001A 通信速度設定状態のビット構成

bit	記号	機 能
0	DSW1	通信設定 SW bit0 ON で "1"
1	DSW2	通信設定 SW bit1 ON で "1"
3,2	SPD1,0	通信設定 bit 読み出し ON で "1" 00:20Mbps,10:10Mbps,01:5Mbps,11:2.5Mbps

表 5.2-4 G9001A 通信速度設定確認ポートの内容

5.2.5 PCI 割込許可

BAR3+08H: INT.E(Read/Write)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	INTE

図 5.2-5 PCI 割込許可設定ポートのビット構成

bit	記号	機 能	備 考
0	INTE	PCIBus への割込を許可します。“1”で許可, “0”で禁止。 (割込要因は G9001A に依ります)	Windows 版ソフトウェアではサポートしていません。
15-1	(未定義)		書き込みした値が読めます

表 5.2-5 PCI 割込許可設定ポートの内容

5.2.6 PCI 割込ステータスポート

BAR3+0AH: INT.S(Read only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	INT.S2	INT.S1

図 5.2-6 PCI 割込ステータスポートのビット構成

bit	記号	機 能	備 考
0	INT.S1	“1”で Line1 から PCIBus への割込が発生していることを示します。	
1	INT.S2	“1”で Line2 から PCIBus への割込が発生していることを示します	
15-2	(未定義)	常に“0”	

表 5.2-6 PCI 割込ステータスポートの内容

5.2.7 ボード ID 確認ポート

BAR3+10H: BID(Read only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	BID4	BID3	BID2	BID1

図 5.2-7 ボード ID 確認ポートのビット構成

bit	記号	機 能
3-0	BID4-1	基板上のボード ID 設定ロータリ SW の状態が読み出せます。
15-4	(未定義)	常に“0”

表 5.2-7 モジュール ID 読出(ボード ID)の内容

5.2.8 ボードコード確認ポート

(BAR3+12H..17:BCOD)(Read only)

アドレス	データ	内容	備考
+12H	4E4D H	N,M	
+14H	5254 H	52,T	
+16H	0000 H	_,0	

表 5.2-8 ボードコード確認ポートの内容

5.3 PCI コンフィギュレーションレジスタ

HPCI-MCAT520MとHCPCI-MNT720MのPCIコンフィギュレーションレジスタの内容は同じです。

PCIコンフィギュレーションレジスタの内容を次表に示します。

31	24	23	16	15	8	7	0	アドレス
デバイスID 1034h				ベンダID 14a9h				00h
デバイスステータス				デバイス制御				04h
クラスコード								リビジョンID (00h)
基本クラス (06h)		サブクラス (80h)		プログラム インターフェース				
セルフテスト		ヘッダタイプ		マスタレイテンシタイム		キャッシュライン		0ch
ベ ー ス ア ド レ ス レ ジ ス タ	00000000h(予 約)							10h
	PCI9030(PCIコンフィギュレーションレジスタ)							14h
	G9001ベースアドレス(BAR2)							18h
	オプションポートベースアドレス(BAR3)							1ch
	00000000h(予 約)							20h
	00000000h(予 約)							24h
カードバスCISポインタ								28h
サブシステムID 1034h				サブシステムベンダID 14a9h				2ch
予 約								30h
								7 fch

表 5.3-1 PCI コンフィギュレーションレジスタ

ベースアドレス	ターゲット	空間	Bus 幅	備考
BAR2	#1G9001A	512 Byte	16bit	
BAR2+200h	#2G9001A	512 Byte	16bit	
BAR3	オプションポート	16 Byte	16bit	

BAR:ベースアドレス

表 5.3-2 PCI アドレス空間

5.4 割り込み機構

motionCAT マスターボードの割り込み機構を次図に示します。

ステータス詳細は「motionCAT シリーズユーザーズマニュアル<運用編>」を参照して下さい。

CMSTS のビット 2: IOPC は、「入力変化ポート IRES」の割り込みがマスクされていないビットがセットされた時に集約的にセットされ、該当ラインの割り込み出力の1つになります。

CMSTS のビット 3: EIOE についても「サイクリック通信エラーフラグ IOERR」がセットされた時に集約的にセットされ、該当ラインの割り込み出力の1つになります。

これらは CEND, BRKF, EDTE, ERAE, CAER の各割り込みソースと OR され該当ラインの割り込みとして1つにまとめられて出力されます。MCAT 上にはこのラインが2つあり OR されて割り込み出力されます。

MCAT からの PCI Bus への割り込み出力はボード割り込みマスク(オプションポート INTE)を経て CPU に割り込みます。

この CPU への割り込み信号処理は、DOS 版ドライバでサポートされますが、ボード割り込みマスクを行い、CMSTS のポーリングで割り込み要因を取り出し処理する事もできます。

- 要因毎の割り込み許可/禁止は環境レジスタ0(RENVO) bit0~bit6 で行います。

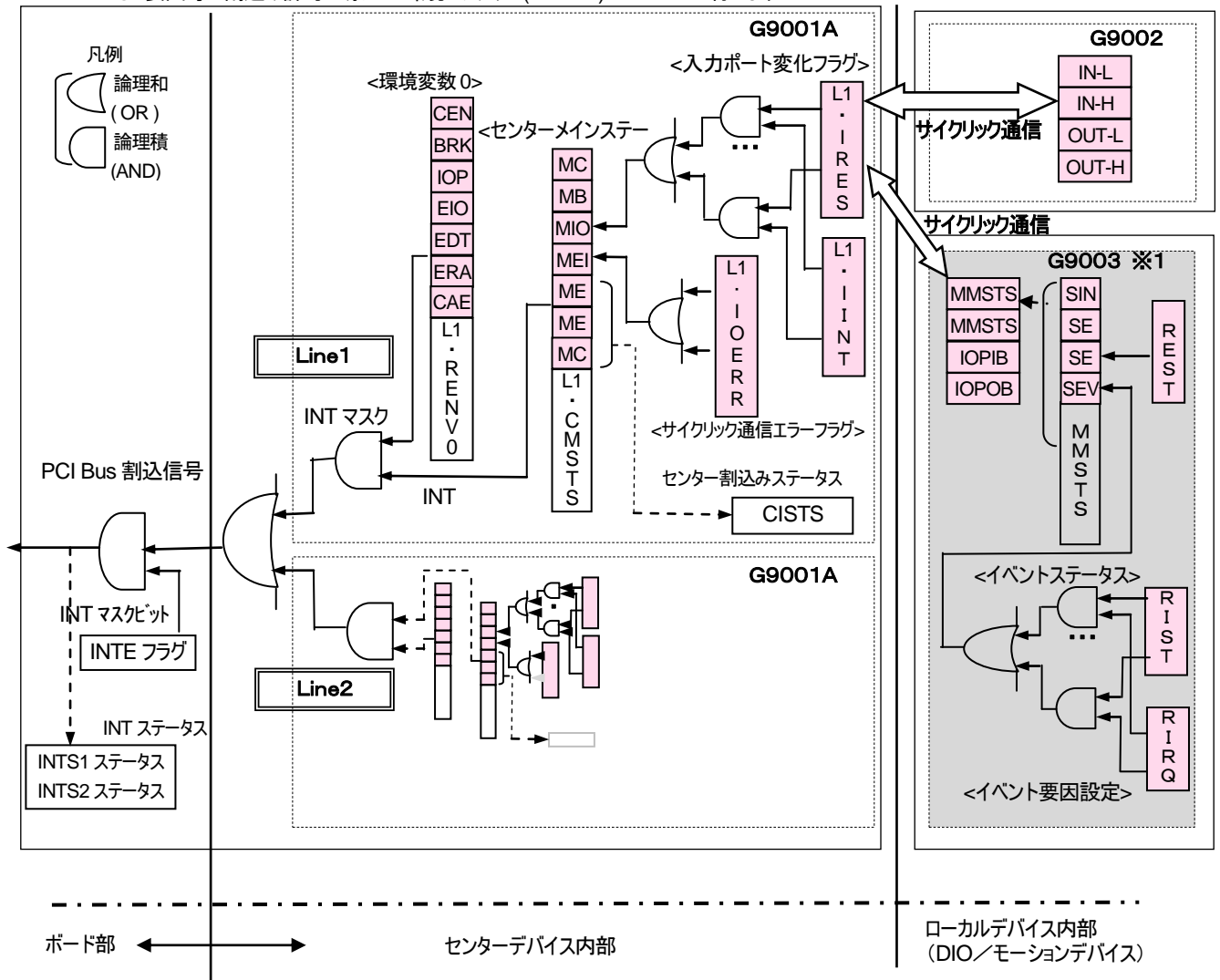


図 5.4-1 MCAT520 ボード内割込みルート

※1 モーションデバイス(G9003)のエラーおよび完了監視を割り込みで使用することは仕様上可能ですが、この場合 G9003 のリセット処理を含む割り込み関連処理全般をすべて割り込み内で行う事が必要となり運用が非常に複雑になります。したがって G9003 エラーおよび完了監視については割り込みを使用せずポーリングにて運用される事を推奨します。

5.5 割り込み使用手順

割り込みを使用する場合の基本的な手順は以下のとおりです。

5.5.1 初期化

(1) マスターボード(G9001A)の初期化

- ① CPU 割り込み設定
割り込みベクターの設定およびユーザ割り込み関数の登録を行います。ドライバ関数に含まれる MCAT オープン関数「mnt520_OpenMstBrd ()」を使用します。
- ② 環境変数 RENV0設定
 - 不要な割り込み要因をマスク。(マスクするビットに1をセット)
ビット1, 2につきましては下記の点ご注意ください。
 - ・ ビット1:MBRK
ブレークフレーム受信による割り込み要因は使用できないため常時マスク‘1’にして使用して下さい。
 - ・ ビット2:MIOP
DIO モジュールなどローカルデバイスからのステータス変化による割り込みを使用する場合はマスクしないで下さい。
 - ステータスビットのクリア方法を設定。
 - 自動ブレーク機能選択
この機能は使用できません。常時‘1’(無効)でご使用下さい。
- ③ 入力ポート変化フラグ(INT.)の設定
ローカルデバイスのポートに対応する入力ポート変化フラグを‘1’に設定すると、入力ポートの状態が変化した時にマスターボードに割り込みが掛かります。通常 DIO モジュールではポート0, 1またはいずれかに‘1’をセットして使用します。
- ④ 割り込みイネーブル
割り込みイネーブルポート(INT.E)を‘1’にします。通常、ローカルデバイスを含めたすべての初期化が終了した後イネーブルにします。

(2) DIO モジュール(G9002)の初期化

割り込みは、マスターボード「入力ポート変化フラグ設定」を設定することで発生します。したがって DIO モジュール本体に対する割り込み設定はありません。

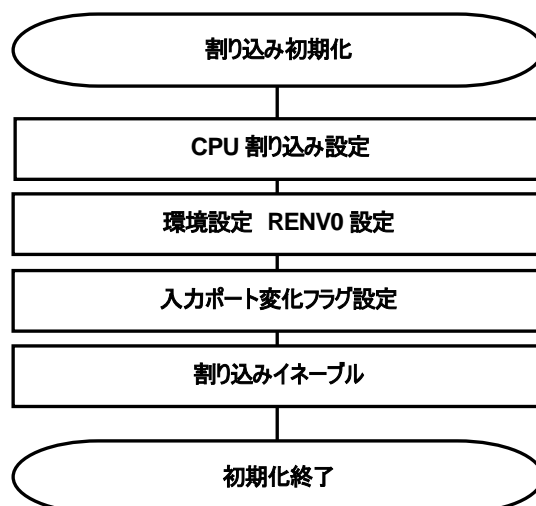


図 5.5-1 初期化手順フロー

5.5.2 運用

MCAT の割り込みは、複数の発生要因が1つの割込ソースに集約されて出力されます。したがって割り込み発生時は、その発生要因をセンターメインステータスから調べ、さらに要因によっては別のレジスタを参照して詳細を調べる等、発生要因に応じた処理を行う必要があります。また、次の割り込みを受け入れるためリセット処理が必要です。

発生要因	他レジスタの参照	リセット操作
CEND BRKF (不使用) EDTE		RENVO Bit9 の設定にが 0 の場合はリードによりクリアされるため特別な操作は不要。 Bit9 が 1 の場合はリセットするビットをクリアコマンドによりリセット
IOPC	入力ポート変化フラグ(IRES)を検索して割り込み発生モジュールを特定	入力ポート変化フラグの該当モジュールに対応するビットに 1 を書き込んでリセット
EIOE	サイクリック通信エラーフラグ(IOERR)を検索してエラー発生モジュールを特定	サイクリック通信エラーフラグのエラーが発生した該当モジュールに対応するビットに 1 を書き込んでリセット
ERAE CAER	センター割り込みステータス(CISTS)を参照して発生原因詳細を取得	RENVO Bit9 の設定にが 0 の場合はリードによりクリアされるため特別な操作は不要。 Bit9 が 1 の場合はリセットするビットをクリアコマンドによりリセット

表 5.5-1 エラー発生要因による参照とリセットの対応

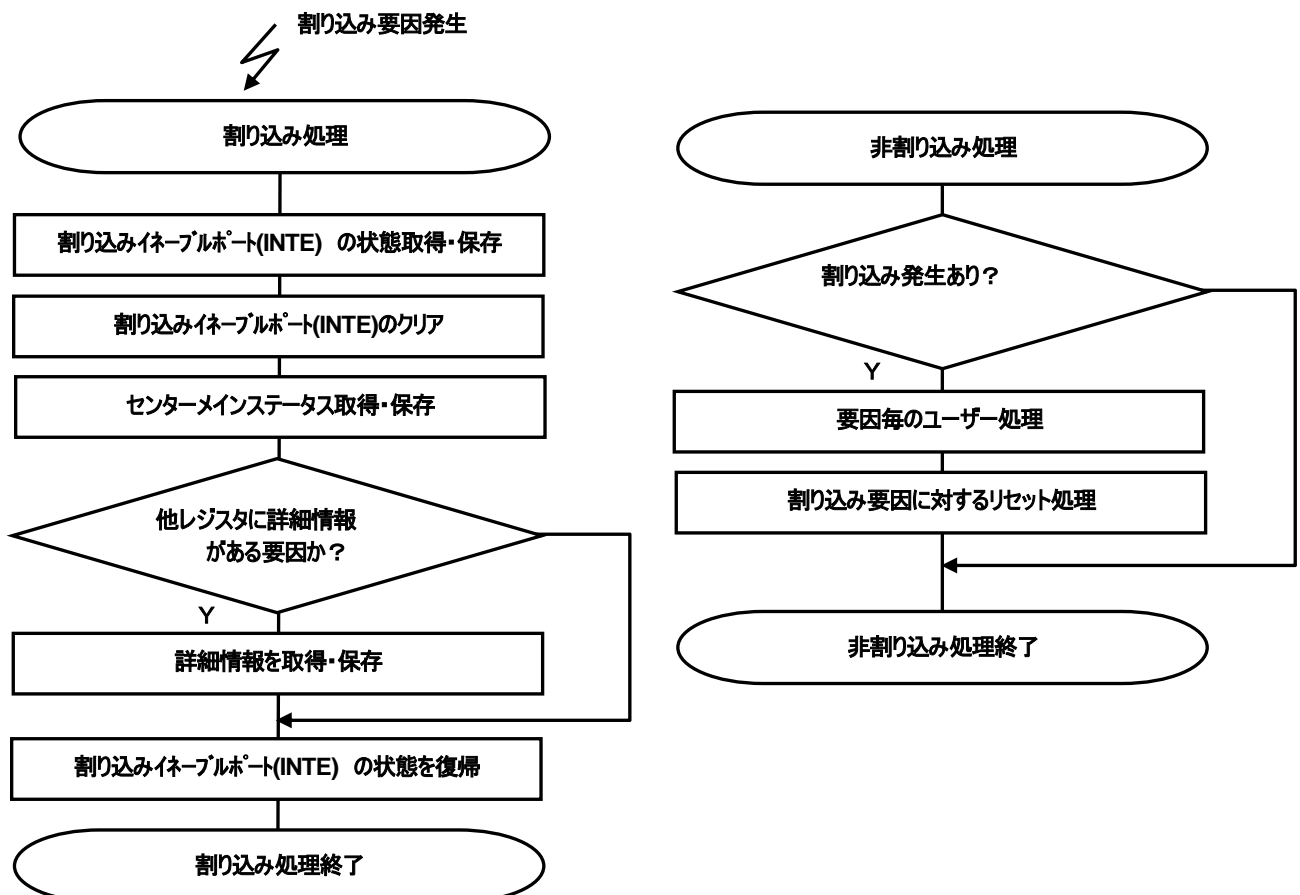


図 5.5-2 割り込み発生時の処理フロー