

PCI Bus, ISA Bus, cPCI Bus,
PC/104, USB I/F

CPDボードシリーズ ユーザーズマニュアル 〈共通編〉

NCボード
多機能・高速 円弧・直線補間・位置決め



株式会社ハイバーテック

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

この説明書は

次のCPDシリーズ のボードに適応しています.

P C I	H P C I	—CPD532
	H P C I	—CPD534
	H P C I	—CPD508
	H P C I	—CPD578
	H P C I	—CPD5212M
I S A	H P C	—CPD234
	H P C	—CPD278
P C / 1 0 4 P C 1 0 4 D	H P C 1 0 4	—CPD132
	H P 1 0 4 D	—CPD364
U S B	H U S B	—CPD434
	H U S B	—CPD434 v 2
C o m p a c t P C I	H C P C I	—CPD734
	H C P C I	—CPD738

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下さい.

Windows 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 ハイバーテック
東京都江東区新大橋 1-8-11
三井生命新大橋ビル 6 F
TEL 03-3846-3801
FAX 03-3846-3773
sales@hivertec.co.jp

第 2. 8 版 2008 年 2 月 20 日発行
不許複製・転載

保証範囲

1. 本製品の保証期間は、お買い上げ頂いた日より3年間です。保証期間中に弊社の判断により欠陥が判明した場合には、本製品を弊社に引き取り、修理または交換を行います。
2. 保証期間内外に関わらず、弊社製品の使用、供給（納期）または故障に起因する、お客様及び第三者が被った、直接、間接、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、予めご了承下さい。

免責事項

1. 本マニュアルに記載された内容に沿わない、製品の取り付け、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、予めご了承下さい。
2. 本製品は、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す恐れのある装置（医療機器・交通機器・燃焼機器・安全装置等）に適用できるような設計、意図、または、承認、保証もされていません。

ゆえに本製品の安全性、品質および性能に関しては、本マニュアル（またはカタログ）に記載してあること以外は明示的にも黙示的にも一切保証するものではありませんので、予めご了承下さい。
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、予めご了承下さい。
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、予めご了承下さい。

マニュアル構成

CPDシリーズのボードには次のマニュアルが添付されています。

1. (個別ボード名) ユーザーズマニュアル < 個別ボード編 >

個々のCPDボードについて、次の項目について説明しています。

- (1) ハードウェアに関する情報
- (2) 添付ソフトウェアのインストール方法
- (3) サンプルソフトの操作
- (4) 「動かしてみる」の操作
- (5) その他ボード固有な機能

2. CPDボードシリーズ ユーザーズマニュアル < ソフトウェア編 >

CPDボードシリーズの次のソフトウェアについて説明しています。

- (1) ライブラリ関数 (ライブラリ関数レベル1: VC++, VB, DOS)
- (2) ドライバ関数 (デバイスドライバI/F用ライブラリ: VC++, VB, DOS)

3. CPDボードシリーズ ユーザーズマニュアル < 共通編 >・・・本マニュアル

CPDボードシリーズに共通した部分について、チュートリアル形式で説明をしています。

- (1) CPDボードの基本的な運用方法 (ライブラリ関数でサンプル表記)
- (2) CPDボードのより応用的な解説
(ドライバ関数でサンプル表記・・・より自在な運用をするためには、参照する必要があります)
- (3) PCL6045, PCL6025及び同等品に基づいた各種レジスタ説明
- (4) その他

目 次

1. はじめに	1
1. 1 このマニュアルについて	1
1. 2 ボードの種類	2
1. 2. 1 使用可能軸と軸名称	2
1. 2. 2 CPDシリーズ機能比較	2
1. 2. 3 軸動作機能	3
1. 2. 4 ボード軸数と軸間の動作組合せ	3
1. 3 添付ソフトウェア	4
1. 3. 1 型式による関数名と説明記述	4
2. 用語解説	5
3. ボード構成とポートアドレス	7
3. 1 ブロック図	7
3. 2 ポートアドレス	8
3. 2. 1 ボードアドレス	8
3. 2. 2 オプションポート	9
3. 2. 3 ボード入出力とデバイスドライバ	9
3. 3 ポートとレジスタ配置	10
4. 基本的な設定と運用	11
4. 1 プログラムの手順	11
4. 2 初期設定	12
4. 2. 1 各種初期設定	13
4. 2. 2 オプションポートの設定 (ELS極性, DLS/PCS選択)	14
4. 2. 3 サーボインターフェースの設定	15
4. 2. 4 原点センサと原点復帰方法の設定	16
4. 2. 5 速度および加減速関係の初期設定	19
4. 3 運 用	22
4. 3. 1 動作モードレジスタ RMD (PRMD) と動作モード設定	22
4. 3. 2 動作速度設定	29
4. 3. 3 移動量設定	30
4. 3. 4 動作コマンド	31
4. 3. 5 その他 (SVON, SVRST)	32
4. 4 スタートコマンド発行後のステータス処理	33
4. 4. 1 ステータス構成	33
4. 4. 2 補間の場合のステータス	34
4. 4. 3 ライブラリ関数による終了ステータス処理例	34
4. 5 各動作のサンプルコーディング	35
4. 5. 1 原点復帰	35
4. 5. 2 位置決め	36
4. 5. 3 連続送りと停止コマンド	36
4. 5. 4 タイマー動作	37
5. ポートおよびレジスタの書き込み, 読出し	37
5. 1 CMD.BUFx書き込み, 読出し方法	37
5. 1. 1 データを伴わないCMD形式	38
5. 1. 2 レジスタ書き込み, 読出し	38
5. 1. 3 MST S、SST Sの読出し	40
6. ステータス解説	41
6. 1 メインステータス MST S (読出し専用)	41
6. 2 サブステータス SST S (読出し専用)	42
6. 3 エラーステータスレジスタ REST (読出し専用)	42
6. 4 イベントステータスレジスタ RIST (読出し専用)	44
6. 5 イベントマスクレジスタ RIRQ	45
6. 6 拡張ステータスレジスタ RST S (読出し専用)	46
6. 7 補間ステータスレジスタ RIPS (読出し専用)	47
6. 8 ステータス運用解説	48

6. 8. 1	イベント報告方式	48
6. 8. 2	補間の場合のステータス運用	50
6. 9	割り込み処理方法	51
7.	コマンド	52
7. 1	動作コマンドおよび制御コマンド	52
7. 2	動作コマンド（同時スタートコマンド，同時ストップコマンド）	53
7. 3	加速ブロック，定速ブロック，減速ブロック コマンド	53
7. 4	残量スタート コマンド	54
7. 5	カウンタ制御機能と コマンド	54
7. 6	サーボ偏差カウンタクリア機能とコマンド	54
8.	レジスタ制御コマンド および レジスタ解説	55
8. 1	レジスタ制御コマンド	55
8. 2	RMV（PRMV） 移動量レジスタ	56
8. 3	RFL（PRFL） ベース速度レジスタ	56
8. 4	RFH（PRFH） 動作速度レジスタ	56
8. 5	RUR（PRUR） 加速レートレジスタ	56
8. 6	RDR（PRDR） 減速レートレジスタ	56
8. 7	RMG（PRMG） 速度倍率レジスタ	57
8. 8	RDP（PRDP） 減速開始点レジスタ	57
8. 9	RMD（PRMD） 動作モードレジスタ	57
8. 10	RIP（PRIP） 円弧補間中心位置レジスタ	57
8. 11	RUS（PRUS） および RDS（PRDS） 加速および減速S字区間レジスタ	58
8. 12	RFA 補助速度レジスタ	58
8. 13	RCTR1 指令位置カウンタ・レジスタ	58
8. 14	RCTR2 機械位置カウンタ・レジスタ	58
8. 15	RCTR3 脱調カウンタ・レジスタ	59
8. 16	RCTR4 汎用カウンタ・レジスタ	59
8. 17	RCMP1 コンパレータ1比較レジスタ	59
8. 18	RCMP2 コンパレータ2比較レジスタ	59
8. 19	RCMP3 コンパレータ3比較レジスタ	60
8. 20	RCMP4 コンパレータ4比較レジスタ	60
8. 21	RCMP5（PCMP5）コンパレータ5比較レジスタ	60
8. 22	RLTC1～4 カウンタラッチレジスタ（読出し専用）	60
8. 22. 1	RLTC1，2，4のレジスタ形式（読出し専用）	60
8. 22. 2	RLTC3（脱調カウンタ，現在速度用）ラッチ レジスタ形式（読出し専用）	61
8. 23	RPLS 移動残パルスレジスタ（読出し専用）	61
8. 24	RSPD 現在速度，その他読出しレジスタ（読出し専用）	61
8. 25	RIPS 補間ステータスレジスタ（読出し専用）	61
9.	環境設定レジスタ	62
9. 1	RENV1 レジスタ	62
9. 2	RENV2 レジスタ	63
9. 3	RENV3 レジスタ	63
9. 4	RENV4 レジスタ	65
9. 5	RENV5 レジスタ	66
9. 6	RENV6 レジスタ	67
9. 7	RENV7 レジスタ	67
10.	各機能と応用	68
10. 1	速度と加減速	68
10. 1. 1	速度，加減速の計算	68
10. 1. 2	急加減速時の注意点	70
10. 2	速度のオーバーライド	71
10. 3	位置のオーバーライド	72
10. 4	PCS「位置決め管理開始信号入力」	73
10. 5	指定軸停止で起動（条件付他軸スタート—1）	74
10. 6	カウンタ および コンパレータ	75
10. 6. 1	カウンタ	75
10. 6. 2	コンパレータ	76

10. 6. 3	ソフトリミット機能.....	76
10. 6. 4	コンパレータ条件成立で他軸起動（条件付他軸スタート—2）.....	77
10. 7	次動作連続実行（プリレジスタ連続使用）.....	79
10. 8	複数のPCL間の動作方法.....	81
10. 9	手動パルス入力モード使用方法.....	82
11.	CPDシリーズ追加機能について.....	83
11. 1	各種レジスタ内の追加ビット.....	83
11. 2	追加機能詳細.....	84

図 表 目 次

1. はじめに	
表 1. 1-1 CPDボードシリーズ製品ラインアップ	1
表 1. 1-2 付属される説明資料	1
表 1. 2-1 使用可能軸と軸名称	2
表 1. 2-2 CPDシリーズ機能比較	2
表 1. 2-3 CPDボード軸数と軸間の動作組合せ	3
表 1. 3-1 ライブラリファイル対応表	4
表 1. 3-2 ライブラリファイル対応表	4
2. 用語解説	
図 2. 1-1 速度パターン図（直線加減速、原点復帰の例）	5
図 2. 1-2 軸センサ（直動軸）	6
図 2. 1-3 指令パルスとモータ・モータドライバ	6
3. ボード構成とボードアドレス	
図 3. 1-1 HPCI-CPD534ブロック図	7
表 3. 2-1 HPCI-CPD578, 534, 532ポート表	8
表 3. 2-2 HPCI-CPD578, 534, 532のボード入出力とドライバ関数	9
図 3. 3-1 HPCI-CPD578, 534, 532のポートとレジスタ配置	10
4. 基本的な設定と運用	
図 4. 1-1 プログラム手順概要	11
表 4. 2-1 DEVOPEN関数における初期化内容	12
表 4. 2-2 HPCI-CPD534, 532のオプションポート設定要件	14
表 4. 2-3 サーボインターフェースの設定要件	15
表 4. 2-4 原点復帰方法選択表	17
表 4. 2-5 原点復帰方法選択表（CTR2参照方式）	18
表 4. 2-6 原点復帰方法選択表（センサ原点DLS方式）	18
表 4. 2-7 速度関係のレジスタ設定	19
図 4. 2-1 速度パターンとレジスタの対応	19
表 4. 2-8 速度倍率設定例	20
図 4. 3-1 RMDレジスタのビット構成	22
表 4. 3-1 RMDレジスタの内容	22
表 4. 3-2 動作モード一覧	24
表 4. 3-3 連続動作モード	24
表 4. 3-4 位置決め動作モード	25
表 4. 3-5 補間動作モード	26
表 4. 3-6 原点復帰動作モード	27
表 4. 3-7 ELS, SLS 動作モード	27
表 4. 3-8 Z相移動モード	27
表 4. 3-9 パルサ動作モード	28
表 4. 3-10 JOG送り動作モード	28
表 4. 3-11 動作コマンド	31
表 4. 3-12 スタートコマンドの速度動作	31
表 4. 3-13 SVON, SVRSTポートの操作	32
図 4. 4-1 メインステータス MSTSのビット構成	33
図 4. 4-2 イベントステータス RISTのビット構成	33
図 4. 4-3 エラーステータス RESTのビット構成	34
5. ポートおよびレジスタの書込み、読出し	
図 5. 1-1 CMDポートの形式	38
図 5. 1-2 動作レジスタ構成	38
図 5. 1-3 レジスタ書込み、読出しのCMD, BUF0, BUF1の形式	39
図 5. 1-4 メインステータス読出し形式	40
図 5. 1-5 サブステータス読出し形式	40
6. ステータス解説	
表 6. 1-1 メインステータス MSTSの内容	41

表 6. 2-1	サブステータス SSTSの内容	4 2
図 6. 3-1	エラーステータスレジスタ RESTのビット構成	4 2
表 6. 3-1	エラーステータスレジスタ RESTの内容	4 3
図 6. 4-1	イベントステータスレジスタ RISTのビット構成	4 4
表 6. 4-1	イベントステータスレジスタ RISTの内容	4 4
図 6. 5-1	イベントマスクレジスタ RIRQのビット構成	4 5
表 6. 5-1	イベントマスクレジスタ RIRQの内容	4 5
図 6. 6-1	拡張ステータスレジスタ RSTSのビット構成	4 6
表 6. 6-1	拡張ステータスレジスタ RSTSの内容	4 6
図 6. 7-1	補間ステータスレジスタ RIPSのビット構成	4 7
表 6. 7-1	補間ステータスレジスタ RIPSの内容	4 7
図 6. 8-1	イベント報告方式のステータス運用フロー例	4 9
図 6. 8-2	補間動作時のステータス運用フロー例	5 0
図 6. 9-1	ボード内割込みルート	5 1
7. コマンド		
表 7. 1-1	動作コマンドおよび制御コマンド一覧	5 2
図 7. 3-1	線分の連続と加速BLK, 定速BLK, . . . , 減速BLK	5 3
図 7. 3-2	加速BLK, 定速BLK, 減速BLK	5 3
図 7. 6-1	サーボ偏差カウンタクリア信号とタイミング関係	5 4
8. レジスタ制御コマンドおよびレジスタ解説		
表 8. 1-1	レジスタコマンド一覧	5 5
図 8. 2-1	RMV (PRMV) 移動量レジスタ	5 6
図 8. 3-1	RFL (PRFL) ベース速度レジスタ	5 6
図 8. 4-1	RFH (PRFH) 動作速度レジスタ	5 6
図 8. 5-1	RUR (PRUR) 加速レートレジスタ	5 6
図 8. 6-1	RDR (PRDR) 減速レートレジスタ	5 6
図 8. 7-1	RMG (PRMG) 速度倍率レジスタ	5 7
図 8. 8-1	RDP (PRDP) 減速開始点レジスタ	5 7
図 8. 9-1	RMD (PRMD) 動作モードレジスタ	5 7
図 8. 10-1	RIP (PRIP) 円弧補間中心レジスタ	5 7
図 8. 11-1	RUS (PRUS) およびRDS (PRDS) S字区間レジスタ	5 8
図 8. 12-1	RFA 補助速度レジスタ	5 8
図 8. 13-1	RCTR1 指令位置カウンタ・レジスタ	5 8
図 8. 14-1	RCTR2 機械位置カウンタ・レジスタ	5 8
図 8. 15-1	RCTR3 脱調カウンタ・レジスタ	5 9
図 8. 16-1	RCTR4 汎用カウンタ・レジスタ	5 9
図 8. 17-1	RCMP1 コンパレータ1比較レジスタ	5 9
図 8. 18-1	RCMP2 コンパレータ2比較レジスタ	5 9
図 8. 19-3	RCMP3 コンパレータ3比較レジスタ	6 0
図 8. 20-1	RCMP4 コンパレータ4比較レジスタ	6 0
図 8. 21-1	RCMP5 コンパレータ5比較レジスタ	6 0
図 8. 22-1	RLTC1, 2, 4のカウンタラッチレジスタ	6 0
図 8. 22-2	RLTC3 のカウンタラッチレジスタ	6 1
図 8. 23-1	RPLS 移動残パルスレジスタ	6 1
図 8. 24-1	RSPD 現在速度レジスタ	6 1
表 8. 24-1	RSPD 現在速度レジスタ内容	6 1
9. 環境設定レジスタ		
図 9. 1-1	環境レジスタ RENV1	6 2
表 9. 1-1	環境レジスタ RENV1の内容	6 2
図 9. 2-1	環境レジスタ RENV2	6 3
表 9. 2-1	環境レジスタ RENV2の内容	6 3
図 9. 3-1	環境レジスタ RENV3	6 3
表 9. 3-1	環境レジスタ RENV3の内容	6 4
図 9. 4-1	環境レジスタ RENV4	6 5
表 9. 4-1	環境レジスタ RENV4の内容	6 5
図 9. 5-1	環境レジスタ RENV5	6 6
表 9. 5-1	環境レジスタ RENV5の内容	6 6

図 9. 6-1	環境レジスタ RENV6	67
表 9. 6-1	環境レジスタ RENV6の内容	67
図 9. 7-1	環境レジスタ RENV7	67
表 9. 7-1	環境レジスタ RENV7の内容	67
10. 各機能と応用		
図 10. 1-1	正しい条件の自動加減速	68
図 10. 1-2	不適正な加減速の場合	68
図 10. 1-3	加減速動作指令と実際の速度制御	70
表 10. 2-1	自動加減速における速度オーバーライド条件	71
図 10. 3-1	位置のオーバーライド	72
表 10. 6-1	カウンタ機能と選択	75
表 10. 6-2	エンコーダカウント周波数	75
図 10. 6-1	各カウンタのカウント入力ルート案内図	75
表 10. 6-3	コンパレータの標準的使用	76
図 10. 6-2	ソフトリミット	77
図 10. 6-3	条件付他軸起動	77
図 10. 9-1	パルス入力	82
11. CPDシリーズ追加機能について		
表 11. 1-1	追加機能一覧表	83
表 11. 1-2	環境レジスタ2追加ビット	83
表 11. 1-3	環境レジスタ4追加ビット	83
表 11. 1-4	環境レジスタ5追加ビット	84
表 11. 2-1	指定軸停止の設定レジスタ	84
表 11. 2-2	指定軸停止の設定詳細	84
図 11. 2-1	指定軸停止による起動	85
表 11. 2-3	マシンロックの設定レジスタ	85
表 11. 2-4	カウンタラッチ後のクリア設定レジスタ	85
表 11. 2-5	回転軸カウンタ機能の設定レジスタ	85

1. はじめに

このマニュアルは、制御LSIにPCL6045（4軸）、PCL6025（2軸）及び同等品（以降は、これらLSIの総称として“PCL”を使用します。）を使用した高速・高機能・円弧、直線補間、位置決めモーター制御ボード「CPDボードシリーズ」の共通取扱説明書です。

「CPDボードシリーズ」はPCI Bus、ISA Bus、Compact PCI Bus、PC/104の各Bus、及びUSB I/F に対して次の製品ラインアップがあります。

軸数	PCI	ISA	Compact PCI	PC/104	PC104D (PC/104)	USB
2 軸	HPCI—CPD532			HPC104—CPD132		
4 軸	HPCI—CPD534	HPC—CPD234	HCPCI—CPD734		HP104D—CPD364	HUSB—CPD434 HUSB—CPD434v2
8 軸	HPCI—CPD578 HPCI—CPD508	HPC—CPD278	HCPCI—CPD738			
12 軸	HPCI—CPD5212M					

表 1. 1-1 CPDボードシリーズ製品ラインアップ

製品には 通常次の説明資料が付属します。

1	CPDシリーズユーザーズマニュアル〈共通編〉	このマニュアル
2	CPDシリーズユーザーズマニュアル〈ソフトウェア編〉	ライブラリ関数の説明
3	（個別ボード名）ユーザーズマニュアル〈個別ボード編〉	製品のハードウェアの説明 ソフトウェアスタートアップ説明
4	添付ソフトウェア（CDまたはFD）	

表 1. 1-2 付属される説明資料

■Bus タイプ、ポート構成、軸数などによって異なる部分の機能説明、コネクタ、ボード上の設定およびソフトウェアのインストール、サンプルプログラムの操作などは

別冊「ユーザーズマニュアル〈個別ボード編〉」を併せて参照して下さい。

■CPDシリーズ製品に共通するライブラリ関数・ドライバ関数（VC++、VB）の説明は

別冊「CPDシリーズ ユーザーズマニュアル〈ソフトウェア編〉」を併せて参照して下さい。

1. 1 このマニュアルについて

この取扱い説明は 標準的な 4 軸CPDボード HPCI—CPD534 を基本に説明します。

■次の章は 是非一読してください。

2 章 用語解説

3 章 ボード構成とポートアドレス

【他型式ボードは個別ユーザーズマニュアル参照】

■通常の動作は さらに4章をご覧ください。

4 章を把握の上、添付ソフトウェアマニュアルをご利用ください。

■モータドライバ、軸センサの接続は製品型式によります。

【他型式ボードは個別ユーザーズマニュアル参照】

■より広範な利用は5章以降もご覧下さい。

1. 2 ボードの種類

1. 2. 1 使用可能軸と軸名称

ボードの軸数（2，4，8，12）と使用可能の名称は次表の通りです。

<div> <div>軸名称上段：CPD5212M以外</div> <div>下段：CPD5212M</div> </div> <div>ボード種類</div>	X	Y	Z	U	V	W	A	B	—	—	—	—
	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
HPCI-CPD532, HPCI104-CPD132	○	○	×	×	×	×	×	×	×	×	×	×
HPCI-CPD534, HPC-CPD234, HUSB-CPD434, HUSB-CPD434v2, HCPCI-CPD734, HP104D-CPD364	○	○	○	○	×	×	×	×	×	×	×	×
HPCI-CPD578, HPCI-CPD508, HCPCI-CPD738, HPC-CPD278	○	○	○	○	○	○	○	○	×	×	×	×
HPCI-CPD5212M	○	○	○	○	○	○	○	○	○	○	○	○

※ ○：使用可，×：使用不可

表 1. 2-1 使用可能軸と軸名称

1. 2. 2 CPDシリーズ機能比較

BUS	型式	軸数	エンコーダ入力	サーボI/F				マシンI/F	
				出力			入力	DLS	PCS
				SVON	SVRST	CTRCL	INPOS		
PCI	HPCI-CPD532	2	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
	HPCI-CPD534	4	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
	HPCI-CPD508	8	×	8点		×	×	—	BOLS と切替
	HPCI-CPD578	8	差動 4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
	HPCI-CPD5212M	12	差動 6.5MHz(x4)(但し 4ch)	8点		○	○	×	×
ISA	HPC-CPD234	4	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
	HPC-CPD278	8	差動 6.5MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
CPCI	HCPCI-CPD734	4	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
	HCPCI-CPD738	8	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
PC/104	HPCI104-CPD132	2	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
	HP104D-CPD364	4	差動 6.5MHz(x4)	○	○	○	○	PCS と切替	DLS と切替
USB	HUSB-CPD434 HUSB-CPD434v2	4	4MHz(x4)	○	○	○	○	PCS と切替	DLS と切替

BUS	型式	軸数	CMP4→STA 外部出力	CMP5→STP 外部出力	CMP5→STP 内部接続	コンパレータ出力	汎用DIO	備考
PCI	HPCI-CPD532	2	オプション	オプション	○	STA/STP で代用	×	ショートサイズ
	HPCI-CPD534	4	オプション	オプション	○	STA/STP で代用	×	ショートサイズ
	HPCI-CPD508	8	×	×	×	×	×	ショートサイズ
	HPCI-CPD578	8	○	○	○	○	×	ショートサイズ
	HPCI-CPD5212M	12	オプション	オプション	○	X1-U1 は標準	×	
ISA	HPC-CPD234	4	オプション	オプション	○	STA/STP で代用	×	ハーフサイズ
	HPC-CPD278	8	オプション	オプション	○	○	×	ハーフサイズ
CPCI	HCPCI-CPD734	4	×	×	○	×	×	3U サイズ
	HCPCI-CPD738	8	オプション	オプション	○	STA/STP で代用	×	6U サイズ
PC/104	HPCI104-CPD132	2	×	×	○	×	×	
	HP104D-CPD364	4	○	○	○	○	16i/16o	
USB	HUSB-CPD434 HUSB-CPD434v2	4	×	×	○	×	×	

表 1. 2-2 CPDシリーズ機能比較

1. 2. 3 軸動作機能

1 式の 4 軸用 P C L の軸動作機能として

- ① 独立軸動作
- ② 1 組の 2 ～ 4 軸直線補間機能
- ③ 1 組の 2 軸円弧補間機能

があります。

1 式の 2 軸用 P C L の軸動作機能として

- ① 独立軸動作
- ② 1 組の 2 軸直線補間機能
- ③ 1 組の 2 軸円弧補間機能

があります。

ここで「独立軸動作」には次の各動作が含まれます。

位置決め (P T P) , 連続送り (停止モードにより終了) ,
原点復帰, 手動パルス送り (手動ハンドル送り) , タイマー動作

以降の動作組合せ表において「同時動作」とは次のことをいいます。

- (1) 複数の独立軸が同時期に動作する. 例: X 動作中, Y 軸が途中から動作.
- (2) 複数組の補間軸が同時期に動作する. 例: X Y 円弧補間中, Z U 直線補間が途中から動作.
- (3) 独立軸と補間軸が同時期に動作する. 例: X Y Z 直線補間中, U 軸が独立軸動作する.

1. 2. 4 ボード軸数と軸間の動作組合せ

2 軸ボードには 2 軸用 P C L が 1 個

4 軸ボードには 4 軸用 P C L が 1 個

8 軸ボードには 4 軸用 P C L が 2 個

1 2 軸ボードには 4 軸用 P C L が 3 個 搭載されています。

これらの P C L は基本的に互いに独立です。

例外的に直線補間において同時スタートコマンドを使用することにより, 複数 P C L 間の直線補間が出来ます。ただし, 加減速を伴わない定速送りにおいて使用します。

ボードの軸数 (2, 4, 8, 1 2) と軸間動作組合せは次表の通りです。

軸数	項	軸動作組合せ	X, Y	X ~ U X1 ~ U1	V ~ B X2 ~ U2	X3 ~ U3	同時スタート コマンド 使用
2 軸	1	全て独立	同時動作可	—	—	—	—
	2	直線補間	直線補間可				
	3	円弧補間	円弧補間可				
4 軸	1	全て独立	—	同時動作可	—	—	—
	2	全て直線補間	—	2 ~ 4 直線補間可			
	3	直線補間と独立	—	同時動作可			
	4	円弧補間と残軸	—	円弧補間 と残軸の 同時動作可			
8 軸	1	全て独立	—	同時動作可	左の動作と関係 なく 1 ~ 4 の 各動作組合可	—	5 ~ 8 軸 直線補間可 (但し定速)
	2	全て直線補間	—	2 ~ 4 直線補間可			
	3	直線補間と独立	—	同時動作可			
	4	円弧補間と残軸	—	円弧補間と残軸の 同時動作可			
12 軸	1	全て独立	—	同時動作可	左の動作と関係 なく 1 ~ 4 の 各動作組合可	左の動作と関係 なく 1 ~ 4 の 各動作組合可	5 ~ 1 2 軸 直線補間可 (但し定速)
	2	全て直線補間	—	2 ~ 4 直線補間可			
	3	直線補間と独立	—	同時動作可			
	4	円弧補間と残軸	—	円弧補間と残軸の 同時動作可			

表 1. 2 - 3 C P D ボード軸数と軸間の動作組合せ

1. 3 添付ソフトウェア

CPDボードには次のソフトウェアが添付されます。

- Windows版ライブラリ関数(レベル1)・・・基本的な動作に必要な関数集を使用出来ます。
- Windows版デバイスドライバ・・・個々のOS毎にデバイスドライバが提供されます。
- Windows版ドライバ関数・・・高度なアプリケーション向きにはドライバ関数から利用(I/F用DLL)出来ます。
- Windows版 サンプルプログラム・・・ライブラリ関数の使用法を解説するサンプルソフトです。
- 「動かしてみる」・・・ボードをパソコンへ装着するだけで、最小限の動作をします。接続確認にも利用できます。

1. 3. 1 型式による関数名と説明記述

このマニュアルのソフトウェア例の個所はHPCI-CPD534の添付ソフトウェアに基づいた説明をしています。

【CPD534ライブラリ関数による記述例】・・・ライブラリ関数による記述箇所

【CPD534ドライバ関数による記述例】・・・ドライバ関数による記述箇所

ライブラリおよびドライバはボードのBusタイプにより関数名が異なります。

CPD534のソフトウェア記述例はボード型式に対応した関数名で置き換えて解釈してください。

以下に、ライブラリファイル対応表(表1. 3-1)と関数名規約(表1. 3-2)を示します。

(1) ライブラリファイル対応表

ボード種別		HPCI		HPC		HUSB	HCPCI	HPC104	HP104D
ボード		CPD534 CPD532 CPD578 CPD508	CPD5212M	CPD234	CPD278	CPD434 CPD434v2	CPD734 CPD738	CPD132	CPD364
Windows版	ライブラリ関数	VC++ソース 拡張子.c	cp53011a	cp52c11a	cp23411a	cp27811a	cp43011a	cp73011a	cp13211a
		VC++ヘッダー 拡張子.h	cp53011a	cp52c11a	cp23411a	cp27811a	cp43011a	cp73011a	cp13211a
		VB標準モジュール 拡張子.bas (.NETの場合.vb)	cp53011a	cp52c11a	cp23411a	cp27811a	cp43011a	cp73011a	cp13211a
	ドライバ関数	VC++インポートライブラリ 拡張子.lib	hicpd530	hicpd52c	hcpd230		hucpd430	hccpd730	hcpd130
		VC++関数結合用ヘッダー 拡張子.h	hicpd530	hicpd52c	hcpd230		hucpd430	hccpd730	hcpd130
		VB関数定義標準モジュール 拡張子.bas(.NETの場合.vb)	hicpd530	hicpd52c	hcpd230		hucpd430	hccpd730	hcpd130
DOS版	ドライバ関数	ドライバ I/F用DLL 拡張子.dll	hicpd530	hicpd52c	hcpd230		hucpd430	hccpd730	hcpd130
	ライブラリ関数	ソース 拡張子.c	cp53011a	cp52c11a	cp23411a	cp27811a.c	Windows版のみ		cp13011a
		ヘッダ 拡張子.h	cp53011a	cp52c11a	---	---			cp13011a
	ライブラリ関数	ラージ 拡張子.lib	Licpd530	Licpd52c	Lcpd230	Lcpd270			Lcpd130
		コンパクト 拡張子.lib	Cicpd530	Cicpd52c	Ccpd230	Ccpd270			Ccpd130
		ミディアム 拡張子.lib	Micpd530	Micpd52c	Mcpd230	Mcpd270			Mcpd130
		スモール 拡張子.lib	Sicpd530	Sicpd52c	Scpd230	Scpd270	Windows版のみ		Scpd130
		ヘッダー 拡張子.h	hicpd530 hcpdtype	hicpd52c hcpdtype	hcpd230	hcpd270			hcpd130

表 1. 3-1 ライブラリファイル対応表

(2) 関数名規約

ボード種別		HPCI		HPC		HUSB	HCPCI	HPC104	HP104D
ボード		CPD532 CPD534 CPD578 CPD508	CPD5212M	CPD234	CPD278	CPD434 CPD434v2	CPD734 CPD738	CPD132	CPD364
関数名	ライブラリ関数	hcp530_xxx()	hcp52c_xxx()	hcp234_xxx()	hcp278_xxx()	hcp430_xxx()	hcp730_xxx()	hcp132xxx()	hcp364xxx()
	ドライバ関数	cp530_xxx()	cp52c_xxx()	cp230_xxx()	cp230_xxx()	cp430_xxx()	cp730_xxx()	cp130_xxx()	cp360_xxx()
	備考	Windows版のみ							

表 1. 3-2 ライブラリファイル対応表

2. 用語解説

- 制御軸 このボードで制御されるモータ（サーボモータ、ステップモータなど）は「制御軸」といい、他種のモータ（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します。制御軸を指示する場合X、Y、Z、Uなどと呼びます。
- 独立軸制御 軸間には何の関係もなく動作するモードの総称。位置決め、原点復帰、連続送り、タイマー動作（ドゥエル）の各動作があります。
- 連続送り 停止位置は指定せず、速度のみ指示して送る。停止コマンドによって停止します。
- 位置決め制御 途中の経路は問題とせず、目標物を早く移動し目的点に位置付けする制御をいいます。速度指定は軸ごとに独立で、各軸にあたえます。
- 補間制御 目標物の移動が希望するプロファイルを描きながら目標点に移動する制御をいいます。速度制御は輪郭速度（線速度、合成速度）が一定になるように制御されます。速度指定は軸数によらず1組の線速度で与えます。CPDボードでは直線補間、円弧補間ができます。
- 補間代表軸 X～U軸の並びで1組の円弧補間または直線補間を行う場合、X軸かX軸に最も近い軸をいいます。8軸ボードでV～B軸使用の円弧補間または直線補間では、V軸かV軸に最も近い軸となります。同様に12軸ボードではX1～U1軸使用の円弧補間または直線補間では、X1軸かX1軸に最も近い軸、X2～U2軸ではX2軸またはX2軸に最も近い軸、X3～U3軸ではX3軸かX3軸に最も近い軸となります。この代表軸には“補間の速度”設定と動作中速度の読み込みを行います。
- 長軸（主軸）、従軸 1組の補間軸の中で最大移動量の軸を長軸（あるいは主軸）といいます。その他の軸を従軸といいます。
- 相対座標 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。CPDでは送り座標はすべて相対座標で指示します。
- 移動単位、速度単位 移動単位、すなわち座標値は[μm]、速度は[μm/sec]すなわち[PPS]で表現します。
- 動作速度 位置決め、補間、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、FHレジスタ（RFH）の値×速度倍率（RMGの値により決まる）が動作速度となります。
- ベース速度 ベース速度はFLレジスタ（RFL）と速度倍率レジスタ（RMG）で規定され、起動時は停止から直接ベース速度に達します。停止時はこの速度から即停止します。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します。（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ます。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ます。図2-1に速度パターン図を示します。

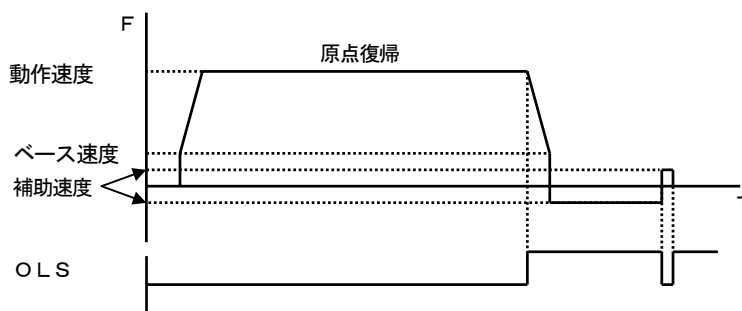


図2. 1-1 速度パターン図（直線加減速、原点復帰の例）

- 速度のオーバーライド 動作中に、動作速度FHレジスタ（RFH）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います。
- 軸センサ モータにより直動するテーブル機構の例を図2-2に示します。図中の「+ELS、-ELS、DLS、OLS」を軸センサとよびます。
「±ELS」 極限センサ（Stroke End—Limit Sensor）は直動軸の両端に配置されます。移動方向のELSがキャリッジを検出した時、ボードは指令パルス出力を直ちに停止、あるいは減速して止めます。ELSの検出状態ではELSの逆方向にのみ移動します。±は座標の方向と一致します。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出来ます。

- 「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します。いずれの機能に使用するかは原点復帰方法の選択によります。
- 「DLS」 DLSは原点復帰時の減速センサです。CPDでは減速センサの機能としてOLSが多く使われます。
- 「Z相」 原点としてエンコーダ（ENC）のZ相信号を原点とする場合に使われます。OLSにより減速を開始して、指定したn回目（nは1～15）のZ相で原点完了となります。
- ENC Z相の代わりに、1回転に1パルスが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相信号に使用出来ます。

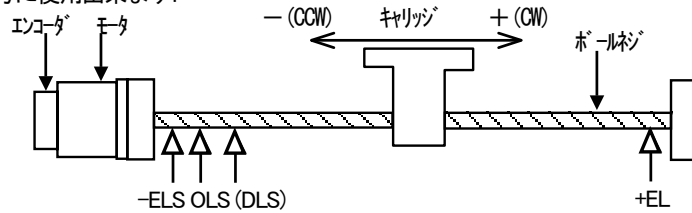


図2. 1-2 軸センサ（直動軸）

○外部同期入出力

- CPDボードではボード外部と同期をとるために「PCS（CPD5212Mは除く）、LTCH（CPD364のみ）、±DR（CPD364のみ）」入力信号が用意されています。
- また、ボード外部と同期をとるための「CMP出力（CPD5212M、CPD578、278、364のみ）、STA、STP」出力があります。
- 「PCS」 PCS連続送りで移動中に、PCS信号が入力された時点から予め設定してある移動量の位置決めが開始されます。PCS信号の代わりに「PCSコマンド」によっても可能です。
- 「LTCH」 LTCH入力によりカウント値をラッチすることができます。
- 「±DR」 動作モードをJOG送りに設定すると、+DR入力ON中で+方向動作、-DR入力ON中で-方向動作します。
- 「CMP 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します。また一定カウンタごとに出力する「同期出力」機能もあります。（コンパレータ4のみ）
- 「STA」 STAは入出力ピンです。（CPD578、278、364以外はオプション）STA入力でボード間の同時スタートができます。同時スタートコマンドまたはコンパレータ4条件一致で出力できます。また、CMP出力の代わりとしても使用できます。（但し、1ボード1出力）
- 「STP」 STPは入出力ピンです。（CPD578、278、364以外はオプション）STP入力でボード間の同期停止ができます。同時ストップコマンドまたはコンパレータ5条件一致で出力できます。また、CMP出力の代わりとしても使用できます。（但し、1ボード1出力、同期出力不可）

○指令パルスとモータ／モータドライバ

「指令パルス」

モータドライバへの出力をいいます。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ます。

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ます。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ます。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」は回転速度となります。

「モータ・ドライバ」

ボード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します。したがって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ります。

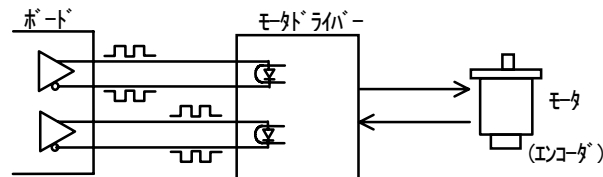


図2. 1-3 指令パルスとモータ・モータドライバ

○位置のオーバーライド

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いいます。

動作の様子は2通りあります。（1）当初の終点値より遠方に変更する場合。（2）当初の終点値より手前に変更した場合です。この場合は指示された時点で停止し、逆方向へ戻り、新たな終点に向います。

CPDボードでは位置決め動作のみ可能です。

○バックラッシュ動作、スリップ動作

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ます。スリップ動作は動作方法に無関係に補正量が入ります。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効です。

3. ボード構成とポートアドレス

3. 1 ブロック図

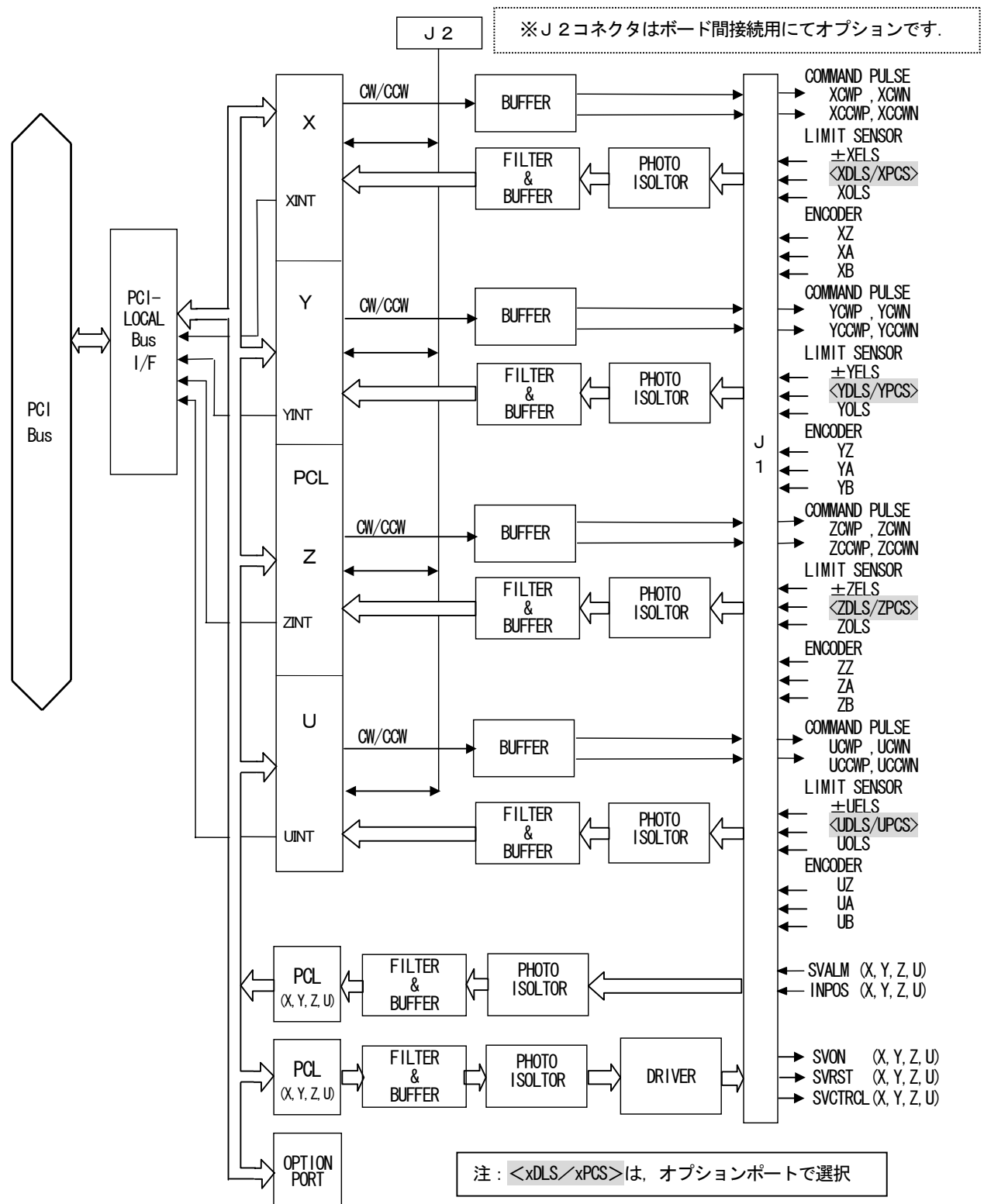


図3. 1-1 HPCI-CPD534ブロック図
【 他型式ボードは個別ユーザーズマニュアル参照 】

3. 2 ポートアドレス

3. 2. 1 ボードアドレス

ポートはすべて I/O マップです。

「添付ソフトウェア ライブラリ (レベル 1) 関数」においては以下の表は意識する必要はありません。

「デバイスドライバ関数」を使用する場合に必要となります。

区分		I/Oアドレス	読み込み (INP)		書き込み (OUT)	
軸数	名称	ポ-ードアドレス	呼称	内 容	呼称	内 容
2	X 軸 (第 1 軸)	+ 0	MSTS	メインステータス	CMD	コマンド
		+ 2	SSTS	サブステータス	OTP	不使用 (予約)
		+ 4	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 6	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	Y 軸 (第 2 軸)	+ 8	MSTS	メインステータス	CMD	コマンド
		+ a	SSTS	サブステータス	OTP	不使用 (予約)
		+ c	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ e	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	Z 軸 (第 3 軸)	+ 1 0	MSTS	メインステータス	CMD	コマンド
		+ 1 2	SSTS	サブステータス	OTP	不使用 (予約)
		+ 1 4	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 1 6	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	U 軸 (第 4 軸)	+ 1 8	MSTS	メインステータス	CMD	コマンド
		+ 1 a	SSTS	サブステータス	OTP	不使用 (予約)
		+ 1 c	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 1 e	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	V 軸 (第 5 軸)	+ 2 0	MSTS	メインステータス	CMD	コマンド
		+ 2 2	SSTS	サブステータス	OTP	不使用 (予約)
		+ 2 4	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 2 6	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	W 軸 (第 6 軸)	+ 2 8	MSTS	メインステータス	CMD	コマンド
		+ 2 a	SSTS	サブステータス	OTP	不使用 (予約)
		+ 2 c	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 2 e	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	A 軸 (第 7 軸)	+ 3 0	MSTS	メインステータス	CMD	コマンド
		+ 3 2	SSTS	サブステータス	OTP	不使用 (予約)
		+ 3 4	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 3 6	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
	B 軸 (第 8 軸)	+ 3 8	MSTS	メインステータス	CMD	コマンド
		+ 3 a	SSTS	サブステータス	OTP	不使用 (予約)
		+ 3 c	BUF0	入出力バッファ IN (15- 0)	BUF0	入出力バッファ OUT (15- 0)
		+ 3 e	BUF1	入出力バッファ IN (31-16)	BUF1	入出力バッファ OUT (31-16)
オプションポート (ボード全体に一組)	+ 8 0	ELPOL	各軸 E L S 極性・状態読込		ELPOL	各軸 E L S 極性設定
	+ 8 2	DLS/PCS	DLS/PCS入力選択状態読込		DLS/PCS	DLS/PCS入力選択設定
		※ボードの種類毎にオプションポートが決められています。 詳細は<個別ボード編>を参照して下さい。				
	+ 9 0	BINTM	ボード割込マスク読込		BINTM	ボード割込出力マスク設定
	+ 9 2	BINTS	ボード割込・状態読込		—	不使用 (予約)

表 3. 2-1 HPCI-CPD578, 534, 532 ポート表

3. 2. 2 オプションポート

オプションポートはボード全体に1組おかれています。次の機能があります。

- (1) 各軸E L S極性の設定と読込 (E L P O L) 電源投入時は' 0 ' : B接
ビットn=' 0 ' : B接 (カプラに電流がOFFでE L S検出), n=' 1 ' : A接 (カプラに電流が流れてE L S検出)

ボード軸数	ビット	7	6	5	4	3	2	1	0
2	機 能	0	0	0	0	0	0	YELS	XELS
4		0	0	0	0	UELS	ZELS	YELS	XELS
8		BELS	AELS	WELS	VELS	UELS	ZELS	YELS	XELS

- (2) 各軸D L S / P C S入力信号の選択と読込 (D L S / P C S) . . . 電源投入時は' 0 ' : D L S

ボード軸数	ビット	7	6	5	4	3	2	1	0
2	機 能	0	0	0	0	0	0	YPCS	XPCS
4		0	0	0	0	UPCS	ZPCS	YPCS	XPCS
8		BPCS	APCS	WPCS	VPCS	UPCS	ZPCS	YPCS	XPCS

x D L S選択時はx P C Sの入力極性をB接に設定します。(9. 1 R E N V 1 レジスタ参照)

x P C S選択時はx D L Sの入力極性をB接に設定します。(9. 1 R E N V 1 レジスタ参照)

- (3) ボード割込マスクの設定と読込 (B I N T M) 電源投入時はb 0=' 0 ' : 割込みマスク

ボードからP C I B u sへの割込みマスクを設定します。

b 0=' 0 ' : 割込みマスク, b 0=' 1 ' : 割込みアンマスク (割込み許可)

ビット	7	6	5	4	3	2	1	0
機 能	0	0	0	0	0	0	0	BINTM

- (4) ボード割込・状態読込 (B I N T S) 電源投入時はb 0=' 1 ' : 割込みなし

ボードからP C I B u sへの割込み状態を表します。

b 0=' 0 ' : 割込みあり, b 0=' 1 ' : 割込みなし

ビット	7	6	5	4	3	2	1	0
機 能	0	0	0	0	0	0	0	BINTS

■オプションポート機能欄の表記について

◆英数字記号のビット . . . 設定時は設定値(1 / 0)を書込み, 読込時は個々の状態(1 / 0)となります。

◆その他 ' 0 ' 表記のビット . . . 設定時はこの値を無視します。読込時にはこの値が読込まれます。

3. 2. 3 ボード入出力とデバイスドライバ

ボード上の各軸 (P C L)およびオプションポートへの入出力とドライバ関数との対応を次表に示します。

区分	軸の指定 ポートの指定	読込み (I N P)		書込み (O U T)	
		呼称	ドライバ関数	呼称	ドライバ関数
軸	引数 : 軸指定 (axis) 0 (X) ~ 7 (B)	MSTS	cp530_rMstsW ()	CMD	cp530_wCmdW ()
		SSTS	cp530_rSstsW ()	OTP	_____
		BUF0	cp530_rReg ()	BUF0	cp530_wReg ()
		BUF1	cp530_rBufDW ()	BUF1	cp530_wBufDW ()
オプション ポート	引数 : ポート(port) 0x80 0x82 (その他)	ELPOL	cp530_rPortB ()	ELPOL	cp530_wPortB ()
		DLS/PCS		DLS/PCS	
		(その他)		(その他)	
	0x90	BINTM	(デバイスドライバ処理)	BINTM	(デバイスドライバ処理)
	0x92	BINTS		_____	_____

表3. 2-2 H P C I - C P D 5 7 8, 5 3 4, 5 3 2のボード入出力とドライバ関数

- (注) 1. デバイスドライバはアプリケーションプログラムの動作するOSの種類毎に提供されます。
2. 対応OSの種類が異なっても、デバイスドライバの関数名は同一です。
3. ドライバ関数の詳細は「CPDボードシリーズ ユーザーズマニュアル<ソフトウェア編>」を参照して下さい。

3. 3 ポートとレジスタ配置

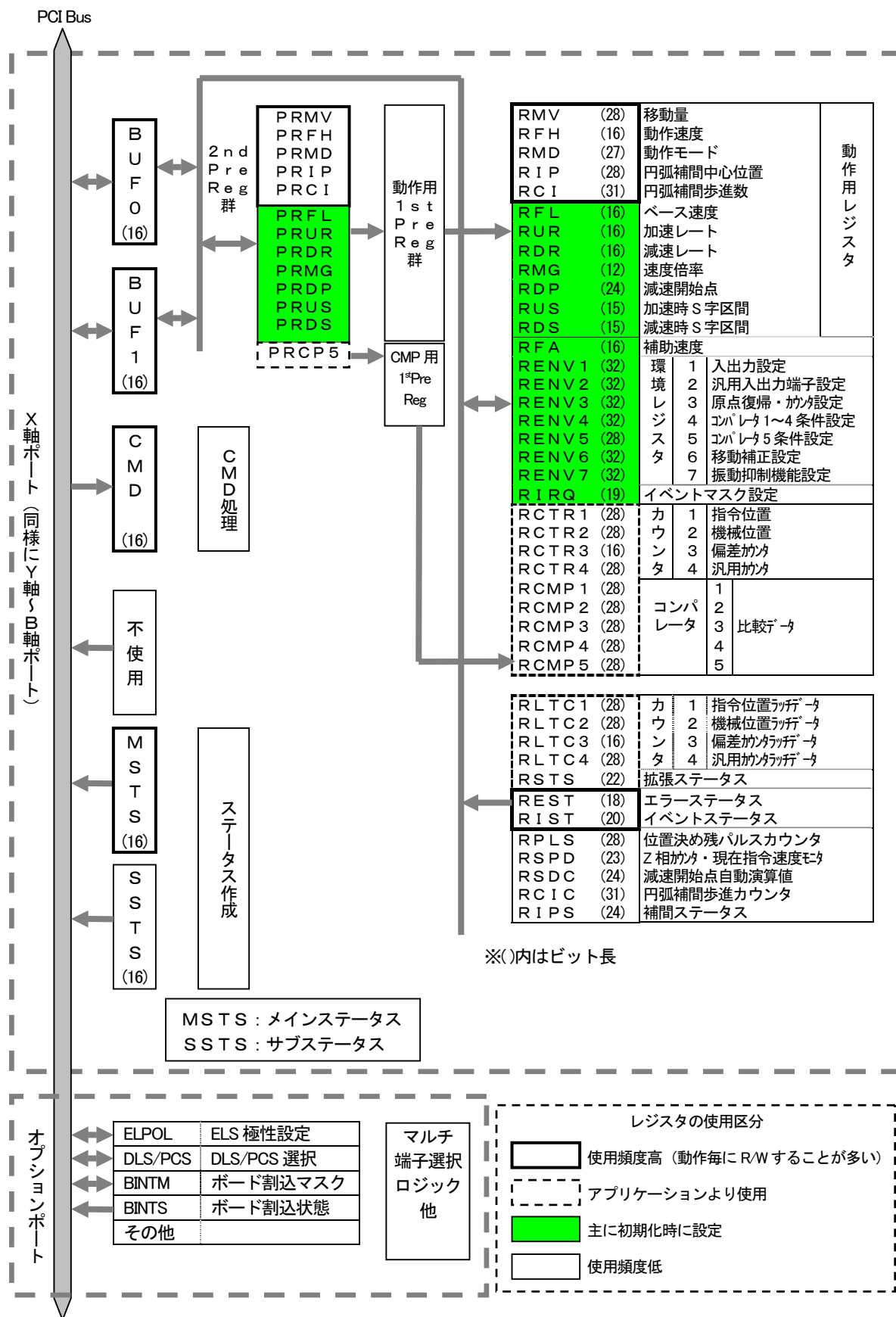


図 3. 3-1 HPCI-CPD578, 534, 532 のポートとレジスタ配置

4. 基本的な設定と運用

この節では、CPDボードの動作および機能をC++用ライブラリ関数に基づいて説明します。動作および機能は基本的運用（プログラム構成）の範囲で説明します。基本的運用とは次の範囲を想定しています。

- (1) 軸数 1～4 軸
- (2) 次動作連続実行（プリバッファの連続使用）はしない。
- (3) 位置のオーバライドは使用しない。
- (4) PCS機能は使用しない。
- (5) 条件付他軸起動は使用しない。
- (6) 2個以上のPCLにまたがった動作はしない。（8軸、12軸ボード）
- (7) ステータス取得はポーリング方式（ステータス割込みは「6.9節 参照」）

以上の条件下で次の動作モードを基本運用とします。

- ①連続送り ②位置決め ③直線補間 ④円弧補間 ⑤タイマー機能（ドゥエル機能）

なお、(1)～(6)以外の運用は「10. 各機能と応用」を参照して下さい。

4. 1 プログラムの手順

プログラムの基本手順は概略 図4. 1-1のフローに示す構成です。以下この手順によって添付ソフトの「ライブラリ関数（レベル1）」を使用して解説します。関数の詳細は「ユーザーズマニュアル<ソフトウェア編>」および「ユーザーズマニュアル<個別編>」のスタートアップガイドを参照して下さい。

なお、CPD534ボードを標準例として記述します。

概略手順は次の通りです。

(1) 初期設定

電源投入後 各軸に対して、ボードの初期設定を行います。初期設定の項目はインターフェースに関する条件(ELS,DLS/PCS, OLS, 指令パルス形式)、原点復帰方法、加速減速方法および加減速時間、ベース速度の設定、速度倍率設定などがあります。設定の場所は「環境設定レジスタ RENV1～RENV7」、速度、加減速関係レジスタ、モードレジスタの一部、オプションポートです。ライブラリ関数を利用する場合は「DevOpen関数」を発行した段階で各軸についてデフォルト値が設定されます。

初期化は各種初期化の関数を必要に応じて発行する必要があります。

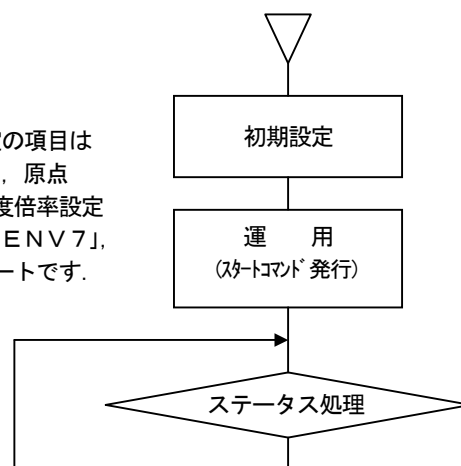


図4. 1-1 プログラム手順概要

(2) 運 用

初期設定が行われていればモータの動作は、各種の動作モードにしたがって、スタートコマンドを発行することにより、動作開始します。

動作モードは「モードレジスタ（PRMD）」に設定します。動作速度は実行前に「FHレジスタ（PRFH）」に設定します。移動量が必要な場合は 移動量を「移動量レジスタ（PRMV）」に相対座標値で書込みます。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）をコマンドバッファに書込むことによって動作開始します。

ライブラリ関数を利用する場合、

動作モードの設定は「WritOpelMode関数」、

動作速度の設定は「WritFHSpd関数」、

移動量の設定は、

位置決めの場合「WritPos関数」、直線補間の場合「WritLine関数」、円弧補間の場合「WritCircI関数」を使用します。

スタートコマンドの発行は、加速スタート「AccStart関数」、定速スタート「CnstStartFH関数」を使用します。

停止コマンドの発行は、減速停止「DecStop関数」、即停止「QuickStop関数」を使用します。

(3) 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メインステータス (MSTS)、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。

位置決め等の独立軸動作においては各軸のこれらのステータスによって動作終了を判断します。

補間動作の場合は補間代表軸 (X, Y, Z, …の順において一番 X に近い軸) の MSTS を取上げて判断します。

MSTS に動作終了またはエラーが検出されたとき各軸の RIST あるいは REST を読み処理終了します。

イベントステータスレジスタは、イベントマスクレジスタによってイベントステータス報告のマスクが出来ます。通常は初期化時にイベントマスク設定をします。

ライブラリ関数の場合は、MSTS 読み込み「ReadMainSts 関数」、RIST 読み込み「ReadEventSts 関数」、REST 読み込み「ReadErrorSts 関数」を使用します。イベントマスクレジスタのマスク状態は DevOpen 関数において「正常終了ビット」以外はすべてマスクされています。

4. 2 初期設定

●プログラムの始めの初期化は 次の関数を発行することで始まります。

(1) GetDevInfo 関数を発行します。【ISA Bus, PC/104 は不要】

現在パソコンに装着されている CPD ボードの数、および デバイス情報 (ボードの BIOS が割付ける資源情報) を取得します。この関数はライブラリの先頭で 1 回発行する必要があります。

(USB の場合はハブを通して接続されている CPD の台数および デバイス情報を取得します。)

(2) DevOpen 関数を発行します。【ISA Bus, PC/104 は先頭で発行】

ボード毎に DevOpen 関数を発行します。この場合 DevOpen 関数の引数に GetDevInfo で得たデバイス情報を与えて発行します。(注) その結果デバイスハンドル (ボード ID, ボード番号) が取得されます (hDevID)。以後 HPCI,

HPCPI, HUSB の各 Bus タイプのボードにおいてはこのデバイスハンドルによってボードを指定します。

【(注) ISA Bus, PC/104 はオープンするボードの軸数, ボードアドレスを与える】

さらに、この関数においては次の初期化を行っています。

①レジスタ類の初期設定 (各軸同一値で初期化を実施)

②オプションポートの初期化 (オプションポートはボードに 1 式あります。)

■オプションポートの初期値

ELS 極性選択ポート ELPOL : 0 x 0 0 (全軸 B 接)

DLS/PCS 切り替えポート DLS/PCS : 0 x f f (全軸 PCS 入力)

PCI, CPCI Bus への割込み禁止マスク BINTM : 0 x 0 0

【HPC はジャンパ設定, HUSB は割込み機能なし】

■レジスタの初期値

レジスタ	内容	初期値	備考
PRFL, RFL	ベース速度	2 0 0	2 0 0 p p s
PRFH, RFH	動作速度	2 0 0 0	2 0 0 0 p p s
PRUR, RUR	加速レート	1 3 6 4	直線加減速, 加減速時間 : 加速時 2 0 0 p p s から 2 0 0 0 p p s まで約 0. 5 秒 減速時 2 0 0 0 p p s から 2 0 0 p p s まで約 0. 5 秒
PRMG, RMG	速度倍率	2 9 9	1 倍
RFA	補助速度	2 0 0	2 0 0 p p s
PRMD, RMD	動作モード	0 x 0 8 0 0 8 0 0 0	
RENV1	環境設定 1	0 x 2 0 4 3 4 0 0 4	
RENV2	環境設定 2	0 x 0 0 2 0 f d 6 5	CPD 3 6 4
		0 x 0 0 2 0 f d 5 5	CPD 5 7 8, CPD 2 7 8, CPD 5 2 1 2 M
		0 x 0 0 2 0 f 5 5 5	上記以外
RENV3	環境設定 3	0 x 0 0 f 0 0 0 0 2	原点復帰モード 2 (OLS+ENC) 等
RIRQ	イベント設定	0 x 0 0 0 0 1	正常停止時
その他		0	PCL の初期値

表 4. 2-1 DEVOPEN 関数における初期化内容

●プログラムの終わり・・・DevClose 関数を発行して終了します。

すべてシステムが終了して電源 OFF する前に DevClose 関数を発行して、使用していた資源を解放します。

【CPD534 ライブラリ関数による記述例】

ステータス処理方法・・・ポーリング（ライブラリではポーリング方式を採っています。）

■GetDevInfo 関数

```
ret=hcp530_GetDevInfo(&count, &HpcDeviceInfo[0]);
```

■DevOpen 関数

```
ret=hcp530_DevOpen(hDevID[0], &HpcDeviceInfo[0]); // 1 番目のボードのデバイスハンドル取得と初期化
ret=hcp530_DevOpen(hDevID[1], &HpcDeviceInfo[1]); // 2 番目のボードのデバイスハンドル取得と初期化
以上の初期化の後、2 番目のボードの X 軸を +10000 (パルス)、Z 軸を -20000 (パルス) 定速動作で移動
させる場合、デバイスハンドル、ボードの軸指定の例は次の通りです。
```

■X 軸を位置決め動作モードにする：WritOpeMode 関数を X に発行する。

```
ret=hcp530_WritOpeMode(hDevID[1], // デバイスハンドルを与え、ボードを指定する。
0, // このボードの中の軸を指定。(0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸)
0x41); // 位置決め動作モードコマンド
```

■X 位置決め移動量をセットする：WritPos 関数を X に発行する。

```
ret=hcp530_WritPos(hDevID[1], // デバイスハンドルを与え、ボードを指定する。
0, // このボードの中の軸を指定。(0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸)
10000); // +方向 10000 パルス
```

■Z 軸を位置決め動作モードにする：WritOpeMode 関数を Z に発行する。

```
ret=hcp530_WritOpeMode(hDevID[1], // デバイスハンドルを与え、ボードを指定する。
2, // このボードの中の軸を指定。(0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸)
0x41); // 位置決め動作モードコマンド
```

■Z 位置決め移動量をセットする：WritPos 関数を Z に発行する。

```
ret=hcp530_WritPos(hDevID[1], // デバイスハンドルを与え、ボードを指定する。
2, // このボードの中の軸を指定。(0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸)
-20000); // -方向 20000 パルス
```

■F L 定速スタート：CnstStartFL 関数を発行する。（注意：スタートコマンドの軸指定は複数軸指定できる）

```
ret=hcp530_CnstStartFL(hDevID[1], // デバイスハンドルを与え、ボードを指定する。
5); // このボードの中の軸を指定する。(01:X, 02:Y, 04:Z, 08:U)
```



＜実際はこの続きに終了ステータス読取り関数を発行するプロセスが必要です。＞

速度は初期化において F L=200PPS および 倍率 RMG=299(1倍)に設定されています。したがって、X 軸と Z 軸は現在の点からそれぞれ 10000、-20000 の位置（相対座標）へ、定速 200PPS で移動します。

4. 2. 1 各種初期設定

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。DevOpen 関数においてデフォルト値による取りあえずの初期化を行っていることは前述しました。現実に沿った初期設定は、各種の初期設定系の関数が用意されています。

- | | | |
|----------------------------------|---|--|
| (1) オプションポートの設定 | ・・・① ELS 使用/不使用、極性(A 接/B 接)
② DLS/PCS 入力選択 | SetElis 関数
SetDisSel 関数 |
| (2) サーボインターフェース
(パルスモーターフェース) | ・・・① 指令パルス出力形式
② サボアーム使用/不使用
③ サボ偏差キャンセル機能使用/不使用
④ Inposition 機能使用/不使用 | SetCmdPulse 関数
SetSvAlm 関数
SetSvCtrl 関数
SetInpos 関数 |
| (3) 原点センサと原点復帰方法 | ・・・① 原点復帰方法設定
② エンコーダ Z 相の設定
③ QLS の極性設定 | SetOrMode 関数
SetEz 関数
SetOls 関数 |
| (4) 加減速度設定 | ・・・① S 字/直線加減速の選択
② 加減速レート | SetAccProfile 関数
SetAccRate 関数, CalAccRate 関数 |
| (5) 速度設定 | ・・・① 速度倍率
② ベース速度、補助速度
③ 動作速度 | SetMult 関数
SetFLSpd 関数, SetAuxSpd 関数
WritFHSpd 関数 |

4. 2. 2 オプションポートの設定 (ELS極性, DLS/PCS選択)

オプションポート次の2項目について、初期化時に設定します。(HPCI-CPD534, 532の場合)

No	設定事項	ポート信号名	選択事項	設定箇所
1	ELS 極性	±xELS	B 接(オフ電流 OFF で検出)／ A 接(オフ電流 ON で検出) ELS 無効 (使用しない) は A 接にしておく	オプションポート ELPOL b3～b0: U, . . . X
2	DLS/PCS 入力選択	xDLS/xPCS	DLS 減速セグ/PCS 信号として使用	オプションポート DLS/PCS

表 4. 2-2 HPCI-CPD534, 532のオプションポート設定要件

オプションポートはボードにより異なりますので、設定については「個別編」を参照してください。

【CPD534 ライブラリ関数による記述例】

(1) ELS 極性 設定の関数

SetEls 関数の機能の1つにELSの極性設定があります。ELSを使用しない場合は A 接とし、かつ ELS 端子を使用しない状態にします。(端子がオープンのままの状態)

この関数には ELS を検出したときの停止方法も指定します。(⇒9.1 RENV1)

《書式》

DWORD hcp530_SetEls(DWORD *hDevID*, WORD *axis*, WORD *pol*, WORD *stop*);

《引数》

- ◆ DWORD *hDevID* . . . 対象デバイスのデバイスハンドル。
- ◆ WORD *axis* . . . 軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *pol* . . . 極性 [0:B 接, 1:A 接]
- ◆ WORD *stop* . . . 停止方法 [0:即停止, 1:減速停止]

■初期値: B 接, 即停止

(2) DLS/PCS 入力選択の関数

SetDisSel 関数の機能の1つにコネクタ信号の xDLS/xPCS の選択があります。さらに、入力極性があります。DLS を選択した場合に、DLS 検出時の動作方法を指定します。

DLS にドグを用いず、DLS 信号がパルスの場合はこの信号をラッチする機能を選択できます。

《書式》

DWORD hcp530_SetDisSel(DWORD *hDevID*, WORD *axis*, WORD *para*, WORD *pol* WORD *motion*, WORD *latch*);

《引数》

- ◆ DWORD *hDevID* 対象デバイスのデバイスハンドル。
- ◆ WORD *axis* 軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *para* DLS/PCS 入力の切替 [0: DLS 入力, 1: PCS 入力, 2: 両方とも不使用]
- ◆ WORD *pol* 入力極性 [0:B 接, 1:A 接]
- ◆ WORD *motion* DLS 入力時の動作 [0:減速のみ, 1:減速停止]
- ◆ WORD *latch* DLS 入力のラッチ [0:ラッチしない, 1:ラッチする]

■初期値: DLS, PCS とも不使用, 減速のみ, ラッチしない。

4. 2. 3 サーボインターフェースの設定

サーボインターフェースの初期化に関係する項目は次表の通りです。

N o	設定事項	ボード信号名	選択事項	設定箇所
1	指令パルス出力形式	xCWP, xCWN xCCWP, xCCWN	独立パルス指令 (CWパルス, CCWパルス) または 共通パルス指令 (符号+パルス列)	環境レジスタ RENv1 b0~b2
2	サーボアラーム 使用する/しない	xSVALM	使用する (停止方法, 信号極性) ／しない (SVALM 信号線を接続せず かつ, 入力極性 A 接に設定)	環境レジスタ RENv1 b8~b9
3	サーボ偏差カウンタ 使用する/しない	xSVCTRCL	●原点完了時 SVCTRCL の使用 ●異常停止時 SVCTRCL の使用 SVCTRCL 信号の出力極性の設定	環境レジスタ RENv1 b10~b11, b15
4	インポジション 使用する/しない	xINPOS	([使用する/しない]は動作モードレジスタで行う) 使用する場合は入力信号極性も設定する	環境レジスタ RENv1 b22

表 4. 2-3 サーボインターフェースの設定要件

【 CPD534 ライブラリ関数による記述例 】

(1) 指令パルス出力形式の設定の関数定

初期に一度設定します。(指令パルス出力形式⇒9.1 RENv1)

《書式》

DWORD hcp530_SetCmdPulse (DWORD hDevID, WORD axis, WORD cmdp/s);

《引数》

- ◆ DWORD hDevID …対象デバイスのデバイスID
- ◆ WORD axis …軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸]
- ◆ WORD cmdp/s …指令パルスの出力形式 [0:個別パルス方式, 1:共通パルス方式]
- 初期値: 個別パルス方式

(2) サーボアラーム使用/不使用設定

モータドライバのアラーム信号をCPDボードに入力する場合に条件設定をします。アラーム信号の極性設定をします。(サーボアラームを使用しない場合はA接にしてボードの信号端子 xSVALM をオープンにしておきます。)

アラームを検出した場合の停止方法も設定します。(減速停止を選んだ場合は, サーボのアラーム信号は停止まで出力されている必要があります)

《書式》

DWORD hcp530_SetSvAlm (DWORD hDevID, WORD axis, WORD pol, WORD stop);

《引数》

- ◆ DWORD hDevID …対象デバイスのデバイスID
- ◆ WORD axis …軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸]
- ◆ WORD pol …極性 [0:B接, 1:A接]
- ◆ WORD stop …停止方法 [0:即停止, 1:減速停止]
- 初期値: B接, 即停止

サーボモータを使用する場合に、原点完了時 あるいはE L S、S V A L Mなど異常検出による即停止時にサーボドライバの偏差カウンタクリア信号をボード信号端子x S V C T R C Lに出力します。

(x SVCTRL 信号の幅 1 3 m s)

パルスモータの場合は不使用とします。

《書式》

DWORD hcp530 SetSvCtrCl (DWORD *hDevID*, WORD *axis*, WORD *enable*, WORD *pol*);

《引数》

- | | | |
|-----------|---------------|--|
| ◆ DWORD | <i>hDevID</i> | ・ 対象デバイスのデバイス ID |
| ◆ WORD | <i>axis</i> | ・ 軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸] |
| ◆ WORD | <i>enable</i> | ・ 使用/不使用
[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] |
| ◆ WORD | <i>pol</i> | ・ 極性 [0 : B 接 1 : A 接] (⇒9.1 RENV 1) |
| ■初期値: 不使用 | | |

(4) インポジション使用設定

サーボモータを使用する場合に、サーボドライバからのインポジション信号（ボード信号端子 x INPOS）の使用／不使用の選択、信号極性の設定を行います。パルスモータの場合は不使用とします。

《書式》

```
DWORD hcp530 SetInpos (DWORD hDev/D, WORD axis, WORD enable, WORD po/);
```

《引数》

- | | | | |
|--------|---------------|------------------------------------|--------------|
| ◆ WORD | <i>hDev/D</i> | ・ 対象デバイスのデバイス I D | |
| ◆ WORD | <i>axis</i> | ・ 軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸] | |
| ◆ WORD | <i>enable</i> | ・ 使用/不使用 [0 : 不使用, 1 : 使用] | |
| ◆ WORD | <i>pol</i> | ・ 極性 [0 : B 接, 1 : A 接] | (⇒9.1 RENV1) |
- 初期値：不使用

4. 2. 4 原点センサと原点復帰方法の設定

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。なお、原点復帰時の速度、加減速、については次節「4. 2. 5 速度および加減速関係の初期設定」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。)

- ①センサ原点(OLS), ②センサ原点とエンコーダZ相,
③ELS兼用原点, ④ELS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「原点復帰方法選択表：表4. 2-4，表4. 2-5，表4. 2-6」により選択して下さい。

■「原点復帰方法選択表:表4. 2-4, 表4. 2-5」において採用する原点復帰 No. を環境設定レジスタ RENW3 の ORGmode (b3~b0) にセットしておきます。これは「SetOrgMode 関数」によって行います。

■原点復帰動作の所定のタイミングでCTR1～CTR4を選択してクリアすることができます。

REN V3 bit20 = '1' ・ ・ CTR1 (指令位置カク), bit21 = '1' ・ ・ CTR2 (機械位置カク)
bit22 = '1' ・ ・ CTR3 (脱調カク), bit23 = '1' ・ ・ CTR4 (汎用カク)

注意：1. ライブラリ関数ではすべての CTR が自動的にクリアされる設定になっています。

2. 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4. 2-5 (CTR2参照方式) 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CTR2をクリアし、減速終了時のCTR2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CTR2は“0”になるとは限りません。

注意：ライブラリ関数を使用した場合のCTR2入力

原点復帰方法No. 9を選択・・・エンコーダZ相は不使用・・・「指令パルス入力」

No. 10~12を選択・・エンコーダZ相を使用・・・「ENC入力」

■表4. 2-6 (センサ原点DLS方式)は原点復帰方法No. 0 にDLS信号を追加することにより原点精度が改善された方法となります。(⇒4. 3. 1 RMD および 9. 1 RENV1)

No	原点復帰方法名	定速原点復帰 (FH定速スタート)	高速原点復帰 (FH加速スタート)
0	OLS原点 RENV3 ORGmode=0	<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	<p>OLS検出で即停止。FA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	<p>OLS検出後n個目のZ相検出で完了</p>	<p>※減速中に最後のZ相があると完了</p>
3	OLS+ENC原点 RENV3 ORGmode=3	<p>OLS検出後n個目のZ相で減速し、ベース速度で完了</p>	<p>※減速中に最後のZ相があると完了</p>
4	OLS反転+ENC原点 RENV3 ORGmode=4	<p>OLS検出で即停止。FA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FA速度で反転。n個目のZ相検出で完了</p>
5	OLS反転+ENC原点 RENV3 ORGmode=5	<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	<p>ELS検出で即停止。FA速度で反転しELS拔出し完了</p>	<p>※ ELS幅>減速距離</p>
7	ELS反転定速+ENC原点 RENV3 ORGmode=7	<p>ELS検出で即停止。FA速度で反転しn個目のZ相検出で完了</p>	<p>※ ELS幅>減速距離</p>
8	ELS反転定速+ENC原点 RENV3 ORGmode=8	<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幅>減速距離</p>

注意：1. No.0,3,5,8の高速減速原点方式では原点精度がサーボゲインの影響をうけます。

2. ●マークはCTR1～CTR4のクリアが設定してある場合のクリアされるタイミングを表します。

表4. 2-4 原点復帰方法選択表

No	原点復帰方法名	高速原点復帰 (FH加速スタート)	
9	OLS反転+CTR2原点 RENV3 ORGmode=9	<p>OLS検出で減速, 同時にCTR2クリア 停止後に反転し, 停止時のCTR2の値 (減速距離) の符号を反転した移動量位置決めをし完了. ※ CTR2カウント入力は指令パルス(推奨) (RENV3 b9-b8=01')</p>	
10	ENC反転+CTR2原点 RENV3 ORGmode=10	<p>OLS検出後のn回目Z相で減速, 同時にCTR2クリア 停止後に反転し, 停止時のCTR2の値 (減速距離) の符号を反転した移動量位置決めをし完了. ※ CTR2カウント入力はエンコーダ(推奨) (RENV3 b9-b8=00') 但し, サーボモータ使用時は, サーボモータ使用時は, 減速停止時の偏差分 CTR2=0 まで動作しません.</p>	
11	ENC反転+CTR2原点 RENV3 ORGmode=11	<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CTR2クリア. 停止後に反転し, 停止時のCTR2の値 (減速距離) の 符号を反転した移動量位置決めをし完了. ※ CTR2カウント入力はエンコーダ(推奨) (RENV3 b9-b8=00') 但し, サーボモータ使用時は, サーボモータ使用時は, 減速停止時の偏差分 CTR2=0 まで動作しません.</p>	
12	ELS反転+CTR2原点 RENV3 ORGmode=12	<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CTR2クリア. 停止後に反転し, 停止時のCTR2の値 (減速距離) の 符号を反転した移動量位置決めをし完了. ※ CTR2カウント入力はエンコーダ(推奨) (RENV3 b9-b8=00') 但し, サーボモータ使用時は, サーボモータ使用時は, 減速停止時の偏差分 CTR2=0 まで動作しません.</p>	

注意: 1. ●マークはCTR1~CTR4のクリアが設定してある場合のクリアされるタイミングを表します.
2. 最小限CTR2はクリアを行う設定が必要です. (RENV3 bit21 = '1')

表4. 2-5 原点復帰方法選択表 (CTR2参照方式)

No	原点復帰方法名	高速原点復帰 (FH加速スタート)	
13	DLS減速OLS原点 RENV3 ORGmode=0	<p>OLS原点復帰(N=0)のOLS減速停止より精度改善 ※DLS減速後 定速で移動継続 (RENV4 b4=0)</p>	

注意: 1. この方式はNo. 0の高速原点復帰においてDLSを使用する設定とします.
(⇒4. 3. 1 RMD b8='1' および 9. 1 RENV1 b4='0')
2. ●マークはCTR1~CTR4のクリアが設定してある場合のクリアされるタイミングを表します.

表4. 2-6 原点復帰方法選択表 (センサ原点DLS方式)

【 CPD534 ライブラリ関数による記述例 】

(1) 原点復帰モード設定

《書式》

```
DWORD hcp530_SetOrgMode(DWORD hDevID, WORD axis, WORD mode);
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイス I D.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *mode* ・・原点復帰モード番号 (ORGmode No.)

例： X 軸を原点復帰方法 No. 1 を選択しておく。

```
ret=hcp530_SetOrgMode(hDevID,
                        0,      //X 軸
                        1);    //原点復帰方法 No. 1 指定
```

4. 2. 5 速度および加減速関係の初期設定

速度および加速度に関する初期化の項目を表 4. 2-7 に示します。

加減速パターン(直線/S 字), 加減速度は通常のアプリケーションでは初期にこれらの関係のレジスタに設定しておきます。

同じくベース速度を規定する F L 速度, 原点復帰時の原点突入速度 F A (補助速度) 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。移動指令時に動作速度レジスタ (PRFH) に設定してスタートコマンドを発行します。

項	レジスタ内容	アドレス	ビット長	設定範囲	初期設定	記 事	
1	ベース速度	PRFL	RFL	16	1～65,535	○	
2	動作速度	PRFH	RFH	16	1～65,535	毎回運用	
3	加速レート	PRUR	RUR	16	1～65,535	○	
4	減速レート	PRDR	RDR	16	0～65,535	○	
5	速度倍率	PRMG	RMG	12	2～4,095	○	
6	減速開始点	PRDP	RDP	24	0～16,777,215	○	
7	加速時S字範囲	PRUS	RUS	15	0～32,767	○	
8	減速時S字範囲	PRDS	RDS	15	0～32,767	○	
9	補助速度	なし	RFA	16	1～65,535	○	

表 4. 2-7 速度関係のレジスタ設定

■ 速度パターンの各部の呼称

表 4. 2-7 の各レジスタは下図 4. 2-1 の各部分に対応します。

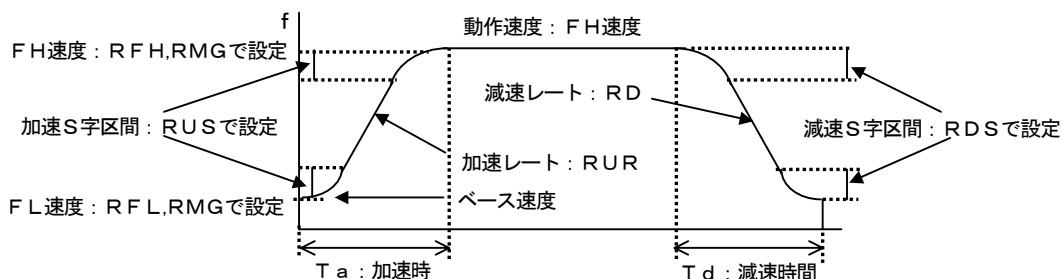


図 4. 2-1 速度パターンとレジスタの対応

減速開始点 : RMD b13 で減速開始点自動を選択している場合で, RDP=0 の時減速点は加減速が等勾配として自動計算される。

■速度倍率：速度レジスタ（RFH, RFL, RFA）は16bitです。

倍率 1倍では1～65,535 PPS範囲の速度設定が出来ます。

より広範囲の速度に対しては倍率レジスタPRMGに倍率を設定しておきます。

倍率	RMGの値	F (PPS)	指令周波数Fの取りうる範囲		
0.1	2999	$0.1 \times RFx$	0.1 PPS	～	6,553.5 KPPS
0.2	1499	$0.2 \times RFx$	0.2 PPS	～	13,107 KPPS
0.5	599	$0.5 \times RFx$	0.5 PPS	～	32,767.5 KPPS
1	299	$1 \times RFx$	1 PPS	～	65,535 KPPS
2	149	$2 \times RFx$	2 PPS	～	131,070 KPPS
5	59	$5 \times RFx$	5 PPS	～	327,675 KPPS
10	29	$10 \times RFx$	10 PPS	～	655,350 KPPS
20	14	$20 \times RFx$	20 PPS	～	1,310,700 KPPS
50	5	$50 \times RFx$	50 PPS	～	3,276,750 KPPS
100	2	$100 \times RFx$	100 PPS	～	6,553,500 KPPS

表 4. 2-8 速度倍率設定例

速度倍率は初期化の段階で決定しておきます。その軸の使用最高速度を考慮した倍率を倍率レジスタ（RMG）に設定します。

このRMGの値はベース速度、補助速度、動作速度の全てに共通します。

補間動作の場合は補間代表軸（X軸に最も近い軸）のRMGが採用されます。

ライブラリ関数は「SetMult 関数」で倍率設定をします。

■動作速度：運用速度であり補間、位置決め、連続送りの場合の送り速度をいいます。

PRFHにスタートコマンドを発行毎に指示します。ただし、速度が前回と同じであれば必要ありません。原点復帰時も原点復帰速度はPRFHに指示します。

■ベース速度：動作速度がベース速度以上になった時加速が開始され、以下になった時減速が行われます。

動作速度が始めからベース速度以下であれば定速送りとなります。

■補助速度：一部の原点復帰時の定速送り部分（原点突入速度）、バックラッシュ動作あるいはスリップ動作の速度に使われます。

動作速度、ベース速度および補助速度 は次の式で与えられます。

$$\text{速度} F = \frac{300 \times PRFx}{PRMG + 1} [\text{PPS}]$$

ただしPRFxはPRFH(動作速度)、PRFL(ベース速度)またはRFA(補助速度)

初期化の段階でベース速度、（原点突入速度として採用する場合など）補助速度を与えておきます。

ライブラリ関数によればベース速度設定は「SetFLSpd 関数」補助速度設定は「SetAuxSpd 関数」を用います。

（「DevOpen 関数」は倍率 1、ベース速度、補助速度は200 PPSに初期化しています。）

なお、動作速度は「WriteSpd 関数」で行います。

■加減速パターン：

動作モードレジスタ（PRMD）bit10で直線加減速（b10='0'）かS字加減速（b10='1'）を初期時に選択します。

S字加減速の場合S字範囲設定レジスタ（RUS, RDS）を'0'以外の値を設定すると部分S字パターン加減速になります。

ライブラリ関数ではS字／直線加減速の選択を「SetAccProfile 関数」で行います。（「DevOpen 関数」は直線加減速を初期値としています。）

■加減速時間：加速時間と減速時間は通常同一として使用します。

Ta=Td（加速・減速時間同一）の場合の直線加減速時間は次式で与えられます。

$$Ta = Td = \frac{(PRFH - PRFL) \times (PRUR + 1) \times 4}{19,660,800} [\text{sec}] \quad \text{この時PRDR} = 0 \text{であること}$$

（S字加減速および非対称加減速 Ta<Tdの場合 ⇒ 10.1 節）

ライブラリ関数ではS字／直線加減速の選択は「SetAccProfile 関数」、加減速レートの設定は「SetAccRate 関数」、加減速レートレジスタ値の計算は「CalcAccRate 関数」で行います。

【 CPD534 ライブラリ関数による記述例 】

(1) 速度倍率設定

《書式》

```
DWORD hcp530_SetMult(DWORD hDevID, WORD axis, DWORD rmg);
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイス I D.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *rmg* ・・倍率設定値

(2) ベース速度, 補助速度設定

《書式》

```
DWORD hcp530_SetFLSpd ( DWORD hDevID, WORD axis, DWORD speed);   //ベース速度設定
```

```
DWORD hcp530_SetAuxSpd( DWORD hDevID, WORD axis, DWORD speed);   //補助速度設定
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイス I D.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *speed* ・・ F L, F A 速度

(3) 加減速パタンの設定

《書式》

```
DWORD hcp530_SetAccProfile( DWORD hDevID, WORD axis, WORD pr );
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイス I D.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *pr* ・・加減速形式 [0 : 直線, 1 : S 字]

(4) 加減速時間, F H, F L 等より加減速レートを計算します.

この関数で得た加減速レートを (5) 項の各加速, 減速レートに代入します.

《書式》

```
DWORD hcp530_CalAccRate( DWORD* rate, WORD time, DWORD fh, DWORD fl, WORD pro, WORD s );
```

《引数》

- ◆ DWORD* *rate* ・・加減速レート [1~65535]
- ◆ WORD *time* ・・加減速時間 (msec)
- ◆ DWORD *fh* ・・ F H レジスタ値 [1~65535]
- ◆ DWORD *fl* ・・ F L レジスタ値 [1~65535]
- ◆ WORD *pro* ・・加減速形式 [0 : 直線, 1 : S 字]
- ◆ WORD *s* ・・常に '0' にして下さい.

(5) 加減速レート設定

《書式》

```
DWORD hcp530_SetAccRate( DWORD hDevID, WORD axis, DWORD rate );
```

```
DWORD hcp530_SetDecRate( DWORD hDevID, WORD axis, DWORD rate );
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイス I D.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *rate* ・・加速レート, 減速レート

4. 3 運 用

この節は連続送り、原点復帰、位置決め、補間動作、等々の動作実行について説明します。
一般的には次の手順でプログラムを進めます。

- (1) 動作モード設定
- (2) 動作速度設定
- (3) 移動量設定
- (4) その他（サーボ ON/OFF など）
- (5) スタートコマンド発行
- (6) スタートコマンド発行後のステータス処理（4. 4 節を参照して下さい。）

すなわち、スタートコマンドを発行する前に「動作モード」、「動作速度」、「移動量」をセットします。
スタートコマンド発行後はステータス処理にはいります。

4. 3. 1 動作モードレジスタ RMD (PRMD) と動作モード設定

(1) 動作モードレジスタの構成

RMD (PRMD) の構成を図 4. 3-1 に示します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	PCS 有効	減速開始点	0	CTR1DIS	acc 方法	INPSE	DLSE	←---- MOD ----→							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	1	FH補正	同他軸停止	同時ストップ	停止指定軸設定	条件付スタート	SEQ No					

図 4. 3-1 RMDレジスタのビット構成

bit	名称	説 明		ラティ初期値
b0-b7	MOD	4. 3. 1. 2 動作モード参照		
b8	DLSE	'1' = DLS 有効 DLS 検出により減速または減速停止（オプションで DLS/PCS で選択時）		0
b9	INPSE	'1' = INPOS 有効		0
b10	acc 方法	加減速方法 '0' = 直線加減速 '1' = S 字加減速		0
b11	CTR1DIS	'1' = 指令位置カウンタ（CTR1） 出力禁止		0
b12	不使用	'0' 固定（完了タイミングを周期完了とする：推奨）		0
b13	減速開始点	'0' = 減速開始点自動計算設定 '1' = 減速開始点手動計算設定		0
b14	PCS 有効	'1' = PCS 入力有効（PCS 信号が入力されると位置決め開始。 （オプションで DLS/PCS で選択時）		0
b15	補間合成速度	'0' = 一定制御 OFF '1' = 一定制御 ON（推奨）		1
b16	SEQ No	動作ブロックを管理する時は、2ビットのシーケンス No. を設定する。MSTS を読み出すことにより 現在実行中の SEQ No が確認できる。SEQ No は動作に影響しない。		00
b17				
b18	条件付	スタート発行時の 軸スタート条件	b19, b18 00 : 即スタート 10 : 他軸起動	00
b19	スタート		01 : STA 入力によるスタート 11 : 指定軸停止によりスタート	
b20	停止指定 軸設定 (※1)	b19, b18=11 の場合 停止確認する軸を 設定する。	b23 - 0001 : X(V, X1, X2, X3) 軸 停止でスタート 0101 : X(V, X1, X2, X3) と Z(A, Z1, Z2, Z3) が共に停止でスタート	0000
b21			0010 : Y(W, Y1, Y2, Y3) 軸 停止でスタート 1111 : 全軸 (X-U, V-B, X1-U1, X2-U2, X3-U3) が停止でスタート	
b22			0100 : Z(A, Z1, Z2, Z3) 軸 停止でスタート 0000 : 条件無し	
b23			1000 : U(B, U1, U2, U3) 軸 停止でスタート	
b24	同時ストップ	'1' = STP(同時停止信号) 入力により減速停止または停止する。		0
b25	同他軸停止	異常停止時に STP(同時停止信号) を出力する。		0
b26	FH補正	'0' = ON（推奨）, '1' = OFF		0
b27	不使用	'1' 固定（円弧自動終点引込み：推奨）		1
b28-31	未定義	常に '0' を設定してください		0

表 4. 3-1 RMDレジスタの内容

※1. 2 軸ボードは Z, U 軸はありません。1 個の PCL 内の軸の組み合わせで有効です。

■解 説

- ① b 0 ~ b 7 (MOD) : スタートコマンド発行以前にMODをセットする. (詳細は 4. 3. 1. 2 参照)
- ② b 8 (DLS E) : 減速センサーDLS有効にすると, DLS検出により減速しベース速度で走行または減速停止する. 減速/減速停止はRENV 1 b 4 により選択する.
DLS信号またはPCS信号の選択はオプションポートで初期化時に選択.
(通常 電源投入直後の初期設定)
- ③ b 9 (INPSE) : インポジジョン(INPOS)有効. サーボモータ使用時に, インポジジョン機能を使用する場合 b 9 を有効にする. 有効にすると, サーボドライバからのINPOS信号(コネクタ端子の x INPOS)を検出して, MST Sは動作終了状態になる.
INPOS無効の場合は, パルス指令が終了すると同時にMST Sは動作終了状態になる. INPOS信号の極性はRENV 1 b 2 2 で設定.
(INPSE は通常 電源投入直後の初期に設定する.)
- ④ b 1 0 (acc 方法) : 加減速方法の選択. (通常 電源投入直後の初期に設定する)
- ⑤ b 1 1 (CTR 1 D I S) : 指令位置カウンタ (CTR 1) のカウント有効/無効. (CTR 2 ~ CTR 4 のカウント有効/無効はRENV 3 b 2 9 ~ b 3 1 で行う)
- ⑥ b 1 2 (不使用) : '0' 固定. (動作完了タイミングは周期完了とする: 推奨)
- ⑦ b 1 3 (減速開始点) : 減速開始点の設定方法. (通常 電源投入直後の初期に設定する)
- ⑧ b 1 4 (PCS有効) : オプションポートにてPCSを選択してある時,
PCS有効にするとPCS信号入力によって位置決め開始する. (⇒ 1 0. 4 節)
- ⑨ b 1 5 (補間合成速度) : '0' : 合成速度一定制御OFF
'1' : 合成速度一定制御ON
- ⑩ b 1 9, 1 8 (条件付スタート) :

スタートコマンド (accSTART, cnstSTART, 同時スタート) 発行時に条件つきスタートとなる.

b19	b18	内 容
0	0	accSTART, cnstSTART は通常スタート. 同時スタートコマンドは無効.
0	1	同時スタートコマンド (06h) のみ有効. 複数のPCLを同時にスタートさせる. (⇒ 1 0. 8 節)
1	0	accSTART, cnstSTART コマンドによってRENV5の条件による他軸スタート. (⇒ 1 0. 6. 4 節)
1	1	accSTART, cnstSTART コマンドにより次項 (11) の指定軸条件によりスタートする. (⇒ 1 0. 5 節)

- ⑪ b 2 3 ~ 2 0 (停止によりスタート) : b 1 9, 1 8 = 1 1 の場合の指定軸条件.
- ⑫ b 2 4 (同時ストップ) : '1' = 同時ストップコマンド (07h) を受け入れる.
減速停止/即停止はRENV 1 b 1 9 による.
- ⑬ b 2 5 (同他軸停止) : b 2 4 により同時ストップコマンドを受け入れた場合, このビットが '1' であれば 2 個のPCLにまたがる他の軸も停止することが出来る.
ただし, 他の軸も b 2 4 = '1' であること.
- ⑭ b 2 6 (FH補正) : b 2 6 によりFH補正 (三角駆動回避) 機能を設定. このビットが '0' でFH補正機能ON (推奨). このビットが '1' であればFH補正機能OFF.
FH補正機能ON時は位置決めまたは補間動作時に加減速動作させた時に, 加減速するために十分な移動量がないと自動的に最高速度を低下させて三角駆動を回避します.
- ⑮ b 2 7 (不使用) : '1' 固定. (円弧補間自動終点引込み: 推奨)

(2) 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7モードあります。指定軸のモードレジスタPRMD bit7～0にMODコードをセットします。ライブラリ関数では「WriteOpenMode 関数」の引数に軸指定をし、動作モードにMOD(16進コード)を書きます。

No	動作モード分類	MOD	動作モード	記 事
1	連続動作モード	00h	＋方向連続動作	(＋手動送り)
		08h	－方向連続動作	(－手動送り)
2	位置決め動作モード	41h	位置決め動作	
		42h	PCS位置決め動作(ライブラリ関数固有)	10. 4節参照
		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	補間動作モード	60h	直線補間連続送り	(直線補間手動送り)
		61h	直線補間	
		62h	2個のPCL間の直線補間連続送り	10. 8節参照
		63h	2個のPCL間の直線補間	
3	補間動作モード	64h	CW方向円弧補間	
		65h	CCW方向円弧補間	
4	原点復帰動作モード	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	
5	ELS, SLS動作モード	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	
6	Z相移動モード	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	
7	パルス動作モード	01h	パルス入力による連続動作	(手動パルス送り)
		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	
8	JOG送り動作モード	02h	±DR入力による連続動作	(JOG送り)
		56h	±DR入力による位置決め動作	

表4. 3-2 動作モード一覧

①連続動作モード

独立軸動作であり、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, ー方向:08hを目的軸のPRMDビットに設定 2. 速度設定 動作速度を目的軸の速度レジスタPRFHに設定 3. スタートコマンド “高速53h/定速51h”を目的軸に発行 4. 停止コマンド 停止は高速・定速によらず減速停止コマンド“4ah”を発行
	ELS	1. +(－)方向に進行時+(－)ELSを検出して停止、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RENV3'b3'で設定

表4. 3-3 連続動作モード

②位置決め動作モード

独立軸動作であり、タイマー動作を除き、ELS・DLS・SLS動作は①～④共通です。

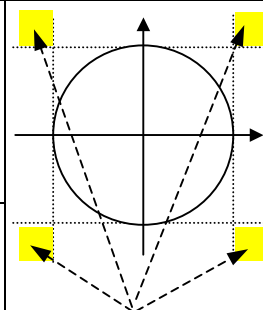
位置決め MOD=41h	動作	1. モード設定 41hを目的軸のPRMDビットに設定 2. 移動量パルス数目的軸のPRMVに相対移動量を設定、動作方向は符号で決まる。 [-134, 217, 728 ≤ 移動量 ≤ +134, 217, 727 (相対位置送り)] 移動量0でスタートさせるとパルスは出力されず即完了 3. 速度設定 動作速度を目的軸の速度レジスタPRFHに設定 4. スタートコマンド “高速53h/定速51h”を目的軸に発行
	ELS	1. +(−)方向に進行時+(−)ELSを検出して停止、 検出後逆方向へはスタートコマンドで動作 2. ELS検出時の停止方法はRENV3“b3”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、指令位置カウン(CTR1)が0になるまで移動	
	動作	1. モード設定 44hを目的軸のPRMDビットに設定 2. 速度設定 動作速度を目的軸の速度レジスタPRFHに設定 3. スタートコマンド “高速53h/定速51h”を目的軸に発行
	[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、機械位置カウン(CTR2)が0になるまで移動	
	動作	1. モード設定 45hを目的軸のPRMDビットに設定 2. 速度設定 動作速度を目的軸の速度レジスタPRFHに設定 3. スタートコマンド “高速53h/定速51h”を目的軸に発行
	[ELS] 前に同じ	
1パルス送り +方向送り MOD=46h −方向送り MOD=4eh	PRMVに±1パルスをセットせずに、スタートコマンドを発行すると1パルス移動して完了	
	動作	1. モード設定 46hを目的軸のPRMDビットに設定 (+1パルス) 4ehを目的軸のPRMDビットに設定 (−1パルス) 2. 速度設定 動作速度を目的軸の速度レジスタPRFHに設定 3. スタートコマンド “高速53h/定速51h”を目的軸に発行
	[ELS] 前に同じ	
タイマー動作 MOD=47h	目標位置レジスタPRMVは単にタイマカウンタとして動作し、PRMVのパルス数と速度レジスタPRFH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：PRMV=100[パルス]、PRFH=1000[PPS]の場合100ms) PRMVの設定値範囲・正数：1～134, 217, 727	
	動作	1. モード設定 47hを目的軸のPRMDビットに設定 2. タイマ速度設定 目的軸PRFHにタイマ速度(PPS)を設定 (1000 PPSを推奨：1ms) 3. 移動量パルス数目的軸のPRMVに停止時間を設定 4. スタートコマンド “定速51h”を目的軸に発行
	注意	1. このPRMDレジスタのb12=0にして使用 (動作完了タイミング MENDTM) 2. ELS他・・・指令パルスは出ない (CTR1非動作)→ELS, DLS, SLSは無視 3. INPOSも無関係。タイマ動作中のSV ALM信号、<EMG入力>は有効

表4. 3-4 位置決め動作モード

③補間動作モード

補間は連続直線補間、直線補間、円弧補間の3種のモードがあります。各軸にモード、移動量（相対値）を指定の上、速度は補間代表軸にのみ設定し、スタートコマンドに（補間）軸選択を立てて発行します。
このモードで補間代表軸の呼称、速度の設定、ステータスの関係を共通事項として説明します。

[補間代表軸]・・・軸がX, Y, Z, Uの順にある時、一組の補間軸において最もXに近い軸を「補間代表軸」と呼ぶ。 [速度指示]・・・補間モードにおいて速度は各軸の合成速度を指示する。 この合成速度は1式のみ補間代表軸のPRFHに与えればよい。 ただし、4軸直線補間では合成速度は $\sqrt{4}$ 制御とはならず $\sqrt{3}$ 制御になる。	
ステータス	<p>(1) 補間モードにおけるMSTSの完了ステータス：SINT(bit5)は各補間軸同じ内容を示す。</p> <p>(2) MSTSの監視は常に特定の1軸(例えば補間代表軸)を監視していれば良い。正しく完了した時はMSTSの完了割込みSINTビットが検出される。この時補間軸の全てのイベント割込みステータス(RIST)を読む必要がある。このことによって各軸のMSTSがクリアされ次の動作の完了検出ができる</p> <p>(3) エラー発生して完了した時はMSTSのSINTはセットされず代わりにSERR(bit4)が各軸に共通にセットされる。ただし、SSTS, RESTなどの要因ビットは発生している軸にのみセットされる。エラー発生軸のRESTを読み込むとRESTはクリアされ、補間軸のMSTSのSERRビットもクリアされ、このことによって各軸のMSTSがクリアされ、次の動作の完了検出ができる。</p>
ELS	ELS, SLSの動作は位置決めと同じ。ただし、補間軸の1つの軸がリミットを検出すると全軸が停止する
スタートコマンド	スタートコマンドの軸SELビットには補間軸をセットして発行する。発行軸はどの補間軸に対しても良いが、補間代表軸に対して書込む方が分かりやすい。スタートコマンドは高速53h、定速51h。
連続直線補間 MOD=60h	<p>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を発行するまで補間動作をします</p> <p>動作</p> <ol style="list-style-type: none"> 1. モード設定 60hを補間各軸のPRMDビットに設定。 2. 移動量パルス数各補間軸のPRMVに相対値で設定。 (連続送りなので設定値は移動方向を与えるだけ)。 3. 速度設定 動作速度を補間代表軸の速度レジスタPRFHに設定 4. スタートコマンド 補間軸SEL+“高速53h/定速51h”を補間代表軸に発行
直線補間 MOD=61h	<p>このモードにしてスタートコマンドを発行すると各軸のPRMVに設定した座標終点まで直線補間で送られる。速度は2軸では$\sqrt{2}$制御、3軸では$\sqrt{3}$制御され合成速度で移動する。 ただし、4軸は$\sqrt{4}$制御とはならず$\sqrt{3}$制御で行われる。</p> <p>動作</p> <ol style="list-style-type: none"> 1. モード設定 61hを補間各軸のPRMDビットに設定。 2. 移動量パルス数各補間軸のPRMVに相対値で設定。 3. 速度設定 動作速度を補間代表軸の速度レジスタPRFHに設定 4. スタートコマンド 補間軸SEL+“高速53h/定速51h”を補間代表軸に発行
円弧補間 CW円弧 MOD=64h CCW円弧 MOD=65h	<p>任意の2軸の円弧補間を行う。</p> <p>(1) 現在点を始点とし、この点から見た終点座標値をPRMVに、始点からみた円の中心座標をPRIPレジスタに設定する。</p> <p>(2) 2軸の終点値が共に(0,0)の場合は真円になる。</p> <p>(3) 終点座標が円周上にない場合、右図の4隅の■部分を除き終点引込みをする</p> <p>(4) 速度は周速一定制御される</p> <p>動作</p> <ol style="list-style-type: none"> 1. モード設定・・・補間各軸に CW円弧64hをPRMDビットに設定 CCW円弧65hをPRMDビットに設定 2. 移動量パルス数各補間軸のPRMVに相対値で設定。 中心点をPRIPに相対値で設定 3. 速度設定 補間代表軸のPRFHに動作速度を設定 4. スタートコマンド 補間軸SEL+“高速53h/定速51h”を補間代表軸に発行 <p>注意</p> <p>次の機能は円弧補間においては出来ません・</p> <ol style="list-style-type: none"> 1. 円弧補間で加減速(直線加速、円弧定速、直線減速ブロックを利用して実現します) 2. バックラッシュ補正動作、スリップ補正。 3. 位置のオーバーライド



このエリア(4ヶ所)に終点が設定された時は停止しません。その他では終点引込み停止

表4. 3-5 補間動作モード

原点復帰モードは原点復帰動作、原点拔出し動作、原点サーチ動作の動作をモードを指定しスタートコマンド発行して実行します。

表4. 3-6 原点復帰動作モード

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。

表4. 3-7 ELS, SLS 動作モード

RENV2のZ相CT設定数分の移動を行います。

表4. 3-8 Z相移動モード

⑦パルス動作モード（HPCI-CPD508は除く）

パルスモード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。

パルス信号形式選択	RENV2：パルス倍率(b25, 24)設定で“通信(x1, x2, x4)またはアップダウン・パルス”を指定 PDIR(b26)でPA/PB入力カウンタ方向を設定			
スタートコマンド	定速コマンドを発行する。パルスの周波数は定速51hスタートコマンドで指定する			
パルス入力 最高周波数	RFHで設定、RFHで制限される。RFH設定速度とパルス周波数(FP)の関係は次の通り。			
	1 通信(x1)	2 通信(x2)	4 通信(x4)	アップダウン・パルス
	FP<RFH	FP<(RFH/2)	FP<(RFH/4)	FP<RFH
ELS	ELS検出でパルス出力停止するが、逆方向は動作可能			
モード解除	即停止コマンド(49h)を発行			
パルス連続送り MOD=01h	動作	パルスの送りによる指令パルス出力(通常のハンドル送り)		
		1. モード設定	01hを目的軸のPRMDビットに設定	
		2. 速度設定	パルス入力最高周波数相当値を目的軸の速度1/2がPRFHに設定	
3. スタートコマンド	“定速51h”を目的軸に発行			
パルス位置決め送り MOD=51h	動作	パルス入力信号(方向は無関係)に同期して位置決め動作、目標位置に達すると動作終了		
		1. モード設定	51hを目的軸のPRMDビットに設定	
		2. 移動量/パルス数	目的軸のPRMVに相対移動量を設定、動作方向は符号で決まる。	
3. 速度設定	パルス入力最高周波数相当値を目的軸の速度1/2がPRFHに設定			
4. スタートコマンド	“定速51h”を目的軸に発行			
パルス同期 指令位置0点復帰 MOD=54h	動作	パルス入力信号(方向は無関係)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を目的軸のPRMDビットに設定	
		2. 速度設定	パルス入力最高周波数相当値を目的軸の速度1/2がPRFHに設定	
3. スタートコマンド	“定速51h”を目的軸に発行			
パルス同期 機械位置0点復帰 MOD=55h	動作	パルス入力信号(方向は無関係)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を目的軸のPRMDビットに設定	
		2. 速度設定	パルス入力最高周波数相当値を目的軸の速度1/2がPRFHに設定	
3. スタートコマンド	“定速51h”を目的軸に発行			

表 4. 3-9 パルス動作モード

⑧JOG送り動作モード（HP104D-CPD364のみ）

JOG送り動作モードにはJOG送り、±DR位置決め送り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±DR信号が有効となり、DR信号がONの間、指令出力にパルスが出力されます。

スタートコマンド	定速コマンドを発行する。		
ELS	ELS検出でパルス出力停止するが、逆方向は動作可能		
SVALM, EMG	SVALM, EMG入力でエラー停止		
モード解除	即停止コマンド（49h）を発行		
JOG送り （±DR連続送り） MOD=02h	動作	±DR入力による指令パルス出力	
		1. モード設定	02hを目的軸のPRMDビットに設定
		2. 速度設定	DR入力中の速度を目的軸の速度の1/2がPRFHに設定
		3. スタートコマンド	“定速51h”を目的軸に発行
		4. 動作方向は+DR入力ONで+方向、-DR入力ONで一方向に動作	
±DR位置決め送り MOD=56h	動作	±DR入力による位置決め指令パルス出力、目標位置に達すると動作終了	
		1. モード設定	56hを目的軸のPRMDビットに設定
		2. 移動量/パルス数	目的軸のPRMVに相対移動量を設定、動作方向は符号で決まる。
		3. 速度設定	DR入力中の速度を目的軸の速度の1/2がPRFHに設定
		4. スタートコマンド	“定速51h”を目的軸に発行

表 4. 3-10 JOG送り動作モード

【 CPD534 ライブラリ関数による記述例 】

■動作モード書き込み関数

《書式》

```
DWORD hcp530_WriteMode( DWORD hDevID, WORD axis, WORD mode );
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイスハンドル.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *mode* ・・動作モード

■補間における動作モード設定は補間軸全てをモード設定します.

例: X, Y 2 軸円弧補間,

```
ret = hcp530_WriteMode( hDevID,          // デバイスハンドル
                        0,                // X 軸
                        0x64 );          // CW円弧

ret = hcp530_WriteMode( hDevID,          // デバイスハンドル
                        1,                // Y 軸
                        0x64 );          // CW円弧
```

4. 3. 2 動作速度設定

動作速度の設定は「4. 2. 5 速度および加減速の初期設定」の「■動作速度」の項で説明されています.
すなわち,

$$\text{速度} F = \frac{300 \times \text{PRFH}}{\text{PRMG} + 1} [\text{PPS}]$$

例: 倍率 10 倍 (PRMG=29) で 200Kpps (F) の速度を希望する場合は PRFH レジスタに 20000 をセットします.

$$\begin{aligned} \text{速度} F &= \frac{300 \times \text{PRFH}}{\text{PRMG} + 1} [\text{PPS}] \text{ より } \text{PRFH} = \frac{F \times (\text{PRMG} + 1)}{300} \\ \therefore \text{PRFH} &= \frac{200,000 \times (29 + 1)}{300} = 20,000 \end{aligned}$$

■動作速度設定の注意:

- (1) 独立軸動作は各軸に速度を設定します.
- (2) 補間軸は「補間代表軸」に速度を設定します.
- (3) タイマー動作(ドウエル)は時間待ちの単位時間を使用している倍率によって決めておきます.
通常単位時間を 1ms (1Kpps) に選びます.
例: 倍率 10 で 1Kpps の場合 $\text{PRFH} = 1000 \times 30 / 300 = 100$ PRFH に 100 をセットします.
この速度設定の上でドウエル時間=移動量(パルス) × 1ms となります.
- (4) 速度レジスタ PRFH に一度設定した値は変更しない限り変わりません.

【 CPD534 ライブラリ関数による記述例 】

(1) 動作速度の書き込み

《書式》

```
DWORD hcp530_WriteFHSpd( DWORD hDevID, WORD axis, DWORD speed );
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイスハンドル.
- ◆ WORD *axis* ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ WORD *speed* ・・FH速度 (実速度は倍率を掛けた値です.)

例:

```
ret = hcp530_WriteFHSpd( hDevID,          // デバイスハンドル
                        1,                // Y 軸を指定
                        5000 );          // 速度倍率が 10 倍ならば速度 = 50Kpps
```


4. 3. 3 移動量設定

移動量は位置決め、連続直線補間、直線補間、円弧補間、パルス位置送り、タイマー動作の各場合に設定する必要があります。前節の「動作モード設定」に記載の通りです。

■移動量はPRMVに相対値（符号付）で設定します。（-134, 217, 728～+134, 217, 727）

■動作中にRMVレジスタを直接変更することにより位置オーバーライドが行えます。

ただし、位置決め動作の場合に限ります。（⇒「10. 3 位置のオーバーライド」参照）

■円弧補間はPRIPレジスタに円の中心座標値を始点から見た相対位置で書きます。

（⇒「4. 3. 1 （2）動作モード ③補間動作モード」参照）

【CPD534 ライブラリ関数による記述例】

（1）位置決めの場合の移動量書込み

《書式》

```
DWORD hcp530_WritPos( DWORD hDevID, WORD axis, long dstnc );
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイスハンドル。
- ◆ WORD *axis* ・・軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸]
- ◆ long *dstnc* ・・移動量

（2）直線補間の場合移動量書込み

《書式》

```
DWORD hcp530_WritLine( DWORD hDevID, WORD axis, long dstnc );
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイスハンドル。
- ◆ WORD *axis* ・・軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸]
- ◆ long *dstnc* ・・移動量

例：X=10000, Y=-15000 を移動量としてセット。

```
ret = hcp530_WritLine( hDevID,          //デバイスハンドル
                      0,                //X軸を指定
                      10000 );          //+方向10000パルス
ret = hcp530_WritLine( hDevID,          //デバイスハンドル
                      1,                //Y軸を指定
                      -15000 );         //-方向15000パルス
```

（3）円弧補間の場合移動量書込み

《書式》

```
DWORD hcp530_WritCircI( DWORD hDevID, WORD axis, long dstnc1, long dstnc2, long center1, long center2 );
```

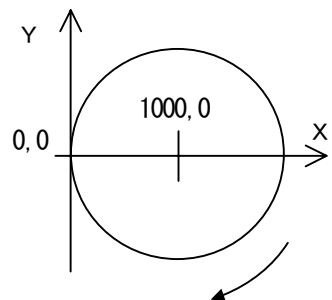
《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイスハンドル。
- ◆ WORD *axis* ・・補間軸組合わせ [0:X-Y軸, 1:X-Z軸, 2:X-U軸, 3:Y-Z軸, 4:Y-U軸, 5:Z-U軸]
- ◆ long *dstnc1* ・・終点位置 1
- ◆ long *dstnc2* ・・終点位置 2
- ◆ long *center1* ・・中心位置 1
- ◆ long *center2* ・・中心位置 2

終点位置及び中心位置 1, 2 の順番はX軸に近い軸の順番になります。

例：半径 1000 パルスの真円（モード設定でXY CWと設定してある場合）

```
ret = hcp530_WritCircI( hDevID,          //デバイスハンドル
                        0,                //X Y軸で円弧補間
                        0,                //終点位置 1 (X軸)
                        0,                //終点位置 2 (Y軸)
                        1000,            //中心位置 1 (X軸)
                        0 );              //中心位置 2 (Y軸)
```



（4）タイマーモード（ドゥエル）の待ち時間

例：速度設定において1Kpps（1ms）に設定されている場合、X軸を1000msタイマーとします。

```
ret = hcp530_WritPos( hDevID,          //デバイスハンドル
                     0,                //X軸ドゥエル
                     1000 );           //1000ms
```

4. 3. 4 動作コマンド

動作コマンドを表 4. 3-11 に示します。動作コマンドの他に制御コマンド等があります。
これらは「7. コマンド」を参照して下さい。

コマンド名		CMD	機 能		略 称	備 考
動作コマンド	スタートコマンド	加速スタート	5 3 h	ベース速度(F L)から動作速度(F H)へ加速し F H定速で動作後減速停止。 減速開始点手動設定時減速開始点=0 で加速ブロック開始となる。	accSTART	通常は使用しない
		F H定速スタート	5 1 h	動作速度(F H)定速スタートで F H 継続移動 定速ブロック開始	cnstSTART	
		F H定速スタート後 減速停止	5 2 h	動作速度(F H)定速スタート後減速停止 減速開始点手動設定で減速ブロック開始	decSTART	
		F L定速スタート	5 0 h	ベース速度(F L)定速スタートで F L 定速移動	baseSTART	
	停止	即停止	4 9 h	即停止	STOP	
		減速停止	4 a h	減速停止	DEC stop	
		非常停止	0 5 h	EMG stop	EMG stop	
	残量スタート	残量 F L 定速スタート	5 4 h	残量をベース速度(F L)で定速スタート	rbaseSTART	7. 1 コマンド 参照
		残量 F H 定速スタート	5 5 h	残量を動作速度(F H)で定速スタート	rcnstSTART	
		残量加速スタート	5 7 h	残量を動作速度(F H)へ加速スタート	raccSTART	
		残量 F H 定速スタート後 減速停止	5 6 h	残量を動作速度(F H)で定速スタート後減速停止	rdecSTART	
	速度変更	F L 定速瞬時速度変更	4 0 h	F H 動作中瞬時にベース速度(F L)に速度変更	FCHGL	
		F H 定速瞬時速度変更	4 1 h	F L 動作中瞬時に動作速度(F H)に速度変更	FCHGH	
		F L 速度まで減速	4 2 h	F H 動作中減速してベース速度(F L)に速度変更	FDECL	
		F H 速度まで加速	4 3 h	F L 動作中加速して動作速度(F H)に速度変更	FACCH	
	同時スタート		0 6 h	2 個以上の P C L に	同時スタート	QMSTA
	同時ストップ (※)		0 7 h	またがった	同時ストップ	QMSTP

※. 次動作使用時の同時ストップの際はプリレジキャンセルコマンドを発行してください。

表 4. 3-11 動作コマンド

(1) スタートコマンドおよび停止コマンド

動作開始のスタートコマンド発行は

- (1) 通常は、あらかじめ設定してある速度レジスタ (P R F H) に対して加速スタートまたは定速スタートコマンドで指定します。(下表 4. 3-12)
- (2) スタートさせる軸を選択します。(補間の場合は補間軸全てを指定します。)

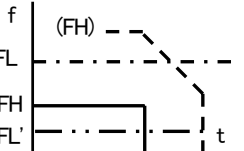
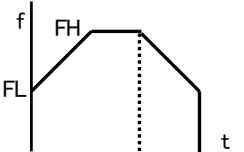
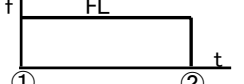
スタートコマンド	速度動作	動作モード：連続モード	動作モード：位置決め又は補間
動作速度 (F H) で 定速スタート (cnst START)	スタートコマンド = F H 定速 (51h) 	① P R F H に設定してある動作速度 F H に即立上がる ② 減速停止 (4 a h), 即停止 (4 9 h) コマンドにより即停止 ※減速停止を想定する場合, $F L' \leq F H$ とする.	① P R F H に設定してある動作速度 F H に即立上がる. ② 移動目標値に達した時, 即停止. ●動作速度 F H が F L 以下の場合の停止 即停止, 減速停止コマンドの停止は②. ●動作速度 F H が F L 以上の場合の停止 減速停止コマンドはベースまで減速して停止 即停止コマンドは常に即停止.
動作速度 (F H) へ 加速スタート (acc START)	スタートコマンド = F H 加速 (53h) 	① P R F L に設定してあるベース速度 F L に即立上がり, 動作速度 P R F H に設定してある速度 F H に加速 注意: 加速スタートコマンドでは必ず $F H \geq F L$ のこと ② 減速停止コマンドはベースまで減速して停止, 即停止コマンドは常に即停止	① P R F L に設定してあるベース速度 F L に即立上がり, 動作速度 P R F H に設定してある速度 F H に加速. ② 残移動量が減速開始点に達して減速し, ベース速度 F L で停止 注意: 円弧補間は定速スタートコマンドを使用 ●停止コマンド 減速停止コマンドはベースまで減速して停止 即停止コマンドは即停止
ベース速度 (F L) で 定速スタート (base START)	スタートコマンド = F L 定速 (50h) 	① P R F L に設定してあるベース速度 F L に即立上がる ② 減速停止 (4 a h), 即停止 (4 9 h) コマンドにより即停止	① P R F L に設定してあるベース速度 F L に即立上がる. ② 移動目標値に達した時, 即停止 ●停止コマンド 減速停止, 即停止コマンドにより即停止

表 4. 3-12 スタートコマンドの速度動作

ライブラリ関数を使用し、独立軸、補間軸にかかわらず動作させる軸を選択します。

【CPD534 ライブラリ関数による記述例】

(1) スタートコマンド発行

《書式》

```
DWORD hcp530_GnstStartFH( DWORD hDevID, WORD axis );      動作速度 (FH) で定速スタート
DWORD hcp530_AccStart ( DWORD hDevID, WORD axis );      動作速度 (FH) へ加速スタート
```

《引数》

- ◆ DWORD hDevID …対象デバイスのデバイスハンドル。
- ◆ WORD axis …軸指定 [0x01 : X軸, 0x02 : Y軸, 0x04 : Z軸, 0x08 : U軸]
軸指定はORしたデータも可能です。

例 : X, Y, Z を一斉に加速スタート

```
ret = hcp530_AccStart( hDevID,      //デバイスハンドル
                      0x07 );      //X、Y、Z軸
```

(2) (位置送りの途中) 停止

《書式》

```
DWORD hcp530_QuickStop( DWORD hDevID, WORD axis );      即停止
DWORD hcp530_DecStop ( DWORD hDevID, WORD axis );      減速停止
```

《引数》

- ◆ DWORD hDevID …対象デバイスのデバイス I D.
- ◆ WORD axis …軸指定 [0x01 : X軸, 0x02 : Y軸, 0x04 : Z軸, 0x08 : U軸]

■動作中にR F Hレジスタに速度を直接書込むことにより移動中の速度オーバーライドが出来ます。
(⇒ 1 0. 2 速度のオーバーライド)

(2) 非常停止コマンド

非常停止コマンドの動作は即停止と同じです。ただし、ステータスの結果が異なります。非常停止はメインステータス (M S T S) にエラー発生 bit4 (SERR) がセットされます。即停止の場合 M S T Sは終了 bit3 (SEND), イベント報告発生 bit5 (SINT) がセットされます。

【CPD534 ライブラリ関数による記述例】

《書式》

```
DWORD hcp530_EmgStop( DWORD hDevID, WORD axis );
```

《引数》

- ◆ DWORD hDevID …対象デバイスのデバイスハンドル。
- ◆ WORD axis …軸指定 [0x01:X 軸,0x02:Y 軸,0x04:Z 軸,0x08:U 軸]
軸指定はORしたデータも可能です。

例 :

```
ret = hcp530_EmgStop( hDevID,      //デバイスハンドル
                      0x07 );      //X、Y、Z 軸非常停止
```

4. 3. 5 その他 (SVON, SVRST)

SVONおよびSVRSTポートは汎用出力制御コマンドを使用します。
これらのポートは基本的に汎用出力ポートなので他の目的に使用できます。

ポートコマンド	CMD	機能名称	略 称	備 考
SVON	1 8 h	SVON ON	SVON	コネクタ端子 x SVONがO [V]
ON/OFF	1 0 h	SVON OFF	SVOFF	同 x SVONがHigh (EXTPOW) レベル
SVRST	1 9 h	SVRST ON	SVRSTON	コネクタ端子 x SVRSTがO [V]
ON/OFF	1 1 h	SVRST OFF	SVRSTOFF	同 x SVRSTがHigh (EXTPOW) レベル

表 4. 3-13 SVON, SVRSTポートの操作

【CPD534 ライブラリ関数による記述例】

ライブラリ関数には パルスモータに対する励磁 ON/OFF (MF ON) の関数があります。サーボ ON と信号が逆のため用意してあります。

《書式》

```
DWORD hcp530_SvOn      ( DWORD hDevID, WORD axis );    //サーボ ON      ( xSVON 端子が ON )
DWORD hcp530_SvOff     ( DWORD hDevID, WORD axis );    //サーボ OFF     ( xSVON 端子が OFF)
DWORD hcp530_SvResetOn ( DWORD hDevID, WORD axis );    //サーボリセット ON ( xSVRST 端子が ON )
DWORD hcp530_SvResetOff ( DWORD hDevID, WORD axis );   //サーボリセット OFF ( xSVRST 端子が OFF)
DWORD hcp530_PMOOn     ( DWORD hDevID, WORD axis );    //パルスモータ ON  ( xSVON 端子が OFF)
DWORD hcp530_PMOff     ( DWORD hDevID, WORD axis );    //パルスモータ OFF ( xSVON 端子が ON )
```

《引数》

- ◆ DWORD *hDevID* ・・対象デバイスのデバイスハンドル。
- ◆ WORD *axis* ・・軸指定 [0x01:X 軸,0x02:Y 軸,0x04:Z 軸,0x08:U 軸]

例：

```
ret = hcp530_SvOn( hDevID,      //デバイスハンドル
                  0x07 );      //X,Y,Z 軸
```

4. 4 スタートコマンド発行後のステータス処理

4. 4. 1 ステータス構成

基本的運用において取上げるステータスは次の3種類です。これら各ステータスについて常にステータス処理の対象となるビットについて運用面の解説をします。

(1) メインステータス (MST S)

ステータスの監視は常に対象となる軸のMST Sに対して行います。

スタートコマンドを発行後はMST Sを常にポーリングします。b 5およびb 4を監視し、これら全てが‘0’の場合は移動中です。

移動が終了すると正常終了ではb 5が‘1’，異常終了ではb 4が‘1’となります。このとき‘1’を示している対応のステータスの処理を読みます。

b 4= ‘1’ の場合は、エラーステータスの処理をします。b 5= ‘1’ の場合はイベントステータスの処理をします。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPDF	SPRF	0	SCMP5	SCMP4	SCMP3	SCMP2	SCMP1	SSC1	SSC0	SINT	SERR	SEND	0	SRUN	SSCM

図 4. 4-1 メインステータス MST Sのビット構成

b3 (SEND) : ‘1’ = 移動終了状態

b4 (SERR) : ‘1’ = エラー生じた (REST をリードすると ‘1’ は自動的に ‘0’ となる)

b5 (SINT) : ‘1’ = 移動が終了した (RIST をリードすると ‘1’ は自動的に ‘0’ となる)

※ b 5が移動終了とは、イベントマスクレジスタ (RIRQ) のb 0のみアンマスクしてある場合です。

注意：b 3 (SEND) は状態を示している bit です。電源投入直後は ‘0’ です。

一度移動実行後終了状態は ‘1’，移動中は ‘0’ を示します。

通常停止中 (= ‘0’) が動作中かの確認したいときに使用します。

(2) イベントステータス (RIST)

RISTはMST S b 5 (SINT) = ‘1’ を検出したとき必ず読まなければなりません。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISOL	ISLT	ISCL	ISC5	ISC4	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISND	ISNM	ISN	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	ISSA	ISMD	ISPD	ISSD

図 4. 4-2 イベントステータス RISTのビット構成

イベントステータスは初期化において「割り込みマスクレジスタ R I R Q」の動作完了 b 0 (I R E N) のみアンマスク (イベント報告イネーブル) で、他はすべてマスクして下さい (推奨) 。 (⇒ 6. 5 節)

- b 0 (I S E N 動作完了) : ‘ 1 ’ = 動作完了した。
 (R I S T をリードすると ‘ 1 ’ は自動的に ‘ 0 ’ となる。 M S T S の b 5 (S I N T) も ‘ 0 ’ となる)
 b 2 (I S N M 動作プリレジ書込み可) : プリレジスタ使用の場合用います。
 (⇒ 1 0. 7 次動作連続実行)
 b 8, 9 (I S C 1, 2 ± S L S 検出した) : ソフトリミット使用の場合用います。
 (⇒ 1 0. 6. 3 ソフトリミット機能)

(3) エラーステータス (R E S T)

R E S T は M S T S b 4 (S E R R) = ‘ 1 ’ を検出したとき必ず読まなければなりません。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESAO	ESPO	ESIP	ESDT	0	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC5	ESC4	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	ESPE	ESEE

図 4. 4-3 エラーステータス R E S T のビット構成

エラーステータスはマスクすることは出来ません。通常生起するエラーは ± E L S, S V A L M です。
 R E S T をリードすると検出されたビットは自動的に ‘ 0 ’ となり、M S T S の b 4 S E R R も ‘ 0 ’ となります。

- b 5 (E S P L 検出) : ‘ 1 ’ = + E L S を検出停止した
 b 6 (E S M L 検出) : ‘ 1 ’ = − E L S を検出停止した
 b 7 (E S A L 検出) : ‘ 1 ’ = サーボアラームを検出停止した

4. 4. 2 補間の場合のステータス

独立軸はそれぞれの軸の M S T S を見ますが、補間の場合は補間代表軸 (X, Y, Z, U の並びで X 軸に近い軸をいいます。) のステータスを取り上げればよく、残りの軸の M S T S ステータス b 3 ~ b 5 ビットは皆同じ状態を示します。

ただし、M S T S の S E R R はそれぞれの軸に生じたエラー内容を示します。M S T S に S E R R を検出した場合は補間軸の全てを讀出して処理する必要があります。讀出さないと、M S T S の S E R R が ‘ 0 ’ になりません。

4. 4. 3 ライブラリ関数による終了ステータス処理例

【 C P D 5 3 4 ライブラリ関数による記述例 】

(1) 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 読込み関数

《書式》

```
DWORD hcp530_ReadMainSts ( DWORD hDevID, WORD axis, WORD* status );
DWORD hcp530_ReadErrorSts ( DWORD hDevID, WORD axis, DWORD* status );
DWORD hcp530_ReadEventSts ( DWORD hDevID, WORD axis, DWORD* status );
```

《引数》

- ◆ DWORD hDevID … 対象デバイスのデバイスハンドル。
- ◆ WORD axis … 軸指定 [0 : X 軸, 1 : Y 軸, 2 : Z 軸, 3 : U 軸]
- ◆ WORD* (DWORD*) status … 読込んだデータが格納されるエリアのアドレス

例 :

```
ret = hcp530_ReadMainSts ( hDevID, //デバイスハンドル
                          1, //Y 軸を指定
                          &msts ); //MSTS の格納先のアドレス
```

(2) 1 軸 のみのステータス処理の例

- ①SLS は不使用. イベントマスクは RIRQ は b0 以外 初期化時にマスクしてある.
- ②エラーステータスは SLS 不使用として b0, b1 は取り上げない. ELS と SVALM を対象にメッセージ表示する.

```
//-----  
// X 軸ステータス読み処理(メッセージ表示)  
//-----  
void TmrXMSts(HWND hwnd)  
{  
    WORD msts; //メインステータス  
    DWORD ests; //エラーステータス  
    DWORD ists; //イベントステータス  
    DWORD dwRet; //戻り値  
  
    ■メインステータスの読み  
    dwRet = hcp530_ReadMainSts( hDevID, 0, &msts );  
    if( msts & 0x0030 ){ ← SERR, SINT  
        ■スタートコマンド終了  
        if(msts & 0x0010){ ← SERR  
            ■エラーステータス発生  
            hcp530_ReadErrorSts( hDevID, 0, &ests ); //エラーステータス読み  
            if( ests & 0x20 ) MessageBox( hwnd, " +ELS による停止", MB_OK ); //メッセージ表示  
            if( ests & 0x40 ) MessageBox( hwnd, " -ELS による停止", MB_OK ); //メッセージ表示  
            if( ests & 0x80 ) MessageBox( hwnd, " SVALM による停止", MB_OK ); //メッセージ表示  
        } else { ← SINT  
            ■イベントステータス発生  
            hcp530_ReadEventSts( hDevID, 0, &ists ); //イベントステータス読み  
            if( ists & 0x1 ) MessageBox( hwnd, " 正常停止", MB_OK ); //メッセージ表示  
        }  
    }  
}
```

4. 5 各動作のサンプルコーディング

代表的な動作のライブラリ関数によるサンプルを掲げます. (全て 1 軸動作です.)

このサンプルでは, コーディングの内容をより明確とするため, ライブラリ関数の戻り値は全て正常として扱っています.

4. 5. 1 原点復帰

原点復帰方法は「X 軸定速 O L S 原点復帰 (ORGmode1)」とします. X 軸は停止状態とします.

```
DWORD ret; //関数の戻り値  
WORD msts; //メインステータス  
DWORD errs, evsts; //エラーステータス, イベントステータス  
WORD run = 0; // 1 : 動作開始, 0 : 動作終了  
  
ret = hcp530_WriteMode( hDevID, 0, 0x18 ); //X 軸 : 動作モード=一方向原点復帰動作  
ret = hcp530_SetOrgMode ( hDevID, 0, 1 ); // : 原点復帰モード=O L S 原点復帰 (ORGmode1)  
ret = hcp530_WriteFHSpd ( hDevID, 0, 10000 ); // : 動作速度の設定 (速度倍率 1 倍: 10Kpps)  
ret = hcp530_CnstStartFH( hDevID, 0x01 ); // : FH 定速スタート  
run = 1; // : 動作開始  
while( run ) {  
    ret = hcp530_ReadMainSts( hDevID, 0, &msts ); //X 軸 : メインステータスの読み  
    if( msts & 0x10 ) { //X 軸 : エラーステータス報告あり  
        ret = hcp530_ReadErrorSts( hDevID, 0, &errs ); // : エラーステータスの読み  
        run = 0; // : 動作終了 (異常)  
    }  
    if( msts & 0x20 ) { //X 軸 : イベントステータス報告あり  
        ret = hcp530_ReadEventSts( hDevID, 0, &evsts ); // : イベントステータスの読み  
        run = 0; // : 動作終了 (正常)  
    }  
}
```

4. 5. 2 位置決め

Y軸に対して10Kppsの速度で20000パルスの位置決めを行います。

この動作では加減速形式、加速・減速レート、減速開始点等は既に設定されているものとします。

```
DWORD    ret;                //関数の戻り値
WORD     msts;               //メインステータス
DWORD    ersts, evsts;       //エラーステータス, イベントステータス
WORD     run = 0;            // 1 : 動作開始, 0 : 動作終了

ret = hcp530_WritOpMode( hDevID, 1, 0x41 );    //Y軸 : 動作モード=位置決め動作
ret = hcp530_WritPos   ( hDevID, 1, 20000 );    //      : +方向に20000パルス移動
ret = hcp530_WritFHSpd ( hDevID, 1, 10000 );    //      : 動作速度の設定 (速度倍率1倍: 10Kpps)
ret = hcp530_AccStart( hDevID, 0x02 );          //      : 加速スタート
run = 1;                                         //      : 動作開始
while( run ) {
    ret = hcp530_ReadMainSts( hDevID, 1, &msts );    //Y軸 : メインステータスの読み込み
    if( msts & 0x10 ) {                               //Y軸 : エラーステータスあり
        ret = hcp530_ReadErrorSts( hDevID, 1, &ersts );    //      : エラーステータスの読み込み
        run = 0;                                         //      : 動作終了 (異常)
    }
    if( msts & 0x20 ) {                               //Y軸 : イベントステータスあり
        ret = hcp530_ReadEventSts( hDevID, 1, &evsts );    //      : イベントステータスの読み込み
        run = 0;                                         //      : 動作終了 (正常)
    }
}
```

4. 5. 3 連続送りと停止コマンド

下記コーディング例中で、“PZSW”スイッチでZ軸を5Kppsの速度で高速の連続送りを行い、“ZSTP”スイッチで減速停止とします。

この動作では加減速形式、加速・減速レート等は既に設定されているものとします。

```
DWORD    ret;                //関数の戻り値
WORD     msts;               //メインステータス
DWORD    ersts, evsts;       //エラーステータス, イベントステータス
WORD     run = 0;            // 1 : 動作開始, 0 : 動作終了

//Z軸 : +方向連続送りのスイッチ指令
case PZSW:
    if( run != 0 )            break;                //動作中の為無視
    //動作開始
    ret = hcp530_WritOpMode( hDevID, 2, 0x00 );    //Z軸 : 動作モード=+方向連続動作
    ret = hcp530_WritFHSpd ( hDevID, 2, 5000 );    //      : 動作速度の設定 (速度倍率1倍: 5Kpps)
    ret = hcp530_AccStart( hDevID, 0x04 );          //      : 加速スタート
    run = 1;                                         //      : 動作開始
    break;

//Z軸 : 減速停止のスイッチ指令
case ZSTP:
    if( run == 0 )            break;                //停止中の為無視
    ret = hcp530_DecStop( hDevID, 0x04 );          //動作中ではZ軸に対して減速停止指令
    while( run ) {
        ret = hcp530_ReadMainSts( hDevID, 2, &msts );    //Z軸 : メインステータスの読み込み
        if( msts & 0x10 ) {                               //Z軸 : エラーステータスあり
            ret = hcp530_ReadErrorSts( hDevID, 2, &ersts );    //      : エラーステータスの読み込み
            run = 0;                                         //      : 動作終了 (異常)
        }
        if( msts & 0x20 ) {                               //Z軸 : イベントステータスあり
            ret = hcp530_ReadEventSts( hDevID, 2, &evsts );    //      : イベントステータスの読み込み
            run = 0;                                         //      : 動作終了 (正常)
        }
    }
    break;
```

4. 5. 4 タイマー動作

U軸で定速位置決め動作を利用して1秒間の停止を行います。

```
DWORD    ret;           //関数の戻り値
WORD     msts;          //メインステータス
DWORD    ersts, evsts;   //エラーステータス, イベントステータス
WORD     run = 0;        // 1 : 動作開始, 0 : 動作終了

ret = hcp530_WritOpMode( hDevID, 3, 0x47 );           //U軸 : 動作モード=タイマー動作
ret = hcp530_WritFHSpd ( hDevID, 3, 1000 );           //      : 動作速度の設定 (速度倍率 1 倍: 1 Kpps)
                                                    //      ( 1 パルスに対して 1 msec に相当)
ret = hcp530_WritPos( hDevID, 3, 1000 );              //      : +方向に 1 0 0 0 パルス
                                                    //      ( 1 0 0 0 msec = 1 秒)
ret = hcp530_CnstStartFH( hDevID, 0x08 );             //      : F H 定速スタート
run = 1;                                               //      : 動作開始
while( run ) {
    ret = hcp530_ReadMainSts( hDevID, 3, &msts );     //U軸 : メインステータスの読込
    if( msts & 0x10 ) {                                //U軸 : エラーステータスあり
        ret = hcp530_ReadErrorSts( hDevID, 3, &ersts ); //      : エラーステータスの読込
        run = 0;                                       //      : 動作終了 (異常)
    }
    if( msts & 0x20 ) {                                //U軸 : イベントステータスあり
        ret = hcp530_ReadEventSts( hDevID, 3, &evsts ); //      : イベントステータスの読込
        run = 0;                                       //      : 動作終了 (正常)
    }
}
```

5. ポートおよびレジスタの書込み, 読出し

本節ではポートへ書込み, ポートからの読出しをドライバを通して行う方法を解説します。ドライバにより直接ポートにアクセスする事により自在な運用が行えます。

5章以降は 主としてドライバ関数による説明を行います。

「図3. 3-1 ポートとレジスタ配置」に示すように各軸にCMD, BUF0, BUF1, MST S, SST Sポートがあります。

PCLはコマンド, データをCMD, BUF x ポート経由で指定するレジスタに書くことによって, 動作条件の書込み, 読出しを行います。動作開始, 停止などはCMDポートに書込むことにより直接行います。

ポートアドレスは「表3. 2-1 HPCI-CPD578, 534, 532ポート表」を参照して下さい。

動作状態はステータス・ポートMST S (メインステータス) とSST S (サブステータス) を読むことによって直接知ることができます。

5. 1 CMD, BUFx 書込み, 読出し方法

CMDポートに与えられるコマンドは次の3通りに分類されます。

- (1) 動作コマンド・・・データを伴わないコマンド。(スタート, ストップ, 速度変更などのコマンド)
- (2) 制御コマンド・・・データを伴わないコマンド。(カウンタリセット, カンタラッチ, 偏差カウンタクリア, プリッジ制御などのコマンド)
- (3) レジスタ制御コマンド・・・BUF0, 1 レジスタ書込みデータを伴うコマンド。
あるいはレジスタデータ読出しコマンド。
(コマンド書込み後, BUF0, 1 にレジスタデータがセットされる)

5. 1. 1 データを伴わないCMD形式

動作および制御コマンド形式は図5. 1-1の通りです。

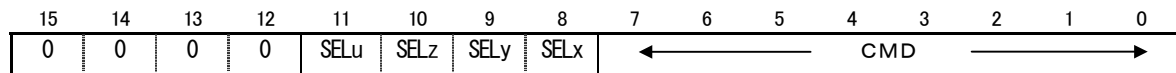


図5. 1-1 CMDポートの形式

■発行手順：CMD部とSEL部を一括してCMDポートへ書く（X軸:Board_ADR+0, Y軸: Board_ADR+8, ...）

■CMD部：動作コマンドあるいは制御コマンド。（⇒7. コマンド）

■SEL部：原則として動作軸を指定する。ただし、全てのSELビットが0の場合は自軸指定と同じ。

動作コマンドあるいは制御コマンドで軸指定の際に考慮すべき場合があります。

◆補間制御以外の場合

自軸を含み他軸も指定した場合は、他軸も同じコマンドで動作する。（同時実行となる。あるいはスタートコマンドの場合は同時スタートとなる。）

他軸のみ指定してコマンドを書いたときは、他軸がこのコマンドで動作する。

<注意> 複数のPCLにまたがる同時スタートは「同時スタートコマンド(06h)」を使用します。

（⇒7. 1）

◆補間制御の場合

自軸を含み補間軸全てを指定する。（該当軸の動作モードを補間モードにしてあること。）

（補間モードを指定しない軸を混在させてコマンドを書いた場合は、該当モードで動作する。）

【CPD534 ドライバ関数による記述例】

■発行例：

<例1> X軸にスタートコマンドを発行する。 ret= cp530_wCmdW(hDevID, 0, 0x051);

<例2> X,Z軸を同時にスタートコマンドを発行する。 ret= cp530_wCmdW(hDevID, 0, 0x551);

<例3> X軸をサボ ON する。 ret= cp530_wCmdW(hDevID, 0, 0x010);

5. 1. 2 レジスタ書き込み、読出し

■レジスタは次のグループに分類されます。

- | | |
|---------------------|--|
| （1）書き込み、読出し可能なレジスタ群 | 「動作レジスタ」（移動など動作に直接関係する）
「環境レジスタ」（初期に設定し動作環境を作る）
その他、カウンタやコンパレータを操作するレジスタ |
| （2）読出し専用レジスタ群 | 拡張ステータスなど状態の読出しのみ出来るレジスタ |

■レジスタ書き込みの時期とグループ

- | | |
|------------------------------|----------------------|
| （1）一度初期化すれば良いレジスタ群 | 「環境レジスタ」 |
| （2）初期化後必要によっては操作するレジスタ群 | カウンタやコンパレータを操作するレジスタ |
| （3）原則として動作コマンド発行前にセットするレジスタ群 | 「動作レジスタ」 |

■「動作レジスタ」書き込みの注意

- （1）動作レジスタにはプリレジスタが置かれています。（図5. 1-2 動作レジスタ構成）
- （2）動作レジスタの書き込みは通常下図のように「プリレジスタ（2nd プリレジスタ）」に対して書き込みます。
（1nd プリレジスタには書き込みません。）
- （3）位置オーバーライド・速度オーバーライドなどは現在動作中の軸の「レジスタ」に直接書き込みます。

■次動作連続実行（⇒10. 7 次動作連続実行）

プリレジスタを連続して使うことにより、短い線分のような動作も切れ目なく実行できます。

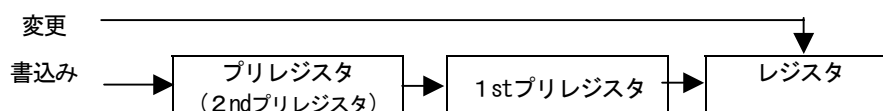
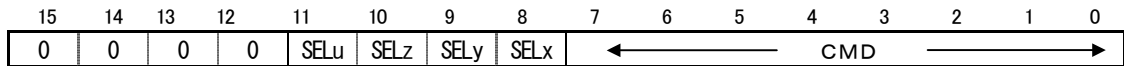


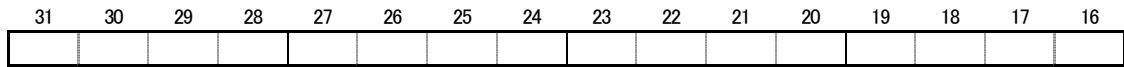
図5. 1-2 動作レジスタ構成

(1) レジスタ書き込み, 読出しのコマンド, データ形式

■ CMD形式



■ BUF 1 形式 (上位)



■ BUF 0 形式 (下位)

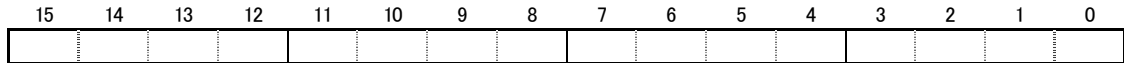


図 5. 1-3 レジスタ書き込み, 読出しのCMD, BUF 0, BUF 1 の形式

(2) レジスタ書き込み

■ 発行手順: ①レジスタへの書き込みデータをBUF 1, BUF 0へ書く。

②書き込み目的レジスタコマンドをCMD部, 書き込み軸をSEL部へ設定し, CMDポートへ書込む。

■ CMD部: 書き込み目的レジスタコマンド (⇒ 8. レジスタ制御コマンド解説)

■ SEL部: ①自軸のレジスタを指定するとき, SEL部を0にする。あるいは自軸のみ指定。

②自軸と他軸を指定した場合, 他軸のレジスタには他軸のBUF 0, 1の内容が書込まれる。

＜注意＞レジスタ書き込みの場合, 同じ値を複数軸に書くときでも各軸のBUF xにはデータをあらかじめ書いておかなければならない。

■ BUF 0, 1: ①BUF 0, BUF 1ポートへの書き込み順は自由。(ドライバ関数使用では一括データ処理)

②数値データは右詰に書込む。(レジスタ長によらず常に32ビット長として書込む)

③論理データはレジスタのビット長以外のビットは0である

【 CPD534 ドライバ関数による記述例 】

■ 発行例:

＜例 1＞ X軸～U軸の「環境レジスタ1」をセットする。

```
ret=cp530_wBufDW( hDevID, 0, 0xac013004 ); //X 軸 BUF に環境条件 1 セット
ret=cp530_wBufDW( hDevID, 1, 0xac013004 ); //Y 軸 BUF に環境条件 1 セット
ret=cp530_wBufDW( hDevID, 2, 0xac013004 ); //Z 軸 BUF に環境条件 1 セット
ret=cp530_wBufDW( hDevID, 3, 0xac013004 ); //U 軸 BUF に環境条件 1 セット
ret=cp530_wCmdW ( hDevID, 0, 0xf9c); //X, ~, U 軸の RENV1 に書き込み
```

＜例 2＞ X, U軸の「動作モードプリレジスタ」を位置決め動作モードにセットする。(同レジスタの他条件もセットする。)

```
ret=cp530_wBufDW( hDevID, 0, 0x441 ); //X 軸 BUF にモードセット, S 字加減速
ret=cp530_wBufDW( hDevID, 3, 0x441 ); //U 軸 BUF にモードセット, S 字加減速
ret=cp530_wCmdW ( hDevID, 0, 0x987 ); //X, U のモードレジスタに BUF 書き込み
```

＜例 3＞ X, U軸の「速度倍率プリレジスタ」に速度倍率を10にセットする。(速度倍率10)

```
ret=cp530_wReg( hDevID, 0, 0x85, 29 ); //X 軸倍率レジスタに倍率 10 書き込み
ret=cp530_wReg( hDevID, 3, 0x85, 29 ); //U 軸倍率レジスタに倍率 10 書き込み
```

＜例 4＞ X, U軸の「FHプリレジスタ」に動作速度100Kppsをセットする。(速度倍率10)

```
ret=cp530_wReg( hDevID, 0, 0x82, 10000 ); //X 軸 PRFH レジスタに速度書き込み
ret=cp530_wReg( hDevID, 3, 0x82, 10000 ); //U 軸 PRFH レジスタに速度書き込み
```

＜例 5＞ X軸の「送り量プリレジスタPRMV」に移動量10000をセットする。

```
ret=cp530_wReg( hDevID, 0, 0x80, 10000 ); //X 軸 PRMV レジスタに移動量書き込み
```

＜例 6＞ U軸の「送り量プリレジスタPRMV」に移動量-10000をセットする。

```
ret=cp530_wReg( hDevID, 3, 0x80, -10000 ); //U 軸 PRMV レジスタに移動量書き込み
```

(3) レジスタ読出し

- 発行手順：①読出し目的レジスタコマンドをCMD部、読込み軸をSEL部へ設定し、CMDポートへ書込む。
②読出されたデータをBUF0、BUF1ポートから読む。
- CMD部：読出し目的のレジスタコマンドを書く。
- SEL部：①自軸のレジスタを指定するとき、SEL部を0にする。あるいは自軸のみ選択する。
②自軸と他軸を指定した場合、他軸のレジスタ内容も該当軸BUF0、1に読出される。
- BUF0、1：①BUF0、BUF1ポートからの読出し順は自由。（ドライバ関数使用では一括データ処理）
②数値データは右詰に読出され符号拡張される。
③論理データはレジスタのビット長以外のビットは0が詰められる。
④レジスタ長によらず常に32ビット長として読む。

【 CPD534 ドライバ関数による記述例 】

■ 発行例：

<例1> X,Y,Z,U軸の「指令位置カウンタ」を一斉(同時)に読取る。

```
ret=cp530_wCmdW( hDevID, 0, 0xfe3 ); //X,~,ZをSEL 読出し
ret=cp530_rBufDw( hDevID, 0, &Data0 ); //X BUF0,1 読出し
ret=cp530_rBufDw( hDevID, 1, &Data1 ); //↓
ret=cp530_rBufDw( hDevID, 2, &Data2 ); //↓
ret=cp530_rBufDw( hDevID, 3, &Data3 ); //U BUF0,1 読出し
```

<例2> U軸の「エラーステータス」をレジスタから読む。

```
ret=cp530_rReg( hDevID, 3, 0xf2, &Data3 );
```

5. 1. 3 MSTS、SSTSの読出し

動作コマンド、制御コマンドの結果はMSTS(メインステータス)で運用します。

(1) メインステータス読出し形式

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPDF	SPRF	SEOR	SCMP5	SCMP4	SCMP3	SCMP2	SCMP1	SSC1	SSC0	SINT	SERR	SEND	予約	SRUN	SSCM

図5. 1-4 メインステータス読出し形式

(2) サブステータス読出し形式

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDLS	SOLS	SMEL	SPEL	SALM	SFC	SFD	SFU	← 不使用(予約) →				不使用		SVRST	SVON

図5. 1-5 サブステータス読出し形式

【 CPD534 ドライバ関数による記述例 】

■ ステータス読出し例：

<例1> Y軸のMSTS読出し

```
ret = cp530_rMstsW( hDevID, //デバイスハンドル
1, //Y軸を指定
&msts ); //ステータス格納先のアドレス
```

6. ステータス解説

ステータスには各軸ごとに、次の種類があります。直接ポートから読出すMSTS(メインステータス)、SSTS(サブステータス)、レジスタで保有する形式のREST(エラーステータス)、RIST(イベントステータス)、およびRIRQ(イベントマスクレジスタ)です。

6. 1 メインステータス MSTS (読出し専用)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPDF	SPRF	予約	SCMP5	SCMP4	SCMP3	SCMP2	SCMP1	SSC1	SSC0	SINT	SERR	SEND	予約	SRUN	SSCM
bit	名称	説明													運用度
b0	SSCM	'1' = スタートが掛かっている (スタートコマンドを書いた、あるいはスタート保留中)													△
b1	SRUN	'1' = RUN 中													○
b2	—	予 約													X
b3	SEND	'1' = 終了状態あるいは停止状態													○
b4	SERR	'1' = 何らかのエラーが発生した。このビットはRESTレジスタを読んだ後クリアされる													○
b5	SINT	'1' = RIRQ でマスクした要因が発生した。このビットはRISTレジスタを読んだ後クリアされる。													○
b6	SSC0	実行中の動作モードレジスタRMDに付属する実行シークスNo. がb7, b6に表示される ●PRMDレジスタb17, b16にユーザが書いた番号が示される。番号順は任意 ●2段あるブリッパを含み実行がどこまで進んでいるか判断の手段													△
b7	SSC1														
b8	SCMP1	●イコール比較条件の場合、比較条件成立中の間のみ'1'となる。(レベル検出) 通過してしまった場合は'0'となるので(過去において)条件が成立したことを監視する場合、イベントステータス SINT を利用する。													X
b9	SCMP2														
b10	SCMP3														
b11	SCMP4														
b12	SCMP5														
b13	—	予 約													X
b14	SPRF	'1' = 動作用ブリッパレジスタ満杯 '0' = 空き (プリレジスタに書込み可)													△
b15	SPDF	'1' = コンパレータ5用ブリッパレジスタ満杯 '0' = 空き (プリレジスタに書込み可)													△

(注) 運用度 ○: 基本的運用のビット △: アプリケーションによっては使用 X: 通常使用しない

表 6. 1-1 メインステータス MSTS の内容

■ 解 説

- (1) b0 (SSCM): スタートコマンドを発行すると '1' となる。スタートコマンドを発行しても移動は始まらない場合がある (条件付スタートのスタート保留状態など)。
(⇒ 4. 3. 1 動作モードレジスタ)
- (2) b1 (SRUN): 動作状態 '1', 停止状態 '0'。
(状態を示している。b3の逆状態表示。電源投入後の表示は正常)
- (3) b3 (SEND): 停止状態 '1', 動作状態 '0'。(状態を示している。動作終了してもボードからCPUへの割り込みは発生しない。電源投入後は停止状態でも '0' なので注意。初めての動作が終了したのち、正しい状態表示となる。)
- (4) b4 (SERR): エラーステータスレジスタRESTのビットにエラーが生じていれば '1' となる。
エラーステータスレジスタを読むとSERRは '0' となる。
このビットはCPUへの割り込み要因となる。
- (5) b5 (SINT): イベントステータスレジスタRISTの (イベント発生報告可として) マスクされていないビットがセットされた場合 '1' となる。イベントステータスレジスタを読むとSINTは '0' となる。このビットはCPUへの割り込み要因となる。
- (6) b8 ~ b12 (SCMP1 ~ 5):
それぞれ、コンパレータの比較条件成立時 '1' で成立状態を示す。
コンパレータ比較条件が一瞬だけ成立するような場合 (例えばイコール比較条件など) を検出する場合、イベントステータスを利用する。
- (7) b14 (SPRF): 動作用ブリッパレジスタが満杯のときは '1'。この時ブリッパレジスタにデータを書込むと上書きされてしまうので '0' となるのを待ってブリッパレジスタにデータを書込む。
(⇒ 10. 7 次動作連続実行)
- (8) b15 (SPDF): CMP5用ブリッパレジスタが満杯のとき '1'。この時ブリッパレジスタにデータを書込むと上書きされてしまうので '0' となるのを待ってブリッパレジスタにデータを書込む。

6. 2 サブステータス SSTS (読出し専用)

ビット構成は前掲の 図 5. 1-5 を参照して下さい。

b 1 1 のサーボアラーム, b 1 2, b 1 3 のエンドセンサー, b 1 4 の原点センサー, b 1 5 の減速センサーなどの状態を試験的に知る必要がある場合 (センサ状態の試験など) 以外は サブステータスは通常の運用では使用しません。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDLS	SOLS	SMEL	SPEL	SALM	SFC	SFD	SFU	不	使	用	(予	約)	SVON

bit	名称	説 明											運用度		
b0	SVON	'1' = SVON ON				(指令状態のモニタ)							△		
b1	SVRST	'1' = SVRST ON													
b2	—	不使用													
b3															
b4															
b5															
b6															
b7															
b8	SFU	'1' = 加速中				●右の状態の間のみ'1'となる。(レベル検出)							×		
b9	SFD	'1' = 減速中													
b10	SFC	'1' = 定速動作中													
b11	SALM	'1' = SVALM (サーボアラーム信号ON中)						●これらのステータスは全てボードに入力されている信号状態を示す。 アプリケーションによってはこれらの状態を確認する場合に利用する。							△
b12	SPEL	'1' = +ELS検出中													
b13	SMEL	'1' = -ELS検出中													
b14	SOLS	'1' = OLS検出中													
b15	SDLS	'1' = DLS検出中													

(注) 運用度 △ : アプリケーションにより使用. X : 通常使用しない.

表 6. 2-1 サブステータス SSTS の内容

6. 3 エラーステータスレジスタ REST (読出し専用)

- エラーが生じてRESTの何れかのビットが '1' になった時, MSTS の b 4 SERR ビットが '1' となります。
- REST の各ビットは割り込みの要因となります。したがって, MSTS の SERR が '1' となった時は, その軸の割り込みマスクビット (RENV 1 およびボード割り込みマスク BINTM) が有効 (マスクされていない) であれば, CPU に割り込みます。(⇒ 6. 9 割り込み処理方法)
- REST は読み出しにより '0' リセットされます。同時に MSTS の SERR ビットもリセットされます。すなわち, MSTS の SERR ビットの '1' を検出した場合は 必ず, REST を読まなければなりません。
- REST 自体の各ビットは割り込みマスクは出来ません。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESAO	ESPO	ESIP	ESDT	0	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC5	ESC4	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	ESPE	ESEE

図 6. 3-1 エラーステータスレジスタ REST のビット構成

bit	名称	説明
b0	ESC1	'1' = CMP 1 条件成立で停止 (+SLS)
b1	ESC2	'1' = CMP 2 条件成立で停止 (-SLS)
b2	ESC3	'1' = CMP 3 条件成立で停止
b3	ESC4	'1' = CMP 4 条件成立で停止
b4	ESC5	'1' = CMP 5 条件成立で停止
b5	ESPL	'1' = +ELS検出で停止
b6	ESML	'1' = -ELS検出で停止
b7	ESAL	'1' = SVALM検出で停止
b8	ESSP	'1' = STP入力ONによる停止
b9	ESEM	'1' = EMG入力ONによる停止 (非常停止マッドによる停止またはオプションで非常停止機能を使用している場合)
b10	ESSD	'1' = DLS検出で減速停止 <オプションポートでPCSを選択時は常に'0'>
b11	未定義	常に'0'
b12	ESDT	'1' = 動作データが不正で停止
b13	ESIP	'1' = 補間動作中に補間の他軸に異常生起で停止
b14	ESPO	'1' = パルス入力速度が設定FHより速すぎて停止
b15	ESAO	'1' = 補間データのレンジオーバー生起で停止
b16	ESEE	'1' = エンコーダ信号エラー生起
b17	ESPE	'1' = パルス入力信号エラー生起
b18-31	未定義	常に'0'

表 6. 3-1 エラーステータスレジスタ RESTの内容

■ 解 説

RESTがセットされる原因は、不正に停止したと考えられる場合です。

- (1) b 0 (ESC1) :
b 1 (ESC2) : ソフトリミット用途で停止によりセットされる。(⇒10. 6. 3 ソフトリミット機能)
- (2) b 2 (ESC3) : 通常 脱調検出用に使用する。脱調検出して停止の場合 '1' となる。
- (3) b 3 (ESC4) : 通常 条件成立で停止、または速度変更に使用する。成立条件が速度変更の場合はこのビットはセットされない。イベントステータスで検出する。
- (4) b 4 (ESC5) : このビットはセットされない。
通常条件付他軸スタートに使用するため CMP 5 成立で停止はすることはない。
- (5) b 5 (ESPL) : エンドリミットで停止により '1' となる。(コネクタ +xELS 端子)
- b 6 (ESML) : エンドリミットで停止により '1' となる。(コネクタ -xELS 端子)
- (6) b 7 (ESAL) : サーボアラームを検出した。(コネクタ xSVALM 端子)
- (7) b 8 (ESSP) : 同時停止信号(STP)入力(または同時ストップコマンド)で停止した。
(<個別編>オプション参照)
次動作使用時の同時停止の際はプリレジキャンセルコマンドを発行してください。
- (8) b 9 (ESEM) : 非常停止信号入力で停止した(<個別編>オプション参照)
- (9) b 10 (ESSD) : 減速センサDLSにより減速停止した。
オプションポートでxPCS入力を選択時は常に '0' 。
- (10) b 12 (ESDT) : 動作データ不正で停止(補間軸 1 軸のみでスタート、3 軸以上で円弧補間、円弧補間の中心座標を(0, 0)とした時など)
- (11) b 13 (ESIP) : 補間動作中他の軸に異常生起して停止した。
- (12) b 14 (ESPO) : パルス送り時の設定速度(FH)より追従出来ないほどのパルス速度により停止した。
- (13) b 15 (ESAO) : 円弧補間軸の移動量が最大座標カウント値がオーバーし停止した。(2⁻²⁸~2²⁸)
- (14) b 16 (ESEE) : A相、B相が同時に変化した。エンコーダエラーは停止とならないので注意する。
- (15) b 17 (ESPE) : A相、B相が同時に変化した。同上

6. 4 イベントステータスレジスタ RIST（読み出し専用）

- イベントマスクレジスタ（RIRQ）でアンマスク（検出有効）されたビットに対応するRISTのビットが‘1’になった時、MSTSのb5SINTビットが‘1’となります。
- RISTのマスクされていないビットは割り込みの要因となります。したがって、MSTSのSINTが‘1’となった時は、その軸の割り込みマスクビット（RENV1およびボード割り込みマスクBINTM）が有効（マスクされていない）であれば、CPUに割り込みます。（⇒6.9 割り込み処理方法）
- RISTは読み出しにより‘0’リセットされます。同時にMSTSのSINTビットもリセットされます。すなわち、MSTSのSINTビットの‘1’を検出した場合は必ず、RISTを読まなければなりません。
- RISTのすべてのビットはイベントマスクレジスタ（RIRQ）によりマスクして使用します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISOL	ISLT	ISCL	ISC5	ISC4	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISND	ISNM	ISN	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	ISSA	ISMD	ISPD	ISSD

図 6. 4-1 イベントステータスレジスタ RISTのビット構成

bit	名称	説 明	RIRQ マスクビット	運用度 (注1)
b0	ISEN	動作が正常完了した。(注2)	IREN	○
b1	ISN	次動作継続スタートした。	IRN	×
b2	ISNM	動作用ブリッジが満杯状態から空ぎができた。	IRNM	×
b3	ISND	CMP5用ブリッジが満杯状態から空ぎができた。	IRND	×
b4	ISUS	加速開始	IRUS	×
b5	ISUE	加速終了	IRUE	×
b6	ISDS	減速開始	IRDS	×
b7	ISDE	減速終了	IRDE	×
b8	ISC1	+SLS検出した	IRCS	×
b9	ISC2	-SLS検出した		
b10	ISC3	CMP3比較条件成立	IRCS	△
b11	ISC4	CMP4比較条件成立		
b12	ISC5	CMP5比較条件成立		
b13	ISCL	CLR入力によるカウンタクリア時(CPD364, CPD434v2のみ)	IRCL	×
b14	ISLT	LTC入力によるカウンタラッチ時(CPD364, CPD434v2のみ)	IRLT	△
b15	ISOL	センサ原点OLSを検出し、カウンタをラッチした(RENV5:b12-13)	IROL	△
b16	ISSD	DLSを検出した(検出報告は“DLS無効(RENV1)”に関係しない)	IRSD	△
b17	ISPD	+DR入力時(CPD364のみ)	IRDR	×
b18	ISMD	-DR入力時(CPD364のみ)		×
b19	ISSA	STA入力時	IRSA	×
b20-31	未定義	常に‘0’		

(注1) 運用度 ○：通常使用。△：アプリケーションによっては使用。×：通常使用しない。

(注2) 正常完了のタイミングとほぼ同時に異常停止要因が発生すると正常終了報告とエラー停止報告の両方が発生する場合があります。この場合はエラー停止報告を優先してください。

表 6. 4-1 イベントステータスレジスタ RISTの内容

■ 解 説

異常停止あるいはCMP成立で停止などはエラーステータスによってMSTSに報告されるので、これらはRISTにおいては、一般的にはマスクして運用する。

(1) b0 (ISEN)：アプリケーションによらずマスクせずに運用するビット。

6. 5 イベントマスクレジスタ RIRQ

■ イベントステータスレジスタに対するイベント報告をマスク（報告せず）設定するレジスタです。

■ マスクは対応 bit を '0' で与えます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IROL	IRLT	IRCL	IRC5	IRC4	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IRND	IRNM	IRN	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	IRSA	IRDR	IRSD

図 6. 5-1 イベントマスクレジスタ RIRQ のビット構成

bit	名称	内 容	運用度 (注 1)
b0	IREN	'0' 動作完了報告マスク（報告しない） '1' にして運用する。	○
b1	IRN	'0' 次動作継続スタート報告マスク	×
b2	IRNM	'0' 動作用ブルジョア書込み可能報告マスク	×
b3	IRND	'0' CMP5 用ブルジョア書込み可能報告マスク	×
b4	IRUS	'0' 加速開始報告マスク	×
b5	IRUE	'0' 加速終了報告マスク	×
b6	IRDS	'0' 減速開始報告マスク	×
b7	IRDE	'0' 減速終了報告マスク	×
b8	IRC1	'0' CMP 1 比較条件成立報告マスク（+SL）	×
b9	IRC2	'0' CMP 1 比較条件成立報告マスク（-SL）	×
b10	IRC3	'0' CMP 3 比較条件成立報告マスク（脱調検出用途）	×
b11	IRC4	'0' CMP 4 比較条件成立報告マスク（一致で停止または速度変更用途）	△
b12	IRC5	'0' CMP 5 比較条件成立報告マスク（一致で他軸起動用途）	△
b13	IRCL	'0' CLR 入力によるカウンタクリア報告マスク	×
b14	IRLT	'0' L T C H 入力によるカウンタラッチ報告マスク	△
b15	IROL	'0' OLS 信号入力時にカウンタ値をラッチ報告マスク	△
b16	IRSD	'0' DLS 信号 ON 割込みマスク	×
b17	IRDR	'0' ±DR 信号 ON 割込みマスク	×
b18	IRSA	'0' STA 信号 ON 割込みマスク	×
b19-31	未定義	常に '0' を設定してください	

（注 1） 運用度 ○：通常使用。△：アプリケーションによっては使用。×：通常使用しない。

表 6. 5-1 イベントマスクレジスタ RIRQ の内容

■ 解 説

- （1） b 0 （ I R E N ）：動作正常終了報告マスク。通常マスクせずに運用する。
- （2） b 8 （ I R C 1 ）：
 - b 9 （ I R C 2 ）：ソフトリミット用途はエラーステータスで M S T S に報告されるのでマスクする。
- （3） b 1 0 （ I R C 3 ）：脱調検出用途はエラーステータスで M S T S に報告されるのでマスクする。
- （4） b 1 1 （ I R C 4 ）：速度変更報告が必要な場合はマスクしない。
- （5） b 1 2 （ I R C 5 ）：他軸起動アプリケーションの場合によりマスクしない。

6. 6 拡張ステータスレジスタ RSTS (読出し専用)

■ 拡張ステータスは通常の運用時は使用しません。ソフトウェア開発中にボードの動作状態を確認する時など使用します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDIN	SLTC	SCLR	SDRM	SDRP	SEZ	SERC	SPCS	SEMG	SSTP	SSTA	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	0	0	0	0	0	0	0	0	0	0	0	SINP

図 6. 6-1 拡張ステータスレジスタ RSTS のビット構成

bit	名称	内 容
b0	CND0	b3~b0=0~15 動作状態を表す
b1	CND1	0000: 停止中 1000: パルス入力待ち
b2	CND2	0001: DR 入力待ち 1001: FA (補助速度) 定速で動作中
b3	CND3	0010: STA 入力待ち 1010: FL 定速で動作中
		0011: 条件付きスタート待ち状態 1011: 加速中
		0100: 他軸の停止待ち状態 1100: FH 定速で動作中
		0101: サーボ偏差カウンタ・クリアタイムオン中 1101: 減速中
		0110: 指令方向信号切替中 1110: INPOS 信号待ち
		0111: バックラッシュ補正中 1111: 動作に入った
b4	SDIR	動作方向 (0: +方向, 1: -方向)
b5	SSTA	'1' = 同時スタート信号オン状態
b6	SSTP	'1' = 同時停止信号オン状態
b7	SEMG	通常 '0' <オプション: '1' = EMG 信号入力中>
b8	SPCS	通常 '0' <オプション: '1' = PCS 位置決めスタート信号入力中>
b9	SERC	'1' = サーボ偏差カウンタ・クリア信号出力中
b10	SEZ	'1' = エンコーダ Z 相信号入力中
b11	SDRP	'1' = +DR 信号入力中
b12	SDRM	'1' = -DR 信号入力中
b13	SCLR	'1' = CLR 信号入力中
b14	SLTC	'1' = L T C H 信号入力中
b15	SDIN	'1' = D L S 信号入力中 <位置決めスタート オプション選択時は '0' >
b16	SINP	'1' = I N P O S 信号入力中
b17	未定義	常に '0'
b21-18	予 約	予 約
b31-22	未定義	常に '0'

表 6. 6-1 拡張ステータスレジスタ RSTS の内容

■ 解 説

b0~b3 (CND):

これらの状態が重なっておきる場合がある。また、多くの場合は短い時間で変化する性格のものが多いので注意する必要がある。

- ① CND=3 h スタートコマンド発行されていて、軸スタート条件成立待ち状態。
- ② CND=4 h 指定軸の停止によりスタート待ち。
- ③ CND=5 h サーボ偏差カウンタ・クリアタイムオン中: RENV1 b14~b12において
サーボ偏差カウンタクリアのパルス幅 (13ms) を与えている。
CND=5は13msの間 '1' となる。
- ④ CND=6 h 指令方向信号切替中: 指令パルス出力形式が共通パルス (RENV1 b2~b0)
の場合方向切替時間 (0.2ms) がある。
- ⑤ CND=7 h バックラッシュ補正中: バックラッシュ補正中の時間は補正パルス量と FA (補助速度) による。
- ⑥ CND=A h 動作モードに関係なく、FL 定速で動作中の全ての場合。
- ⑦ CND=C h 動作モードに関係なく、FH 定速で動作中の全ての場合。
- ⑧ CND=F h RMDレジスタのスタート方法が「即スタート」である場合、
スタートコマンド書き込み後、動作中になるまでの状態。

6. 7 補間ステータスレジスタ RIPS（読出し専用）

- RIPSは補間設定状態と動作状態を示します。読出し専用です。
- 全軸共通レジスタであり、どの軸から読みだしても同一内容です。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPFu	IPFz	IPFy	IPFx	IPSu	IPSz	IPSy	IPSx	IPEu	IPEz	IPEy	IPEx	IPLu	IPLz	IPLy	IPLx
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	SED1	SED0	SDM1	SDM0	IPCC	IPCW	IPE	IPL

図6. 7-1 補間ステータスレジスタ RIPSのビット構成

bit	名称	内 容
b0	IPLx	'1'=X (V, X1, X2, X3) 軸が直線補間モード
b1	IPLy	'1'=Y (W, Y1, Y2, Y3) 軸が直線補間モード
b2	IPLz	'1'=Z (A, Z1, Z2, Z3) 軸が直線補間モード
b3	IPLu	'1'=U (B, U1, U2, U3) 軸が直線補間モード
b4	IPEx	'1'=X (V, X1, X2, X3) 軸がPCL間の直線補間モード
b5	IPEy	'1'=Y (W, Y1, Y2, Y3) 軸がPCL間の直線補間モード
b6	IPEz	'1'=Z (A, Z1, Z2, Z3) 軸がPCL間の直線補間モード
b7	IPEu	'1'=U (B, U1, U2, U3) 軸がPCL間の直線補間モード
b8	IPSx	'1'=X (V, X1, X2, X3) 軸が円弧補間モード
b9	IPSy	'1'=Y (W, Y1, Y2, Y3) 軸が円弧補間モード
b10	IPSz	'1'=Z (A, Z1, Z2, Z3) 軸が円弧補間モード
b11	IPSu	'1'=U (B, U1, U2, U3) 軸が円弧補間モード
b12	IPFx	'1'=X (V, X1, X2, X3) 軸が合成速度一定指定
b13	IPFy	'1'=Y (W, Y1, Y2, Y3) 軸が合成速度一定指定
b14	IPFz	'1'=Z (A, Z1, Z2, Z3) 軸が合成速度一定指定
b15	IPFu	'1'=U (B, U1, U2, U3) 軸が合成速度一定指定
b16	IPL	'1'=直線補間で動作中
b17	IPE	'1'=PCL間の直線補間で動作中
b18	IPCW	'1'=CW方向円弧補間動作中
b19	IPCC	'1'=CCW方向円弧補間動作中
b20	SDM0	円弧補間の現在象限 b21=0, b20=0 : 第1象限 b21=0, b20=1 : 第2象限 b21=1, b20=0 : 第3象限 b21=1, b20=1 : 第4象限
b21	SDM1	
b22	SED0	円弧補間の終点象限 b23=0, b22=0 : 第1象限 b23=0, b22=1 : 第2象限 b23=1, b22=0 : 第3象限 b23=1, b22=1 : 第4象限
b23	SED1	
b24-31	未定義	常に'0'

表6. 7-1 補間ステータスレジスタ RIPSの内容

※. 2軸ボードはZ, U軸はありません。

6. 8 ステータス運用解説

6. 8. 1 イベント報告方式

ステータス運用はポーリング方式か割り込み方式かによらずイベントステータス レジスタ、エラーステータス レジスタよりそれぞれの発生報告をMSTSのSINT (b 5), SERR (b 4) を監視してステータス運用します。

この方法によれば コマンド発行後MSTSのみを読めばよく、b 5, b 4の何れかに‘1’を検出したときのみ、対象とする軸のRISTあるいはRESTを読んで、終了処理あるいはエラー処理をします。
(イベントステータス処理)

■イベントマスクの設定

イベント報告方式ではマスクの設定が重要です。イベントマスクレジスタRI RQのb 0 (動作完了マスク) は常に‘1’として運用します。これは初期化の段階で設定しておきます。(ライブラリ関数ではDevOpen 関数で実施)

RI RQの他のビットは必要に応じてアンマスクを実施します。この場合注意すべき点は、条件が成立している状態のビットに対してアンマスク (マスクを解除) したとき、MSTSのSINTビットがセットされます。この場合は必ずイベントステータス処理を行っておきます。

■イベントステータス処理

イベント報告方式ではMSTSのb 5またはb 4に‘1’を検出したときは、それぞれRIST, RESTを読まなければなりません。これは要因解析をするまでもない場合 (b 0動作終了のみで運用) においても、要因の発生元を読まないでMSTSのb 5, b 4はリセットされないからです。イベント報告方式はパルス状の瞬時生起するステータスも報告できます。(コンパレータのイコール条件成立報告、エンコーダなどのパルス状に生起するエラーなど。)

■イベント報告方式の処理フロー

図6. 8-1 にイベント報告方式の処理フロー示します。

■イベント報告方式のステータス運用例：

図6. 8-1 イベント報告方式のステータス運用フローのプログラム例を以下に示します。
ただし、説明の平易のため1軸のみの処理例です。

【CPD534 ドライバ関数による記述例】

```
WORD      xsts;      //MSTS
DWORD     rest,rist; //REST,RIST

cp530_wReg(hDev, 0, 0x80, 10000); //X 軸移動量
cp530_wReg(hDev, 0, 0x87, 0x8008041); //X 軸位置決めモード
cp530_wCmdW(hDev, 0, 0x53); //X 軸加速スタート

// 動作終了確認
do {
    cp530_rMstsW(hDev, 0, &xsts); //X 軸メインステータス読込
    if(xsts & 0x0030) { //b5 or b4 検出した
        if(xsts & 0x0010) {
            //エラー処理
            cp530_rReg(hDev, 0, 0xf2, &rest); //X 軸エラーステータス読込
            { エラー解析&処理 }
        } else {
            //イベントステータスクリア
            cp530_rReg(hDev, 0, 0xf3, &rist); //X 軸イベントステータス読込
        }
    }
} while(!(xsts & 0x0030));
```

■ イベント報告方式のステータス運用フロー例

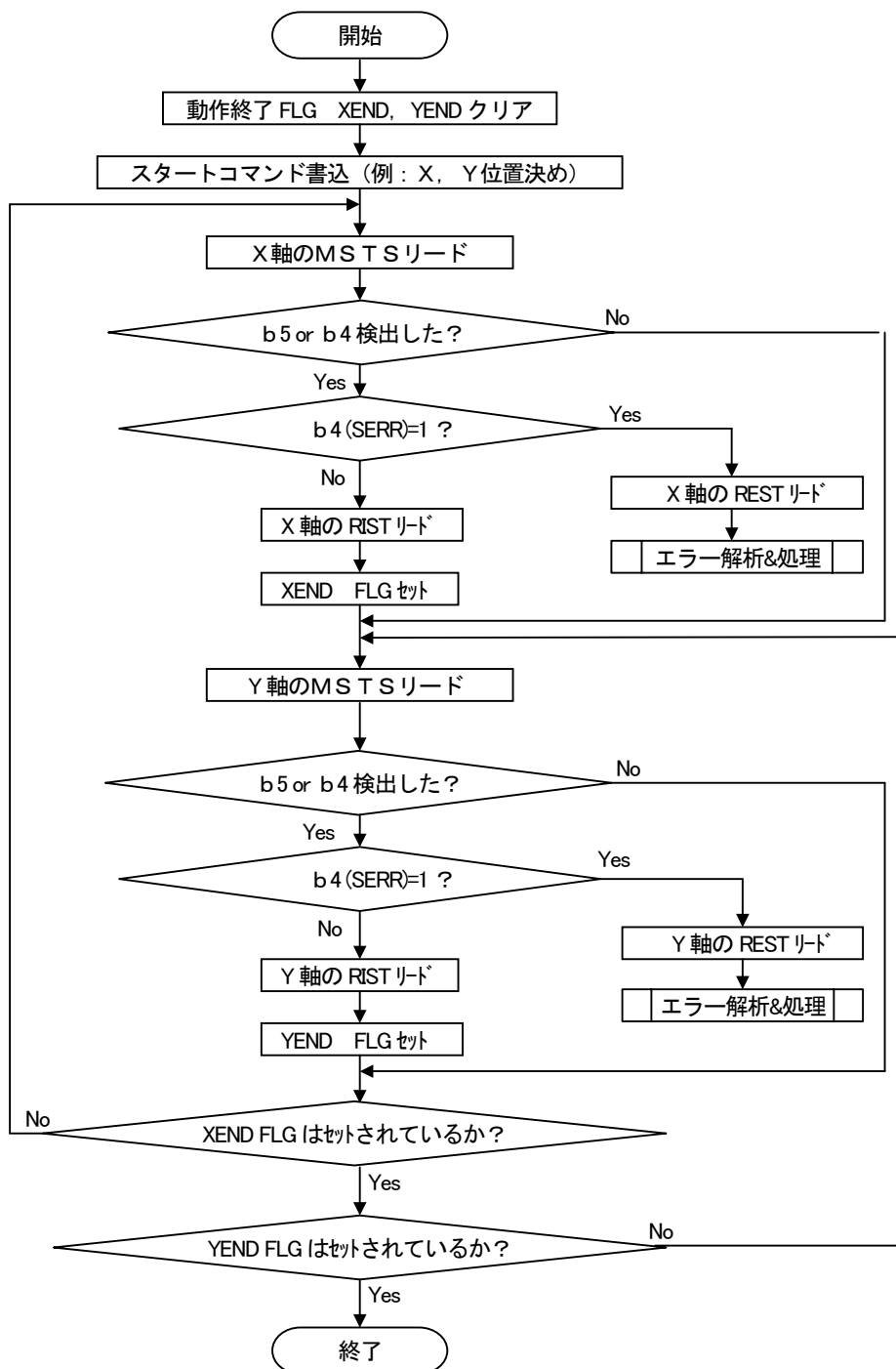


図6. 8-1 イベント報告方式のステータス運用フロー例

6. 8. 2 補間の場合のステータス運用

補間動作の場合のステータス処理はイベント報告方式によって 次のようにします。

- (1) MST Sは代表軸を読みます。
- (2) MST Sのb 5 または b 4に '1' を検出したら全ての補間軸のR I S TまたはR E S Tを読みます。
- (3) 正常終了の場合はR I S Tのb 0（動作終了）だけの運用の場合は各軸のb 0がセットされます。
- (4) 異常終了の場合は 異常の生じた軸のR E S Tの該当ビットに要因がセットされ、他の軸にはb 1 3がセットされます。

■ 補間動作時のステータス運用フロー例

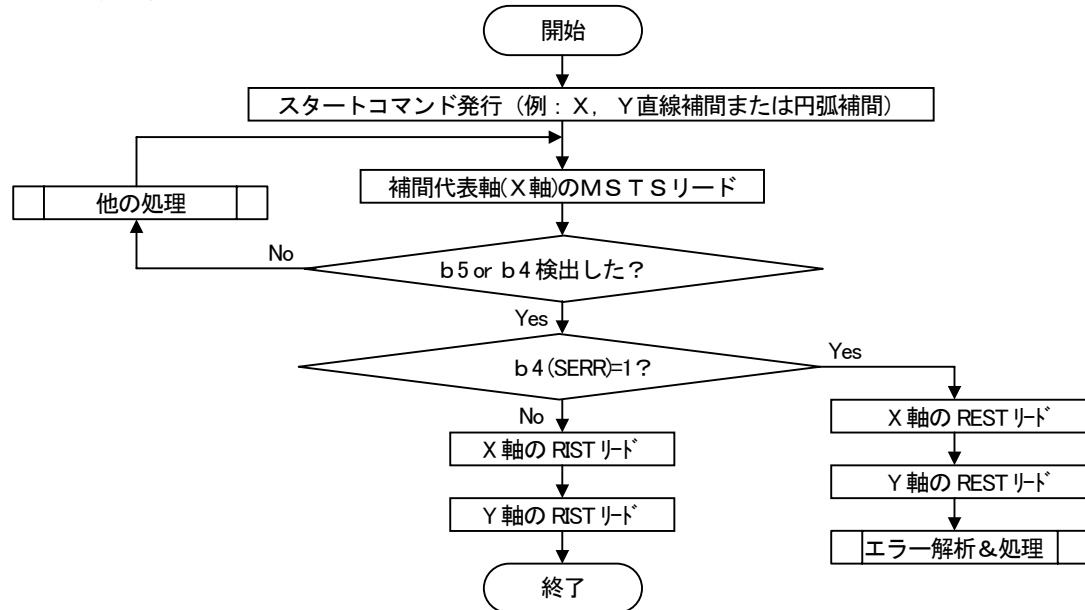


図6. 8-2 補間動作時のステータス運用フロー例

■補間動作時のステータス運用例：

図6. 8-2 補間動作時のステータス運用フローのプログラム例を以下に示します。

【CPD534 ドライバ関数による記述例】

```

WORD    xend;           //動作終了フラグ
WORD    xsts;           //MSTS
DWORD   xrest, yrest;   //REST
DWORD   xrist, yrist;   //RIST
cp530_wReg(hDev, 0, 0x80, 10000); //X軸移動量
cp530_wReg(hDev, 0, 0x87, 0x8008061); //X軸直線補間モード
cp530_wReg(hDev, 1, 0x80, 50000); //Y軸移動量
cp530_wReg(hDev, 1, 0x87, 0x8008061); //Y軸直線補間モード
xend = 1;               //動作終了フラグセット
cp530_wCmdW(hDev, 0, 0x353); //加速スタート
// 動作終了確認
while(xend) {
    cp530_rMstsW(hDev, 0, &xsts); //代表軸(X軸)メインステータス読み
    if(xsts & 0x0030) {           //b5 or b4 検出した
        xend = 0;                //X軸動作完了
        if(xsts & 0x0010) {
            //エラー処理
            cp530_rReg(hDev, 0, 0xf2, &xrest); //X軸:エラーステータス読み
            { エラー解析&処理 }
            cp530_rReg(hDev, 1, 0xf2, &yrest); //Y軸:エラーステータス読み
            { エラー解析&処理 }
        } else {
            //イベントステータスクリア
            cp530_rReg(hDev, 0, 0xf3, &xrist); //X軸:イベントステータス読み
            cp530_rReg(hDev, 1, 0xf3, &yrist); //Y軸:イベントステータス読み
        }
    }
}
  
```

6. 9 割込み処理方法

ステータス割込み方式は DOS あるいは RTOS 等のリアルタイム性の得られるOSのもとで可能です。

割り込みによりステータス処理を行う場合はBus に割り込み信号を出します。その方法は「図6. 9-1 ボード内割り込みルート」を参照して下さい。

(1) Bus への割込みマスク設定

各軸の割込みマスク（RENV1 b 2 9）および「ボード割込みマスク BINTM」（オプションポート）の設定を初期時にしておきます。

各軸の割込みマスクは軸の RENV1 INTM(b29) = '0' にします。(割込み許可, アンマスク)

使用しない軸はマスクします. (INTM(b29) = '1')

オプションポートはBINTM b0 = '1' (割込み許可, アンマスク) にしておきます.

(2) 割込み処理 (具体的なプログラムは「DOS版サンプル」を参照して下さい。)

CPUから割込みを受ける割込みエントリーにおいて行う事は次の通りです。

- ①オプションポートの割込みマスクBINTM b0 = '0' (割込み禁止)にする。
- ②オプションポートのボード割込みステータスBINTSを読む。b0 = '0' (割込み発生中)
- ③イベント報告方式のステータス運用処理をする。
- ④割込みから抜出すときにオプションポートの割込みマスクBINTM b0 = '1' (割込み許可)にする。

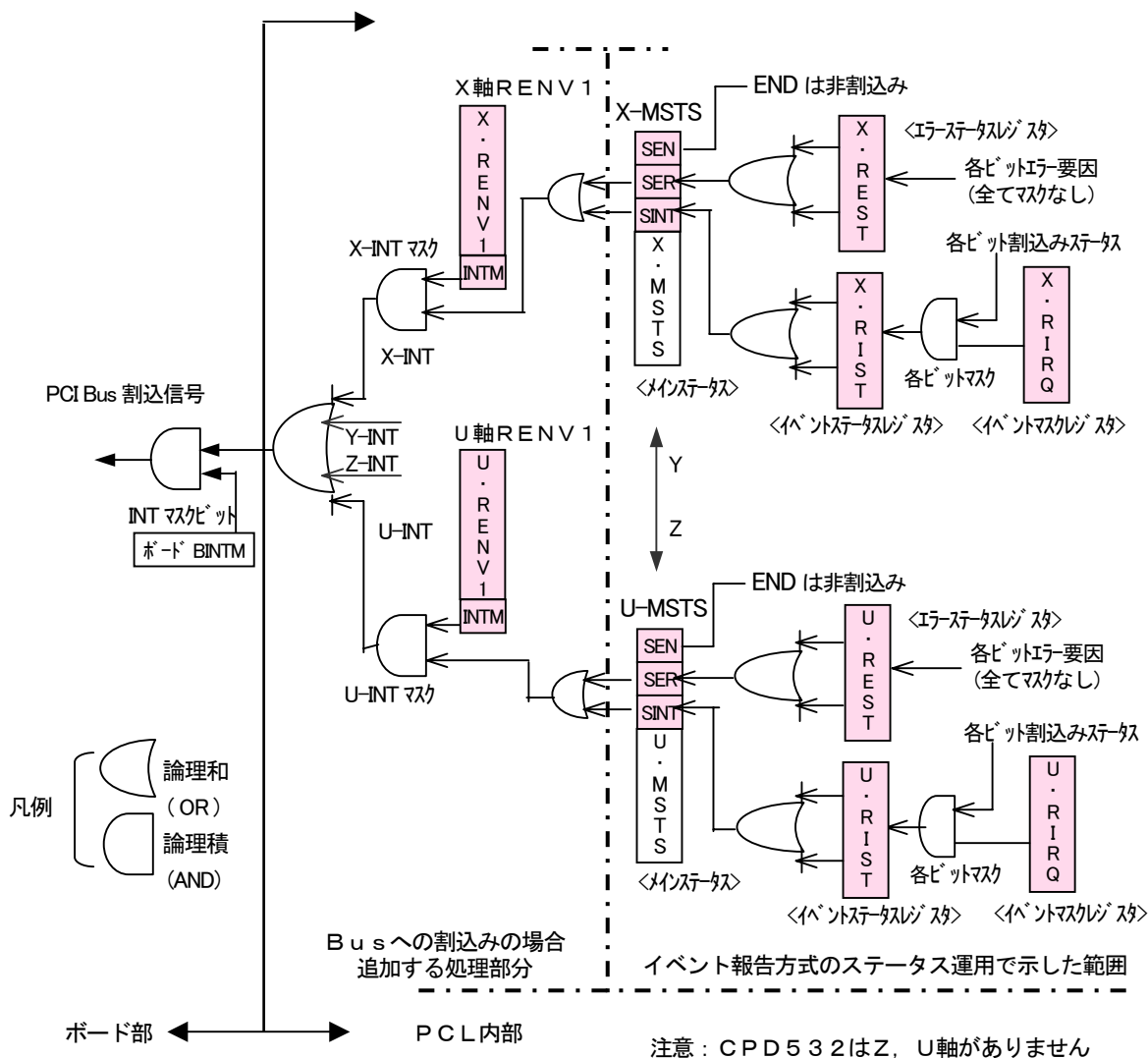


図6. 9-1 ボード内割込みルート

7. コマンド

7. 1 動作コマンドおよび制御コマンド

コマンドには「5. 1 CMD, BUF x 書込み, 読出し方法」で説明した通り, 動作コマンド, 制御コマンド, レジスタ制御コマンドがあります。

動作コマンドについては「4. 3. 4 動作コマンド」で基本のスタート, ストップコマンドを説明してあります。この節ではその他の動作コマンドと制御コマンドについて説明します。

レジスタ制御コマンドは次章「8. レジスタ制御コマンドおよびレジスタ解説」を参照して下さい。

コマンド名			CMD	機 能	略 称	備 考
動作 コマ ン ド	スタート コマ ン ド	加速スタート	5 3 h	ベース速度 (F L) から動作速度 (F H) へ加速し F H 定速で動作後減速停止。減速開始点手動設定時減速開始点 = 0 で加速ブロック開始となる	accSTART	「4. 3. 4 動作 コマンド」参照
		F H 定速スタート	5 1 h	動作速度 (F H) 定速スタートで F H 継続移動定速ブロック開始	cnstSTART	
		F H 定速スタート後 減速停止	5 2 h	動作速度 (F H) 定速スタート後減速停止 減速開始点手動設定で減速ブロック開始	decSTART	
		F L 定速スタート	5 0 h	ベース速度 (F L) 定速スタートで F L 定速移動	baseSTART	
	停止	即停止	4 9 h	即停止	STOP	「4. 3. 4 動作 コマンド」参照
		減速停止	4 a h	減速停止	DEC stop	
		非常停止	0 5 h	EMG stop	EMG stop	
	残量 スタート	残量 F L 定速スタート	5 4 h	残量をベース速度 (F L) で定速スタート	rbaseSTART	円弧補間では 使用不可
		残量 F H 定速スタート	5 5 h	残量を動作速度 (F H) で定速スタート	rcnstSTART	
		残量加速スタート	5 7 h	残量を動作速度 (F H) へ加速スタート	raccSTART	
		残量 F H 定速スタート後 減速停止	5 6 h	残量を動作速度 (F H) で定速スタート後 減速停止	rdecSTART	
	速度 変 更	F L 定速瞬時速度変更	4 0 h	F H 動作中瞬時にベース速度 (F L) に速度変更	FCHGL	
		F H 定速瞬時速度変更	4 1 h	F L 動作中瞬時に動作速度 (F H) に速度変更	FCHGH	
		F L 速度まで減速	4 2 h	F H 動作中減速してベース速度 (F L) に速度変更	FDECL	
		F H 速度まで加速	4 3 h	F L 動作中加速して動作速度 (F H) に速度変更	FACCH	
	同時スタート		0 6 h	同時スタート	CMSTA	2 個以上の P C L
	同時ストップ		0 7 h	同時ストップ	CMSTP	にまたがった場合
制 御 コ マ ン ド	ソフトウェアリセット		0 4 h	P C L をリセット (POWON 状態)	SRST	
	カ ウ ン タ リ セ ッ ト	CTR 1 リセット	2 0 h	CTR 1 (指令位置) を 0 クリア	CTR1CL	
		CTR 2 リセット	2 1 h	CTR 2 (機械位置) を 0 クリア	CTR2CL	
		CTR 3 リセット	2 2 h	CTR 3 (脱調検出偏差) を 0 クリア	CTR3CL	
		CTR 4 リセット	2 3 h	CTR 4 (汎用) を 0 クリア	CTR4CL	
	サーボ 偏差 カウンタ制御	信号出力	2 4 h	サーボ偏差カウンタクリア (SVCTRL) 出力	SVCTRL	信号幅は R E N V 1 設定
		信号リセット	2 5 h	SVCTRL 出力レベル時の SVCTRL 出力リセット	SVCTRLRST	
	ブ リ ジ 制 御 コ マ ン ド	動作ブリッジキャンセル	2 6 h	動作ブリッジの内容をキャンセル	PREGCAN	「1 0. 7 次動作連続 実行」参照
		CMP 5 ブリッジ キャンセル	2 7 h	CMP 5 用比較ブリッジの内容をキャンセル	PMPCAN	
		動作ブリッジ 次段シフト	2 b h	動作ブリッジの内容を次段にシフト	PREGSFT	
		CMP 5 ブリッジ 次段シフト	2 c h	CMP 5 用比較ブリッジの内容を次段にシフト	PMPSFT	
	カウンタ ラッチ		2 9 h	CTR 1 ~ CTR 4 を一斉にラッチ	LTCH	
	PCS コマンド		2 8 h	PCS 信号の代わりに PCS を実行	PCSON	
	サーボ ポ ー ト コ マ ン ド	サーボオン ON	1 8 h	サーボオン ON	SVON	SVON 端子 : 0 [V]
		サーボオン OFF	1 0 h	サーボオン OFF	SVOFF	SVON 端子 : High (EXTPOW) レベル
		サーボリセット ON	1 9 h	サーボリセット ON	SVRSTON	SVRST 端子 : 0 [V]
		サーボリセット OFF	1 1 h	サーボリセット OFF	SVRSTOFF	SVRST 端子 : High (EXTPOW) レベル

表 7. 1-1 動作コマンドおよび制御コマンド一覧

7. 2 動作コマンド（同時スタートコマンド，同時ストップコマンド）

同時スタート，同時ストップコマンドは 2 個以上の P C L に対して同時にスタートまたは ストップコマンドを発行する場合に使用します。具体的には 8 軸または 1 2 軸 C P D ボードの場合あるいは複数ボード間で 同時スタート，ストップ・オプションコネクタがある場合に使用します。

- 同時スタートコマンドは，動作モードレジスタのスタート条件を同時スタート (b19, b18= '01') にあらかじめ設定しておきます。
- 同時ストップコマンドは，動作モードレジスタのストップ条件を同時ストップ有効 (b24= '01') に設定しておきます。

4 軸以下の場合には同時スタート，ストップのコマンドは使用しません。通常のスタート，停止コマンドを使用します。（⇒ 4. 3. 4 動作コマンド）

7. 3 加速ブロック，定速ブロック，減速ブロック コマンド

加速ブロック (B L K)，(複数の) 定速ブロック (B L K)，減速ブロック (B L K) で構成するような送り制御は次の代表例で説明されます。

緩やかな自由曲線を直線線分により近似したプロファイルを高速に送る場合を考えます。①加速して直線移動する「加速 B L K」accSTART コマンドによりでスタートします。②以降を直線線分で構成された B L K によって cnstSTART コマンド による定速 (F H) で③まで移動します。最後は ④decSTART コマンドにより減速停止します。

(図 7. 3-1)

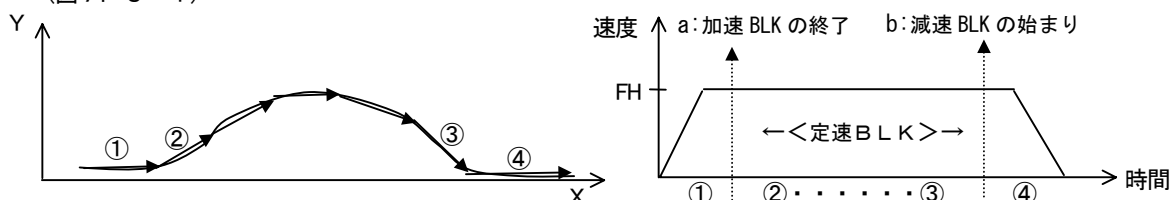


図 7. 3-1 線分の連続と加速 B L K，定速 B L K，・・・，減速 B L K

実際は，動作プリレジスタを使用して動作させます。加速 B L K，減速 B L K の加減速設定は減速開始点手動計算 (モードレジスタ P R M D の b13= '0' とする) として減速開始点 (R D P) などを設定する必要があります。(詳細 ⇒ 10. 7 次動作連続実行 および 6. 8. 3 プリレジスタ連続使用のステータス運用)

- ①加速 B L K：加速条件の時に減速開始点を '0' に設定します。(P R D P= '0')
移動量は加速終了して F H に達する距離を P R M V に設定します。
この条件で accSTART コマンドを発行すると，加速 B L K は図 7. 3-2 のように移動します。
すなわち，加速後 F H に達し加速 B L K の移動を終了します。
- ②定速 B L K：加速 B L K が実行中に，次の移動量を P R M V にセットし，定速 B L K 開始コマンド (51 h)
↓
を C M D ポートに書いておきます。以下，次の定速 B L K もプリレジスタに同様に書いておきます。(速度は変更無しとして P R F H を書く必要はありません。)
- ③
- ④減速 B L K：減速レートを P R D R に，減速開始点を P R D P に与えます。
移動量を P R M V にセットして，decSTART コマンド (52 h) を C M D ポートに書いておきます。

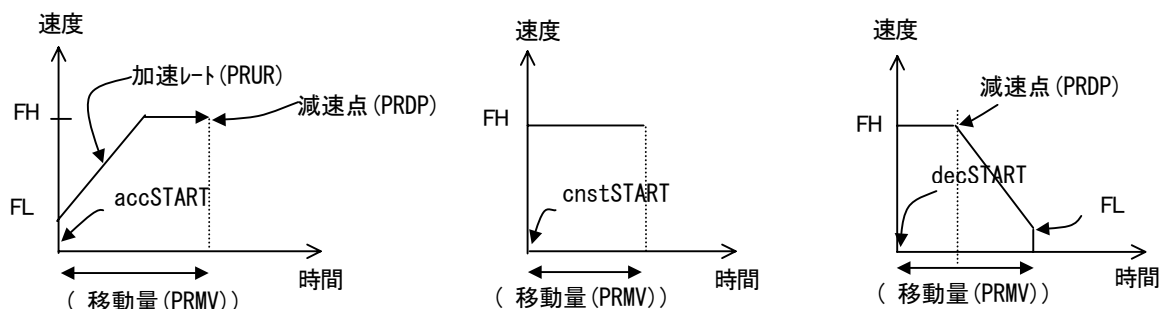


図 7. 3-2 加速 B L K，定速 B L K，減速 B L K

7. 4 残量スタートコマンド

残量スタートコマンドは直線補間送り、位置決め送りにおいて終点になる前に停止コマンドによって一時停止させ、そのまま継続スタートさせる場合に使用します。適当なF H速度による残量スタートコマンドを発行します。ただし、残量スタートコマンドは、円弧補間においては使用出来ません。

自動実行中に、一時停止してから手動送りを行うなどして後、再び、手動動作した移動点から残量スタートする場合（手動介入）は制限があります。すなわち、

- （１）プリパッファを使用しない運用であること
- （２）途中停止させ アプリの責任で動作レジスタ(RMV, RFH, RMD, RIP)を直接読み、セーブします。

手動介入して移動後、再びセーブした動作レジスタを元の通りそれぞれのレジスタに戻します。その後、残量スタートコマンドを発行します。この場合、その他のレジスタに影響がある手動介入は出来ません。

7. 5 カウンタ制御機能とコマンド

カウンタを制御するコマンドは以下の２つです。コマンド以外に、パラメータによりカウンタ動作を条件づけるビットがあります。

（１）カウンタリセット コマンド

コマンドに対応するカウンタ(CTR1～4)を‘0’リセットします。

（２）カウンタラッチ コマンドはカウンタの値(CTR1～4)を一斉にラッチレジスタ(RLTC1～4)にコピーします。読出しはRLTCxからおこないます。

（３）カウンタ制御に関するパラメータ機能

■原点復帰完了時に対応のRCTRxをリセットする選択がRENV3 b20～b23にあります。

■カウントする／しないはRENV3 b29～b31によって行うことができます。

ただし、指令位置カウンタ(CTR1)に対しては、動作モードレジスタ(RMD) b11で行います。

■バックラッシュ動作時にもカウント動作有効にする機能はRENV3 b24～b27にあります。（保守用途）

■カウンタラッチをコマンド以外で行う機能として、次の何れかを選択できます。RENV5 b12, b13で行います。

- ①OLS信号入力時にCTR1～4をラッチする。
- ②コンパレータ4 (CMP4) 成立時にCTR1～4をラッチする。
- ③コンパレータ5 (CMP5) 成立時にCTR1～4をラッチする。

7. 6 サーボ偏差カウンタクリア機能とコマンド

サーボモータ使用時に指令パルスの停止後、モータの停止はサーボドライバ内の偏差カウンタが‘0’になるまで遅れます。指令パルスの停止後サーボモータを即時に停止させるためには「サーボ偏差カウンタクリア信号（信号端子xSVCTRCL）」をサーボドライバに出力します。

この動作は通常は短い時間に行われるので、クリアコマンドは一般的に使用しません。

「サーボ偏差カウンタクリア」コマンドはこのことを考慮して使用する必要があります。

xSVCTRCLが自動的に出力される場合は以下の通りです。（いずれも出力可否の設定可）

- ①原点復帰完了時に自動的に出力する。（RENV1 b11により出力可否選択）
- ②ELS, SVALM, EMG(オプション)信号入力による停止時に出力。（EMGコマンドによる停止も同様）（RENV1 b10により出力可否選択）

xSVCTRCL信号出力はステータスと指令出力には次のタイミング関係があります。

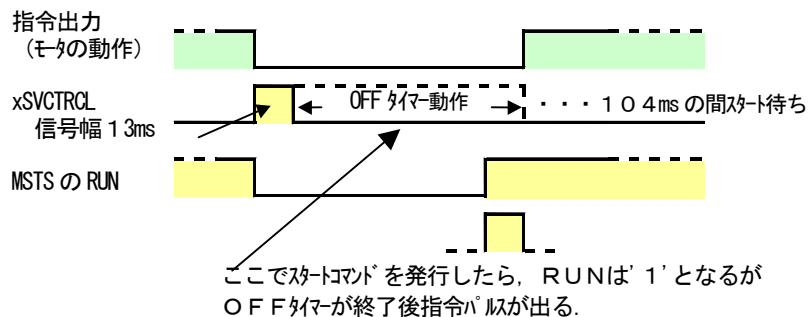


図7. 6-1 サーボ偏差カウンタクリア信号とタイミング関係

8. レジスタ制御コマンド および レジスタ解説

8. 1 レジスタ制御コマンド

ライブラリ関数を使用しない場合、全てのレジスタとプリレジスタの初期値はゼロです。
レジスタ制御コマンド一覧表を 表 8. 1-1 に掲げます。

No	レジスタ名	内 容	ビット長	読み込みコマンド	書き込みコマンド	プリレジスタ名	読み込みコマンド	書き込みコマンド	備 考
1	RMV	移動量	28	d0h	90h	PRMV	c0h	80h	動作レジスタは プリレジスタがある
2	RFL	ベース速度	16	d1h	91h	PRFL	c1h	81h	
3	RFH	動作速度	16	d2h	92h	PRFH	c2h	82h	
4	RUR	加速レート	16	d3h	93h	PRUR	c3h	83h	
5	RDR	減速レート	16	d4h	94h	PRDR	c4h	84h	
6	RMG	速度倍率	12	d5h	95h	PRMG	c5h	85h	
7	RDP	減速開始点	24	d6h	96h	PRDP	c6h	86h	
8	RMD	動作モード	27	d7h	97h	PRMD	c7h	87h	
9	RIP	円弧補間中心	28	d8h	98h	PRIP	c8h	88h	
10	RUS	加速時 S 字区間	15	d9h	99h	PRUS	c9h	89h	
11	RDS	減速時 S 字区間	15	dah	9ah	PRDS	cah	8ah	
12	RFA	補助速度	16	dbh	9bh				
13	RENW1	環境設定 1	32	dch	9ch				
14	RENW2	環境設定 2	27	ddh	9dh				
15	RENW3	環境設定 3	32	deh	9eh				
16	RENW4	環境設定 4	32	dfh	9fh				
17	RENW5	環境設定 5	22	e0h	a0h				
18	RENW6	環境設定 6	32	e1h	a1h				
19	RENW7	環境設定 7	32	e2h	a2h				
20	RCTR1	カウンタ 1 (指令位置用)	28	e3h	a3h				
21	RCTR2	カウンタ 2 (機械位置用)	28	e4h	a4h				
22	RCTR3	カウンタ 3 (脱調検出用)	16	e5h	a5h				
23	RCTR4	カウンタ 4 (汎用)	28	e6h	a6h				
24	RCMP1	コンパレータ 1 用データ	28	e7h	a7h				
25	RCMP2	コンパレータ 2 用データ	28	e8h	a8h				
26	RCMP3	コンパレータ 3 用データ	28	e9h	a9h				
27	RCMP4	コンパレータ 4 用データ	28	eah	aah				
28	RCMP5	コンパレータ 5 用データ	28	ebh	abh	PROCP5	cbh	8bh	コンパレータ 5 はプリレジ付
29	RIRQ	イベントマスク設定	19	ech	ach				
30	RLTC1	カウンタ 1 ラッチデータ	28	edh					読み込み専用
31	RLTC2	カウンタ 2 ラッチデータ	28	eeh					
32	RLTC3	カウンタ 3 ラッチデータ	16	efh					
33	RLTC4	カウンタ 4 ラッチデータ	28	f0h					
34	RSTS	拡張ステータス	17	f1h					
35	REST	エラーステータス	18	f2h					
36	RIST	イベントステータス	20	f3h					
37	RPLS	移動残量位置決め	28	f4h					
38	RSPD	Z 相カウンタ、速度	23	f5h					
39	RSDC	減速開始点自動計算値	24	f6h					
40	RIPS	補間ステータス	24	ffh					

表 8. 1-1 レジスタコマンド一覧

8. 2 RMV (PRMV) 移動量レジスタ

補間動作、位置決め動作において移動量を相対位置で設定します。

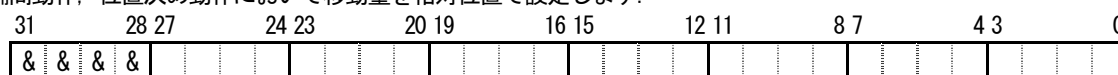


図8. 2-1 RMV (PRMV) 移動量レジスタ

- 設定範囲 -134, 217, 728 ~ +134, 217, 727
- 書込みはプリレジスタ PRMV に書込む。
- RMVに直接書込むと位置のオーバライドとなる。(⇒10. 3 位置のオーバライド)
- &表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。(符号拡張)

8. 3 RFL (PRFL) ベース速度レジスタ

加減速送りに於いて加速開始する速度、減速時に減速終了する速度。

(⇒4. 2. 5 速度および加減速関係の初期設定)

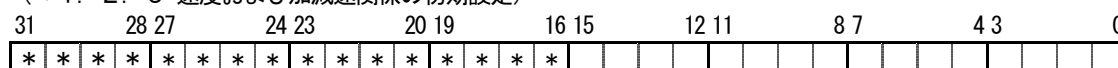


図8. 3-1 RFL (PRFL) ベース速度レジスタ

- 設定範囲 1 ~ 65, 535 ただし、速度(PPS)は速度倍率レジスタ(RMG)の速度倍率によります。
- $$\text{速度}F = \text{RFL} \times \frac{300}{\text{RMG} + 1} [\text{PPS}]$$
- 書込みはプリレジスタ PRFL に書込む。
 - *表記のビットは、書込み時には無視され、読出し時には0になる。

8. 4 RFH (PRFH) 動作速度レジスタ

動作速度を設定するレジスタです。(⇒4. 2. 5 速度および加減速関係の初期設定)

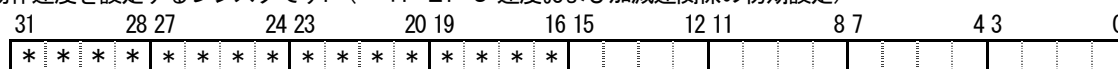


図8. 4-1 RFH (PRFH) 動作速度レジスタ

- 設定範囲 1 ~ 65, 535 ただし、実際の速度(PPS)は速度倍率レジスタ(RMG)の速度倍率によります。
- $$\text{速度}F = \text{RFH} \times \frac{300}{\text{RMG} + 1} [\text{PPS}]$$
- 書込みはプリレジスタ PRFH に書込む。
 - *表記のビットは、書込み時には無視され、読出し時には0になる。

8. 5 RUR (PRUR) 加速レートレジスタ

加速レートを設定するレジスタです。(⇒4. 2. 5 速度および加減速関係の初期設定)

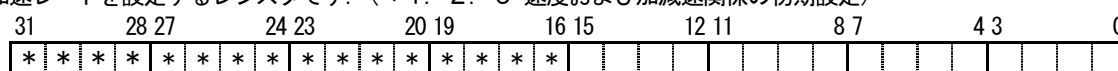


図8. 5-1 RUR (PRUR) 加速レートレジスタ

- 設定範囲 1 ~ 65, 535
- 書込みはプリレジスタ PRUR に書込む。
- *表記のビットは、書込み時には無視され、読出し時には0になる。

8. 6 RDR (PRDR) 減速レートレジスタ

減速レートを設定するレジスタです。(⇒4. 2. 5 速度および加減速関係の初期設定)

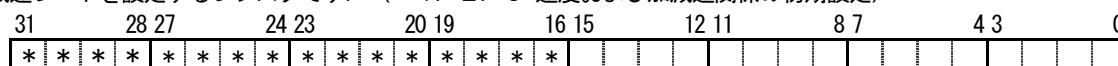


図8. 6-1 RDR (PRDR) 減速レートレジスタ

- 設定範囲 1 ~ 65, 535
- 書込みはプリレジスタ PRDR に書込む。
- *表記のビットは、書込み時には無視され、読出し時には0になる。

8. 7 RMG (PRMG) 速度倍率レジスタ

速度倍率を設定するレジスタです。(⇒4. 2. 5 速度および加減速関係の初期設定)

31	28 27	24 23	20 19	16 15	12 11	8 7	4 3	0
* * * *	* * * *	* * * *	* * * *	* * * *	* * * *			

図8. 7-1 RMG (PRMG) 速度倍率レジスタ

■設定範囲 2~4, 095

■書込みはプリレジスタ PRMG に書込む。

■*表記のビットは、書込み時には無視され、読出し時には0になる。

8. 8 RDP (PRDP) 減速開始点レジスタ

加減速動作を伴う位置決め時の減速開始点を設定するレジスタです。

自動減速開始点計算 (RMDレジスタ b13= '0') の時は自動計算値に対するオフセット値となります。

手動減速開始点計算の場合は終点からの減速移動量を設定します。(⇒10. 1 速度, 加減速の計算)

31	28 27	24 23	20 19	16 15	12 11	8 7	4 3	0
# # # #	# # # #							

図8. 8-1 RDP (PRDP) 減速開始点レジスタ

■#表記のビットは、書込み時に無視され、読出し時にRMDレジスタ (b13 減速開始点計算方法) によって異なる。

- (1) RMD b13= '0' 自動設定値に対するオフセット量を示し、正数は早めに減速する値、負数は遅めの減速。#ビットはビット23と同じ。
- (2) RMD b13= '1' 手動減速開始点計算設定。移動残量が設定値以下になると減速開始をする。

8. 9 RMD (PRMD) 動作モードレジスタ

解説は「4. 3. 1 動作モードレジスタ RMD (PRMD) と動作モード設定」を参照して下さい。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
合成速度	PCS 有効	減速開始点	0	CTRIDIS	acc 方法	INPSE	DLSE	← MOD →							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	1	FH補正	同他軸停止	同時ストップ	停止指定軸設定				条件付スタート		SEQ No	

図8. 9-1 RMD (PRMD) 動作モードレジスタ

8. 10 RIP (PRIP) 円弧補間中心位置レジスタ

円弧補間時の中心位置または2つ以上のPCLにまたがる直線補間の長軸移動量を設定するレジスタ。

31	28 27	24 23	20 19	16 15	12 11	8 7	4 3	0
& & & &								

図8. 10-1 RIP (PRIP) 円弧補間中心レジスタ

■RIPは、RMDレジスタのMOD (b6~b0) が次の動作モードの時有効。

- (1) 円弧補間 (MOD=64h, 65h)
- (2) 2個以上のPCL間の直線連続送り (MOD=62h)
- (3) 2個以上のPCL間の直線補間 (MOD=63h)

■円弧補間の場合は円弧中心位置を相対値で設定する。(⇒4. 3. 1. 2 (3) 補間動作モード)

■2個以上のPCL間の直線連続送り、直線補間の場合は長軸の移動量を相対値で設定する。

(⇒10. 8 複数のPCL間の動作方法)

■設定範囲 -134,217,727~+134,217,727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。(符号拡張)

8. 1 1 RUS (PRUS) およびRDS (PRDS) 加速および減速S字区間レジスタ

RUSはS字加速時のS字区間を設定するレジスタ。RDSはS字減速区間を設定するレジスタ。

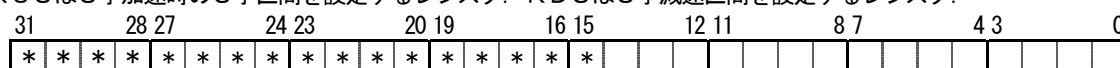
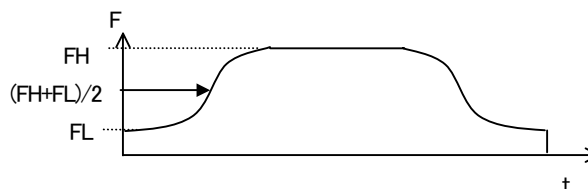


図8. 1 1-1 RUS (PRUS) およびRDS (PRDS) S字区間レジスタ

■ *表記のビットは、書き込み時には無視され、読出し時には0になる。

■ 設定範囲 1~32, 767

■ 0を設定すると $\frac{RFH-RFL}{2}$ の値が自動的に設定される。
すなわち、RUS=RDS=0の場合には完全にS字となる。



8. 1 2 RFA 補助速度レジスタ

バックラッシュ、スリップ補正時の速度および原点復帰モード4, 6, 7において原点突入速度として使用されます。

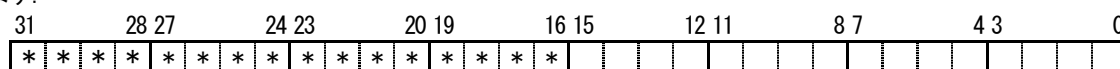


図8. 1 2-1 RFA 補助速度レジスタ

■ 設定範囲 1~65, 535 ただし、速度 (PPS) は速度倍率レジスタの設定値により変化する。

$$\text{速度} F = RFA \times \frac{300}{RMG+1} [PPS]$$

■ *表記のビットは、書き込み時には無視され、読出し時には0になる。

8. 1 3 RCTR1 指令位置カウンタ・レジスタ

指令位置パルス用 カウンタ・レジスタです。カウンタはアップダウン・カウンタです。CTR1への書き込みはRCTR1に行います。

特定のタイミングで読出したカウンタ値に精度が必要な場合は“ラッチ”を使用し、RLTC1から読出します。

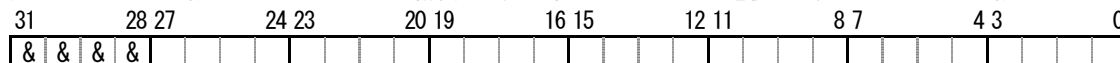


図8. 1 3-1 RCTR1 指令位置カウンタ・レジスタ

■ 設定範囲 -134, 217, 728~+134, 217, 727

■ &表記のビットは、書き込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。
(符号拡張)

8. 1 4 RCTR2 機械位置カウンタ・レジスタ

機械位置パルス用 カウンタ・レジスタです。カウンタはアップダウン・カウンタです。CTR2への書き込みはRCTR2に行います。

エンコーダ信号、パルサ信号および指令パルスが選択入力できます。(RENV3 b8, 9)

特定のタイミングで読出したカウンタ値に精度が必要な場合は“ラッチ”を使用し、RLTC2から読出します。

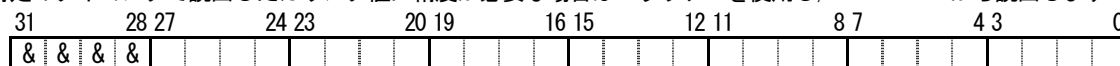


図8. 1 4-1 RCTR2 機械位置カウンタ・レジスタ

■ 設定範囲 -134, 217, 728~+134, 217, 727

■ &表記のビットは、書き込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。
(符号拡張)

8. 15 RCTR3 脱調カウンタ・レジスタ

脱調検出用 偏差カウンタ・レジスタです。CTR3への書込みはRCTR3に行います。

2つの信号の偏差（差分）をカウントします。指令パルスとエンコーダ信号、指令パルスとパルサ信号の何れかを選択できます。（RENV3 b10,11）

特定のタイミングで読出したカウンタ値に精度が必要な場合は“ラッチ”を使用し、RLTC3から読出します。

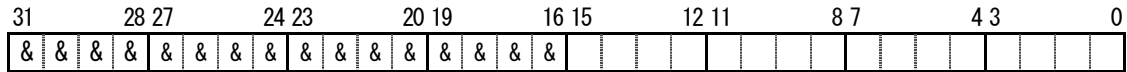


図8. 15-1 RCTR3 脱調カウンタ・レジスタ

■設定範囲 -32,768～+32,767

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。（符号拡張）

8. 16 RCTR4 汎用カウンタ・レジスタ

汎用パルス用 カウンタです。カウンタはアップダウン・カウンタです。CTR4への書込みはRCTR4に行います。

指令パルス、エンコーダ信号、パルサ信号および9.8304MHz クロック信号が選択入力できます。

（RENV3 b12,13）

特定のタイミングで読出したカウンタ値に精度が必要な場合は“ラッチ”を使用し、RLTC4から読出します。

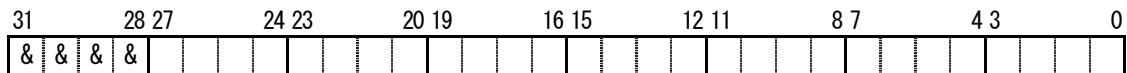


図8. 16-1 RCTR4 汎用カウンタ・レジスタ

■設定範囲 -134,217,728～+134,217,727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。（符号拡張）

8. 17 RCMP1 コンパレータ1比較レジスタ

コンパレータ1用比較データを設定します。

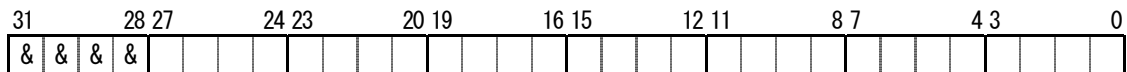


図8. 17-1 RCMP1 コンパレータ1比較レジスタ

■設定範囲 -134,217,728～+134,217,727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。（符号拡張）

8. 18 RCMP2 コンパレータ2比較レジスタ

コンパレータ2用比較データを設定します。

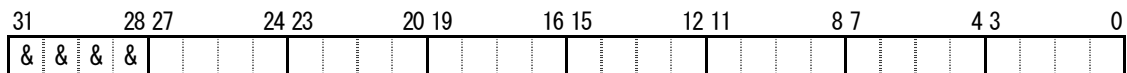


図8. 18-1 RCMP2 コンパレータ2比較レジスタ

■設定範囲 -134,217,728～+134,217,727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。（符号拡張）

8. 19 RCMP 3 コンパレータ 3 比較レジスタ

コンパレータ 3 用比較データを設定します。

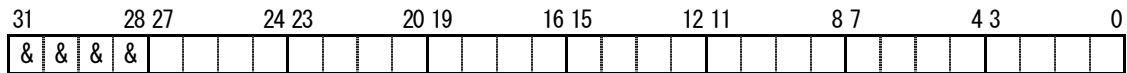


図 8. 19-3 RCMP 3 コンパレータ 3 比較レジスタ

■設定範囲 -134, 217, 728 ~ +134, 217, 727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。
(符号拡張)

8. 20 RCMP 4 コンパレータ 4 比較レジスタ

コンパレータ 4 用比較データを設定します。

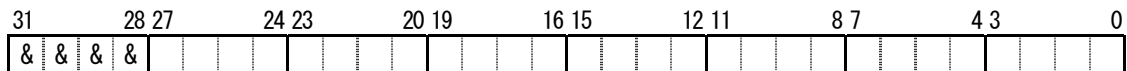


図 8. 20-1 RCMP 4 コンパレータ 4 比較レジスタ

■設定範囲 -134, 217, 728 ~ +134, 217, 727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。
(符号拡張)

8. 21 RCMP 5 (PCMP 5) コンパレータ 5 比較レジスタ

プリレジスタがあります。コンパレータ 5 用比較データをプリレジスタに設定します。

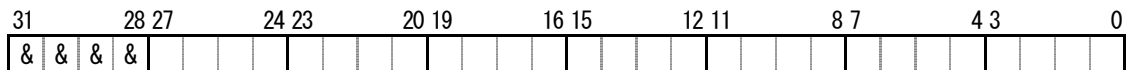


図 8. 21-1 RCMP 5 コンパレータ 5 比較レジスタ

■設定範囲 -134, 217, 728 ~ +134, 217, 727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。
(符号拡張)

8. 22 RLTC1~4 カウンタ ラッチレジスタ (読出し専用)

以下の方法によりラッチ指示されるとCTR1~CTR4の内容を一斉に読出して、データをRLTC1~RLTC4にラッチします。

CTRが動作中でも、ラッチレジスタから読出した各CTRの内容は正しく読出せます。

カウンタラッチ指示方法は「7.5 カウンタ制御機能とコマンド」参照して下さい。

8. 22. 1 RLTC1, 2, 4のレジスタ形式 (読出し専用)

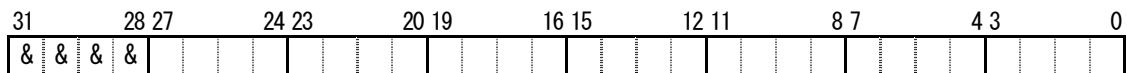


図 8. 22-1 RLTC1, 2, 4のカウンタ ラッチレジスタ

■設定範囲 -134, 217, 728 ~ +134, 217, 727

■&表記のビットは、書込み時には無視され、読出し時には空欄表示の最上位ビットと同一になります。
(符号拡張)

8. 2.2. 2 R L T C 3 (脱調カウンタ, 現在速度用) ラッチ レジスタ形式 (読出し専用)

R L T C 3は脱調カウンタ (CTR3) または現在速度カウンタの内容をラッチします。脱調カウンタか現在速度カウンタの選択は RENV5 b14 で設定します。ラッチ指示方法は「7. 5 カウンタ制御機能とコマンド」参照して下さい。

31	28 27	24 23	20 19	16 15	12 11	8 7	4 3	0
\$	\$	\$	\$	\$	\$	\$	\$	\$

図8. 2 2-2 R L T C 3 のカウンタ ラッチレジスタ

■設定範囲 脱調カウンタの場合 -32,768~+32,767

速度カウンタの場合 0~65,535

■\$表記のビットは, RENV5 b14 = '0' (脱調カウンタ選択) の時はビット15と同一 (符号拡張) となり, b14 = '1' (速度カウンタ選択) の時は 0 となる。

8. 2.3 R P L S 移動残パルスレジスタ (読出し専用)

位置決めカウンタの移動残量を確認出来ます。読出し専用です。スタート時にRMVレジスタの絶対値となり、パルス出力ごとにダウンカウントされます。途中停止時残パルスが読めます。

31	28 27	24 23	20 19	16 15	12 11	8 7	4 3	0
0	0	0	0					

図8. 2 3-1 R P L S 移動残パルスレジスタ

8. 2.4 R S P D 現在速度, その他読出しレジスタ (読出し専用)

現在速度の他にエンコーダZ相カウント値, アイドリングカウント値が読めます。読出し専用です。

速度は読んだ値に速度倍率を掛けた結果が指令速度となります。

なお, ある時点での速度を読むにはカウンタラッチコマンドを発行後R L T C 3を読みます。

(RENV5 b14 = '1')

31	28 27	24 23	20 19	16 15	12 11	8 7	4 3	0
0	0	0	0	0	0	0	0	0
IDL CT				ZCT		CRNT VELO		

図8. 2 4-1 R S P D 現在速度レジスタ

bit	名称	説 明
b0~b15	CRNT VELO	この値に速度倍率を掛けた値が指令速度。停止時は '0'
b16~b19	ZCT	原点復帰に使用されているZ相カウント値
b20~b22	IDL CT	アイドリングカウント値を読出す
b23~b31	未定義	常に '0'

表8. 2 4-1 R S P D 現在速度レジスタ内容

8. 2.5 R I P S 補間ステータスレジスタ (読出し専用)

解説は「6. 7 補間ステータスレジスタ R I P S」を参照して下さい。

9. 環境設定レジスタ

9. 1 RENV1 レジスタ

■指令出力、軸インターフェイス信号、サポインターフェイス信号など入出力極性、パルス幅、動作などを設定します。初期化時設定。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERCTL	EPW2	EPW1	EPW0	ERCTO	ERCTE	ALML	ALMM	OLL	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	INTM	DTMF	DRF	FLTR	DR	PCS	LTC	INPS	CLRO	CSTP	CSTA	1	1	

図9. 1-1 環境レジスタ RENV1

ビット	名 称	説 明				リセット初期値				
b0-b2	PMD0 ~ PMD2	信号端子名	b 2-b 0 = 1 0 0 : 個別パルス方式		指令出力	' 100' 個別パルス 方式				
					CWパルス出力					
					CCWパルス出力					
		信号端子名	b 2-b 0 = 0 0 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 '				
b3	ELM	E L S 検出時の停止方法			即停止	減速停止	' 0 '			
b4	SDM	D L S 検出時の動作方法			減速後 定速移動継続	減速停止	' 0 '			
b5	SDLT	D L S センサのラッチ機能			使用しない	使用する	' 0 '			
b6	SDL	D L S センサの入力極性			B 接	A 接	' 0 '			
b7	OLL	O L S センサの入力極性			B 接	A 接	' 0 '			
b8	SVALMM	S V A L M 入力時停止方法			即停止	減速停止	' 0 '			
b9	SVALML	S V A L M 信号入力極性			B 接	A 接	' 0 '			
b10	ERCTE	S V A L M 入力で“即停止”時のサボ 偏差からの信号 (xSVCTRCL) 自動出力			しない	する	' 0 '			
b11	ERCTO	原点復帰完了時サボ 偏差からの信号出力			しない	する	' 0 '			
b12-b14	EPW0-EPW2	サボ 偏差からの信号 (xSVCTRCL) 出力幅 (b 1 4 - 1 2 = ' 1 0 0 ' を推奨)					' 100' (13ms)			
		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0		1 0 1	1 1 0	1 1 1
		1 2 us	0. 1 ms	0. 4 ms	1. 6 ms	1 3 ms		5 2 ms	1 0 4 ms	レベル
b15	SVCTRCL	サボ 偏差からの信号の出力論理 (' 0 ' = 負論理を推奨)					' 0 '			
b16-b17	ETW0-ETW1	サボ 偏差からの信号出力後のイレ時間 (b 1 7 - 1 6 = ' 1 1 ' を推奨)			0 0	0 1	1 0	1 1	' 11'	
					0	1 2 us	1. 6 ms	1 0 4 ms		
b18	CSTA	S T A 入力仕様			' 0 ' = レベルトリガ		' 1 ' = エッジトリガ		' 0 '	
b19	CSTP	S T P 入力時停止方法 (※)			' 0 ' = 即停止		' 1 ' = 減速停止		' 0 '	
b20-b21	CLR	C L R 入力仕様			0 0	0 1	1 0	1 1	' 00'	
					立下がり	立上がり	L レベル	H レベル		
b22	INPOS	インポジション信号入力極性			' 0 ' = B 接		' 1 ' = A 接		' 1 '	
b23	LTC	ラッチ入力信号極性			' 0 ' = B 接		' 1 ' = A 接		' 0 '	
b24	PCS	P C S 入力信号極性			' 0 ' = B 接		' 1 ' = A 接		' 0 '	
b25	DR	D R 入力信号極性			' 0 ' = B 接		' 1 ' = A 接		' 0 '	
b26-b27	不使用	b 2 7 - 2 6 = ' 0 0 ' 固定							' 00'	
b28	DTMF	方向変化タイマON/OFF			' 0 ' = ON		' 1 ' = OFF		' 0 '	
b29	INTM	この軸からの割込信号をマスク (割込禁止) (ボードの最終段階割込マスクは別途あり)			' 0 ' = 割込信号出力		' 1 ' = 割込禁止		' 0 '	
b30	不使用	常に ' 0 ' を設定してください							' 0 '	
b31		常に ' 0 ' を設定してください							' 0 '	

※. 次動作使用時に減速停止に設定した場合は、STPを外部から入力する場合は停止するまで入力ON状態を保持してください。同時ストップコマンド使用時はプリレジキャンセルコマンドを発行してから同時ストップコマンドを発行してください。

表9. 1-1 環境レジスタ RENV1の内容

9. 2 RENV2 レジスタ

- エンコーダ A, B 相倍率, カウント方向, Z 相極性, パルサ A, B 相倍率, カウント方向を設定します。
■ 初期化時に設定します。

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

図9. 2-1 環境レジスタ RENV2

bit	名称	説明						初期値
b0 f b19	モード仕様 (固定)	汎用入出力ポートの仕様設定。b19～b0 は必ず次のように初期設定する。 CPD364 .. 0x0fd65 CPD578, CPD278, CPD5212M .. 0x0fd55 上記以外 .. 0x0f555						固定
b20 b21	ENCM 倍率	エンコーダ通倍設定	b21, b20 通倍	00 x1	01 x2	10 x4	11 up/down パルス	10 (x4)
b22	EDIR	'1' = エンコーダ A 相, B 相カウンタ方向逆転						0
b23	EZL	エンコーダ Z 相入力極性反転 '1' = 反転						0
b24 b25	PULSR 倍率	パルス通倍設定	b25, b24 通倍	00 x1	01 x2	10 x4	11 up/down パルス	00 (x1)
b26	PDIR	'1' = パルス A 相, B 相カウンタ方向逆転						0
b27	不使用	常に '0' を設定してください						
31～28	—	「11. CPD シリーズ追加機能について」参照						'0000'

表9. 2-1 環境レジスタ RENV2 の内容

9. 3 RENV3 レジスタ

- 原点復帰方法指定、原点復帰Z相カウント設定、CTR2～4の入力形式設定、CTR1～4の出力動作方法の設定などを行います。
- 初期化時に設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0		CTR4G		CTR4IP		CTR3IP		CTR2IP		Z相CT				ORmode		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CTR4DIS	CTR3DIS	CTR2DIS	0	CTR4BS	CTR3BS	CTR2BS	CTR1BS	CTR4ORG	CTR3ORG	CTR2ORG	CTR1ORG	CTR4CLR	CTR3CLR	CTR2CLR	CTR1CLR	

図9. 3-1 環境レジスタ RENV3

bit	名 称	説 明			ライブリ初期値	
b0 └ b3	ORGmode	復帰 No	b3-b0	【復帰方法名称】	No. 2 '0010'	
		0	0000	【セグ原点復帰 (OLS 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	【セグ原点復帰 (OLS 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	【セグ+エンコーダ 原点復帰 (OLS 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	【セグ+エンコーダ 原点復帰 (OLS 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	【セグ反転+エンコーダ 原点 (OLS 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	【セグ高速反転+エンコーダ 原点 (OLS 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	【ELS 兼用セグ原点】 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	【ELS 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	【ELS 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 CTR2 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			「表 4. 2-5 原点復帰方法選択表」参照
b4 └ b7	Z 相 C T	原点復帰時のエンコーダ Z 相カウンタ値を設定する 設定値範囲: 0000 (1 回目 Z 相カウンタで原点) ~ 1111 (16 回目)			0	
b8 b9	CTR2IP	CTR2 (機械位置カウンタ) カウント入力ソースの選択	b9, b8 =	0 0 ENC (パルサ) 入力 0 1 指令パルス入力 1 0 ENC (パルサ) 入力	ENC 入力とパルサ入力はコネクタ端子で共通です (択一)	
b10 b11	CTR3IP	CTR3 (偏差カウンタ) カウント入力ソースの選択	b11, b10 =	0 0 指令パルスと ENC (パルス) 入力の差分 0 1 指令パルスと ENC (パルス) 入力の差分 1 0 不使用		
b12 b13	CTR4IP	CTR4 (汎用カウンタ) カウント入力ソースの選択	b13, b12 =	0 0 指令パルス入力 0 1 ENC (パルサ) 入力 1 0 ENC (パルサ) 入力 1 1 9.8304MHz CLK カウント (最大 27s)		
b14	CTR4G	'1' = 動作中の期間のみ CTR4 カウント可とする。 (MSTS の bit1 SRUN = 1 の間)			0	
b15	未定義	常に '0' を設定してください			0	
b16	CTR1CLR	CLR 入力時	CTR1	を '1' = クリアする, '0' = クリアしない	0	
b17	CTR2CLR		CTR2		0	
b18	CTR3CLR		CTR3		0	
b19	CTR4CLR		CTR4		0	
b20	CTR1ORG	原点復帰完了時	CTR1	を '1' = クリアする, '0' = クリアしない	1	
b21	CTR2ORG		CTR2		1	
b22	CTR3ORG		CTR3		1	
b23	CTR4ORG		CTR4		1	
b24	CTR1BS	バックラッシュ あるいは スリップ補正時も	CTR1	を '1' = 動作させる, '0' = 動作させない	0	
b25	CTR2BS		CTR2		0	
b26	CTR3BS		CTR3		0	
b27	CTR4BS		CTR4		0	
b28	未定義	常に '0' を設定してください				
b29	CTR2DIS	CTR2 CTR3 CTR4	を '1' = カウント不可にする, '0' = カウント可にする	指令位置カウンタ CTR1 は RMD レジスタ CTR1DIS (bit11) で行う	0	
b30	CTR3DIS				0	
b31	CTR4DIS				0	

表 9. 3-1 環境レジスタ RENV3 の内容

9. 4 RENV4 レジスタ

コンパレータ (CMP) は5組あります。CMP 1~4をRENV4で設定します。(CMP 5はRENV5)

■CMP1~4の比較カウンタ選択, 比較方法, 条件成立時の動作方法を指定します。初期化時に設定。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CMP2M	CMP2D	CMP2C	0	CMP1M	CMP1D	CMP1C								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP4M	CMP4D	CMP4C	0	CMP3M	CMP3D	CMP3C									

図9. 4-1 環境レジスタ RENV4

(注1) 比較対象カウンタとしてCTR3(脱調用)を選択した場合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る。
(絶対値範囲: 0~32,767)

ビット	名称	説 明						ラブリ初期値
b0-b1	CMP1C	CMP 1 比較CTR	b 1-0=0 0 CTR 1(指令位置)	b 1-0=0 1 CTR 2(機械位置)	b 1-0=1 0 CTR 3(偏差)	b 1-0=1 1 CTR 4(汎用)	'00' (指令位置)	
b2-b4	CMP1D	CMP 1 比較方法	b 4-2=0 0 0	b 4-2=0 0 1	b 4-2=0 1 0	b 4-2=0 1 1	'000' (+SLS 不使用)	
			比較せず	RCMP 1=比較CTR				
				カウント方向無関係	カウントアップ中	カウントダウン中		
			b 4-2=1 0 0	b 4-2=1 0 1	b 4-2=1 1 0	b 4-2=1 1 1		
			RCMP 1>比較CTR	RCMP 1<比較CTR	-SLSとして使用 (RCMP 1<CTR 1)	比較せず		
b5-b6	CMP1M	CMP 1条件 成立動作処理	b 6-5=0 0 処理無し	= 0 1 即停止	= 1 0 減速停止	= 1 1 動作データをプリレジスタのデータに変更(速度変更)	'00' 処理無し	
b7	未定義	常に '0' を設定してください						'0'
b8-b9	CMP2C	CMP 2 比較CTR	b 9,b 8=0 0 CTR 1(指令位置)	b 9,b 8=0 1 CTR 2(機械位置)	b 9,b 8=1 0 CTR 3(偏差)	b 9,b 8=1 1 CTR 4(汎用)	'00' (指令位置)	
b10-b12	CMP2D	CMP 2 比較方法	b 1 2-1 0=0 0 0	b 1 2-1 0=0 0 1	b 1 2-1 0=0 1 0	b 1 2-1 0=0 1 1	'000' (-SLS 不使用)	
			比較せず	RCMP 2=比較CTR				
				カウント方向無関係	カウントアップ中	カウントダウン中		
			b 1 2-1 0=1 0 0	b 1 2-1 0=1 0 1	b 1 2-1 0=1 1 0	b 1 2-1 0=1 1 1		
			RCMP 2>比較CTR	RCMP 2<比較CTR	-SLSとして使用 (RCMP 2>CTR 1)	比較せず		
b13-b14	CMP2M	CMP 2条件 成立動作処理	b 1 4-1 3=0 0 処理無し	= 0 1 即停止	= 1 0 減速停止	= 1 1 動作データをプリレジスタのデータに変更(速度変更)	'00' 処理無し	
b15	未定義	常に '0' を設定してください						'0'
b16-b17	CMP3C	CMP 3 比較CTR	b 1 7-1 6=0 0 CTR 1(指令位置)	b 1 7-1 6=0 1 CTR 2(機械位置)	b 1 7-1 6=1 0 CTR 3(偏差)	b 1 7-1 6=1 1 CTR 4(汎用)	'00' (指令位置)	
b18-b20	CMP3D	CMP 3 比較方法	b 2 0-1 8=0 0 0	b 2 0-1 8=0 0 1	b 2 0-1 8=0 1 0	b 2 0-1 8=0 1 1	'000'	
			比較せず	RCMP 3=比較CTR				
				カウント方向無関係	カウントアップ中	カウントダウン中		
			b 2 0-1 8=1 0 0	b 2 0-1 8=1 0 1	b 2 0-1 8=1 1 0	b 2 0-1 8=1 1 1		
			RCMP 3>比較CTR	RCMP 3<比較CTR	設定禁止	比較せず		
b21-b22	CMP3M	CMP 3条件 成立動作処理	b 2 2-2 1=0 0 処理無し	= 0 1 即停止	= 1 0 減速停止	= 1 1 動作データをプリレジスタのデータに変更(速度変更)	'00' 処理無し	
b23	未定義	常に '0' を設定してください						'0'
b24-b25	CMP4C	CMP 4 比較CTR	b 2 5-2 4=0 0 CTR 1(指令位置)	b 2 5-2 4=0 1 CTR 2(機械位置)	b 2 5-2 4=1 0 CTR 3(偏差)	b 2 5-2 4=1 1 CTR 4(汎用)	'00' (指令位置)	
b26-b29	CMP4D	CMP 4 比較方法	b 2 9-2 6=0 0 0 0	b 2 9-2 6=0 0 0 1	b 2 9-2 6=0 0 1 0	b 2 9-2 6=0 0 1 1	'0000'	
			比較せず	RCMP 4=比較CTR				
				カウント方向無関係	カウントアップ中	カウントダウン中		
			b 2 9-2 6=0 1 0 0	b 2 9-2 6=0 1 0 1	b 2 9-2 6=0 1 1 0	b 2 9-2 6=0 1 1 1		
			RCMP 3>比較CTR	RCMP 3<比較CTR	比較せず	比較せず		
			b 2 9-2 6=1 0 0 0	b 2 9-2 6=1 0 0 1	b 2 9-2 6=1 0 1 0	b 2 9-2 6=1 0 1 1		
			同期 (定ピッチ) 信号出力				比較せず	
			カウント方向無関係	カウントアップ中	カウントダウン中			
b30-b31	CMP4M	CMP 4条件 成立動作処理	b 3 1-3 0=0 0 処理無し	= 0 1 即停止	= 1 0 減速停止	= 1 1 動作データをプリレジスタのデータに変更(速度変更)	'00' 処理無し	

(注2) +SLS, -SLSに設定する場合は比較対象カウンタに CTR1(指令位置)を選択する。

表9. 4-1 環境レジスタ RENV4 の内容

9. 5 RENV5 レジスタ

■CMP5の比較方法選択, 比較方法, 条件成立時の動作方法, その他, アイドリングパルス数, カウンタの特殊なラッチ条件, 他軸スタート条件などの設定を行います。

■他軸スタート関係を除き, 初期化時に設定します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CTLCHF	CTLCH	0	IDLP	CMP5M	CMP5D	CMP5C								
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	TRIGRIN sel	TRIGROT sel				

図9. 5-1 環境レジスタ RENV5

bit	名称	説明	ラブリ初期値
b0 └ b2	CMP5C	CMP5の比較対象 CTRを選択 b2-b0= 000 CTR1 (指令位置) 011 CTR4 (汎用) 001 CTR2 (機械位置) 100 位置管理カウンタ (PCTR) 010 CTR3 (偏差) 101 現在速度レジスタ	'000'
b3 └ b5	CMP5D	CMP5の比較方法の選択 (RCMP5は比較データレジスタ) b5-b3= 000 比較せず 100 RCMP5>比較カウンタ 001 RCMP5= 方向に無関係 101 RCMP5<比較カウンタ 010 比較カウンタ カウント up中比較 110 比較せず 011 down中比較 111	'000'
b6 └ b7	CMP5M	CMP5条件成立時の動作処理指定 b7, b6= 00 01 10 11 処理なし 即停止 減速停止 動作データをプルパルスのデータに変更 (速度変更)	'00'
b8 └ b10	IDLP	パルスモータ用機能として, アイドリングパルス数(0~7パルス)を設定 (特別な用途として「速度制御」にも使用可能⇒「10. 2. 2 急加減速での速度制御」)	'000'
b11	未定義	常に「0」を設定してください	0
b12 └ b13	CTLCH	CTR1~4を一斉に次の条件でラッチする b13, b12= 00 01 10 11 不使用 OLS信号on時 CMP4比較条件成立時 CMP5比較条件成立時	'00'
b14	CTLCHF	「1」=上記機能(b12, b13)においてCTR3 (偏差) の代わりに「現在速度レジスタ」をラッチ	0
b15	不使用	常に「0」を設定してください	0
b16 └ b19	TRIGROT sel	他軸をトリガ (スタート) させる条件設定 (⇒10. 6. 4 条件付き他軸スタート) b19-b16= 0000 トリガoff 1000 加速開始時 0001 CMP1 1001 加速終了時 0010 CMP2 1010 減速開始時 0011 CMP3 条件成立時 1011 減速終了時 0100 CMP4 0101 CMP5 その他 トリガoff	'0000'
b20 └ b21	TRIGRIN sel (※1)	この軸が(スタート)を受けるトリガ元の選択 b21, b20 = 00 = 01 = 10 = 11 X (V, X1, X2, X3) Y (W, Y1, Y2, Y3) Z (A, Y1, Y2, Y3) U (B, U1, U2, U3) 軸トリガ 軸トリガ 軸トリガ 軸トリガ	'00'
b23-b22	未定義	常に「0」を設定してください	'00'
b27-b24	—	「11. CPDシリーズ追加機能について」参照	'0000'
b31-b28	未定義	常に「0」を設定してください	'0000'

表9. 5-1 環境レジスタ RENV5の内容

※1. 2軸ボードはZ, U軸はありません。1個のPCL内の軸の組み合わせで有効です。

9. 6 RENV6 レジスタ

■バックラッシュ補正, スリップ補正などの選択, データを設定します.

■初期化時に設定します.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	BLSH sel		バックラッシュ補正量/スリップ補正量 (BLSH)											
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9. 6-1 環境レジスタ RENV6

bit	名称	説明	ライブラリ初期値
b0 └ b11	BLSH	b0~b11の12ビットで バックラッシュ補正量 または スリップ補正量 を設定する	all '0'
b12 b13	BLSH sel	補正方法の選択 b13, b12: 00=補正機能 off 01=バックラッシュ補正 10=スリップ補正	'00' off
b14-b31	不使用	常に '0' を設定してください	all '0'

表9. 6-1 環境レジスタ RENV6の内容

9. 7 RENV7 レジスタ

■パルスモータの停止時特性を改善する「制振機能」を設定します.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
逆転パルス挿入位置時間 (RT)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
正転パルス挿入位置時間 (FT)															

図9. 7-1 環境レジスタ RENV7

ビット	名称	説明	ライブラリ初期値
b0-b15	RT 逆転時間	パルスモータへの最終出力パルス後の逆転パルス出力タイミング RT: 0, 1~65535 (16us~104ms)	共に '0' (制振機能 off)
b16-b31	FT 正転時間	上記逆転パルス出力後の正転パルス出力タイミング FT: 0, 1~65535 (16us~104ms)	

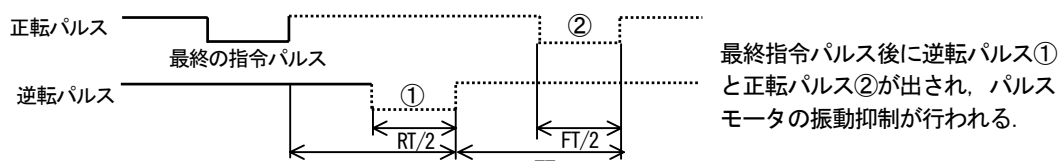


表9. 7-1 環境レジスタ RENV7の内容

10. 各機能と応用

10.1 速度と加減速

10.1.1 速度、加減速の計算

速度パターンとレジスタの対応は 図4. 2-1 を参照して下さい。

- (1) RFL : FL速度設定レジスタ (16ビット)

FL定速動作の速度及び、高速動作(加減速動作)の場合のスタート速度を1~65,535(0FFFFh)の範囲で設定します。

速度はRMG(速度倍率)の設定値との計算値になります。

$$\text{FL速度[PPS]} = \frac{\text{RFL} \times 300}{\text{RMG} + 1}$$

- (2) RFH : FH速度設定レジスタ (16ビット)

FH定速動作の速度及び、高速動作(加減速動作)の場合のスタート速度を1~65,535(0FFFFh)の範囲で設定します。

高速動作(加減速動作)の場合にはRFLの設定値より大きな値を設定してください。

速度はRMGの設定値との計算値になります。

$$\text{FH速度[PPS]} = \frac{\text{RFH} \times 300}{\text{RMG} + 1}$$

- (3) RUR : 加速レート設定レジスタ (16ビット)

高速動作(加減速動作)の場合の加速特性を1~65,535(0FFFFh)の範囲で設定します。

設定値と加速時間の関係は次式の様になります。

直線加速 (RMDレジスタの 'b10 acc 方法' = 0)

$$\text{加速時間[sec]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 4}{19,660,800}$$

直線部分のないS字加速 (RMDレジスタの 'b10 acc 方法' = 1 かつ RUSレジスタ > 0)

$$\text{加速時間[sec]} = \frac{(\text{RFH} - \text{PRFL}) \times (\text{RUR} + 1) \times 8}{19,660,800}$$

直線部分のあるS字加速 (RMDレジスタの 'b10 acc 方法' = 1 かつ RUSレジスタ > 0)

$$\text{加速時間[sec]} = \frac{(\text{RFH} - \text{RFL} + 2 \times \text{RUS}) \times (\text{RUR} + 1) \times 4}{19,660,800}$$

- (4) RDR : 減速レート設定レジスタ (16ビット)

高速動作(加減速動作)の場合の減速特性を1~65,535(0FFFFh)の範囲で設定します。

減速開始点を自動計算 (RMDレジスタの 'b13: 減速開始点' = 0) に設定する場合でも、RDRレジスタの設定が減速レートとして使用されます。

ただし、RDR=0に設定した場合、減速レートはRURで設定した値になります。

減速開始点を自動計算に設定した場合は、独立動作では(減速時間) ≤ (加速時間 × 2)、補間動作では(減速時間) = (加速時間)となるように設定してください。

注意: 次のような場合は、減速開始点は手動計算して設定します。

独立動作で(減速時間) > (加速時間 × 2)、補間動作で(減速時間) > (加速時間)

この場合は、停止時にFL速度まで減速しきれない場合がありますので、減速開始点手動計算

(RMDレジスタの 'b13: 減速開始点' = 1) に設定し、減速開始点(RDP)を設定してください。

詳しくは次ページ「(6) RDP: 減速開始点設定レジスタ (24ビット)」を参照してください。

■減速開始点自動設定で

(減速時間) ≤ (加速時間 × 2) の時

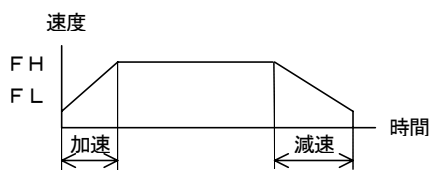


図10.1-1 正しい条件の自動加減速

■減速開始点自動設定で

(減速時間) > (加速時間 × 2) の時

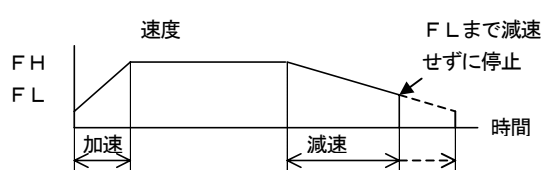


図10.1-2 不適正な加減速の場合

設定値と減速時間の関係は次式の様になります。

■直線減速（RMDレジスタの‘b10 acc 方法’ = 0）

$$\text{減速時間[sec]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RDR} + 1) \times 4}{19,660,800}$$

■直線部分のないS字加速（RMDレジスタの‘b10 acc 方法’ = 1 かつ RDSレジスタ > 0）

$$\text{減速時間[sec]} = \frac{(\text{RFH} - \text{PRFL}) \times (\text{RDR} + 1) \times 8}{19,660,800}$$

■直線部分のあるS字加速（RMDレジスタの‘b10 acc 方法’ = 1 かつ RUSレジスタ > 0）

$$\text{減速時間[sec]} = \frac{(\text{RFH} - \text{RFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1) \times 4}{19,660,800}$$

(5) RMG：速度倍率設定レジスタ（12ビット）

RFL、RFH設定値と速度の関係を2～4,095（0FFFh）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ので、通常はできるだけ小さな倍率を使用して下さい。

設定値と速度倍率の関係は次式の様になります。

$$\text{速度倍率[倍]} = \frac{300}{\text{RMG} + 1}$$

(6) RDP：減速開始点設定レジスタ（24ビット）

加減速・位置決め動作の場合の減速開始点（終点からのパルス数で指定。残移動量が減速開始点以下になると減速を開始します）を設定します。RMDレジスタの「減速開始点の設定方法（b13 減速開始点）」の状態により、RDPに設定する値の意味は異なります。

(I) マニュアル設定時（RMDレジスタの‘b13 減速開始点=1’）の場合

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

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b10 acc 方法 = 0）

$$\text{最適値[ハルス]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 32,768}$$

■直線部分のないS字減速（RMDレジスタのb10 acc 方法 = 1 かつ RDSレジスタ = 0）

$$\text{最適値[ハルス]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1) \times 2}{(\text{RMG} + 1) \times 32,768}$$

■直線部分のあるS字減速（RMDレジスタのb10 acc 方法 = 1 かつ RUSレジスタ > 0）

$$\text{最適値[ハルス]} = \frac{(\text{RFH} + \text{RFL}) \times (\text{RFH} - \text{RFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 32,768}$$

（位置決めカウンタ値）≤（RDP設定値）のタイミングで減速を開始します。

(II) 自動設定時（RMDレジスタの‘b10 acc 方法’ = 0）の場合

自動で設定される減速開始点に対してオフセットとなり、

–8,388,608（800000h）～ 8,388,607（7FFFFh）の範囲で設定します。

オフセット量が正数の場合、早めに減速を開始して、減速後FL速度で動作します。

負数の場合、減速開始が遅れます。オフセットが必要ない場合は‘0’に設定します。

減速開始点の設定値より小さい場合、停止時の速度はFLより早くなります。

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。

(7) RUS：加速S字範囲設定レジスタ（15ビット）

S字加減速動作のS字加速区間を1～32,767（7FFFh）の範囲で設定します。

S字加速区間の範囲SsuはRMGの設定値との計算値になります。

$$\text{Ssu [PPS]} = \frac{\text{RUS} \times 300}{\text{RMG} + 1}$$

つまり、FL速度～（FL速度+Ssu）までと（FH速度–Ssu）～FH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 $\frac{\text{RFH} - \text{RFL}}{2}$ 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。

10. 1. 2 急加減速時の注意点

加減速指令時の初速及び終速は、理論的にFLよりやや高い値（通常は僅かに高い値）になります。特に、加減速勾配が急な場合には、その差異が大きくなるので注意が必要な場合があります。

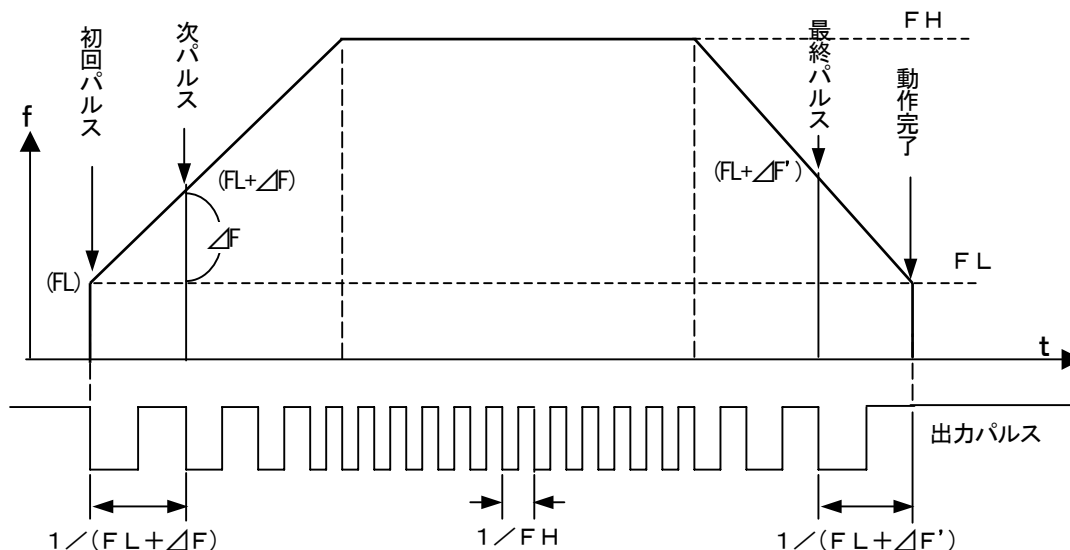


図10. 1-3 加減速動作指令と実際の速度制御

- 動作開始直後、理論上の速度FLでパルス出力し、次パルス出力時の理論上の速度はFL + ΔFになっています。
したがって、最初の2パルスの間隔は1/FLではなく、1/FL + ΔFになります。
- 同様に、動作完了時の理論上の速度はFLなので、最終パルス出力時の理論上の速度はFL + ΔF'になります。
- また、減速開始点自動計算時、FLが極端に小さい時に動作時間が延びてしまうのを防ぐため、減速開始点の自

動

演算結果が整数でない場合、減速開始点は減速開始が遅くなる方向に丸められます。

したがって、減速開始点自動設定時ΔF'は、ΔFよりも大きくなります。

特に“移動量が少なく、速度がFHまで到達しない場合”ΔF'はΔFよりもかなり大きくなります。

- 通常運転時、ΔF、ΔF'は僅かですので、運用上 初速=FL、終速=FLと考えて差し支えないですが、加減速勾配が極端に大きい場合や、移動量が少ない場合はΔF、ΔF'（特にΔF'）は大きくなるので、この動作が機械振動抑制等の観点から問題となる場合は、以下の対策を行うことにより、動作終了時の速度がFLとなるような補正が行えます。
（特に問題にならないければ補正の必要はありません）

加減速動作において、機械振動抑制等の為に、「初速 = 終速 = FL」とする方法については、「テクニカルノート」をご請求ください。

10.2 速度のオーバーライド

動作中に速度を変更することが出来ます。ただし、次の表の条件によります。

自動加減速モード(RMD b13= '0')における速度オーバーライド(速度途中変更)		
定速送り		
◎全ての動作		
加減速制御送り	S字加減速時速度変更 	直線加減速時速度変更
◎連続送り ◎直線補間連続送り ◎位置決め ◎直線補間 (直線加減速)	①加速中に目標動作速度より低いRFHを指示する。RFH<変更時点の速度なら、RFHの速度までS字減速する。 ②加速中に目標動作速度より低いRFHを指示する。RFH≥変更時点の速度なら、S字せずRFH速度になる。 ③加速中に目標の動作速度より大きいRFHを与えた場合は目標速度到達後、RFH速度までS字加速する。 ④、⑤加減速終了後にRFHを変更すると、その速度までS字で加減速する。	①加速中に目標速度より低いRFHを指示するとその速度まで加減速する。 ②、③加減速終了後にRFHを変更すると、その速度まで加減速する。

表 10. 2-1 自動加減速における速度オーバーライド条件

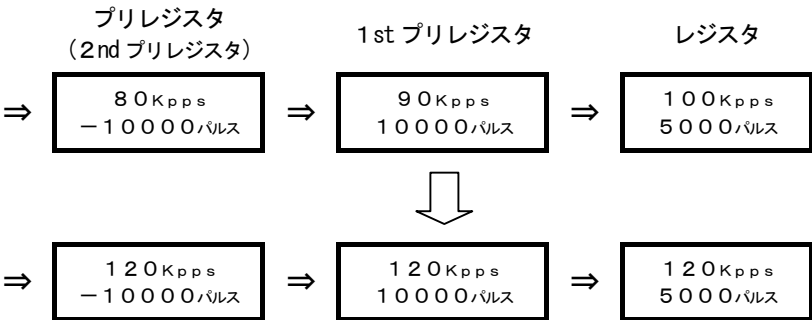
■自動加減速モードでの速度オーバーライド

自動加減速モード(RMD b13= '0')では、RFL, RUR, RUS, RDSを変更してはなりません。速度変更は直接動作速度レジスタRFHを書き換えます。プリレジスタを連続して使用する「次動作連続実行」の場合は、プリレジスタのFH速度も変更する必要があります。

■速度オーバーライドの例

次の例は、X軸が現在の動作速度が100Kppsで実行されていて、次動作連続実行として、1st プリレジスタに速度90Kpps,位置決め量10000パルスがセットされ、プリレジスタに速度80Kpps,位置決め量10000パルスがセットされている。

このとき、現在実行の速度から全て120Kppsにする。速度倍率は全てx10で運用する。



【 CPD534 ドライバ関数による記述例 】

●速度のオーバーライドの運用例

RMD,RFL,RUR,RMG 等はすでに設定してあるものとします.

```
// スタート
case IDC_BTN_STRT:
    cp230_wReg (hDevID, 0, 0x82, 10000); // 動作速度設定
    cp230_wReg (hDevID, 0, 0x80, 5000); // 移動量設定
    cp230_wCmdW (hDevID, 0, 0x53); // 加速スタート
    cp230_wReg (hDevID, 0, 0x82, 9000); // 動作速度設定
    cp230_wReg (hDevID, 0, 0x80, 10000); // 移動量設定
    cp230_wCmdW (hDevID, 0, 0x53); // 加速スタート
    cp230_wReg (hDevID, 0, 0x82, 8000); // 動作速度設定
    cp230_wReg (hDevID, 0, 0x80, -10000); // 移動量設定
    cp230_wCmdW (hDevID, 0, 0x53); // 加速スタート
    break;

// 速度変更
case IDC_BTN_CHNG:
    cp230_wReg (hDevID, 0, 0x92, 12000); // 動作速度を RFH に書込む
    cp230_wCmdW (hDevID, 0, 0x26); // プリレジスタキャンセル
    cp230_wReg (hDevID, 0, 0x82, 12000); // 動作速度を PRFH に書込む
    cp230_wCmdW (hDevID, 0, 0x53); // 加速スタート
    cp230_wCmdW (hDevID, 0, 0x53); // 加速スタート
    break;
```

10. 3 位置のオーバーライド

位置決め動作中に、自由に目標位置のオーバーライド（目標位置の書き換え）ができます。
変更はスタート位置を基準とした新しい目標位置データ（RMVレジスタ値）を書き換えます。

- (1) 加速または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すれば、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。
- (2) 減速中に新データを最初の目標値より遠方に変更すれば、その位置から F H 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。
- (3) 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。

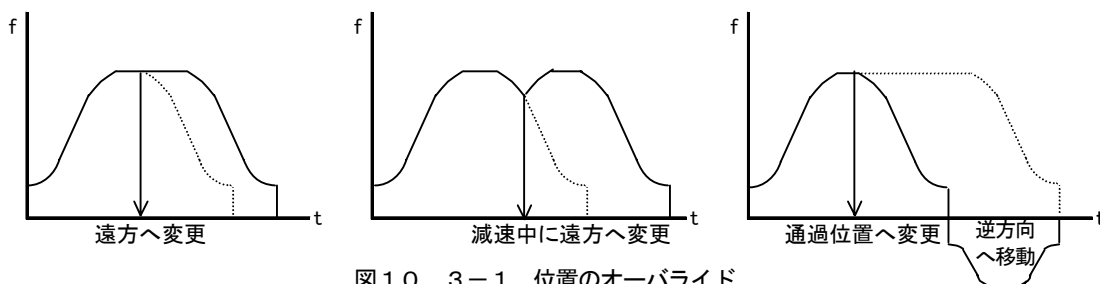


図 10. 3-1 位置のオーバーライド

- (4) 停止直前のオーバーライド指令を行う場合

- ① 次動作データが無い時、無視される可能性があります。（位置の読み出しで確認）
- ② 次動作データがある時、次動作がオーバーライドされる可能性があります。

■位置のオーバーライドの例

次の例は、位置決め動作中に移動量が 10000 から 15000 へ変更されます。

【CPD534 ドライバ関数による記述例】

- 次の例では、すでにレジスタ及びオプションポートの初期化が
終わっているものとします。

```

DWORD    ret;                //関数の戻り値

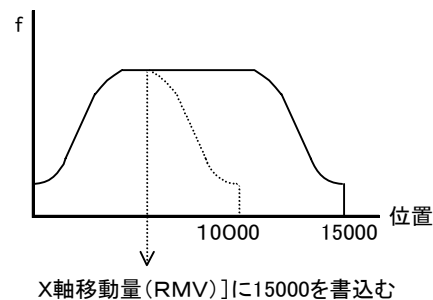
//X軸移動量設定(移動量 10000)
ret = cp530_wReg( hDevID,    //デバイスハンドル
                  0,          //X軸を指定
                  0x80,       //移動量 (PRMV) を指定
                  10000 );    //移動量 10000

//X軸動作モード設定(位置決め動作)
ret = cp530_wReg( hDevID,    //デバイスハンドル
                  0,          //X軸を指定
                  0x87,       //動作モード (PRMD) を指定
                  0x41 );    //位置決め動作

//スタートコマンド発行
ret = cp530_wCmdW( hDevID,   //デバイスハンドル
                  0,          //X軸を指定
                  0x53 );    //高速スタート

//動作途中で移動量 15000 パルスに変更、X軸移動量設定(移動量 15000)
ret = cp530_wReg( hDevID,    //デバイスハンドル
                  0,          //X軸を指定
                  0x90,       //移動量 (RMV) を指定
                  15000 );    //移動量 15000

```



10.4 PCS「位置決め管理開始信号入力」

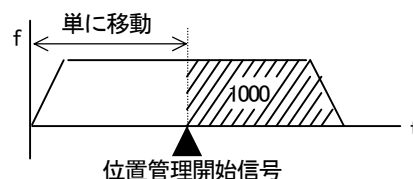
RMD（動作モード）レジスタのPCS有効(bit14) = ‘1’ に設定することにより、動作スタート後、コマンド書き込み、またはPCS入力信号が‘ON’のタイミングから、RMVレジスタに設定した移動量分の位置決め動作を行う事ができます。（PCS信号の代わりにコマンド(28h)により位置決め管理開始が出来ます。）

- 位置決め管理開始の例・・・連続送り中にPCS信号到来により位置決め開始10000に位置決め。

【CPD534 ドライバ関数による記述例】

ドライバI/F用DLL関数の応用例を記します。

- 以下の例では、すでにレジスタ及びオプションポートの
初期化が終わっているものとします。
更に、PCS入力有効、PCS極性設定がすすんでいる
ものとします。X軸のPCS信号入力の時点から位置
決め(移動量 10000)します。



```

DWORD    ret;                //関数の戻り値

ret = cp530_wReg( hDevID, 0, 0x80, 10000 ); //X軸 移動量 (PRMV) = 10000
ret = cp530_wReg( hDevID, 0, 0x87, 0x4041 ); //X軸 動作モード (PRMD) = PCS位置決め動作
ret = cp530_wCmdW( hDevID, 0, 0x53 );       //X軸 高速スタート

```

10.5 指定軸停止で起動（条件付他軸スタート—1）

指定軸停止での起動にはここで説明する「条件付他軸スタート—1」の他にコンパレータ成立で他軸起動の「条件付他軸スタート—2」があります。（10.6.4 コンパレータ条件成立で他軸起動 参照）

「条件付他軸スタート—1」による動作は同一PCL内の軸の停止により自軸がスタートします。

■設定箇所

- (1) RMD b19, b18=11 と設定する。「指定軸の停止によりスタート」の状態となります。
（通常はb19,b18=00 即スタートの設定）
- (2) RMD b23~b20において停止軸の選択をしておく。

RMD : b23, 22, 21, 20	停止軸指定	記 事
0001	X(V, X1, X2, X3) (※) 軸停止でスタート	各ビット位置の組合せで任意の軸組合せが取れます。 (同一PCL内) 自軸の起動条件に自軸を停止軸指定してはいけません。 2軸ボードはZ, U軸はありません。
0010	Y(W, Y1, Y2, Y3) (※) 軸停止でスタート	
0100	Z(A, Z1, Z2, Z3) (※) 軸停止でスタート	
1000	U(B, U1, U2, U3) (※) 軸停止でスタート	
0101	X(V, X1, X2, X3) (※) 軸と Z(A, Z1, Z2, Z3) (※) 軸停止でスタート	
1110	Y(W, Y1, Y2, Y3) (※) 軸と Z(A, Z1, Z2, Z3) (※) 軸と U(B, U1, U2, U3) (※) 軸停止でスタート	

※. 4軸ボードではX~U軸の組合せ, 8軸ボードではX~U軸またはV~B軸の組合せ,
12軸ボードではX1~U1軸, X2~U2軸, X3~U3軸の組合せで使用できます。

- (3) スタートコマンドを発行します。(ステータスMSTS b0= '1' スタート保留となる.)
即停止コマンド(49h)を発行するとスタート保留は解除されます。

●指定軸停止で起動の例・・・X軸の位置決め動作終了(移動量10000)でY軸がスタート。

【CPD534 ドライバ関数による記述例】

ドライバI/F用DLL関数の応用例を記します。

X軸停止でY軸スタート。

```
DWORD ret; //関数の戻り値
```

```
//X軸移動量設定(移動量10000)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
0, //X軸を指定
0x80, //移動量(PRMV)を指定
10000 ); //移動量10000
```

```
//X軸動作モード設定(位置決め動作)
```

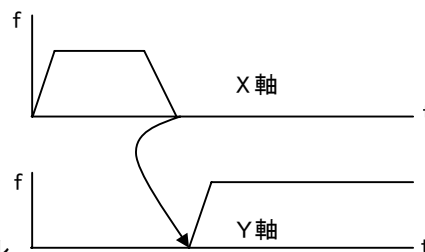
```
ret = cp530_wReg( hDevID, //デバイスハンドル
0, //X軸を指定
0x87, //動作モード(PRMD)を指定
0x41 ); //位置決め動作
```

```
//Y軸動作モード設定(X軸停止で+方向連続送りスタート)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
1, //Y軸を指定
0x87, //動作モード(PRMD)を指定
0x1c0000 ); //X軸停止で+方向連続送りスタート
```

```
//スタートコマンド発行
```

```
ret = cp530_wCmdW( hDevID, //デバイスハンドル
0, //X Y軸高速スタート
0x353 ); //X軸は即スタート, Y軸はX軸停止でスタート
```



10.6 カウンタ およびコンパレータ

CPD534の各軸にカウンタ(CTR1~4)とコンパレータ(CMP1~5)を使用することにより、次の機能が可能です。

- 指令位置、機械位置管理(CTR1,CTR2使用)
- ソフトリミット機能(CTRとCMPの使用)
- コンパレータ条件成立により他軸スタートさせる(CTRとCMPの使用)
- コンパレータ条件成立により停止または速度変更(CTRとCMPの使用)
- パルスモータ脱調検出機能(CTR3使用)

10.6.1 カウンタ

カウンタ機能、カウンタ入力選択、カウント信号形式などを表10.6-1に一括して掲げます。

	CTR1	CTR2	CTR3	CTR4	設定箇所
用 途	指令位置 カウント	機械位置 カウント	(パルス) モータ 脱調検出	汎 用	カウントテーブル CTR1 : RMD b11
機 能	UP/DOWN CTR	UP/DOWN CTR	偏差カウンタ	UP/DOWN CTR	CTR2~CTR4 : RENV3 b29~b31
ビット長	28	28	16	28	
カウント入力 の選択	指令出力	指令出力	指令出力	指令出力	カウント入力の選択: CTR2 : RENV3 b9,b8 CTR3 : RENV3 b11,b10 CTR4 : RENV3 b13,b12 カウント・CLK (注)
	—	エンコーダ	エンコーダ	エンコーダ	
	—	パルサ	パルサ	パルサ	
	—	—	—	カウント・CLK	
入力信号形式	—	位相差または UP/DOWNパルス	位相差または UP/DOWNパルス	位相差または UP/DOWNパルス	RENV2 参照

(注) カウント入力の「カウント・CLK」を選択した場合は内部CLK 9.8304MHz (約0.102μs) をカウントする。

表10.6-1 カウンタ機能と選択

ボード種別	x1時	x4時
CPD5212M, CPD278, CPD364	1.6MHz	6.5MHz
上記以外	1MHz	4MHz

表10.6-2 エンコーダカウント周波数

- 各カウンタのカウント入力ルート案内図を図10.6-1に示します。

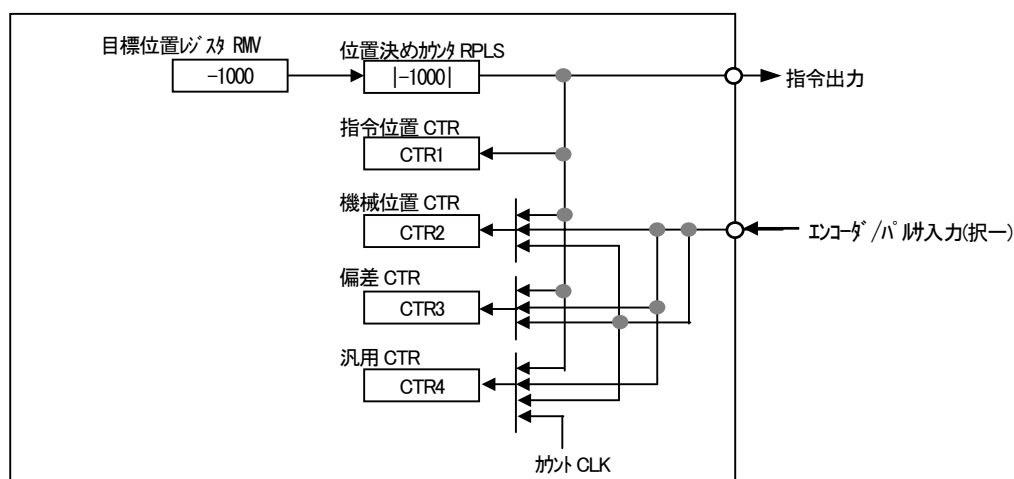


図10.6-1 各カウンタのカウント入力ルート案内図

10. 6. 2 コンパレータ

コンパレータの標準的使用を「表 10. 6-2 コンパレータの標準的使用」に示します。

用途	設定箇所	ソフトリミット				脱調検出		CMP成立で 停止または速度変更		CMP成立で 他軸起動			
使用コンパレータ		CMP1 (+SLS), CMP2 (-SLS)				CMP3		CMP4		CMP5			
比較カウンタ		CTR1				CTR3 (偏差)		CTR1/CTR2/CTR4		CTR1/CTR2/CTR4/速度			
比較カウンタ		CMP1 の比較		CMP2 の比較		CMP3 の比較カウンタ		CMP4 の比較カウンタ		CMP5 の比較カウンタ			
	REN4	b1, b0	カウンタ	b9, b8	カウンタ	b17, b16	選択	b25, b24	選択	REN5	b2-b0	選択	
		0 0	CTR1 選択	0 0	CTR1 選択	1 0	CTR3 選択	0 0	CTR1		0 0 0	CTR1	
								0 1	CTR2		0 0 1	CTR2	
								1 1	CTR4		0 1 1	CTR4	
比較方法	in SLS	RCMP1<CTR1		RCMP2>CTR1		RCMP3<CTR3 で脱調		RCMP4=比較カウンタ		RCMP5=比較カウンタ			
	REN4	b4-b2		b12-b10		b20-b18		b29-b26		REN5	b5-b3		
		1 1 0		1 1 0		1 0 1	RCMP3<CTR3	0 0 0 0	比較 ※1		0 0 1	比較 ※1	
		0 0 0	不使用	0 0 0	不使用	0 0 0	不使用	0 0 1 0	比較 ※2		0 1 0	比較 ※2	
								0 0 1 1	比較 ※3		0 1 1	比較 ※3	
成立時動作選択	REN4	b6, b5		動作		b14, b13		動作		b31, b30		動作	
		0 0		処理なし		0 0		処理なし		0 0		処理なし	
		0 1		即停止		0 1		即停止		0 1		即停止	
		1 0		減速停止		1 0		減速停止		1 0		減速停止	
比較データの 設定		RCMP1 (+側)		RCMP2 (-側)		RCMP3		RCMP4		RCMP5			
		+数値		-数値		0 ~ 3 2, 7 6 7		2 ⁻²⁸ ~ 2 ²⁸		2 ⁻²⁸ ~ 2 ²⁸			
他軸起動条件の コンパレータ選択											REN5: b19-b16		
											0 1 0 1		CMP5 成立で 他軸起動
											0 0 0 0		非選択
起動する 他軸を選択											REN5: b21, b20		
											0 0		X (V, X1, X2, X3) 軸から
											0 1		Y (W, Y1, Y2, Y3) 軸から
											1 0		Z (A, Z1, Z2, Z3) 軸から
											1 1		U (B, U1, U2, U3) 軸から

※1：カウント方向に無関係に比較， ※2：カウントアップ中に比較， ※3：カウントダウン中に比較

表 10. 6-3 コンパレータの標準的使用

10. 6. 3 ソフトリミット機能

コンパレータ 1 (CMP 1) と 2 (CMP 2) を利用して、ソフトリミット (SLS) 機能が行えます。

CMP 1 と CMP 2 の比較カウンタには、CTR 1 (指令位置) を選択して下さい。

CMP 1 を (+) 側リミット値、CMP 2 を (-) 側リミット値として使用し、比較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ことができます。

- 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。

また、スタートコマンド書き込み時に SLS が ON 状態の場合、SLS が ON 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。

【設定例】

REN4 = 00003838h : CMP 1 を (+) 側, CMP 2 を (-) 側 SLS に使用.
 リミット動作時は即停止に設定してある.
 RCMP1 = 100,000 : CMP 1 レジスタに (+) 側リミット値
 RCMP2 = -100,000 : CMP 2 レジスタに (-) 側リミット値

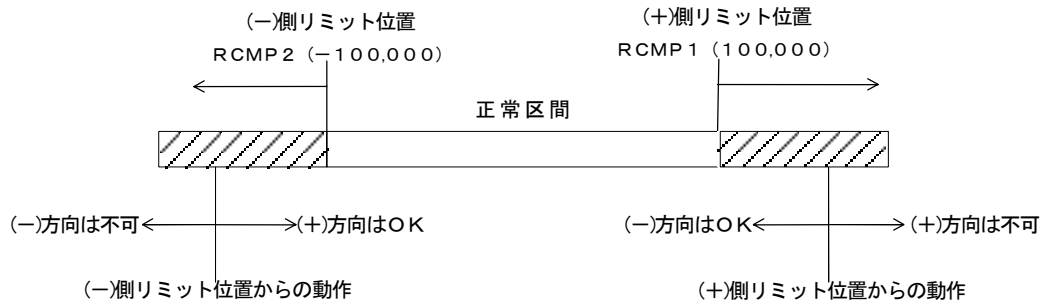


図 10. 6-2 ソフトリミット

■ソフトリミットの設定例

【 CPD534 ライブラリ関数による記述例 】

デバイスハンドルで指定された CPD 530 の指定された軸のソフトリミットを設定します。

```
DWORD hcp530_SetSls( DWORD hDevID, WORD axis, long psIs, long msIs, WORD selctr, WORD stop );
```

《引数》

- ◆ DWORD hDevID : 対象デバイスのデバイスハンドル.
- ◆ WORD axis : 軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]
- ◆ long psIs : プラスソフトリミット [パルス数]
- ◆ long msIs : マイナスソフトリミット [パルス数]
- ◆ WORD selctr : カタの選択 [0:不使用, 1:カタ 1]
- ◆ WORD stop : 停止方法 [0:即停止, 1:減速停止]

※初期値は selctr = 0 不使用になっています.

(注) psIs は必ず msIs より大きくして下さい.

10. 6. 4 コンパレータ条件成立で他軸起動 (条件付他軸スタート—2)

よくある応用例は図 10. 6-3 のように X 軸が上下軸, Y 軸が水平軸の場合の例で, 今 X 軸が上昇して P1 点を通過したとき Y 軸は移動開始する.

次に Y 軸が P2 を通過した時 X 軸は降下開始をする. すなわち, 「X 軸 P1 座標一致で Y 軸起動, Y 軸の P2 座標一致で X 軸起動」がコンパレータ条件成立で他軸起動の機能例です.

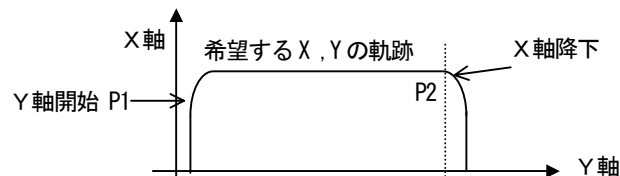


図 10. 6-3 条件付他軸起動

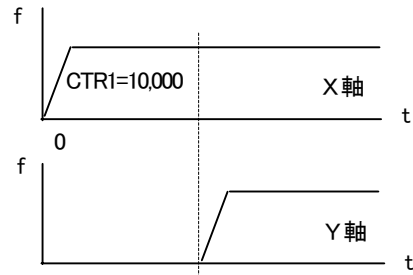
この例では, 実際は CMP 5 のプリレジスタを利用して P2 もセットする行程が必要です.

■コンパレータ条件成立で他軸の簡単な例・・・次ページは X 軸が P1 点を通過した時 Y 軸を起動する例です.

【CPD534 ドライバ関数による記述例】

ドライバ I/F 用 DLL 関数の応用例を記します。
次の例では、すでにレジスタ及びオプションポートの
初期化が終わっているものとします。

X軸のコンパレータ条件一致でY軸スタート
以下の場合、X軸の指令位置が10000になった
時点でY軸がスタートします。



```
DWORD ret; //関数の戻り値
```

```
//X軸コンパレータ5データ設定(比較データ=10000)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
                  0, //X軸を指定
                  0x8B, //コンパレータ5 (PRCP5) を指定
                  10000 ); //コンパレータ5用の比較データ
```

```
//X軸環境レジスタ5設定(コンパレータ比較条件設定, コンパレータ成立時の処理設定)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
                  0, //X軸を指定
                  0xA0, //環境レジスタ5 (REN5) を指定
                  0x50008 ); //カンタ1=コンパレータ5で他軸起動出力
```

```
//Y軸環境レジスタ5設定(他軸起動入力選択)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
                  1, //Y軸を指定
                  0xA0, //環境レジスタ5 (REN5) を指定
                  0 ); //X軸からの他軸起動入力を選択
```

```
//X軸動作モード設定(+方向連続送り)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
                  0, //X軸を指定
                  0x87, //動作モード (PRMD) を指定
                  0 ); //+方向連続送り
```

```
//Y軸動作モード設定(X軸からの他軸起動入力で+方向連続送り)
```

```
ret = cp530_wReg( hDevID, //デバイスハンドル
                  1, //Y軸を指定
                  0x87, //動作モード (PRMD) を指定
                  0x80000 ); //X軸からの他軸起動入力で+方向連続送り
```

```
//スタートコマンド発行(X軸は即スタート, Y軸はX軸からの他軸起動入力でスタート)
```

```
ret = cp530_wCmdW( hDevID, //デバイスハンドル
                  0,
                  0x353 ); //X Y軸高速スタート
```

10. 7 次動作連続実行（プリレジスタ連続使用）

動作レジスタの「プリレジスタ（2nd プリレジスタ）」と「1st プリレジスタ」を切れ目なく使うことを「プリレジスタ連続使用」あるいは「次動作連続実行」といいます。（⇒ 5. 1. 2 レジスタ書き込み、読出し）

（1）プリレジスタ連続使用の要領

MST S b 1 4（動作作用プリレジフル）をチェックしながら、以下のように運用する。

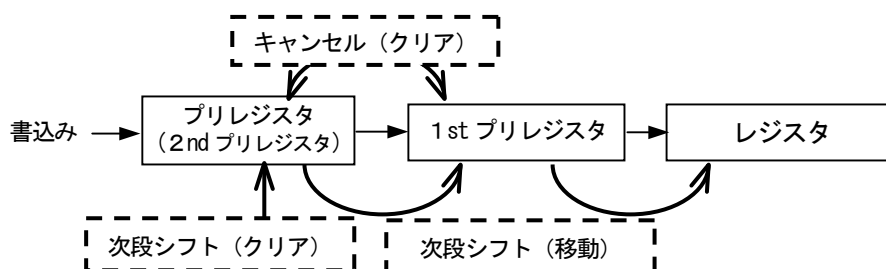
- ①MST S b 1 4 = '1'（動作作用プリレジフル）を検出したら、b 1 4 = '0'（動作作用プリレジ空き）になるまで待ち、b 1 4 = '0'になったら、次のデータ・スタートコマンドをプリレジスタに書く。
- ②全てのデータの書き込みが終了したら全ての動作が完了するのを待つ。
全ての動作が完了した時のMST Sは b 0 = 0（スタートコマンドが書込まれた）、かつ b 3 = 1（停止中）の状態になる。

（2）プリレジ制御コマンド

コマンド名	CMD	機 能	略 称
動作プリレジ キャンセル	2 6 h	動作プリレジの内容をキャンセル	PREGCAN
CMP 5 プリレジ キャンセル	2 7 h	CMP 5 用比較プリレジの内容をキャンセル	PCMPCAN
動作プリレジ 次段シフト	2 b h	動作プリレジの内容を次段へシフト	PREGSFT
CMP 5 プリレジ 次段シフト	2 c h	CMP 5 用比較プリレジの内容を次段へシフト	PCMPSFT

■動作プリレジキャンセル、CMP 5 プリレジキャンセルは「プリレジスタ」と「2nd プリレジスタ」の内容が全てキャンセルされます。（現在レジスタで動作中にキャンセルコマンドを発行すると、レジスタの動作が終了した時点で通常に動作終了することになる。）

■次段シフトコマンドを発行すると各レジスタがレジスタ方向にシフトされ、プリレジスタは空になります。レジスタには「1st プリレジスタ」の内容が入りますので、動作中は注意が必要です。



（3）次動作連続実行の例・・・XY 2 軸補間で点 P 1 ～点 P 7 の次動作連続実行を行います。

【 CPD534 ドライバ関数による記述例 】

P 1 から P 2 まで「加速ブロック」として直線補間で動作し、そこから半径 1 0 0 0 0 の円弧補間で P 3 まで「定速ブロック」移動し、P 4 まで「減速ブロック」として直線補間で動作します。

そこからもう一度同じ動作を、P 4 ～直線補間・加速～P 5 ～円弧補間・定速～P 6 ～直線補間・減速～P 7 で繰り返します。

また速度等の設定はベース速度 5 0 0 p p s (RFL = 5 0 0)、動作速度 5 0 0 0 p p s (RFH = 5 0 0 0)、倍率 1 倍(RMG = 2 9 9)、加減速時間 0. 5 s e c (RUR = RDR = 5 4 5)、直線加減速としてあります。

[C 言語]

// 連続実行データの構造体（X 軸：補間代表軸）

typedef struct {

long rmd; //XY 動作モード

long xmv; //X 移動量

long ymv; //Y 移動量

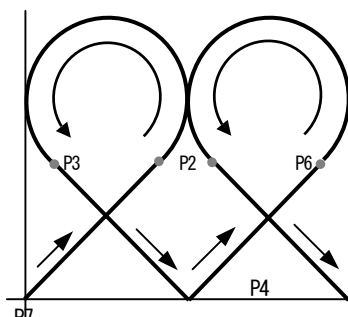
long xcn; //X 中心

long ycn; //Y 中心

long rdp; //X 減速開始点

WORD cmd; //X スタートコマンド

} DATABLK;



各位置の座標データ

P 1 : (0, 0)

P 2 : (17071, 17071)

P 3 : (2929, 17071)

P 4 : (20000, 0)

P 5 : (37071, 17071)

P 6 : (22929, 17071)

P 7 : (40000, 0)

```

// 設定データ
DATABLE dt_blk[6]={
//動作モード, X 移動量, Y 移動量, X 中心, Y 中心, 減速開始点(※1), スタートコマンド
{ 0x800a061, 17071, 17071, 0, 0, 0, 0x353 }, //dt_blk[0] 加速 1
{ 0x8008065, -14142, 0, -7071, 7071, 0, 0x351 }, //dt_blk[1] 定速 1
{ 0x800a061, 17071, -17071, 0, 0, 972, 0x352 }, //dt_blk[2] 減速 1
{ 0x800a061, 17071, 17071, 0, 0, 0, 0x353 }, //dt_blk[3] 加速 2
{ 0x8008065, -14142, 0, -7071, 7071, 0, 0x351 }, //dt_blk[4] 定速 2
{ 0x800a061, 17071, -17071, 0, 0, 972, 0x352 } };

WORD ptr = 0; //データポインタ
WORD msts; //メインステータス
DWORD ersts; //エラーステータス
DWORD evsts; //イベントステータス

case IDC_BTN_LIN: //スタートボタンがクリックされた
// 動作プロファイルスタックを監視しながらデータ設定
while(1) {
if(ptr > 5) break; //データ終了(最大6データ)
while(1) {
cp530_rMstsW(hDev, 0, &msts); //X:メインステータス読み込み
if(msts & 0x10) { //エラー発生
cp530_rReg(hDev, 0, 0xf2, &ersts); //X:エラーステータス読み込み
cp530_rReg(hDev, 1, 0xf2, &ersts); //Y:エラーステータス読み込み
cp530_wCmdW(hDev, 0, 0x326); //X:動作プロファイルスタックキャンセル
return 0; //処理キャンセル(終了)
}
if((msts & 0x4000)==0) break; //動作プロファイルスタックでなければループを抜ける
}
// データ書き込み
cp530_wReg(hDev, 0, 0x87, dt_blk[ptr].rmd); //X:動作モード
cp530_wReg(hDev, 1, 0x87, dt_blk[ptr].rmd); //Y:
cp530_wReg(hDev, 0, 0x80, dt_blk[ptr].xmv); //X:移動量
cp530_wReg(hDev, 1, 0x80, dt_blk[ptr].ymv); //Y:
cp530_wReg(hDev, 0, 0x88, dt_blk[ptr].xcn); //X:中心位置
cp530_wReg(hDev, 1, 0x88, dt_blk[ptr].ycn); //Y:
cp530_wReg(hDev, 0, 0x86, dt_blk[ptr].rdp); //X:減速開始点
cp530_wCmdW(hDev, 0, dt_blk[ptr].cmd); //X:スタート(X, Y)
ptr++;
}

// 2. 全書き込みデータの動作完了を待つ
while(1) {
cp530_rMstsW(hDev, 0, &msts); //X:メインステータス読み込み
if(msts & 0x0010) { //エラー発生
cp530_rReg(hDev, 0, 0xf2, &ersts); //X:エラーステータス読み込み
cp530_rReg(hDev, 1, 0xf2, &ersts); //Y:エラーステータス読み込み
cp530_wCmdW(hDev, 0, 0x326); //X:動作プロファイルスタックキャンセル
return 0; //処理キャンセル(終了)
}
if(msts & 0x0020) cp530_rReg(hDev, 0, 0xf3, &evsts); //X:イベントステータス読み込み
if((msts & 0x0009) == 0x0008) break; //動作終了
}
break;

case IDC_BTN_STP: // 停止ボタンがクリックされた
cp530_wCmdW(hDev, 0, 0x326);
cp530_wCmdW(hDev, 0, 0x34a);
break;
}
break;

```

※ 1. 減速ブロックの減速開始点(SDP)の計算 (2軸直線補間 45° の場合の近似計算)

$$SDP[ハルス] = \frac{(RFH^2 - RFL^2) \times (RDR + 1)}{(RMG + 1) \times 32,768} \times (\text{長軸の移動量のスカラー} / \text{補間移動量のスカラー})$$

$$SDP[ハルス] = \frac{(5,000^2 - 500^2) \times (545 + 1)}{(299 + 1) \times 32,768} \times (17,071 / \sqrt{17,071^2 + 17,071^2})$$

$$SDP[ハルス] \div 972$$

10.8 複数のPCL間の動作方法

8軸または12軸ボードでは、ここにいるPCL間の直線補間動作が組合せとして使用できます。

■ 8軸ボードの例・・・2個のPCLにまたがる4軸直線補間を行います。

【CPD534 ドライバ関数による記述例】

X, Z, W, A軸の直線補間を行います。長軸はX軸とします。

長軸の速度は1000ppsとし、FH定速(※1)で動作します。

すでに速度データ(RFH = 1000)は補間軸すべてに設定されているものとします。

また動作モードは0x40063(動作モードは直線補間動作: 0x63(※2), 同時スタート入力によりスタート)とし、補間軸すべてに設定されているものとします。

[C言語]

```

DWORD   ret;           //関数の戻り値

ret = cp530_wReg( hDevID, 0,           //X軸移動量設定(移動量 10000)
                 0x80,                //X軸を指定
                 10000 );              //移動量(PRMV)を指定
                                     //移動量 10000
                                     //長軸の移動量設定(移動量 10000)
ret = cp530_wReg( hDevID, 0,           //X軸を指定
                 0x88,                //長軸の移動量(PRIIP)を指定
                 10000 );              //移動量 10000
                                     //Z軸移動量設定(移動量 5000)
ret = cp530_wReg( hDevID, 2,           //Z軸を指定
                 0x80,                //移動量(PRMV)を指定
                 5000 );              //移動量 5000
                                     //長軸の移動量設定(移動量 10000)
ret = cp530_wReg( hDevID, 2,           //Z軸を指定
                 0x88,                //長軸の移動量(PRIIP)を指定
                 10000 );              //移動量 10000
                                     //W軸移動量設定(移動量 1000)
ret = cp530_wReg( hDevID, 5,           //W軸を指定
                 0x80,                //移動量(PRMV)を指定
                 1000 );              //移動量 1000
                                     //長軸の移動量設定(移動量 10000)
ret = cp530_wReg( hDevID, 5,           //W軸を指定
                 0x88,                //長軸の移動量(PRIIP)を指定
                 10000 );              //移動量 10000
                                     //A軸移動量設定(移動量 2000)
ret = cp530_wReg( hDevID, 6,           //A軸を指定
                 0x80,                //移動量(PRMV)を指定
                 2000 );              //移動量 2000
                                     //長軸の移動量設定(移動量 10000)
ret = cp530_wReg( hDevID, 6,           //A軸を指定
                 0x88,                //長軸の移動量(PRIIP)を指定
                 10000 );              //移動量 10000
                                     //スタートコマンド発行(X, Z軸)
ret = cp530_wCmdW( hDevID, 0,          //X～U軸のグループを指定
                 0x551 );              //X, Z軸FH定速スタート
                                     //スタートコマンド発行(W, A軸)
ret = cp530_wCmdW( hDevID, 5,          //V～B軸のグループを指定
                 0x651 );              //W, A軸FH定速スタート
                                     //同時スタートコマンド発行
ret = cp530_wCmdW( hDevID, 0, 0x06 ); //同時スタート

```

[設定データ]

設定値	X軸	Z軸	W軸	A軸
RMD(動作モード)	0x40063	0x40063	0x40063	0x40063
RMV(移動量)	10000	5000	1000	2000
RIP(長軸の移動量)	10000	10000	10000	10000
RFH(動作速度)	1000	1000	1000	1000
長軸／従軸	長軸	従軸	従軸	従軸

※1. この場合の補間動作は定速のみです。

※2. この直線補間動作の動作モードは'0x63',連続直線補間動作の動作モードは'0x62'となります。

10.9 手動パルス入力モード使用方法

手動パルス入力を利用する方法を以下に示します。パルスのモード設定は「4. 3. 1 動作モードレジスタ (2) 動作モード ⑦パルス動作モード」を参照して下さい。

■接続：各軸のエンコーダ入力端子にパルスを接続します。（+／－パルス方式も⇒ 9. 2 RENV2）
パルスと各軸出力の対応は1対1です。（下図）

■使用上の制限：エンコーダ入力には各軸に1組ですのでパルス入力に使用した場合はフィードバックのエンコーダ信号は入力出来ません。

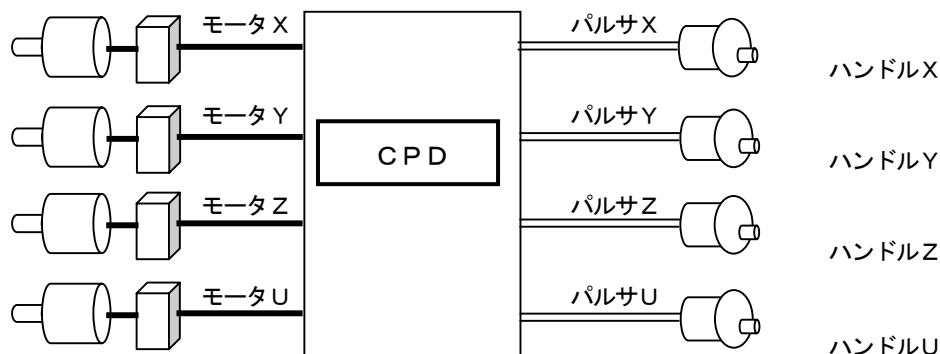


図10. 9-1 パルス入力

■通常の手動パルス送り

ハンドルを+または一方向に操作するとそのパルスの速さでモータが送られる動作です。操作中にELSを検出するとそれ以上はモータは送られません。ELSを抜出す方向には動作します。

パルスの送りパルスレートの上限值を動作速度レジスタPRFHに設定します。もし、パルスの送り速度がこの値を超えた場合は指令パルスは停止します。（⇒6. 3 REST b15）

■ステータス処理

(1) 手動パルス入力モードにおいては「イベント報告方式」のステータス処理に於いて RIST b0 は使用しません。すなわち、RIRQのb0 = '0' マスクして使用します。（ハンドル方向が反転する度に MST b5 SINT が報告されるため）

(2) ELSを検出し停止はするが、MST b4はセットされません。
すなわちエラーステータス（REST）のb5, b6はセットされません。

■手順

- (1) 対象軸の動作モードレジスタに「パルスによる連続動作(01h)」を設定。
- (2) その軸RENV2レジスタに「90度位相差入力倍率」を設定。
- (3) その軸に「FH定速スタート」を発行します。以後パルスの回転に同期して軸動作します。
この時、FH速度を最高速とし、FH速度をこえてパルスを回し続けるとエラー停止します。（予めFHに適切な値を設定しておきます。）
- (4) 「即停止(49h)」を発行することで、パルス連続動作モードが解除されます。

【CPD534 ドライバ関数による記述例】

環境設定2の90度位相差通倍を設定してあるものとする。

```
switch (LOWORD(wParam)) {
    // 手動パルス開始
    case IDC_BTN_STR:
        // 動作モード設定(パルスによる連続動作)
        cp530_wReg(hDevID, 0, 0x87, 0x00000001); //パルスによる連続動作
        cp530_wReg(hDevID, 0, 0x82, 5000);        // 動作速度設定 FH = 5000
        cp530_wCmdW(hDevID, 0, 0x51);            // FH 定速スタート
        break;

    // 手動パルス終了
    case IDC_BTN_STP:
        cp530_wCmdW(hDevID, 0, 0x49);            // 「即停止」
        break;
}
```

1 1. CPDシリーズ追加機能について

CPDシリーズの4軸、8軸、12軸ボードで基板の半田面にあるシリアル番号の末尾にBがあるもの（例、「□○○○○○B」（□は英字、○は数字））は以下の機能が追加されています。

No	項 目	機能概要	備 考
1	自軸を含めた指定軸停止起動	自軸を含めた指定軸の停止により軸スタート	RENV2のbit29=1で有効
2	マシンロック	指令出力マスク	RENV2のbit28=1で有効
3	カウンタラッチ後クリア	OLS入力等によるカウンタラッチ後、カウンタを0クリア	RENV5 bit24=1でCTR1をクリア bit25=1でCTR2をクリア bit26=1でCTR3をクリア bit27=1でCTR4をクリア
4	回転軸カウンタ機能	回転軸を対象として、CTR1(CTR2)はRCMP1(RCMP2)の設定値と0の間をリングカウント	RENV4 bit7=1でCTR1をリングカウント bit15=1でCTR2をリングカウント

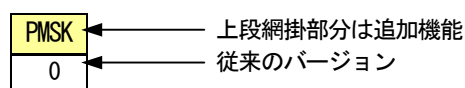
表 1 1. 1-1 追加機能一覧表

これらの機能を使用する場合は、ドライバ関数を使用します。

1 1. 1 各種レジスタ内の追加ビット

従来未使用となっていたレジスタ内のビットに新しい機能が追加されました。この追加は上下2段に分割されたビットの上段が該当します。

【 凡 例 】



(1) 環境レジスタ2 (RENV2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POFF	EOFF	SMA	PMSK	0	PDIR	PULSR	EZL	EDIR	ENCM	0	0	0	0	0	0
0	0	0	0												

ビット	名 称	機 能	記 事
28	PMSK	'1' = マシンロック	外部への指令出力パルスをマスク
29	SMA	'1' = 「停止指定軸設定(自軸可能)」で自軸指定を有効	
30	EOFF	'1' = エンコーダA相／B相入力をマスク	
31	POFF	'1' = パルサA相／B相入力をマスク	

表 1 1. 1-2 環境レジスタ2追加ビット

(2) 環境レジスタ4 (RENV4)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2RM	CMP2M	CMP2D	CMP2C	C1RM	CMP1M	CMP1D	CMP1C								
0				0											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP4M	CMP4D	CMP4C	0	CMP3M	CMP3D	CMP3C									

ビット	名 称	機 能	記 事
7	C1RM	'1' = コンパレータ1と対でCTR1を回転軸カウンタとして使用	
15	C2RM	'1' = コンパレータ2と対でCTR2を回転軸カウンタとして使用	

表 1 1. 1-3 環境レジスタ4追加ビット

(3) 環境レジスタ5 (RENV5)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CTLCHF	CTLCH	0	IDLP	CMP5M	CMP5D	CMP5C								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	CJ4L	CJ3L	CJ2L	CJ1L	0	0	TRIGRIN sel	TRIGROT sel				
				0	0	0	0								

ビット	名 称	機 能	記 事
24	CJ1L	'1' = CTR1のラッチ時, CTR1を0クリア	
25	CJ2L	'1' = CTR2のラッチ時, CTR2を0クリア	
26	CJ3L	'1' = CTR3のラッチ時, CTR3を0クリア	
27	CJ4L	'1' = CTR4のラッチ時, CTR4を0クリア	

表 11. 1-4 環境レジスタ5追加ビット

11. 2 追加機能詳細

(1) 自軸を含めた指定軸停止起動

RENV2 : bit29(SMAX) = 1 で, 自軸を含めた指定軸停止によるスタートが有効になります.

■設定箇所

環境設定2 (RENV2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POFF	EOFF	SMAX	PMSK	IEND	PDIR	PULSR		EZL	EDIR	ENCM		PINF	EINF	0	0
ビット	名 称		機 能							記 事					
29	SMAX		'1' = 「停止指定軸設定(自軸可能)」で自軸指定を有効												

動作モードレジスタ (RMD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	1	0	同他軸 停止	同時 ストップ	停止指定軸設定 (自軸可能)				条件付スタート = 1 1		SEQ No	
ビット		名 称		機 能			記 事								
20,23		停止指定軸設定		全指定ビット軸の停止 で動作開始			条件付きスタート(b18, 19) = 11 (指定軸停止でスタート)の時 b20=X (V, X1, X2, X3), b21=Y (W, Y1, Y2, Y3), b22=Z (A, Z1, Z2, Z3), b23=U (B, U1, U2, U3)								

表 11. 2-1 指定軸停止の設定レジスタ

■設定に伴う機能

停止指定軸				条件	機 能	記 事
b23 U(B, U1, U2, U3)	b22 Z(A, Z1, Z2, Z3)	b21 Y(W, Y1, Y2, Y3)	b20 X(V, X1, X2, X3)	b19, 18		
0	0	0	1	11	X(V, X1, X2, X3) 軸停止でスタート	RENV2 の bit29 = '1' では 自軸停止を含めて 自軸のスタートが可能 RENV2 の bit29 = '0' では 他軸の停止でのみ 自軸のスタート可能
0	0	1	0	11	Y(W, Y1, Y2, Y3) 軸停止でスタート	
0	1	0	0	11	Z(A, Z1, Z2, Z3) 軸停止でスタート	
1	0	0	0	11	U(B, U1, U2, U3) 軸停止でスタート	
0	0	1	1	11	X(V, X1, X2, X3) 軸と Y(W, Y1, Y2, Y3) 軸停止でスタート	
1	1	1	1	11	全軸停止でスタート	

表 11. 2-2 指定軸停止の設定詳細

■設定例

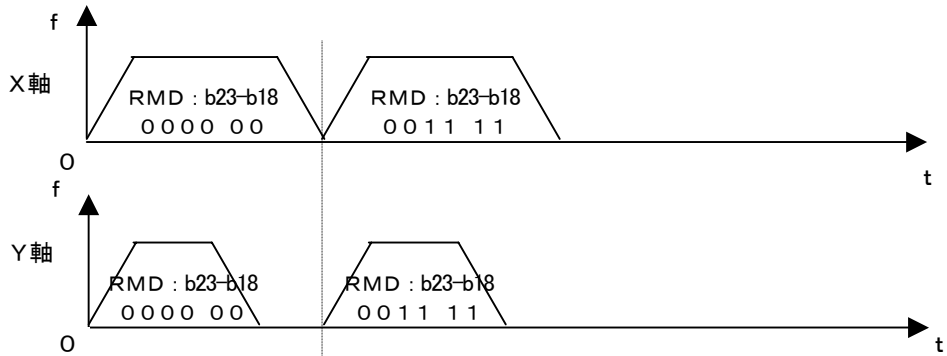


図 11. 2-1 指定軸停止による起動

(2) マシンロック

環境設定 2 (RENV2) : bit28 (PMSK) = 1 で指令出力パルスをマスクします。

指令位置カウンタは動作します。

これにより、メカを動作させずに、現在位置でアプリケーションの指令状況をチェックします。

環境設定 2 (RENV2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POFF	EOFF	SMAx	PMSK	IEND	PDIR	PULSR	EZL	EDIR	ENCM	PINF	EINF	0	0		
ビット	名 称	機 能										記 事			
28	PMSK	'1' = マシンロック										外部への指令出力パルスをマスク			

表 11. 2-3 マシンロックの設定レジスタ

(3) カウンタラッチ後クリア

OLS入力等によるカウンタラッチ後、カウンタを0クリアします。

環境設定 5 (RENV5)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	CJ4L	CJ3L	CJ2L	CJ1L	0	0	TRIGRIN sel	TRIGROT sel				
ビット	名 称	機 能										記 事			
24	CJ1L	'1' = CTR 1 のラッチ時、CTR 1 を0クリア													
25	CJ2L	'1' = CTR 2 のラッチ時、CTR 2 を0クリア													
26	CJ3L	'1' = CTR 3 のラッチ時、CTR 3 を0クリア													
27	CJ4L	'1' = CTR 4 のラッチ時、CTR 4 を0クリア													

表 11. 2-4 カウンタラッチ後のクリア設定レジスタ

(4) 回転軸カウンタ機能

環境設定 4 レジスタの設定でCTR 1・CTR 2が回転軸カウンタとして利用できます。

環境設定 4 (RENV4)

CTR 2 を回転軸カウンタに設定								CTR 1 を回転軸カウンタに設定							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2RM = 1	CMP2M = 0 0		CMP2D = 0 0 0		CMP2C = 0 0			C1RM = 1	CMP1M = 0 0		CMP1D = 0 0 0		CMP1C = 0 0		
RCMP 2 の設定値 = 1 回転当たりのパルス数 - 1								RCMP 1 の設定値 = 1 回転当たりのパルス数 - 1							
CTR 2 の値 = 0 ~ "RCMP 2 設定値"								CTR 1 の値 = 0 ~ "RCMP 1 設定値"							
1 回転パルス数が 3, 6 0 0 パルスの場合、カウンタ値は 0 ~ 3, 5 9 9															

表 11. 2-5 回転軸カウンタ機能の設定レジスタ