

モーション制御用省配線システム

PC/104 規格 motionCAT Series Master Board

HPC104-MCAT110M

ユーザーズマニュアル

Motion by Control Automation Technology



株式会社ハイバーテック

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

このマニュアルは
次のボードに適応しています。

H P C 1 0 4 - M C A T 1 1 0 M

本書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。

Windows98, Windows2000, WindowsXP Home Edition, WindowsXP Professional, VisualC++, Visual Basic,
Microsoft C/C++ MS-DOS はMicrosoft Corporation の米国及びその他の国における登録商標
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 - テック
東京都江東区新大橋 1 - 8 - 11
三井生命新大橋ビル 6 F
T E L 03 - 3846 - 3801
F A X 03 - 3846 - 3773
sales@hivertec.co.jp

第 1 . 1 1 版 2 0 0 8 年 1 0 月 8 日 発行

不許複製・転載

保証範囲

1. 本シリーズ製品の保証期間は、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、本シリーズ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、2次的な損害あるいは、遺失利益の損害に付いて、弊社は本シリーズ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。

免責事項

1. 本マニュアル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. 本シリーズ製品は、一般電子機器用（工作機械・計測機器・F A / O A 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。
ゆえに本シリーズ製品の安全性、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. 本マニュアル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。

安全にお使い頂くために

この度は、弊社motionCATシリーズ製品をご採用頂きまして、誠に有り難う御座います。

本書は、本シリーズ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、本書が添付された本シリーズ製品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

本シリーズ製品のご使用前に、必ずこのユーザ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下さい。本シリーズ製品の知識、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下さい。

本ユーザズマニュアルでは、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. 対象ユーザー





注意





本シリーズ製品およびマニュアルは、以下の様な、ユーザーを対象としています。

- ・拡張用ボードの増設および配線に付いて基本的な知識を有している方。
- ・制御用電子機器およびパソコン等に付いて基本的な知識を有している方。








2 . 適合 B u s

 警 告	
	H P C 1 0 4 - M C A T 1 1 0 MはPC/104 Bus Ver2.5 適合したボードです。 PC/104 Bus Ver2.5 が動作する環境以外では使用しないでください。

3 . 環境条件

 警 告	
	本シリーズ製品は、下記の環境条件下で保管・ご使用ください。 <ul style="list-style-type: none">・ 動作周囲温度 0 ~ + 5 0・ 動作周囲湿度 2 0 % R H ~ 8 5 % R H (結露せぬこと)・ 保存周囲温度 - 1 5 ~ + 7 5・ 保存周囲湿度 1 0 % R H ~ 9 0 % R H (結露せぬこと)・ 雰 囲 気 腐食性ガス・引火性ガス・オイルミスト・塵埃のないこと・ 標 高 海拔3 0 0 0 m以下 (3 0 0 m毎に2 の上限値を下げた範囲で使用してください)










4 . 適合ケーブルおよび通信システム

 警 告	
	Motionnet に適合した製品以外と通信ケーブルを接続しないでください。
	通信ケーブルはシールド付きLANケーブルCAT5eまたはCAT6を使用してください。 これら以外の通信ケーブルを使用すると誤動作の原因となります。
	スレーブ間またはマスターとスレーブ間のケーブル長は6 0 c m以上にしてください。 ケーブルが短すぎると誤動作の原因となります。
	合計の伝送距離は5 0 m以内にしてください。伝送距離が長すぎると誤動作の原因となります。
	1 系統に接続するモジュールの数は3 2 個以内にしてください。3 3 個以上を接続した場合、 誤動作の原因となります。
	1 系統に接続するモジュールのモジュールID (アドレス) は重複しないようにしてください。 モジュールIDが重複すると誤動作および故障の原因となります。




5 . 運搬・取り付け

 警 告	
	本製品にふれる前に、金属に触り身体の静電気を取り除いてください。 静電気は、本ボードの故障の原因になります。
	本製品を静電気の帯びやすい梱包材（エアーキャップなど）でくるまないでください。 静電気は、本ボードの故障の原因になります。
	本製品のコネクタピン部分に触らないで下さい。 コネクタピン部分が汚れますと、誤動作の原因になります。
	本製品の上に重いものを載せないで下さい。 重いものを乗せますと、部品が損傷し故障の原因になります。
	本製品のジャンパー設定は、スタッキング前に行ってください。 電源がONの状態 設定しますと、設定を正しく認識しないで誤動作の原因になります。
	本製品のジャンパー設定は、正しく行ってください。 設定を間違えますと 誤動作の原因 になります。
	本製品をスタッキングする時は、必ず電源をOFFにし、電源コードを抜いてから作業 を行ってください。 電源コードを抜かないで作業を行った場合、故障の原因になります。また、装置が思わ ぬ動作をすることがあります。
	本製品をスタッキングする時は、ボードが平行になるように、金メッキ部分のコネクタ ピンを深く挿入してください。 ボードが斜めに取り付けられたり、挿入が浅かったりし ますと、接触不良などにより誤動作、故障の原因になります。
	本製品をスタッキングする時は、取り付け金具を、取り付けネジにより確実に固定して ください。 取り付けが不十分ですと誤動作の原因になります。
	本製品を落としたり乱暴に扱わないでください。 衝撃や振動が故障の原因となります。
	本製品の半田面を手で直接触らないでください。 部品の突起などにより怪我をする恐れがあります。



6. 配 線

 警 告	
	外線用コネクタへの配線作業や外線用コネクタの着脱は、パソコン等の電源をOFFし電源コードを抜いてから行ってください。 電源コードを抜かないで作業を行った場合、故障の原因になります。また、装置が思ぬ動作をすることがあります。
	外線用コネクタへの配線は、コネクタ信号表などをよく確認し、正しく配線してください。間違った配線をしますと、故障・焼損の原因になります。
	外部から供給する電源は、必ず定格以内でご使用下さい。定格以外で使用されますと、故障・焼損・誤動作の原因となります。
	入出力回路に接続する回路は、必ず定格電流・電圧以内でご使用ください。定格以外で使用されますと、故障・焼損・誤動作の原因となります。
	外部配線用コネクタは、推奨のコネクタをご使用ください。推奨以外のコネクタを使用されますと、接触不良などにより誤動作の原因となります。
	外部配線用コネクタは、必ずロックしてご使用ください。ロックしないで使用されますと、コネクタが外れる、または接触不良などにより誤動作の原因となります。
	外部配線用ケーブルは、引っ張る、または重い荷重を掛けしないでください。コネクタが外れる、または接触不良などにより誤動作の原因となります。
	外部配線用ケーブルは、モータの配線やAC電源ケーブルなど、ノイズの多い配線とは出来るだけ離してください。配線が近いとノイズが 誤動作の原因となります。

7. 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。

8. 廃 棄

 警 告	
	本シリーズ製品を廃棄する時は、関連する法律・規則に従って処理してください。

【 用語解説 】

(1) motionCATシリーズ製品用語

用 語	説 明
システム通信	この通信により各ローカルデバイスの接続状態，デバイスの種類，および入出力ポートの設定等を取得する．
サイクリック通信	ローカルデバイスとのサイクリックな通信（番号の若いローカルデバイスから順番に通信を行い，最後の番号との通信が終了したら，再度若い番号からのデバイスとの通信を開始する）．入出力ポートの状態を入出力する．
データ通信	この通信はモーションデバイス（G9003）等とデータの通信を行う．センターデバイスでは，サイクリック通信をサイクリック的に連続して行うが，CPUからのデータ通信コマンドにより，サイクリック通信に割込むかたちでデータ通信を行う．サイクリック通信を開始していないとデータ通信はできない．
セン タ ー デ バ イ ス （G9001A）	motionCATシステムのローカルデバイスと通信を行うデバイス．
ローカルデバイス	センターデバイスにより制御されるD I Oデバイス（G9002）とモーションデバイス（G9003）の総称
D I Oデバイス（G9002）	入出力ポートを4個持ったD I Oデバイス．センターデバイスより制御される．
モ ー シ ョ ン デ バ イ ス （G9003）	1軸のモーション制御を行うモーションデバイス． センターデバイスからの通信による各種コマンドにより制御される．
L i n e	センターデバイスのシリアル通信ライン，通信回線，伝送ライン，系統．
ライン番号	マスターボードに搭載されているセンターデバイスの通信ラインの識別番号
ローカルデバイス情報	システム通信により取得される1ライン上にあるローカルデバイスの情報
モジュールID	モジュールを識別するためのID．00h～3Fh（0～63）まで設定可能．
マスターボード	センターデバイスを搭載したボード． motionCATシステムの通信の制御はこのボードで行う．
スレーブ	通信ボードと複数のモジュールで構成される．
モジュール	スレーブを構成する基板．D I Oモジュールとモーションモジュールがある．
D I Oスレーブ	通信ボードとD I Oモジュールで構成されるスレーブ
モーションスレーブ	通信ボードとモーションモジュールで構成されるスレーブ
複合スレーブ	通信ボードとモーションモジュールとD I Oモジュールで構成されるスレーブ
D I Oモジュール	D I Oデバイスを搭載した基板
モーションモジュール	モーションデバイスを搭載した基板

表1 motionCATシリーズ製品用語

(2) モーションモジュール用語

用 語	説 明
制御軸	このモーションモジュールで制御されるモータ(サーボモータ, パルスモータなど)は「制御軸」と云い他の種モータ(例えばスピンドル回転のモータ)とは制御対象を別に表現する.
連続送り	停止位置は指定せず, 速度のみ指示して起動する. 停止コマンドによって停止する.
位置決め制御	途中の経路は問題とせず, 指定目標位置に移動後停止する制御を云う.
相対座標 (インクリメンタル)	現在点を相対的な原点と考え目標位置をその原点から見た座標値と考える. モーションモジュールでは送り座標はすべて相対座標で指示する.
移動単位, 速度単位	移動単位, すなわち座標値は[パルス], 速度は[パルス/sec]すなわち[PPS]で表現する.
動作速度	位置決め, 原点復帰, 連続送りなどの指示速度は総称して「動作速度」と言い, F Hレジスタ(RFH)の値×速度倍率が動作速度となる.
ベース速度	ベース速度はF Lレジスタ(RFL)と速度倍率設定レジスタ(RMG)で規定され, 起動時は停止から直接ベース速度に達する. 停止時はこの速度から即停止する. 加速は動作速度がベース速度以上の時に行われ, 減速は動作速度がベース速度になったとき終了する.(自動加減速時)
補助速度	一部の原点復帰において, 原点突入速度に使われる. あるいはバックラッシュ動作, スリップ動作時の速度に使われる. 図1に速度パターン図を示す.
軸センサ	モータにより直動するテーブル機構の例を図2に示す. 図中の「+ELS, -ELS, DLS, OLS」を軸センサとよぶ. 「±ELS」 極限センサ(Stroke End - Limit Sensor)は直動軸の両端に配置される. 移動方向のELSがキャリッジを検出した時, モーションモジュールは指令パルス出力を即停止, あるいは減速して止める. ELSの検出状態ではELSの逆方向にのみ移動できる. ±は座標の方向と一致する. 原点復帰方法の選択により, ELSをセンサ原点と兼用することもできる. 「OLS」 センサ原点 あるいは 高速原点復帰時の減速センサの機能を果たす. いずれの機能に使用するかは原点復帰方法の選択による. 「DLS」 DLSは減速センサ. DLS検出でベース速度に減速する. モーションモジュールでは原点復帰時の減速センサの機能としてOLSが多く使われる. 「Z相」 原点としてエンコーダ(ENC)のZ相信号を使用する場合に使われる. OLSにより減速を開始して, 指定したn回目(nは1~15)のZ相で原点完了となる. ENC Z相の代わりに, 1回転に1パルスを発生するように回転軸に設けたフォトインタラプタなどもZ相信号に使用できる.

(次ページへ続く)

(前ページからの続き)

用 語	説 明
外部同期入出力	<p>モーションモジュールでは外部と同期をとるためにL T C H信号，C L R信号，C M P出力があります．</p> <p>「L T C H」 L T C H入力によりカウント値をラッチすることができる． モーションモジュールのD I P S Wで入力可にする．</p> <p>「C L R」 C L R入力によりカウント値をクリアすることができる． モーションモジュールのD I P S Wで入力可にする．</p> <p>「C M P出力」 指定したカウンタと設定したコンパレータデータを比較して，コンパレータ条件成立時に外部へ出力する． また一定カウントごとに出力する「同期出力」機能もある． (コンパレータ3のみ)</p>
指令パルスと モータドライバ	<p>「指令パルス」 モータドライバへの出力を云う．指令パルス列信号がサーボドライバまたはパルスモータドライバに与えられる． 指令1パルスは，パルスモータの場合はモータを1ステップ角回転させる．サーボモータの場合は，位置の検出器（通常モータに組込まれているエンコーダ）を1パルス分を回転させる． したがって「指令パルス」1パルスは位置(移動量)を有し， 「指令パルス周波数」は回転速度となる．</p> <p>「ドライバ」 モーションモジュールに接続されるモータドライバはパルス列入力形式であれば適合する．したがってモータドライバがパルス列入力形式であればモータはサーボモータ，リニアモータあるいはパルスモータでも制御対象となる．</p> <p style="text-align: center;">図3 指令パルスとモータ・モータドライバ</p>
速度のオーバー ライド	動作中に，動作速度F Hレジスタ（R F H）に速度データ書き込みを行うことで，動作速度の変更を行う．
位置のオーバー ライド	<p>位置のオーバーライドは，当初位置決め送りを(例えば移動量10，000)スタートさせた後終点に達する前に，異なる終点に変更する動作を云う． 動作の様子は2通りある．</p> <p>(1) 当初の終点値より遠方に変更する場合．</p> <p>(2) 当初の終点値より手前に変更した場合．</p> <p>(2)の場合は指示された時点で停止し，逆方向へ戻り，新たな終点に向う．</p>
バックラッシュ 動作， スリップ動作	<p>バックラッシュ補正は動作方向が変化する毎に，指令動作の直前に予め設定してある補正量が挿入される．</p> <p>スリップ動作は動作方法に無関係に補正量が入る．</p> <p>スリップ動作は(送り滑りのあるような)フリクションドライブ機構などに有用．</p>

表2 モーションモジュール用語

目 次

1 . はじめに	1
2 . motionCAT(Motion by Control Automation Technology)とは	1
2 . 1 motionCAT シリーズ (P C / 1 0 4 タイプ) 製品の特長	1
2 . 2 motionCATスレーブの接続ルール	2
2 . 3 通信仕様	2
2 . 4 motionCATシリーズ製品スレーブの構成	2
2 . 5 制御概要	3
2 . 5 . 1 制御概要	3
(1) motionCATの構成	3
(2) モジュール I D	4
2 . 5 . 2 通信の概要	5
2 . 5 . 3 通信時間の計算方法	8
2 . 5 . 4 通信ケーブル断線検出・監視機能	9
3 . マスターボードハードウェア編	10
3 . 1 ブロック図	10
3 . 2 ポートアドレス	10
4 . センターデバイス (G 9 0 0 1 A) ・オプションポート	12
4 . 1 センターデバイス (G 9 0 0 1 A)	12
4 . 1 . 1 ローカルデバイス情報 (D I N F O)	12
4 . 1 . 2 サイクリック通信エラーフラグ (I O E R R)	13
4 . 1 . 3 入力ポート変化フラグ設定 (I I N T)	14
4 . 1 . 4 入力ポート変化フラグ (I R E S)	15
4 . 1 . 5 ポートデータ (D A T A)	16
4 . 1 . 6 センターメインステータス (C M S T S)	17
4 . 1 . 7 センター割込ステータス (C I S T S)	18
4 . 1 . 8 センターデバイスコマンド (C C M D)	19
4 . 1 . 9 センターデバイスレジスタ	22
4 . 2 オプションポート	24
4 . 3 ボード上の設定及びコネクタ位置	26
4 . 3 . 1 H P C 1 0 4 - M C A T 1 1 0 M 設定箇所	26
4 . 3 . 2 マスターボードの設定	26
4 . 3 . 3 汎用入出力と E M G 入力回路	29
4 . 4 コネクタ信号表	30
4 . 5 マスターボード仕様	31
4 . 5 . 1 マスターボード H P C 1 0 4 - M C A T 1 1 0 M ハードウェア仕様	31
4 . 5 . 2 H P C 1 0 4 - M C A T 1 1 0 M マスターボード形寸	32
5 . ローカルデバイス	33
5 . 1 D I O デバイス (G 9 0 0 2)	33
5 . 1 . 1 D I O モジュール I / O ポート	33
5 . 2 モーションデバイス (G 9 0 0 3)	33
5 . 2 . 1 特 長	33
5 . 2 . 2 モーションモジュール I / O ポート	35
5 . 2 . 3 モーションメインステータス (M M S T S)	35
5 . 2 . 4 汎用出力状態ポート (I O P I B)	36
5 . 2 . 5 汎用出力ポート (I O P O B)	36
5 . 2 . 6 モーション動作コマンド・コントロールコマンド・レジスタ制御コマンド	36
5 . 2 . 7 モーションデバイスレジスタ	40
5 . 2 . 8 モーションデバイス環境設定レジスタ	49
6 . ソフトウェア編	55
6 . 1 ソフトウェアの構成	55
6 . 1 . 1 W i n d o w s 版のソフトウェア構成	55
6 . 1 . 2 ソフトウェアの構成	56
6 . 1 . 3 ソフトウェアの使用方法	56
6 . 2 W i n d o w s 用デバイスドライバのインストールとアンインストール	56
6 . 2 . 1 W i n d o w s 用デバイスドライバのインストール	56
6 . 2 . 2 W i n d o w s 用デバイスドライバのアンインストール	56
6 . 3 アプリケーション構築準備	57
6 . 4 ボードアクセス方法	58
6 . 4 . 1 デバイス情報構造体	58

6.4.2	ボードアクセス準備手順と終了処理	59
6.5	ドライバ関数	60
6.5.1	ドライバ関数一覧	60
6.5.2	ドライバ関数の戻り値一覧	61
6.5.3	ドライバ関数仕様	62
6.6	ライブラリ関数	72
6.6.1	ライブラリ関数一覧	72
6.6.2	ライブラリ関数の戻り値	73
6.6.3	ライブラリ関数詳細	74
6.7	添付ソフトウェア	88
6.7.1	「動かしてみる」	88
6.8	DOS版添付ソフトウェア	93
6.8.1	Windows版ソフトウェアとの違い	93
6.8.2	ソフトウェア構成	93
6.8.3	デバイスドライバ	93
6.8.4	割り込み	94
6.8.5	割り込み使用手順	95
6.8.6	DOS用ドライバ関数のデータ型と構造体の定義	97
6.8.7	ドライバ関数の戻り値	98
6.8.8	ドライバ関数仕様	99
6.8.9	ドライバ関数を割り込み環境で使用する場合の注意事項	104
6.8.10	ライブラリ関数	104
6.8.11	DOS版サンプルプログラム	105
7.	モーションモジュールの運用	116
7.1	プログラム手順	116
7.1.1	マスターボード初期設定	116
7.2	モーションモジュール(デバイス)初期設定	117
7.2.1	マシンセンサー入力仕様設定	117
7.2.2	サーボインターフェース設定	118
7.2.3	原点復帰方法設定	119
(1)	原点センサ構成と原点復帰方法	120
7.2.4	イベント要因の設定	122
7.3	モーションモジュール(デバイス)の運用	122
7.3.1	動作モード	122
7.3.2	速度パターン設定	130
7.3.3	速度パターン	131
7.3.4	速度パターン設定レジスタ	132
7.4	ステータス処理	137
7.5	応用	139
7.5.1	STA入力を利用した同時スタート	139
7.5.2	グループスタート	139
7.5.3	コンパレータ設定	140
7.5.4	ソフトリミット設定	142
7.5.5	定ピッチ出力設定	143
7.5.6	コンパレータ3を利用した同一スレーブ内他モーションモジュールスタート	144

図 表 目 次

表 2 . 1 - 1	伝送周期	1
表 2 . 3 - 1	通信仕様	2
表 2 . 4 - 1	motionCAT製品構成・型式.....	2
図 2 . 5 - 1	motionCAT構成	3
図 2 . 5 - 2	HMG型スレーブ構成	3
表 2 . 5 - 1	デバイス種別	3
図 2 . 5 - 3	スレーブの構成・名称とモジュールID	4
図 2 . 5 - 4	サイクリック通信のワンセット.....	5
表 2 . 5 - 2	サイクリック通信で取り扱うデータ.....	5
表 2 . 5 - 3	ポートデータアドレス	6
図 2 . 5 - 5	データ通信	7
図 2 . 5 - 6	データ通信は連続しない.....	7
図 2 . 5 - 7	データ通信コマンドとシステム通信コマンドのビット構成	7
表 2 . 5 - 4	データ通信コマンドとシステム通信コマンドの内容	7
図 2 . 5 - 8	断線検出	9
図 3 . 1 - 1	ブロック図	10
表 3 . 2 - 1	ボードアドレス(8バイトモード).....	10
表 3 . 2 - 2	ボードアドレス(512バイトモード).....	11
表 4 . 1 - 1	ローカルデバイス情報アドレス.....	12
表 4 . 1 - 2	サイクリック通信エラーフラグアドレス.....	13
表 4 . 1 - 3	入力ポート変化フラグ設定アドレス.....	14
表 4 . 1 - 4	入力ポート変化フラグアドレス.....	15
表 4 . 1 - 5	ポートデータアドレス	16
図 4 . 1 - 1	センターメインステータス(CMSTS)のビット構成	17
表 4 . 1 - 6	センターメインステータス(MSTS)の内容	17
図 4 . 1 - 2	センター割込ステータス(CISTS)のビット構成	18
表 4 . 1 - 7	センター割込ステータス(CISTS)の内容	18
表 4 . 1 - 8	センター動作コマンドの内容.....	20
表 4 . 1 - 9	センターレジスタ制御コマンドの内容	20
表 4 . 1 - 10	ローカルデバイス制御コマンドの内容	21
図 4 . 1 - 3	RENV0レジスタのビット構成	22
表 4 . 1 - 11	RENV0レジスタの内容	22
図 4 . 1 - 4	RSYCNT	23
図 4 . 1 - 5	RDJADD	23
図 4 . 1 - 6	RVER	23
図 4 . 2 - 1	汎用入力ポートのビット構成.....	24
表 4 . 2 - 1	汎用入力ポートの内容	24
図 4 . 2 - 2	汎用出力設定及び出力状態確認ポートのビット構成	24
表 4 . 2 - 2	汎用出力設定及び出力状態確認ポートの内容	24
図 4 . 2 - 3	G9001Aステータス確認ポートのビット構成	24
表 4 . 2 - 3	G9001Aステータス確認の内容	24
図 4 . 2 - 4	G9001A通信速度確認ポートのビット構成	24
表 4 . 2 - 4	G9001A通信速度確認ポートの内容	24
図 4 . 2 - 5	割込みイネーブルポートのビット構成	25
表 4 . 2 - 5	割込みイネーブルポートの内容	25
図 4 . 2 - 6	割込み確認ポートのビット構成	25
表 4 . 2 - 6	割込み確認ポートの内容	25
図 4 . 2 - 7	設定スイッチ確認ポートのビット構成	25
表 4 . 2 - 7	設定スイッチ確認ポートの内容	25
図 4 . 3 - 1	HPC104-MCAT110Mコネクタ, スイッチ位置	26
図 4 . 3 - 2	MCAT110M 割込ジャンパ端子.....	26
図 4 . 3 - 3	通信速度SW	26
表 4 . 3 - 1	アドレス, バス幅, 使用空間モード, アドレス空間の設定スイッチ機能説明	27
表 4 . 3 - 2	設定スイッチによるアドレス スイッチの有効/無効状態	27
図 4 . 3 - 6	非常停止 有効入力	28
図 4 . 3 - 7	非常停止 12/24V切替え.....	28
表 4 . 3 - 2	汎用入出力回路とEMG入力回路	29

図4.4-1	PC/104 Bus コネクタ信号表	30
表4.4-1	RJ45 (J4) コネクタ信号表	30
表4.4-2	DIO部ピンアサイン (J3)	31
表4.5-1	HPC104-MCAT110M 仕様	31
図4.5-2	HPC104-MCAT110Mボード形状	32
図5.1-1	DIOモジュールポート割付	33
図5.2-1	モーションモジュールポート割付	35
図5.2-2	モーションメインステータス (MMSTS) のビット構成	35
表5.2-1	モーションメインステータス (MMSTS) の内容	35
図5.2-3	汎用出力状態ポート (IOPIB) のビット構成	36
表5.2-2	汎用出力状態ポート (IOPOB) の内容	36
図5.2-4	汎用出力設定ポート (IOPOB) のビット構成	36
表5.2-3	汎用出力ポート (IOPOB) の内容	36
表5.2-4	動作コマンド	37
表5.2-5	速度変更コマンド	37
表5.2-6	停止コマンド	37
表5.2-7	停止コマンド	37
表5.2-8	モーションコントロールコマンド	38
表5.2-9	モーションレジスタ書き込みコマンド	38
表5.2-10	モーションレジスタ読み出しコマンド	39
図5.2-5	RMV: 移動量レジスタのビット構成	40
図5.2-6	RFL: ベース速度レジスタのビット構成	40
図5.2-7	RFH: 動作速度レジスタのビット構成	40
図5.2-8	RUR: 加速レートレジスタのビット構成	41
図5.2-9	RDR: 減速レートレジスタのビット構成	41
図5.2-10	RMG: 速度倍率設定レジスタのビット構成	41
図5.2-11	RDP: 減速開始点レジスタのビット構成	41
図5.2-12	RMD: 動作モードレジスタのビット構成	42
表5.2-11	RMD: 動作モードレジスタの内容	42
図5.2-13	RUS: 加速S字区間レジスタのビット構成	43
図5.2-14	RDS: 減速S字区間レジスタのビット構成	43
図5.2-15	RFA: 補助速度レジスタのビット構成	43
図5.2-16	RIRQ: イベント要因設定レジスタのビット構成	44
表5.2-12	RIRQ: イベント要因設定レジスタの内容	44
図5.2-17	RIST: イベントステータスレジスタのビット構成	44
表5.2-13	RIST: イベントステータスレジスタの内容	44
図5.2-18	REST: エラーステータスレジスタのビット構成	45
表5.2-14	REST: エラーステータスレジスタの内容	45
図6.2-19	RSTS: 拡張ステータスレジスタのビット構成	46
表5.2-15	RSTS: 拡張ステータスレジスタの内容	46
図5.2-20	RCTR1: 指令位置カウンタレジスタのビット構成	46
図5.2-21	RCTR2: 機械位置カウンタレジスタのビット構成	47
図5.2-22	RCTR3: 汎用・偏差カウンタレジスタのビット構成	47
図5.2-23	RCMP1~3: コンパレータ1~3レジスタのビット構成	47
図5.2-24	RLTC1: カウンタ1ラッチレジスタのビット構成	47
図5.2-25	RLTC2: カウンタ2ラッチレジスタのビット構成	47
図5.2-26	RLTC3: カウンタ3ラッチレジスタのビット構成	48
図5.2-27	RPLS: 移動残パルスレジスタのビット構成	48
図5.2-28	RSPD: 速度モニタレジスタのビット構成	48
表5.2-16	RSPD: 速度モニタレジスタの内容	48
図5.2-29	RSDC: 減速開始点自動演算値レジスタのビット構成	48
図5.2-30	RENV1: 環境設定1レジスタのビット構成	49
表5.2-17	RENV1: 環境設定1レジスタの内容	49
図5.2-31	RENV2: 環境設定2レジスタのビット構成	51
表5.2-18	RENV2: 環境設定2レジスタの内容	51
図5.2-32	RENV3: 環境設定3レジスタのビット構成	51
表5.2-19	RENV3: 環境設定3レジスタの内容	52
図5.2-33	RENV4: 環境設定4レジスタのビット構成	53
表5.2-20	RENV4: 環境設定4レジスタの内容	53
図5.2-34	RENV5: 環境設定5レジスタのビット構成	54
表5.2-21	RENV5: 環境設定5レジスタの内容	54

図5.2-35	RENV6：環境設定6レジスタのビット構成	54
表5.2-22	RENV6：環境設定6レジスタの内容	54
図5.2-36	振動抑制用逆転パルス	54
図6.1-1	ソフトウェアの関連図	56
図6.2-1	デバイスパラメータの設定画面	56
表6.5-1	ドライバ関数一覧	60
表6.5-2	ドライバ関数の戻り値	61
表6.6-1	ライブラリ関数一覧	72
表6.6-2	ライブラリ関数の戻り値	73
表6.6-3	モーションモジュールのレジスタの初期値	74
図6.7-1	起動画面	88
表6.7-1	「動かしてみる」G9003レジスタ初期値	89
図6.7-2	モーションモジュール操作画面	90
図6.7-3	信号入力状態	90
図6.7-4	動作パラメータ設定	91
図6.7-5	動作開始ボタン	91
図6.7-6	DIOモジュール操作画面	92
図6.8-1	ソフトウェアの関連図	93
図6.8-2	ボード内の割込みルート	94
図6.8-3	初期化手順フロー	95
図6.8-4	割り込み発生時の処理フロー	96
表6.8-3	ドライバ関数の戻り値	98
図7.1-1	プログラム基本手順	116
表7.2-1	モーションデバイス初期設定項目	117
表7.2-2	マシンセンサー入力仕様設定	117
表7.2-3	サーボインターフェース設定	118
表7.2-4	原点復帰方法設定要件	119
表7.2-5	原点復帰方法	120
表7.2-6	原点復帰方法（CTR2参照方式）	121
図7.2-1	RIRQ：イベント要因設定レジスタのビット構成	122
表7.3-1	動作モード一覧	122
表7.3-2	コマンドによる連続動作モード	123
表7.3-3	位置決め動作	124
表7.3-4	原点復帰動作設定項目と関連ステータス	124
表7.3-5	原点復帰動作	125
表7.3-6	ELS, SLS動作	126
表7.3-7	Z相移動モード	126
表7.3-8	FH速度[pps]とパルス入力周波数FP[pps]との関係例	128
表7.3-9	パルスA/B相入力関係の設定	129
表7.3-10	手動パルス動作	129
表7.3-11	速度パターン設定レジスタ	130
図7.3-1	加減速動作時のレジスタ使用箇所	130
表7.3-12	速度パターン	131
表7.3-13	速度倍率設定例	134
図7.4-1	モーションメインステータス（MMSTS）のビット構成	137
表7.4-1	モーションメインステータス（MMSTS）の内容	137
図7.4-2	RIST：イベントステータスレジスタのビット構成	137
図7.4-3	REST：エラーステータスレジスタのビット構成	138
表7.4-2	REST：エラーステータスレジスタの内容	138
図7.4-4	ステータス運用例	138
図7.5-1	ソフトウェアリミット設定例	142
図7.5-2	定ピッチ出力	143
図7.5-3	モーションモジュール上面	144

1 . はじめに

このマニュアルは、motionCATシリーズ製品の

(1) motionCATの特長・仕様・構成

(2) motionCATマスターボード「HPC104-MCAT110M」のハードウェア等についての取扱説明書です。

motionCATシリーズ製品のスレーブの接続・設定等については各々のスレーブマニュアルを参照してください。

尚、この説明書ではHPC104-MCAT110MをMCAT110Mと呼びます。

設定頂く、ベースアドレスをBARと表記します。

2 . motionCAT(Motion by Control Automation Technology)とは

motionCATは、Motionnet®を使用したモーション制御用省配線システムです。

特に接続されるモータードライバーはメーカーを選ばず、接続できます。

通信マスターは、複数のスレーブとLANケーブルで接続され、マルチドロップ形式で通信制御します。

. Motionnet は日本パルスモーター株式会社の登録商標です。日本パルスモーター製 G9000 シリーズを使用し、LANケーブルを介して高速通信によりモータ制御、I/O制御を行います。

2 . 1 motionCATシリーズ(PC / 104タイプ) 製品の特長

(1) 省配線システム 省コスト配線 一般のLANケーブルを使用しています。

(2) モータ制御として1系統に接続できるモータは32台が制御できます。

(3) I/O制御なら 32点×32台=1024点を制御できます。

(4) 1系統のLANケーブル長は最大50mです。

(5) モータ制御スレーブはパルス列入力形式のドライバーと接続します。

サーボモータ、パルスモータなどメーカーフリーです。

(6) CPDボードシリーズからのソフトウェアの移植が容易に出来ます。

(7) マスターボード上のI/Oポート用メモリが伝送周期ごとに自動的に更新され、I/O制御が容易にできます。

(8) 短い伝送時間、早いレスポンス 20Mbps

伝送周期は次表の通りです。

1系統の接続数	8台	16台	32台	記 事
伝送周期	0.12ms	0.24ms	0.49ms	データ通信時間 19.3us/6byte (モーションデバイスの1レジスタ書込み)

表 2 . 1 - 1 伝送周期

2.2 motionCATスレーブの接続ルール

MCAT110Mとの接続について、次の「通信ケーブル接続ルール」を守って下さい。

通信ケーブル接続ルール

- 使用ケーブル・・・・・・・・・・シールド付きLANケーブル CAT5e 又は CAT6を使用する。
- 各系統の総モジュール数・・・・・・・・32モジュール以下であること。
- 各系統のケーブル総延長・・・・・・・・50m以下であること。
- スレーブ間の最短ケーブル長・・・・・・・・60cm以上であること。
- 最遠端モジュールを終端処理・・・通信設定SW（終端設定）をONにする。
（各スレーブのユーザズマニュアル参照）
- LAN コネクタの出入り接続・・・・・・・・スレーブの2個のLAN コネクタの接続順はない。（接続順フリー）

2.3 通信仕様

項 目	内 容
基準クロック	40MHzまたは80MHz（製品により異なります）
通信速度	2.5M, 5M, 10M, 20Mbps
通信符号	NRZ符号
通信プロトコル	日本パルスモータオリジナル（Motionnet）
通信方式	半2重通信
通信I/F	RS-485, パルストランス結合
接続方式	マルチ・ドロップ接続
伝送距離	50m
ローカルデバイス接続個数	32個
サイクリック通信周期	20Mbps時 1ラインにローカルデバイスを 8個使用時・・・0.12ms 1ラインにローカルデバイスを16個使用時・・・0.24ms 1ラインにローカルデバイスを32個使用時・・・0.49ms
データ通信時間	3ワード（ ）通信時（モーションデータの1ワード書き込）・・・19.3μs
通信データ長	1～3ワード/フレーム
通信ケーブル断線対策	断線検出モーション停止機能有り

・1ワード=2バイト=16ビット

表2.3-1 通信仕様

2.4 motionCATシリーズ製品スレーブの構成

motionCATは、マスターボードからマルチドロップ形式で、複数のスレーブを接続できます。

MCAT110Mは通信ラインには最大32個のモジュールを接続できます。

スレーブは1つの通信ボードと、これに追加する1～4（6）枚のモジュールで構成されます。通信ボードは通信機能を有します。モジュールにはモーションモジュールとDIOモジュールがあります。

下表に弊社スレーブの製品構成・型式を示します。

種 別	製 品 名	型 式 名	記 事
モーションスレーブ	HMG 1軸モーションスレーブ（1軸×1）	HMG - P1	パルス列 位置決め
	HMG 2軸モーションスレーブ（1軸×2）	HMG - P2	
	HMG 3軸モーションスレーブ（1軸×3）	HMG - P3	
	HMG 4軸モーションスレーブ（1軸×4）	HMG - P4	
	HMG 5軸モーションスレーブ（1軸×5）	HMG - P5	
	HMG 6軸モーションスレーブ（1軸×6）	HMG - P6	
DIO スレーブ	HMG DIO 32スレーブ（16IN/16OUT×1）	HMG - D1	ケーブル絶縁型 汎用入出力
	HMG DIO 64スレーブ（16IN/16OUT×2）	HMG - D2	
	HMG DIO 96スレーブ（16IN/16OUT×3）	HMG - D3	
	HMG DIO128スレーブ（16IN/16OUT×4）	HMG - D4	
複合スレーブ	HMG 1軸 + DIO32 スレーブ	HMG - P1D1	上記の複合
	HMG 1軸 + DIO64 スレーブ	HMG - P1D2	
	HMG 1軸 + DIO96 スレーブ	HMG - P1D3	
	HMG 2軸 + DIO32 スレーブ	HMG - P2D1	
	HMG 2軸 + DIO64 スレーブ	HMG - P2D2	
	HMG 3軸 + DIO32 スレーブ	HMG - P3D1	

表2.4-1 motionCAT製品構成・型式

2.5 制御概要

2.5.1 制御概要

(1) motionCATの構成

motionCATシステムはマスターボードとスレーブで構成されます。
MCAT110Mマスターボードには通信回線（ライン）を制御するセンターデバイス（G9001A）が1個搭載され、1系統のラインを制御できます。また、ひとつのスレーブは通信ボードと、これに追加する1～4個のモジュールで構成されます。モジュールにはモーションモジュールとDIOモジュールがあります。
1個のモーションモジュールには1軸制御用のモーションデバイス（G9003）が1個搭載されています。
1個のDIOモジュールには16IN/16OUTのI/Oを制御するDIOデバイス（G9002）が1個搭載されています。
また、各モジュールに搭載されているモーションデバイスとDIOデバイスを総じてローカルデバイスと呼びます。

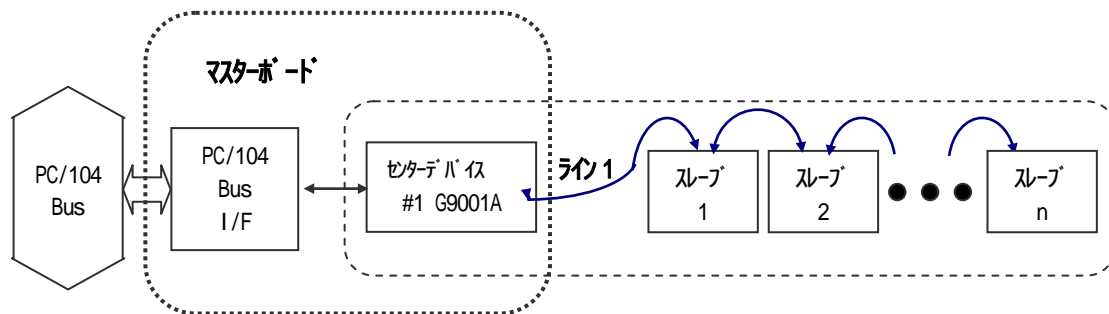


図2.5-1 motionCAT構成

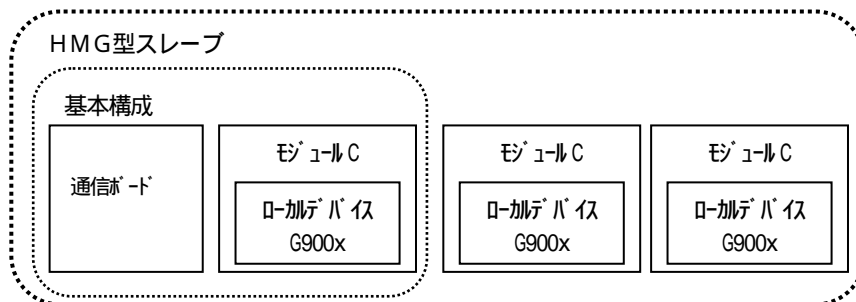


図2.5-2 HMG型スレーブ構成

デバイス名		型 名	機 能
センターデバイス		G9001A	1ラインに接続されている全てのローカルデバイスに対して通信制御を行う。
ローカルデバイス	モーションデバイス	G9003	1軸分のモーションコントロールを行う。
	DIOデバイス	G9002	入力2ポート、出力2ポートを持ち16IN/16OUTのI/O制御を行う

表2.5-1 デバイス種別

(2) モジュールID

- (1) モジュールID (M ID) はモジュールを識別するためのアドレス (ID) です。
 通信ボードの M ID ロータリSW で設定します。
 ベースとなるモジュールの 'M ID' を HEX 00 ~ 3C (0 ~ 60) で設定します。
 同一LINE内に接続するM IDは重複しないようにして下さい。
注意：ソフトウェアの関数で指定する 'M ID' を ロータリSWに設定します。

(2) 後続モジュールの 'M ID'

2以降のモジュールの 'M ID' は重ね順に + 1 したアドレスになります。

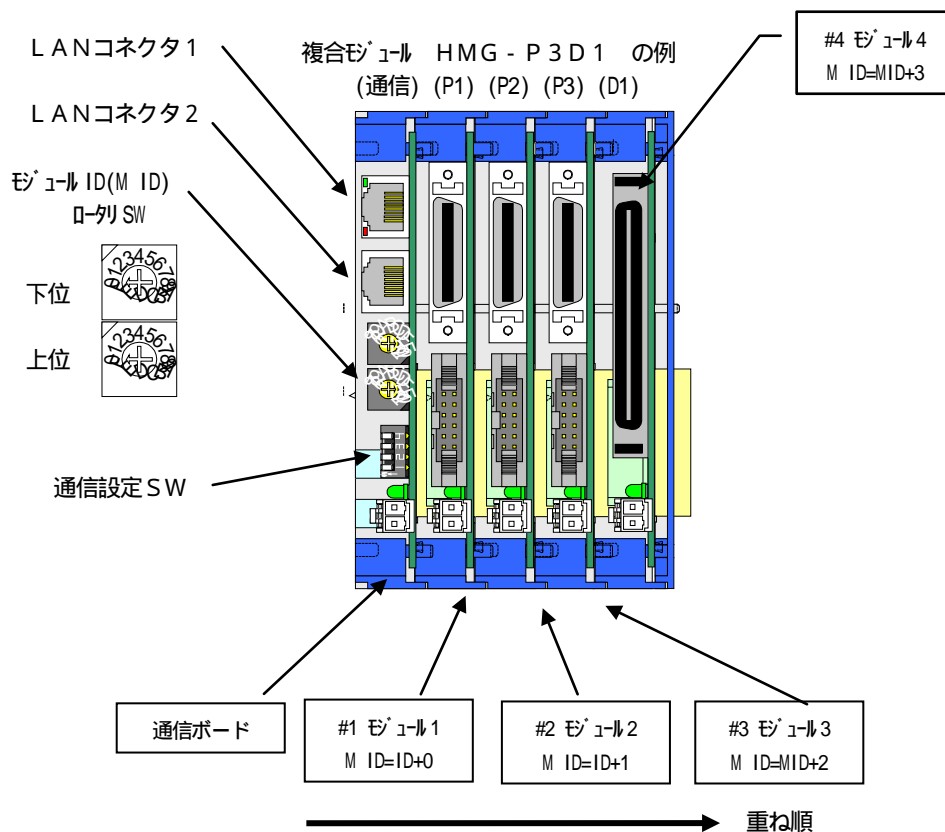


図2.5-3 スレーブの構成・名称とモジュールID

2.5.2 通信の概要

motionCAT はMotionnet通信を使用しています。
Motionnetの通信にはサイクリック通信とデータ通信があります。

(1) サイクリック通信

Motionnetでは常に通信回線の制御権はセンターデバイスが持っています。
各ローカルデバイスはセンターデバイスの許可がなければデータの送受信ができません。
センターデバイスは接続されている各ローカルデバイスのうち、一番若いモジュールIDを持つものから順番に通信を行います。まずセンターデバイスから、最初のローカルデバイスへ通信を行います。この通信を受け取れるデバイスは通信回線（1ライン）上にひとつしか存在しません（モジュールIDの重複は不可）。
そして通信を受け取ったローカルデバイスが、次に通信回線を使用する権利を得ることとなり、ローカルデバイスからセンターデバイスに向けて通信が行われます。これがワンセットの通信です。
センターデバイスはこのような応答を、モジュールID順にすべてのローカルデバイスに対して行います。
そして、最後のモジュールIDとの応答が終了すると、再度一番若いモジュールIDのローカルデバイスから順番に同じ処理を繰り返します。このように順番にすべてのローカルデバイスとの応答を繰り返す通信をサイクリック通信と呼びます。

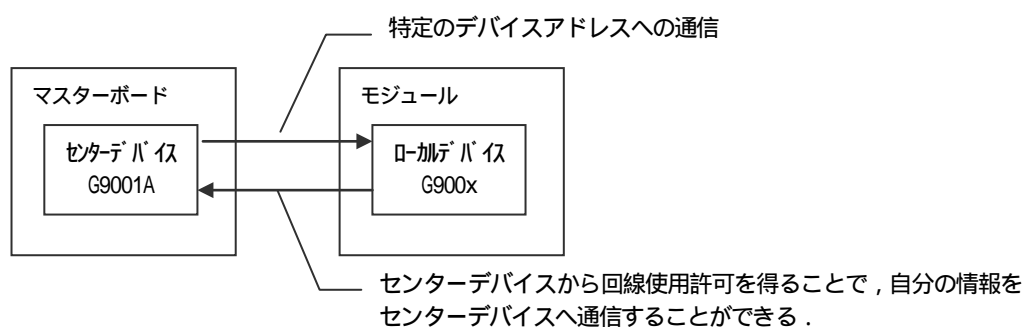


図2.5-4 サイクリック通信のワンセット

【 サイクリック通信で取り扱うデータ 】

サイクリック通信はソフトウェア側からの停止コマンドがない限り、常に行われます。
サイクリック通信でやりとりされるデータのサイズは4バイトです。
このサイズ内で通信されるデータは、ローカルデバイスごとに内容が異なります。

デバイス名称	センタからローカルへの通信内容	ローカルからセンタへの返信内容
G9002(DI0デバイス)	出力ポートへの出力データ	入力ポートから入力されたデータ
G9003 (モーションデバイス)	汎用出力ポートへの出力データ	汎用入力ポートからの入力データ モーションデバイスのステータス

表2.5-2 サイクリック通信で取り扱うデータ

通信時間については「2.5.3 通信時間の計算方法」で後述します。

【 サイクリック通信によるデータ取得とモジュールID 】

MCAT110MマスターボードにはI/Oポート用メモリ（ポートデータ）があります。
このメモリはサイクリック通信により、各モジュールの入出力ポートの状態に更新されます。
これにより、通常のメモリをアクセスするようにI/Oの状態を操作することができます。
下表にI/Oポート用メモリのアドレスマップ（512バイト、16bit設定時）を記します。

アドレス	読み込み（INP） ポートデータ（DATA）	書き込み（OUT） ポートデータ（DATA）
BAR+100	ライン1モジュールID 0ポート1,0データ	ライン1モジュールID 0ポート1,0設定
+102	ライン1モジュールID 0ポート3,2データ	ライン1モジュールID 0ポート3,2設定
+104	ライン1モジュールID 1ポート1,0データ	ライン1モジュールID 1ポート1,0設定
+106	ライン1モジュールID 1ポート3,2データ	ライン1モジュールID 1ポート3,2設定
+108	ライン1モジュールID 2ポート1,0データ	ライン1モジュールID 2ポート1,0設定
+10A	ライン1モジュールID 2ポート3,2データ	ライン1モジュールID 3ポート3,2設定
⋮		
+1FC	ライン1モジュールID 63ポート1,0データ	ライン1モジュールID 63ポート1,0設定
+1FE	ライン1モジュールID 63ポート3,2データ	ライン1モジュールID 63ポート3,2設定
⋮		
+300	ライン2モジュールID 0ポート1,0データ	ライン2モジュールID 0ポート1,0設定
+302	ライン2モジュールID 0ポート3,2データ	ライン2モジュールID 0ポート3,2設定
+304	ライン2モジュールID 1ポート1,0データ	ライン2モジュールID 1ポート1,0設定
+306	ライン2モジュールID 1ポート3,2データ	ライン2モジュールID 1ポート3,2設定
+308	ライン2モジュールID 2ポート1,0データ	ライン2モジュールID 2ポート1,0設定
+30A	ライン2モジュールID 2ポート3,2データ	ライン2モジュールID 3ポート3,2設定
⋮		
+3FC	ライン2モジュールID 63ポート1,0データ	ライン2モジュールID 63ポート1,0設定
+3FE	ライン2モジュールID 63ポート3,2データ	ライン2モジュールID 63ポート3,2設定

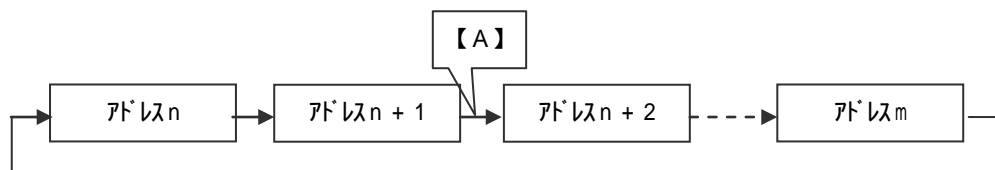
表 2 . 5 - 3 ポートデータアドレス

(2) データ通信

モーションデバイスに対する動作指示等はデータ通信で行います。

データ通信は、任意のタイミング（ソフトウェアからの通信指示のタイミング）で、サイクリック通信に割り込む形で行われます。（注1）

例えば以下のような順序でサイクリック通信が行われていると仮定します。



ここで“アドレスn”とは、モジュールID = nのローカルデバイスを表現しています。図中の矢印は、サイクリック通信が行われる順番を表します。ここで【A】のタイミングでデータ通信コマンドが発行されると、以下ようになります。

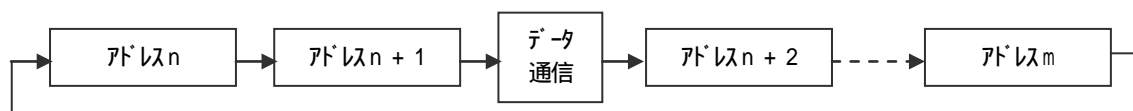


図2.5-5 データ通信

サイクリック通信の1周期に要する時間が延びますが、目的のデバイスとのやりとりは迅速に行えます。

データ通信も、サイクリック通信と同様に特定のローカルデバイスに対してデータが送られた後、そのローカルデバイスからの応答通信がセンターデバイスに対して送られます。

通信時間については「2.6.3 通信時間の計算方法」で後述します。

注1. 任意のタイミングと説明しましたが若干の制約があります。

データ通信を連続して行いたい場合や、通信エラーによりデータ通信をリトライする場合などは、必ず1回のサイクリック通信を挟んで行います。

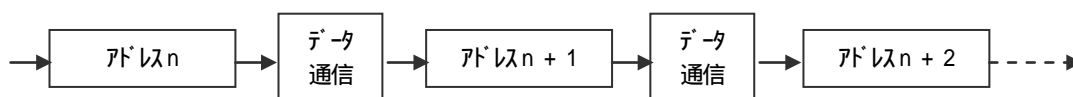


図2.5-6 データ通信は連続しない

【 データ通信コマンドとモジュールIDの指定 】

次表で指定したモジュールIDのデータ通信コマンドについて説明します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	0	0	0	モジュールID 00h~3Fh					

図2.5-7 データ通信コマンドとシステム通信コマンドのビット構成

コマンド種類	コマンド	説明
データ通信	0100 0000 00## ##### (4000h~403Fh)	指定したデバイスへ送信用FIFO内のデータを送信します。 応答データは受信用FIFOに格納されます。

注. #記号のビット: #ビットの上位から順に、モジュールIDの上位ビットを設定します。

表2.5-4 データ通信コマンドとシステム通信コマンドの内容

2.5.3 通信時間の計算方法

通信時間の計算は以下のように分けられます．

- (1) サイクリック通信に要する時間
- (2) 1回のデータ通信に要する時間
- (3) データ通信を含んだトータルのサイクリック時間

ただし，N：接続するローカルデバイス数

B：送信データのバイト数（2バイトデータ送信時：B = 2）

K：通信速度係数

通信速度(Mbps)	K
20	1
10	2
5	4
2.5	8

とします．

(1) サイクリック周期に要する時間

基本項目	所要時間 (μs)
1 ローカルデバイスあたりにかかる通信時間 (CT)	$7.7 \times K$

$$\text{サイクリック周期に要する時間 (μs)} = (CT + 7.4) \times N$$

例．通信速度 = 20 Mbps で，ローカルデバイスを30個接続の時のサイクリック時間

$$\text{サイクリック時間} = (7.7 \times 1 + 7.4) \times 30 = 453 \mu s$$

(2) 1回のデータ通信に要する時間

データ通信には以下の2タイプがあります．

- (1) ローカルデバイスからの返信にデータが存在する場合（データ長は可変）
- (2) ローカルデバイスからの返信にデータが存在しない場合．

基本項目	所要時間 (μs)
送信時間 (ST)	$(B \times 0.6 + 3.25) \times K$
応答時間：返信データ有り (JT)	$(B \times 0.6 + 5.65) \times K$
応答時間：返信データ無し (JT)	$5.05 \times K$

$$1 \text{ 回のデータ通信時間 (μs)} = ST + JT + 7.4$$

例1．通信速度 = 20 Mbps，モーションデバイスのひとつのレジスタに書き込む時間

モーションデバイスのレジスタに書き込む時は書き込みコマンド（2バイト）+ レジスタデータ（4バイト）の計6バイトの送信になります．また返信データはありません．（応答フレームのみ）

$$\begin{aligned} \text{データ通信時間} &= (6 \times 0.6 + 3.25) \times 1 + 5.05 \times 1 + 7.4 \\ &= 19.3 \mu s \end{aligned}$$

例2．通信速度 = 20 Mbps，モーションデバイスのひとつのレジスタを読み出す時間

モーションデバイスのレジスタに書き込む時は読み出しコマンド（2バイト）の送信になります．また返信データは読み出しコマンド（2バイト）+ レジスタデータ（4バイト）の計6バイトになります．

$$\begin{aligned} \text{データ通信時間} &= (2 \times 0.6 + 3.25) \times 1 + (6 \times 0.6 + 5.65) \times 1 + 7.4 \\ &= 21.1 \mu s \end{aligned}$$

(3) データ通信を含んだ、トータルの通信時間

サイクリック時間にデータ通信時間を加算すれば計算できます。

例1．通信速度 = 20Mbps，ローカルデバイス32個接続してサイクリック通信間に送信2バイト，受信6バイトのデータ通信（モーションデバイスのレジスタ読み出し）を4回行った場合のサイクル時間

$$\begin{aligned}\text{トータルの通信時間} &= \text{サイクリック時間} + (\text{データ通信時間}) \times \text{データ通信回数} \\ &= (7.7 \times 1 + 7.4) \times 32 + \{(2 \times 0.6 + 3.25) \times 1 + (6 \times 0.6 + 5.65) \times 1 + 7.4\} \times 4 \\ &= 483.2 + 21.1 \times 4 \\ &= 567.6 \mu\text{s}\end{aligned}$$

注1．上記計算値は余裕時間を含んでいるため、実際の通信時間は計算値より小さくなります。

注2．通信エラーなどの発生があった場合、リトライの通信があるため上記計算値より大きな値となります。

注3．上記通信時間はBusの遅延時間、OS・アプリケーションソフトウェアの遅延時間を含みません。

2.5.4 通信ケーブル断線検出・監視機能

省配線から生じる万が一の事故（LANケーブルの断線など）が生じた場合のモーション停止機能です。

断線検出は各スレーブがサイクリック通信を監視することによって行われます。

（最後の通信から20msecの間に通信がない場合断線とみなします：通信速度20Mbps時）従って、断線箇所よりマスタから見て遠方の、即ち終端側に位置するスレーブは通信が途絶えますから、断線検出を自己検出しスレーブにつながるモーションモジュールはリセットされます。

一方、断線箇所よりマスタに近いスレーブは断線しても通信が保持される場合はリセットされないことになります。よって、マスタのプログラムはサイクリック通信エラーを監視し（4.1.2 サイクリック通信エラーフラグ（IOERR）参照）、サイクリック通信停止処置をすることによりこれらのスレーブもリセットされます。

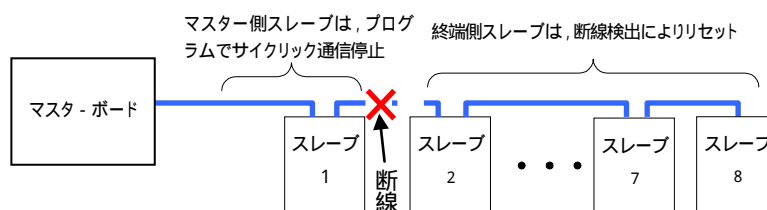


図2.5-8 断線検出

3．マスターボードハードウェア編

3．1 ブロック図

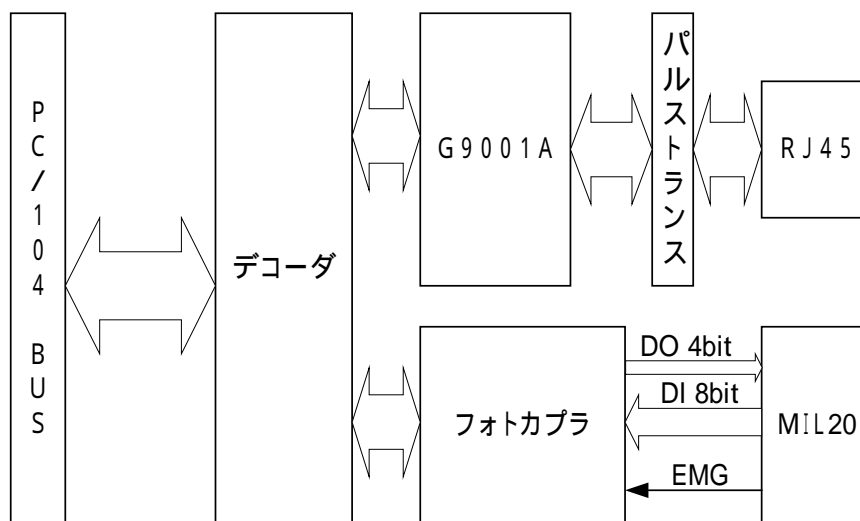


図3．1 - 1 ブロック図

3．2 ポートアドレス

HPC104-MCAT110Mで使用しているG9001Aには、512バイトモードと8バイトモードの2種類のアクセス設定があります。下記にそれぞれの特徴を記載します。

8バイトモード

特徴：512バイトモードに比べ、少ないアドレス空間のみ必要。
 （アドレス空間：オプションポートを含め、32バイト空間を使用。）
 スレーブ側にアクセスする際、入出力バッファを使用する為、512バイトモードに比べ時間がかかる。

512バイトモード

特徴：スレーブ側にアクセスするアドレスが全て存在する為、制御データのリード、ライトが短い時間で完了する。
 アドレス空間として、オプションポートを含め、1024バイト空間を必要とする。

HPC104-MCAT110Mアドレスマップです。
 8バイトモード時（バス幅16bit）

区分	アドレス	読み込み（INP）		書き込み（OUT）	
		呼称	内 容	呼称	内 容
G9001A	BAR + 0	CMSTS	センターメインステータス(15- 0)	CCMD	コマンド(15- 0)
	+ 2	CISTS	センター割込ステータス(15- 0)	-	不使用（予約）
	+ 4	CBUF	入出力バッファ IN (15- 0)	CBUF	入出力バッファ OUT (15- 0)
	+ 6	RFIFO	データ受信FIFO(15- 0)	SFIFO	データ送信FIFO(15- 0)
	+ 6				
		-	不使用（予約）	-	不使用（予約）
	+ F				
オプション ポート	+ 10	DIN	汎用入力状態（J4: ）	-	不使用（予約）
	+ 12	DOUT	汎用出力状態	DOUT	汎用出力設定（J4: ）
	+ 14	STS	G9001Aステータス	-	不使用（予約）
	+ 16	SPD	通信速度設定状態	-	不使用（予約）
	+ 18	INTEN	割込みイネーブル	INTEN	割込みイネーブル読み
	+ 1A	INTST	割込み確認	-	不使用（予約）
	+ 1C	LDSW	DIP-SW状態読み	-	不使用（予約）
	+ 1E	-	不使用（予約）	-	不使用（予約）
	+ 1F	-	不使用（予約）	-	不使用（予約）

表3．2 - 1 ボードアドレス(8バイトモード)

512バイトモード時（バス幅16bit）

区分	アドレス	読み込み（INP）		書き込み（OUT）	
		呼称	内 容	呼称	内 容
G9001A	BAR + 0	CMSTS	センターメインステータス(15- 0)	CCMD	コマンド(15- 0)
	+ 2	CISTS	センター割込ステータス(15- 0)	-	不使用（予約）
	+ 4	CBUF	入出力バッファ IN (15- 0)	CBUF	入出力バッファ OUT (15- 0)
	+ 6	RFIFO	データ受信FIFO(15- 0)	SFIFO	データ送信FIFO(15- 0)
	+ 8				
	+ 77	-	不使用（予約）	-	不使用（予約）
	+ 78	DINFO	ローカルデバイス情報(78-87)	DINFO	ローカルデバイス情報(78-87)
	+ B8	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 リセット(B8-BF)
	+ C0	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)
	+ E0	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)
	+ 100	DATA	ポートデータ IN (100-1FF)	DATA	ポートデータ OUT (100-1FF)
オプション ポート	+ 200	DIN	汎用入力状態（J4: ）	-	不使用（予約）
	+ 202	DOUT	汎用出力状態	DOUT	汎用出力設定（J4: ）
	+ 204	STS	G9001Aステータス	-	不使用（予約）
	+ 206	SPD	通信速度設定状態	-	不使用（予約）
	+ 208	INTEN	割込みイネーブル	INTEN	割込みイネーブル読み込み
	+ 20A	INTST	割込み確認	-	不使用（予約）
	+ 20C	LDSW	DIP-SW状態読み込み	-	不使用（予約）
	+ 20D				
	+ 3FF	-	不使用（予約）	-	不使用（予約）

表3.2-2 ボードアドレス(512バイトモード)

4. センターデバイス (G9001A) ・オプションポート

マスターボードにはセンターデバイス (G9001A) が搭載され、付加機能としてオプションポートが1組あります。

4.1 センターデバイス (G9001A)

センターデバイス (G9001A) はモーションデバイスやDIOデバイスなどのローカルデバイスと通信します。この節ではセンターデバイス (G9001A) のステータス、コマンド、レジスタ等の解説をします。

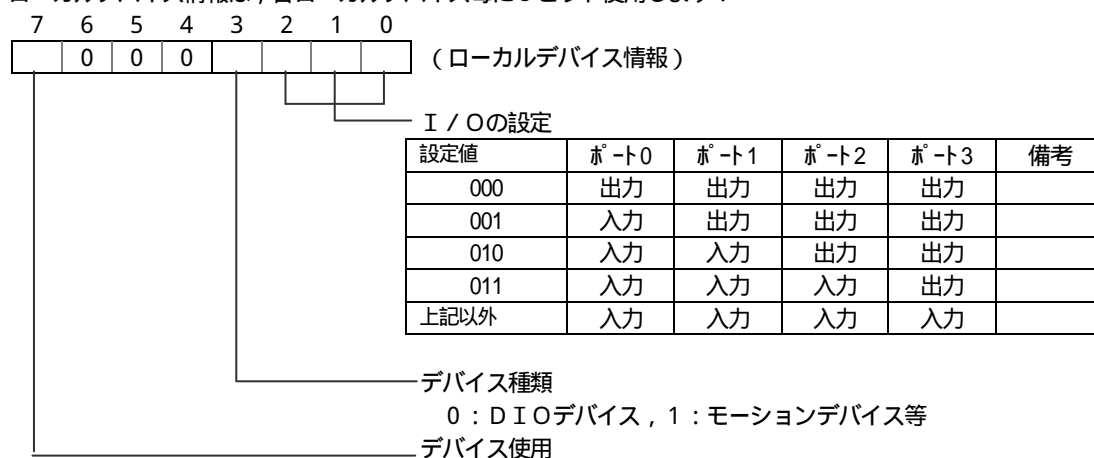
運用についてはソフトウェア編を合わせて参照ください。また、この節のレジスタのビット説明で、"0" は書き込み時 "0" 以外禁止、及び読み出し時 "0" 固定を表します。

4.1.1 ローカルデバイス情報 (DINFO)

システム通信により、全ローカルデバイス (デバイス番号0～63) を順次ポーリングし、ローカルデバイスからの応答により、各ローカルデバイスの接続状態、デバイスの種類、および入出力ポートの設定等の確認を行い、「ローカルデバイス情報」を更新します。

「ローカルデバイス情報」が既に判っている場合には、アプリケーションから書込む事もできます。

ローカルデバイス情報は、各ローカルデバイス毎に8ビット使用します。



<ローカルデバイス情報例>

HM - DIO32Cの時は82h

HM - P100Cの時は 8Bh

アドレス	読み込み (INP)	書き込み (OUT)
BAR + 78	ライン1モジュールID=0 の-カルデバイス情報	ライン1モジュールID=0 の-カルデバイス情報設定
+ 79	ライン1モジュールID=1 の-カルデバイス情報	ライン1モジュールID=1 の-カルデバイス情報設定
+ 80	ライン1モジュールID=2 の-カルデバイス情報	ライン1モジュールID=2 の-カルデバイス情報設定
~	~	~
+ B5	ライン1モジュールID=61 の-カルデバイス情報	ライン1モジュールID=61 の-カルデバイス情報設定
+ B6	ライン1モジュールID=62 の-カルデバイス情報	ライン1モジュールID=62 の-カルデバイス情報設定
+ B7	ライン1モジュールID=63 の-カルデバイス情報	ライン1モジュールID=63 の-カルデバイス情報設定

表4.1-1 ローカルデバイス情報アドレス

4.1.2 サイクリック通信エラーフラグ (IOERR)

ローカルデバイスの入出力ポートへの通信は全てサイクリック通信で行われます。

このサイクリック通信で、同じモジュールIDのローカルデバイスに対して、3回連続して通信エラーが発生した場合にサイクリック通信エラーとして扱います。(断線時もサイクリック通信エラーとなります)

エラー時には以下のような規則でサイクリック通信エラーフラグの特定のビットが“1”になります。

これを確認することでエラー対象のローカルデバイスを特定できます。

エラーによって“1”になったビットを“0”に戻すには、そのビットに対して“1”を書き込むことで行えます。

フラグをリセットするひとつの方法としてリードした“サイクリック通信エラーフラグ”のデータを、そのままサイクリック通信エラーフラグリセットへ書き込めばフラグをリセットできます。

< ビット構成 >

読出し . . . 0 : エラーなし, 1 : エラー有り

書込み . . . 0 : 無視, 1 : フラグリセット

“BAR + 0B8h”を読出すとライン1のモジュールID 15 ~ 0を見ることができます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

“BAR + 2BAh”を読出すとライン2のモジュールID 31 ~ 16を見ることができます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

上記のように最下位ビットが一番若いモジュールIDを持つローカルデバイスのエラー状態になります。

サイクリック通信エラーフラグのアドレスの決定は以下に行います(小数点以下切り捨て)

$$\text{アドレス} = \text{BAR} + 0\text{B}8\text{h} + (\text{モジュールID} / 8)$$

アドレス(HEX)	読込み (INP)	書込み (OUT)
BAR + B8	ライン1モジュールID 15- 0 サイクリック通信エラーフラグ	ライン1モジュールID 15- 0 サイクリック通信エラーフラグ リセット
+ BA	ライン1モジュールID 31-16 サイクリック通信エラーフラグ	ライン1モジュールID 31-16 サイクリック通信エラーフラグ リセット
+ BC	ライン1モジュールID 47-32 サイクリック通信エラーフラグ	ライン1モジュールID 47-32 サイクリック通信エラーフラグ リセット
+ BE	ライン1モジュールID 63-48 サイクリック通信エラーフラグ	ライン1モジュールID 63-48 サイクリック通信エラーフラグ リセット

表4.1-2 サイクリック通信エラーフラグアドレス

4.1.3 入力ポート変化フラグ設定 (I I N T)

サイクリック通信により、接続されているD I Oデバイスなどのポート情報は自動的に取得できます。
またモーションデバイスなどでも、デバイスのステータス情報をサイクリック通信により、センタデバイスは定期的に取得しています。
このような入力ポートの変化や、モーションデバイスのステータス変化を検出する時に入力ポート変化フラグを使用します。
入力ポート変化フラグ設定は、状態変化を監視したいローカルデバイスの番号に対応するビットを“ 1 ”にすることで指定します。

< ビット構成 >

読出し . . . 0 : 入力ポート変化フラグ設定なし, 1 : 入力ポート変化フラグ設定状態
書込み . . . 0 : 入力ポート変化を監視しない, 1 : 入力ポート変化を監視する

“ B A R + 0 C 0 h ” を読出すとライン 1 のモジュール I D 3 ~ 0 の入力ポート変化フラグ設定状態を確認できます。

“ B A R + 0 C 0 h ” へ書込むとライン 1 のモジュール I D 3 ~ 0 の入力ポート変化フラグを設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
モジュール ID=3				モジュール ID=2				モジュール ID=1				モジュール ID=0			
ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0	ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0	ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0	ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0

“ B A R + 2 C 2 h ” を読出すとライン 2 のモジュール I D 7 ~ 4 の入力ポート変化フラグ設定状態を確認できます。

“ B A R + 2 C 2 h ” へ書込むとライン 2 のモジュール I D 7 ~ 4 の入力ポート変化フラグを設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
モジュール ID=7				モジュール ID=6				モジュール ID=5				モジュール ID=4			
ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0	ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0	ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0	ポ-ト 3	ポ-ト 2	ポ-ト 1	ポ-ト 0

上記のように4ビット単位ごとに1つのローカルデバイスが対応します。最下位4ビットが、一番若いモジュール I D を持つローカルデバイスの入力ポート変化フラグ設定エリアになります。

この4ビット単位の下位ビットがポート 0 に対応し、順にポート 3 までそれぞれ対応してゆきます。

別のローカルデバイスに対して設定したい場合、以下の規則によりアドレスを決定してください。

(小数点以下切り捨て)

$$\text{アドレス} = 0 C 0 h + (\text{モジュール I D} / 2)$$

アドレス	読込み (I N P)	書込み (O U T)
B A R + C 0	ライン1モジュールID 3- 0 入力ポート変化フラグ 設定状態	ライン1モジュールID 3- 0 入力ポート変化フラグ 設定
+ C 2	ライン1モジュールID 7- 4 入力ポート変化フラグ 設定状態	ライン1モジュールID 7- 4 入力ポート変化フラグ 設定
+ C 4	ライン1モジュールID 11- 8 入力ポート変化フラグ 設定状態	ライン1モジュールID 11- 8 入力ポート変化フラグ 設定
+ C 6	ライン1モジュールID 15-12 入力ポート変化フラグ 設定状態	ライン1モジュールID 15-12 入力ポート変化フラグ 設定
+ C 8	ライン1モジュールID 19-16 入力ポート変化フラグ 設定状態	ライン1モジュールID 19-16 入力ポート変化フラグ 設定
+ C A	ライン1モジュールID 23-20 入力ポート変化フラグ 設定状態	ライン1モジュールID 23-20 入力ポート変化フラグ 設定
+ C C	ライン1モジュールID 27-24 入力ポート変化フラグ 設定状態	ライン1モジュールID 27-24 入力ポート変化フラグ 設定
+ C E	ライン1モジュールID 31-24 入力ポート変化フラグ 設定状態	ライン1モジュールID 31-24 入力ポート変化フラグ 設定
+ D 0	ライン1モジュールID 35-32 入力ポート変化フラグ 設定状態	ライン1モジュールID 35-32 入力ポート変化フラグ 設定
+ D 2	ライン1モジュールID 39-36 入力ポート変化フラグ 設定状態	ライン1モジュールID 39-36 入力ポート変化フラグ 設定
+ D 4	ライン1モジュールID 43-40 入力ポート変化フラグ 設定状態	ライン1モジュールID 43-40 入力ポート変化フラグ 設定
+ D 6	ライン1モジュールID 47-44 入力ポート変化フラグ 設定状態	ライン1モジュールID 47-44 入力ポート変化フラグ 設定
+ D 8	ライン1モジュールID 51-48 入力ポート変化フラグ 設定状態	ライン1モジュールID 51-48 入力ポート変化フラグ 設定
+ D A	ライン1モジュールID 55-52 入力ポート変化フラグ 設定状態	ライン1モジュールID 55-52 入力ポート変化フラグ 設定
+ D C	ライン1モジュールID 59-56 入力ポート変化フラグ 設定状態	ライン1モジュールID 59-56 入力ポート変化フラグ 設定
+ D E	ライン1モジュールID 63-60 入力ポート変化フラグ 設定状態	ライン1モジュールID 63-60 入力ポート変化フラグ 設定

表 4. 1 - 3 入力ポート変化フラグ設定アドレス

4.1.4 入力ポート変化フラグ (I R E S)

入力ポート変化フラグ設定を行ったポートの状態に変化があると、そのポートに対応した入力ポート変化フラグのビットが“ 1 ”になります。

入力ポート変化フラグを監視することで状態変化があったローカルデバイス、ポート番号 (ステータス) を知ることができます。

< ビット構成 >

読出し . . . 0 : 入力ポート変化なし, 1 : 入力ポート変化有り

書込み . . . 0 : 無視, 1 : 入力ポート変化フラグリセット

“ B A R + 0 E 0 h ” を読出すとライン 1 のモジュール I D 3 ~ 0 の入力ポート変化フラグが読出されます。

“ B A R + 0 E 0 h ” へ書込むとライン 1 のモジュール I D 3 ~ 0 の入力ポート変化フラグをリセットします。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
モジュール ID=3				モジュール ID=2				モジュール ID=1				モジュール ID=0			
ポート 3	ポート 2	ポート 1	ポート 0	ポート 3	ポート 2	ポート 1	ポート 0	ポート 3	ポート 2	ポート 1	ポート 0	ポート 3	ポート 2	ポート 1	ポート 0

“ B A R + 0 E 2 h ” を読出すとライン 2 のモジュール I D 7 ~ 4 の入力ポート変化フラグが読み出されます。

“ B A R + 0 E 2 h ” へ書込むとライン 2 のモジュール I D 7 ~ 4 の入力ポート変化フラグをリセットします。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
モジュール ID=7				モジュール ID=6				モジュール ID=5				モジュール ID=4			
ポート 3	ポート 2	ポート 1	ポート 0	ポート 3	ポート 2	ポート 1	ポート 0	ポート 3	ポート 2	ポート 1	ポート 0	ポート 3	ポート 2	ポート 1	ポート 0

上記のように 4 ビット単位ごとに 1 つのローカルデバイスが対応します。最下位 4 ビットが、一番若いモジュールデバイス I D を持つローカルデバイスの入力ポート変化フラグデータエリアになります。

この 4 ビット単位の下位ビットがポート 0 に対応し、順にポート 3 までそれぞれ対応してゆきます。

別のローカルデバイスに対して確認したい場合、以下の規則によりアドレスを決定してください。

(小数点以下切り捨て)

アドレス = 0 E 0 h + (モジュール I D / 2)

フラグをリセットするひとつの方法としてリードした “ 入力ポート変化フラグ ” のデータを、そのまま入力ポート変化フラグリセットへ書き込めばフラグをリセットできます。

アドレス	読込み (I N P)	書込み (O U T)
B A R + E 0	ライン1モジュールID 3- 0 入力ポート変化フラグ	ライン1モジュールID 3- 0 入力ポート変化フラグ リセット
+ E 2	ライン1モジュールID 7- 4 入力ポート変化フラグ	ライン1モジュールID 7- 4 入力ポート変化フラグ リセット
+ E 4	ライン1モジュールID 11- 8 入力ポート変化フラグ	ライン1モジュールID 11- 8 入力ポート変化フラグ リセット
+ E 6	ライン1モジュールID 15-12 入力ポート変化フラグ	ライン1モジュールID 15-12 入力ポート変化フラグ リセット
+ E 8	ライン1モジュールID 19-16 入力ポート変化フラグ	ライン1モジュールID 19-16 入力ポート変化フラグ リセット
+ E A	ライン1モジュールID 23-20 入力ポート変化フラグ	ライン1モジュールID 23-20 入力ポート変化フラグ リセット
+ E C	ライン1モジュールID 27-24 入力ポート変化フラグ	ライン1モジュールID 27-24 入力ポート変化フラグ リセット
+ E E	ライン1モジュールID 31-24 入力ポート変化フラグ	ライン1モジュールID 31-24 入力ポート変化フラグ リセット
+ F 0	ライン1モジュールID 35-32 入力ポート変化フラグ	ライン1モジュールID 35-32 入力ポート変化フラグ リセット
+ F 2	ライン1モジュールID 39-36 入力ポート変化フラグ	ライン1モジュールID 39-36 入力ポート変化フラグ リセット
+ F 4	ライン1モジュールID 43-40 入力ポート変化フラグ	ライン1モジュールID 43-40 入力ポート変化フラグ リセット
+ F 6	ライン1モジュールID 47-44 入力ポート変化フラグ	ライン1モジュールID 47-44 入力ポート変化フラグ リセット
+ F 8	ライン1モジュールID 51-48 入力ポート変化フラグ	ライン1モジュールID 51-48 入力ポート変化フラグ リセット
+ F A	ライン1モジュールID 55-52 入力ポート変化フラグ	ライン1モジュールID 55-52 入力ポート変化フラグ リセット
+ F C	ライン1モジュールID 59-56 入力ポート変化フラグ	ライン1モジュールID 59-56 入力ポート変化フラグ リセット
+ F E	ライン1モジュールID 63-60 入力ポート変化フラグ	ライン1モジュールID 63-60 入力ポート変化フラグ リセット

表 4. 1 - 4 入力ポート変化フラグアドレス

4.1.5 ポートデータ (DATA)

ポートデータからの読み出しはD I Oデバイスの入力ポートからのデータの確認，またはモーションデバイスのモーションメインステータスの確認，汎用出力ポート状態の確認に使用します．

ポートデータへの書込みはD I Oデバイスの出力ポートへのデータ設定，またはモーションデバイスの場合は汎用出力ポートへのデータ設定に使用されます．

例えば，モジュールID = 0 のデバイスがD I Oデバイスならば，“BAR + 1 0 0 h” の読み出しで，汎用入力ポートのデータが確認できます．

同じく“BAR + 1 0 2 h” の読み出しで汎用出力状態の確認，書込みで汎用出力設定となります．

アクセスは，下記アドレスマップに記載されているモジュールIDと，ポート番号のとおりです．

アドレス	読み込み (I N P) ポートデータ (DATA)	書込み (O U T) ポートデータ (DATA)
BAR + 1 0 0	ライノモジュールID 0 ポート1,0データ	ライノモジュールID 0 ポート1,0設定
+ 1 0 2	ライノモジュールID 0 ポート3,2データ	ライノモジュールID 0 ポート3,2設定
+ 1 0 4	ライノモジュールID 1 ポート1,0データ	ライノモジュールID 1 ポート1,0設定
+ 1 0 6	ライノモジュールID 1 ポート3,2データ	ライノモジュールID 1 ポート3,2設定
+ 1 0 8	ライノモジュールID 2 ポート1,0データ	ライノモジュールID 2 ポート1,0設定
+ 1 0 A	ライノモジュールID 2 ポート3,2データ	ライノモジュールID 3 ポート3,2設定
5		
+ 1 F C	ライノモジュールID 63 ポート1,0データ	ライノモジュールID 63 ポート1,0設定
+ 1 F E	ライノモジュールID 63 ポート3,2データ	ライノモジュールID 63 ポート3,2設定

表 4.1-5 ポートデータアドレス

4.1.6 センターメインステータス (CMSTS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BBSY	DBSY	RBSY	SBSY	0	RDBB	TDBB	REF	0	CAER	ERAE	EDTE	EIOE	IOPC	BRKF	CEND

図4.1-1 センターメインステータス (CMSTS) のビット構成

bit	名 称	説 明	備 考
0	CEND	データ送信FIFO書き込み可能時に1になります。システム通信、またはデータ通信が完了して、データ送信FIFOに次データの書き込みが可能になった時に1なり、割り込み信号(#INT)を出力します。本ビットのクリア方法は、RENV0レジスタのb9の状態によります。	
1	BRKF	ブレークフレーム受信時に1になり、割り込み信号を出力します。本ビットのクリア方法は、RENV0レジスタのb9の状態によります。	
2	IOPC	「入力変化割り込み検定」を1にセットした入力ポートの状態が変化した場合に1になり、割り込み信号を出力します。「入力ポート変化フラグ」の全256ビットのOR信号です。全ビットが0になると、このビットは0に戻ります。	
3	EIOE	サイクリック通信エラー発生時に1になり、割り込み信号を出力します。「サイクリック通信エラーフラグ」の全64ビットのOR信号です。全ビットが0になると、このビットは0に戻ります。	
4	EDTE	データ通信エラー発生時に1になり、割り込み信号を出力します。本ビットのクリア方法は、RENV0レジスタのb9の状態によります。	
5	ERAE 注1	ローカルデバイス側受信処理エラー発生時に1になり、割り込み信号を出力します。エラーの発生条件は、「5.1.2 センター割込ステータス」の「ERA」を参照ください。このセンター割込ステータスで、エラーが発生したローカルデバイスとエラーの内容の確認ができます。本ビットのクリア方法は、RENV0レジスタのb9の状態によります。	
6	CAER	アプリケーションのアクセスエラーです。送信データが空のままデータ送信コマンドを書き込む等、アプリケーションから不適切なアクセスがあると1になり、割り込み信号を出力します。エラーの発生条件は、「5.1.2 センター割込ステータス」の「CAE」を参照ください。ここでエラーの内容の確認ができます。本ビットのクリア方法は、RENV0レジスタのビット9の状態によります。	
7	(未定義)	常に0になります。	
8	REF	未送信の出力ポートデータがある時1になります。出力ポートエリアにデータを書き込むと1になり、全ポートへのサイクリック通信を2回以上エラー無しで行った後に0に戻ります。	
9	TDBB	データ送信FIFOに送信データがある時に1になります。データ送信FIFOに書き込むと1になり、データ送信コマンド、または送信FIFOリセットコマンドを書き込んだ時に0に戻ります。	
10	RDBB	データ受信FIFOに受信データがある時に1になります。データデバイスからデータを受信すると1になり、アプリケーションが受信データを全て読み出すと0に戻ります。	
11	(未定義)	常に0になります。	
12	SBSY	サイクリック通信スタート中に1になります。	
13	RBSY	リセット処理中に1になります。	
14	DBSY	システム通信中、またはデータ通信中に1になります。	
15	BBSY	RENV0(8) = "1" でブレーク通信コマンド(0610h)を発行したとき、ブレーク通信が完了するまで"1"となります。それ以外は"0"です。	

注1. 複数ローカルデバイスにエラーが発生した場合は、ERAの内容は最後にエラーになったローカルデバイスの情報になります。

表4.1-6 センターメインステータス (MSTS) の内容

4.1.7 センター割込ステータス (CISTS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAE3	CAE2	CAE1	CAE0	ERA3	ERA2	ERA1	ERA0	LNRV	0	EDN5	EDN4	EDN3	EDN2	EDN1	EDN0

図 4.1-2 センター割込ステータス (CISTS) のビット構成

bit	名 称	説 明	備 考										
5 ~ 0	EDN 5 ~ 0	MSTSの, EDTE = 1またはERA E = 1のエラー発生時のモジュールIDで, 次のエラー発生まで記憶されます.											
6	(未定義)	常に0になります.											
7	LNRV	ローカルデバイス側データ未受信時に1になります. データ通信がエラーで終了した (EDTE = 1) 場合, ローカルデバイス側がセンターデ バイスからのデータを受信出来なかった時 (ローカルデバイス側は無応答) は1になり, 受 信出来た時 (ローカルデバイス側からの通信が, 通信障害などで破壊され, センタデバイス 側が正常に受信できなかった場合. この場合, ローカルデバイス側が正常にデータを受け取 れたかどうかを確認する作業が必要となります) は0になります. 次のエラー発生まで記憶されます.											
11 ~ 8	ERA 3 ~ 0	ローカルデバイス側が正常受信したにもかかわらず, そのバケットの内容がローカルデバイスの種 類にマッチしていない場合に発生します. このとき以下のようなコードを本ビット部分に記憶します. このコードは次のエラー発生まで記 憶されます. <table><tr><th>コード</th><th>発 生 条 件</th></tr><tr><td>0001</td><td>センターデバイス内の “ デバイス情報エリア ” の I / O の設定情報と, ローカルデバイ ス側の I / O ポートの組合せが異なっている場合</td></tr><tr><td>0010</td><td>D I O デバイスがデータ通信を受信した時</td></tr><tr><td>0011</td><td>モーションデバイスが自己の持っている受信バッファ容量 (6 バイト) 以上のデータ 通信を受信した時 その他のデバイスが自己の持っている受信バッファ容量以上のデータ通信を受信した時</td></tr></table>	コード	発 生 条 件	0001	センターデバイス内の “ デバイス情報エリア ” の I / O の設定情報と, ローカルデバイ ス側の I / O ポートの組合せが異なっている場合	0010	D I O デバイスがデータ通信を受信した時	0011	モーションデバイスが自己の持っている受信バッファ容量 (6 バイト) 以上のデータ 通信を受信した時 その他のデバイスが自己の持っている受信バッファ容量以上のデータ通信を受信した時			
コード	発 生 条 件												
0001	センターデバイス内の “ デバイス情報エリア ” の I / O の設定情報と, ローカルデバイ ス側の I / O ポートの組合せが異なっている場合												
0010	D I O デバイスがデータ通信を受信した時												
0011	モーションデバイスが自己の持っている受信バッファ容量 (6 バイト) 以上のデータ 通信を受信した時 その他のデバイスが自己の持っている受信バッファ容量以上のデータ通信を受信した時												
15 ~ 12	CAE 3 ~ 0	アプリケーションが G 9 0 0 1 A に対して不正なアクセスを行った場合, 以下のようなコードを本 ビット部分に記憶します. このコードは次のエラー発生まで記憶されます. <table><tr><th>コード</th><th>発 生 条 件</th></tr><tr><td>0001</td><td>使用ローカルデバイス数がゼロでサイクリック通信スタートコマンドを書き込んだ時</td></tr><tr><td>0010</td><td>送信データを送信用 F I F O に設定しないでデータ送信スタートコマンド書き込んだ時</td></tr><tr><td>0011</td><td>D B S Y = 1 の時に, もしくは のアクセスを行った時 受信用 F I F O からの読み出し, または送信用 F I F O への書き込みを行った時 システム通信, またはデータ通信スタートコマンドを書き込んだ時</td></tr><tr><td>0100</td><td>デバイス情報エリアのビット7が “ 0 ” となっている (未使用デバイス扱い となってい る) ローカルデバイスに対して, データ通信を行った時</td></tr></table>	コード	発 生 条 件	0001	使用ローカルデバイス数がゼロでサイクリック通信スタートコマンドを書き込んだ時	0010	送信データを送信用 F I F O に設定しないでデータ送信スタートコマンド書き込んだ時	0011	D B S Y = 1 の時に, もしくは のアクセスを行った時 受信用 F I F O からの読み出し, または送信用 F I F O への書き込みを行った時 システム通信, またはデータ通信スタートコマンドを書き込んだ時	0100	デバイス情報エリアのビット7が “ 0 ” となっている (未使用デバイス扱い となってい る) ローカルデバイスに対して, データ通信を行った時	
コード	発 生 条 件												
0001	使用ローカルデバイス数がゼロでサイクリック通信スタートコマンドを書き込んだ時												
0010	送信データを送信用 F I F O に設定しないでデータ送信スタートコマンド書き込んだ時												
0011	D B S Y = 1 の時に, もしくは のアクセスを行った時 受信用 F I F O からの読み出し, または送信用 F I F O への書き込みを行った時 システム通信, またはデータ通信スタートコマンドを書き込んだ時												
0100	デバイス情報エリアのビット7が “ 0 ” となっている (未使用デバイス扱い となってい る) ローカルデバイスに対して, データ通信を行った時												

表 4.1-7 センター割込ステータス (CISTS) の内容

4.1.8 センターデバイスコマンド (CCMD)

(1) センター動作 コマンド

コマンド種類	コマンド	説 明	備 考																
無効コマンド	0000 0000 0000 0000 (0000h)	無効コマンド																	
ソフトウェア リセットコマンド	0000 0001 0000 0000 (0100h)	センターデバイス (G 9 0 0 1 A) をリセットします。 このコマンド発行後は1 0 0 μ s e c 以上ウェイトして下さい。 リセット後の状態 「コマンド」…………… 0 0 0 0 h 「センターメインステータス」………… 0 0 0 0 h 「センター割込ステータス」………… 0 0 0 0 h 「入出力バッファ」………… 0 0 0 0 h 「データ送信用 F I F O」………… 不定 「データ受信用 F I F O」………… 不定 「デバイス情報」…………… 全て 0 0 h 「サイクリック通信エラーフラグ」… 全て 0 0 h 「入力ポート変化フラグ設定」………… 全て 0 0 h 「入力ポート変化フラグ」…………… 全て 0 0 h 「ポートデータ」…………… 全て 0 0 h 「レジスタ類」…………… R V E R = 0 0 0 1 h , それ以外全て 0 0 0 0 h																	
送信用 F I F O リセットコマンド	0000 0010 0000 0000 (0200h)	データ送信用の F I F O だけをリセットします。																	
受信用 F I F O リセットコマンド	0000 0011 0000 0000 (0300h)	データ受信用の F I F O だけをリセットします。																	
センター割込 ステータスの クリアコマンド	0000 0100 0\$\$\$ 00\$\$ (04xxh)	コマンドのbit 0 , 1 , 4 , 5 , 6 は以下のように対応します。 <table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>CAER</td><td>ERAE</td><td>EDTE</td><td>0</td><td>0</td><td>BRKF</td><td>CEND</td></tr></table> 各ビットを“ 1 ”とすることで、対応するステータスをクリアします。 ただし、R E N V 0 (9) = “ 0 ” の時は無効です。	7	6	5	4	3	2	1	0	0	CAER	ERAE	EDTE	0	0	BRKF	CEND	
7	6	5	4	3	2	1	0												
0	CAER	ERAE	EDTE	0	0	BRKF	CEND												
エラーカウンタ クリアコマンド	0000 0110 0000 0000 (0600h)	エラーカウンタレジスタをゼロクリアします。																	
ブレーク通信 コマンド	0000 0110 0001 0000 (0610h)	R E N V 0 (8) = “ 1 ” と設定して自動ブレーク機能を無効としたとき、 このコマンドで任意のタイミングでブレーク通信を発行できます。 R E N V 0 (8) = “ 0 ” の時は無効です。																	
全デバイスへの システム通信 コマンド	0001 0000 0000 0000 (1000h)	全デバイス(モジュール I D 0 ~ 6 3) を順次ポーリングし、モジュール I D に対応した「ローカルデバイス情報」のエリアを更新します。 「ローカルデバイス情報」には、以下の内容が含まれます。 ローカルデバイス使用：応答無しの時に 0 , 有りの時に 1 に更新 ローカルデバイス種類：モーションデバイスの時に 1 に更新。 ローカルデバイスの I / O ポートの設定情報 システム通信に対する無応答はエラーではありません。 ただし C R C ミスマッチによるエラーはあり得ます。																	
サイクリック通信 除外中の全ローカル デバイスへの システム通信	0001 0001 0000 0000 (1100h)	「ローカルデバイス情報」の内、“ローカルデバイス使用”のビットが 0 になっている全ローカルデバイスを順次ポーリングし、ローカルデバイス 番号に対応した「ローカルデバイス情報」のエリアを更新します。更新内 容は、コマンド 1 0 0 0 h と同じです。システム通信に対する無応答は エラーではありません。ただし C R C ミスマッチによるエラーはあり得ま す。																	
指定したローカル デバイスへの システム通信	0001 0010 00## ##### (1200h ~ 123Fh)	指定したローカルデバイスだけをポーリングし、モジュール I D に対応し た「ローカルデバイス情報」のエリアを更新します。更新内容は、コマン ド 1 0 0 0 h と同じです。システム通信に対する無応答はエラーではあり ません。ただし C R C ミスマッチによるエラーはあり得ます。																	

(次ページに続く)

(前ページからの続き)

コマンド種類	コマンド	説 明	備 考
指定したローカルデバイスの属性情報の取得	0001 0011 00## ##### (1300h~133Fh)	ポーリングの応答フレームは、ローカルデバイス属性情報になっています。このコマンドにより、指定したローカルデバイスのポーリングが行われ、属性情報がデータ受信用FIFOにコピーされます。尚、「ローカルデバイス情報」のエリアは変化しません。 データ受信用FIFOの内容は下記の通りです。 bit 4~ 0 : (データ最長バイト数) / 8 - 1 bit 7~ 5 : 未使用 (不定) bit15~ 8 : 機種コード (DIOデバイス: 01h, モーションデバイス: 81h) bit18~16 : 入出力ポートの設定 bit19 : 0 固定 bit31~20 : モーションデバイス: 000h, その他のローカルデバイス: 001h	
指定したグループへのコマンド送信	0010 0ggg cccc cccc (2*01h) (2*02h) (2*04h)	指定したグループのデータデバイスに対しコマンドを送信します。 ggg = 000の時は全グループ cccc cccc 0000 0001(01h):スタート 0000 0010(02h):ストップ 0000 0100(04h):ソフトウェアリセット	
サイクリック通信の開始	0011 0000 0000 0000 (3000h)	「ローカルデバイス情報」の「ローカルデバイス使用」のビットが1のデバイスを対象に、サイクリック通信を開始します。	
サイクリック通信の停止	0011 0001 0000 0000 (3100h)	サイクリック通信を停止します。	
データ通信	0100 0000 00## ##### (4000h~403Fh)	指定したデバイスへ送信用FIFO内のデータを送信します。 応答データは受信用FIFOに格納されます。	
データ通信キャンセル	0100 0001 0000 0000 (4100h)	データ通信を中断させ、送信用FIFOをリセットします。 本コマンドは、送信終了後は無効です。	

注1. #記号のビット: #ビットの上位から順に、モジュールIDの上位ビットを設定します。

&記号のビット: ポート0, 1の時は0を設定し, ポート2, 3の時は1を設定します。

\$記号のビット: 設定はどちらの値でも大丈夫です。

*記号のビット: 設定はどちらの値でも大丈夫です。

表4. 1 - 8 センター動作コマンドの内容

(2) センターレジスタ制御コマンド

センターデバイス (G9001A) の持つレジスタは、アドレスマップ上に割り付けられていません。

よって、レジスタへアクセスするためには、センターレジスタ制御コマンドを使用しなければなりません。

コマンド種類	コマンド	説 明	備 考
RENV0 ライトコマンド	0101 0101 0000 0000 (5500h)	入出力バッファヘデータをセットし本コマンドを発行すると入出力バッファの値がRENV0レジスタへコピーされます。	
RENV0 リードコマンド	0110 0101 0000 0000 (6500h)	このコマンドを発行すると、RENV0レジスタの値が入出力バッファへコピーされます。入出力バッファをリードすることで、RENV0の値を得ることができます。	
エラーカウンタ リードコマンド	0110 0101 0000 0001 (6501h)	このコマンドを発行すると、エラーカウンタレジスタの値が入出力バッファへコピーされます。入出力バッファをリードすることで、エラーカウンタの値を得ることができます。	
サイクリック周期 レジスタ リードコマンド	0110 0101 0000 0010 (6502h)	このコマンドを発行すると、サイクリック周期レジスタの値が入出力バッファへコピーされます。入出力バッファをリードすることで、サイクリック周期レジスタの値を得ることができます。	
受信アドレス レジスタ リードコマンド	0110 0101 0000 0011 (6503h)	このコマンドを発行すると、受信アドレスレジスタの値が入出力バッファへコピーされます。入出力バッファをリードすることで、受信アドレスレジスタの値を得ることができます。	
バージョン情報 レジスタ リードコマンド	0110 0101 0000 0100 (6504h)	このコマンドを発行すると、バージョン情報レジスタの値が入出力バッファへコピーされます。入出力バッファをリードすることで、バージョン情報レジスタの値を得ることができます。	

表4. 1 - 9 センターレジスタ制御コマンドの内容

(3) ローカルデバイス制御コマンド

8バイトモードに設定した場合、下表のコマンドを書き込む事で、システム通信により、全ローカルデバイス（デバイス番号0～63）を順次ポーリングし、ローカルデバイスからの応答により、各ローカルデバイスの接続状態、デバイスの種類、および入出力ポートの設定等の確認を行い、「ローカルデバイス情報」を更新します。

512バイトモードに設定した場合は、本コマンドは必要ありません。

コマンド種類	コマンド	説明	備考
ローカルデバイス情報 ライトコマンド	0101 0000 0xxx xxxx (5000h～507Fh)	ローカルデバイス情報エリアへの書き込み 入出力バッファの内容がデバイス情報エリア内の1ワードに書込まれます	
サイクリック通信エラーフラグ ライトコマンド	0101 0001 0xxx xxxx (5100h～517Fh)	サイクリック通信エラーフラグへの書き込み 入出力バッファの内容が上記エリアの1ワードに書込まれます	
入力ポート変化フラグ設定状態 ライトコマンド	0101 0010 0xxx xxxx (5200h～527Fh)	入力ポート変化フラグ設定状態への書き込み 入出力バッファの内容が上記エリアの1ワードに書込まれます	
入力ポート変化フラグ ライトコマンド	0101 0011 0xxx xxxx (5300h～537Fh)	入力ポート変化フラグへの書き込み 入出力バッファの内容が上記エリアの1ワードに書込まれます	
ポートデータ ライトコマンド	0101 0100 0xxx xxxx (5400h～547Fh)	ポートデータへの書き込み 入出力バッファの内容が上記エリアの1ワードに書込まれます	
ローカルデバイス情報 リードコマンド	0110 0000 0xxx xxxx (6000h～607Fh)	ローカルデバイス情報エリアへの読出し 入出力バッファの内容がデバイス情報エリア内の1ワードにコピーされます	
サイクリック通信エラーフラグ リードコマンド	0110 0001 0xxx xxxx (6100h～617Fh)	サイクリック通信エラーフラグへの読出し 入出力バッファの内容が上記エリアの1ワードにコピーされます	
入力ポート変化フラグ設定状態 リードコマンド	0110 0010 0xxx xxxx (6200h～627Fh)	入力ポート変化フラグ設定状態への読出し 入出力バッファの内容が上記エリアの1ワードにコピーされます	
入力ポート変化フラグ リードコマンド	0110 0011 0xxx xxxx (6300h～637Fh)	入力ポート変化フラグへの読出し 入出力バッファの内容が上記エリアの1ワードにコピーされます	
ポートデータ リードコマンド	0110 0100 0xxx xxxx (6400h～647Fh)	ポートデータへの読出し 入出力バッファの内容が上記エリアの1ワードにコピーされます	

表4.1-10 ローカルデバイス制御コマンドの内容

(4) センターレジスタへのデータ読み出しと書き込み

読み出し方法

読み出す対象のセンターデバイス（ライン番号）へ対象レジスタリードコマンドを書込みます。
読み出す対象のセンターデバイス（ライン番号）の入出力バッファを読み出します。

書き込み方法

書き込む対象のセンターデバイス（ライン番号）の入出力バッファへデータを書込みます。
書き込む対象のセンターデバイス（ライン番号）へ対象レジスタライトコマンドを書込みます。

4.1.9 センターデバイスレジスタ

(1) RENV0レジスタ (Read/Write)

センターデバイス環境設定用の16ビットのレジスタです。リセット直後の状態は、"0000h"です。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	MCLR	BKOF	0	MCSE	MERE	MEDE	MEIE	MIOP	MBRK	MCED

図4.1-3 RENV0レジスタのビット構成

bit	名 称	説 明	備 考
0	MCED	“1”でCEND割込出力をマスク。センター割込ステータスは変化します。	無用な割込を無視
1	MBRK	“1”でBRKF割込出力をマスク。センター割込ステータスは変化します。	
2	MIOP	“1”でIOPC割込出力をマスク。センター割込ステータスは変化します。	
3	MEIE	“1”でEIOE割込出力をマスク。センター割込ステータスは変化します。	
4	MEDE	“1”でEDTE割込出力をマスク。センター割込ステータスは変化します。	
5	MERE	“1”でERAE割込出力をマスク。センター割込ステータスは変化します。	
6	MCSE	“1”でCAER割込出力をマスク。センター割込ステータスは変化します。	
7	(未定義)	常に“0”としてください。	
8	BKOF	“1”で、自動ブレーク機能を無効とします。	
9	MCLR	以下のステータスビットのクリア方法を選択します。 CEND, BRKF, EDTE, ERAE, CAER 0: ステータスリードによりクリアされます (デフォルト) 1: ステータスリードではクリアされません。 クリアするには、割込ステータスのクリアコマンド(04xxh)を使用します。 ステータスリードによる自動クリア機能を抑止することができます。	
15~10	(未定義)	常に“0”としてください。	

表4.1-11 RENV0レジスタの内容

(2) RERCNT (Read only)

エラーカウンタ用の16ビットレジスタです。無応答やCRCエラーなどの、通信エラー数を累計カウントします。エラー数が65535回以上となった場合、65535という数値でカウンタは停止します。カウンタをクリアするには、カウンタクリアコマンド(0600h)を発行してください。またシステム通信に対する無応答もエラーとしてカウントするので注意してください。(システム通信に対する無応答は、本来はエラーとしては扱っていません)

(3) RSYCNT (Read only)

サイクリック周期測定用の16ビットレジスタです。MSYN(サイクリック周期毎にレベル反転する信号)が変化する間の時間を、1 μ s単位でカウントします。カウントは常に行われ、直前のMSYN変化前のカウント数を参照できます。カウント値は65535(65535 μ s)が上限で、MSYN信号の幅がそれ以上になると測定できません。サイクリック通信時間の計算値は余裕時間を見ているため、通常は、このレジスタで計測した値は計算値より小さくなります。

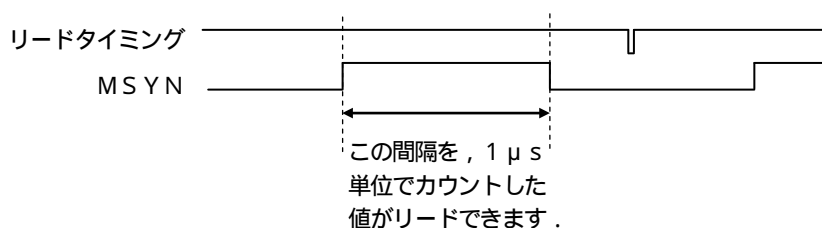


図4.1-4 RSYCNT

(4) RDJADD (Read only)

最後に正常受信したデータ通信のモジュールIDが保持されます。

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	最後に正常に受信したデータ通信 でのモジュールID					

図4.1-5 RDJADD

(5) RVER (Read only)

このレジスタは、G9001とG9001Aをソフトウェア的に区別するためのレジスタです。弊社motionCATマスターボードは全て最新版G9001Aが搭載されていますので、使用する必要はありません。常に以下の値(0001h)となります。

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	1

図4.1-6 RVER

4.2 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。

(1) 汎用入力ポート

(BAR+200H:DIN) (Read only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	IN8	IN7	IN6	IN5	IN4	IN3	IN2	IN1

図4.2-1 汎用入力ポートのビット構成

bit	記号	機 能	備 考
0~7	IN1~8	汎用入力状態 ONで"1"	
8~15	(未定義)	常に"1"	

表4.2-1 汎用入力ポートの内容

(2) 汎用出力設定ポート

(BAR+202H:DOUT) (Read/Write)

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

図4.2-2 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
0~3	OUT1~4	汎用出力設定 "1" でON, 汎用出力状態 ONで"1"	
4~7	(未定義)	常に"0"	
8~15	(未定義)	常に"1"	

表4.2-2 汎用出力設定及び出力状態確認ポートの内容

(3) G9001Aステータス

(BAR+204H:STS) (Read only)

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

図4.2-3 G9001Aステータス確認ポートのビット構成

bit	記号	機 能	備 考
0	xMCRY	信号ライン信号検出で一定時間Lレベル	
1	xMERR	異常フレーム受信時と、無応答時に一定時間Lレベル	
2	xMERF	エラー応答フレーム受信時一定時間Lレベル	
3	xMSYN	サイクリック周期毎にレベル反転	

表4.2-3 G9001Aステータス確認の内容

(4) 通信速度確認レジスタ

(BAR+06H:SPD) (Read only)

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

図4.2-4 G9001A通信速度確認ポートのビット構成

bit	記号	機 能			備 考
0	DSW1	通信設定SW bit0 ONで"1"			
1	DSW2	通信設定SW bit1 ONで"1"			
2, 3	SPD1.0	通信設定bit読み出し ONで"1"			
		SPD0	SPD1	通信速度	
		0	0	20Mbps	
		1	0	10Mbps	
		0	1	5Mbps	
		1	1	2.5Mbps	

表4.2-4 G9001A通信速度確認ポートの内容

(5) 割込みイネーブル

(BAR+208H:INTEN) (Read/Write)

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

図4.2-5 割込みイネーブルポートのビット構成

bit	記号	機 能	備 考
0	INTEN	割込み許可状態で“1”	
1~7	(未定義)	常に“0”	
8~15	(未定義)	常に“1”	

表4.2-5 割込みイネーブルポートの内容

(6) 割込み確認

(BAR+20AH:INTST) (Read only)

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

図4.2-6 割込み確認ポートのビット構成

Bit	記号	機 能	備 考
0	INTST	“1”で割込み有り，“0”で割込み無し	
1~7	(未定義)	常に“0”	
8~15	(未定義)	常に“1”	

表4.2-6 割込み確認ポートの内容

(7) 設定スイッチ確認ポート

(BAR+20CH:LDSW) (Read only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	0	0	EMG	予約	ALEON	IMEM	8512	816

図4.2-7 設定スイッチ確認ポートのビット構成

スイッチの設定状態及びEMG状態を確認できます。 ON:1 OFF:0

Bit	記号	機 能	備 考
0	816	スイッチ設定16bitの時 '1' が読み込み出来ます。 スイッチ設定8bitの時 '0' が読み込み出来ます。	
1	8512	スイッチ設定512バイトモードの時 '1' が読み込み出来ます。 スイッチ設定8バイトモードの時 '0' が読み込み出来ます。	
2	IMEM	スイッチ設定MEMの時 '1' が読み込み出来ます。 スイッチ設定I/Oの時 '0' が読み込み出来ます。	
3	ALEON	スイッチ設定ALE ONの時 '1' が読み込み出来ます。 スイッチ設定ALE OFFの時 '0' が読み込み出来ます。	
4	予約	予備スイッチ設定 ONの時 '1' が読み込み出来ます。 予備スイッチ設定 OFFの時 '0' が読み込み出来ます。	
5	EMG	EMGオプション 無し の時 '0' が読み込み出来ます。 EMGオプション付きで標準動作時 '0' が読み込み出来ます。 EMGオプション付きで非常停止時 '1' が読み込み出来ます。	
6~7	(未定義)	常に“0”	
8~15	(未定義)	常に“1”	

表4.2-7 設定スイッチ確認ポートの内容

4.3 ボード上の設定及びコネクタ位置

4.3.1 HPC104 - MCAT110M設定箇所

HPC104 - MCAT110Mの設定箇所は6箇所 コネクタは4箇所です。

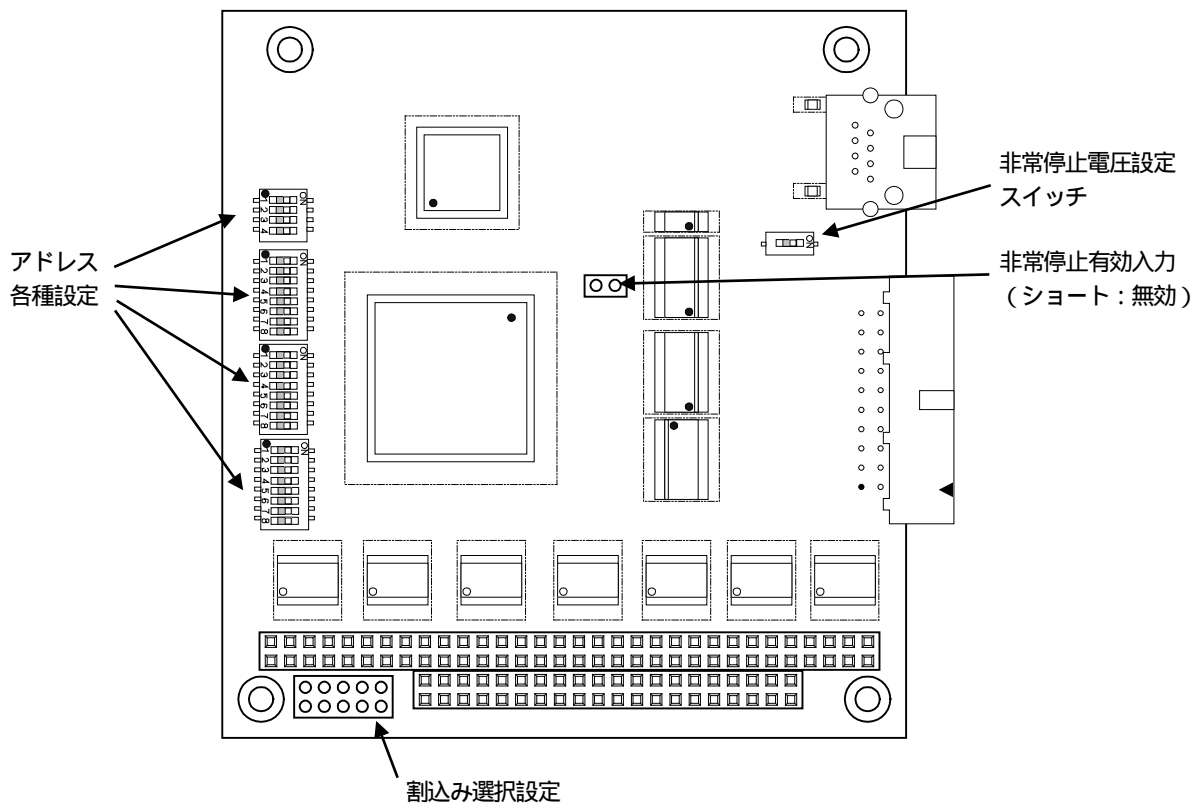


図4.3-1 HPC104 - MCAT110Mコネクタ, スイッチ位置

4.3.2 マスターボードの設定

(1) 割り込み選択設定

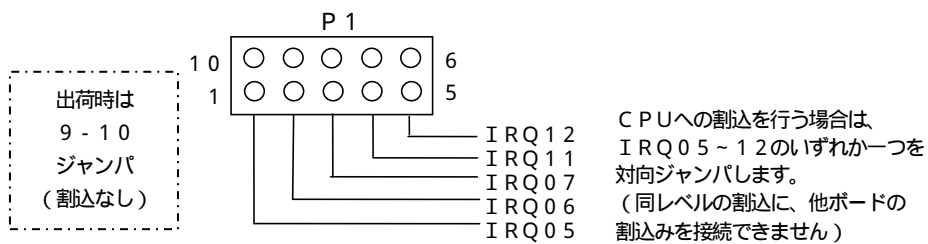
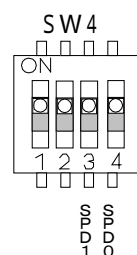


図4.3-2 MCAT110M 割り込みジャンパ端子

(2) 通信速度設定

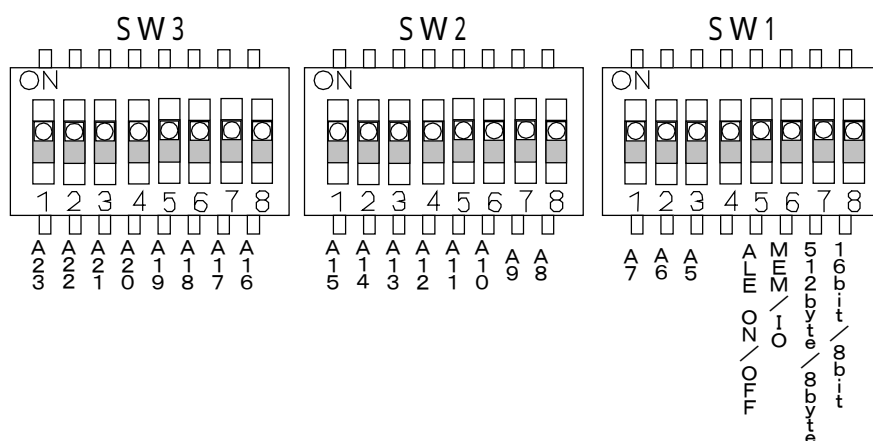
通信速度を設定します。接続するスレーブの通信速度設定も同じ値にします。



	SPD1	SPD0
2.5Mbps	ON	ON
5Mbps	ON	OFF
10Mbps	OFF	ON
20Mbps	OFF	OFF

図4.3-3 通信速度SW

(3) アドレス、バス幅、使用空間モード、アドレス空間の設定



名称	機能	ON	OFF
バス幅 16bit/8bit	CPUボードからのバスアクセスを設定します。 (使用されるCPUボードに合わせて設定願います)	16bit	8bit
使用空間モード 512バイトモード/ 8バイトモード	G9001Aへのアクセスモードを設定 512バイトアクセス設定 (全1024バイト空間) 8バイトアクセス設定 (全32バイト空間) 通常はこちらを御利用願います。 CPUボードのリソースが少ない 場合に御利用願います。	512バイト モード	8バイト モード
アドレス空間 MEM/I/O	ボードに対してメモリアクセス又はI/Oアクセスを切り替えます。 (メモリ:メモリーマップドI/Oアクセス) (I/O:I/OマップドI/Oアクセス)	メモリ	I/O
ALE信号 ON/OFF	ALEを使用してアドレスをラッチするかしないかを選択します。 (CPUボードによってはALE信号を出力できないボードがございます ので御注意願います。)	有効	無効
A9 - A5	アドレス設定 A9 - A5 (8バイトアクセス時のみ有効)	1	0
A15 - A10	アドレス設定 A15 - A10	1	0
A19 - A16	アドレス設定 A19 - A16 (メモリーマップドI/O時のみ有効)	1	0
A23 - A20	アドレス設定 A23 - A20 (メモリーマップドI/O時のみ有効)	1	0

表4.3-1 アドレス、バス幅、使用空間モード、アドレス空間の設定スイッチ機能説明

バス幅	アドレス空間	使用空間モード	ALE 信号	アドレス設定			
8/16bit	I/O/MEM	8/512バイトモード	ON/OFF	A23 - A20	A19 - A16	A15 - A10	A9 - A5
8	I/O	8	OFF	無効		有効	有効
8	I/O	8	ON				無効
8	I/O	512	OFF				無効
8	I/O	512	ON				無効
8	MEM	8	OFF	左記設定時（8bit/MEM）、 自動的にアドレス空間を I/O にしますので無効です			
8	MEM	8	ON				
8	MEM	512	OFF				
8	MEM	512	ON				
16	I/O	8	OFF	無効		有効	有効
16	I/O	8	ON				無効
16	I/O	512	OFF				無効
16	I/O	512	ON				無効
16	MEM	8	OFF	有効			
16	MEM	8	ON				
16	MEM	512	OFF				
16	MEM	512	ON				

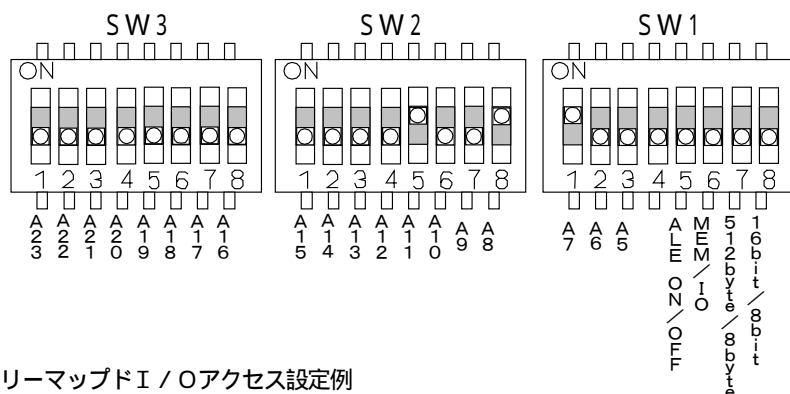
表4.3-2 設定スイッチによるアドレス スwitchの有効/無効状態

印: 設定スイッチの出荷設定です。

(4) アドレス, バス幅, 使用空間モード, アドレス空間の設定例

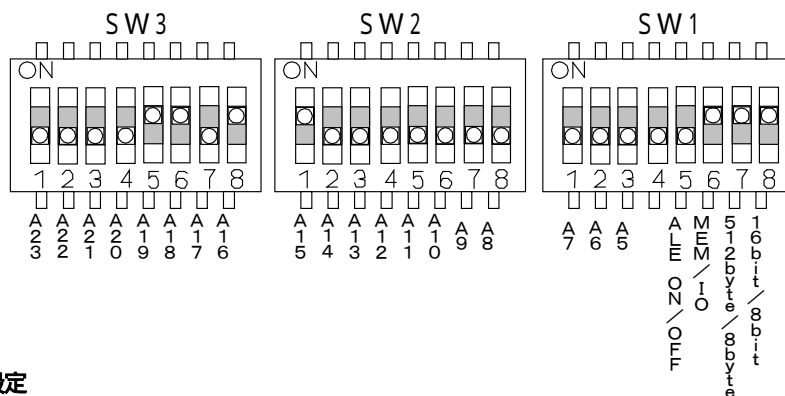
I/OマップドI/Oアクセス設定例 (出荷時設定)

- ・アドレス設定 : '000980H'
- ・ALE信号 : 使用しない
- ・アドレス空間 : I/OマップドI/Oアクセス
- ・使用空間モード : 8バイトモード (実使用空間: 32バイト空間)
- ・バス幅 : 8bit



メモリーマップドI/Oアクセス設定例

- ・アドレス設定 : '0D8000H'
- ・ALE信号 : 使用しない
- ・アドレス空間 : メモリーマップドI/Oアクセス
- ・使用空間モード : 512バイトモード (実使用空間: 1024バイト空間)
- ・バス幅 : 16bit



(5) 非常停止の設定

非常停止入力を使用する場合はショートピンをニッパーでカットしてください。

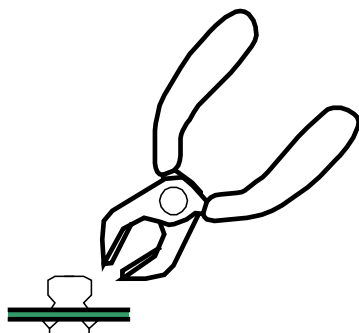
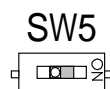


図4.3-6 非常停止 有効入力

非常停止回路に使用する電源に合わせてSW5を設定してください。

- 24V入力時 : OFF (出荷時)
- 12V入力時 : ON

図4.3-7 非常停止 12/24V切替え



4.3.3 汎用入出力とEMG入力回路

入出力回路を次表に示します

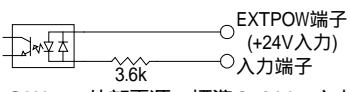
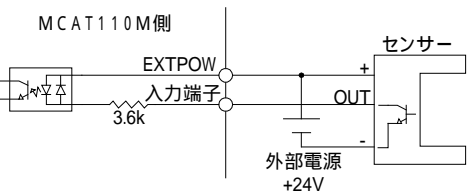

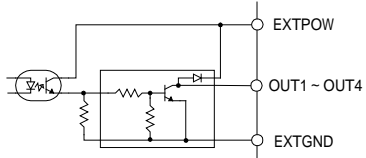
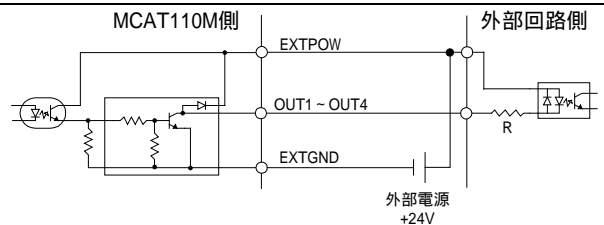
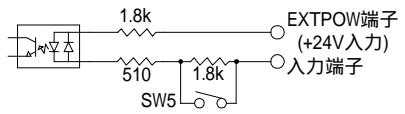
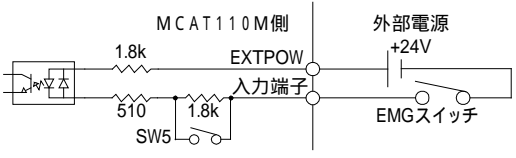
項	項 目		内 容
1	回路形式1	汎用入力 (IN1 ~ IN8)	 <p>EXTPOW: 外部電源 標準24V 入力端子</p>
2	外部との接続	フォトセンサ入力	
		リミットスイッチ入力	
3	回路形式2	汎用出力 (OUT1 ~ OUT4)	 <p>定格負荷電圧 DC12V ~ DC24V 使用負荷電流 80mA以下 / 1点 (但し, 4点合計負荷電流 150mA以下)</p>
4	外部との接続		
5	回路形式3	EMG入力	 <p>EXTPOW: 外部電源 標準24V 入力 (SW5 OFF)</p>
6	外部との接続	EMGスイッチ入力	

表4.3-2 汎用入出力回路とEMG入力回路

4.4 コネクタ信号表

(1) PC/104 BUS コネクタ : J1, J2

下記にPC/104コネクタ信号表を記載します。

信号表青部：常時使用している信号

信号表緑部：ディップスイッチの設定により変更される信号

信号表赤部：ジャンパ設定により変更される信号

J2		
Pin	Row D	ROW C
0	GND	GND
1	MEMCS16	SBHE
2	IOCS16	LA23
3	IRQ10	LA22
4	IRQ11	LA21
5	IRQ12	LA20
6	IRQ15	LA19
7	IRQ14	LA18
8	DACK0	LA17
9	DRQ0	MEMR
10	DACK5	MEMW
11	DRQ5	SD8
12	DACK6	SD9
13	DRQ6	SD10
14	DACK7	SD11
15	DRQ7	SD12
16	+5V	SD13
17	MASTER	SD14
18	GND	SD15
19	GND	KEY

J1		
Pin	Row A	Row B
1	IOCHK	GND
2	SD7	RESET
3	SD6	+5V
4	SD5	IRQ9
5	SD4	-5V
6	SD3	DRQ2
7	SD2	-12V
8	SD1	SRDY
9	SD0	+12V
10	IOCHRDY	KEY
11	AEN	SMEMW
12	SA19	SMEMR
13	SA18	IOW
14	SA17	IOR
15	SA16	DACK3
16	SA15	DRQ3
17	SA14	DACK1
18	SA13	DRQ1
19	SA12	REFRESH
20	SA11	BCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2
27	SA4	TC
28	SA3	BALE
29	SA2	+5V
30	SA1	OSC
31	SA0	GND
32	GND	GND

図4.4-1 PC/104 Bus コネクタ信号表

(2) Motionnet通信(RJ45) : J4

J4: Motionnet通信ライン

Pin	内 容
1	フレームアースにイベルダンスで接続
2	フレームアースにイベルダンスで接続
3	S+
4	予約
5	予約
6	S-
7	フレームアースにイベルダンスで接続
8	フレームアースにイベルダンスで接続

表4.4-1 RJ45(J4)コネクタ信号表

(3) 汎用入出力用 : J3

Pin	信号名	Pin	信号名
1	EMG +	2	EMG -
3	EXTPOW (+ 2.4V 入力)	4	EXTPOW (+ 2.4V 入力)
5	IN1	6	IN2
7	IN3	8	IN4
9	IN5	10	IN6
11	IN7	12	IN8
13	EXTPOW (+ 2.4V 入力)	14	EXTPOW (+ 2.4V 入力)
15	OUT1	16	OUT2
17	OUT3	18	OUT4
19	EXTGND (+ 2.4V GND)	20	EXTGND (+ 2.4V GND)

オムロン : XG4C-2034

推奨ハーネス(ケーブル側) 圧接タイプ : XG4M-2030-T

バラ線タイプ : XG5M-2032-N

表4.4-2 D I O部ピンアサイン(J3)

4.5 マスターボード仕様

4.5.1 マスターボード HPC104-MCAT110Mハードウェア仕様

区 分	項 目	仕 様	備 考
【基本仕様】	制御Line数 通信速度 通信方式 通信I/F 通信ケーブル 接続方式	最大1Line制御 1Line最大32Bit接続 (1Lineケーブル合計長50m以内) 20, 10, 5, 2.5Mbps 半2重通信 RS-485, パルストランス結合 LANケーブルCAT5e, CAT6(シールド付) マルチドロップ	制御LSI: G9001A (日本パルスマーター製)
【Bus仕様】	PC/104 Bus	PC/104 Specification Version 2.5 データバス 8/16ビット切替SW有	
【汎用入出力】	カプラ絶縁汎用入出力 入力ポート 出力ポート	8IN/4OUT D I Oポート 入力論理 : 入力フォトカプラON時 '1' 定格入力電圧 : DC12V~DC24V 使用入力電圧範囲: DC12V~DC25V 定格入力電流 : 5mA/1点 出力論理 : '1'書込時 出力ON 定格負荷電圧: DC12V~DC24V 負荷電流 : 80mA以下/1点	
【EMG入力】	カプラ絶縁EMG入力 EMG入力	1IN 入力論理 : 入力フォトカプラOFF時停止 定格入力電圧 : DC12V~DC24V 使用入力電圧範囲: DC12V~DC25V 定格入力電流 : 5mA	ジャンパーにより 許可/禁止 有り 出荷時 禁止
【周囲条件】	供給電源 消費電流 温度条件 ボード形寸	+5V±5% 400mA MAX 0 ~ 50 但し結露せぬこと 横90.2mm × 縦95.9mm(PC/104サイズ)	

表4.5-1 HPC104-MCAT110M 仕様

4.5.2 HPC104-MCAT110Mマスターボード形寸

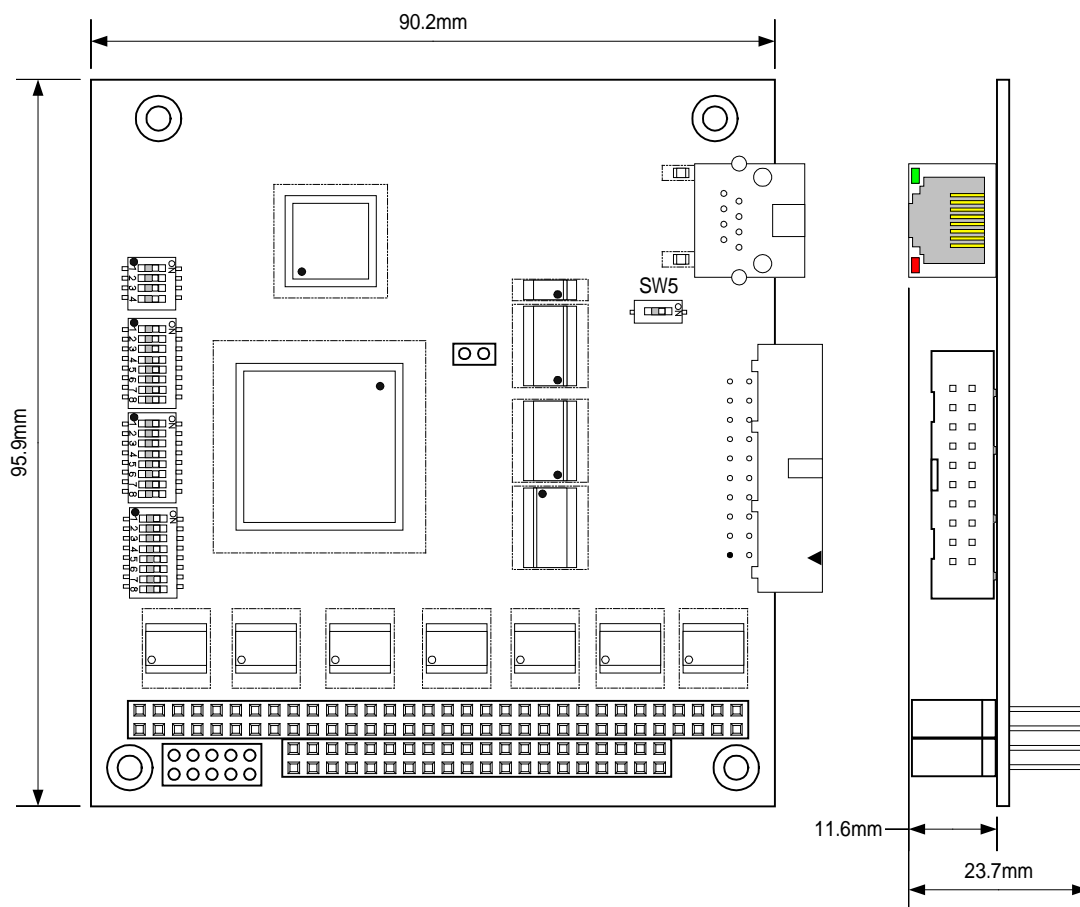


図4.5-2 HPC104-MCAT110Mボード形寸

5. ローカルデバイス

5.1 DIOデバイス (G9002)

DIOデバイスは、motionCATシステム用のI/Oデバイスです。

弊社motionCATシリーズ製品ではDIOモジュールに搭載され、センターデバイスから入力ポート2ポート、出力ポート2ポートの制御ができます。(1ポートは8ビット)

5.1.1 DIOモジュールI/Oポート

motionCAT DIOモジュールのポートは4つのポートにより構成されており、ポート3、ポート2が出力、1、0が入力となっています。

汎用入力状態は「1」でON中、「0」でOFF中となります。

汎用出力状態は「1」でON中、「0」でOFF中となります。

汎用出力設定は「1」書き込みでON、「0」書き込みでOFFとなります。(初期化時はすべて0)

	ポート3	ポート2	ポート1	ポート0
読み出し	汎用出力状態 (OUT16-9)	汎用出力状態 (OUT8-1)	汎用入力状態 (IN16-9)	汎用入力状態 (IN8-1)
書き込み	汎用出力設定 (OUT16-9)	汎用出力設定 (OUT8-1)	不使用	不使用

図5.1-1 DIOモジュールポート割付

5.2 モーションデバイス (G9003)

モーションデバイス (G9003) は、motionCATシステム用の軸制御デバイスです。

センターデバイス (G9001A) からの通信による各種コマンドにより、ステッピングモータ、サーボモータ駆動用の高速パルスが出力できます。定速、直線加減速、S字加減速により、多種多様な連続動作、位置決め動作、原点復帰動作等の制御が行えます。また、通信により動作状態確認が行えます。

また、この節のレジスタのビット説明で、「0」は書き込み時「0」以外禁止、及び読み出し時「0」固定を表します。

5.2.1 特長

(1) 通信

伝送速度は最大で20Mbpsです。

(2) 加減速制御

直線加減速とS字加減速が行えます。

S字加減速時には、中間部分に直線加減速部分を付けられます。(S字範囲設定)

(3) 速度のオーバーライド

全ての動作モードにおいて、動作中に速度の変更ができます。

(4) 位置のオーバーライド

位置決めモードで動作中に目標位置(移動量)の変更ができます。

既に新データ位置を通過している時は減速停止(定速動作時は即停止)後、逆方向に動作を開始し、位置決めを行います。

(5) 三角駆動回避機能 (FH補正機能)

位置決めモードにおいて、移動量が少ない時に最高速度を自動的に低下させて三角駆動を回避します。

(6) 豊富なカウンタ回路

下記の 3 カウンタがあります .

名 称	ビット長	説 明	カウンタ入力
C T R 1 指令位置カウンタ	2 8	指令位置管理用カウンタ	指令パルス
C T R 2 機械位置カウンタ	2 8	機械位置管理用カウンタ	エンコーダ (パルス) 入力 指令パルス
C T R 3 汎用・偏差カウンタ	1 6	指令位置と機械位置との偏差カウンタ コンパレータ一致出力用カウンタ 定ピッチ信号出力用カウンタ	指令パルス エンコーダ (パルス) 入力 基準加算の 1/4069 指令パルスとエンコーダ (パルス) 入力の偏差

全てのカウンタは、コマンド書き込み、及び C L R 信号入力によりリセットできます .

また、コマンド書き込み、L T C H 入力、O L S 入力によりカウンタデータをラッチする事ができます .

C T R 3 は、指定したカウンタ範囲を繰り返すリングカウンタ機能があります .

(7) コンパレータ機能

3 個のコンパレータがあり、設定値と内部カウンタ値との比較を行えます .

比較できるカウンタは、C T R 1 (指令位置カウンタ)、C T R 2 (機械位置カウンタ)、C T R 3 (汎用・偏差カウンタ)より選択できます .

(8) ソフトリミット機能

C M P 1、C M P 2 を使用してソフトリミットの設定ができます .

ソフトリミット範囲に入ると即停止または減速停止します . その後は、逆方向のみ動作できます .

(9) バックラッシュ補正

バックラッシュ補正機能があります .

バックラッシュ補正は動作方向が変わる毎に移動量の補正を行います .

(1 0) 定ピッチ出力機能

指定した一定間隔毎にパルス信号を出力する事ができます .

(1 1) 振動抑制機能 (パルスモータ用)

予め、制御定数を指定しておいて、停止直前に逆転と正転の 2 パルスを付加します .

この機能により停止時の振動を低減することができます .

(1 2) 手動パルス入力機能

手動パルスの信号をモジュールのエンコーダ入力端子に入力して直接モータを動作させる事ができます .

入力信号は、9 0 度位相差信号 (1 , 2 , 4 通倍)、または、アップ信号とダウン信号です .

E L S、ソフトリミット設定は有効で、指令パルス出力は停止します . 反対方向へは動作できます .

(1 3) パルスモータ脱調検出機能

指令パルスとエンコーダ信号 (E A / E B) により動作する偏差カウンタ (C T R 3) があります .

コンパレータを使用してパルスモータの脱調検出や位置決め確認に使用できます .

(1 4) 指令パルス仕様

共通指令パルスモード (共通パルス + 方向信号) と個別指令パルスモード (C W / C C W) とから選択できます .

(1 5) アイドリングパルス出力機能 (パルスモータ用)

加速スタート時に自起動周波数 (F L) のパルス数を設定する事ができます .

パルスモータの加減速制御で、初速を高め設定した場合に脱調しにくくなります .

(1 6) 動作モード

基本動作は、連続動作、位置決め動作、原点復帰動作です .

オプション的な動作モードビットの設定により、様々な動作を行えます .

<動作モード例>

コマンドによるスタート/ストップ .

手動パルスによる連続動作、位置決め動作 .

原点復帰動作

コマンドによる位置決め動作 .

(1 7) 多様な原点復帰シーケンス

<原点復帰例>

定速動作でOLS信号ONで停止	(センサー原点)
加減速動作でDLS信号ONで減速し、OLS信号ONで停止。	(センサー原点)
加減速動作でOLS信号ONで減速停止後逆転しOLS拔出し後、定速で再突入	(センサー原点)
定速動作でOLS信号ON後のEZ信号カウントで停止。	(センサー + Z相)
定速動作でOLS信号ONで逆転し、EZ信号カウントで停止。	(センサー + Z相)
加減速動作でOLS信号ONで減速し、EZ信号カウントで停止。	(センサー + Z相)
加減速動作でOLS信号ONで減速停止後逆転し、EZカウントで停止。	(センサー + Z相)
定速動作でELS信号ONで停止。(正常停止)	(ELS兼用)
定速動作でELS信号ONで逆転し、EZ信号カウントで停止。	(ELS兼用 + Z相)
加減速動作でELS信号ONで減速停止後逆転し、EZ信号カウントで停止。	(ELS兼用 + Z相)

5 . 2 . 2 モーションモジュールI / Oポート

motionCATモーションモジュールのポートは4つのポートにより構成されており、最上位のポート3が出力、ポート2, 1, 0が入力となっています。

下図のように、上位から、汎用出力設定、汎用出力状態、モーションメインステータスの上位バイト、下位バイトになっています。但し、汎用出力設定、汎用出力状態はRENV2レジスタ設定の初期化後有効となります。

ポート3	ポート2	ポート1	ポート0
汎用出力設定ポート IOPOB	汎用出力状態ポート IOPIB	モーション メインステータス MMSTS(bit15~8)	モーション メインステータス MMSTS(bit7~0)

図 5 . 2 - 1 モーションモジュールポート割付

5 . 2 . 3 モーションメインステータス (MMSTS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	GRP2	GRP1	GRP0	0	0	0	SBSY	0	0	0	0	SEVT	SERR	SEND	SINT

図 5 . 2 - 2 モーションメインステータス (MMSTS) のビット構成

bit	記 号	機 能	備 考
0	SINT	MMSTSのbit1, 2, 3のいずれかが1で1になります。	
1	SEND	動作停止により1になります。RENV1のbit28=1の時スタート時にリセットされます。または割り込みリセットコマンド(0008h)で0になります。	
2	SERR	エラー停止、位置のオーバーライド失敗、エンコーダ信号異常発生により1になります。REST読み出しで0になります。	
3	SEVT	RIRQで設定されたイベント発生により1になります。RIST読み出しで0になります。	
7 ~ 4	(未定義)	常に“0”	
8	SBSY	パルス出力開始で1になります。動作停止で0になります。(=BSY)	
11 ~ 9	(未定義)	常に“0”	
14 ~ 12	GRP2 ~ GRP0	グループ 設定状態 無所属グループ : 000, グループ 1 : 001, グループ 2 : 010 グループ 3 : 011, グループ 4 : 100, グループ 5 : 101 グループ 6 : 110, グループ 7 : 111	
15	(未定義)	常に“0”	

表 5 . 2 - 1 モーションメインステータス (MMSTS) の内容

注 . MMSTSはサイクリック通信により更新されます。従って実際のG9003の状態が反映されるのに最大サイクリック通信の周期分の遅れが生じます。

5.2.4 汎用出力状態ポート (I O P I B)

このポートは R E N V 2 レジスタ設定の初期化後有効となります。

7	6	5	4	3	2	1	0
ELL	GRP2	GRP1	GRP0	SVGAIN	SVTL	SVRST	SVON

図 5.2-3 汎用出力状態ポート (I O P I B) のビット構成

bit	記号	機能	備考
0	SVON	サーボON出力状態 (' 1 ' でON)	
1	SVRST	サーボリセット出力状態 (' 1 ' でON)	
2	SVTL	サーボトルク制御出力状態 (' 1 ' でON)	
3	SVGAIN	サーボゲイン切替出力状態 (' 1 ' でON)	
4	GRP0	グループ設定状態 無所属グループ : 111 , グループ 1 : 110 , グループ 2 : 101 グループ 3 : 100 , グループ 4 : 011 , グループ 5 : 010 グループ 6 : 001 , グループ 7 : 000	
5	GRP1		
6	GRP2		
7	ELL	ELS極性 (0 : A接 , 1 : B接)	

表 5.2-2 汎用出力状態ポート (I O P O B) の内容

注 . I O P I B はサイクリック通信により更新されます . 従って実際の G 9 0 0 3 の状態が反映されるのに最大サイクリック通信の周期分の遅れが生じます .

5.2.5 汎用出力ポート (I O P O B)

このポートは R E N V 2 レジスタ設定の初期化後有効となります。

7	6	5	4	3	2	1	0
ELL	GRP2	GRP1	GRP0	SVGAIN	SVTL	SVRST	SVON

図 5.2-4 汎用出力設定ポート (I O P O B) のビット構成

bit	記号	機能	備考
0	SVON	サーボON出力 (' 1 ' でON)	
1	SVRST	サーボリセット出力 (' 1 ' でON)	
2	SVTL	サーボトルク制御出力 (' 1 ' でON)	
3	SVGAIN	サーボゲイン切替出力 (' 1 ' でON)	
4	GRP0	グループ設定 無所属グループ : 111 , グループ 1 : 110 , グループ 2 : 101 グループ 3 : 100 , グループ 4 : 011 , グループ 5 : 010 グループ 6 : 001 , グループ 7 : 000	
5	GRP1		
6	GRP2		
7	ELL	ELS極性切り替え (0 : A接 1 : B接)	

表 5.2-3 汎用出力ポート (I O P O B) の内容

注 . I O P O B はサイクリック通信により G 9 0 0 3 に反映します . 従って G 9 0 0 3 へ反映されるのに最大サイクリック通信周期分の遅れが生じます .

5.2.6 モーション動作コマンド・コントロールコマンド・レジスタ制御コマンド

センタースタイルからデータ通信により , 以下のコマンドを設定することにより軸制御が行えます .

コマンドにはデータ無しのコマンドとデータ (レジスタデータ) 付きのコマンドがあります .

データ無しのコマンドにデータを付けて通信を行った場合 , データは無視されます .

データ付きコマンドで , 有効ビット数以上のデータの通信を行った場合 , 有効ビット数以上のデータは無視されます .

また , データ付きコマンドにデータを設定せずにセンタースタイルから通信を行った場合 , データは更新されません .

5ワード (1ワードは2バイト) 以上のデータの通信を行った場合 , センタースタイル側でローカル側受信処理エラー (E R A E) が発生します . この時コマンドは更新されません .

(1) モーション動作コマンド

スタートコマンド

スタートコマンド

停止時に書き込むとスタートします。

残量スタートコマンド

位置決め動作を途中停止させた後に書き込むと、位置決めカウンタの残パルス数分動作します。

移動量付きスタートコマンド

停止時に移動量とスタートコマンドを書き込みます。移動量は位置決め動作時に有効です。

移動量を設定せずに通信を行った場合、RMVレジスタデータ(移動量データ)は0となり、移動量0の動作になります。

同時スタートコマンド

RMDのb14 = 1の時、動作をスタート

コマンド名	記号	1ワード目 hex	2ワード目 hex	3ワード目 hex	内容	応答
FL定速スタート	STAF L	0050			FL定速スタート	応答ルームのみ
FH定速スタート	STAF H	0051			FH定速スタート	応答ルームのみ
加速スタート	STAUD	0053			加速スタート(加速 FH定速 減速)	応答ルームのみ
残量FL定速スタート	CNTFL	0054			残量FL定速スタート	応答ルームのみ
残量FH定速スタート	CNTFH	0055			残量FH定速スタート	応答ルームのみ
残量加速スタート	CNTUD	0057			残量加速スタート	応答ルームのみ
移動量付 FL定速スタート	RMSTFL	0058	RMV 下位データ	RMV 上位データ	RMVレジスタ書き込み+FL定速スタート	応答ルームのみ
移動量付 FH定速スタート	RMSTFH	0059	RMV 下位データ	RMV 上位データ	RMVレジスタ書き込み+FH定速スタート	応答ルームのみ
移動量付 加速スタート	RMSTUD	005B	RMV 下位データ	RMV 上位データ	RMVレジスタ書き込み+加速スタート	応答ルームのみ
同時スタート	CMSTA	0006			STA出力(スレーブ内で有効)	応答ルームのみ
	SPSTA	002A			STA入力代行(自軸のみ有効)	応答ルームのみ

表5.2-4 動作コマンド

速度変更コマンド

動作中に書き込むと動作速度を変更します。停止中の書き込みは無視されます。

コマンド名	記号	1ワード目 hex	2ワード目 hex	3ワード目 hex	内容	応答
FL定速へ瞬時速度変更	FCHGL	0040			FL定速へ瞬時速度変更	応答ルームのみ
FH定速へ瞬時速度変更	FCHGH	0041			FH定速へ瞬時速度変更	応答ルームのみ
FL速度減速	FSCHL	0042			FL速度まで減速	応答ルームのみ
FH速度加速	FSCHH	0043			FH速度まで加速	応答ルームのみ

表5.2-5 速度変更コマンド

停止コマンド

動作中に書き込むと停止します。

コマンド名	記号	1ワード目 hex	2ワード目 hex	3ワード目 hex	内容	応答
即停止	STOP	0049			即停止	応答ルームのみ
減速停止	SDSTP	004A			減速停止	応答ルームのみ

表5.2-6 停止コマンド

コマンド名	記号	1ワード目 hex	2ワード目 hex	3ワード目 hex	内容	応答
同時停止(STP出力)	CMSTP	0007			同時停止(STP出力)	応答ルームのみ

表5.2-7 停止コマンド

(2) モーションコントロールコマンド

カウンタのリセット等の各種コントロールを行います。

コマンド名	記 号	1ワード目 hex	2ワード目 hex	3ワード目 hex	内 容	応 答
NOPコマンド	NOP	0000			無効コマンド	応答ルームのみ
SEND割り込み リセット	INTRS	0008			MMSTS bit1 (SEND) の リセット	応答ルームのみ
ソフトウェア リセット	SRST	0004			ソフトウェアリセット モーションデバイス (G9003) の レジスタおよびコマンドをリセットし ます (通信関連を除く)	応答ルームのみ
カウンタリセット	CUN1R	0020			CTR1 (指令位置) リセット	応答ルームのみ
	CUN2R	0021			CTR2 (機械位置) リセット	応答ルームのみ
	CUN3R	0022			CTR3 (汎用・偏差) リセット	応答ルームのみ
サーボ偏差カウンタ クリア出力制御	SVCTR CLOUT	0024			SVCTRCL信号の出力	応答ルームのみ
	SVCTR CLRST	0025			SVCTRCL信号のリセット	応答ルームのみ
PCSコマンド	PCSON	0028			PCS信号の入力代行 RMDレジスタ (動作モード) でPCS 入力有効時に, PCS入力ONと同等の 処理を行います。	応答ルームのみ
カウンタラッチ	LTCR	0029			LTCR信号の入力代行	応答ルームのみ

表5.2-8 モーションコントロールコマンド

(3) モーションレジスタ制御コマンド

モーションレジスタ書込みコマンド

コマンド名	記 号	1ワード目 hex	2ワード目 hex	3ワード目 hex	応 答
RMVオーバライド書込み	WRMVOR	0080	下位データ	上位データ	応答ルームのみ
RMV書込み	WRMV	0090	下位データ	上位データ	応答ルームのみ
RFL書込み	WRFL	0091	下位データ	上位データ	応答ルームのみ
RFH書込み	WRFH	0092	下位データ	上位データ	応答ルームのみ
RUR書込み	WRUR	0093	データ		応答ルームのみ
RDR書込み	WRDR	0094	データ		応答ルームのみ
RMG書込み	WRMG	0095	データ		応答ルームのみ
RDP書込み	WRDP	0096	下位データ	上位データ	応答ルームのみ
RMD書込み	WRMD	0097	下位データ	上位データ	応答ルームのみ
RUS書込み	WRUS	0099	データ		応答ルームのみ
RDS書込み	WRDS	009A	データ		応答ルームのみ
RFA書込み	WRFA	009B	下位データ	上位データ	応答ルームのみ
RENV1書込み	WRENV1	009C	下位データ	上位データ	応答ルームのみ
RENV2書込み	WRENV2	009D	下位データ	上位データ	応答ルームのみ
RENV3書込み	WRENV3	009E	下位データ	上位データ	応答ルームのみ
RENV4書込み	WRENV4	009F	下位データ	上位データ	応答ルームのみ
RENV5書込み	WRENV5	00A0	下位データ	上位データ	応答ルームのみ
RENV6書込み	WRENV6	00A1	下位データ	上位データ	応答ルームのみ
RCTR1書込み	WRCTR1	00A3	下位データ	上位データ	応答ルームのみ
RCTR2書込み	WRCTR2	00A4	下位データ	上位データ	応答ルームのみ
RCTR3書込み	WRCTR3	00A5	データ		応答ルームのみ
RCMP1書込み	WRCMP1	00A7	下位データ	上位データ	応答ルームのみ
RCMP2書込み	WRCMP2	00A8	下位データ	上位データ	応答ルームのみ
RCMP3書込み	WRCMP3	00A9	下位データ	上位データ	応答ルームのみ
RIRQ書込み	WRIRQ	00AC	下位データ	上位データ	応答ルームのみ

表5.2-9 モーションレジスタ書込みコマンド

モーションレジスタ読出しコマンドと応答データ

コマンド名	記 号	コマンド	応答データ				
		1ワード目 hex	応答データの種類	1ワード目 hex	2ワード目 hex	3ワード目 hex	有効 ビット数
RMV読出し	RRMV	00D0	ARMV データ	00D0	下位データ	上位データ	28
RFL読出し	RRFL	00D1	ARFL データ	00D1	下位データ	上位データ	17
RFH読出し	RRFH	00D2	ARFH データ	00D2	下位データ	上位データ	17
RUR読出し	RRUR	00D3	ARUR データ	00D3	データ	0	16
RDR読出し	RRDR	00D4	ARDR データ	00D4	データ	0	16
RMG読出し	RRMG	00D5	ARMG データ	00D5	データ	0	11
RDP読出し	RRDP	00D6	ARDP データ	00D6	下位データ	上位データ	24
RMD読出し	RRMD	00D7	ARMD データ	00D7	下位データ	上位データ	25
RUS読出し	RRUS	00D9	ARUS データ	00D9	データ	0	16
RDS読出し	RRDS	00DA	ARDS データ	00DA	データ	0	16
RFA読出し	RRFA	00DB	ARFA データ	00DB	下位データ	上位データ	17
RENV1読出し	RRENV1	00DC	ARENV1 データ	00DC	下位データ	上位データ	30
RENV2読出し	RRENV2	00DD	ARENV2 データ	00DD	下位データ	上位データ	23
RENV3読出し	RRENV3	00DE	ARENV3 データ	00DE	下位データ	上位データ	31
RENV4読出し	RRENV4	00DF	ARENV4 データ	00DF	下位データ	上位データ	28
RENV5読出し	RRENV5	00E0	ARENV5 データ	00E0	下位データ	上位データ	32
RENV6読出し	RRENV6	00E1	ARENV6 データ	00E1	下位データ	上位データ	32
RCTR1読出し	RRCTR1	00E3	ARCTR1 データ	00E3	下位データ	上位データ	28
RCTR2読出し	RRCTR2	00E4	ARCTR2 データ	00E4	下位データ	上位データ	28
RCTR3読出し	RRCTR3	00E5	ARCTR3 データ	00E5	データ	符号拡張	16
RCMP1読出し	RRCMP1	00E7	ARCMP1 データ	00E7	下位データ	上位データ	28
RCMP2読出し	RRCMP2	00E8	ARCMP2 データ	00E8	下位データ	上位データ	28
				00E9	下位データ	上位データ	28
RCMP3読出し	RRCMP3	00E9	ARCMP3 データ				
RIRQ読出し	RRIRQ	00EC	ARIRQ データ	00EC	データ	0	15
RLTC1読出し	RRLTC1	00ED	ARLTC1 データ	00ED	下位データ	上位データ	28
RLTC2読出し	RRLTC2	00EE	ARLTC2 データ	00EE	下位データ	上位データ	28
RDTC3読出し	RRLTC3	00EF	ARLTC3 データ	00EF	下位データ	上位データ	17
RSTS読出し	RRSTS	00F1	ARSTS データ	00F1	下位データ	上位データ	23
REST読出し	RREST	00F2	AREST データ	00F2	データ	0	15
RIST読出し	RRIST	00F3	ARIST データ	00F3	データ	0	15
RPLS読出し	RRPLS	00F4	ARPLS データ	00F4	下位データ	上位データ	28
RSPD読出し	RRSPD	00F5	ARSPD データ	00F5	下位データ	上位データ	27
RSDC読出し	RRSDC	00F6	ARSDC データ	00F6	下位データ	上位データ	24

表5. 2 - 10 モーションレジスタ読出しコマンド

注 1. レジスタ読み出しコマンドに4ワード目までデータを付加して通信を行った場合は無応答となり、センタデバイス側でデータ通信エラー(EDTE)が発生します。

5.2.7 モーションデバイスレジスタ

すべてのレジスタの初期値は"0"です。設定すべき値が前回と同じ時には、再度書き込む必要はありません。

< 表記説明 >

*表示のビットは、書き込み時には無視され、読み出し時には0になります。

&表示のビットは、書き込み時には無視され、読み出し時には空欄表示中の最上位ビットと同一になります。

(符号拡張)

(1) RMV: 移動量レジスタ (28ビット)

位置決め動作において移動量を設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図5.2-5 RMV: 移動量レジスタのビット構成

動作モードにより設定内容が変わります。設定範囲は -134,217,728 ~ +134,217,727 です。

動作中にRMVレジスタを変更する事により、位置のオーバーライドが行えます。

(2) RFL: ベース速度レジスタ (17ビット)

加減速付き動作において、ベース速度 (初速, 停止速度) を設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	

図5.2-6 RFL: ベース速度レジスタのビット構成

RFL定速動作の速度及び、加減速動作の場合のベース速度を 1 ~ 100,000(186A0h)の範囲で設定します。

100,000 ~ 131,071(186A0h ~ 1FFFFh) の範囲は全て 100,000 と見なされます。

実際の速度はRMG (速度倍率設定レジスタ) の設定値との計算値になります。

$$\text{ベース速度[pps]} = \text{RFL} \times \frac{200}{(\text{RMG} + 1)}$$

(3) RFH: 動作速度設定レジスタ (17ビット)

動作速度を設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	

図5.2-7 RFH: 動作速度レジスタのビット構成

動作中にRFHレジスタを変更する事により、速度のオーバーライドが行えます。

RFH定速動作の速度及び、加減速動作の場合の動作速度を 1 ~ 100,000(186A0h)の範囲で設定します。

100,000 ~ 131,071(186A0h ~ 1FFFFh) の範囲は全て 100,000 と見なされます。

加減速動作の場合にはRFLの設定値よりも大きな値を設定して下さい。

実際の速度はRMG (速度倍率設定レジスタ) の設定値との計算値になります。

$$\text{動作速度[pps]} = \text{RFH} \times \frac{200}{(\text{RMG} + 1)}$$

(4) RUR : 加速レートレジスタ (16ビット)

加速レートを設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

図5.2-8 RUR : 加速レートレジスタのビット構成

加減速動作の場合の加速特性を1~65,535(FFFFh)の範囲で設定します。

(5) RDR : 減速レートレジスタ (16ビット)

減速レートを設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

図5.2-9 RDR : 減速レートレジスタのビット構成

加減速動作の場合の減速特性を通常は1~65,535(FFFFh)の範囲で設定します。

減速開始点を自動(RMDレジスタのMSDP(b12)=0)に設定する場合でも、RDRレジスタの設定値が減速レートとして使用されます。RDR=0に設定した場合、減速レートはRURで設定した値になります。

(6) RMG : 速度倍率設定レジスタ (11ビット)

速度倍率を設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*										

図5.2-10 RMG : 速度倍率設定レジスタのビット構成

RFL, RFH及びRFA設定値と速度の関係を2~2,047(07FFh)の範囲で設定します。

高倍率になるほど設定できる速度間隔は粗くなります。

動作速度[PPS]は、速度倍率と速度レジスタ設定値との積になります。

(7) RDP : 減速開始点レジスタ (24ビット)

位置決め動作において、減速開始点を設定するレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
#	#	#	#	#	#	#	#																								

図5.2-11 RDP : 減速開始点レジスタのビット構成

加減速・位置決め動作の場合の減速開始点を決定するための値を設定します。

#表示のビットは、書き込み時には無視され、読み出し時にはRMDレジスタのMSDP(b12)設定により設定内容は変化します。

MSDP	設 定 内 容	#ビット
0	自動設定値に対するオフセットとなります。 正数の時は早めに減速開始してベース速度(F L速度)区間が長くなり、 負数の時は遅めに減速開始してベース速度(F L速度)に到達しなくなります。	ビット23と同じ (符号拡張)
1	移動残量が設定値以下になった時に減速を開始します。	0

(8) RMD:動作モードレジスタ(25ビット)

動作モードを設定するレジスタです。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSPE	MSY	MPCS	MSDP	METM	MSMD	MINP	MSDE	0							MOD
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	MIOR	MFH	MUB	MMPH	MPH	MINT	MMSK	MADJ	MSPO

図5.2-12 RMD:動作モードレジスタのビット構成

bit	記号	内 容
6 ~ 0	MOD	動作モードを設定します。
		MOD hex 動作モード
		0 0 コマンド制御による (+)方向連続動作
		0 8 コマンド制御による (-)方向連続動作
		1 0 (+)方向 原点復帰動作
		1 8 (-)方向 原点復帰動作
		1 2 (+)方向 原点抜け出し動作
		1 A (-)方向 原点抜け出し動作
		1 5 (+)方向 原点サーチ動作
		1 D (-)方向 原点サーチ動作
		2 0 + E Lまたは+ S L位置まで動作
		2 8 - E Lまたは- S L位置まで動作
		2 2 - E Lまたは- S L抜け出し動作
		2 A + E Lまたは+ S L抜け出し動作
		2 4 (+)方向に E Zカウント分だけ動作
		2 C (-)方向に E Zカウント分だけ動作
		4 1 位置決め動作(目標相対位置指定)
		4 4 指令位置(CTR1)0点(復帰動作)
		4 5 機械位置(CTR2)0点(復帰動作)
		4 6 (+)方向 1/Vレズ動作
		4 E (-)方向 1/Vレズ動作
		4 7 タイマ動作
		0 1 パルス(PA/PB)入力による連続動作
		5 1 パルス(PA/PB)入力による位置決め動作
		5 4 パルス(PA/PB)入力による指令位置0点(復帰動作)
		5 5 パルス(PA/PB)入力による機械位置0点(復帰動作)
7	未定義	(常に0に設定して下さい。)
8	MSDE	0: DLS入力は無効になります。(RSTSでの確認は可) 1: DLS入力ONにより減速(減速停止)します。
9	MINP	0: INPOS入力による動作完了遅延は無効になります。(RSTSでの確認は可) 1: INPOS入力ONで動作完了になります。
10	MSMD	加減速動作時の加減速特性を設定。(0:直線加減速 1: S時加減速)
11	METM	動作完了タイミングを設定。(0:周期完了 1:パルス完了) 振動抑制機能を使用する時は、パルス完了にします。
12	MSDP	加減速動作時の減速開始点を設定。位置決め動作時に有効。(0:自動設定 1:手動設定)(注1)
13	MPCS	PCS入力の有効/無効設定。 1: 位置決め動作時に、PCS(SVRDY)入力ONからパルス数管理を行います。 SVRDY信号をサーボレディとして使用している場合は0に設定して下さい。
14	MSY	0:即スタート 1: STA入力、または同時スタートコマンドでスタート
15	MSPE	1: STP入力または同時停止コマンドで停止。停止方法はRENV1のb19で設定
16	MSPO	1: 異常停止時にSTP信号自動出力
17	MADJ	FH補正機能の設定。(0:ON 1:OFF)
18	MMSK	1: パルス出力をマスクします。
19	MINT	1: 割込出力(SINT)をマスク。(インターナ、バートナは変化します。)
23 ~ 20	-	(予約: 常に0を設定して下さい。)
24	MIOR	汎用 I/O ポートの出力設定ビットのモタ方法を選択します。 出力ポートの変化時に、セクタバスの入力変化割り込みを動作させたくない時に使用します。 0: 出力設定ビットの状態をポート2から読み出せます。 1: 出力設定ビットの状態に関わらず、ポート2の対応ビットは"0"になります。
31 ~ 25	未定義	(常に0を設定して下さい。)

表5.2-11 RMD:動作モードレジスタの内容

注1. 減速開始点自動設定で(減速時間) > (加速時間 × 2)に設定した場合、FLまで減速しきれません。
減速時間が加速時間の2倍を越える場合は、減速開始点をマニュアル設定にして下さい。

(9) RUS : 加速S字区間レジスタ (16ビット)

S字加速時のS字区間を設定するレジスタです .

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

図 5 . 2 - 1 3 RUS : 加速S字区間レジスタのビット構成

S字加減速動作のS字加速区間を 1 ~ 50,000 (C350h) の範囲で設定します .

50,000 ~ 65,535 (C350h ~ FFFFh) の範囲は全て 50,000 と見なされます .

S字加速区間の範囲 S_{su} は R M G (速度倍率設定レジスタ) の設定値との計算値になります .

"0" を設定した場合 , 内部演算により $(RFH-RFL)/2$ が代用され , 直線加速部分のないS字加速動作となります .

$(RFH-RFL)/2$ より大きな値を設定した場合 , 最大加速度まで到達しなくなり , 加速時間が計算式と異なります .

このため $(RFH-RFL)/2$ 以下の値を設定して下さい .

(10) RDS : 減速S字範囲設定レジスタ (16ビット)

S字減速時のS字区間を設定するレジスタです .

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

図 5 . 2 - 1 4 RDS : 減速S字区間レジスタのビット構成

S字加減速動作のS字減速区間を 1 ~ 50,000 (C350h) の範囲で設定します .

50,000 ~ 65,535 (C350h ~ FFFFh) の範囲は全て 50,000 と見なされます .

S字減速区間の範囲 S_{sd} は R M G (速度倍率設定レジスタ) の設定値との計算値になります .

"0" を設定した場合 , 内部演算により $(RFH-RFL)/2$ が代用され , 直線減速部分のないS字減速動作となります .

$(RFH-RFL)/2$ より大きな値を設定した場合 , 最大加速度まで到達しなくなり , 減速時間が計算式と異なります .

このため $(RFH-RFL)/2$ 以下の値を設定して下さい .

(11) RFA : 補助速度レジスタ (17ビット)

原点復帰動作での逆転定速度 , バックラッシュ補正時の定速度を設定するレジスタです .

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

図 5 . 2 - 1 5 RFA : 補助速度レジスタのビット構成

原点復帰動作での逆転定速度として使用します .

または , バックラッシュ時の移動量補正速度 (F A 速度) の速度を 1 ~ 100,000 (186A0h) の範囲で設定します .

100,000 ~ 131,071 (186A0h ~ 1FFFFh) の範囲は全て 100,000 と見なされます .

実際の動作速度は R M G (速度倍率設定レジスタ) の設定値との計算値になります .

(12) RIRQ: イベント要因設定レジスタ (13ビット)

イベント要因を設定します。イベントを発生させたい内容に対応するビットを"1"にします。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	IRNP	IRNA	IRSA	IRSD	IROL	IRLT	IRCL	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IREN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図5.2-16 RIRQ: イベント要因設定レジスタのビット構成

bit	記号	内 容
0	I REN	正常停止時
1	I RUS	加速開始時
2	I RUE	加速終了時
3	I RDS	減速開始時
4	I RDE	減速終了時
5	I RC 1	コンパレータ1条件成立時
6	I RC 2	コンパレータ2条件成立時
7	I RC 3	コンパレータ3条件成立時
8	I RCL	CLR信号入力によるカウント値のリセット時
9	I RLT	LTC H入力によるカウント値のラッチ時
10	I ROL	OLS入力によるカウント値のラッチ時
11	I RSD	DLS入力ON時
12	I RSA	STA入力ON時
13	I RNA	グループスタート時
14	I RNP	グループストップ時
31~15	未定義	(常に0を設定して下さい。)

表5.2-12 RIRQ: イベント要因設定レジスタの内容

(13) RIST: イベントステータスレジスタ (13ビット)

イベント発生要因を確認できます。(読み出し専用)

イベントが発生した時に、対応するビットが"1"になります。

このレジスタは、読み出しによりリセットされます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ISNP	ISNA	ISSA	ISSD	ISOL	ISLT	ISCL	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISEN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図5.2-17 RIST: イベントステータスレジスタのビット構成

bit	記号	内 容
0	I SEN	正常停止時
1	I SUS	加速開始時
2	I SUE	加速終了時
3	I SDS	減速開始時
4	I SDE	減速終了時
5	I SC 1	コンパレータ1条件成立時
6	I SC 2	コンパレータ2条件成立時
7	I SC 3	コンパレータ3条件成立時
8	I SCL	CLR信号入力によるカウント値のリセット時
9	I SLT	LTC H入力によるカウント値のラッチ時
10	I SOL	OLS入力によるカウント値のラッチ時
11	I SSD	DLS入力ON時
12	I SSA	STA入力ON時
13	I SNA	グループスタート時
14	I SNP	グループストップ時
31~15	未定義	(常に0になります。)

表5.2-13 RIST: イベントステータスレジスタの内容

(14) REST: エラーステータスレジスタ (15ビット)

エラー要因を確認できます。(読み出し専用)

エラーが発生した時に対応するビットが"1"になります。このレジスタは読み出しによりリセットされます。

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

0	ESPE	ESEE	ESOR	0	ESNT	ESPO	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC3	ESC2	ESC1
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図5.2-18 R E S T : エラーステータスレジスタのビット構成

bit	記 号	内 容	
0	E S C 1	コンパレータ1条件成立による停止時 (+ S L S)	
1	E S C 2	コンパレータ2条件成立による停止時 (- S L S)	
2	E S C 3	コンパレータ3条件成立による停止時	
3	E S P L	+ E L S入力ONによる停止時	
4	E S M L	- E L S入力ONによる停止時	
5	E S A L	S V A L M入力ONによる停止時	
6	E S S P	S T P入力による停止時	
7	E S E M	(予約 : 不定)	
8	E S S D	D L S入力ONによる減速停止時	
9	E S P O	手動パルスバッファカウンタのオーバーフロー発生時	
1 0	E S N T	通信エラー発生による停止時	
1 1	未定義	(常に0になります .)	
1 2	E S O R	位置のオーバーライドが実行できなかった時	
1 3	E S E E	エンコーダ入力エラー発生時	注意 : 停止しない
1 4	E S P E	パルス入力エラー発生時	
31 ~ 15	未定義	(常に0になります .)	

表5.2-14 R E S T : エラーステータスレジスタの内容

(15) RSTS : 拡張ステータスレジスタ (28ビット)

拡張ステータスを確認できます。(読み出し専用)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERC	SPCS	SEMG	SSTP	SSTA	SDIN	SSD	SORG	SMEL	SPEL	SALM	SDIR	CND3	CND2	CND1	CND0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	SPH4	SPH3	SPH2	SPH1	SPLS	SCP3	SCP2	SCP1	SINP	SLTC	SCLR	SEZ

図 6. 2 - 19 RSTS : 拡張ステータスレジスタのビット構成

bit	記 号	内 容	
3 ~ 0	CND3 ~ 0	動作状態を表します .	
		0 0 0 0 : 停止中	1 0 0 0 : 加速中
		0 0 0 1 : STA入力待ち	1 0 0 1 : FH定速動作中
		0 0 1 0 : SVCTRCLタイマ完了待ち	1 0 1 0 : 減速中
		0 0 1 1 : 方向変化タイマ完了待ち	1 0 1 1 : INPOS待ち
		0 1 0 0 : バックラッシュ補正中	1 1 1 1 : その他(スタート制御中)
		0 1 0 1 : 手動パ ー入力待ち	1 1 0 0 : 未定義
		0 1 1 0 : F A定速動作中	1 1 0 1 : "
0 1 1 1 : F L定速動作中	1 1 1 0 : "		
4	SDIR	動作方向 (0 : + 方向 1 : - 方向)	
5	SALM	ALM入力信号がON状態の時に1になります .	
6	SPEL	+EL入力信号がON状態の時に1になります .	
7	SMEL	-EL入力信号がON状態の時に1になります .	
8	SORG	ORG入力信号がON状態の時に1になります .	
9	SSD	DLS入力信号がON状態の時に1になります . DLS入力極性 (RENV1ビット6) , ラッチ機能 (RENV1ビット5) が反映されます .	
10	SDIN	DLS入力端子がHighの時に1になります . (入力端子の状態)	
11	SSTA	不使用	
12	SSTP	不使用	
13	SEMG	不使用	
14	SPCS	SVRDY (PCS) 入力信号がON状態の時に1になります .	
15	SERC	SVCTRCL出力信号がON状態の時に1になります .	
16	SEZ	エンコーダZ相入力信号がON状態の時に1になります .	
17	SCLR	CLR入力信号がON状態の時に1になります .	
18	SLTC	LTCH入力信号がON状態の時に1になります .	
19	SINP	INPOS入力信号がON状態の時に1になります .	
20	SCP1	CMP1比較条件成立時に1になります .	
21	SCP2	CMP2比較条件成立時に1になります .	
22	SCP3	CMP3比較条件成立時に1になります .	
23	SPLS	パルス出力 (±) がON状態の時に1になります . 注1	
24	SPH1	不使用	
25	SPH2	不使用	
26	SPH3	不使用	
27	SPH4	不使用	
31 ~ 28	未定義	(常に0になります)	

表 5. 2 - 15 RSTS : 拡張ステータスレジスタの内容

注1. CNV/CCW 信号の論理和出力です。

(16) RCTR1 : 指令位置カウンタレジスタ (28ビット)

RCTR1 (指令位置カウンタ) です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図 5. 2 - 20 RCTR1 : 指令位置カウンタレジスタのビット構成

指令パルスのカウント専用です。

範囲は, -134,217,728 ~ +134,217,727 (符号付28ビット) です。

(17) RCTR2: 機械位置カウンタレジスタ (28ビット)

CTR2 (機械位置カウンタ) です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図5.2-21 RCTR2: 機械位置カウンタレジスタのビット構成

エンコーダ信号(EA/EB入力), パルサ信号(PA/PB入力), 指令パルスがカウントできます。

範囲は, -134,217,728 ~ +134,217,727 (符号付28ビット) です。

(18) RCTR3: 汎用・偏差カウンタレジスタ (16ビット)

CTR3 (汎用・偏差カウンタ) です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図5.2-22 RCTR3: 汎用・偏差カウンタレジスタのビット構成

指令パルスとエンコーダ信号/パルサ信号の偏差をカウントできます。

汎用カウンタとしても使用できます。

範囲は, -32,768 ~ +32,767 です。

範囲以上のカウントはせず, 最大値を示します。

(19) RCMP1~3: コンパレータ1~3レジスタ (28ビット)

コンパレータ1~3用の比較データを設定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図5.2-23 RCMP1~3: コンパレータ1~3レジスタのビット構成

設定範囲は, -134,217,728 ~ +134,217,727 (符号付28ビット) です。

(20) RLTC1: カウンタ1ラッチレジスタ (28ビット)

CTR1 (指令位置)のラッチデータです。(読み出し専用)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図5.2-24 RLTC1: カウンタ1ラッチレジスタのビット構成

LTCH, OLS入力, LTCHコマンドによりCTR1の内容がコピーされます。

データ範囲は, -134,217,728 ~ +134,217,727 (符号付28ビット) です。

(21) RLTC2: カウンタ2ラッチレジスタ (28ビット)

CTR2 (機械位置)のラッチデータです。(読み出し専用)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

図5.2-25 RLTC2: カウンタ2ラッチレジスタのビット構成

LTCH, OLS入力, LTCHコマンドによりCTR2の内容がコピーされます。

データ範囲は, -134,217,728 ~ +134,217,727 (符号付28ビット) です。

(22) RLTC3: カウンタ3ラッチレジスタ(17ビット)

CTR3 (偏差)または現在速度のラッチデータです。(読み出し専用)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	%																

図5.2-26 RLTC3: カウンタ3ラッチレジスタのビット構成

L T C H, O L S 入力, L T C H コマンドにより C T R 3 または現在指令速度の内容がコピーされます。

R E N V 4 レジスタの L T F D=0 の時は C T R 3 を, L T F D=1 の時は現在指令速度をラッチします。

なお, L T F D=1 で 停止状態の時は, ラッチデータは 0 になります。

L T F D=0 の時のデータ範囲は, -32,768 ~ +32,767 となり, L T F D=1 の時は 0 ~ 100,000 です。

C T R 3 データラッチ時(RENV4 の L T F D(b26)=0)は, \$ 及び%表示のビットは, ビット 15 と同一状態(符号拡張)になります。

現在速度データラッチ時(RENV4 の L T F D(b26)=1)は, \$ 表示のビットは“ 0 ”になり, %を含む下位 17 ビットは現在速度データになります。

(23) RPLS: 移動残パルスレジスタ(28ビット)

位置決めカウンタの値(移動残パルス数)を確認できます。(読み出し専用)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0																												

図5.2-27 RPLS: 移動残パルスレジスタのビット構成

スタート時に R M V レジスタの絶対値になり, パルス出力毎にダウンカウントします。

(24) RSPD: 速度モニタレジスタ(27ビット)

E Z カウント値と, 現在速度を確認できます。(読み出し専用)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AS15	AS14	AS13	AS12	AS11	AS10	AS9	AS8	AS7	AS6	AS5	AS4	AS3	AS2	AS1	AS0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	IDC2	IDC1	IDC0	ECZ3	ECZ2	ECZ1	ECZ0	0	0	0	AS16

図5.2-28 RSPD: 速度モニタレジスタのビット構成

bit	記号	内容
16~0	AS16~0	現在速度をステップ値(RFL, RFHと同一単位)として読出せます。 停止時には“ 0 ”になります。
19~17	未定義	(常に0になります。)
23~20	ECZ3~0	原点復帰に使用するEZ入力のカウンタ値を読み出せます。
26~24	IDC2~0	アイドリングカウンタ値を読み出せます。
31~27	未定義	(常に0になります。)

表5.2-16 RSPD: 速度モニタレジスタの内容

(25) RSDC: 減速開始点自動演算値レジスタ(24ビット)

位置決め動作における, 減速開始点自動演算値を確認できます。(読み出し専用)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0																								

図5.2-29 RSDC: 減速開始点自動演算値レジスタのビット構成

5.2.8 モーションデバイス環境設定レジスタ

(1) RENV1:環境設定1レジスタ(30ビット)

環境設定1用レジスタです。主に入出力端子の仕様を設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERCL	EPW2	EPW1	EPW0	EROR	EROE	ALML	ALMM	ORGL	SDL	SDLT	SDM	ELM	PMD2	PMD1	PMD0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	PDTC	SEDR	SEDM	DTMF	FLTR	PCSL	LTCL	INPL	CLR1	CLR0	STPM	STAM	ETW1	ETW0

図5.2-30 RENV1:環境設定1レジスタのビット構成

bit	記 号	内 容								
2 ~ 0	PMD 0 ~ PMD 2	信号端子名	b 2 - b 0 = 1 0 0 : 個別パルス方式					指令出力		
								CWパルス出力		
								C CWパルス出力		
		信号端子名	b 2 - b 0 = 0 1 0 : 共通パルス方式					指令出力		
								パルス列		
								方向出力		
		設 定 値							' 0 '	' 1 '
3	ELM	ELS入力ON時の処理 (注1)							即停止	減速停止
4	SDM	DLS入力ON時の処理 (注2)							減速のみ	減速停止
5	SDLT	DLS入力のラッチ機能 . (注2) DLS信号幅が短い時にONにします . スタート時にDLS入力がOFFの時は , ラッチ信号はリセットされます . また , SDLT = 0にしても , ラッチ信号はリセットされます							設定しない	設定する
6	SDL	DLS信号入力極性							B接	A接
7	ORGL	OLS信号入力極性							B接	A接
8	ALMM	SVALM入力ON時の処理							即停止	減速停止
9	ALML	SVALM信号入力極性							B接	A接
10	EROE	± ELS , SVALM入力により即停止した時に偏差カウンタクリア (SVCTRCL) 信号を自動出力 ただし , 減速停止した時には出力しません . また , RMDレジスタのMOD = "010 X000" (ELS位置まで動作) にして , ELS停止を正常停止とした場合でも即停止であれば出力します							自動出力しない	自動出力する
11	EROR	原点復帰完了時に偏差カウンタクリア (SVCTRCL) 信号を自動出力							自動出力しない	自動出力する
14 ~ 12	EPW2 ~ 0	偏差カウンタクリア出力 (SVCTRCL) 信号の出力パルス幅								
		000	001	010	011	100	101	110	111	
		12 μs	102 μs	409 μs	1.6ms	13ms	52ms	104ms	レベル出力	
15	ERCL	偏差カウンタクリア出力 (SVCTRCL) 信号出力論理設定 (0 : 負論理 1 : 正論理)								
17 ~ 16	ETW1 ~ 0	SVCTRCL信号出力後ディレー時間					00	01	10	11
							0 μs	12 μs	1.6ms	104ms
18	STAM	STA入力仕様					0 : レベルトリガ		1 : エッジトリガ	
19	STPM	STP入力停止設定時 (RMDのb15 = 1) 停止方法 .					0 : 即停止		1 : 減速停止	
21 ~ 20	CLR1 ~ 0	CLR入力仕様	00		01		10		11	
			立ち下がりエッジ		立ち上がりエッジ		Lレベル		Hレベル	
22	INPL	INPOS信号入力極性					0 : B接		1 : A接	
23	LTCL	LTC信号入力仕様					0 : 立ち下がり		1 : 立ち上がり	
24	PCSL	SVRDY (PCS) 信号入力極性					0 : B接		1 : A接	
25	FLTR	不使用 (常に0に設定して下さい .)								
26	DTMF	方向変化タイマ (0.2ms) 機能					0 : ON		1 : OFF	
27	SEDM	SEND出力禁止 . (モーションデータは変化します .)					0 : 許可		1 : 禁止	
28	SEDR	1 : スタート時にSENDがリセットされます .								
29	PDTC	不使用 (常に0に設定して下さい .)								
31 , 30	未定義	(常に0に設定して下さい .)								

表5.2-17 RENV1:環境設定1レジスタの内容

注1. ELS入力ON時の処理を減速停止(ELM=1)に設定した場合、ELS入力ONで減速を開始しますので、ELS位置を通過して停止します。機械系への衝突等に十分御注意下さい。

注2．RMD(動作モード)レジスタのMSDE = 0の場合，DLS信号は無視されます．

DLS信号入力を有効にした場合，動作中にDLS信号がONすると，SDM(bit4)，SDLT(bit5)の設定により，<(1)減速 (2)ラッチ・減速 (3)減速停止 (4)ラッチ・減速停止>します．

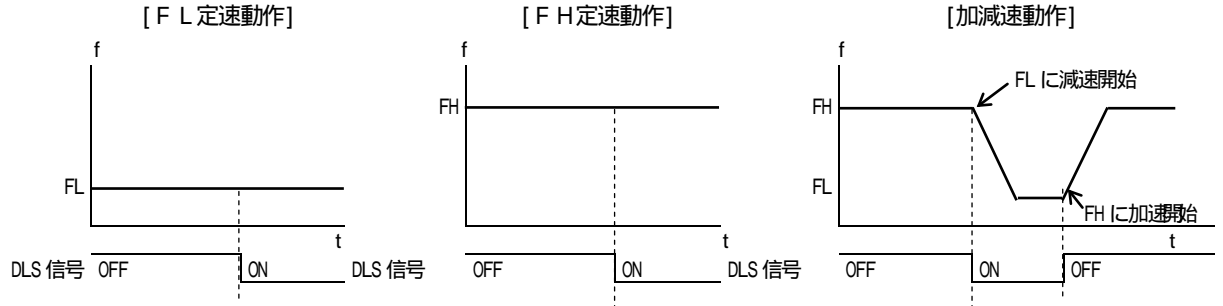
(1) 減速 <RENV1レジスタのSDM(bit4)=0，SDLT(bit5)=0>

定速動作中はDLS信号を無視します．

加減速動作は，DLS信号ONによりFL速度まで減速します．

減速後，または減速中にDLS信号がOFFになるとFHまで加速します．

加速スタートコマンド書き込み時にDLS信号がONしている場合は，FL速度で動作しますが，DLS信号がOFFになるとFHまで加速します．



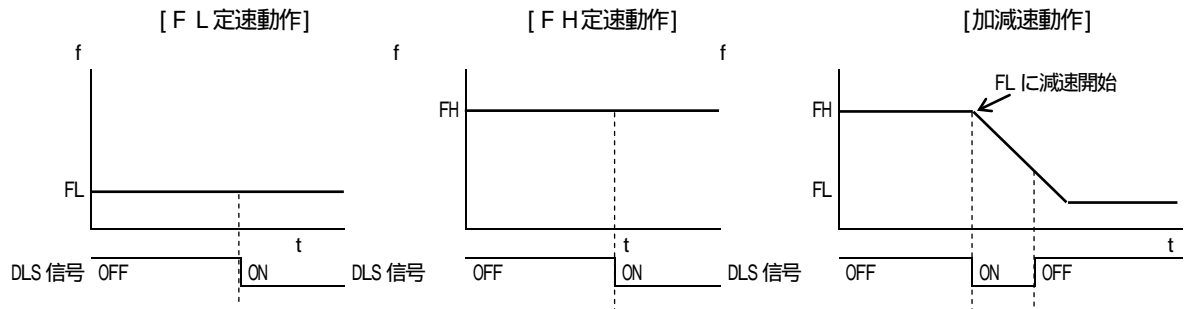
(2) ラッチ・減速 <RENV1レジスタのSDM(bit4)=0，SDLT(bit5)=1>

定速動作中はDLS信号を無視します．

加減速動作中は，DLS信号ONによりFL速度まで減速します．

減速後，または減速中にDLS信号がOFFになってもFL速度を維持し，FH速度に加速しません．

加速スタートコマンド書き込み時にDLS信号がONしている場合は，FL速度で動作しますが，DLS信号がOFFになっても，FH速度に加速はしません．



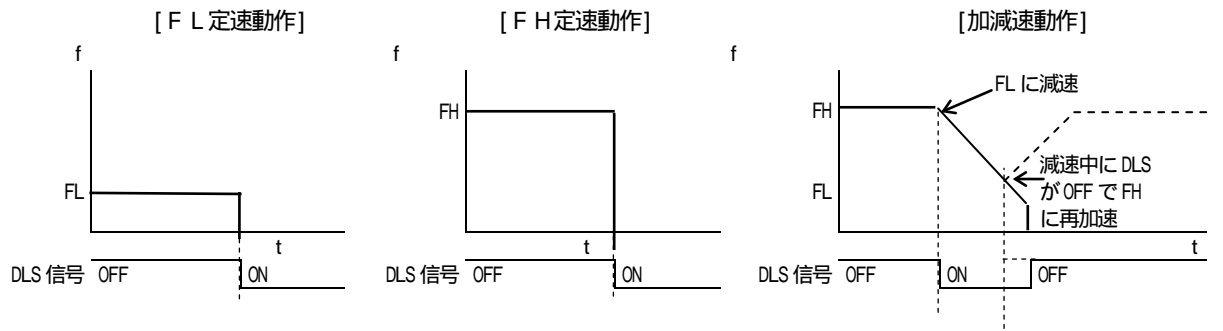
(3) 減速停止 <RENV1レジスタのSDM(bit4)=1，SDLT(bit5)=0>

定速動作中にDLS信号がONすると停止します．

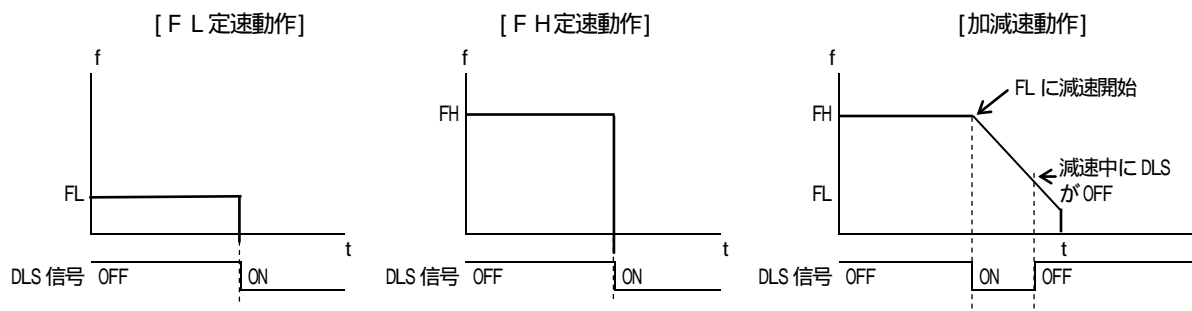
加減速動作中はDLS信号ONにより，FL速度まで減速後停止します．

減速中にDLS信号がOFFになるとFHまで加速します．

スタートコマンド書き込み時にDLS信号がONしている場合は，スタートせず動作完了になります．
停止時にイベント発生します．



- (4) ラッチ・減速停止 < RENV1レジスタの SDM(bit4)=1, SDLT(bit5)=1>
 定速動作中にDLS信号がONすると停止します。
 加減速動作中はDLS信号ONにより, FL速度まで減速後停止します。
 減速中にDLS信号がOFFになっても加速しません。
 スタートコマンド書き込み時にDLS信号がONしている場合は, スタートせず動作完了になります。
 停止時にイベント発生します。



(2) RENV2:環境設定2レジスタ(23ビット)

環境設定2用レジスタです。汎用ポート, エンコーダ入力, パルス入力の仕様設定を行います。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIM1	PIMO	PINF	EZL	EDIR	EIM1	EIMO	EINF	1	1	1	1	1	1	1	1
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	IDL2	IDL1	IDL0	0	POFF	EOFF	PDIR

図5.2-31 RENV2:環境設定2レジスタのビット構成

bit	記号	内容
7~0	POM	全て'1'を設定(ボード仕様)
8	EINF	(予約:常に0を設定して下さい。)
10~9	EIM1~0	エンコーダA/B相入力仕様 b10,9: 00: 1通倍, 01: 2通倍, 10: 4通倍, 11: UP/DOWN
11	EDIR	エンコーダ入力カウント極性 (0: A相進相, 1: A相遅相)
12	EZL	エンコーダZ相信号の入力極性 (0: 立ち上がりエッジ, 1: 立ち上がりエッジ)
13	PINF	(予約:常に0を設定して下さい。)
15~14	PIM1~0	パルスA/B相入力仕様 b10,9: 00: 1通倍, 01: 2通倍, 10: 4通倍, 11: UP/DOWN
16	PDIR	パルス入力カウント極性 (0: A相進相, 1: A相遅相)
17	EOFF	エンコーダ入力マスク
18	POFF	パルス入力マスク
19	未定義	(常に0を設定して下さい。)
22~20	IDL2~0	アイドリングパルス数設定 (0~7パルス)
31~23	未定義	(常に0を設定して下さい。)

表5.2-18 RENV2:環境設定2レジスタの内容

(3) RENV3:環境設定3レジスタ(31ビット)

環境設定3用レジスタです。主に, 原点復帰動作方法とカウンタ動作仕様を設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	BSYC	CI32	CI31	CI30	CI21	CI20	EZD3	EZD2	EZD1	EZD0	ORM3	ORM2	ORM1	ORM0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	CU3H	CU2H	CU1H	0	CU3B	CU2B	CU1B	0	CU3R	CU2R	CU1R	0	CU3C	CU2C	CU1C

図5.2-32 RENV3:環境設定3レジスタのビット構成

(次ページに続く)

(前ページからの続き)

bit	記 号	内 容			
3 ~ 0	ORM3 ~ ORM0	原点復帰方法を設定します。			
		ORG モード	ORM3 ~ 0	原点復帰内容	
		0	0000	【センサ原点復帰 (O L S on 検出) 】 定速復帰 : OLS on で即停止し完了 高速復帰 : OLS on 検出で減速停止で完了 (DLS 減速 , OLS 原点も可能)	
		1	0001	【センサ原点復帰 (O L S 検出後拔出し再突入完了) 】 定速復帰 : OLS on で即停止後 , 反転拔出し , 再度 OLS 検出で完了 高速復帰 : OLS on で減速停止後 , 反転拔出し , 再度 OLS 検出で完了	
		2	0010	【センサ+エンコーダ原点復帰 (O L S on 検出) 】 定速復帰 : OLS on 検出後の Z 相で即停止し完了 高速復帰 : OLS on で減速し , 検出後の Z 相で即停止し完了	
		3	0011	【センサ+エンコーダ原点復帰 (O L S on 検出) 】 定速復帰 : OLS on 検出後の Z 相で即停止し完了 高速復帰 : OLS on 検出後の Z 相で減速停止し完了	
		4	0100	【センサ反転+エンコーダ原点 (O L S on 反転) 】 定速復帰 : OLS on で即停止 , 反転定速の Z 相検出で完了 高速復帰 : OLS on で減速停止 , 反転定速の Z 相検出で完了	
		5	0101	【センサ高速反転+エンコーダ原点 (O L S on 反転) 】 定速復帰 : OLS on で即停止 , 反転定速の Z 相検出で完了 高速復帰 : OLS on で減速停止 , 反転高速の Z 相検出で減速停止し完了	
		6	0110	【 E L S 兼用センサ原点 】 定速復帰 : ELS on で即停止後 , 反転定速拔出し , ELS off 検出で完了 高速復帰 : ELS on で減速停止後 , 反転定速拔出し , ELS off 検出で完了	
		7	0111	【 E L S 反転低速+エンコーダ原点 】 定速復帰 : ELS on で即停止後 , 反転定速の Z 相検出で完了 高速復帰 : ELS on で減速停止後 , 反転定速の Z 相検出で完了	
		8	1000	【 E L S 反転定速+エンコーダ原点 】 定速復帰 : ELS on で即停止後 , 反転定速の Z 相検出で完了 高速復帰 : ELS on で減速停止後 , 反転高速の Z 相検出で減速停止し完了	
		以下は C T R 2 クリアを利用した原点復帰 [CTR2 (機械カウンタ) = 0] の方法			
	9 ~ 12	1001 ~ 1100			
7 ~ 4	E Z D 3 ~ 0	原点復帰で使用する E Z カウント値を設定します。0 0 0 0 (1 回目) ~ 1 1 1 1 (16 回目)			
9 ~ 8	C I 21 ~ 20	CTR2 (機械位置) のカウント入力選択	b9,8	0 0	エンコーダ (パルス) 入力
				0 1	指令パルス
				1 0	エンコーダ (パルス) 入力
12 ~ 10	C I 32 ~ 30	CTR3 (汎用・偏差) のカウント入力選択	b12-10	0 0 0	出力パルス
				0 0 1	エンコーダ (パルス) 入力
				0 1 0	
				0 1 1	内部基準クロック (40MHz) の 1 / 4096 分周クロック
				1 0 0	出力パルスとエンコーダ (パルス) 入力
				1 0 1	(偏差カウント)
13	B S Y C	1 : 動作中 (B S Y = L) の間だけ , C T R 3 を動作させます。			
15, 14	未定義	(常に 0 を設定して下さい。)			
16	C U 1 C	C L R 入力 O F F O N 時	CTR 1 (指令位置)	を 0 : リセットしない 1 : リセットする	
17	C U 2 C		CTR 2 (機械位置)		
18	C U 3 C		CTR 3 (汎用・偏差)		
19	未定義	(常に 0 を設定して下さい。)			
20	C U 1 R	原点復帰動作完了時	CTR 1 (指令位置)	を 0 : リセットしない 1 : リセットする	
21	C U 2 R		CTR 2 (機械位置)		
22	C U 3 R		CTR 3 (汎用・偏差)		
23	未定義	(常に 0 を設定して下さい。)			
24	C U 1 B	バックラッシュ補正時	CTR 1 (指令位置)	を 0 : カウンタを動作させない 1 : カウンタを動作させる	
25	C U 2 B		CTR 2 (機械位置)		
26	C U 3 B		CTR 3 (汎用・偏差)		
27	未定義	(常に 0 を設定して下さい。)			
28	C U 1 H	CTR 1 (指令位置)	0 : カウンタを停止しない 1 : カウンタを停止する		
29	C U 2 H	CTR 2 (機械位置)			
30	C U 3 H	CTR 3 (汎用・偏差)			
31	未定義	(常に 0 を設定して下さい。)			

表 5. 2 - 19 RENV3: 環境設定 3 レジスタの内容

(4) RENV4：環境設定4レジスタ(28ビット)

環境設定4用レジスタです。コンパレータ1～4の設定を行います。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	C2D1	C2D0	C2S2	C2S1	C2S0	C2C1	C2C0	0	C1D1	C1D0	C1S2	C1S1	C1S0	C1C1	C1C0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	LTOF	LTFD	LTM1	LTM0	C3D1	C3D0	C3S3	C3S2	C3S1	C3S0	C3C1	C3C0

図5.2-33 RENV4：環境設定4レジスタのビット構成

bit	記号	内 容			
1,0	C1C1~0	コンパレータ1 比較カウンタ(注1)	b1,0 = 00 CTR1(指令位置)	b1,0 = 01 CTR2(機械位置)	b1,0 = 10 CTR3(汎用・偏差)
4~2	C1S2~0	コンパレータ1 比較方法(注2)	b4-2 = 000 常に比較条件 不成立	b4-2 = 001 RCMP1データ = 比較CTR 加算方向無関係	b4-2 = 010 カウントアップ中
			b4-2 = 100 RCMP1 > 比較CTR	b4-2 = 101 RCMP1 < 比較CTR	b4-2 = 110 + SLSとして (RCMP1 < CTR1)
					b4-2 = 111 常に比較条件 不成立
6,5	C1D1~0	コンパレータ1条件成立時処理選択	b6,5 = 00 処理なし	b6,5 = 01 即停止	b6,5 = 10 減速停止
7	未定義	(常に0を設定して下さい。)			
9,8	C2C1~0	コンパレータ2 比較カウンタ(注2)	b9,8 = 00 CTR1(指令位置)	b9,8 = 01 CTR2(機械位置)	b9,8 = 10 CTR3(汎用・偏差)
12~10	C2S2~0	コンパレータ2 比較方法(注2)	b12-10 = 000 常に比較条件 不成立	b12-10 = 001 RCMP2データ = 比較CTR 加算方向無関係	b12-10 = 010 カウントアップ中
			b12-10 = 100 RCMP2 > 比較CTR	b12-10 = 101 RCMP2 < 比較CTR	b12-10 = 110 - SLSとして (RCMP2 > CTR1)
					b12-10 = 111 常に比較条件 不成立
14,13	C2D1~0	コンパレータ2条件成立時処理選択	b14,13 = 00 処理なし	b14,13 = 01 即停止	b14,13 = 10 減速停止
15	未定義	(常に0を設定して下さい。)			
17,16	C3C1~0	コンパレータ3 比較カウンタ(注3)	b17,16 = 00 CTR1(指令位置)	b17,16 = 01 CTR2(機械位置)	b17,16 = 10 CTR3(汎用・偏差)
21~18	C3S3~0	コンパレータ3 比較方法(注3)	b21-18 = 0001 RCMP3データ = 比較CTR 加算方向無関係	b21-18 = 0010 カウントアップ中	b21-18 = 0011 カウントダウン中
			b21-18 = 0100 RCMP3 > 比較CTR	b21-18 = 0101 RCMP3 < 比較CTR	b21-18 = 0111 設定禁止
			b21-18 = 1000 定ピッチ信号出力として使用(注3)	b21-18 = 1001	b21-18 = 1010
23,22	C3D1~0	コンパレータ3条件成立時処理選択	b23,22 = 00 処理なし	b23,22 = 01 即停止	b23,22 = 10 減速停止
25,24	LTM1~0	カウンタ(CTR1~3)の ラッチタイミングを設定	b25,24 = 00 LTCH OFF ON	b25,24 = 01 OLS 入力	b25,24 = 10 CMP2 条件成立
26	LTFD	1: CTR3の代わりに現在速度データをラッチ			
27	LTOF	1: ハードウェアラッチ停止(ソフトラッチのみ有効)			
31~28	未定義	(常に0を設定して下さい。)			

表5.2-20 RENV4：環境設定4レジスタの内容

注1. 比較カウンタとして CTR3 (偏差に設定時) を選択した場合には、カウンタ値の絶対値と、コンパレータデータとの比較となります。(絶対値範囲: 0 ~ 32,767)

注2. C1S2~0 = 110 (+ソフトリミット)、または C2S2~0 = 110 (-ソフトリミット) に設定する場合には、比較カウンタに CTR1 (指令位置) を選択して下さい。また、ソフトリミット設定時は、C1D1~0 及び C2D1~0 の設定にかかわらずに停止します。

注3. C3S3~0 = 1000 ~ 1010 (定ピッチ信号出力) にする場合には、比較カウンタに CTR3 (汎用に設定) を選択して下さい。他のカウンタは選択できません。また、コンパレータの設定値は正の値にして下さい。

(5) RENV5:環境設定5レジスタ(32ビット)

環境設定5用レジスタです。主に移動量の補正データを設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSTP	0	0	ADJ	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMG4	PMG3	PMG2	PMG1	PMG0	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

図5.2-34 RENV5:環境設定5レジスタのビット構成

bit	記号	内容
11~0	BR11~0	バックラッシュ補正量設定 [設定範囲: 0~4,095]
12	ADJ	移動量補正方法選択。(0:補正機能OFF, 1:バックラッシュ補正)
14,13	未定義	(常に0を設定して下さい。)
15	PSTP	(常に0を設定して下さい。)
26~16	PD10~0	パルス入力の変換比設定 (設定値)/2048 に分周 [設定範囲: 0~2,047] 0を設定した場合には, 分周回路はOFF (=2048/2048)
31~27	PMG4~0	パルス入力の変換比設定 (設定値+1)倍に逡倍 [設定範囲: 0~31]

表5.2-21 RENV5:環境設定5レジスタの内容

(6) RENV6:環境設定6レジスタ(32ビット)

環境設定6用レジスタです。振動抑制機能用の時間設定を行います。

RT, FTデータが共に0以外の時に振動抑制機能がONになります。

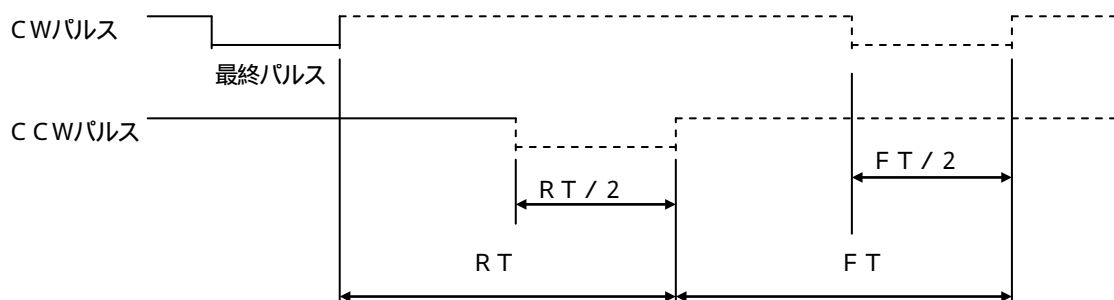
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0

図5.2-35 RENV6:環境設定6レジスタのビット構成

bit	記号	内容
15~0	RT15~0	下図RTの時間を設定します。[設定範囲: 0~65,535] 設定単位は $25\text{nsec} \times 64 (1.6\mu\text{s})$ です。
31~16	FT15~0	下図FTの時間を設定します。[設定範囲: 0~65,535] 設定単位は $25\text{nsec} \times 64 (1.6\mu\text{s})$ です。

表5.2-22 RENV6:環境設定6レジスタの内容

下図の破線部分が振動抑制機能により付加されるパルスです。



設定時間 [RT, FT] = 設定値 $\times 1.6 (\mu\text{s})$

図5.2-36 振動抑制用逆転パルス

6．ソフトウェア編

6．1 ソフトウェアの構成

6．1．1 Windows版のソフトウェア構成

弊社の提供するソフトウェアは、ライブラリ関数、ドライバ関数、デバイスドライバの3種類です。

デバイスドライバはmotionCATマスターボードHPC104-MCAT110Mへの入出力を行うソフトウェアです。

ドライバ関数はアプリケーションとデバイスドライバをつなぐ入出力関数「デバイスドライバI/F用ライブラリ」であり、Win32API関数としてDLLファイルで提供されています。

ライブラリ関数はドライバ関数で構成され、モーションモジュールHMG-Pxの初期化、原点復帰、位置決め動作等の基本的な動作を制御することができます。ライブラリ関数はソースファイルで提供されていますので、適宜変更追加することができます。また各開発言語用ライブラリ関数の仕様は同様になっています。

(1) Windows版デバイスドライバ

WindowsVista・XP/Windows2000用・・・hm110wdm.sys

(2) Windows版デバイスドライバI/F用DLL

デバイスドライバI/F用DLLに含まれる各種関数を「ドライバ関数」と称します。

WindowsVista・XP/Windows2000用・・・hmct110.dll

(3) Windows版モーションモジュール用ライブラリ関数

アプリケーションプログラム用の「ドライバ関数を使用した特定機能処理を行うライブラリ関数」であり、ソースプログラムで提供されます。このライブラリ関数の内容は自由に変更できます。

ドライバ関数と対比して「ライブラリ関数」と称します。

VC++用ライブラリ関数ソースファイル・・・h1px11a.c

VC++用ライブラリ関数ヘッダーファイル・・・h1px11a.h

VB6.0用ライブラリ関数ソースファイル・・・h1px11a.bas

VB.NET用ライブラリ関数ソースファイル・・・h1px11a.vb

6.1.2 ソフトウェアの構成

アプリケーションプログラムとこれらのソフトウェアの関連は下図の通りです。

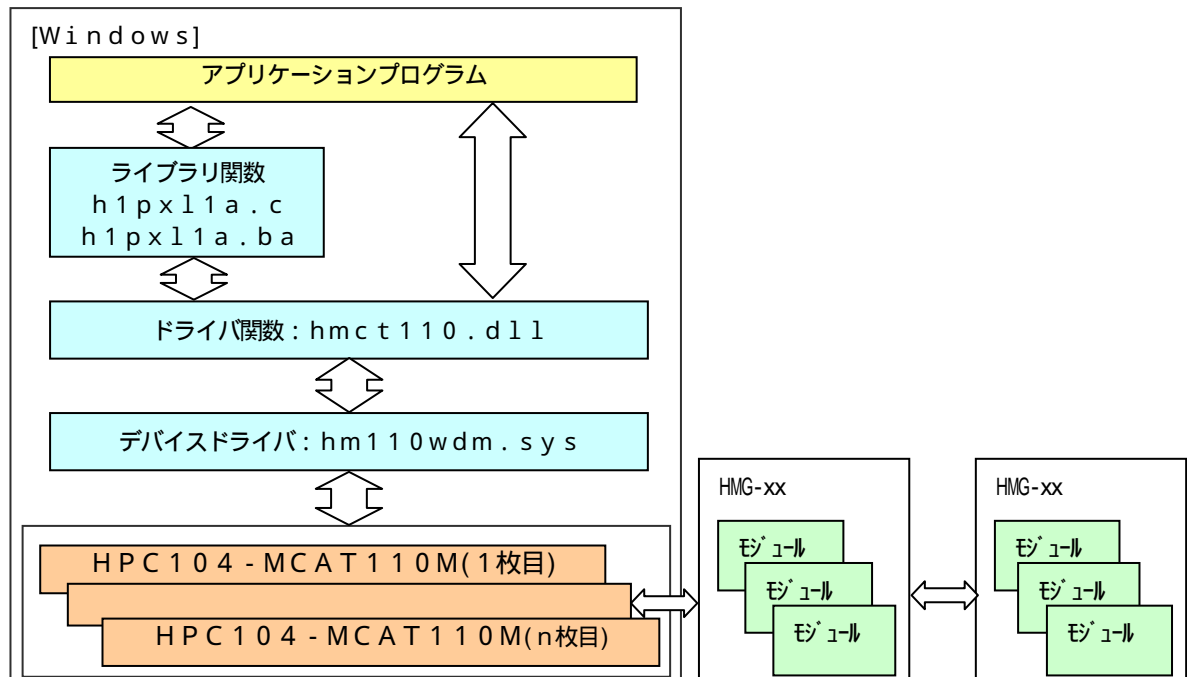


図6.1-1 ソフトウェアの関連図

6.1.3 ソフトウェアの使用方法

「hmct.dll」をカレントフォルダにコピーして使用します。

6.2 Windows用デバイスドライバのインストールとアンインストール

6.2.1 Windows用デバイスドライバのインストール

- (1) 添付ディスクをCDドライブに挿入します。
- (2) エクスプローラを起動し、CDドライブ: ¥x86¥install.bat を起動します。
- (3) 以下の画面が出力されますので、プログラムの指示に従い、アドレス空間の選択、アクセスモードの選択、アドレスの設定を行って下さい。バス番号には通常“0”を設定して下さい。

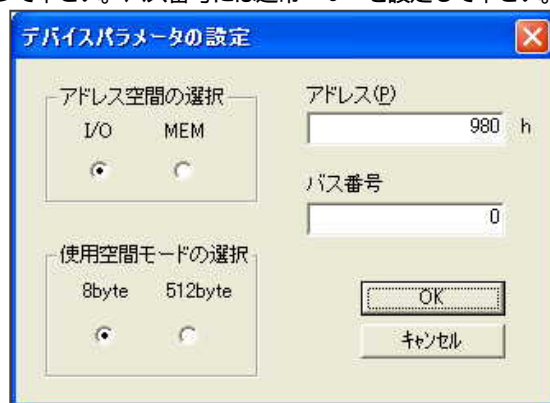


図6.2-1 デバイスパラメータの設定画面

- (4) プログラムの指示に従い、システムを再起動させて完了です。

6.2.2 Windows用デバイスドライバのアンインストール

- (1) 添付ディスクをCDドライブに挿入します。
- (2) エクスプローラを起動し、CDドライブ: ¥mc110uin.exe を実行します。

6.3 アプリケーション構築準備

(1) Visual C++ (6.0以上)によるアプリケーションの構築

次のファイルをプロジェクトへ追加します。

プロジェクト追加ファイル

ライブラリ関数用・・・h1pxl1a.c・・・VC++アプリケーション用ライブラリ関数ソースファイル

ライブラリ関数のコードがC言語で記述されています。

ドライバ関数用・・・hmct110.lib・・・ドライバ関数インポートライブラリ

インクルードファイル

ライブラリ関数用・・・h1pxl1a.h・・・VC++アプリケーション用ライブラリ関数ヘッダーファイル

“hpxl1a.c”ファイルの中で“#include”されています。

(ライブラリ関数を使用せずにアプリケーションの構築も可能です)

ドライバ関数用・・・hmct110.h・・・ドライバ関数結合用ヘッダーファイル

“hpxl1a.h”ファイルの中で“#include”されています。

(2) Visual Basic (6.0)によるアプリケーションの構築

次のファイルをプロジェクトへ追加します。

ライブラリ関数用・・・h1pxl1a.bas・・・VB6.0アプリケーション用ライブラリ標準モジュールファイル

このファイルにライブラリ関数の関数定義と関数コードが記述されています。

(ライブラリ関数を使用せずにアプリケーションの構築も可能です)

ドライバ関数用・・・hmct110.bas・・・ドライバI/F用DLL関数定義標準モジュールファイル

このファイルに外部関数宣言(Declare)、及びユーザー定義型宣言が記述されています。

(3) Visual Basic.NETによるアプリケーションの構築

次のファイルをプロジェクトへ追加します。

ライブラリ関数用・・・h1pxl1a.vb・・・VB.NETアプリケーション用ライブラリソースファイル

このファイルにライブラリ関数の関数定義と関数コードが記述されています。

(ライブラリ関数を使用せずにアプリケーションの構築も可能です)

ドライバ関数用・・・hmct110.vb・・・ドライバI/F用DLL関数定義ソースファイル

このファイルに外部関数宣言(Declare)、及びユーザー定義型宣言が記述されています。

6.4 ボードアクセス方法

ドライバ関数・ライブラリ関数群では複数のマスターデバイスを制御することができます。

ある1つのマスターデバイスにアクセスするためには、まずこのデバイスをオープンし、制御するデバイスハンドルを取得する必要があります。

デバイスをオープンするためには、どのようなハードウェアリソースを持つデバイスをオープンするのかという情報が必要となります。この情報をデバイス情報と呼びます。

6.4.1 デバイス情報構造体

ボード認識のために次に示す HMCT110INFO 型構造体を、ボード枚数最大4枚として、使用枚数分用意します。

[C 言語: Visual C++]

```
typedef struct _HMCT110INFO {    // デバイス情報
    DWORD    dwBaseAddress;    // ベースアドレス
    DWORD    dwIrqNo;          // IRQ 番号
    DWORD    dwNumber;         // 管理番号
    DWORD    dwAddressSpace;   // アドレス空間 ( 0:IO / 1:MEM )
    DWORD    dwAccessMode;     // 使用空間モード (0:8バイト / 1:512バイト)
    DWORD    dwRsrvd;          // 予約
} HMCT110INFO, * PHMCT110INFO;
```

[Visual Basic]

```
Public Type HMCT110INFO
    dwIoAdr      As Long      ' ベースアドレス
    dwIrqNo      As Long      ' IRQ 番号
    dwNumber     As Long      ' 管理番号
    dwAddressSpace As Long     ' アドレス空間 ( 0:IO / 1:MEM )
    dwAccessMode As Long      ' 使用空間モード (0:8バイト / 1:512バイト)
    dwRsrvd      As Long      ' 予約
End Type
```

[Visual Basic .NET]

```
Public Structure HMCT110INFO
    Dim dwIoAdr      As Integer ' ベースアドレス
    Dim dwIrqNo      As Integer ' IRQ 番号
    Dim dwNumber     As Integer ' 管理番号
    Dim dwAddressSpace As Integer ' アドレス空間 ( 0:IO / 1:MEM )
    Dim dwAccessMode As Integer ' 使用空間モード (0:8バイト / 1:512バイト)
    Dim dwRsrvd      As Integer ' 予約
End Structure
```

6.4.2 ボードアクセス準備手順と終了処理

準備手順

(1) ボード毎にデバイスオープン

まずMCATの枚数取得し、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デバイス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。
マスターデバイス枚数が2枚以上の場合には、個々のMCAT毎にこの処理を行います。

```
mct110_GetMstBrdCount()    ... MCATの枚数取得
mct110_GetMstBrdInfo()    ... MCATのデバイス情報取得
mct110_OpenMstBrd()       ... MCATのオープン処理
```

(2) 初期設定

マスターデバイス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。

```
mct110_rLclInfo()          ... ローカルデバイス(DIO, モーション)の情報取得
mct110_wCenCmd()           ... センターデバイスへのコマンド書込み
mct110_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
mct110_wIoPortB()          ... DIOデバイスのポートへの書込み
mct110_wIoPortW()          ... DIOデバイスのポートへの書込み
mct110_wLclSetInt()        ... ローカルデバイスのポート入力変化割込設定
mct110_wPclPort()          ... モーションデバイスのポートへの書込み
mct110_wLclSetInt()        ... モーションデバイスのポート入力変化割込設定
mct110_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。

```
mct110_rIoPortB()          ... DIOデバイスのポートからの読出し
mct110_wIoPortB()          ... DIOデバイスのポートへの書込み
mct110_rIoPortW()          ... DIOデバイスのポートからの読出し
mct110_wIoPortW()          ... DIOデバイスのポートへの書込み
mct110_rLclInt()           ... ローカルデバイスのポート入力変化割込読出し
mct110_wLclInt()           ... ローカルデバイスのポート入力変化割込リセット
mct110_rPclPort()          ... モーションデバイスのポートからの読出し
mct110_wPclPort()          ... モーションデバイスのポートへの書込み
mct110_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
mct110_wPclCmd()           ... モーションデバイスへのコマンド書込み
mct110_rPclReg()           ... モーションデバイスのレジスタからの読出し
mct110_wPclReg()           ... モーションデバイスのレジスタへの書込み
```

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。

マスターデバイス枚数が2枚以上の場合には、個々のマスターデバイス毎にこの処理を行います。

```
mct110_CloseMstBrd()      ... デバイスのクローズ処理
```

6.5 ドライバ関数

ドライバ関数はアプリケーションとデバイスドライバをつなぐ入出力関数「デバイスドライバI/F用ライブラリ」であり、Win32 API関数としてDLLファイルで提供されています。

6.5.1 ドライバ関数一覧

No	関 数 名	機 能	備 考
1	mct110_GetMstBrdCount()	M C A Tの枚数取得	M C A T 1 1 0
2	mct110_GetMstBrdInfo()	M C A Tのデバイス情報取得	
3	mct110_OpenMstBrd()	M C A Tのオープン	
4	mct110_CloseMstBrd()	M C A Tのクローズ	
5	mct110_rCenMsts()	センターメインステータス読出し	センターデバイスの制御関数
6	mct110_wCenCmd()	センターデバイスコマンド書込み	
7	mct110_rCenIsts()	センター割込ステータス読出し	
8	mct110_rCenBuf()	センターデバイス入出力バッファ読出し	
	mct110_wCenBuf()	センターデバイス入出力バッファ書込み	
9	mct110_rCenRFiFo()	センターデバイス受信用F I F O読出し	
	mct110_wCenSFiFo()	センターデバイス送信用F I F O書込み	
10	mct110_rLclCycErr()	サイクリック通信エラーフラグ読出し	サイクリック通信
	mct110_wLclCycErr()	サイクリック通信エラーフラグリセット	
11	mct110_rLclInfo()	ローカルデバイス(D I O , モーション)情報読出し	システム通信
	mct110_wLclInfo()	ローカルデバイス(D I O , モーション)情報書込み	
12	mct110_rLclSetInt()	ローカルデバイス入力ポート変化フラグ 設定状態読出し	サイクリック通信
	mct110_wLclSetInt()	ローカルデバイス入力ポート変化フラグ 設定書込み	
13	mct110_rLclInt()	ローカルデバイス入力ポート変化フラグ 読出し	
	mct110_wLclInt()	ローカルデバイス入力ポート変化フラグ リセット	
14	mct110_rIoPortB()	D I Oデバイス指定ポート1バイト読出し	
	mct110_wIoPortB()	D I Oデバイス指定ポート1バイト書込み	
	mct110_rIoPortW()	D I Oデバイス指定ポート2バイト読出し	
	mct110_wIoPortW()	D I Oデバイス指定ポート2バイト書込み	
15	mct110_rPclPort()	モーションデバイス汎用出力ポート出力状態読出し	データ通信
	mct110_wPclPort()	モーションデバイス汎用出力ポート出力設定書込み	
16	mct110_rPclMsts()	モーションメインステータス読出し	
17	mct110_wPclCmd()	モーションデバイス制御コマンド書込	
18	mct110_rPclReg()	モーションデバイスレジスタ読出し	M C A T 1 1 0
	mct110_wPclReg()	モーションデバイスレジスタ書込	
19	mct110_rOptPortB()	オプションポート1バイト読出し	
	mct110_rOptPortW()	オプションポート2バイト読出し	
	mct110_wOptPortB()	オプションポート1バイト書込み	
	mct110_wOptPortW()	オプションポート2バイト書込み	
20	mct110_rCenPortW()	センターデバイス指定ポート2バイト読出し	
	mct110_wCenPortW()	センターデバイス指定ポート2バイト書込み	

表 6.5 - 1 ドライバ関数一覧

6.5.2 ドライバ関数の戻り値一覧

ドライバの諸関数を使用する時、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行います。なお、装置に起因する異常はこの異常報告に含まれません。個々の要因毎に、異常発生内容を明確にすると共に、適切な処置が求められます。

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	デバイスドライバが存在しない デバイスドライバがインストールされていない デバイスドライバが所定のフォルダに格納されていない
3	ALREADY_OPENED	00000002	既にオープン済のデバイスをオープン オープン済みデバイスに更にオープン指令 オープンしたデバイスはクローズするまで使用 (多重のオープン禁止) ボード2枚以上を使用する場合、オープンするデバイス情報の更新を確認します。
4	INSUFFICIENT_MEMORY	00000004	デバイス情報格納メモリが不足 アプリケーション用のメモリ不足 パソコン主記憶メモリの不足 システムリソース(OS用メモリ)の不足 多数のアプリケーション起動 1度に多数のウィンドウを開いた
5	INVALID_HANDLE	00000008	無効なデバイスハンドルを指定 デバイスオープンで得られた"デバイスハンドル"の不使用 このデバイスは既にクローズされている
6	NOT_READY	00000010	デバイスの入出力ポートが使用できない デバイス(ボード)内部の入出力ポートがない (システム不整合が考えられます。弊社までご連絡下さい)
7	ILLEGAL_DEVICE	00000020	ボード固有情報が不正 デバイス情報は正常ですが、ボード機能が不一致です (システム不整合が考えられます。弊社までご連絡下さい)
8	ILLEGAL_ADDRESS	00000040	不正なアドレス
9	ILLEGAL_IRQNUM	00000080	不正な IRQ 番号
10	ILLEGAL_PARAM	00000100	関数の引数の値が異常 その他引数の設定値を確認
11	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
12	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
13	DATA_COM_ERROR	00040000	データ通信エラー
14	COM_OTHER_ERROR	00080000	その他の通信エラー センター割込ステータスで詳細を確認して下さい
15	OTHER_ERROR	80000000	その他のエラー

表 6.5-2 ドライバ関数の戻り値

6.5.3 ドライバ関数仕様

ここではVC++表記で説明します。VB、VB.NETを使用される場合、データ型の対応は下表の通りです。

引数のデータ型、16進数表記の対応

言語	VC++言語	VB	VB.NET
データ型	8bit	BYTE	Byte
	16bit	WORD	Integer
	32bit	DWORD	Long
データ例	0x0000	&H0	&H0
	0x1000	&H1000	&H1000

(1) mct110_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。	
Win VC++	書式	DWORD WINAPI mct110_GetMstBrdCount (DWORD* count);
	引数	DWORD* 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「7.2 ドライバ関数の戻り値」参照
	呼出例	DWORD count; // MNTの枚数 DWORD ret; // 関数の戻り値 ret = mct110_GetMstBrdCount(&count);

(2) mct110_GetMstBrdInfo () MCATのデバイス情報の取得		
機能	現在システムに装着されているMCATのデバイス情報を取得します。	
Win VC++	書式	DWORD WINAPI mct110_GetMstBrdInfo (DWORD* count, HMCT110INFO* hInfo);
	引数	DWORD* count・・・取得したMCAT枚数の格納アドレス HMCT110INFO* hInfo・・・取得したMCATのデバイス情報の先頭アドレス
	戻り値	'0'で正常, '0'以外は「7.2 ドライバ関数の戻り値」参照
	呼出例	MCAT110が2個PCに装着されているとします。 DWORD count=2; // MCATは2枚 DWORD ret; // 関数の戻り値 HMCT110INFO hMctInfo[2]; // デバイス情報(2枚分) ret = mct110_GetMstBrdInfo(&count, &hMctInfo[0]);

(3) mct110_OpenMstBrd () MCATのオープン		
機能	MCATをオープンし、このMCATのデバイスハンドルを取得します。	
Win VC++	書式	DWORD WINAPI mct110_OpenMstBrd (DWORD* hDev, HMCT110INFO* hInfo);
	引数	DWORD* hDev・・・取得したデバイスハンドルの格納アドレス HMCT110INFO* hInfo・・・オープンするMCATデバイス情報の先頭アドレス
	戻り値	'0'で正常, '0'以外は「7.2 ドライバ関数の戻り値」参照
	呼出例	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル HMCT110INFO hInfo[2]; // デバイス情報(2枚分) // 1枚目のMCATをオープンします。 ret = mct110_OpenMstBrd(&hDev, &hInfo[0]);

(4) mct110_CloseMstBrd () MCATのクローズ		
機能	MCATをクローズします。	
Win VC++	書式	DWORD WINAPI ubmct_CloseMstBrd(DWORD hDev);
	引数	DWORD hDev・・・デバイスハンドル
	戻り値	'0'で正常, '0'以外は「7.2 ドライバ関数の戻り値」参照
	呼出例	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル ret = mct110_CloseMstBrd(hDev);

(5) mct110_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。	

Win VC++	書 式	DWORD WINAPI mct110_rCenMsts (DWORD hDev,WORD wLine,WORD* wSts);
	引 数	DWORD hDev .. デバイスハンドル WORD wLine .. 予約 (常に0を設定してください) WORD* wSts .. センターメインステータス
	戻り値	'0'で正常, '0'以外は「7.2 ドライバ関数の戻り値」参照
	呼出例	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル WORD sts; // センターメインステータス // センターメインステータス読出し ret = mct110_rCenMsts(hDev, 0, &sts);

(6) mct110_wCenCmd () センターデバイスへのコマンド書き込み		
機 能		センターデバイスへコマンドを書込みます.
Win VC++	書 式	DWORD WINAPI mct110_wCenCmd (DWORD hDev,WORD wLine,WORD wCmd);
	引 数	DWORD hDev .. デバイスハンドル WORD wLine .. 予約 (常に0を設定してください) WORD 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「7.2 ドライバ関数の戻り値」参照
	呼出例	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル // MCATセンターデバイスのサイクリック通信開始 (3000h) ret =mct110_wCenCmd(hDev, 0, 0x3000);

(7) mct110_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。
Win VC++	書 式
	引 数
	戻り値
	呼出例

(8) mct110_rCenBuf () センターデバイスの入出力バッファ読出し mct110_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。
Win VC++	書 式
	引 数
	戻り値
	呼出例

(9) mct110_rCenRFifo () センターデバイスの受信用 FIFO 読出し mct110_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。
Win VC++	書 式
	引 数
	戻り値
	呼出例

(1 0) mct110_rLclCycErr () サイクリック通信エラーフラグ読出し mct110_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。
Win VC++	書 式 DWORD WINAPI mct110_rLclCycErr (DWORD hDev, WORD wLine, WORD wMidPrm, WORD* wFlag); DWORD WINAPI mct110_wLclCycErr (DWORD hDev, WORD wLine, WORD wMidPrm, WORD wFlag);
	引 数 DWORD hDev ・ ・ デバイスハンドル WORD wLine ・ ・ 予約 (常に 0 を設定してください) WORD wMidPrm ・ ・ 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 ~ 63) WORD* wFlag ・ ・ エラーフラグ WORD wFlag ・ ・ エラーフラグ
	戻り値 ' 0 ' で正常 , ' 0 ' 以外は「 7 . 2 ドライバ関数の戻り値」参照
	呼出例 DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル WORD wFlag; // サイクリック通信エラーフラグ // MCATMID = 0 ~ 15 のサイクリック通信エラーフラグ読出し ret = mct110_rLclCycErr(hDev, 0, 0, &wFlag);
	備 考 読み出されたエラーフラグのデータを書込むことでエラーフラグをリセットします。

(1 1) mct110_rLclInfo () ローカルデバイス情報読出し mct110_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。
Win VC++	書 式 DWORD WINAPI mct110_rLclInfo (DWORD hDev, WORD wLine, WORD wMid, BYTE* byData); DWORD WINAPI mct110_wLclInfo (DWORD hDev, WORD wLine, WORD wMid, BYTE byData);
	引 数 DWORD hDev ・ ・ デバイスハンドル WORD wLine ・ ・ 予約 (常に 0 を設定してください) WORD wMid ・ ・ モジュール ID (0 : MID = 0 , . . . , 63 : MID = 63) BYTE* byData ・ ・ ローカルデバイス情報 BYTE byData ・ ・ ローカルデバイス情報
	戻り値 ' 0 ' で正常 , ' 0 ' 以外は「 7 . 2 ドライバ関数の戻り値」参照
	呼出例 DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル BYTE byData; // ローカルデバイス情報 // MCAT デバイス 0 のローカルデバイス情報読出し ret = mct110_rLclInfo(hDev, 0, 0, &byData);
	備 考 モジュール ID は 0 ~ 63 まで指定できますが , 1 ライン接続できるモジュールは 32 個までです。

(1 2) mct110_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 	
---	--

(1 3) mct110_rLclInt ()		ローカルデバイス入力ポート変化フラグ読出し
mct110_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。
Win VC++	書 式	DWORD WINAPI mct110_rLclInt (DWORD hDev,WORD wLine,WORD wMidPrm,WORD* wFlag); DWORD WINAPI mct110_wLclInt (DWORD hDev,WORD wLine,WORD wMidPrm,WORD wFlag);
	引 数	DWORD hDev ・ デバイスハンドル WORD wLine ・ 予約 (常に 0 を設定してください) WORD 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) WORD* 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 WORD 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「 7 . 2 ドライバ関数の戻り値」参照
	呼出例	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル WORD wFlag; // 入力変化フラグ BYTE byData; // 入力ポートデータ // MCATデバイス0 ~ 3のポート0 ~ 3の入力変化を監視 ret = mct110_rLclInt(hDev, 0, 0, &wFlag); if(wFlag & 0x0001) { // デバイス0のポート0に入力変化があった // デバイス0のポート0読出し ret = mct110_rIoPortB(hDev, 0, 0, 0, &byData); ret = mct110_wLclInt(hDev, 0, 0, wFlag); // フラグリセット }
	備 考	1 . 読み出された入力ポート変化フラグのデータを書込むことで入力ポート変化フラグをリセット します。 2 . 入力ポート変化フラグ 例 . DIOデバイス, wMidPrm = 0の時 デバイス3 デバイス2 デバイス1 デバイス0 bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

(1 4) mct110_rIoPortB () D I Oデバイスポート1バイト読出し mct110_rIoPortW () D I Oデバイスポート2バイト読出し mct110_wIoPortB () D I Oデバイスポート1バイト書込み mct110_wIoPortW () D I Oデバイスポート2バイト書込み	
機 能	
指定したD I Oデバイスのポートから読出します。 指定したD I Oデバイスのポートへ書込みます。	
Win VC++	書 式
	DWORD WINAPI mct110_rIoPortB (DWORD hDev,WORD wLine,WORD wMid,WORD wPrtno, BYTE* byData); DWORD WINAPI mct110_rIoPortW (DWORD hDev,WORD wLine,WORD wMid,WORD wPrtno, WORD* wData); DWORD WINAPI mct110_wIoPortB (DWORD hDev,WORD wLine,WORD wMid,WORD wPrtno, BYTE byData); DWORD WINAPI mct110_wIoPortW (DWORD hDev,WORD wLine,WORD wMid,WORD wPrtno, WORD wData);
	引 数
	DWORD hDev ・ ・ デバイスハンドル WORD wLine ・ ・ 予約 (常に0を設定してください) WORD wMid ・ ・ モジュールID (0 : MID = 0 , . . . , 63 : MID = 63) WORD wPrtno ・ ・ ポート番号 (0 : ポート0 , . . . , 3 : ポート3) BYTE* byData ・ ・ 読出されたポートデータ (1 バイト) の格納アドレス BYTE byData ・ ・ 書込むポートデータ (1 バイト) WORD* wData ・ ・ 読出されたポートデータ (2 バイト) の格納アドレス WORD 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「 7 . 2 ドライバ関数の戻り値」参照
Win VC++	呼出例
	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル BYTE byData; // 入力ポートデータ // デバイス0のポート0 (汎用入力状態) 読出し ret = mct110_rIoPortB (hDev, 0, 0, 0, &byData);
Win VC++	備 考
	モジュールIDは0 ~ 63まで指定できますが、1ラインに接続できるモジュールは32個までです。

(1 5) mct110_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 mct110_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。
Win VC++	書 式 DWORD WINAPI mct110_rPclPort (DWORD hDev, WORD wLine, WORD wMid, BYTE* byData); DWORD WINAPI mct110_wPclPort (DWORD hDev, WORD wLine, WORD wMid, BYTE* byData);
	引 数 DWORD hDev .. デバイスハンドル WORD wLine .. 予約 (常に0を設定してください) WORD wMid .. モジュールID (0 : MID = 0 , . . . , 63 : MID = 63) BYTE* byData .. 入出力ポートから読出されたデータ (1バイト) の格納アドレス BYTE 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「7.2 ドライバ関数の戻り値」参照
	呼出例 DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル BYTE byData; // 入力ポートデータ // デバイス0の出力ポートへ書込み (汎用出力) ret = mct110_wPclPort(hDev, 0, 0, 0x01);
	備 考 モジュールIDは0 ~ 63まで指定できますが, 1ラインに接続できるモジュールは32個までです。

(1 6) mct110_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。
Win VC++	書 式 DWORD WINAPI mct110_rPclMsts (DWORD hDev, WORD wLine, WORD wMid, WORD* wSts);
	引 数 DWORD hDev .. デバイスハンドル WORD wLine .. 予約 (常に0を設定してください) WORD wMid .. モジュールID (0 : MID = 0 , . . . , 63 : MID = 63) WORD* 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「7.2 ドライバ関数の戻り値」参照
	呼出例 DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル WORD wSts; // モーションメインステータス // デバイス0のモーションメインステータス読出し ret = mct110_rPclMsts(hDev, 0, 0, &wSts);
	備 考 モジュールIDは0 ~ 63まで指定できますが, 1ラインに接続できるモジュールは32個までです。

(17) mct110_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。
Win VC++	書 式
	引 数
	戻り値
	呼出例
	備 考

(18) mct110_rPclReg () モーションデバイスレジスタからの読み出し mct110_wPclReg () モーションデバイスレジスタへの書込み	
機 能	指定したモーションデバイスのレジスタから読み出します。 指定したモーションデバイスのレジスタへ書込みます。
Win VC++	書 式
	引 数
	戻り値
	呼出例
	備 考

(1 9) mct110_rOptPortB () オプションポート1バイト読出し mct110_rOptPortW () オプションポート2バイト読出し mct110_wOptPortB () オプションポート1バイト書込み mct110_wOptPortW () オプションポート2バイト書込み	
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込みます。	
Win VC++	書 式
	DWORD WINAPI mct110_rOptPortB (DWORD hDev, WORD wAdrs, BYTE* byData); DWORD WINAPI mct110_rOptPortW (DWORD hDev, WORD wAdrs, WORD* wData); DWORD WINAPI mct110_wOptPortB (DWORD hDev, WORD wAdrs, BYTE byData); DWORD WINAPI mct110_wOptPortW (DWORD hDev, WORD wAdrs, WORD wData);
	引 数
	DWORD hDev ・ ・ デバイスハンドル WORD wAdrs ・ ・ オプションポート部ボード内アドレス BYTE* byData ・ ・ 読出しデータ (1バイト) WORD* wData ・ ・ 読出しデータ (2バイト) BYTE byData ・ ・ 書込みデータ (1バイト) WORD wData ・ ・ 書込みデータ (2バイト)
	戻り値
	' 0 'で正常, ' 0 '以外は「 7. 2 ドライバ関数の戻り値」参照
Win VC++	呼出例
	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル // オプションポートへ書込み ret = mct110_wOptPortB(hDev, 0x0010, 1); // OUT1出力
備 考	

(2 0) mct110_rCenPortW () センターデバイス指定ポート2バイト読出し mct110_wCenPortW () センターデバイス指定ポート2バイト書込み	
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込みます。	
Win VC++	書 式
	DWORD WINAPI mct110_rCenPortW (DWORD hDev, WORD wAdrs, WORD* wData); DWORD WINAPI mct110_wCenPortW (DWORD hDev, WORD wAdrs, WORD wData);
	引 数
	DWORD hDev ・ ・ デバイスハンドル WORD wAdrs ・ ・ センターデバイス部ボード内アドレス WORD* wData ・ ・ 読出しデータ (2バイト) WORD wData ・ ・ 書込みデータ (2バイト)
	戻り値
	' 0 'で正常, ' 0 '以外は「 7. 2 ドライバ関数の戻り値」参照
Win VC++	呼出例
	DWORD ret; // 関数の戻り値 DWORD hDev; // デバイスハンドル // センターデバイス送信用FIFOにデータ書込み ret = mct110_wCenPortW(hDev, 0x0006, 0x0090);
備 考	

6.6 ライブラリ関数

ライブラリ関数はドライバ関数で構成され、モーションモジュールHMG - P xの初期化、原点復帰、位置決め動作等の基本的な動作を制御することができます。また各開発言語用ライブラリ関数の仕様は同様になっています。

6.6.1 ライブラリ関数一覧

No		関数名	機 能	備 考
1	デバイス関係	h1px_DevOpenEx	マスターボードのオープンとモーションモジュール初期化	
2		h1px_DevClose	マスターボードのクローズ	
3		h1px_SetOrgMode	原点復帰モードの設定	
4		h1px_SetEls	E L S入力の設定	
5		h1px_SetOls	O L S入力の設定	
6		h1px_SetDls	D L S入力の設定	
7		h1px_SetLtc	L t c h入力の設定	
8		h1px_SetClr	C L R入力の設定	
9		h1px_SetCmp	コンパレータ条件の設定	
10	初期設定	h1px_SetEz	エンコーダZ相の設定	
11		h1px_SetInpos	I N P O S入力の設定	
12		h1px_SetSvAlm	S V A L M入力の設定	
13		h1px_SetSvCtrCl	S V C T R C L出力の設定	
14		h1px_SetSvRdy	S V R D Y入力の設定	
15		h1px_SetCmdPulse	指令パルス出力形式の設定	
16		h1px_SetAccProfile	S字/直線加減速の切替	
17		h1px_SetAutoDec	減速開始点の計算の自動/マニュアル切替	
18		h1px_SetSls	ソフトウェアリミットの設定	
19		h1px_SetCtr3	カウンタ3の入力ソースの設定	
20		h1px_SetFhAdj	F H補正機能のON/OFF	
21		h1px_ReadMainSts	メインステータスの読出し	
22		h1px_ReadErrorSts	エラーステータスの読出し	
23		h1px_ReadEventSts	イベントステータスの読出し	
24	状態読出し	h1px_ReadExSts	拡張ステータスの読出し	
25		h1px_ReadOutp	出力ポート状態読出し	
26		h1px_ReadSpd	指令速度の読出し	
27		h1px_ReadCtr	カウンタの読出し	
28		h1px_SetFLSpd	ベース速度	
29		h1px_SetAuxSpd	補助速度の設定	
30		h1px_SetAccRate	加速レートの設定	
31		h1px_SetDecRate	減速レートの設定	
32		h1px_SetMult	倍率設定値の設定	
33		h1px_SetEventMask	イベントマスクの設定	
34	動作設定	h1px_SetDecPoint	減速開始点の設定	
35		h1px_WriteOpMode	動作モードの設定	
36		h1px_WriteSta	STA 入力時の動作設定	
37		h1px_WriteStp	STP 入力時の動作設定と STP 自動出力の設定	
38		h1px_WriteFhSpd	動作速度の設定	
39		h1px_WritePos	移動量の設定	
40		h1px_WriteCtr	カウンタプリセット	
41		h1px_DecStop	減速停止	
42		h1px_QuickStop	即停止	
43		h1px_EmgStop	非常停止	
44		h1px_AccStart	加速スタート	
45		h1px_CnstStartFH	F H定速スタート	
46		h1px_CnstStartFL	F L定速スタート	
47		h1px_MvAccStart	移動量設定 + 加速スタート	
48		h1px_MvCnstStartFH	移動量設定 + F H定速スタート	
49		h1px_MvCnstStartFL	移動量設定 + F L定速スタート	
50		h1px_SetGroup	グループ設定	
51	動作制御指令	h1px_GrpStop	グループ停止	
52		h1px_GrpAccStart	グループ加速スタート	
53		h1px_GrpCnstStartFH	グループF H定速スタート	
54		h1px_GrpCnstStartFL	グループF L定速スタート	
55		h1px_SvOn	サーボオン	
56		h1px_SvOff	サーボオフ	
57		h1px_SvResetOn	サーボリセットオン	
58		h1px_SvResetOff	サーボリセットオフ	
59		h1px_SvTlOn	サーボトルク制限オン	
60		h1px_SvTlOff	サーボトルク制限オフ	
61		h1px_SvGainOn	サーボゲイン切替オン	
62		h1px_SvGainOff	サーボゲイン切替オフ	
63		h1px_PMOOn	パルスモータ励磁オン	
64		h1px_PMOOff	パルスモータ励磁オフ	
65	加減速レート計算	h1px_CalAccRate	加減速レート計算	
66	減速開始点計算	h1px_CalDecPoint	減速開始点計算	

表 6.6 - 1 ライブラリ関数一覧

6.6.2 ライブラリ関数の戻り値

ライブラリ関数を使用する時、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行います。複数のエラー要因がある時に、戻り値はORされて戻ることがあります。

なお、装置に起因する異常はこの異常報告に含まれません。個々の要因毎に、異常発生内容を明確にすると共に、適切な処置が求められます。

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	デバイスドライバが存在しない デバイスドライバがインストールされていない デバイスドライバが所定のフォルダに格納されていない
3	ALREADY_OPENED	00000002	既にオープン済のデバイスをオープン オープン済みデバイスに更にオープン指令 オープンしたデバイスはクローズするまで使用 (多重のオープン禁止) ボード2枚以上を使用する場合、オープンするデバイス情報の更新を確認します。
4	INSUFFICIENT_MEMORY	00000004	デバイス情報格納メモリが不足 アプリケーション用のメモリ不足 パソコン主記憶メモリの不足 システムリソース(OS用メモリ)の不足 多数のアプリケーション起動 1度に多数のウィンドウを開いた
5	INVALID_HANDLE	00000008	無効なデバイスハンドルを指定 デバイスオープンで得られた"デバイスハンドル"の不使用 このデバイスは既にクローズされている
6	NOT_READY	00000010	デバイスの入出力ポートが使用できない デバイス(ボード)内部の入出力ポートがない (システム不整合が考えられます。弊社までご連絡下さい)
7	ILLEGAL_DEVICE	00000020	ボード固有情報が不正 デバイス情報は正常ですが、ボード機能が不一致です (システム不整合が考えられます。弊社までご連絡下さい)
8	ILLEGAL_ADDRESS	00000040	不正なアドレス
9	ILLEGAL_IRQNUM	00000080	不正な IRQ 番号
10	ILLEGAL_PARAM	00000100	関数の引数の値が異常 その他引数の設定値を確認
11	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
12	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
13	DATA_COM_ERROR	00040000	データ通信エラー
14	COM_OTHER_ERROR	00080000	その他の通信エラー センター割込ステータスで詳細を確認して下さい
15	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。
16	MODULE_COUNT_ERROR	00200000	モジュールの数が合いません。モジュールのID設定、モジュールの接続、電源供給状態、ケーブル等確認して下さい。
17	ILL_ACCESS_COM	00400000	通信中の不正なアクセス
18	SEND_DATA_NONUSE	00800000	未使用デバイスにデータ送信
19	OTHER_ERROR	80000000	その他のエラー

表6.6-2 ライブラリ関数の戻り値

6.6.3 ライブラリ関数詳細

ここではVC++表記で説明します。VB, VB.NETを使用される場合、データ型の対応は下表の通りです。

引数のデータ型, 16進数表記の対応

言 語	VC++言語	VB	VB.NET
データ型	8bit	BYTE	Byte
	16bit	WORD	Short
	32bit	DWORD	Integer
データ 例	0x0000	&H0	&H0
	0x1000	&H1000	&H1000

デバイス関係

(1) h1px_DevOpenEx () デバイスのオープン, レジスタとオプションポートの初期化

機 能	指定したデバイス情報を持つマスターボードをオープンし, 他のマスターボードと識別するためのデバイスハンドルを取得します。以降このデバイスハンドルは, このマスターボードにアクセスするために使用します。またオープンしたマスターボードに接続されたモーションモジュールの初期化 (1) をします。
書 式	DWORD h1px_DevOpenEx(DWORD* hDevID, HMCT110INFO* hInfo);
引 数	DWORD* hDevID ... デバイスハンドル HMCT110INFO* hInfo ... オープンするデバイス情報構造体のポインタ HMCT110INFO* hMdl ... オープンするモジュール情報構造体のポインタ
備 考	<p>< 呼び出し例 ></p> <p>パソコンにマスターボードが2枚装着されていることを想定します。 デバイス情報構造体として HMCT110INFO 型の配列 hInfo[2], モジュール情報構造体として HUBMDLINFO 型の配列 hMdl[2] を準備し, これらの中には既に情報が入っているものとします。 DWORD ret; // 関数の戻り値 DWORD hDevID[2]; // デバイスハンドル取得エリア</p> <p>ret = h1px_DevOpenEx(hDevID[0], &hInfo[0], &hMdl[0]); // 1 番目のデバイスオープン ret = h1px_DevOpenEx(hDevID[1], &hInfo[1], &hMdl[1]); // 2 番目のデバイスオープン</p>

【モーションモジュールのレジスタの初期値 (1)】

レジスタ	内容	レジスタ初期値	備 考
RFL	ベース速度	200	200pps
RFH	動作速度	2000	2000pps
RUR	加速レート	1387	200pps 2,000ppsの加減速時間約500msec (直線加減速時)
RMG	速度倍率	199	1倍
RFA	補助速度	200	200pps
REN1	環境設定 1	0x01434004	指令パルス出力形式: CW/CCW, SVCTRL自動出力しない, SVCTRL出力パルス幅: 13ms, DLS, OLS, SVALM: B接, ELS, SVALM入力時即停止, DLSタッチしない, SVRDY, INPOS: A接, LATC, CLR: 立下りエッジ
REN2	環境設定 2	0x000004FF	サーボ I/F出力設定, エンコーダ入力4通倍
REN3	環境設定 3	0x00700002	原点復帰モード2 (OLS+Z), 原点復帰完了時カウンタ1~3をクリア
RIRQ	イベントマスク設定	0x00001	正常停止時

本表以外のレジスタは初期値0 (レジスタの詳細は5.「ローカルデバイス」を参照)

1. レジスタの初期値は関数内で直接与えています。

表6.6-3 モーションモジュールのレジスタの初期値

(2) h1px_DevClose () デバイスのクローズ

機 能	デバイスハンドルで指定されたマスターボードをクローズします。 以降、このデバイスハンドルは無効となります。
書 式	DWORD h1px_DevClose(DWORD hDevID);
引 数	DWORD* hDevID・・・デバイスハンドル
備 考	デバイスのクローズ処理ではデバイスハンドルを無効にするのみで、パルス出力の停止や、サーボオフは行いません。モーションモジュールに対する終了処理をした後でデバイスをクローズして下さい。

初期設定

(3) h1px_SetOrgMode () 原点復帰モードの設定

機 能	引数で指定されたマスターボード、モジュールIDのモーションモジュールの原点復帰モードを設定します。
書 式	DWORD h1px_SetOrgMode(DWORD hDevID, WORD wLine, WORD wMid, WORD wMode);
引 数	DWORD* hDevID・・・デバイスハンドル WORD wLine・・・予約[常に0を設定してください] WORD wMid・・・M_ID 指定[0:モジュールID=0, ~ , 63:モジュールID=63] WORD wMode・・・原点復帰モード[0:ORGmode0, 1:ORGmode1, ~ , 12:ORGmode12] (表6.2-5 原点復帰方法, 表6.2-5 原点復帰方法(CTR2参照方式)参照)
備 考	

(4) h1px_SetEIs () ELSの設定

機 能	引数で指定されたマスターボード、モジュールIDのモーションモジュールのELSの入力極性と入力停止方法を設定します。
書 式	DWORD h1px_SetEIs(DWORD hDevID, WORD wLine, WORD wMid, WORD wPol, WORD wStop);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に0を設定してください] WORD wMid・・・M_ID 指定[0:モジュールID=0, ~ , 63:モジュールID=63] WORD wPol・・・極性 [0: B 接, 1: A 接] WORD wStop・・・停止方法 [0: 即停止, 1: 減速停止]
備 考	

(5) h1px_SetOIs () OLSの設定

機 能	引数で指定されたマスターボード、モジュールIDのモーションモジュールのOLSの入力極性を設定します。
書 式	DWORD h1px_SetOIs(DWORD hDevID, WORD wLine, WORD wMid, WORD wPol);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に0を設定してください] WORD wMid・・・M_ID 指定[0:モジュールID=0, ~ , 63:モジュールID=63] WORD wPol・・・極性 [0: B 接, 1: A 接]
備 考	

(6) h1px_SetDis () D L Sの設定

機 能	引数で指定されたマスターボード , モジュール I D のモーションモジュールの D L S の入力極性を設定します .
書 式	DWORD h1px_SetDis(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wEnbl</i> , WORD <i>wPol</i> , WORD <i>wStop</i> , WORD <i>wLtc</i>);
引 数	DWORD <i>hDevID</i> . . デバイスハンドル WORD <i>wLine</i> . . 予約[常に 0 を設定してください] WORD <i>wMid</i> . . M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD <i>wEnbl</i> . . DLS 有効 / 無効[0:無効, 1:有効] WORD <i>wPol</i> . . 極性 [0: B 接 , 1: A 接] WORD <i>wStop</i> . . D L S 入力時の動作 [0:減速のみ, 1:減速停止] WORD <i>wLtc</i> . . D L S 入力のラッチ [0:ラッチしない, 1:ラッチする]
備 考	<i>wEnbl</i> =0(DLS 無効設定)の時は <i>wStop</i> , <i>wLtc</i> は無効です .

(7) h1px_SetLtc () L A T C 入力の設定

機 能	引数で指定されたマスターボード , モジュール I D のモーションモジュールの LATC 入力極性を設定します .
書 式	DWORD h1px_SetLtc(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wPol</i>);
引 数	DWORD <i>hDevID</i> . . デバイスハンドル WORD <i>wLine</i> . . 予約[常に 0 を設定してください] WORD <i>wMid</i> . . M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD <i>wPol</i> . . 極性[0:立下がりエッジ, 1:立上がりエッジ]
備 考	

(8) h1px_SetClr () C T R 入力の設定

機 能	引数で指定されたマスターボード , モジュール I D のモーションモジュールの C L R 入力設定とクリアするカウンタを設定します .
書 式	DWORD h1px_SetClr(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wPol</i> , WORD <i>wCtr</i>);
引 数	DWORD <i>hDevID</i> . . デバイスハンドル WORD <i>wLine</i> . . 予約[常に 0 を設定してください] WORD <i>wMid</i> . . M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD <i>wPol</i> . . 入力設定[0:立下がりエッジ, 1:立上がりエッジ, 2:L レベル, 3:H レベル] WORD <i>wCtr</i> . . クリアするカウンタの選択[0x0001:CTR1, 0x0002:CTR2, 0x0004:CTR3] (O R した値で選択)
備 考	

(9) h1px_SetCmp () コンパレータ条件の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールのコンパレータ条件を設定します．
書 式	DWORD h1px_SetCmp(DWORD hDevID, WORD wLine, WORD wMid, WORD wCtr, WORD wCon, WORD wIndex, long lCmp);
引 数	DWORD hDevID ・・ デバイスハンドル WORD wLine ・・ 予約[常に 0 を設定してください] WORD wMid ・・ M_ID 指定[0: モジュール ID=0, ~ , 63: モジュール ID=63] WORD wCtr ・・ コンパレータ比較カウンタ [0: CTR1, 1: CTR2, 2: CTR3] WORD wCon ・・ コンパレータ条件[0: 常に条件不成立, 1: cmp=比較カウンタ(カウント方向無関係), 2: cmp=比較カウンタ(カウントアップ中), 3: cmp=比較カウンタ(カウントダウン中), 4: cmp>比較カウンタ, 5: cmp<比較カウンタ] WORD wIndex ・・ 同期(定ピッチ)出力有効 / 無効 [0: 無効, 1: 有効] long lCmp ・・ コンパレータ比較データ値 [-134,217,728 ~ +134,217,727]
備 考	同期(定ピッチ)出力有効(wIndex=1)にした場合は wCon=4,5 の設定は不可

(1 0) h1px_SetEz () エンコーダ Z 相の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールのエンコーダ Z 相の入力処理を設定します．
書 式	DWORD h1px_SetEz(DWORD hDevID, WORD wLine, WORD wMid, WORD wCount, WORD wPol);
引 数	DWORD hDevID ・・ デバイスハンドル WORD wLine ・・ 予約[常に 0 を設定してください] WORD wMid ・・ M_ID 指定[0: モジュール ID=0, ~ , 63: モジュール ID=63] WORD wCount ・・ 原点復帰時の Z 相回数[0: 1 回目, ~ , 15: 16 回目] WORD wPol ・・ 極性[0: 立下がりエッジ, 1: 立上がりエッジ]
備 考	

(1 1) h1px_SetInpos () INPOS の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの INPOS の入力信号処理方法を設定します．
書 式	DWORD h1px_SetInpos(DWORD hDevID, WORD wLine, WORD wMid, WORD wEnbl, WORD wPol);
引 数	DWORD hDevID ・・ デバイスハンドル WORD wLine ・・ 予約[常に 0 を設定してください] WORD wMid ・・ M_ID 指定[0: モジュール ID=0, ~ , 63: モジュール ID=63] WORD wEnbl ・・ INPOS 制御有効 / 無効[0: 無効, 1: 有効] WORD wPol ・・ 極性 [0: B 接, 1: A 接]
備 考	

(1 2) h1px_SetSvalm () SVALM の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの SVALM の入力極性と入力停止方法を設定します．
書 式	DWORD h1px_SetSvalm(DWORD hDevID, WORD wLine, WORD wMid, WORD wPol);
引 数	DWORD hDevID ・・ デバイスハンドル WORD wLine ・・ 予約[常に 0 を設定してください] WORD wMid ・・ M_ID 指定[0: モジュール ID=0, ~ , 63: モジュール ID=63] WORD wPol ・・ 極性 [0: B 接, 1: A 接] WORD wStop ・・ 停止方法 [0: 即停止, 1: 減速停止]
備 考	

(1 3) h1px_SetSvCtrCl () 偏差カウンタクリア出力の設定

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールの偏差カウンタクリア出力の設定をします。
書 式	DWORD h1px_SetSvCtrCl(DWORD hDevID, WORD wLine, WORD wMid, WORD wEnbl);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wEnbl・・・自動出力設定 [0:不使用, 1:原点完了時, 2:異常停止時, 3:原点完了及び異常停止時]
備 考	

(1 4) h1px_SetSvRdy () サーボレディ入力の設定

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールの S V R D Y の入力極性を設定します。
書 式	DWORD h1px_SetSvRdy(DWORD hDevID, WORD wLine, WORD wMid, WORD wPol);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wPol・・・極性 [0: B 接, 1: A 接]
備 考	

(1 5) h1px_SetCmdPulse () 指令パルスの出力形式の設定

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールの指令パルスの出力形式を設定します。
書 式	DWORD h1px_SetCmdPulse(DWORD hDevID, WORD wLine, WORD wMid, WORD wCmdpls);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wCmdpls・・・指令パルスの出力形式[0:個別指令方式, 1:共通指令方式]
備 考	

(1 6) h1px_SetAccProfile () S 字/直線加減速の切替

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールの S 字 / 直線加減速の設定をします。
書 式	DWORD h1px_SetAccProfile(DWORD hDevID, WORD wLine, WORD wMid, WORD wPro);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wPro・・・加減速形式 [0:直線, 1: S 字]
備 考	

(1 7) h1px_SetAutoDec () 減速開始点の設定方式の自動/マニュアル切替

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールの減速開始点の設定をマニュアル(手動)か自動が設定します。
書 式	DWORD h1px_SetAutoDec(DWORD hDevID, WORD wLine, WORD wMid, WORD wAuto);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wAuto・・・減速開始点の設定方式 [0:自動設定, 1:マニュアル設定]
備 考	

(1 8) h1px_SetSls () ソフトリミットの設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールのソフトリミット処理を設定します．
書 式	DWORD h1px_SetSls(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , long <i>IPsl</i> , long <i>ImSl</i> , WORD <i>wEnbl</i> , WORD <i>wStop</i>);
引 数	DWORD <i>hDevID</i> ・ ・ デバイスハンドル WORD <i>wLine</i> ・ ・ 予約 [常に 0 を設定してください] WORD <i>wMid</i> ・ ・ M_ID 指定 [0: モジュール ID=0, ~ , 63: モジュール ID=63] long <i>IPsl</i> ・ ・ + S L S [パルス数] long <i>ImSl</i> ・ ・ - S L S [パルス数] WORD <i>wEnbl</i> ・ ・ 使用 / 不使用 [0: 不使用, 1: 使用] WORD <i>wStop</i> ・ ・ 停止方法 [0: 即停止, 1: 減速停止]
備 考	1 . ソフトリミット使用時は <i>psls</i> は必ず <i>msls</i> より大きくして下さい． 2 . ソフトリミット不使用時は <i>msls</i> = <i>psls</i> = 0 とし， <i>enable</i> = 0 とします． 3 . スタートコマンド書込み時に S L S が O N 状態の場合，S L S が O N 状態の方向へは，スタートはできません．逆方向へはスタートできます．

(1 9) h1px_SetCtr3 () カウンタ 3 の入力ソースの設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールのカウンタ 3 の入力ソースを設定します．
書 式	DWORD h1px_SetCtr3(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wSrc</i>);
引 数	DWORD <i>hDevID</i> ・ ・ デバイスハンドル WORD <i>wLine</i> ・ ・ 予約 [常に 0 を設定してください] WORD <i>wMid</i> ・ ・ M_ID 指定 [0: モジュール ID=0, ~ , 63: モジュール ID=63] WORD <i>wSrc</i> ・ ・ 入力ソース [0: 指令パルス, 1: エンコーダ入力, 2: 予約, 3: 予約, 4: 指令パルスとエンコーダ入力の偏差カウント]
備 考	

(2 0) h1px_SetFHAdj () FH 補正機能の ON/OFF

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの F H 補正機能（三角駆動自動回避機能）の O N / O F F をします．
書 式	DWORD h1px_SetFHAdj(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wEnbl</i>);
引 数	DWORD <i>hDevID</i> ・ ・ デバイスハンドル WORD <i>wLine</i> ・ ・ 予約 [常に 0 を設定してください] WORD <i>wMid</i> ・ ・ M_ID 指定 [0: モジュール ID=0, ~ , 63: モジュール ID=63] WORD <i>wEnbl</i> ・ ・ F H 補正設定 [0: O F F, 1: O N]
備 考	

状態読出し

- (2 1) h1px_ReadMainSts () メインステータスの読出し
 (2 2) h1px_ReadErrorSts () エラーステータスの読出し
 (2 3) h1px_ReadEventSts () イベントステータスの読出し
 (2 4) h1px_ReadExSts () 拡張ステータスの読出し

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールのメインステータス，エラーステータス，イベントステータス，拡張ステータスを読出し，指定したエリアに格納します．ステータスデータについては，「 5 . 2 . 7 モーションデバイスレジスタ」を参照して下さい．
書 式	DWORD h1px_ReadMainSts (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD* <i>wSts</i>); DWORD h1px_ReadErrorSts (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , DWORD* <i>dwSts</i>); DWORD h1px_ReadEventSts (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , DWORD* <i>dwSts</i>); DWORD h1px_ReadExSts (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , DWORD* <i>dwSts</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD* <i>wSts</i> ・・・メインステータスが格納されるエリアのアドレス DWORD* <i>dwSts</i> ・・・エラー・イベント・拡張ステータスが格納されるエリアのアドレス
備 考	h1px_ReadErrorSts(), h1px_ReadEventSts() はステータス読出し後，割込みリセットコマンド (0008h) を発行しています．(MMSTS の b1 のビット)

- (2 5) h1px_ReadOutp () 出力ポートの読出し

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの出力ポートを読出し，指定したエリアに格納します．
書 式	DWORD h1px_ReadOutp (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , BYTE* <i>byData</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] BYTE* <i>byData</i> ・・・読出されたデータが格納されるエリアのアドレス
備 考	

- (2 6) h1px_ReadSpd () 現在速度の読出し

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの指令速度を読出し，指定したエリアに格納します．実指令速度への変換は読出し値に速度倍率をかけます．
書 式	DWORD h1px_ReadSpd (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , DWORD* <i>dwSpd</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD* <i>dwSpd</i> ・・・読出されたデータが格納されるエリアのアドレス
備 考	

- (2 7) h1px_ReadCtr () カウンタの読出し

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの，指定されたカウンタを読出し，指定したエリアに格納します．
書 式	DWORD h1px_ReadCtr (DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wCtr</i> , long* <i>lValue</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD <i>wCtr</i> ・・・カウンタ選択[1:カウンタ 1, 2:カウンタ 2, 3:カウンタ 3] long* <i>lValue</i> ・・・読出されたデータが格納されるエリアのアドレス
備 考	

動作設定

(28) h1px_SetFLSpd () ベース速度の設定

(29) h1px_SetAuxSpd () 補助速度の設定

機能	引数で指定されたマスターボード、モジュールIDのモーションモジュールの ベース速度・加減速動作時の立ち上がりの速度（本速度から加速，本速度まで減速して停止） 補助速度・・・一部の原点復帰において，原点突入速度等を使用 を設定します．実速度（pps）を倍率で除算した値を設定します．
書式	DWORD h1px_SetFLSpd (DWORD hDevID, WORD wLine, WORD wMid, DWORD dwSpd); DWORD h1px_SetAuxSpd(DWORD hDevID, WORD wLine, WORD wMid, DWORD dwSpd);
引数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に0を設定してください] WORD wMid・・・M_ID 指定[0:モジュールID=0, ~ , 63:モジュールID=63] DWORD dwSpd・・・ベース速度(FL), 補助速度(FA)レジスタ値 [1~100,000, FL(FA) < FH(動作速度)]
備考	

(30) h1px_SetAccRate () 加速レートの設定

(31) h1px_SetDecRate () 減速レートの設定

機能	引数で指定されたマスターボード、モジュールIDのモーションモジュールの加速レート（減速レート）を設定します．
書式	DWORD h1px_SetAccRate(DWORD hDevID, WORD wLine, WORD wMid, DWORD dwRur); DWORD h1px_SetDecRate(DWORD hDevID, WORD wLine, WORD wMid, DWORD dwRdr);
引数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に0を設定してください] WORD wMid・・・M_ID 指定[0:モジュールID=0, ~ , 63:モジュールID=63] DWORD dwRur・・・加速レート[1~65535] DWORD dwRdr・・・減速レート[0~65535](1)
備考	1．加速レート（減速レート）と加減速時間の関係 RFH, RFL, RUR (RDR) (1)直線加減速の時加減速時間(秒)=(RFH - RFL) × (RUR + 1) × 4 / 40,000,000 (2) S 字加減速の時加減速時間(秒)=(RFH - RFL) × (RUR + 1) × 8 / 40,000,000 減速レート=0の時は減速レート=加速レートとなります．

(32) h1px_SetMult () 倍率設定値(MG)の設定

機能	引数で指定されたマスターボード、モジュールIDのモーションモジュールの速度の倍率設定値(MG)(1)を設定します．
書式	DWORD h1px_SetMult(DWORD hDevID, WORD wLine, WORD wMid, DWORD dwRmg);
引数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に0を設定してください] WORD wMid・・・M_ID 指定[0:モジュールID=0, ~ , 63:モジュールID=63] DWORD dwRmg・・・速度倍率レジスタ値[1~7fff](1)
備考	1．速度と倍率の関係及び倍率設定値(MG)と速度倍率の関係 (RFxは速度レジスタ(RFH, RFL, RFA)の値) 速度(pps) = RFx × 倍率 倍率設定値(MG) = (200 / 速度倍率) - 1

[倍率設定例]

設定値	倍率	出力速度範囲(pps)	設定値	倍率	出力速度範囲(pps)
1999(0x7cf)	0.1	0.1 ~ 10,000	39(0x027)	5	5 ~ 500,000
999(0x3e7)	0.2	0.2 ~ 20,000	19(0x013)	10	10 ~ 1,000,000
399(0x18f)	0.5	0.5 ~ 50,000	9(0x009)	20	20 ~ 2,000,000
199(0x0c7)	1	1 ~ 100,000	3(0x003)	50	50 ~ 5,000,000
99(0x063)	2	2 ~ 200,000	2(0x002)	66.6	100 ~ 6,666,666

(3 3) h1px_SetEventMask () イベントマスクの設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールのイベントマスクを設定します．
書 式	DWORD h1px_SetEventMask(DWORD hDevID, WORD wLine, WORD wMid, DWORD dwMsk);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] DWORD dwMsk・・・イベントマスクデータ 下表の値を O R したデータを与えることで 2 種類以上のイベント報告指定となります．
備 考	

[イベントマスク設定値]

設定値	イベントステータス報告のイベント	設定値	イベントステータス報告のイベント
0x00001	正常停止時	0x00080	コンパレータ 3 条件成立時
0x00002	加速開始時	0x00100	CLR 入力によるカウント値のクリア時
0x00004	加速終了時	0x00200	LTC 入力によるカウント値のラッチ時
0x00008	減速開始時	0x00400	OLS 入力によるカウント値のラッチ時
0x00010	減速終了時	0x00800	DLS 入力 O N 時
0x00020	コンパレータ 1 条件成立時	0x01000	STA 入力 O N 時
0x00040	コンパレータ 2 条件成立時		

(3 4) h1px_SetDecPoint () 減速開始点の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの減速開始点自動計算をマニュアル設定とした時の減速開始点，または減速開始点自動計算を自動に設定した時の減速開始点オフセット値を設定します．減速開始点自動計算マニュアル設定時は設定値は残移動量の値です． 減速開始点自動計算手動設定時の適正な減速開始点の計算方法については，「 6 . 3 . 4 速度パターン設定レジスタの (7) R D P : 減速開始点レジスタ」を参照して下さい．減速開始点自動計算自動設定時は設定値は自動計算値のオフセット値となります．その時に正の値を設定した場合早めに減速します．負の値を設定した場合は遅めに減速します．
書 式	DWORD h1px_SetDecPoint(DWORD hDevID, WORD wLine, WORD wMid, long IRdp);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] long IRdp・・・減速開始点パルス数
備 考	

運用設定

(3 5) h1px_WritOpeMode () 動作モードの設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの動作モードを設定します．
書 式	DWORD h1px_WritOpeMode(DWORD hDevID, WORD wLine, WORD wMid, WORD wMode);
引 数	DWORD hDevID ・・・デバイスハンドル WORD wLine ・・・予約[常に 0 を設定してください] WORD wMid ・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wMode ・・・動作モード（次表参照）
備 考	

[動作モード]

mode (Hex)	動作モード	備考
0 x 0 0	コマンド制御による + 方向連続動作	
0 x 0 8	コマンド制御による - 方向連続動作	
0 x 0 1	パルス入力による連続動作	
0 x 1 0	+ 方向原点復帰動作	
0 x 1 8	- 方向原点復帰動作	
0 x 1 2	+ 方向原点抜け出し動作	
0 x 1 a	- 方向原点抜け出し動作	
0 x 1 5	+ 方向原点サーチ動作	h1px_WritPos() 関数で引出量を設定
0 x 1 d	- 方向原点サーチ動作	
0 x 2 0	+ E L 又は + S L 位置まで動作	
0 x 2 8	- E L 又は S L 位置まで動作	
0 x 2 2	+ E L 又は + S L 抜け出し動作	
0 x 2 a	- E L 又は S L 抜け出し動作	
0 x 2 4	+ 方向に E Z カウント分だけ動作	
0 x 2 c	- 方向に E Z カウント分だけ動作	
0 x 4 1	位置決め動作	h1px_WritPos() 関数で移動量を設定
0 x 4 2	P C S 位置決め動作（ライブラリ関数作成動作）	
0 x 4 4	指令位置 0 点復帰動作	
0 x 4 5	機械位置 0 点復帰動作	
0 x 4 6	+ 方向 1 パルス動作	
0 x 4 e	- 方向 1 パルス動作	
0 x 4 7	タイマー動作	h1px_WritPos() 関数で移動量(タイマ-時間)を設定
0 x 5 1	パルス入力による位置決め動作	
0 x 5 4	パルス入力による指令位置 0 点復帰動作	
0 x 5 5	パルス入力による機械位置 0 点復帰動作	

(3 6) h1px_WritSta () S T A 入力時の動作設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの S T A 入力時の動作を設定します．
書 式	DWORD h1px_WritSta(DWORD hDevID, WORD wLine, WORD wMid, WORD wEnbl, WORD wTrg);
引 数	DWORD hDevID ・・・デバイスハンドル WORD wLine ・・・予約[常に 0 を設定してください] WORD wMid ・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wEnbl ・・・有効，無効設定[0:無効, 1:有効] WORD wTrg ・・・トリガ指定[0:レベルトリガ, 1:エッジトリガ]
備 考	モジュールを同期させる時などに予めこの設定をします．

(3 7) h1px_WritStp () S T P 入力時の動作設定と S T P 自動出力の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの S T P 入力時の動作と異常停止時の S T P 自動出力の設定します．
書 式	DWORD h1px_WritSta(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , WORD <i>wEnbl</i> , WORD <i>wStop</i> , WORD <i>wOut</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ～ , 63:モジュール ID=63] WORD <i>wEnbl</i> ・・・有効，無効設定[0:無効, 1:有効] WORD <i>wStop</i> ・・・S T P 入力時の停止方法[0:即停止, 1:減速停止] WORD <i>wOut</i> ・・・異常停止時 S T P 出力[0:出力しない, 1:出力する]
備 考	モジュールを異常停止時に同一スレーブ上の他のモジュールを停止させたい時などに用います．

(3 8) h1px_WritFHSpd () 動作速度の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの動作速度（実速度を倍率で除算）を設定します．
書 式	DWORD h1px_WritFHSpd(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , DWORD <i>dwSpd</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ～ , 63:モジュール ID=63] DWORD <i>dwSpd</i> ・・・F H（動作速度）レジスタ値[1～100,000]
備 考	

(3 9) h1px_WritPos () 位置決め移動量の設定

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの位置決め動作の移動量を設定します．
書 式	DWORD h1px_WritPos(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , long <i>IDst</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ～ , 63:モジュール ID=63] long <i>IDst</i> ・・・移動量[-134,217,728～134,217,727(28ビット)]
備 考	

(4 0) h1px_WritCtr () カウンタプリセット

機 能	引数で指定されたマスターボード，モジュール I D のモーションモジュールの指定されたカウンタをプリセットします．
書 式	DWORD h1px_WritCtr(DWORD <i>hDevID</i> , WORD <i>wLine</i> , WORD <i>wMid</i> , long <i>IPre</i> , WORD <i>wCtr</i>);
引 数	DWORD <i>hDevID</i> ・・・デバイスハンドル WORD <i>wLine</i> ・・・予約[常に 0 を設定してください] WORD <i>wMid</i> ・・・M_ID 指定[0:モジュール ID=0, ～ , 63:モジュール ID=63] long <i>IPre</i> ・・・プリセット値[-134,217,728～134,217,727(28ビット)] WORD <i>wCtr</i> ・・・カウンタ選択[1:カウンタ 1, 2:カウンタ 2, 3:カウンタ 3]
備 考	

動作制御指令

(4 1) h1px_DecStop () 減速停止

(4 2) h1px_QuickStop () 即停止

(4 3) h1px_EmgStop () 非常停止

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールを減速停止、即停止、非常停止します。
書 式	DWORD h1px_DecStop (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_QuickStop(DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_EmgStop (DWORD hDevID, WORD wLine, WORD wMid);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63]
備 考	

(4 4) h1px_AccStart () 加速スタート

(4 5) h1px_CnstStartFH () F H 定速スタート

(4 6) h1px_CnstStartFL () F L 定速スタート

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールを加速スタート、F H 定速スタート、F L 定速スタートします。
書 式	DWORD h1px_AccStart (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_CnstStartFH (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_CnstStartFL (DWORD hDevID, WORD wLine, WORD wMid);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63]
備 考	

(4 7) h1px_MvAccStart () 移動量設定 + 加速スタート

(4 8) h1px_MvCnstStartFH () 移動量設定 + F H 定速スタート

(4 9) h1px_MvCnstStartFL () 移動量設定 + F L 定速スタート

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールに移動量設定した後で加速スタート、F H 定速スタート、F L 定速スタートします。
書 式	DWORD h1px_MvAccStart (DWORD hDevID, WORD wLine, WORD wMid, long lDst); DWORD h1px_MvCnstStartFH (DWORD hDevID, WORD wLine, WORD wMid, long lDst); DWORD h1px_MvCnstStartFL (DWORD hDevID, WORD wLine, WORD wMid, long lDst);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] long lDst・・・符号付移動量
備 考	h1px_AccStart(), h1px_CnstStartFH(), h1px_CnstStartFL() を使用する場合は移動量設定(データ通信)をした後でコマンド発行(データ通信)という手順になりデータ通信 2 回になります。 h1px_MvAccStart(), h1px_MvCnstStartFH(), h1px_MvCnstStartFL() を使用する場合は移動量設定とコマンド発行を一度のデータ通信で行います。

(5 0) h1px_SetGroup () グループ設定

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールを指定したグループに設定します。 同じグループに設定されたモーションモジュールはライブラリ関数により減速停止、即停止、加速スタート、F H 定速スタート、F L 定速スタートできます。
書 式	DWORD h1px_GrDecStop (DWORD hDevID, WORD wLine, WORD wMid, WORD wGroup); DWORD h1px_GrQuickStop(DWORD hDevID, WORD wLine, WORD wMid, WORD wGroup);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63] WORD wGroup・・・グループ指定[1:グループ 1, ~ , 7:グループ 7]
備 考	

(5 1) h1px_GrpStop () グループモジュール停止

機 能	引数で指定されたマスターボード、グループのモーションモジュールを S T A 入力時の設定に従って減速停止，即停止します．
書 式	DWORD h1px_GrpStop (DWORD hDevID, WORD wLine, WORD wGroup);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wGroup・・・グループ指定[0:全てのグループ, 1:グループ 1, ~ , 7:グループ 7]
備 考	予めグループ設定及び S T P 入力時の動作設定をする必要があります． 指定したグループのモジュールがない場合は戻り値は 0 (正常) が戻り，何もしません．

(5 2) h1px_GrpAccStart () グループ加速スタート

(5 3) h1px_GrpCnstStartFH () グループ F H 定速スタート

(5 4) h1px_GrpCnstStartFL () グループ F L 定速スタート

機 能	引数で指定されたマスターボード、グループのモーションモジュールを加速スタート，F H 定速スタート，F L 定速スタートします．
書 式	DWORD h1px_GrpAccStart (DWORD hDevID, WORD wLine, WORD wGroup); DWORD h1px_GrpCnstStartFH (DWORD hDevID, WORD wLine, WORD wGroup); DWORD h1px_GrpCnstStartFL (DWORD hDevID, WORD wLine, WORD wGroup);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wGroup・・・グループ指定[0:全てのグループ, 1:グループ 1, ~ , 7:グループ 7]
備 考	予めグループ設定及び S T A 入力時の動作設定をする必要があります． 指定したグループのモジュールがない場合は戻り値は 0 (正常) が戻り，何もしません．

(5 5) h1px_SvOn () サーボオン

(5 6) h1px_SvOff () サーボオフ

(5 7) h1px_SvResetOn () サーボリセットオン

(5 8) h1px_SvResetOff () サーボリセットオフ

(5 9) h1px_SvTlOn () サーボトルク制限オン

(6 0) h1px_SvTlOff () サーボトルク制限オフ

(6 1) h1px_SvGainOn () サーボゲイン切替オン

(6 2) h1px_SvGainOff () サーボゲイン切替オフ

(6 3) h1px_PMOOn () パルスモータ励磁オン

(6 4) h1px_PMOOff () パルスモータ励磁オフ

機 能	引数で指定されたマスターボード、モジュール I D のモーションモジュールのサーボオン，サーボリセット，サーボトルク制限，サーボゲイン切替，パルスモータ励磁出力をオン / オフします．
書 式	DWORD h1px_SvOn (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvOff (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvResetOn (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvResetOff (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvTlOn (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvTlOff (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvGainOn (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_SvGainOff (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_PMOOn (DWORD hDevID, WORD wLine, WORD wMid); DWORD h1px_PMOOff (DWORD hDevID, WORD wLine, WORD wMid);
引 数	DWORD hDevID・・・デバイスハンドル WORD wLine・・・予約[常に 0 を設定してください] WORD wMid・・・M_ID 指定[0:モジュール ID=0, ~ , 63:モジュール ID=63]
備 考	

加減速レート計算

(6 5) h1px_CalAccRate () 加減速レート計算

機 能	加減速時間, R F H, R F L 等より加減速レートを計算します。
書 式	DWORD h1px_CalAccRate(DWORD* <i>dwRur</i> , DWORD <i>dwTim</i> , DWORD <i>dwRfh</i> , DWORD <i>dwRfl</i> , WORD <i>wPro</i> , DWORD <i>dwRs</i>);
引 数	DWORD* <i>dwRur</i> ・・加減速レート(RUR)の計算結果のポインタ(加減速レート[1~65,535]) DWORD <i>dwTim</i> ・・加減速時間(msec) DWORD <i>dwRfh</i> ・・RFH レジスタ値[1~100,000] DWORD <i>dwRfl</i> ・・RFL レジスタ値[1~100,000] WORD <i>wPro</i> ・・加減速形式 [0:直線, 1: S 字] DWORD <i>dwRs</i> ・・RUS(RDS)レジスタ値(S 字加減速時の S 字速度区間)
備 考	

減速開始点の計算

(6 6) h1px_CalDecPoint () 減速開始点の計算

機 能	減速開始点手動設定時の減速開始点最適値を計算します。 位置決め動作時に, この値より大きな値を減速開始点レジスタに設定するとベース速度で動作する時間が長くなり, 逆に小さな値を減速開始点レジスタに設定するとベース速度に到達する前に停止します。
書 式	DWORD h1px_CalDecPoint(DWORD* <i>dwRdp</i> , DWORD <i>dwRfh</i> , DWORD <i>dwRfl</i> , DWORD <i>dwRmg</i> , DWORD <i>dwRdr</i> , WORD <i>wPro</i> , DWORD <i>dwRds</i>);
引 数	DWORD* <i>dwRdp</i> ・・減速開始点(パルス)の計算結果のポインタ(減速開始点[0~16,777,215]) DWORD <i>dwRfh</i> ・・RFH レジスタ値[1~100,000] DWORD <i>dwRfl</i> ・・RFL レジスタ値[1~100,000] DWORD <i>dwRmg</i> ・・RMG レジスタ値[2~2,047] DWORD <i>dwRdr</i> ・・RDR レジスタ値(減速レート[1~65,535]) WORD <i>wPro</i> ・・加減速形式 [0:直線, 1: S 字] DWORD <i>dwRds</i> ・・RDS レジスタ値(S 字加減速時の減速時 S 字速度区間[0~50,000])
備 考	

6.7 添付ソフトウェア

6.7.1 「動かしてみる」

「動かしてみる」プログラムは、motionCATシステムの試運転・動作確認用ソフトウェアです。
添付ソフトウェアディスクの「¥test¥Release¥t1mct00.exe」を実行して下さい。

【 「動かしてみる」実行上の注意事項 】

複数のモジュールを使用する場合は、1ライン上のモジュールIDは重複してはいけません。

(1) モジュール選択

使用するMCATのアドレスと接続されるモーションモジュール、DIOモジュールの個数を設定します。

「File」「Open」でデバイスをオープンし、システム通信を行い、モジュール数のチェックを行います。

その後、サイクリック通信を開始します。ここでモジュールIDを指定し、使用するモジュールを選択します。

モーションモジュールが選択された場合はモーションモジュール操作画面が開き、DIOモジュールが選択された場合はDIOモジュール操作画面が開きます。

このプログラムでは1度に使用できるMCATは一つですが、各々のMCATに接続されているすべてのモジュールを操作することができます。

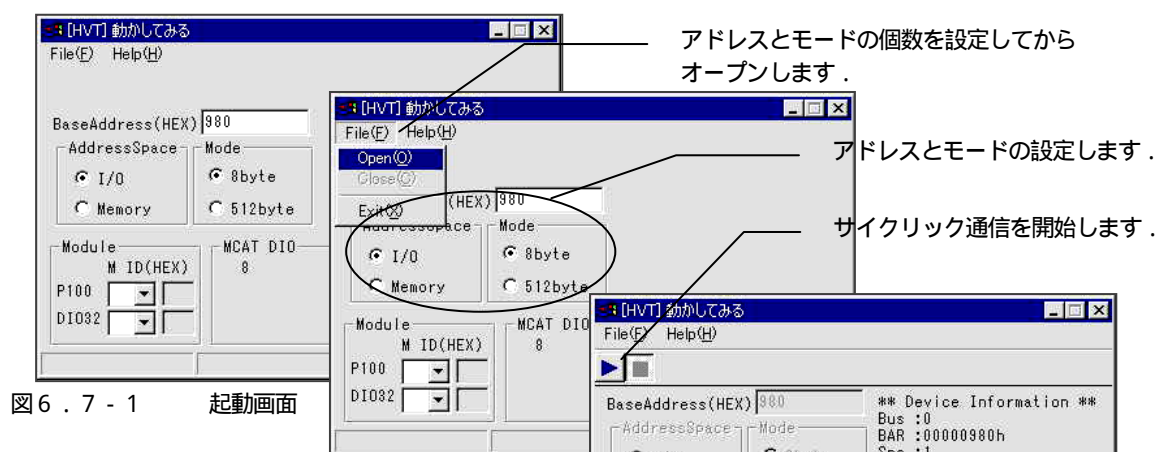
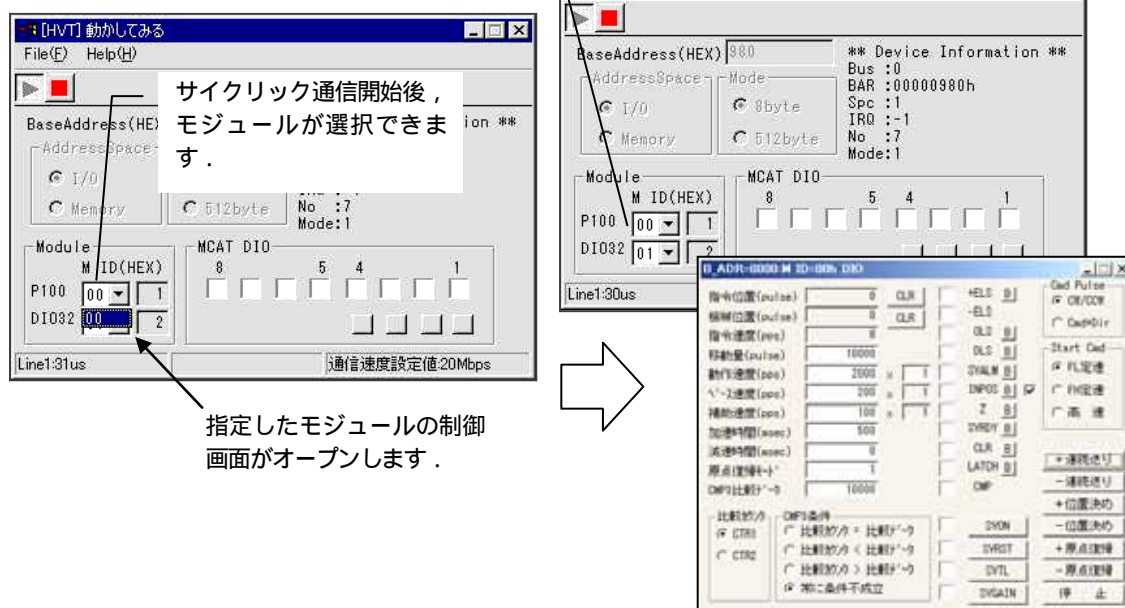


図 6.7 - 1 起動画面

オープンに成功するとMCATの汎用入出力が使用可となります。

サイクリック通信開始後、モジュールが選択できます。



(2) モーションモジュールの操作

機 能

「動かしてみる」のモーションモジュールの操作画面では以下の機能があります。

- (1) 1軸位置決め
- (2) ±連続送り
- (3) 原点復帰
- (4) サーボI/F出力(SVON, SVRST, SVTL, SVGAIN)
- (5) 指令カウンタ(CTR1)・機械カウンタ(CTR2)の表示
- (6) 指令速度の表示
- (7) マシンセンサー状態の表示(±ELS, OLS, DLS)
- (8) サーボI/F入力状態の表示(SVALM, INPOS, SVRDY, Z相)
- (9) サーボI/F出力状態の表示(SVON, SVRST, SVTL, SVGAIN)
- (10) サーボI/F出力(SVON, SVRST, SVTL, SVGAIN)
- (11) 各パラメータ設定
位置決め動作の移動量, 動作速度, ベース速度, 補助速度, 速度倍率, 加速時間, 減速時間,
CMP3比較データ, 原点復帰モード

モーションモジュール レジスタ初期値

「動かしてみる」では各軸の初期化は一部ソースプログラムで固定されています。

初期化の条件を変更して動作させたい場合には、ソースプログラムを変更してリビルドして下さい。

レジスタ	内 容	初期値	備 考
RMV	移動量	10000	10000パルス
RFL	ベース速度	200	200pps
RFH	動作速度	2000	2000pps
RUR	加速レート	1387	200pps 2000ppsの直線加減速時間 約500msec, 加速距離約550パルス
RMG	速度倍率	199	1倍
RFA	補助速度	200	200pps
RMD	動作モード	0x00000030	+方向連続動作, DLS, INPOS有効, 直線加減速, 減速開始点自動計算, FH補正ON
RENV1	環境設定1	0x00034004	指令パルス出力形式: 個別パルス, DLS, OLS, SVALM, INPOS, SVRDY入力極性: B接, DLSラッチしない, ELS, SVALM入力時即停止, 原点復帰完了時SVCTRCL自動出力, 異常停止時SVCTRCL自動出力しない, SVCTRCL出力パルス幅13msec, SVCTRCL出力後のパルス出力ディレイ時間104msec
RENV2	環境設定2	0x000004FF	エンコード入力4倍倍出力
RENV3	環境設定3	0x00700001	原点復帰モード1
RENV4	環境設定4	0x00040000	原点復帰時CTRリセット
RCMP3	CMP3比較データ	10000	RCMP3=CTR1でエバレータ出力
RIRQ	イベント設定	0x00000001	正常停止時

本表以外のレジスタは初期値0

表6.7-1 「動かしてみる」G9003レジスタ初期値

モーションモジュール操作画面

モーションモジュール選択すると以下の操作画面が表示されます。

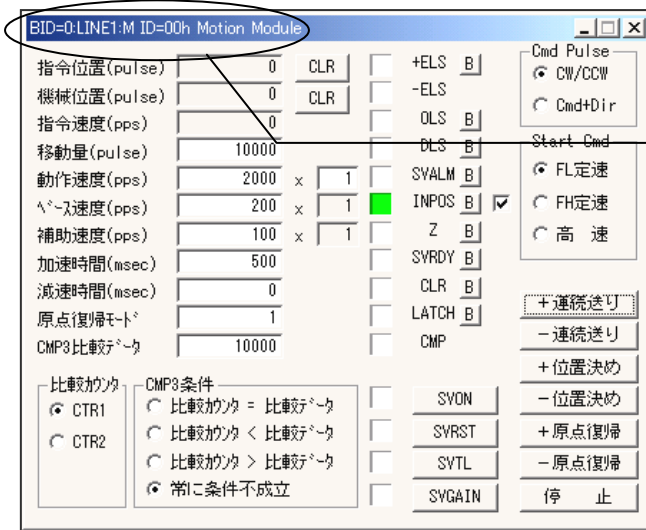


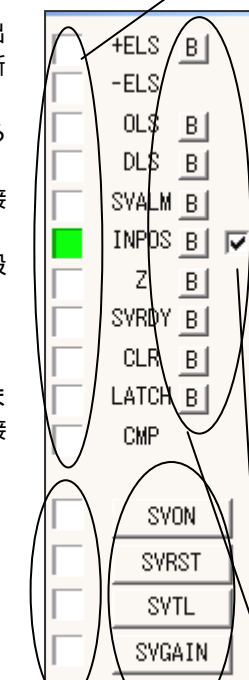
図 6. 7 - 2 モーションモジュール操作画面

信号入力極性設定と入力状態の表示

入出力状態の表示はレベル検出で約 110ms 周期で更新しています。
各端子が正しく接続されていることの確認をします。
未接続の端子の入力極性はA接に設定します。
使用される端子の入力極性を設定します。
INPOS有効で使用される場合はINPOS有効/無効のチェックBOXをチェックします。またサーボI/F出力の接続を確認します。

図 6. 7 - 3 信号入力状態

SVON : SVON出力ON/OFF
白: 出力なし, 緑: 出力中
SVRST : SVRST出力ON/OFF
白: 出力なし, 緑: 出力中
SVTL : SVTL出力ON/OFF
白: 出力なし, 緑: 出力中
SVGAIN : SVGAIN出力ON/OFF
白: 出力なし, 緑: 出力中



この操作画面が、どのモジュールが表示します。
左図の場合、
ボードID = 0のマスターボード、
ライン1に接続された
モジュールID = 00hのモジュール
です。
またサイクリック通信が停止している場合
「サイクリック通信停止中」と表示されます。

+ELS : +ELS入力状態
白: 入力なし, 赤: 入力有り
-ELS : -ELS入力状態
白: 入力なし, 赤: 入力有り
OLS : OLS入力状態
白: 入力なし, 緑: 入力有り
DLS : DLS入力状態
白: 入力なし, 緑: 入力有り
SVALM : SVALM入力状態
白: 入力なし, 赤: 入力有り
INPOS : INPOS入力状態
白: 入力なし, 緑: 入力有り
Z : EZ入力状態
白: 入力なし, 緑: 入力有り
SVRDY : SVRDY入力状態
白: 入力なし, 緑: 入力有り
CLR : CLRY入力状態
白: 入力なし, 緑: 入力有り
LATCH : LATCH入力状態
白: 入力なし, 緑: 入力有り
CMP : CMP条件成立状態
白: 条件不成立 緑: 条件成立

INPOS有効/無効チェックBOX
チェック: 有効, チェックなし: 無効

センサ入力状態及び入力極性切り替え
B... B接設定 (ノーマルクローズ)
A... A接設定 (ノーマルオープン)
クリックでA/B接反転

SVON : SVON出力ON/OFF
SVRST : SVRST出力ON/OFF
SVTL : SVTL出力ON/OFF
SVGAIN : SVGAIN出力ON/OFF

動作パラメータ設定

安全のため装置を動作させる時は最初は速度の低いところで動作を確認して下さい。

指令位置(pulse) 0 CLR
 機械位置(pulse) 0 CLR
 指令速度(pps) 0
 移動量(pulse) 1000
 動作速度(pps) 2000 x 1
 ベース速度(pps) 200 x 1
 補助速度(pps) 100 x 1
 加速時間(msec) 100
 減速時間(msec) 100
 原点復帰モード 1
 CMP3比較データ 10000

比較条件
☒ CTR1
☐ CTR2

CMP3条件
☐ 比較データ = 比較データ
☐ 比較データ < 比較データ
☐ 比較データ > 比較データ
☒ 常に条件不成立

指令位置・・・CTR1の値 指令パルス出力をカウント
 機械位置・・・CTR2の値 エンゲ F/Bを4倍でカウント
 指令速度・・・指令パルスの速度。(単位: PPS)
 移動量・・・位置決め動作の移動量(符号なし)。(単位:パルス)
 動作速度・・・RFHに設定される値。
 $RFH \times \text{速度倍率} = \text{動作速度}$
 ベース速度・・・RFLに設定される値。
 $RFL \times \text{速度倍率} = \text{ベース速度}$
 補助速度・・・原点モード1, 4, 6, 7で使用。
 $RFA \times \text{速度倍率} = \text{補助速度}$
 速度倍率・・・速度レジスタ(RFH, RFL, RFA)の値にこの値をかけた速度が実速度。
 加速時間・・・加減速スタート時のベース速度から動作速度までの加速時間。
 減速時間・・・加減速スタート時の動作速度からベース速度までの減速時間。
 原点復帰モード・・・原点復帰モードを指定。
 CMP3比較データ・コンパレータ比較データを設定。
 CMP3比較条件・コンパレータ比較条件を設定。
 比較データ選択・・・コンパレータ比較データはCTR1(指令位置)またはCTR2(機械位置)を選択。

図6.7-4 動作パラメータ設定

動作開始ボタン

Cmd Pulse
☒ CW/CCW
☐ Cmd+Dir

Start Cmd
☒ FL定速
☐ FH定速
☐ 高速

+連続送り
 -連続送り
 +位置決め
 -位置決め
 +原点復帰
 -原点復帰
 停止

指令パルス出力形式選択・・・指令パルスの出力形式を選択
 接続されるドライバのパルス入力形式と合わせます。
 スタートコマンド選択・・・FL定速, FH定速, 高速スタート(加減速付)を選択
 +連続送り・・・設定された速度で+方向へ動作。
 停止ボタンまたは+E L S, SVALM入力で停止
 -連続送り・・・設定された速度で-方向へ動作。
 停止ボタンまたは-E L S, SVALM入力で停止
 +位置決め動作・・・設定された速度, 移動量で+方向位置決め。
 停止ボタンまたは+E L S, SVALM入力で停止
 +位置決め動作・・・設定された速度, 移動量で-方向位置決め。
 停止ボタンまたは-E L S, SVALM入力で停止
 +方向原点復帰・・・設定された速度, 移動量で+方向へ原点復帰。
 原点復帰完了, 停止ボタン, または+E L S, SVALM入力
 で停止。
 原点復帰完了時にSVCTRCL出力とCTR1,CTR2リセット
 -方向原点復帰・・・設定された速度, 移動量で-方向へ原点復帰。
 原点復帰完了, 停止ボタン, または-E L S, SVALM入力
 で停止。
 原点復帰完了時にSVCTRCL出力とCTR1,CTR2リセット
 停止・・・加減速動作減速停止
 FL定速, FH定速動作時は即停止

図6.7-5 動作開始ボタン

(3) D I Oモジュールの操作

(1) 機 能

指定したD I Oモジュールの入力状態の表示 (I N 1 ~ I N 1 6)

指定したD I Oモジュールの出力状態の表示 (O U T 1 ~ O U T 1 6)

指定したD I Oモジュールの出力のO N / O F F (O U T 1 ~ O U T 1 6)

(2) D I Oモジュール操作画面

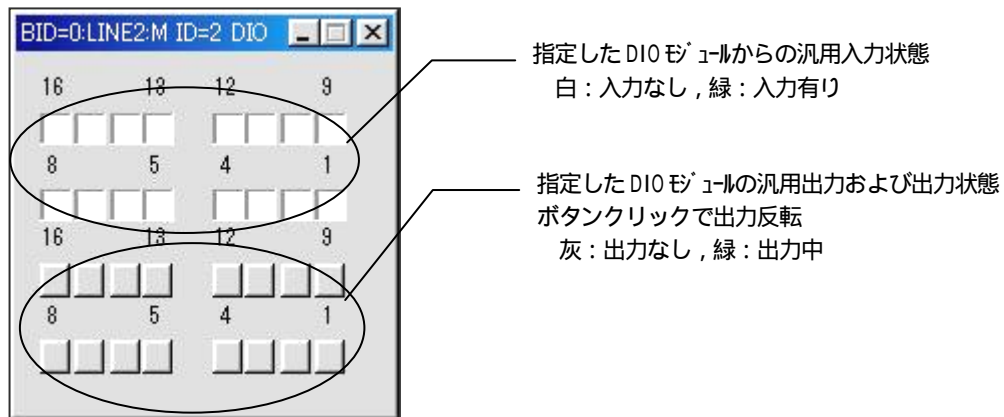


図 6 . 7 - 6 D I Oモジュール操作画面

6.8 DOS版添付ソフトウェア

6.8.1 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
- ドライバー関数の仕様
- 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。

6.8.2 ソフトウェア構成

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motionCATマスターボードをつなぐ入出力関数をドライバー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。

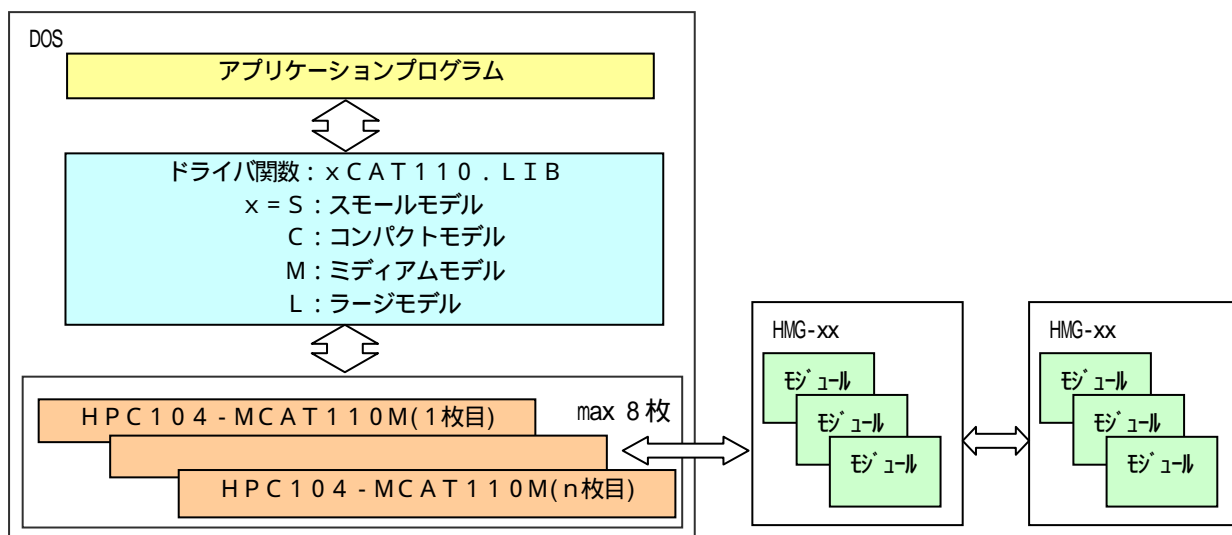


図6.8-1 ソフトウェアの関連図

6.8.3 デバイスドライバー

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ー関数内に含ませたため、デバイスドライバーのインストールは不要です。

6.8.4 割り込み

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4.1節「センターデバイス」を参照して下さい)

CMSTSのビット2:IOPCは、「入力ポート変化フラグ 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(ビットに対応したポートの状態が変化するとき)に集約的にセットされ、一つの割り込みソースになります。また、CMSTSのビット3:EIOEは、サイクリック通信エラーがモジュールのいずれかで発生し、「サイクリック通信エラーフラグ IOERR」が1にセットされた場合に集約的にセットされ、1つの割り込みソースになります。

これらはCEND, BRKF, EDTE, ERAE, CAERの各割り込みソースとORされ、1つにまとめられて出力されます。

マスターボードからのISA Busへの割り込み出力は、ボード割り込み(オプションポート INTE)がイネーブルである場合にジャンパーで設定されたIRQ番号でCPUに割り込みます。割り込みを使用しない場合はINTEをマスクしたままCMSTSのポーリングで割り込み要因を取り出して処理します。

要因毎の割り込み許可/禁止は環境レジスタ0(RENV0) bit 0 ~ bit 6で行います。

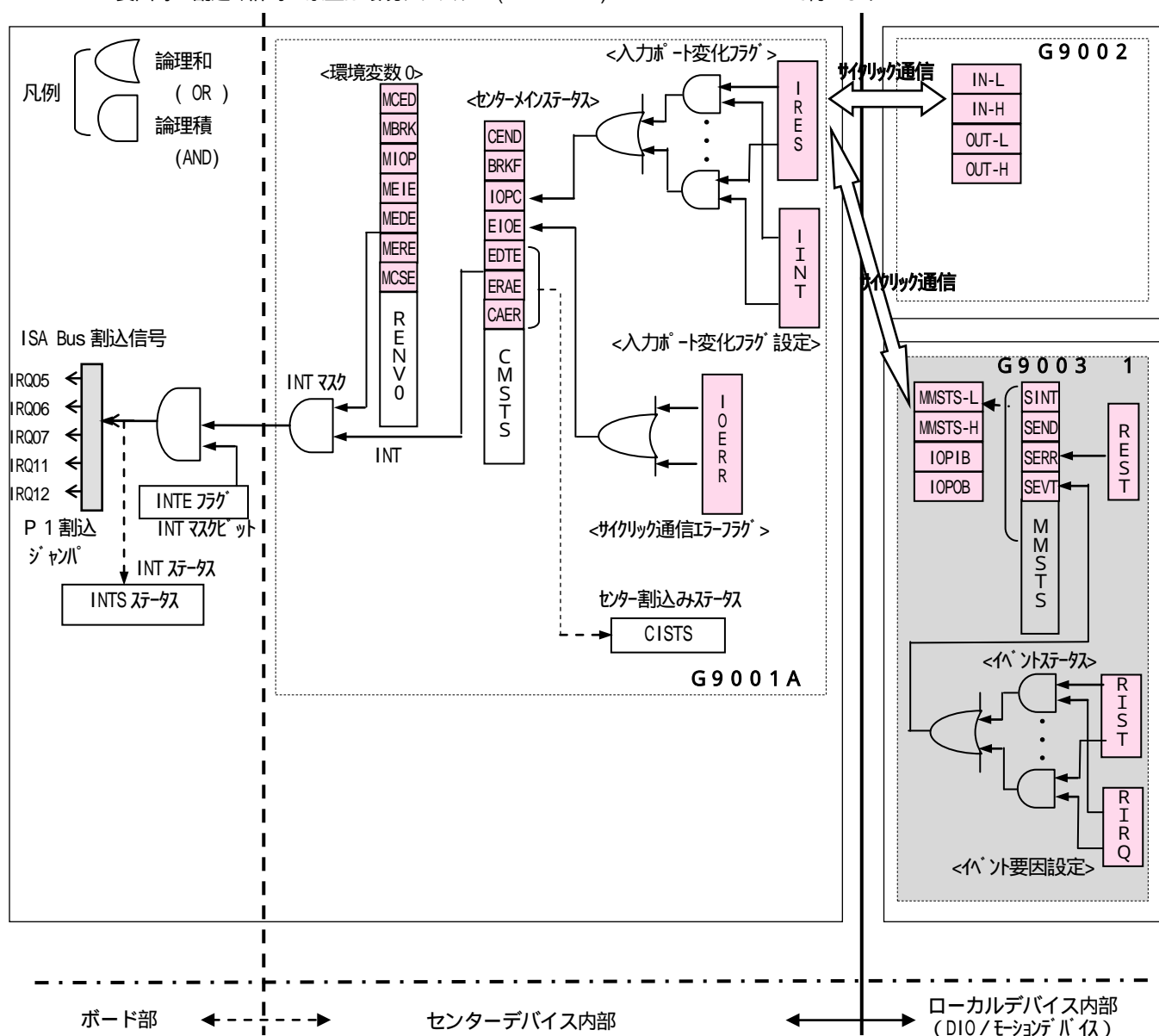


図6.8-2 ボード内の割り込みルート

- 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。

6.8.5 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。

初期化

(1) マスターボード (G9001A) の初期化

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。ドライバー関数に含まれるMCATオープン関数「mct110_OpenMstBrd ()」を使用します。

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: MIOPI

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。

入力ポート変化フラグ (IINT) 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。

割り込みイネーブル

割り込みイネーブルポート (INTE) 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6.8-3 初期化手順フロー

運用

M C A T の割り込みは、複数の発生要因が 1 つの割り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
	C E N D		RENVO Bit9 の設定に 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 の場合はリセットビットをクリアメントによりリセット
	B R K F (不使用)		
	E D T E		
	I O P C	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ビットに対応するビットに 1 を書き込んでリセット
	E I O E	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ビットに対応するビットに 1 を書き込んでリセット
	E R A E	セクタ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 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 の場合はリセットビットをクリアメントによりリセット
	C A E R		

表 6 . 8 - 1 エラー発生要因による参照とリセットの対応

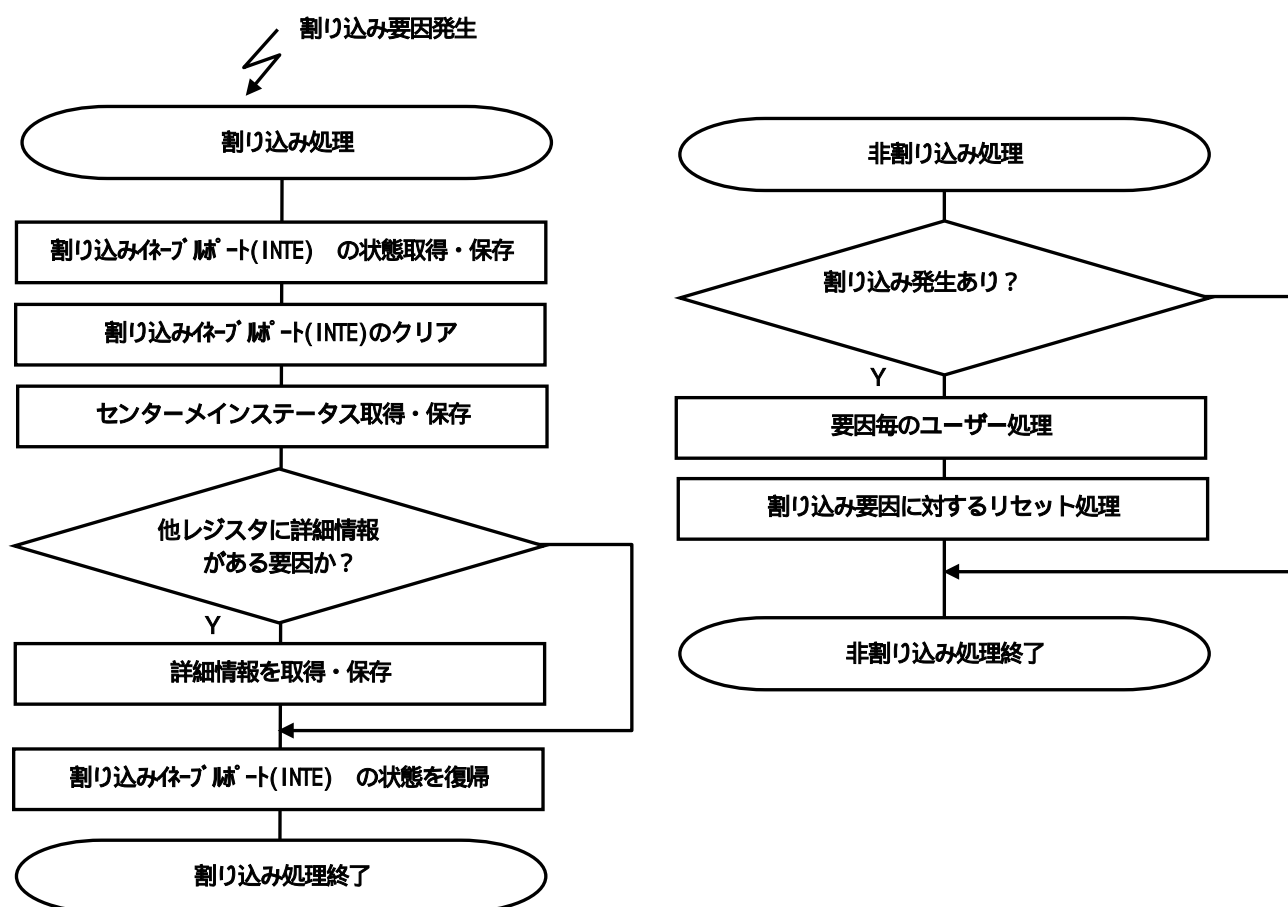


図 6 . 8 - 4 割り込み発生時の処理フロー

6.8.6 DOS用ドライバ関数のデータ型と構造体の定義

データ型

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

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 HMCT110_INFO 型構造体を、DOS 版ではボード枚数最大 8 枚として使用枚数分用意します。

```
typedef struct {
    U_LONG    adrs; /* 1: 0: ページアドレス (IOマップ時 16bit / MEMマップ時: 32bit) */
    Short     maps; /* 2: 4: アドレス空間 ( 0: IO / 1: MEM ) */
    Short     bussz; /* 3: 6: バス幅 ( 0: 8bit / 1: 16bit ) */
    Short     acsmode; /* 4: 8: 使用空間モード ( 0: 8バイトモード / 1: 512バイトモード ) */
    Short     intno; /* 5: 10: 割り込み番号 ( 0: なし / 5, 6, 7, 11, 12: IRQ 番号 ) */
} HMCT110_INFO;
```

6.8.7 ドライバ関数の戻り値

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D SW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_ADDRESS	00000040	不正なアドレス I/OマップドI/O時、0から0×1 f f fのアドレスを使用した メモリーマップドI/O時、0から0×9 f f f以下のアドレスまたは0 X F F F E 0 0以上のアドレスを使用した 8バイトモード使用時、アドレス下位5ビットに0以外を指定した 5 1 2バイトモード使用時、アドレス下位9ビットに0以外を指定した
6	ILL_IRQNUM	00000080	不正なI R Q番号 I R Q 5 , 6 , 7 , 1 1 , 1 2以外を指定した 指定したI R Q番号が既に使用されている
7	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
8	ILL_CPU_MODE	00002000	C P Uモード不正 WindowsまたはWindows上のDOS窓で実行した E M M 3 8 6 . E X Eを使用している
9	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。
10	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
11	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
12	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
13	DATA_COM_ERROR	00040000	データ通信エラー
14	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
15	MODULE_OVERCOUNT	00100000	1ラインに接続されているモジュールの数が3 3個以上 1ラインに接続できるモジュールの数は3 2個以下です。
16	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
17	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
18	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 6.8-3 ドライバ関数の戻り値

6.8.8 ドライバ関数仕様

DOS用ライブラリの関数仕様は以下になります。Windows用ドライバ関数とは、戻り値、引数のデータ型が多少異なります。

(1) mct110_GetDrvVerNo() MCATライブラリバージョンの取得		
機 能		現在システムに使用されているライブラリーのバージョン情報を取得します。
DOS	書 式	<code>void mct110_GetDrvVerNo (short* ver);</code>
	引 数	<code>short* ver</code> ・・・取得したライブラリバージョンの格納アドレス ライブラリは2つのライブラリ(メモリアクセライブラリおよびホスト I/Fライブラリ)が結合して構成されておりバージョン番号取得では下位1バイトにメモリアクセライブラリ、上位1バイトにホスト I/Fライブラリのバージョン番号がそれぞれ格納されます。(1バイトの構成は上位4bitがジャー、下位4bitがマイバージョン)
	戻り値	'0'で正常、'0'以外は「6.8.6 ドライバ関数の戻り値」参照

(2) mct110_OpenMstBrd() MCATのオープン		
機 能		MCATをオープンし、このMNTのデバイスハンドルを取得します。
DOS	書 式	<code>U_LONG mct110_OpenMstBrd (short* hDev, HMCT110_INFO* hInfo);</code>
	引 数	<code>short* hDev</code> ・・・取得したデバイスハンドルの格納アドレス <code>HMCT110_INFO* hInfo</code> ・・・オープンするMCATのデバイス情報格納先頭アドレス
	戻り値	'0'で正常、'0'以外は「6.8.6 ドライバ関数の戻り値」参照

(3) mct110_CloseMstBrd() MCATのクローズ		
機 能		MCATをクローズします。
DOS	書 式	<code>U_LONG mct110_CloseMstBrd (short hDev);</code>
	引 数	<code>short hDev</code> ・・・デバイスハンドル
	戻り値	'0'で正常、'0'以外は「6.8.6 ドライバ関数の戻り値」参照

(4) mct110_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	<code>U_LONG mct110_rCenMsts (short hDev, short wLine, U_SHORT* wSts);</code>
	引 数	<code>short hDev</code> ・・・デバイスハンドル
		<code>short wLine</code> ・・・予約(常に0を設定してください)
		<code>U_SHORT* wSts</code> ・・・センターメインステータス格納アドレス
	戻り値	'0'で正常、'0'以外は「6.8.6 ドライバ関数の戻り値」参照

(5) mct110_wCenCmd() センターデバイスへのコマンド書き込み		
機 能		センターデバイスへコマンドを書込みます。
DOS	書 式	<code>U_LONG mct110_wCenCmd (short hDev, short wLine, U_SHORT wCmd);</code>
	引 数	<code>short hDev</code> ・・・デバイスハンドル
		<code>short wLine</code> ・・・予約(常に0を設定してください)
		<code>U_SHORT wCmd</code> ・・・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「6.8.6 ドライバ関数の戻り値」参照

(6) mct110_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
	引 数
	戻り値

(7) mct110_rCenBuf () センターデバイスの入出力バッファ読出し mct110_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
	引 数
	戻り値

(8) mct110_rCenRFifo () センターデバイスの受信用 FIFO 読出し mct110_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
	引 数
	戻り値

(9) mct110_rCenReg () センターデバイスのレジスタ読出し mct110_wCenReg () センターデバイスのレジスタ書込み	
機 能	センターデバイスのレジスタから読出します。 センターデバイスのレジスタへ書込みます。
DOS	書 式
	引 数
	戻り値
	注意 事項

(15) mct110_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 mct110_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 mct110_rPclPort(short hDev,short wLine, short wMid,U_CHAR* byData); U_LONG mct110_wPclPort(short hDev,short wLine, short wMid,U_CHAR* byData);
	引 数 short hDev .. デバイスハンドル short wLine .. 予約 (常に0を設定してください) 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「6.8.6 ドライバ関数の戻り値」参照

(16) mct110_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 mct110_rPclMsts (short hDev,short wLine,short wMid,U_SHORT* wSts);
	引 数 short hDev .. デバイスハンドル short wLine .. 予約 (常に0を設定してください) 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「6.8.6 ドライバ関数の戻り値」参照

(17) mct110_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 mct110_wPclCmd (short hDev,short wLine,short wMid,U_SHORT wCmd);
	引 数 short hDev .. デバイスハンドル short wLine .. 予約 (常に0を設定してください) 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「6.8.6 ドライバ関数の戻り値」参照

(18) mct110_rPclReg () モーションデバイスレジスタからの読出し mct110_wPclReg () モーションデバイスレジスタへの書込み	
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込みます。
DOS	書 式 U_LONG mct110_rPclReg(short hDev,short wLine, short wMid,U_SHORTwCmd,U_LONG* dwReg); U_LONG mct110_wPclReg(short hDev,short wLine, short wMid,U_SHORTwCmd,U_LONG dwReg);
	引 数 short hDev .. デバイスハンドル short wLine .. 予約 (常に0を設定してください) 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 .. レジスタへ書込むレジスタデータ
	戻り値 '0'で正常, '0'以外は「6.8.6 ドライバ関数の戻り値」参照

(19) mct110_rOptPortB () オプションポート1バイト読出し mct110_rOptPortW () オプションポート2バイト読出し mct110_wOptPortB () オプションポート1バイト書込み mct110_wOptPortW () オプションポート2バイト書込み	
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込みます。
DOS	書 式 U_LONG mct110_rOptPortB (short hDev, U_SHORT wAdrs, U_CHAR* byData); U_LONG mct110_rOptPortW (short hDev, U_SHORT wAdrs, U_SHORT* wData); U_LONG mct110_wOptPortB (short hDev, U_SHORT wAdrs, U_CHAR byData); U_LONG mct110_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 ・ 読出しデータ格納アドレス (1バイト) U_SHORT* wData ・ 読出しデータ格納アドレス (2バイト) U_CHAR byData ・ 書込みデータ (1バイト) U_SHORT wData ・ 書込みデータ (2バイト)
	戻り値 ' 0 'で正常, ' 0 '以外は「 6 . 8 . 6 ドライバ関数の戻り値」参照

(20) mct110_rCenPortW () センターデバイス指定ポート2バイト読出し mct110_wCenPortW () センターデバイス指定ポート2バイト書込み	
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込みます。
DOS	書 式 U_LONG mct110_rCenPortW (short hDev, U_SHORT wAdrs, U_SHORT* wData); U_LONG mct110_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 ・ 読出しデータ格納アドレス (2バイト) U_SHORT wData ・ 書込みデータ (2バイト)
	戻り値 ' 0 'で正常, ' 0 '以外は「 6 . 8 . 6 ドライバ関数の戻り値」参照

6 . 8 . 9 ドライバ関数を割込み環境で使用する際の注意事項

- (1) ドライバ関数内では、実行時に以下条件下では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)
- ・ データバス幅を 8 ビットで使用した場合
 - ・ F I F O に対してアクセスする場合
 - ・ コマンド送信によりアクセスする場合 (センターデバイス, ローカルデバイス)
- (2) 関数 mct110_wPciCmd (No. 17) , mct110_rPciReg , mct110_wPciReg (No. 18) は、前述したドライバ関数のいくつかを組み合わせで作られており、通信完了待ち、タイムアウト監視などの処理も含んでいます。そのため利用するアプリケーション側の構造・構成によってはそのままご利用できない場合があります。そのような場合が想定される場合は、ユーザ様にてドライバ関数を組み合わせ、代替の関数を作成して頂く必要があります。
(これら関数のサンプルソースコードが必要な場合は弊社までお問い合わせください)

6 . 8 . 10 ライブラリ関数

ドライバ関数を使用して作られたライブラリー関数 (Windows 版での H p x _ x x x 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 版ライブラリー関数の大半は、DOS 版で使用が可能ですが、十分検証を行ったうえでご使用下さい。

6.8.1.1 DOS版サンプルプログラム

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。

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。

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」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。

(1) 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。

CLK.BAT	・ ・ 実行ファイル作成用バッチファイル(MS-C V60)
SMP110M.EXE	・ ・ サンプルプログラム実行ファイル
SMP110M.C	・ ・ サンプルプログラム・メインソース
SMPPXX.C	・ ・ P x x x サンプルプログラムソース
SMPDIO.C	・ ・ D I O x x サンプルプログラムソース
HC110LIB.H	・ ・ ドライバ関数参照用ヘッダーファイル
	“ SMP110M.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
	“ SMP110M.C, SMPPXX.C, SMPDIO.C で “#include ”
LCAT110.LIB	・ ・ リンク用ドライバ関数(ラージモデル用)

(2) 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。またアドレス空間をメモリーマップドI/Oで使用する場合は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。

(3) 起動画面

実行ファイル“ Smp110m.exe ” を起動した時，下記画面が表示されます．

```
*** HPC104-MCAT110M : DOS(16) Device Driver Sample ***
----- Lcat110.lib Ver1.0 (Mem:Ver1.0) -----

Set Address Space [0:I/O 1:MEM] (Default:0) =
```

画面上で操作対象ボードの以下アクセス方法の設定を行います。

- ・ アドレス空間選択 ・ ・ I / O マップまたはメモリーマップ I O の選択をします。
- ・ バスサイズ選択 ・ ・ 8 ビットまたは 16 ビットデータバスの選択をします。
- ・ モード選択 ・ ・ 8 バイトまたは 512 バイトの G9001 へのアクセスモード選択をします。
- ・ 割込 (IRQ) 選択 ・ ・ 割込み使用 / 不使用の選択。割込みは画面に表示された IRQ 5, 6, 7
11(b), 12(c) のいずれかで、ユーザーに解放された割込みのみ使用できます。

```

*** HPC104-MCAT110M : DOS(16) Device Driver Sample ***
----- Lcat110.lib Ver1.0 (Mem:Ver1.0) -----
      SPA:I/O  ADRS:0x000980  BUS:8bit  MODE:8byte  IRQ:--
Set Address Space [0:I/O 1:MEM] (Default:0) =
Set Data Bus Size [0:8bit 1:16bit] (Default:0) =
Set G9001 Access Mode [0:8byte 1:512byte] (Default:0) =
Set Board Address (Default:0x000980) =
Int. IRQ Select Slave[...cb...] Master[765.....] : IRQ No.=

```

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および設定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>
```

(4) 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。

```

*** HPC104-MCAT110M : DOS(16) Device Driver Sample ***
----- Lcat110.lib Ver1.0 (Mem:Ver1.0) -----
ID:0000 SPA:I/O ADRS:0x000980 BUS:8bit MODE:8byte IRQ:--

Slave Module Line1: 4 [ DI032: 1 / P100: 3 ] Connect ISTS=00*

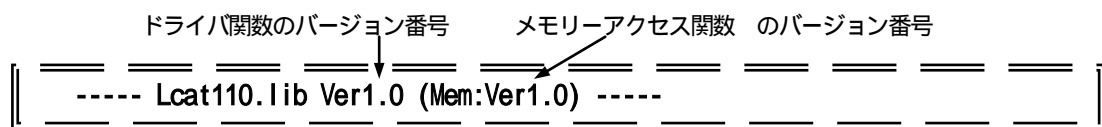
[ Master Board ]

Line1
Main Status 0001x000x0000000( 0x1000 )
Int. Status 00000000x000000( 0x0000 )
General DIO
  in 8-1 00000000( 0x00 )
  out 8-1 xxxx0000( 0x00 )

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

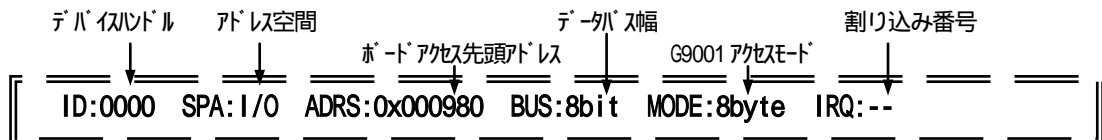
```

ソフトのバージョン番号



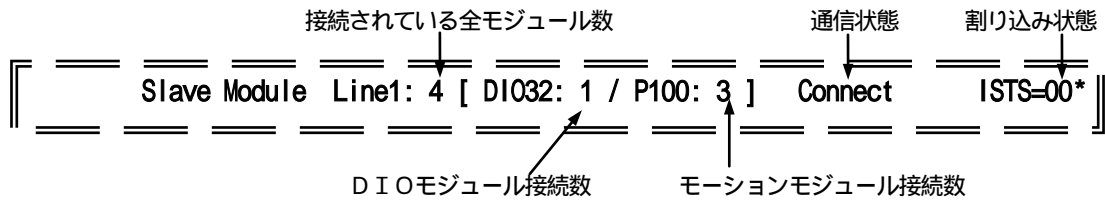
ドライバ関数は、メモリーマッピング使用時に使用するメモリーアクセスの為の関数を結合して構成しています。メモリーアクセス関数とはこのメモリーアクセス関数の部分を指します。

ボードアクセス情報



- ・ 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。
- ・ 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。
SPA : アドレス空間
ADRS : ボードアクセス先頭アドレス
BUS : データバス幅
DATA : G9001アクセスモード
IRQ : 割り込みIRQ番号。' 'は割り込み不使用の表示

接続モジュール情報



・モジュール数

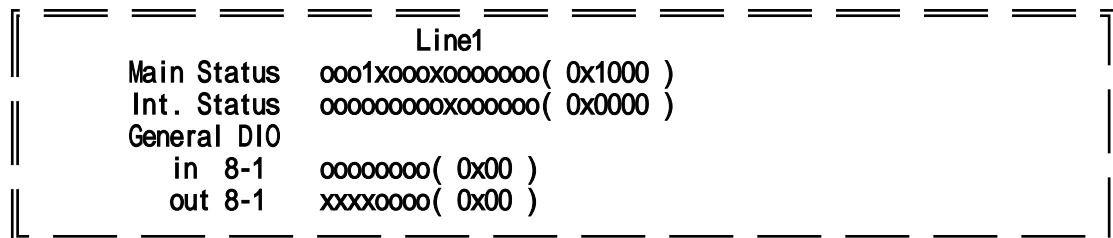
接続されているモジュール数が表示されます。通信状態が Disconnect の場合は “--” で表示されます。ライン上にモジュールが 1 つも接続されていない場合も同様の表示になります。

・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。

マスターボード状態 (選択ボードのみ)



- ・ 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
- ・ General DIO: マスターボード汎用入出力 入力 = 8 ビット, 出力 = 4 ビット表示

ステータス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。

	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名 称	---	CAER	ERAE	EDTE	EIOE	IOPC	BRKF	CEND
Main	1 表示	?	1	1	1	1	1	1	1
	0 表示	X	0	0	0	0	0	0	0
	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	0	0	0	0	X	0	0	0
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	0	0	0	0	0	0
	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	1
	0 表示	0	0	0	0	0	0	0	0

D I O 入出力状態は次の表記で表されます。

D I	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名 称	in8	in7	in6	in5	in4	in3	in2	in1
	1 表示	1	1	1	1	1	1	1	1
D O	0 表示	0	0	0	0	0	0	0	0
	ビット値	b7	b6	b5	b4	b3	b2	b1	b0
	名 称	----	----	----	----	out4	out3	out2	out1
	1 表示	?	?	?	?	1	1	1	1
	0 表示	X	X	X	X	0	0	0	0

操作メインメニュー

```

[ 0:End  1:CONNECT  2:MASTER DIO  3:DIO  4: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： サイクリック通信開始 / 停止を行います。
- ・ "2"キー： マスターボード上D I Oの出力操作を行います。
- ・ "3"キー： 指定D I OモジュールのD I O入出力状態表示と出力操作を行います。
- ・ "4"キー： 指定モーションモジュールの各種操作を行います。

(5) サイクリック通信開始 / 停止

各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
Disconnect?( Y,N ) = _

```

- ・ 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場合は通信を開始できません。

(6) マスターボードD I O操作

マスターボード上のD I Oの状態を表示、またD Oの出力操作をします。

```

General DIO
  in  8-1  00000000( 0x00 )
  out 8-1  xxxxo1o1( 0x05 )

DIO Output( Select: , ,0-7 Fin.:ESC Invert:0thers )

```

D O出力操作を以下のキーを使って行います。

- ・ 矢印キー (,)： 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。
D Iは0 ~ 3, D Oは0 ~ 7が有効
- ・ E S C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。

(7) DIOモジュール

DIO32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。

DIOモジュール選択画面

```
*** HPC104-MCAT110M : DOS(16) Device Driver Sample ***
----- Lcat110.lib Ver1.0 (Mem:Ver1.0) -----
ID:0000 SPA:I/O ADRS:0x000980 BUS:8bit MODE:8byte IRQ:--

      Slave Module Line1: 4 [ DI032: 1 / P100: 3 ]   Connect      ISTS=00*

[ DIO Module ]

      Port   I/O           Status

Valid Module MID = 03

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

- ・ Select Line: 操作ライン番号 (1or2) 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。

出力操作画面

```
=====
[ DIO Module ] MID:03 Intb: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 )
=====
```

DIO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。

(8) モーションモジュール

P100モーションモジュールの操作および設定を行います。

モーションモジュール選択画面

```
*** HPC104-MCAT110M : DOS(16) Device Driver Sample ***
----- Lcat110.lib Ver1.0 (Mem:Ver1.0) -----
ID:0000 SPA:I/O ADRS:0x000980 BUS:8bit MODE:8byte IRQ:--

Slave Module Line1: 4 [ DI032: 1 / P100: 3 ] Connect ISTS=00*

[ Motion Module ]

Valid Module MID = 01 02 03

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

- ・ Select Line: 操作ライン番号 (1or2) 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。

操作メイン画面

```
=====
[ Motion Module ] MID:03 Intb: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 00000000( 0x00 )
ExsLH 00000111( 0x07 ) ===== 3.Register Disp & set =====
ExsHL 00000000( 0x00 ) 0:BasS 1:MovS 2:Acc 3:Mult 4:Mode
ExsHH xxxxxxxx( 0x00 ) 5:AuxS 6:RENV1 7:RENV2 8:RENV3 1:INT
IOPB  00000000( 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。

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

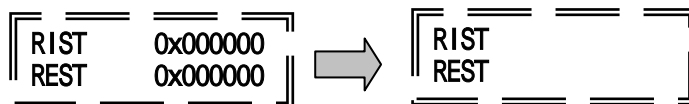
ステータスは各ラインについて状態を次の表現で表示します。表示は英数字または記号1文字で表されます。
なお、ステータス右隣に表示される()内は、ステータスの状態を16進数値で表記した値です。

M s s L	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	0	0	0	0
M s s H	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	0	0	0	x	x	x	0
E x s L 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	0	0	0	0	0	0	0	0
E x s L 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	0	0	0	0	0	0	0	0
E x s H 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	0	0	0	0	0	0	0	0
E x s H 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	0	0	0	0
I O P B	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	0	0	0	0	0	0	0	0

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。



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。

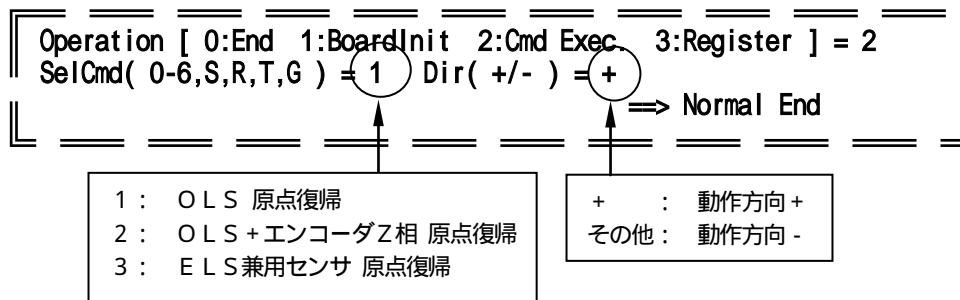
- ・ ソフトウェアリセット(オプションポートの状態はリセットされません)
- ・ レジスタへ初期値書き込み

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（'+' 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。



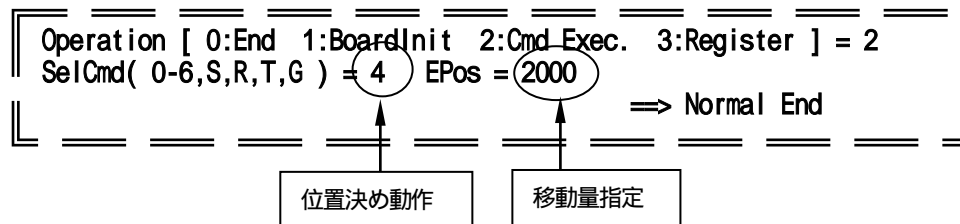
“Normal End” の表示はモーションモジュールに正常にコマンドを送れた事を示すものであり、完了を示すものではありません。（以降同様）

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）

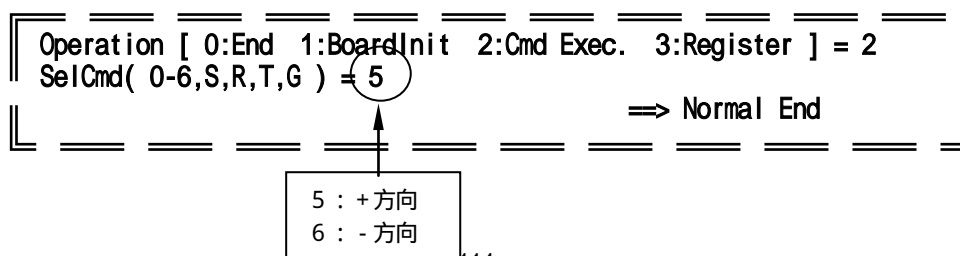
位置決め動作中に手動停止（キー押下）、異常検出、移動方向のリミットを検出すると動作を中断して停止します。



連続運転

連続運転を行います。

- ・ + 方向連続
+ 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。
- ・ - 方向連続
- 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き込みが行われます。

選択表示	キー	変更内容	選択表示	キー入力	変更内容
0 : B a s e	0	ベース速度	I : I N T	I	イベントマスク
1 : M o v S	1	動作速度	D : D L S	D	D L S 入力極性反転
2 : A c c	2	加速レート	O : O L S	O	O L S 入力極性反転
3 : M u l t	3	速度倍率	E : E L S	E	E L S 入力極性反転
4 : M o d e	4	動作モード	A : A L M	A	A L M 入力極性反転
5 : A u x S	5	移動量補正速度			
6 : R E N V 1	6	環境1			
7 : R E N V 2	7	環境2			
8 : R E N V 3	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   xxxx0000( 0x00 )
=====

```

```

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

```

⇒ ELS N.C

E L S 検出から検出なしの状態に変化

```

=====
ExsLL  00000000( 0x00 )
ExsLH   00000111( 0x07 )
ExsHL   00000000( 0x00 )
ExsHH   xxxx0000( 0x00 )
=====

```

7. モーションモジュールの運用

この節ではモーションモジュールの以下の説明をします。

プログラム手順

初期設定

原点センサ構成と原点復帰方法

速度パターン設定

動作モード

ステータス、レジスタ、オプションポート、コマンド

7.1 プログラム手順

プログラムの基本手順は概略 右図のフローに示す構成です。

以下この手順に従って解説します。

7.1.1 マスターボード初期設定

プログラムの始めはセンターデバイスの初期化、システム通信によるローカルデバイス情報取得、サイクリック通信開始で始めてモーションデバイスに対しデータ通信可能となります。

(1) ボード毎にデバイスオープン

(2) センターデバイス初期設定

ボード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。

ローカルデバイス(DIO, モーション)の情報取得

センターデバイスへのコマンド書込み

センターメインステータス読出し

ローカルデバイスのポート入力変化割込設定

モーションデバイスのポート入力変化割込設定

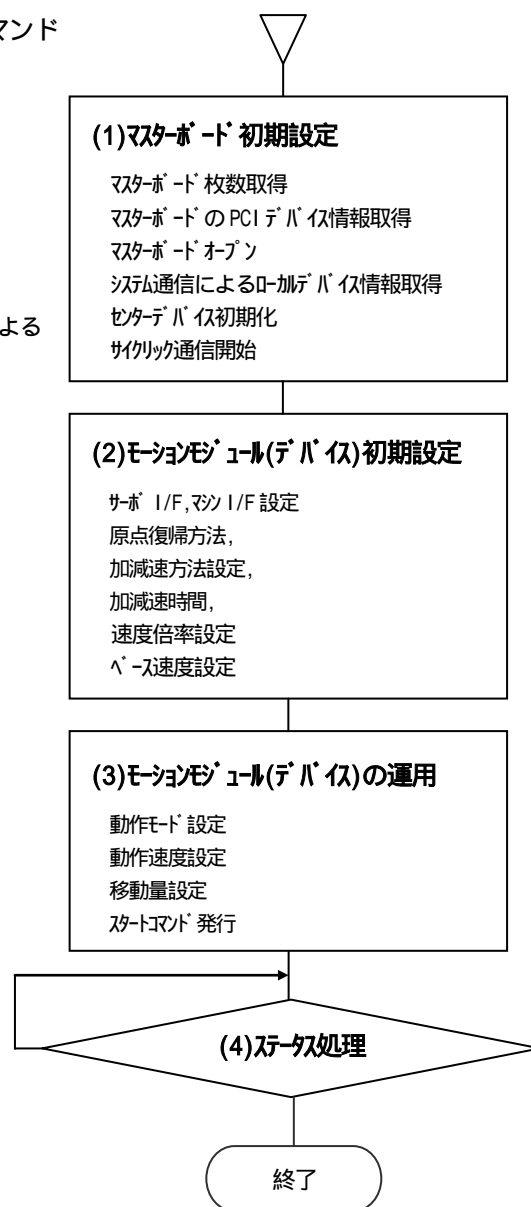


図 7.1 - 1 プログラム基本手順

7.2 モーションモジュール(デバイス)初期設定

プログラム起動後，センタデバイス初期化が終了し，サイクリック通信開始し，データ通信が可能になった後，各モーションデバイスに対して，初期設定を行います．

設定箇所は環境設定レジスタ RENV1～RENV6，動作モードレジスタの一部，イベント要因設定レジスタ，汎用出力ポートです．

以下の関数を用いてマシンセンサー入力仕様設定，サーボI/F設定，原点復帰方法設定，イベント要因設定を行います．

この節では各種初期設定について，次の項目にしたがって解説します．

1	マシンセンサー入力仕様設定	±ELS入力 DLS入力 OLS入力
2	サーボインターフェース設定	指令パルス出力形式 エンコーダA相/B相入力形式 サーボアラーム入力 インポジション入力 サーボレディ入力 サーボ偏差カウンタクリア出力 サーボオン，サーボリセット，サーボトルク，サーボゲイン出力
3	原点復帰方法設定	原点復帰方法 エンコーダZ相入力仕様
4	イベント要因の設定	<input type="checkbox"/> イベント要因設定

表7.2-1 モーションデバイス初期設定項目

7.2.1 マシンセンサー入力仕様設定

マシンセンサー入力仕様の初期化に関する項目は次表の通りです．

No.	項目	信号名	設定箇所等	
1	ELS 入力仕様	±ELS	1. 入力極性 RENV2:b7-0 の設定後有効(RENV2:b7-0 設定:11111111)	IOPOB:ELL (b7)
			B 接(加電流 OFF で検出)	1
			A 接(加電流 ON で検出)	0
			ELS 不使用時は A 接(IOPOB:ELL=0)，端子は接続しない	
			2. ELSon 時の動作	RENV1:b3
			即停止	0
2	DLS 入力仕様	DLS	減速停止	1
			1. DLS 有効/無効	RMD:b8
			DLS 入力を見逃す(RSTS での入力状態の確認は可)	0
			DLSon で減速(減速停止)する	1
			2. 入力極性	RENV1:b6
			B 接(加電流 OFF で検出)	0
			A 接(加電流 ON で検出)	1
			DLS 不使用時は A 接(RENV1:b6=1,RMD:b8=0)，端子は接続しない	
			3. DLSon 時の動作	RENV1:b4
			減速のみ	0
			減速停止	1
			4. DLS 入力ラッチ(DLS 信号幅が短い場合使用)	RENV1:b5
3	OLS 入力極性	OLS	ラッチしない	0
			ラッチする	1
			1. 入力極性	RENV1:b7
			B 接(加電流 OFF で検出)	0
			A 接(加電流 ON で検出)	1
			OLS 不使用時は A 接(RENV1:b7=1)，端子は接続しない	

表7.2-2 マシンセンサー入力仕様設定

7.2.2 サーボインターフェース設定

サーボインターフェースの初期化に関する項目は次表の通りです。

No.	項 目	信号名	設定箇所等	
1	指令パルス出力形式	CWP, CWN CCWP, CCWN	1. 指令パルス出力形式	RENV1:b0-b2
			個別指令パルスモード (CW/CCW)	100
			共通指令パルスモード (共通パルス+DIR 信号)	000
2	エンコーダ A 相/B 相 入力形式	AP, AN BP, BN	1. 通信設定	RENV2:b10,9
			1 通信	00
			2 通信	01
			4 通信	10
			UP/DOWN	11
			2. 入力禁止 (一部のサボドライバ使用時は原点復帰完了まで禁止)	RENV2:b18,17
3	サボアラーム 入力仕様	SVALM	入力許可	00
			入力禁止	11
			1. 入力極性	RENV1:b9
			B 接(加電流 OFF で検出)	0
			A 接(加電流 ON で検出)	1
			2. 入力時動作	RENV1:b8
4	インポジション 入力仕様	INPOS	即停止	0
			減速停止	1
			SVALM 不使用時は A 接(RENV1:b9=1), 端子は接続しない	
			1. INPOS 制御使用 / 不使用	RMD:b9
			動作完了の直後に INPOS を反映しない	0
			INPOS 入力 ON で動作完了	1
5	サボレディ 入力仕様	SVRDY	2. 入力極性	RENV1:b22
			B 接(加電流 OFF で検出)	0
			A 接(加電流 ON で検出)	1
			INPOS 不使用時は A 接(RENV1:b22=1, RMD:b9=0), 端子は NC	
			1. 入力極性	RENV1:b24
			B 接(加電流 OFF で検出)	0
6	サボ偏差 カウンタ 出力仕様	SVCTRCL	A 接(加電流 ON で検出)	1
			SVRDY 不使用時は A 接(RENV1:b24=1), 端子は接続しない	
			サボレディ入力として使用時は PCS 機能使用不可 PCS 機能使用時はサボレディ入力としては使用不可	
			1. 原点復帰完了時自動出力	RENV1:b11
			自動出力しない	0
			自動出力する	1
7	サボオン, サボリセット, サボトルク, サボゲイン, 出力仕様	SVON SVRST SVTL SVGAIN	2. ±ELS, SVALM 入力即停止時自動出力	RENV1:b10
			自動出力しない	0
			自動出力する	1
			3. 出力パルス幅設定	RENV1:b14-12
			1 3msec (推奨)	100
			1. 汎用出力ポートにより制御 RENV2:b7-0 の設定後有効(RENV2:b7-0 設定:11111111)	IOP0B:b3-0
			SVON	b0=1 で出力 ON
			SVRST	b1=1 で出力 ON
			SVTL	b2=1 で出力 ON
			SVGAIN	b3=1 で出力 ON

表 7.2-3 サーボインターフェース設定

7.2.3 原点復帰方法設定

原点が必要な機械系において原点に関する初期設定しておきます。

No.	項 目	設定箇所等
1	原点復帰方法(ORGmode)	RENV3 の ORGmode(b3-0) 0000-1100 次ページの「(1) 原点センサ構成と原点復帰方法」を参照
2	O L S 入力極性	RENV1 の ORGL(b7) 0: A 接, 1: B 接
3	エンコーダZ相信号の入力極性	RENV2 の EZL(b12) 0: 立下りエッジ, 1: 立ち上がりエッジ
4	エンコーダZ相カウント数の設定	RENV3 の EZD3~0(b7-4) 0000 (1 回目) ~1111 (16 回目) (カウント数 - 1)
5	± E L S 入力極性	IOP0B の ELL(b7) 0: A 接, 1: B 接
6	± E L S on時の停止方法	RENV1 の ELM(b3) 0: 即停止, 1: 減速停止
7	補助速度の設定 (ORGmode =1,4,6,7 で使用)	R F A, R M G
8	ベース速度の設定	R F L, R M G
9	動作速度の設定	R F H, R M G
10	原点復帰完了時の設定	RENV3 の CU3R-CU1R(b22-20) CU1R(b20) = 1: CTR1 (指令位置) をリセットします。 CU2R(b21) = 1: CTR2 (機械位置) をリセットします。 CU3R(b22) = 1: CTR3 (汎用・偏差) をリセットします。
11	SVCTRCL 信号自動出力の設定	RENV1 の ERROR(b11) 0: 原点復帰完了時に SVCTRCL 信号を出力しない。 1: 原点復帰完了時に SVCTRCL 信号を自動出力する。

表 7.2-4 原点復帰方法設定要件

(1) 原点センサ構成と原点復帰方法

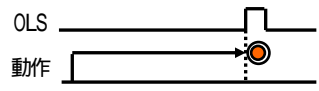
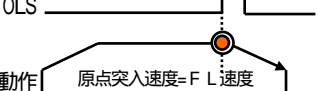
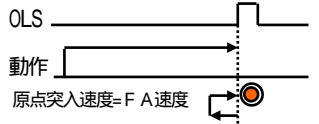
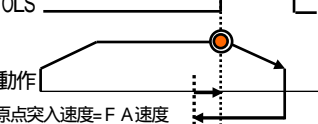
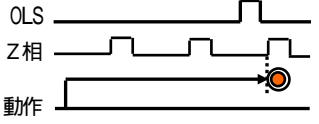
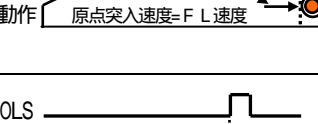

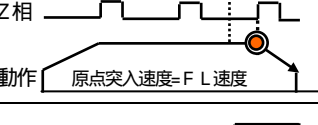
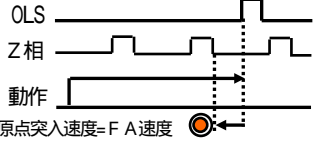
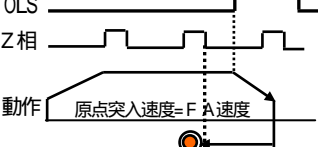
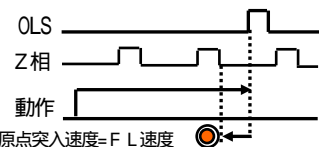

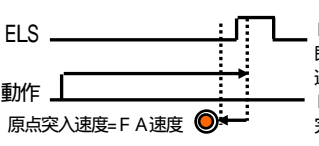
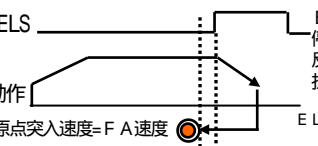
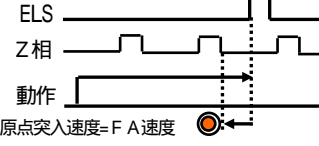
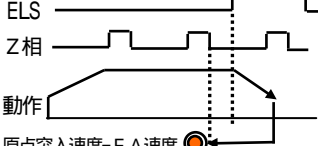
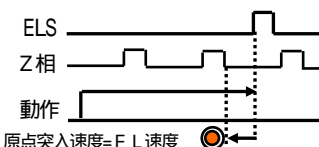
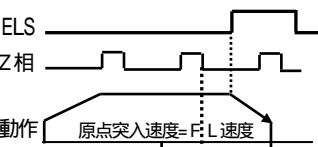
No	原点復帰方法名	定速原点復帰 (F H定速スタート)	高速原点復帰 (加速スタート)
0	OLS 原点 RENV3 ORGmode=0 (0000)	 <p>OLS 検出で完了</p>	 <p>OLS 検出で減速し、ベース速度 (FL) で完了</p>
1	OLS 反転原点 RENV3 ORGmode=1 (0001)	 <p>OLS 検出で即停止・F A 速度で反転し、OLS 拔出し再びOLS検出で完了</p>	 <p>OLS 検出で減速停止・ベース速度でOLSを反転拔出し、再びOLS検出で完了</p>
2	OLS +ENC 原点 RENV3 ORGmode=2 (0010)	 <p>OLS 検出後 n 個目の Z 相検出で完了</p>	 <p>OLS 検出で減速，n 個目の Z 相検出で完了 減速中に最後の Z 相があると完了</p>
3	OLS +ENC 原点 RENV3 ORGmode=3 (0011)	 <p>OLS 検出後 n 個目の Z 相検出で完了</p>	 <p>OLS 検出後 n 個目の Z 相で減速し、ベース速度で完了</p>
4	OLS 反転 +ENC 原点 RENV3 ORGmode=4 (0100)	 <p>OLS 検出で即停止，F A 速度で反転し n 個目の Z 相検出で完了</p>	 <p>OLS 検出で減速，ベース速度になった時点で F A 速度で反転・n 個目 Z 相検出で完了</p>
5	OLS 反転 +ENC 原点 RENV3 ORGmode=5 (0101)	 <p>OLS 検出で即停止，ベース速度で反転し n 個目の Z 相検出で完了</p>	 <p>OLS 検出で減速，ベース速度になった時点で反転加速，n 個目の Z 相で減速しベース速度で完了</p>
6	ELS 兼用センサ原点 RENV3 ORGmode=6 (0110)	 <p>ELS 検出で即停止，F A 速度で反転し ELS 検出で完了</p>	 <p>ELS 検出で減速停止・F A 速度で反転し，ELS 拔出し時点で完了 ELS 幅 > 減速距離</p>
7	ELS 反転定速 +ENC 原点 RENV3 ORGmode=7 (0111)	 <p>ELS 検出で即停止，F A 速度で反転し n 個目の Z 相検出で完了</p>	 <p>ELS 検出で減速停止・F A 速度で反転し，n 個目の Z 相検出で完了 ELS 幅 > 減速距離</p>
8	ELS 反転定速 +ENC 原点 RENV3 ORGmode=8 (1000)	 <p>ELS 検出で即停止，ベース速度で反転し n 個目の Z 相検出で完了</p>	 <p>ELS 検出で減速，ベース速度になった時点で反転加速，n 個目の Z 相で減速しベース速度で完了 ELS 幅 > 減速距離</p>

表 7 . 2 - 5 原点復帰方法

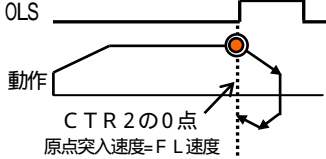
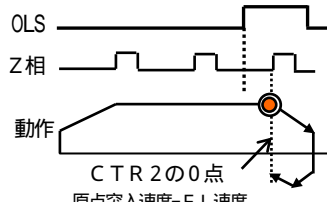
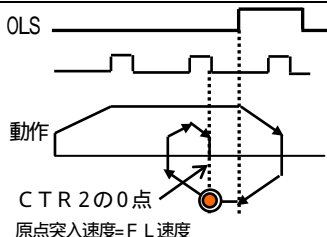
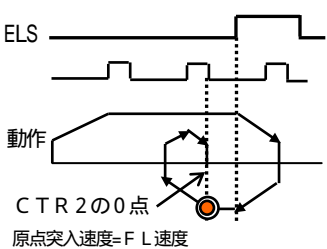
No	原点復帰方法名	高速原点復帰（加速スタート）	
9	OLS反転+CTR 2原点 RENV3 ORGmode=9 (1001)		<p>OLS検出で減速，同時にCTR 2クリア 停止後に反転し，CTR 2の0点で完了． CTR 2カウント入力是指令パルス(推奨) (RENV3 b9-b8='01')</p>
10	ENC反転+CTR 2原点 RENV3 ORGmode=10 (1010)		<p>OLS検出後のn回目Z相で減速，同時にCTR 2クリア 停止後に反転し，CTR 2の0点で完了． CTR 2カウント入力はエンコーダ(推奨) (RENV3 b9-b8='00')</p>
11	ENC反転+CTR 2原点 RENV3 ORGmode=11 (1011)		<p>OLS検出で減速，停止後反転． n回目Z相で減速停止と同時にCTR 2クリア． 停止後に反転し，CTR 2の0点で完了． CTR 2カウント入力はエンコーダ(推奨) (RENV3 b9-b8='00')</p>
12	ELS反転+CTR 2原点 RENV3 ORGmode=12 (1100)		<p>ELS検出で減速，停止後反転． n回目Z相で減速停止と同時にCTR 2クリア． 停止後に反転し，CTR 2の0点で完了． CTR 2カウント入力はエンコーダ(推奨) (RENV3 b9-b8='00')</p>

表7.2-6 原点復帰方法（CTR 2参照方式）

原点が必要な機械系において原点に関する初期設定しておきます．なお，原点復帰時の速度，加減速，については次節「8.3.2 速度パターン設定」を参照して下さい．

- (1) 原点の構成・・・大別して次の4通りの原点構成から選択します．(13通りの構成方法があります．)
センサ原点(OLS)， センサ原点とエンコーダZ相，
ELS兼用原点， ELS兼用原点とエンコーダZ相
- (2) 原点復帰速度・・・定速で復帰，高速復帰（加減速を伴う速度）

原点復帰方法は「表6.2-5：原点復帰方法，表6.2-6：原点復帰方法（CTR 2参照方式）」により選択して下さい．

「表8.2-5：原点復帰方法，表8.2-6：原点復帰方法（CTR 2参照方式）」において採用する原点復帰方法のORGmodeをRENV3のORGmode(b3~b0)にセットしておきます．

原点復帰動作の所定のタイミングでCTR 1～CTR 4を選択してクリアすることが出来ます．

RENV3 bit20 = '1'・・・CTR 1（指令位置加減）， bit21 = '1'・・・CTR 2（機械位置加減）
bit22 = '1'・・・CTR 3（汎用・偏差加減）

注．クリアするタイミングは図中の⊙マークで示してあります．

表6.2-6：原点復帰方法（CTR 2参照方式）のNo. 9～12の原点復帰方法は原点復帰途上の原点検出点でCTR 2をクリアし，減速終了時のCTR 2の符号を反転した移動量の位置決めをする方式です．

したがって原点復帰完了時のCTR 2は“0”になるとは限りません．

7.2.4 イベント要因の設定

RIRQ（イベント要因レジスタ）を初期化時に設定します。
RIRQに設定した要因が発生すると、MMSTSのb3 = '1' となります。
この時、RISTを読み出しイベント要因を確認できます。
通常はb0のみを'1' とします。（正常終了報告）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	IRSA	IRSD	IROL	IRLT	IRCL	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IREN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図7.2-1 RIRQ：イベント要因設定レジスタのビット構成

7.3 モーションモジュール（デバイス）の運用

初期設定後、各動作は、各種の動作モードにしたがって、スタートコマンド書き込みで動作開始します。
動作モードはRMD（動作モードレジスタ）のMOD（動作モード）に設定します。
必要に応じて速度パターンをRFL（ベース速度）、RFH（動作速度）、RUR/RDR（加速/減速レート）、RMG（速度倍率）、RDP（減速開始点）、RUS/RDS（加速/減速S字区間）、RFA（補助速度）に設定します。移動量が必要な場合は 移動量（相対座標値）をRMV（移動量レジスタ）に書き込みます。
最後にスタートコマンド（加速スタート、定速スタートなど）を書込むことによって動作開始します。

7.3.1 動作モード

動作モード“MOD”はスタートコマンドを発行する前に設定します。MODは大別して6モードあります。
動作モードはRMD（動作モードレジスタ）のMOD（b6-0）で設定します。

No	動作モード分類	MOD	動作モード
1	コマンドによる連続動作	00h	+方向連続動作
		08h	-方向連続動作
2	位置決め動作	41h	位置決め動作
		44h	指令位置0点復帰動作
		45h	機械位置0点復帰動作
		46h	+方向1パルス動作
		4Eh	-方向1パルス動作
		47h	タイマー動作
3	原点復帰動作	10h	+方向原点復帰動作
		18h	-方向原点復帰動作
		12h	+方向原点拔出し
		1Ah	-方向原点拔出し
		15h	+方向原点サーチ
		1Dh	-方向原点サーチ
4	ELS, SLS まで動作	20h	+ELSまたは+SLS位置まで動作
		28h	-ELSまたは-SLS位置まで動作
		22h	+ELSまたは+SLS拔出し動作
		2Ah	-ELSまたは-SLS拔出し動作
5	Z相移動	24h	+方向にZ相カウント分動作
		2Ch	-方向にZ相カウント分動作
6	手動パルス動作	01h	パルス入力による連続動作
		51h	パルス入力による位置決め動作
		54h	パルス入力による指令位置0点復帰動作
		55h	パルス入力による機械位置0点復帰動作

表7.3-1 動作モード一覧

(1) コマンドによる連続動作

このモードにしてスタートコマンドを発行すると、停止コマンドが発行されるまで動作します。

+ 方向送り MOD = 00h - 方向送り MOD = 08h	動作	1. モード設定 + 方向:"00h", - 方向:"08h"をMODに設定 2. 速度設定 動作速度をRFH(動作速度)レジスタに設定 3. スタートコマンド 加速"0053h"/FH定速"0051h"を書込み 4. 停止コマンド 停止は減速停止"004Ah"/即停止"0049h"を書込み
	ELS	1. + (-)方向に進行時+ (-)ELSを検出して停止, ELS検出後逆方向へはスタートコマンドで動作 2. ELS検出時の停止方法はRENV1'b3'で設定

表7.3-2 コマンドによる連続動作モード

(2) 位置決め動作

タイマ・動作を除き、ELS・DLS・SLS動作は共通です。

位置決め MOD = 41h	このモードでスタートを発行すると、RMV(移動量レジスタ)設定値分移動 動作方向はRMVの符号による(0 < RMV: + 方向, 0 > RMV: - 方向)	
	動作	1. モード設定 "41h"をMODに設定 2. 移動量パルス数 RMVに相対移動量を設定, 動作方向は符号で決まる. [-134,217,728 移動量 +134,217,727(相対位置送り)] 移動量0でスタートするとパルスは出力されず即完了 3. 速度設定 動作速度をRFH(動作速度レジスタ)に設定 4. スタートコマンド 加速"0053h"/FH定速"0051h"を書込み 5. 停止コマンド 途中停止は減速停止"004Ah"/即停止"0049h"を書込み
	ELS	1. + (-)方向に進行時+ (-)ELSを検出して停止, ELS検出後逆方向へはスタートコマンドで動作 2. ELS検出時の停止方法はRENV1'b3'で設定
指令位置0点復帰 MOD = 44h	このモードでスタートを発行すると、CTR1(指令位置カウンタ)が0になるまで移動	
	動作	1. モード設定 "44h"をMODに設定 2. 速度設定 動作速度をRFH(動作速度レジスタ)に設定 3. スタートコマンド 加速"0053h"/FH定速"0051h"を書込み 4. 停止コマンド 途中停止は減速停止"004Ah"/即停止"0049h"を書込み
	ELS	前に同じ
機械位置0点復帰 MOD = 45h	このモードでスタートを発行すると、CTR2(機械位置カウンタ)が0になるまで移動	
	動作	1. モード設定 "45h"をMODに設定 2. 速度設定 動作速度をRFH(動作速度レジスタ)に設定 3. スタートコマンド 加速"0053h"/FH定速"0051h"を書込み 4. 停止コマンド 途中停止は減速停止"004Ah"/即停止"0049h"を書込み
	ELS	前に同じ
1パルス送り + 方向送り MOD = 46h - 方向送り MOD = 4Eh	RMV(移動量レジスタ)に設定せずに、スタートコマンド書き込みで1パルス移動して完了	
	動作	1. モード設定 +1パルス:"46h", -1パルス:"4Eh"をMODに設定 2. 速度設定 動作速度をRFH(動作速度レジスタ)に設定 3. スタートコマンド FH定速"0051h"を書込み
	ELS	前に同じ

(次ページに続く)

(前ページからの続き)

タイマー動作 MOD = 47h	内部動作時間をタイマーとして使用するモード RMV (移動量レジスタ) はタイマーカウンタとして動作。 RMV の設定値範囲 [1 ~ 134,217,727 : 正数] RMV のパルス数と RFH (動作速度レジスタ) と RMG (速度倍率設定レジスタ) に設定した速度の積がタイマ時間となる。カウントが 0 になった時完了となる。 (例: RMV = 100 [パルス], RFH = 1000, RMG = 199 の場合 100ms)	
	動作	1. モード設定 "47h" を MOD に設定 2. タイマ速度設定 RFH, RMG にタイマ速度 (PPS) を設定 (例: RFH=1000, RMG=199:1000pps=1ms) 3. 移動量パルス数 RMV に停止時間を設定 4. スタートコマンド F H 定速 "0051h" を書込み 5. 停止コマンド 途中停止は即停止 "0049h" を書込み
	注意	1. RMD レジスタの b11=0 にして使用 (動作完了タイミング METM) 2. 指令パルス出力なし。CTR1 (指令位置) のカウントは停止 3. ±ELS, DLS, SLS, INPOS は無視 4. タイマ動作中の SVALM 信号は有効 5. バックラッシュ補正, 振動抑制機能, 方向変化時タイマーの機能は停止。

表 7.3-3 位置決め動作

(3) 原点復帰動作

原点復帰動作は、動作方法によって OLS 入力, EZ 入力, ±ELS 入力を使用します。

OLS 入力信号は、論理設定を RENV1 (環境設定 1) レジスタで行い、端子の状態は RSTS (拡張ステータス) レジスタでモニタできます。

エンコーダ Z 相入力極性を RENV2 (環境設定 2) で、原点復帰完了条件のエンコーダ Z 相入力カウント数の設定を RENV3 (環境設定 3) で行い、入力状態は RSTS レジスタでモニタできます。

±ELS 入力極性を汎用出力ポート (IOPOB) で設定し、入力 ON 時の動作 (即停止 / 減速停止) を RENV1 で行います。また入力状態は RSTS (拡張ステータス) でモニタできます。

原点復帰動作は原点復帰動作、原点拔出し動作、原点サーチ動作を RMD の MOD (動作モード) で指定し、スタートコマンド 書込みで実行します。

< 原点復帰動作設定項目と関連ステータス >

No.	項目	設定レジスタ等
1	原点復帰方法 (ORGmode)	RENV3 の ORGmode (b3-0) 0000 ~ 1100 「8.2.3 原点復帰方法設定」を参照
2	OLS 入力極性	RENV1 の ORGL (b7) 0: A 接, 1: B 接
3	OLS 入力状態読出し	RSTS の SORG (b8) 0: OFF, 1: ON
4	エンコーダ Z 相信号の入力極性	RENV2 の EZL (b12) 0: 立下りエッジ, 1: 立ち上がりエッジ
5	エンコーダ Z 相入力カウント数の設定	RENV3 の EZD3 ~ 0 (b7-4) 0000 (1 回目) ~ 1111 (16 回目) (カウント数 - 1)
6	エンコーダ Z 相入力状態の読出し	RSTS の SEZ (b16) 0: OFF, 1: ON
7	±ELS 入力極性	IOPOB の ELL (b7) 0: A 接, 1: B 接
8	±ELS on 時の停止方法	RENV1 の ELM (b3) 0: 即停止, 1: 減速停止
9	±ELS 入力状態の読出し	RSTS の SPEL (b6), SMEL (b7) 0: OFF, 1: ON
10	補助速度の設定 (ORGmode = 1, 4, 6, 7 で使用)	RFA, RMG
11	ベース速度の設定	RFL, RMG
12	動作速度の設定	RFH, RMG
13	原点復帰完了時の設定	RENV3 の CU3R-CU1R (b22-20) CU1R (bit 20) = 1: CTR1 (指令位置) をリセットします。 CU2R (bit 21) = 1: CTR2 (機械位置) をリセットします。 CU3R (bit 22) = 1: CTR3 (汎用・偏差) をリセットします。
14	SVCTRCL 信号自動出力の設定	RENV1 の ERROR (b11) 0: 原点復帰完了時に SVCTRCL 信号を出力しない。 1: 原点復帰完了時に SVCTRCL 信号を自動出力する。

表 7.3-4 原点復帰動作設定項目と関連ステータス

原点復帰動作 + 方向復帰 MOD = 10h - 方向復帰 MOD = 18h	動作	スタートマツト書き込み後，原点復帰完了条件が成立するまで動作． 原点復帰完了時，自動的にカウンタリセット，SVCTRCL(偏差カウンタ)信号出力が可． 基本的な原点復帰方法，原点復帰完了時のカウンタリセットの有無は RENV3 に設定． SVCTRCL 信号自動出力の有無は RENV1 に設定． 1. モード設定 + 方向: "10h", - 方向: "18h" を MOD に設定 2. 速度設定 動作速度を R F H (動作速度レジスタ) に設定 3. スタートマツト 加速"0053h" / F H 定速"0051h" を書き込み 4. 停止マツト 途中停止は減速停止"004Ah" / 即停止"0049h" を書き込み
	E L S	原点が E L S 兼用原点に設定してある場合を除き， 原点復帰中に E L S を検出した場合は通常の E L S 動作 (即停止/減速停止)
原点拔出し動作 + 方向拔出し MOD = 12h - 方向拔出し MOD = 1Ah	動作	スタートマツト書き込み後，原点(OLSon 状態)を拔出すまで動作． 動作開始前の O L S 状態により O L S = O F F 指令を行っても直ちに完了 (正常停止) O L S = O N O L S が O F F となり 1 パルス拔出で完了 1. モード設定 + 方向: "12h", - 方向: "1Ah" を MOD に設定 2. 速度設定 動作速度を R F H (動作速度レジスタ) に設定 3. スタートマツト F H 定速"0051h" を書き込み 4. 停止マツト 途中停止は即停止"0049h" を書き込み
	E L S	原点拔出し中に E L S を検出した場合は通常の E L S 動作 (即停止/減速停止)
原点サーチ動作 + 方向サーチ MOD = 15h - 方向サーチ MOD = 1Dh	動作	原点復帰動作に機能を追加したモード．下記の動作ブロックで構成 指定と逆方向の「原点復帰動作 (ORGmode=0 の動作: OLSoff OLSon まで動作)」 指定と逆方向の「位置決め動作による原点抜け出し動作」 指定方向の「原点復帰動作 (RENV3:ORGmode の設定に従った動作)」 現在位置と O L S の位置関係で，3 通りの動作があります (1) O L S が O F F : (2) O L S が O N : (3) O L S と E L S : E L S on で反転後 1. モード設定 + 方向: "15h", - 方向: "1Dh" を MOD に設定 2. 速度設定 動作速度を R F H (動作速度レジスタ) に設定 3. スタートマツト 加速"0053h" / F H 定速"0051h" を書き込み 4. 停止マツト 途中停止は減速停止"004Ah" / 即停止"0049h" を書き込み
	注意	1. 原点復帰動作は " RENV3:ORGmode " 設定に従います． 2. 位置決め動作による原点抜け出し動作の移動量は " R M V " に設定された値 であり，原点を拔出すまで繰返しこの値が使用されます． R M V には正数 (1 ~ 134,217,727) を設定して下さい．

表 7. 3 - 5 原点復帰動作

(4) E L S , S L S 動作

リミットに対して移動するモードであり，2 種類の動作があります．

± E L S 入力極性を汎用出力ポート (I O P O B) で設定し，入力 ON 時の動作 (即停止 / 減速停止) を R E N V 1 で行います．また入力状態は R S T S (拡張ステータス) でモニタできます．

S L S (ソフトリミット) の設定に関しては，" . ソフトリミット機能 " をご参照下さい．

E L S, S L S まで移動 + E L S, + S L S まで MOD = 2 0 h - E L S, - S L S まで MOD = 2 8 h	動作	E L S および S L S が O F F の場合に動作指令を行うと, E L S または S L S が O N の位置で正常終了 E L S または S L S が O N の場合に動作指令を行うと, 動作せずに正常終了.
		1. モード設定 + 方向: "20h", - 方向: "28h" を MOD に設定 2. 速度設定 動作速度を R F H (動作速度レジスタ) に設定 3. スタートコマンド 加速 "0053h" / F H 定速 "0051h" を書き込み 4. 停止コマンド 途中停止は減速停止 "004Ah" / 即停止 "0049h" を書き込み
	E L S	E L S 検出での停止は, R E N V 1: E L M (bit 3) 設定で即停止 (0) / 減速停止 (1)
E L S, S L S 拔出し動作 + E L S, + S L S 拔出し MOD = 2 2 h - E L S, - S L S 拔出し MOD = 2 A h	動作	E L S が O N または S L S が O N の場合に指令動作を行うと, E L S または S L S が O F F の位置で正常終了 E L S および S L S が O N の場合に動作指令を行うと, 動作せずに正常終了
		1. モード設定 + 方向: "22h", - 方向: "2Ah" を MOD に設定 2. 速度設定 動作速度を R F H (動作速度レジスタ) に設定 3. スタートコマンド F H 定速 "0051h" を書き込み 4. 停止コマンド 途中停止は即停止 "0049h" を書き込み

表 7. 3 - 6 E L S, S L S 動作

(5) Z 相移動動作

R E N V 3 の (E Z D 設定値 + 1) 回だけエンコーダ Z 相信号をカウントするまで動作するモードです。
スタートコマンド書き込み後, 設定回数分エンコーダ Z 相信号をカウントすると即停止します。

E Z 入力信号は, 入力極性を R E N V 2 (環境設定 2) で, エンコーダ Z 相信号カウント数の設定を R E N V 3 (環境設定 3) で行い, エンコーダ Z 相入力状態は R S T S (拡張ステータス) レジスタでモニタできます。

+ 方向へ移動 MOD = 2 4 h - 方向へ移動 MOD = 2 c h	動作	Z 相の設定回数は 1 ~ 1 6 回 (E Z D 設定値: 0 ~ 1 5) で必ず " 定速 " 動作
		1. モード設定 + 方向: "24h", - 方向: "2Ch" を MOD に設定 2. 速度設定 動作速度を R F H (動作速度レジスタ) に設定 3. スタートコマンド F H 定速 "0051h" を書き込み 4. 停止コマンド 途中停止は即停止 "0049h" を書き込み

表 7. 3 - 7 Z 相移動モード

(6) 手動パルス動作

パルス入力で動作するモードです。

手動パルス動作にはパルス連続送り、パルス位置決め送り、パルス同期指令位置0点復帰、パルス同期機械位置0点復帰の動作モードがあります。

いずれも、スタートコマンドを発行してからパルス信号が入力可能となり、パルス入力すると指令出力にパルスが出力されます。パルス入力はエンコーダ入力と同一入力端子です。パルスとして使用した場合はエンコーダは接続できません。

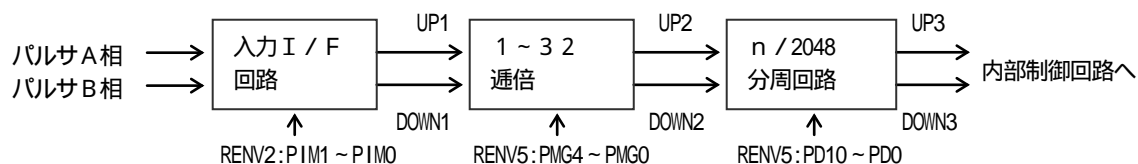
パルス信号の入力I/F回路により、パルスA/B相入力仕様はRENV2(環境設定2)レジスタの設定により、4通りの入力形式が選択できます。

90度位相差信号(1, 2, 4 通倍)を入力

2パルス入力(UP/DOWN)

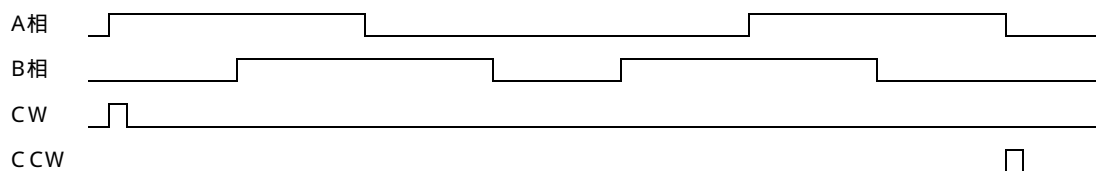
また、上記1～4通倍の他に、1～32通倍回路と(1～2048)/2048分周回路が内蔵されています。

1～32通倍設定はRENV5のPMG4～PMG0で行い、n/2048分周設定はRENV5のPD10～PD0で行います。

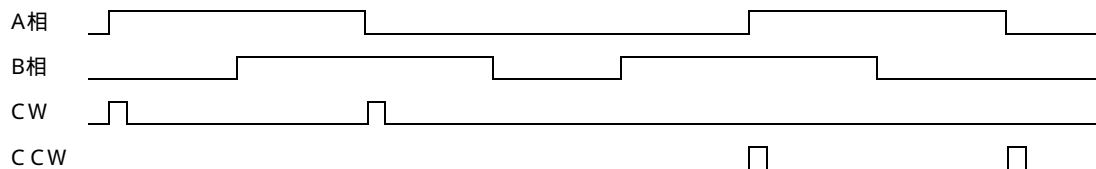


CW, CCWの信号はRENV2のPIM1～PIM0の設定により、下記のようになります。

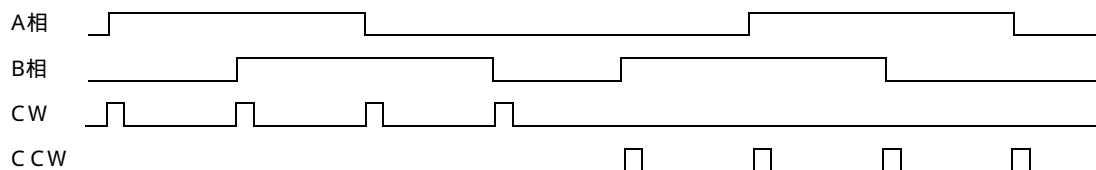
90度位相差信号1通倍入力(PIM=00)の時



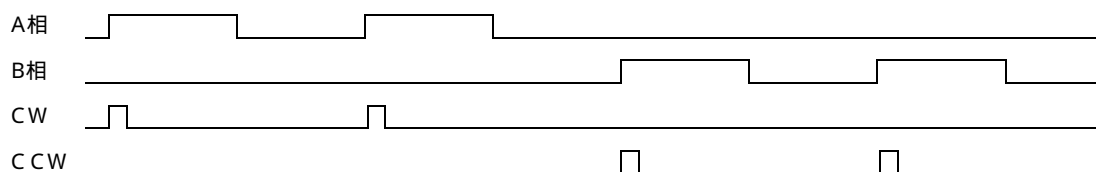
90度位相差信号2通倍入力(PIM=01)の時



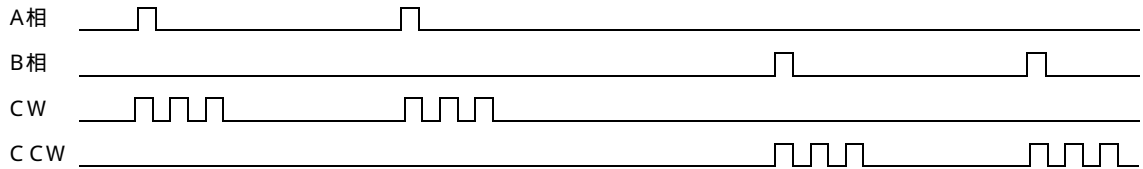
90度位相差信号4通倍入力(PIM=10)の時



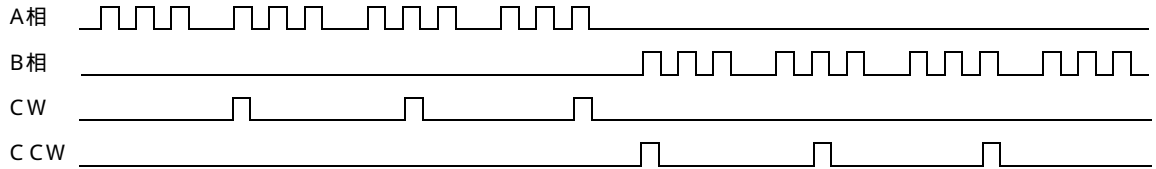
2パルス入力の時(PIM=11)の時



1 ~ 3 2 通倍回路を 3 通倍(REN5 の PMG=2)にした場合の動作は、下記のようになります。



n / 2048 分周回路の分周比を 512 / 2048 (REN5 の PD=512) にした時の動作は、下記のようになります。



パルス入力モードはF H定速スタートモード (0051h) によりスタートさせます。

パルス入力によりF H速度の内部パルスを歯抜けさせて出力します。

従って、パルス入力と出力パルスのタイミングには、最大で内部パルス周期分の誤差が発生します。

パルス信号の最高入力周波数はF H速度により制限されます。

エンコーダ端子から入力されるパルス信号入力と同時に変化した場合と入力周波数がオーバーして、入力用バッファカウンタ(16ビット)がオーバーフローした場合にエラーが発生します。

エラー要因はR E S T (エラーステータス)で確認できます。

パルス入力周波数をF Pとすると

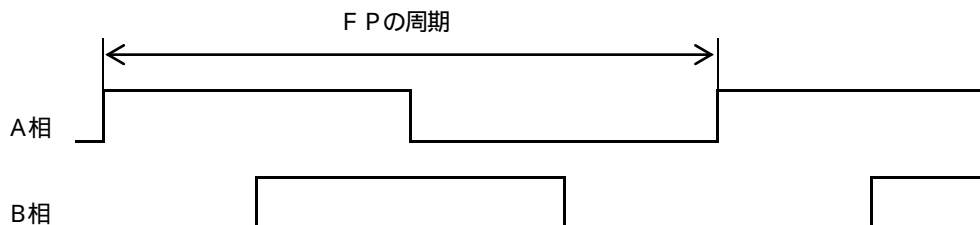
$$F P < (\text{設定速度}) / (\text{入力 I} / \text{F 通倍値}) / (\text{PMG 設定値} + 1) / (\text{PD 設定値} / 2048) \quad \text{PD 設定値} = 0$$

$$F P < (\text{設定速度}) / (\text{入力 I} / \text{F 通倍値}) / (\text{PMG 設定値} + 1) \quad \text{PD 設定値} = 0$$

< F H速度[pps]とパルス入力周波数F P[pps]との関係例>

パルス A / B 相 入力設定	PMG (通倍) 設定値 REN5:PMG4 ~ PMG0	PD (分周比) 設定値 REN5:PD10 ~ PD0	使用範囲
90度位相差 1 通倍 (REN2:PIM=00)	0 (1 倍)	0 (0000h)	$F P < F H$
	0 (1 倍)	1 0 2 4 (0400h)	$F P < F H \times 2$
	2 (3 倍)	0 (0000h)	$F P < F H / 3$
" 2 通倍 (REN2:PIM=01)	0 (1 倍)	0 (0000h)	$F P < F H / 2$
	0 (1 倍)	1 0 2 4 (0400h)	$F P < F H$
	2 (3 倍)	0 (0000h)	$F P < F H / 6$
" 4 通倍 (REN2:PIM=10)	0 (1 倍)	0 (0000h)	$F P < F H / 4$
	0 (1 倍)	1 0 2 4 (0400h)	$F P < F H / 2$
	2 (3 倍)	0 (0000h)	$F P < F H / 6$
2 パルス入力 (REN2:PIM=11)	0 (1 倍)	0 (0000h)	$F P < F H$
	0 (1 倍)	1 0 2 4 (0400h)	$F P < F H \times 2$
	2 (3 倍)	0 (0000h)	$F P < F H / 3$

表 7 . 3 - 8 F H速度[pps]とパルス入力周波数F P[pps]との関係例



注 . パルス A / B 相入力周波数が変動する場合には平均周期ではなく最短周期を上記の「F Pの周期」として下さい。

< パルスA / B相入力関係の設定 >

項 目	レジスタ:ビット	設定値				
入力仕様	RENV2 : PIM1,0(b15,14)	b15,14	00	01	10	11
		逓倍	1 逓倍	2 逓倍	4 逓倍	UP/DOWN
カウント極性	RENV2 : PDIR (b16)	0 : A相進相, 1 : A相遅相				
入力マスク	RENV2 : POFF (b18)	0 : 入力許可, 1 : 入力禁止				
動作状態の読み出し	RSTS : CND (b3-0)	0 1 0 1 : パルスA相 / B相入力待ち				
同相入力エラー	REST : ESPE (b14)	1 : パルスA相 / B相同相入力エラー発生				
入力用パッファオーバーフロー	REST : ESPO (b9)	1 : パッファオーバーフロー発生				

表 7. 3 - 9 パルスA / B相入力関係の設定

スタートコマンド	F H定速コマンド"0051h"を発行する。パルス入力最高周波数はR F HとR M Gで指定する	
E L S	E L S検出でパルス出力停止するが、逆方向は動作可能 E L S入力による停止時に、エラーステータスは変化なし	
モード解除	即停止コマンド"0049h"を発行	
パルス連続送り MOD = 0 1 h	動作	パルス入力により連続動作するモード。パルス送りによる指令パルス出力（通常のパルス送り）スタートコマンドを書き込み後、パルス信号が入力されるとパルス入力信号と同期して指令パルス出力。動作方向は、パルス信号の入力とパルス入力カウント極性設定により決定。 1. モード設定 "01h"をMODに設定 2. パルス入力最高周波数設定 パルス入力最高周波数をR F HとR M Gに設定 3. スタートコマンド F H定速"0051h"を書込み
パルス位置決め送り MOD = 5 1 h	動作	パルス入力信号(方向は無視)に同期して位置決め動作、目標位置に達すると動作終了 R M V設定値を相対位置データとして、パルス入力(方向は無視)に同期して位置決め動作。 目標位置に達すると動作終了。以降のパルス入力は無視される。 R M Vを "0" に設定してスタートさせると、パルス出力せずに即停止。 1. モード設定 "51h"をMODに設定 2. 移動量パルス数 R M Vに相対移動量を設定、動作方向は符号で決定 3. パルス入力最高周波数設定 パルス入力最高周波数をR F HとR M Gに設定 4. スタートコマンド F H定速"0051h"を書込み
パルス同期 指令位置0点復帰 MOD = 5 4 h	動作	パルス入力信号(方向は無視)に同期して“指令位置:CTR1=0”となるまで移動。 動作方向と移動量は、動作指令時のCTR1(指令位置)で決まる。 1. モード設定 "54h"をMODに設定 2. パルス入力最高周波数設定 パルス入力最高周波数をR F HとR M Gに設定 3. スタートコマンド F H定速"0051h"を書込み
パルス同期 機械位置0点復帰 MOD = 5 5 h	動作	パルス入力信号(方向は無視)に同期して“機械位置:CTR2=0”となるまで移動。 動作方向と移動量は、動作指令時のCTR2(機械位置)で決まる。 1. モード設定 "55h"をMODに設定 2. パルス入力最高周波数設定 パルス入力最高周波数をR F HとR M Gに設定 3. スタートコマンド F H定速"0051h"を書込み
注 意	1. パルス入力モードでもバックラッシュ補正機能は動作しますが、バックラッシュ補正中にパルス入力を逆転させた場合には、対応できません。 2. PIM1,0,PMG4-0設定により逓倍動作を行っている場合でも即停止コマンド"0049h"を書込むと、即停止します。 この場合、総出力パルス数は逓倍値の整数倍になるとは限りません。	

表 7. 3 - 1 0 手動パルス動作

7.3.2 速度パターン設定

速度パターンの設定は下表のレジスタとRMD（動作モードレジスタ）の一部で行います。

加減速パターン(直線 / S字：RMDのb10で設定)，加減速度（加速レート，減速レート）は通常，初期にこれらの関係のレジスタに設定しておきます。

同じく速度倍率を設定するRMG，ベース速度を設定するRFL，補助速度（原点復帰時の原点突入速度等）を設定するRFAも初期設定しておきます。

動作速度は初期化の対象にはしません。移動指令時にRFH（動作速度レジスタ）に設定してスタートコマンドを発行します。

またこれらのレジスタは設定すべき値が前回と同じ時には，再度書き込む必要はありません。

レジスタによって，“0”は設定範囲外になる場合がありますので注意して下さい。

レジスタ	内 容	bit長	設 定 範 囲	R / W
RMV	移動量	28	-134,217,728(08000000h) ~ 134,217,727(07FFFFFFh)	R / W
RFL	ベース速度	17	1 ~ 100,000 (000186A0h) (注2)	R / W
RFH	動作速度	17	1 ~ 100,000 (000186A0h) (注2)	R / W
RUR	加速レート	16	1 ~ 65,535 (0000FFFFh)	R / W
RDR	減速レート (注1)	16	0 ~ 65,535 (0000FFFFh)	R / W
RMG	速度倍率	11	2 ~ 2,047 (00007FFh)	R / W
RDP	減速開始点	24	0 ~ 16,777,215 (00FFFFFFh)	R / W
RUS	加速時S字区間	16	0 ~ 50,000 (0000C350h) (注3)	R / W
RDS	減速時S字区間	16	0 ~ 50,000 (0000C350h) (注3)	R / W
RFA	補助速度	17	1 ~ 100,000 (000186A0h) (注2)	R / W

表7.3-11 速度パターン設定レジスタ

注1．RDR = 0 に設定した場合，減速レートはRURで設定した値になります。

注2．186A0h~1FFFFFFh(100000~131071)の間は全て186A0h(100000)と見なされます。

注3．0C350h~0FFFFFFh(50000~65535)の間は全て0C350h(50000)と見なされます。

【 加減速動作時のレジスタデータ使用箇所 】

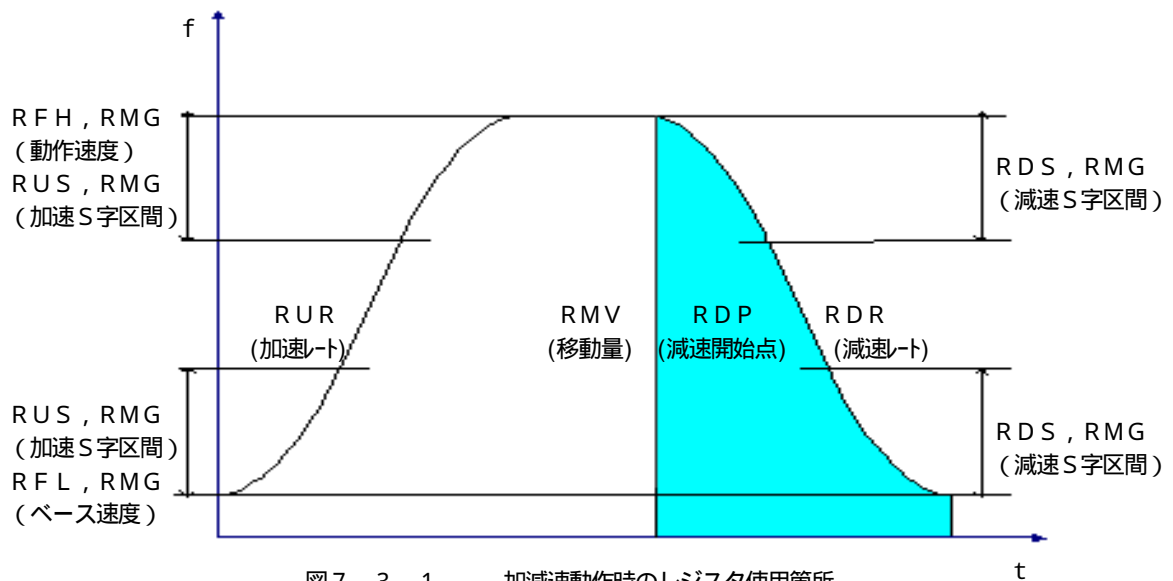


図7.3-1 加減速動作時のレジスタ使用箇所

7.3.3 速度パターン

スタートコマンド，停止コマンドにより速度パターンが変わります．

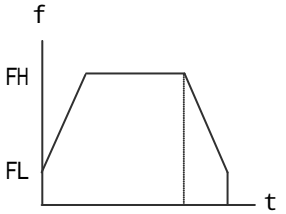
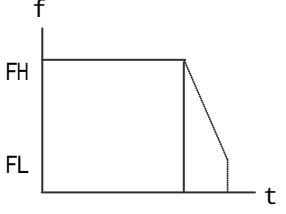
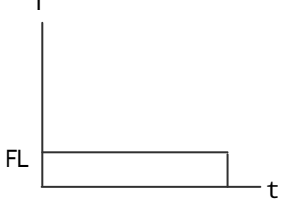
	速度パターン	連続モード	位置決め動作モード
加減速動作		加速スタートコマンド (0053h) の書込み 減速停止コマンド (004Ah) の書込み	加速スタートコマンド (0053h) の書込み 減速開始点到達または減速停止コマンド (004Ah) の書込み RMD(動作モードレジスタ)のb12(MSDP) = 1 (減速開始点手動設定)の時， RDP(減速開始点レジスタ) = 0の場合， 即停止
		即停止コマンド (0049h) の書込みで即停止	
FH定速動作		FH定速スタートコマンド (0053h) の書込み 即停止コマンド (0049h) の書込み	FH定速スタートコマンド (0051h) の書込み 位置決め加減速 = 0 または 即停止 (0049h) コマンド 書込み
		減速停止コマンド (004Ah) の書込みで減速停止	
FL定速動作		FL定速スタートコマンド (0050h) の書込み 即停止 (0049h) / 減速停止 (004Ah) コマンド の書込み	FL定速スタートコマンド (0050h) の書込み 位置決め加減速 = 0 または 即停止 (0049h) / 減速停止 (004Ah) コマンド の書込み

表 7.3-12 速度パターン

7.3.4 速度パターン設定レジスタ

(1) RMV: 移動量レジスタ (28ビット)

位置決め動作において目標位置を設定するレジスタです。動作モードにより設定内容が変わります。

設定範囲は -134,217,728 ~ +134,217,727 です。

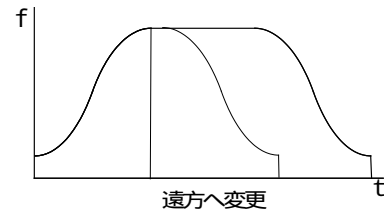
動作中にRMVレジスタを変更する事により、目標位置のオーバーライドが行えます。

< 移動量のオーバーライド >

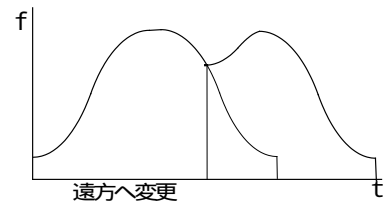
RMVレジスタを書き換える事により、目標位置のオーバーライドができます。

スタート位置を基準として目標位置を変更します。

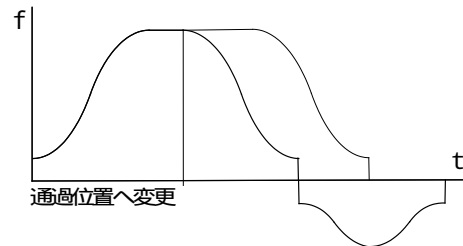
加速・定速動作中に新データを最初の目標位置より遠方に変更したら、そのままの速度パターンで動作し、新データ位置(新RMV値)で位置決め完了になります。



減速中に新データを最初の目標位置より遠方に変更したら、その位置から動作速度まで再加速後、新データ位置(新RMV値)で位置決め完了になります。変更時の速度をFuとした時、再加速カーブは、RFL=Fuの場合の通常加速カーブと同一になります。



新データ位置をすでに通過している時、または減速中で最初の目標位置より手前に変更した時は、減速停止後に逆転して新データ位置(新RMV値)で位置決め完了になります。



加速/減速動作を行うのは加速スタート時のみです。

また、目標位置データ(RMVレジスタ値)の書き換えは、位置決め動作完了まで何度でも可能です。

注1. 位置のオーバーライドは、動作中(FL/FH 定速動作中, 加/減速中, バックラッシュ中)にだけ有効です。

停止直前にオーバーライドを行った場合、オーバーライドが受け付けられない場合があります。

オーバーライドが無視された場合はMMSTSのSERR(bit2) = 1となり、エラー要因は、RESTレジスタで確認できます。

なお、このエラーは停止中にRMVレジスタへのオーバーライド書き込み(0080h)を行った時に発生します。従って、スタート前の停止時に、RMVレジスタへのオーバーライド書き込み(0080h)を行っても位置のオーバーライド失敗イベントが発生します。

(2) RFL: ベース速度レジスタ (17ビット)

加減速動作において、ベース速度(停止速度)を設定するレジスタです。

FL定速動作の速度及び、加減速動作の場合のベース速度を1~100,000(186A0h)の範囲で設定します。

100,000~131,071(186A0h~1FFFFh)の範囲は全て100,000と見なされます。

実際の動作速度はRMG(速度倍率設定レジスタ)の設定値との計算値になります。

$$\text{ベース速度[pps]} = \text{RFL} \times \frac{200}{(\text{RMG} + 1)}$$

(3) RFH : 動作速度レジスタ (17ビット)

動作速度を設定するレジスタです。

動作中にRFHレジスタを変更する事により、速度のオーバーライドが行えます。

FH定速動作の速度及び、加減速動作の場合の動作速度を1~100,000(186A0h)の範囲で設定します。

100,000~131,071(186A0h~1FFFFh)の範囲は全て100,000と見なされます。

加減速動作の場合にはRFL(ベース速度)の設定値よりも大きな値を設定して下さい。

実際の動作速度はRMG(速度倍率設定レジスタ)の設定値との計算値になります。

$$\text{動作速度[pps]} = \text{RFH} \times \frac{200}{(\text{RMG} + 1)}$$

(4) RUR : 加速レートレジスタ (16ビット)

加速レートを設定するレジスタです。

加減速動作の場合の加速特性を1~65,535(FFFFh)の範囲で設定します。

設定値と加速時間の関係は次式のようになります。

直線加速(RMDレジスタのbit10(MSMD)=0)

$$\text{加速時間[秒]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1)}{5,000,000}$$

直線部分のないS字加速(RMDレジスタのbit10(MSMD)=1かつRUS=0)

$$\text{加速時間[秒]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1)}{2,500,000}$$

直線部分のあるS字加速(RMDレジスタのbit10(MSMD)=1かつRUSレジスタ>0)

$$\text{加速時間[秒]} = \frac{(\text{RFH} - \text{RFL} + 2 \times \text{RUS}) \times (\text{RUR} + 1)}{5,000,000}$$

< 加速レート計算例 >

直線加速(RMDレジスタのbit10(MSMD)=0) ,

ベース速度=10pps, 動作速度=110kpps, 加速時間=300ms の場合

110kppsを出力するため倍率を2倍モードに設定する。

RMG=99(63h)。

動作速度を110kppsにするために、倍率を2倍モードに設定したので、

RFH=55000(D6D8h)

スタート速度を10ppsにするためには、倍率を2倍モードに設定したので、

RFL=5(0005h)

加速時間300msにするためには、加速時間とRUR設定値の関係式よりRUR=26.275に設定する。

$$\begin{aligned} \text{加速時間[秒]} &= \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1)}{5,000,000} \\ 0.3[\text{秒}] &= \frac{(55000 - 5) \times (\text{RUR} + 1)}{5,000,000} \end{aligned}$$

これより、

$$\text{RUR} = \frac{5,000,000 \times 0.3}{(55000 - 5)} - 1 = 26.275$$

ただし、RURに設定できるのは整数のみなので、RURには"26" または"27"を設定します。

つまり、実際の加速時間は、297ms(RUR="26"に設定した場合)または308ms(RUR="27"に設定した場合)となります。

(5) RDR : 減速レートレジスタ (16ビット)

減速レートを設定するレジスタです。

加減速動作の場合の減速特性を1~65,535(FFFFh)の範囲で設定します。

ただし、RDR=0に設定した場合、減速レートはRURで設定した値が使用されます。

設定値と減速時間の関係は次式ようになります。

$$\text{直線減速(RMDレジスタのbit10(MSMD)=0)} \\ \text{減速時間[秒]} = \frac{(RFH - RFL) \times (RDR + 1)}{5,000,000}$$

$$\text{直線部分のないS字減速(RMDレジスタのbit10(MSMD)=1かつRDS=0)} \\ \text{減速時間[秒]} = \frac{(RFH - RFL) \times (RDR + 1)}{2,500,000}$$

$$\text{直線部分のあるS字減速(RMDレジスタのbit10(MSMD)=1かつRDSレジスタ>0)} \\ \text{減速時間[秒]} = \frac{(RFH - RFL + 2 \times RDS) \times (RDR + 1)}{5,000,000}$$

減速開始点を自動に設定(RMDレジスタのbit12(MSDP)=0)する場合、

(減速時間) (加速時間×2)となるようにRDRを設定して下さい。

(減速時間) > (加速時間×2)となるようにRDRを設定した場合は、停止時にベース速度まで減速しきれない場合があります。この場合、減速開始点は手動設定(RMDレジスタのbit12(MSDP)=1)に設定して下さい。

(6) RMG: 速度倍率設定レジスタ(11ビット)

速度倍率を設定するレジスタです。

RFL, RFH及びRFA設定値と速度の関係を2~2,047(07FFh)の範囲で設定します。

高倍率になるほど設定できる速度間隔が粗くなります。設定値と速度倍率の関係は次式ようになります。

$$\text{速度倍率[倍]} = \frac{200}{(RMG + 1)}$$

< 速度倍率設定例 >

設 定 値	倍率	速度範囲(pps)	設 定 値	倍率	速度範囲(pps)
1999 (7CFh)	0.1	0.1 ~ 10,000.0	39 (027h)	5	5 ~ 500,000
999 (3E7h)	0.2	0.2 ~ 20,000.0	19 (013h)	10	10 ~ 1,000,000
399 (18Fh)	0.5	0.5 ~ 50,000.0	9 (009h)	20	20 ~ 2,000,000
199 (0C7h)	1	1 ~ 100,000	3 (003h)	50	50 ~ 5,000,000
99 (063h)	2	2 ~ 200,000	2 (002h)	66.6	66.6 ~ 6,666,666.6

表7.3-13 速度倍率設定例

(7) RDP : 減速開始点レジスタ (24ビット)

加減速・位置決め動作の場合の減速開始点を設定します。

RMDレジスタの減速開始点設定方法(bit 12:MSDP)の状態により、RDPに設定する値の意味は異なります。

() マニュアル設定時(RMDレジスタのbit 12(MSDP) = 1)

減速開始点のパルス数を0~16,777,215(00FFFFFFh)の範囲で設定します。

(位置決めカウンタの値) (RDP設定値)のタイミングで減速を開始します。

減速開始点の設定値が最適値より小さい場合、停止時の速度はFL速度より速くなります。

逆に、最適値より大きい場合、減速終了後にFL定速動作をします。

減速開始点の最適値は次式の様になります。

直線減速(RMDレジスタのbit 10(MSMD) = 0)

$$\text{減速開始点の最適値[パルス]} = \frac{(RFH^2 - RFL^2) \times (RDR + 1)}{(RMG + 1) \times 50,000}$$

ただし、FH補正機能をOFFに設定した状態(RMDレジスタのbit 17(MADJ) = 1)で、RFHレジスタに設定する値を変更せずに三角駆動させた場合の最適値は次式のようにになります。

(アイドリング制御を行う場合、以下に示す計算式のRMVにはRMVレジスタの設定値からアイドリングパルス数を差し引いた値を代入して下さい。アイドリングパルス数は、RENV2のIDL = 0~7"の時"1~6"となります)

$$\text{減速開始点の最適値[パルス]} = \frac{RMV \times (RDR + 1)}{RUR + RDR + 2}$$

直線部分のないS字減速(RMDレジスタのbit 10(MSMD)かつRDSレジスタ = 0)

$$\text{減速開始点の最適値[パルス]} = \frac{(RFH^2 - RFL^2) \times (RDR + 1) \times 2}{(RMG + 1) \times 50,000}$$

直線部分のあるS字減速(RMDレジスタのbit 10(MSMD)かつRDSレジスタ > 0)

$$\text{減速開始点の最適値[パルス]} = \frac{(RFH + RFL) \times (RFH - RFL + 2 \times RDS) \times (RDR + 1)}{(RMG + 1) \times 50,000}$$

(8) RUS : 加速S字区間レジスタ (16ビット)

S字加速時のS字区間を設定するレジスタです。

S字加減速動作のS字加速区間を1~50,000(C350h)の範囲で設定します。

50,000~65,535(C350h~FFFFh)の範囲は全て50,000と見なされます。

S字加速区間の範囲 S_{su} はRMG(速度倍率設定レジスタ)の設定値との計算値になります。

$$S_{su}[\text{pps}] = RUS \times \frac{200}{(RMG + 1)}$$

つまり、ベース速度~(ベース速度 + S_{su})までと(動作速度 - S_{su})~動作速度までがS字加速動作となり中間部分は直線加速動作となります。ただし、"0"を設定した場合、内部演算によりS字加速区間は(RFH-RFL)/2が代用され、直線加速部分のないS字加速動作となります。

最小値の"1"を設定した場合、ほぼ直線加速動作と同様になります。

"(RFH-RFL)/2"より大きな値を設定した場合、最大加速度まで到達しなくなり、加速時間が計算式と異なります。このため"(RFH-RFL)/2"以下の値を設定して下さい。

(9) RDS : 減速S字区間レジスタ (16ビット)

S字減速時のS字区間を設定するレジスタです。

S字加減速動作のS字減速区間を1~50,000(C350h)の範囲で設定します。

50,000~65,535(C350h~FFFFh)の範囲は全て50,000と見なされます。

S字減速区間の範囲 S_{SD} はRMG (速度倍率設定レジスタ) の設定値との計算値になります。

$$S_{SD}[\text{pps}] = \text{RDS} \times \frac{200}{(\text{RMG} + 1)}$$

つまり、動作速度~(動作速度 - S_{SD})までと(ベース速度 + S_{SD})~動作速度までがS字減速動作となり中間部分は直線減速動作となります。ただし、"0"を設定した場合、内部演算によりS字減速区間は(RFH-RFL)/2 が代用され、直線減速部分のないS字減速動作となります。

最小値の"1"を設定した場合、ほぼ直線減速動作と同様になります。

"(RFH-RFL)/2"より大きな値を設定した場合、最大加速度まで到達しなくなり、減速時間が計算式と異なります。

このため"(RFH-RFL)/2"以下の値を設定して下さい。

(10) RFA : 補助速度レジスタ (17ビット)

原点復帰動作での逆転定速度、バックラッシュ補正時の定速度を設定するレジスタです。

原点復帰動作での逆転定速度として使用します。

または、バックラッシュ時の移動量補正速度(F A速度)の速度を1~100,000(186A0h)の範囲で設定します。

100,000~131,071(186A0h~1FFFFh)の範囲は全て100,000と見なされます。

実際の動作速度はRMG (速度倍率設定レジスタ) の設定値との計算値になります。

$$\text{補助速度}[\text{pps}] = \text{RFA} \times \frac{200}{(\text{RMG} + 1)}$$

7.4 ステータス処理

動作の終了はポーリングによるステータス読取りによって判断します。

ステータス読取りはモーションメインステータス (MMSTS)、イベントステータス (RIST)、エラーステータス (REST) によって動作終了、エラー発生など判断し次の動作に継続させます。

位置決め等において各モーションバスのこれらのステータスによって動作終了を判断します。

MMSTSに動作終了またはエラーが検出された時はRISTあるいはRESTを読み処理を終了します。

RISTは、RIRQに設定されたイベント報告の要因発生で変化します。

通常は初期化時にRIRQを設定をします。

(1) モーションメインステータス (MMSTS)

まず、初期化時にRIRQのb0を‘1’に設定します。(正常終了報告)

ステータスの監視は常に対象となるモーションバスのMMSTSに対して行います。

スタートコマンド書込み後はMMSTSをポーリングします。

b1を監視し、移動が終了するとb1が‘0’ ‘1’となります。

この時、正常終了ではb3が‘1’、異常終了ではb2が‘1’となります。

b2 = ‘1’の場合は、エラーステータスを読み出し、割り込みリセットコマンドを発行します。

b3 = ‘1’の場合は、イベントステータスを読み出し、割り込みリセットコマンドを発行します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	GRP2	GRP1	GRP0	0	0	0	SBSY	0	0	0	0	SEVT	SERR	SEND	SINT

図7.4-1 モーションメインステータス (MMSTS) のビット構成

bit	記号	機能	備考
0	SINT	MMSTSのbit1,2,3のいずれかが1で1になります。	
1	SEND	動作停止により1になります。RENV1のbit28=1の時スタート時リセットされます。または割り込みリセットコマンド(0008h)で0になります。	
2	SERR	エラー停止、位置のオーバーライド失敗、エンコーダ信号異常発生により1になります。REST読み出しで0になります。	
3	SEVT	RIRQで設定されたイベント発生により1になります。RIST読み出しで0になります。	
7~4	(未定義)	常に“0”	
8	SBSY	パルス出力開始で1になります。動作停止で0になります。(=BSY)	
11~9	(未定義)	常に“0”	
14~12	GRP2 ~ GRP0	グループ設定状態 無所属グループ:000,グループ1:001,グループ2:010 グループ3:011,グループ4:100,グループ5:101 グループ6:110,グループ7:111	
15	(未定義)	常に“0”	

表7.4-1 モーションメインステータス (MMSTS) の内容

b3が移動終了とは、RIRQ (イベント要因設定レジスタ) のb0 = 1 (正常終了) の場合です。

(2) イベントステータス (RIST)

RISTはMMSTS b3 (SEVT) = ‘1’を検出したとき必ず読まなければなりません。

イベントステータスは初期化においてRIRQ (イベント要因設定レジスタ) のb0 = 1 (正常終了) に設定して下さい。(基本的な運用時)

RISTを読み出し後、RISTは自動的に‘0’となり、MMSTSのb3も自動的に‘0’となります。

注: MMSTSはサイクリック通信で更新されますので、G9003の状態が反映されるまで最大サイクリック通信周期分の遅れが生じます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ISNP	ISNA	ISSA	ISSD	ISOL	ISLT	ISCL	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISEN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図7.4-2 RIST: イベントステータスレジスタのビット構成

b0 (ISEN): ‘1’ = 正常終了

(3) エラーステータス (REST)

RESTはMMSTS b4 (SERR) = '1'を検出したとき必ず読まなければなりません。

エラーが発生した時に対応するビットが"1"になります。エラーステータスはマスクすることは出来ません。

RESTを読み出し後、RESTは自動的に'0'となり、MMSTSのb2も自動的に'0'となります。

注：MMSTSはサイクリック通信で更新されますので、G9003の状態が反映されるまで最大サイクリック通信周期分の遅れが生じます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ESPE	ESEE	ESOR	0	ESNT	ESPO	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC3	ESC2	ESC1
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

図7.4-3 REST: エラーステータスレジスタのビット構成

bit	記 号	内 容	
0	E S C 1	コンパレータ1条件成立による停止時 (+ S L S)	
1	E S C 2	コンパレータ2条件成立による停止時 (- S L S)	
2	E S C 3	コンパレータ3条件成立による停止時	
3	E S P L	+ E L S入力ONによる停止時	
4	E S M L	- E L S入力ONによる停止時	
5	E S A L	S V A L M入力ONによる停止時	
6	E S S P	(予約)	
7	E S E M	(予約)	
8	E S S D	D L S入力ONによる減速停止時	
9	E S P O	手動パルスバッファカウンタのオーバーフロー発生時	
1 0	E S N T	通信エラー発生による停止時	
1 1	未定義	(常に0になります.)	
1 2	E S O R	位置のオーバライドが実行できなかった時	
1 3	E S E E	エンコーダ入力エラー発生時	注意：停止しない
1 4	E S P E	パルス入力エラー発生時	
31 ~ 15	未定義	(常に0になります.)	

表7.4-2 REST: エラーステータスレジスタの内容

< ステータス運用例 (RIRQのb0 = 1のみ (正常終了報告) 設定されている場合) >

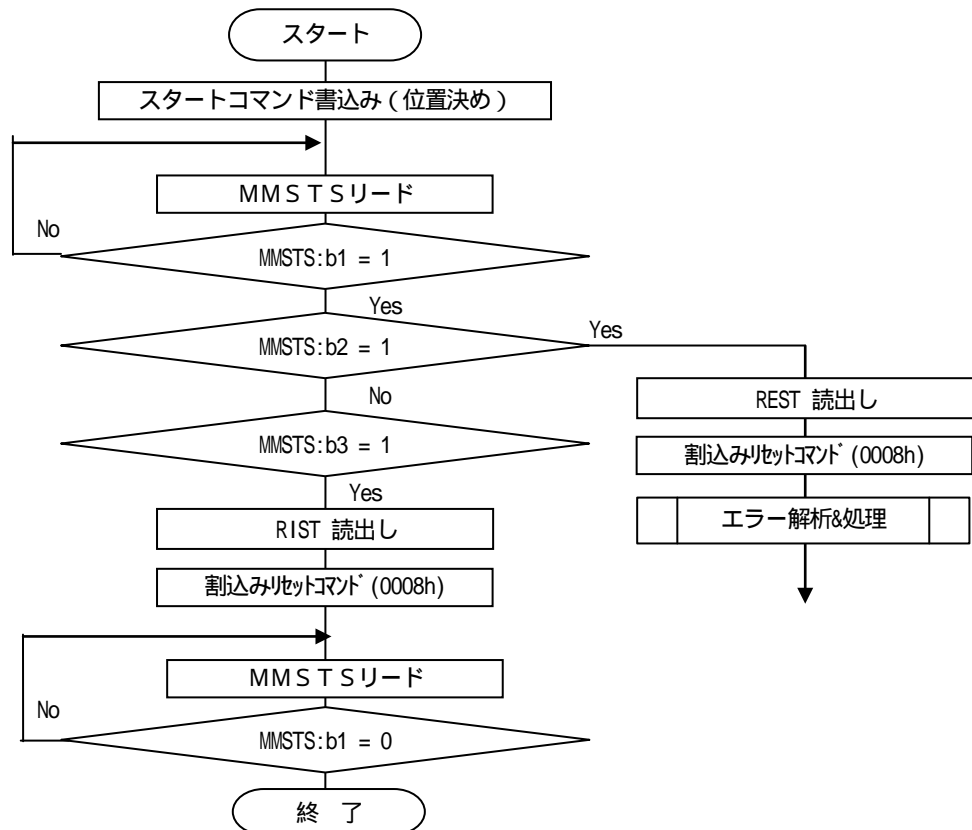


図7.4-4 ステータス運用例

7.5 応 用

7.5.1 STA入力を利用した同時スタート

モーションスレーブはSTA入力を利用してスレーブ内のモーションモジュールの同時スタートができます。
事前に同時にスタートしたいモーションモジュールのRMD(動作モード)のMSY(bit14) = "1"(STA入力待ち)に設定し、同時スタートコマンドでスタートします。
RIRQ(イベント要因)レジスタに設定する事により、同時スタート時(STA入力ON時)にイベントを発生させる事ができます。その時RISTレジスタを読み出すことにより、イベント発生要因を確認できます。
また、RSTS(拡張ステータス)レジスタを読み出すことにより、動作状態(STA入力待ち)、及びSTA入力状態をモニタできます。

<同時スタートの方法>

同時スタートさせるモーションモジュールのRMDレジスタのMSY(bit14) = "1"に設定後、スタートコマンドを書き込み、"STA入力待ち"状態にします。

その後、同時スタートさせる任意のモーションモジュールに同時スタートコマンド(0006h)の書き込み同時スタートします。

"STA入力待ち"状態の解除には、即停止コマンド(0049h)を書き込んで下さい。

STA入力待ちの設定	<RMDのMSY(bit14)に設定>
1: STA入力によりスタートします。	

STA信号の読み出し	<RSTSのSSTA(bit11)>
0: STA信号OFF	
1: STA信号ON	
動作状態の読み出し	<RSTSのCND(bit3~0)>
0001: STA入力待ち	
イベント要因の設定	<RIRQのIRSA(bit12)に設定>
1: STA入力ON時にイベントを発生します	
イベント要因の読み出し	<RISTのISSA(bit12)>
1: STA信号ON時	
同時スタートコマンド	<CMSTA:動作コマンド(0006h)>
自軸のみ同時スタートコマンド	<SPSTA:動作コマンド(002Ah)>
自軸のみにSTA信号が入力されたのと同等の処理を行う	

7.5.2 グループスタート

同じラインに接続されたモーションモジュールは予めグループ設定をすることでグループスタートができます。
事前にグループスタートしたいモーションモジュールをグループ設定し、またRMD(動作モード)のMSY(bit14) = "1"(STA入力待ち)に設定し、グループスタートコマンドでスタートします。
RIRQ(イベント要因)レジスタに設定する事により、グループスタート時にイベントを発生させる事ができます。
その時RISTレジスタを読み出すことにより、イベント発生要因を確認できます。
また、RSTS(拡張ステータス)レジスタを読み出すことにより、動作状態(STA入力待ち)をモニタできます。

<グループスタートの方法>

グループスタートさせるモーションモジュールをグループ設定します。その後のグループスタートさせるモーションモジュールのRMDレジスタのMSY(bit14) = "1"に設定後、スタートコマンドを書き込み、"STA入力待ち"状態にします。

その後、グループを接続しているセンターデバイスにグループスタートコマンド(2*01h: *はグループ番号)を発行します。

"STA入力待ち"状態の解除には、即停止コマンド(0049h)を書き込んで下さい。

グループ設定 <IOPOB の GRP2-GRP0(bit6-4)に設定>	
0 0 0 : グループ 7 , 0 0 1 : グループ 6 , 0 1 0 : グループ 5 , 0 1 1 : グループ 4 , 1 0 0 : グループ 3 , 1 0 1 : グループ 2 , 1 1 0 : グループ 1 , 1 1 1 : 無所属	
グループ設定状態 <MMSTS の GRP2-GRP0(bit14-12)に設定>	
0 0 0 : 無所属 , 0 0 1 : グループ 1 , 0 1 0 : グループ 2 , 0 1 1 : グループ 3 , 1 0 0 : グループ 4 , 1 0 1 : グループ 5 , 1 1 0 : グループ 6 , 1 1 1 : グループ 7	
S T A 入力待ちの設定 <RMD の MSY(bit14)に設定>	
0 1 : S T A 入力(またはグループスタートコマンド)によりスタートします.	
動作状態の読み出し <RSTS の CND(bit3~0)>	
0 0 0 1 : S T A 入力待ち	
イベント要因の設定 <RIRQ の IRNA(bit12)に設定>	
1 : グループスタート時にイベントを発生します	
イベント要因の読み出し <RIST の ISNA(bit12)>	
1 : グループスタート時	
グループスタートコマンド <CCMD>	
2 * 0 1 : * はグループ指定 (0 : 全グループ , 1 : グループ 1 , . . . , 7 : グループ 7)	

7.5.3 コンパレータ設定

モーションモジュールは、28ビットのコンパレータを3式内蔵しており、RCMP1~3レジスタ設定値とカウンタ値とを比較します。コンパレータは比較カウンタとしてCTR1~3を選択できます。

また、色々な比較方法と3種類の条件成立時の処理方法があります。

これらコンパレータの条件はRENV4(環境設定4)レジスタに設定します。

このコンパレータを利用することにより、

- (1) コンパレータ3比較結果の外部出力
- (2) 動作の即停止/減速停止
- (3) コンパレータ1, 2を利用した、ソフトリミット機能
- (4) CTR3(偏差)とコンパレータを利用した、ステッピングモータの脱調検出
- (5) CTR3(汎用)とコンパレータ3を利用した、定ピッチ出力
- (6) コンパレータ3を利用した同一スレーブ内他モジュールスタート

等が行えます。

(1) 比較データ

各コンパレータは比較データを下表の中から選択出来ます。

比較データ	コンパレータ1 C1C1~0	コンパレータ2 C2C1~0	コンパレータ3 C3C1~0
CTR1(指令位置)	"00"	"00"	"00"
CTR2(機械位置)	"01"	"01"	"01"
CTR3(汎用・偏差)	"10"	"10"	"10"
主な用途	+ S L S	- S L S	脱調検出, 等ピッチ出力

1. は比較可能
2. + S L S , - S L S はソフトリミットを表します。
3. 比較カウンタとして偏差カウンタに設定したCTR3を選択した場合には、カウンタ値の絶対値とコンパレータデータとの比較となります。(絶対値範囲: 0 ~ 32,767)
比較データの選択は、RENV4 の C1C1~0(bit1,0) , C2C1~0(bit9,8) , C3C1~0(bit17,16)にて行います。

(2) 比較方法

各コンパレータは比較方法を下表の中から選択できます。

比較方法	コンパレータ1 C1S2 ~ 0	コンパレータ2 C2S2 ~ 0	コンパレータ3 C3S3 ~ 0
コンパレータ = 比較カウンタ (カウント方向無関係)	"001"	"001"	"0001"
コンパレータ = 比較カウンタ (カウントアップ時のみ)	"010"	"010"	"0010"
コンパレータ = 比較カウンタ (カウントダウン時のみ)	"011"	"011"	"0011"
コンパレータ > 比較カウンタ	"100"	"100"	"0100"
コンパレータ < 比較カウンタ	"101"	"101"	"0101"
ソフトリミットとして使用	"110"	"110"	
定ピッチ出力 (カウント方向無関係)			"1000"
定ピッチ出力 (カウントアップ時のみ)			"1001"
定ピッチ出力 (カウントダウン時のみ)			"1010"

1. は比較可能、空欄は比較不可。
2. コンパレータ3のC3S3 ~ 0 = "0111"は設定禁止です。その他の設定値は常に比較条件不成立になります。
3. コンパレータ3のC3S3 ~ 0 = "1000" ~ "1010" <定ピッチ出力>の時には、比較カウンタはC T R 3 (汎用・偏差)を選択して下さい。他のカウンタは選択できません。
また、コンパレータの設定値は正の値に設定して下さい。
4. ソフトリミットとして使用する場合、コンパレータ1は(+)側リミット値になり、比較方法は"コンパレータ < 比較カウンタ"になります。また、コンパレータ2は(-)側リミット値になり、比較方法は"コンパレータ > 比較カウンタ"になります。比較カウンタはC O U N T E R 1 (指令位置)を選択して下さい。
5. 比較方法の選択は、RENV4 の C1S2 ~ 0(bit4 ~ 2), C2S2 ~ 0(bit12 ~ 10), C3S3 ~ 0(bit21 ~ 18) にて行います。

(3) コンパレータ条件成立時の処理方法

条件成立時の処理方法を下表の中から選択できます。

条件成立時の処理方法	コンパレータ1 C1D1 ~ 0	コンパレータ2 C2D1 ~ 0	コンパレータ3 C3D1 ~ 0
処理無し	"0 0"	"0 0"	"0 0"
動作の即停止	"0 1"	"0 1"	"0 1"
動作の減速停止	"1 0"	"1 0"	"1 0"
比較条件不成立	"1 1"	"1 1"	"1 1"

処理方法の選択は、RENV4 の C1D1 ~ 0(bit 6 ~ 5), C2D1 ~ 0(bit14 ~ 13), C3D1 ~ 0(bit23 ~ 22) にて行います。

イベント要因の設定 <RIRQ の IRC3 ~ 1(bit7 ~ 5)に設定>	
IRC1(bit5) = 1 : コンパレータ1条件成立時にイベントを発生。	
IRC2(bit6) = 1 : コンパレータ2条件成立時にイベントを発生。	
IRC3(bit7) = 1 : コンパレータ3条件成立時にイベントを発生。	
イベント要因の読み出し <RIST の ISC3 ~ 1(bit7 ~ 5)>	
IRC1(bit5) = 1 : コンパレータ1条件成立時。	
IRC2(bit6) = 1 : コンパレータ2条件成立時。	
IRC3(bit7) = 1 : コンパレータ3条件成立時。	
コンパレータ条件成立状態の読み出し <RSTS の SCP3 ~ 1(bit22 ~ 20)>	
SCP1(bit20) = 1 : コンパレータ1条件成立時。	
SCP2(bit21) = 1 : コンパレータ2条件成立時。	
SCP3(bit22) = 1 : コンパレータ3条件成立時。	
エラー割り込み要因の読み出し < REST の ESC3 ~ 1(bit2 ~ 0)>	
ESC1(bit0) = 1 : コンパレータ1条件成立による停止時。(+SLS)	
ESC2(bit1) = 1 : コンパレータ2条件成立による停止時。(-SLS)	
ESC3(bit2) = 1 : コンパレータ3条件成立による停止時。	

7.5.4 ソフトリミット設定

コンパレータ1と2を利用して、ソフトリミット機能が行えます。

コンパレータ1、2の比較カウンタには、CTR1(指令位置)を選択して下さい。

コンパレータ1を(+)側リミット値、コンパレータ2を(-)側リミット値として使用し、コンパレート結果と動作方向とから停止管理を行います。

ソフトリミット動作時には下記の処理を行う事ができます。

パルス出力を即停止させる。

パルス出力を減速停止させる。

ソフトリミット機能を使用する場合、コンパレート条件成立時の処理(C1D,C2D)として減速停止を選択すると、高速スタート(コマンド 0052h)中にソフトリミットに到達すると減速停止します。条件成立時の処理として他を選択した時や定速スタート中には即停止します。

また、スタートコマンド書き込み時にソフトリミットがON状態の場合、ソフトリミットがON状態の方向へはスタートは出来ません。逆方向へはスタートできます。

[設定例]

REN4=00003838h : コンパレータ1を(+)側、コンパレータ2を(-)側ソフトリミットに使用。

ソフトリミット動作時は即停止に設定。

RCMP1= 100,000 : (+)側リミット値

RCMP2= -100,000 : (-)側リミット値

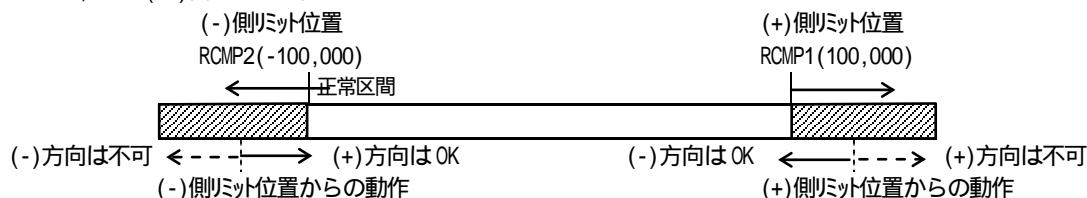


図 7.5 - 1 ソフトウェアリミット設定例

コンパレータ1の比較方法の設定 <REN4のC1S2~0(bit4~2)に設定>	
001: RCMP1データ = 比較カウンタ (カウント方向に無関係)	
010: " (カウントアップ中)	
011: " (カウントダウン中)	
100: RCMP1データ > 比較カウンタ	
101: " < "	
110: (+)側ソフトリミットとして使用 (RCMP1 < CTR1)	
その他: 常に比較条件不成立状態にします	
コンパレータ1条件成立時の処理の設定 <REN4のC1D1~0(bit6~5)に設定>	
01: 即停止	
10: 減速停止	
コンパレータ2の比較方法の設定 <REN4のC2S2~0(bit12~10)に設定>	
001: RCMP2データ = 比較カウンタ (カウント方向に無関係)	
010: " (カウントアップ中)	
011: " (カウントダウン中)	
100: RCMP2データ > 比較カウンタ	
101: " < "	
110: (-)側ソフトリミットとして使用 (RCMP2 > CTR1)	
その他: 常に比較条件不成立状態にします	
コンパレータ2条件成立時の処理の設定 <REN4のC2D1~0(bit14~13)に設定>	
01: 即停止	
10: 減速停止	

注: 太字の部分は設定例の部分です。

7.5.5 定ピッチ出力設定

コンパレータ3と汎用カウンタ設定のCTR3(汎用・偏差カウンタ)を使用して、一定間隔ごとにマシンI/FコネクタのCMP3端子へ信号を出力することができます。

C3C1~C3C0="10"(CTR3), C3S3~C3S0="1000"~"1010"(定ピッチ出力)にすると、定ピッチ出力設定になります。CTR3のカウンタ範囲は0~(RCMP3 設定値[$\text{MAX } 32,767$])になり、0からダウンカウントすると(RCMP3 設定値), (RCMP3 設定値)からアップカウントすると0になります。

CTR3の入力は、RENV3のCI32~CI30により選択できます。

CTR3(偏差)入力の設定		<RENV3のCI32~30(bit12~10)に設定>
000:	指令パルス	
001:	ENC入力	
011:	内部基準クロック(CLK=40MHz)の1/4096分周クロック	
100:	指令パルスとENC入力で偏差カウント	
コンパレータ3の比較カウンタの選択		<RENV4のC3C1~0(bit17~16)に設定>
00:	CTR1(指令位置)	10: CTR3(汎用・偏差)
01:	CTR2(機械位置)	11: 常に比較条件不成立
コンパレータ3の比較方法の設定		<RENV4のC3S3~0(bit21~18)に設定>
0001:	RCMP3データ = 比較カウンタ(カウント方向に無関係)	
0010:	"	(カウントアップ中)
0011:	"	(カウントダウン中)
0100:	RCMP3データ > 比較カウンタ	
0101:	RCMP3データ < 比較カウンタ	
0111:	設定禁止	
1000:	定ピッチ信号出力用として使用(カウント方向に無関係)	
1001:	"	(カウントアップ中)
1010:	"	(カウントダウン中)
その他:	常に比較条件不成立状態にします	

注: 太字の部分は定ピッチ出力での設定可能な部分を示します。

[出力例]

動作方向に関係なく、出力パルスにより定ピッチ信号を出力。カウンタ範囲は0~4。

設定値: RENV3=00000000h, RENV4=00220000h, RCMP4=4

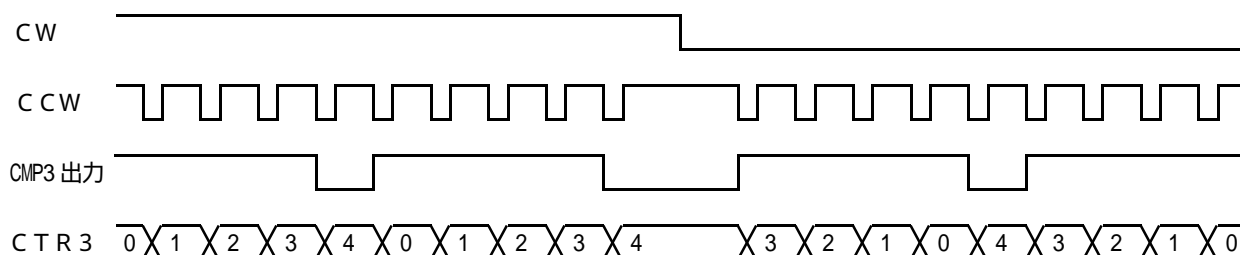


図7.5-2 定ピッチ出力

7.5.6 コンパレータ3を利用した同スレーブ内他モーションモジュールスタート

コンパレータ3と汎用カウンタ設定のCTR3(汎用・偏差カウンタ)を使用して、コンパレータ条件一致で同スレーブ内の他のモーションモジュールを起動することができます。

【 起動元となるモーションモジュールの設定 】

モーションモジュールのスイッチ設定
エンコーダ形式選択 ディップSW1 - 4をONにする

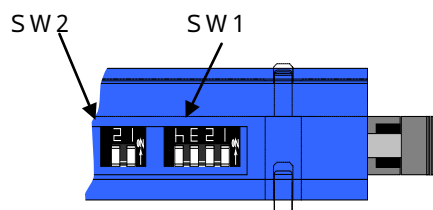


図7.5-3 モーションモジュール上面

CTR3(偏差)入力の設定 <REN3のCI32~30(bit12~10)に設定>	
000: 指令パルス	
001: ENC入力	
011: 内部基準クロック(CLK=40MHz)の1/4096分周クロック	
100: 指令パルスとENC入力で偏差カウント	
コンパレータ3の比較カウンタの選択 <REN4のC3C1~0(bit17~16)に設定>	
00: CTR1(指令位置)	10: CTR3(汎用・偏差)
01: CTR2(機械位置)	11: 常に比較条件不成立
コンパレータ3の比較方法の設定 <REN4のC3S3~0(bit21~18)に設定>	
0001: RCMP3データ = 比較カウンタ(カウント方向に無関係)	
0010: " (カウントアップ中)	
0011: " (カウントダウン中)	
0100: RCMP3データ > 比較カウンタ	
0101: RCMP3データ < 比較カウンタ	
0111: 設定禁止	
1000: 定ピッチ信号出力用として使用(カウント方向に無関係)	
1001: " (カウントアップ中)	
1010: " (カウントダウン中)	
その他: 常に比較条件不成立状態にします	

【 起動されるモーションモジュールの設定 】

STA入力待ちの設定 <RMDのMSY(bit14)に設定>
1: STA入力によりスタートします。

注: 表の太字の部分はコンパレータ条件一致で同スレーブ内の他モーションモジュールを起動する場合の設定可能な部分を示します。

<コンパレータ条件一致で同スレーブ内の他のモーションモジュールを起動する手順>

(通常の初期化、速度、移動量等は予め設定されているものとする)

- (1) 「起動元となるモーションモジュール」のコンパレータ条件設定。
 - (2) 「起動されるモーションモジュール」のRMDレジスタのMSY(bit14) = "1"に設定。
 - (3) 「起動されるモーションモジュール」にスタートコマンドを書き込み、"STA入力待ち"状態にする。
 - (4) 「起動元となるモーションモジュール」を起動。
 - (5) 「起動元となるモーションモジュール」のコンパレータ条件一致で「起動されるモーションモジュール」が起動される
- "STA入力待ち"状態の解除には、即停止コマンド(0049h)を書き込んで下さい。