

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 〈ソフトウェア編〉

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	—CPD534
	H P C I	—CPD532
	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	H P C 1 0 4	—CPD132
P C 1 0 4 D	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.

WindowsVista, WindowsXP, Windows2000, WindowsNT 4.0, Windows98, VisualC++, Visual Basic, MS-DOS は Microsoft Corporation の米国及びその他の国における登録商標です. その他, 記載されている会社名, 製品名は, 各社の商標又は登録商標です.

株式会社 ハイバーテック
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3.40版 2008年 2月20日発行
不許複製・転載

はじめに

この度は、弊社NCボードシリーズをご採用頂きまして、誠に有り難う御座います。

本書は、添付ソフトウェアをご使用して頂く場合の取り扱い、留意点について記入してありますので、必ずご一読の上ご利用をお願い致します。

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

安全上の注意

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



警告

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



注意

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

1. 対象ユーザー



注意



本製品およびマニュアルは、以下の様な、ユーザーを対象としています。

- ・制御用電子機器およびパソコン等に付いて基本的な知識を有している方。
- ・OSの操作およびソフトウェア開発環境に付いて基本的な知識を有している方。

2. 適合OS



注意



本製品は、WindowsXP Home Edition / Professional, Windows2000, WindowsNT4.0, Windows98及びDOSにおいてボードの制御を行う為のソフトウェアです。上記以外のOSでのご使用については、弊社営業までお問合せ下さい。

なお、HUSBボードについてはWindowsNT4.0, DOSが、HCPCIボードについてはWindows95, DOSが対象外となります。

3. ハードウェアの設定・取付け・接続



警告



ボードの取付け、配線に際しては、ユーザーズマニュアル<個別ボード編>を良くお読みいただき、これらの内容にしたがって実施願います。

4. ユーザープログラムの作成



警告



ユーザープログラムの作成にあたっては、モータや装置の特性を考慮し、必要なインターロック・安全対策処理等を十分盛り込んだ設計として下さい。

プログラムコード・データのわずかな違いにより予想外の動作をして、機器や人体に損傷を与える恐れがあります。プログラム作成・試運転時共、十分な注意をお願いいたします。

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>・・・本マニュアル

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>

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

本マニュアルは「1. ユーザーズマニュアル<個別ボード編>」及び「3. ユーザーズマニュアル<共通編>」を一読されたものとして、本ボード添付ソフトウェアを使用する場合の取り扱い、留意点について説明しています。本マニュアルと合わせてこれらのマニュアルを適宜参照・確認頂きますようお願い致します。

目 次

1. はじめに.....	1
1. 1 概要.....	4
1. 2 準備.....	5
1. 2. 1 ライブラリ関数の使用.....	5
1. 2. 2 ドライバ関数の使用.....	6
1. 2. 3 C++アプリケーションでの使用.....	6
1. 3 関数の戻り値.....	7
2. ライブラリ関数.....	8
2. 1 VC++用ライブラリ関数.....	10
■デバイス関係.....	10
(1) hcpxxx_GetDevInfo() ボード枚数の取得, デバイス情報の取得.....	10
(2) hcpxxx_DevOpen() デバイスのオープン, レジスタとオプションポートの初期化.....	10
(3) hcpxxx_DevClose() デバイスのクローズ.....	11
■初期設定.....	11
(4) hcpxxx_SetOrgMode() 原点復帰モードの設定.....	11
(5) hcpxxx_SetEls() E L Sの設定.....	13
(6) hcpxxx_SetOls() O L Sの設定.....	14
(7) hcpxxx_SetSvAlm() S V A L Mの設定.....	14
(8) hcpxxx_SetEz() エンコーダZ相の設定.....	14
(9) hcpxxx_SetDlsSel() D L S / P C Sの選択, 設定.....	15
(10) hcpxxx_SetInpos() I N P O Sの設定.....	15
(11) hcpxxx_SetSvCtrCl() 偏差カウンタクリア出力の設定.....	15
(12) hcpxxx_SetSls() ソフトリミットの設定.....	16
(13) hcpxxx_SetCmdPulse() 指令パルスの出力形式の設定.....	16
(14) hcpxxx_SetAccProfile() S字/直線加減速の切替.....	16
(15) hcpxxx_SetAutoDec() 減速開始点計算方式の自動計算/手動計算切替.....	17
(16) hcpxxx_mSetDls() 汎用入力/±D R / P C S / D L S入力選択, 設定.....	17
(17) hcpxxx_mSetLtc() 汎用入力/ L A T C H入力の選択, 設定.....	18
(18) hcpxxx_mSetInpos() I N P O S入力/汎用入力の選択, 設定.....	18
(19) hcpxxx_mSetSvCtrCl() 汎用出力/ S V C T R C L出力選択, 設定.....	18
■状態読出し.....	19
(20) hcpxxx_ReadMainSts() メインステータスの読出し.....	19
(21) hcpxxx_ReadErrorSts() エラーステータスの読出し.....	19
(22) hcpxxx_ReadEventSts() イベントステータスの読出し.....	19
(23) hcpxxx_ReadSubSts() サブステータスの読出し.....	19
(24) hcpxxx_ReadExSts() 拡張ステータスの読出し.....	19
(25) hcpxxx_ReadIpSts() 補間ステータスの読出し.....	19
(26) hcpxxx_ReadSpd() 現在速度の読出し.....	19
(27) hcpxxx_ReadCtr() カウンタの読出し.....	20
■動作設定.....	21
(28) hcpxxx_SetFLSpd() ベース速度の設定.....	21
(29) hcpxxx_SetAuxSpd() 補助速度の設定.....	21
(30) hcpxxx_SetAccRate() 加速レートの設定.....	21
(31) hcpxxx_SetDecRate() 減速レートの設定.....	21
(32) hcpxxx_SetMult() 速度倍率設定値 (M G) の設定.....	22
(33) hcpxxx_SetEventMask() イベントマスクの設定.....	22
(34) hcpxxx_SetDecPoint() 減速開始点の設定.....	23
■運用設定.....	24
(35) hcpxxx_WritOpeMode() 動作モードの設定.....	24
(36) hcpxxx_WritFHSpd() 動作速度の設定.....	25
(37) hcpxxx_WritPos() 位置決め移動量の設定.....	25
(38) hcpxxx_WritLine() 直線補間の移動量の設定.....	25
(39) hcpxxx_WritCircl() 円弧補間の移動量の設定.....	26
(40) hcpxxx_WritCtr() カウンタプリセット.....	26
■動作制御指令.....	27
(41) hcpxxx_DecStop() 減速停止.....	27

(4 2) hcpxxx_QuickStop()	即停止	27
(4 3) hcpxxx_EmgStop()	非常停止	27
(4 4) hcpxxx_SyDecStop()	同時減速停止	27
(4 5) hcpxxx_SyQuickStop()	同時即停止	27
(4 6) hcpxxx_AccStart()	加速スタート	27
(4 7) hcpxxx_CnstStartFH()	F H 定速スタート	27
(4 8) hcpxxx_CnstStartFL()	F L 定速スタート	27
(4 9) hcpxxx_CnstStartByDec()	F H 定速スタート後減速停止	27
(5 0) hcpxxx_SvOn()	サーボオン	28
(5 1) hcpxxx_SvOff()	サーボオフ	28
(5 2) hcpxxx_SvResetOn()	サーボリセットオン	28
(5 3) hcpxxx_SvResetOff()	サーボリセットオフ	28
(5 4) hcpxxx_PMOOn()	パルスモータ励磁オン	28
(5 5) hcpxxx_PMOOff()	パルスモータ励磁オフ	28
■加減速レート計算		28
(5 7) hcpxxx_CalAccRate()	加減速レートの計算	28
2. 2 V B用ライブラリ関数		29
■デバイス関係		29
(1) hcpxxx_GetDevInfo()	ボード枚数の取得, デバイス情報の取得	29
(2) hcpxxx_DevOpen()	デバイスのオープン, レジスタとオプションポートの初期化	29
(3) hcpxxx_DevClose()	デバイスのクローズ	30
■初期設定		30
(4) hcpxxx_SetOrgMode()	原点復帰モードの設定	30
(5) hcpxxx_SetEls()	E L S の設定	32
(6) hcpxxx_SetOls()	O L S の設定	33
(7) hcpxxx_SetSvAlm()	S V A L M の設定	33
(8) hcpxxx_SetEz()	エンコーダZ相の設定	33
(9) hcpxxx_SetDlsSel()	D L S / P C S の選択, 設定	34
(1 0) hcpxxx_SetInpos()	I N P O S の設定	34
(1 1) hcpxxx_SetSvCtrCl()	偏差カウンタクリア出力の設定	34
(1 2) hcpxxx_SetSls()	ソフトリミットの設定	35
(1 3) hcpxxx_SetCmdPulse()	指令パルスの出力形式の設定	35
(1 4) hcpxxx_SetAccProfile()	S 字/直線加減速の切替	35
(1 5) hcpxxx_SetAutoDec()	減速開始点計算方式の自動計算/手動計算切替	36
(1 6) hcpxxx_mSetDls()	汎用入力/±D R / P C S / D L S 入力選択, 設定	36
(1 7) hcpxxx_mSetLtc()	汎用入力/L A T C H 入力の選択, 設定	37
(1 8) hcpxxx_mSetInpos()	I N P O S 入力/汎用入力の選択, 設定	37
(1 9) hcpxxx_mSetSvCtrCl()	汎用出力/S V C T R C L 出力選択, 設定	37
■状態読出し		38
(2 0) hcpxxx_ReadMainSts()	メインステータスの読出し	38
(2 1) hcpxxx_ReadErrorSts()	エラーステータスの読出し	38
(2 2) hcpxxx_ReadEventSts()	イベントステータスの読出し	38
(2 3) hcpxxx_ReadSubSts()	サブステータスの読出し	38
(2 4) hcpxxx_ReadExSts()	拡張ステータスの読出し	38
(2 5) hcpxxx_ReadIpSts()	補間ステータスの読出し	38
(2 6) hcpxxx_ReadSpd()	現在速度の読出し	39
(2 7) hcpxxx_ReadCtr()	カウンタの読出し	39
■動作設定		40
(2 8) hcpxxx_SetFLSpd()	ベース速度の設定	40
(2 9) hcpxxx_SetAuxSpd()	補助速度の設定	40
(3 0) hcpxxx_SetAccRate()	加速レートの設定	40
(3 1) hcpxxx_SetDecRate()	減速レートの設定	40
(3 2) hcpxxx_SetMult()	速度倍率設定値 (M G) の設定	41
(3 3) hcpxxx_SetEventMask()	イベントマスクの設定	41
(3 4) hcpxxx_SetDecPoint()	減速開始点の設定	42
■運用設定		43
(3 5) hcpxxx_WritOpeMode()	動作モードの設定	43
(3 6) hcpxxx_WritFHSpd()	動作速度の設定	44
(3 7) hcpxxx_WritPos()	位置決め移動量の設定	44

(3 8) hcpxxx_WritLine()	直線補間の移動量の設定	44
(3 9) hcpxxx_WritCircI()	円弧補間の移動量の設定	45
(4 0) hcpxxx_WritCtr()	カウンタプリセット	45
■動作制御指令		46
(4 1) hcpxxx_DecStop()	減速停止	46
(4 2) hcpxxx_QuickStop()	即停止	46
(4 3) hcpxxx_EmgStop()	非常停止	46
(4 4) hcpxxx_SyDecStop()	同時減速停止	46
(4 5) hcpxxx_SyQuickStop()	同時即停止	46
(4 6) hcpxxx_AccStart()	加速スタート	46
(4 7) hcpxxx_CnstStartFH()	F H定速スタート	46
(4 8) hcpxxx_CnstStartFL()	F L定速スタート	46
(4 9) hcpxxx_CnstStartByDec()	F H定速スタート後減速停止	46
(5 0) hcpxxx_SvOn()	サーボオン	47
(5 1) hcpxxx_SvOff()	サーボオフ	47
(5 2) hcpxxx_SvResetOn()	サーボリセットオン	47
(5 3) hcpxxx_SvResetOff()	サーボリセットオフ	47
(5 4) hcpxxx_PMOOn()	パルスモータ励磁オン	47
(5 5) hcpxxx_PMOOff()	パルスモータ励磁オフ	47
■加減速レート計算		47
(5 7) hcpxxx_CalAccRate()	加減速レートの計算	47
2. 3 DOS版ライブラリ関数		48
■デバイス関係		48
(1) hcpxxx_GetDevInfo()	ボード枚数の取得, デバイス情報の取得	48
(2) hcpxxx_DevOpen()	デバイスのオープン, レジスタとオプションポートの初期化	48
(3) hcpxxx_DevClose()	デバイスのクローズ	48
■初期設定		48
(4) hcpxxx_SetOrgMode()	原点復帰モードの設定	48
(5) hcpxxx_SetEIs()	E L Sの設定	48
(6) hcpxxx_SetOIs()	O L Sの設定	49
(7) hcpxxx_SetSvAlm()	S V A L Mの設定	49
(8) hcpxxx_SetEz()	エンコーダZ相の設定	49
(9) hcpxxx_SetDIsSel()	D L S / P C Sの選択, 設定	49
(1 0) hcpxxx_SetInpos()	I N P O Sの設定	49
(1 1) hcpxxx_SetSvCtrCl()	偏差カウンタクリア出力の設定	49
(1 2) hcpxxx_SetSIs()	ソフトリミットの設定	50
(1 3) hcpxxx_SetCmdPulse()	指令パルスの出力形式の設定	50
(1 4) hcpxxx_SetAccProfile()	S字/直線加減速の切替	50
(1 5) hcpxxx_SetAutoDec()	減速開始点の設定方式の自動/手動切替	50
(1 6) hcpxxx_mSetDIs()	汎用入力/±DR/PCS/DLS入力選択, 設定	50
(1 7) hcpxxx_mSetLtc()	汎用入力/LATCH入力の選択, 設定	50
(1 8) hcpxxx_mSetInpos()	I N P O S入力/汎用入力の選択, 設定	51
(1 9) hcpxxx_mSetSvCtrCl()	汎用出力/SVCTRL出力選択, 設定	51
■状態読出し		51
(2 0) hcpxxx_ReadMainSts()	メインステータスの読出し	51
(2 1) hcpxxx_ReadErrorSts()	エラーステータスの読出し	51
(2 2) hcpxxx_ReadEventSts()	イベントステータスの読出し	51
(2 3) hcpxxx_ReadSubSts()	サブステータスの読出し	51
(2 4) hcpxxx_ReadExSts()	拡張ステータスの読出し	52
(2 5) hcpxxx_ReadIpSts()	補間ステータスの読出し	52
(2 6) hcpxxx_ReadSpd()	現在速度の読出し	52
(2 7) hcpxxx_ReadCtr()	カウンタの読出し	52
■動作設定		52
(2 8) hcpxxx_SetFLSpd()	ベース速度の設定	52
(2 9) hcpxxx_SetAuxSpd()	補助速度の設定	52
(3 0) hcpxxx_SetAccRate()	加速レートの設定	53
(3 1) hcpxxx_SetDecRate()	減速レートの設定	53
(3 2) hcpxxx_SetMult()	倍率設定値(MG)の設定	53
(3 3) hcpxxx_SetEventMask()	イベントマスクの設定	53

(3 4) hcpxxx_SetDecPoint()	減速開始点の設定	53
■運用設定		53
(3 5) hcpxxx_WritOpeMode()	動作モードの設定	53
(3 6) hcpxxx_WritFHSpd()	動作速度の設定	54
(3 7) hcpxxx_WritPos()	位置決め移動量の設定	54
(3 8) hcpxxx_WritLine()	直線補間の移動量の設定	54
(3 9) hcpxxx_WritCircl()	円弧補間の移動量の設定	54
(4 0) hcpxxx_WritCtr()	カウンタプリセット	54
■動作制御指令		55
(4 1) hcpxxx_DecStop()	減速停止	55
(4 2) hcpxxx_QuickStop()	即停止	55
(4 3) hcpxxx_EmgStop()	非常停止	55
(4 6) hcpxxx_AccStart()	加速スタート	55
(4 7) hcpxxx_CnstStartFH()	F H 定速スタート	55
(4 8) hcpxxx_CnstStartFL()	F L 定速スタート	55
(4 9) hcpxxx_CnstStartByDec()	F H 定速スタート後減速停止	56
(5 0) hcpxxx_SvOn()	サーボオン	56
(5 1) hcpxxx_SvOff()	サーボオフ	56
(5 2) hcpxxx_SvResetOn()	サーボリセットオン	56
(5 3) hcpxxx_SvResetOff()	サーボリセットオフ	56
(5 4) hcpxxx_PMOOn()	パルスモータ励磁オン	56
(5 5) hcpxxx_PMOff()	パルスモータ励磁オフ	57
■汎用D I O		57
(5 6) hcpxxx_mGetOptInp()	オプションポート設定の汎用入力ポートの確認	57
■加減速レート計算		57
(5 7) hcpxxx_CalAccRate()	加減速レートの計算	57
3. ドライバ関数		58
3. 1 ドライバ関数の種類		58
3. 2 ドライバ関数の詳細		59
(1) cpxxx_GetDeviceCount()	ボード枚数の取得	59
(2) cpxxx_GetDeviceInfo()	デバイス情報の取得	60
(3) cpxxx_OpenDevice()	デバイスのオープン	61
(4) cpxxx_CloseDevice()	デバイスのクローズ	61
(5) cpxxx_rMstsW()	メインステータスの読出し	62
(6) cpxxx_rSstsW()	サブステータスの読出し	63
(7) cpxxx_wCmdW()	制御コマンド書込み	64
(8) cpxxx_rReg()	レジスタ読出し	65
cpxxx_wReg()	レジスタ書込み	65
(9) cpxxx_rPortB()	オプションポートバイト読出し	67
cpxxx_wPortB()	オプションポートバイト書込み	67
cpxxx_rPortW()	オプションポートワード読出し	67
cpxxx_wPortW()	オプションポートワード書込み	67
(1 0) cpxxx_rBufDW()	入出力バッファ読出し	68
cpxxx_wBufDW()	入出力バッファ書込み	68
(1 1) cpxxx_SetIntCall()	割込処理関数の登録・削除	69
(1 2) cpxxx_GetDevVerNo()	バージョン番号の取得	69
(1 3) cpxxx_GetBoardCode()	ボード固有コードの取得	70
4. アプリケーション作成上の注意		71
4. 1 マルチスレッド使用時の注意点		71

1. はじめに

このマニュアルでは、CPDボードシリーズのソフトウェアの共通部分について解説を行います。
ソフトウェアの共通部分は次の通りです。

対応するOSとして次の5種類があります。

- Windows Vista (32bit) の各バージョン (一部の製品が未対応です)
(以降WinVistaと記します)
- Windows XP Professional/Home Edition
(以降WinXPと記します)
- Windows 2000 (以降Win2Kと記します)
- Windows NT 4.0 (以降WinNTと記します)
- Windows 98 SE (以降Win98と記します)
- DOS (MS-DOS, PC DOS)

CPDボード・ライブラリ関数レベル1として次の3種類があります。

ライブラリ関数と称し、この関数はソースプログラムで提供されます。

CPDボードの標準的な位置決め動作及び直線・円弧補間の基本的な動作を制御するための関数群です。

Windows版

- Microsoft Visual C++用 ・・以降は”VC++”と称します。
- Microsoft Visual Basic5.0用
Microsoft Visual Basic6.0用 ・・以降は”VB”と称します。

DOS版として各種C言語対応

- Microsoft C(V6), C++(V7), etc 用

デバイスドライバI/Fライブラリとして次の2種類があります。

ドライバ関数と称し、デバイスドライバとのI/F用ライブラリです。

Windows版

- デバイスドライバI/F DLL・・・WinXP, Win2K, WinNT, Win98 共通

DOS版

- デバイスドライバI/Fライブラリ・・・メモリモデル対応に4種類
(ラージ, コンパクト, ミディアム, スモール)

デバイスドライバとして、個々のOS毎に5種類あります。

- WinXP
- Win2K
- WinNT
- Win98
- DOS・・・割込み処理がサポートされます。

CPDボードに搭載されている日本パルスモータ社製コントローラ「PCL6045, 6025及び同等品
(以降これらの総称としてPCLと呼称します)」及びボードのオプションポートとの入出力を制御します。

(注)

1. デバイスドライバのインストール, アンインストールは<個別ボード編>に記載されています。
2. ボードのオプションポート詳細については、<個別ボード編>を参照して下さい。
3. 日本パルスモータ社製コントローラPCLの使用方法については<共通編>を参照して下さい。

(1) ライブラリファイル対応表

ボード種別		H P C I		H P C (※ 1)		H U S B (※ 2)	H C P C I (※ 3)	H P C 1 0 4 (※ 1)	H P 1 0 4 D (※ 1)	
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	cp36411a
		VC++ヘッダー 拡張子.h	cp53011a	cp52c11a	cp23411a	cp27811a	cp43011a	cp73011a	cp13211a	cp36411a
	VB標準モジュール 拡張子.bas (.NET の場合.vb)	cp53011a	cp52c11a	cp23411a	cp27811a	cp43011a	cp73011a	cp13211a	cp36411a	
	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	hcpd360
		VC++関数結合用ヘッダー 拡張子.h	hicpd530	hicpd52c	hcpd230		hucpd430	hccpd730	hcpd130	hcpd360
		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	hcpd360
		ドライバ I/F 用 DLL 拡張子.dll	hicpd530	hicpd52c	hcpd230		hucpd430	hccpd730	hcpd130	hcpd360
DOS 版	ライブラリ関数	ソース 拡張子.c	cp53011a	cp52c11a	cp23411a.c	cp27811a.c	Windows 版のみ	cp13011a	cp36011a	
		ヘッダ 拡張子.h	cp53011a	cp52c11a	——	——		cp13011a	cp36011a	
	ドライバ関数	ラージ 拡張子.lib	Licpd530	Licpd52c	Lcpd230	Lcpd270		Lcpd130	Lcpd360	
		コンパクト 拡張子.lib	Cicpd530	Cicpd52c	Ccpd230	Ccpd270		Ccpd130	Ccpd360	
		ミディアム 拡張子.lib	Micpd530	Micpd52c	Mcpd230	Mcpd270		Mcpd130	Mcpd360	
		スモール 拡張子.lib	Sicpd530	Sicpd52c	Scpd230	Scpd270		Scpd130	Scpd360	
		ヘッダー 拡張子.h	hicpd530 hcpdtype	hicpd52c hcpdtype	hcpd230	hcpd270		hcpd130	hcpd360	

表 1. 1-1 ライブラリファイル対応表

(2) 関数名規約

ボード種別		H P C I		H P C		H U S B	H C P C I	H P C 1 0 4	H P 1 0 4 D
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_yyy()	cp52c_yyy()	cp230_yyy()	cp230_yyy()	cp430_yyy()	cp730_yyy()	cp130_yyy()	cp360_()
	備考	Windows 版のみ							

表 1. 1-2 ライブラリファイル対応表

- ※1. H P C, H P C 1 0 4, H P 1 0 4 D の対応 OS は WinXP, Win2K, Win98, DOS です。
また、VB、NET サンプルは付属されていません。
- ※2. H U S B の対応 OS は WinVista, WinXP, Win2K, Win98 です。
- ※3. H C P C I の対応 OS は WinXP, Win2K, Win98 です。
また、VB、NET サンプルは付属されていません。
- ※4. I N T E L 互換の CPU を搭載したマシン用です。その他のプラットフォームには対応していません。
ライブラリ関数名の "xxx" 部分については、次ページの一覧表に記載されています。
ドライバ関数名の "yy" 部分については「3. 1 ドライバ関数の種類 (P66)」一覧表に記載されています。

以降、H P C I (CPD534, 532, 578, 508) のファイル名、関数名で説明します。

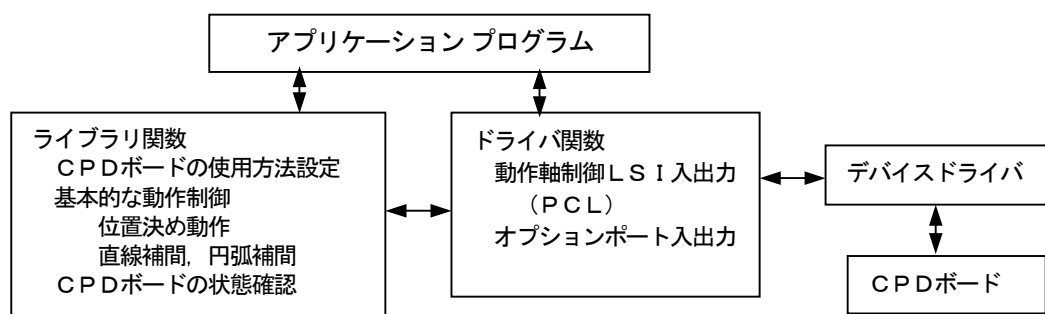
[ライブラリ関数とボード種別の対応表]

No	関数名		HPC1-CPD				HPC-CPD		HUSB-CPD	HPC1-CPD		HPC104-CPD	HP104D-CPD
	関数名 “xxx”		534, 532	578	508	5212M	234	278	434 (v2)	734	738	132	364
1	デバイス 関係	hcpxxx_GetDevInfo	○	○	○	○	×	×	○	○	○	×	×
2		hcpxxx_DevOpen	○	○	○	○	○	○	○	○	○	○	○
3		hcpxxx_DevClose	○	○	○	○	○	○	○	○	○	○	○
4	初期設定	hcpxxx_SetOrgMode	○	○	○	○	○	○	○	○	○	○	○
5		hcpxxx_SetEIs	○	○	○	○	○	○	○	○	○	○	○
6		hcpxxx_SetOIs	○	○	○	○	○	○	○	○	○	○	○
7		hcpxxx_SetSvAlm	○	○	○	○	○	○	○	○	○	○	○
8		hcpxxx_SetEz	○	○	○	○	○	○	○	○	○	○	○
9		hcpxxx_SetDisSel	○	○	×	×	○	○	○	○	○	○	×
10		hcpxxx_SetInpos	○	○	×	○	○	○	○	○	○	○	×
11		hcpxxx_SetSvCtrCl	○	○	×	○	○	○	○	○	○	○	×
12		hcpxxx_SetSIs	○	○	○	○	○	○	○	○	○	○	○
13		hcpxxx_SetCmdPulse	○	○	○	○	○	○	○	○	○	○	○
14		hcpxxx_SetAccProfile	○	○	○	○	○	○	○	○	○	○	○
15		hcpxxx_SetAutoDec	○	○	○	○	○	○	○	○	○	○	○
16		hcpxxx_mSetDis	×	×	×	×	×	×	×	×	×	×	○
17		hcpxxx_mSetLtc	×	×	×	×	×	×	×	×	×	×	○
18		hcpxxx_mSetInpos	×	×	×	×	×	×	×	×	×	×	○
19		hcpxxx_mSetSvCtrCl	×	×	×	×	×	×	×	×	×	×	○
20	状態読み出し	hcpxxx_ReadMainSts	○	○	○	○	○	○	○	○	○	○	○
21		hcpxxx_ReadErrorSts	○	○	○	○	○	○	○	○	○	○	○
22		hcpxxx_ReadEventSts	○	○	○	○	○	○	○	○	○	○	○
23		hcpxxx_ReadSubSts	○	○	○	○	○	○	○	○	○	○	○
24		hcpxxx_ReadExSts	○	○	○	○	○	○	○	○	○	○	○
25		hcpxxx_ReadIpSts	○	×	×	×	○	×	○	○	×	○	○
26		hcpxxx_ReadSpd	○	○	○	○	○	○	○	○	○	○	○
27		hcpxxx_ReadCtr	○	○	○	○	○	○	○	○	○	○	○
28	動作設定	hcpxxx_SetFLSpd	○	○	○	○	○	○	○	○	○	○	○
29		hcpxxx_SetAuxSpd	○	○	○	○	○	○	○	○	○	○	○
30		hcpxxx_SetAccRate	○	○	○	○	○	○	○	○	○	○	○
31		hcpxxx_SetDecRate	○	○	○	○	○	○	○	○	○	○	○
32		hcpxxx_SetMult	○	○	○	○	○	○	○	○	○	○	○
33		hcpxxx_SetEventMask	○	○	○	○	○	○	○	○	○	○	○
34		hcpxxx_SetDecPoint	○	○	○	○	○	○	○	○	○	○	○
35		hcpxxx_WritOpeMode	○	○	○	○	○	○	○	○	○	○	○
36	運用設定	hcpxxx_WritFHSpd	○	○	○	○	○	○	○	○	○	○	○
37		hcpxxx_WritPos	○	○	○	○	○	○	○	○	○	○	○
38		hcpxxx_WritLine	○	○	○	○	○	○	○	○	○	○	○
39		hcpxxx_WritCircl	○	○	○	○	○	○	○	○	○	○	○
40		hcpxxx_WritCtr	○	○	○	○	○	○	○	○	○	○	○
41	動作制御指令	hcpxxx_DecStop	○	○	○	○	○	○	○	○	○	○	○
42		hcpxxx_QuickStop	○	○	○	○	○	○	○	○	○	○	○
43		hcpxxx_EmgStop	○	○	○	○	○	○	○	○	○	○	○
44		hcpxxx_SyDecStop	×	○	○	○	×	○	×	×	○	×	×
45		hcpxxx_SyQuickStop	×	○	○	○	×	○	×	×	○	×	×
46		hcpxxx_AccStart	○	○	○	○	○	○	○	○	○	○	○
47		hcpxxx_CnstStartFH	○	○	○	○	○	○	○	○	○	○	○
48		hcpxxx_CnstStartFL	○	○	○	○	○	○	○	○	○	○	○
49		hcpxxx_CnstStartByDec	○	○	○	○	○	○	○	○	○	○	○
50		hcpxxx_SvOn	○	○	○	○	○	○	○	○	○	○	○
51		hcpxxx_SvOff	○	○	○	○	○	○	○	○	○	○	○
52		hcpxxx_SvResetOn	○	○	×	○	○	○	○	○	○	○	○
53		hcpxxx_SvResetOff	○	○	×	○	○	○	○	○	○	○	○
54		hcpxxx_PMOOn	○	○	○	○	○	○	○	○	○	○	○
55		hcpxxx_PMOOff	○	○	○	○	○	○	○	○	○	○	○
56	汎用 DIO	hcpxxx_mGetOptInp	×	×	×	×	×	×	×	×	×	×	○
57	加減速レート計算	hcpxxx_CalAccRate	○	○	○	○	○	○	○	○	○	○	○

表 1. 1-3 ライブラリ関数とボード種別の対応表

1. 1 概要

弊社の提供するソフトウェアは、ライブラリ関数、ドライバ関数、デバイスドライバの3種類であり、アプリケーションプログラムとのソフトウェアの関連は次図の通りです。



ライブラリ関数（レベル1）は、CPDボードの標準的な位置決め動作、及び直線・円弧補間の基本的な動作を制御することができます。

Windows版のVC++用ライブラリ関数のソースプログラムは「**cp53011a.c**」、VB用ライブラリ関数のソースコードは「**cp53011a.bas**」に記述されていますので、適宜変更追加することができます。VC++用ライブラリ関数とVB用ライブラリ関数の仕様は同様になっています。

DOS版のライブラリ関数はWindows版VC++用ライブラリ関数とほぼ同一の内容です。

ドライバ関数は「デバイスドライバI/Fライブラリ」であり、OSの種類で異なるデバイスドライバへのアクセスをドライバ関数を使用する事で、アプリケーションを共通化しています。

この為に、ドライバ関数はCPDボードとの入出力を行う基本的な関数集となっています。

このマニュアルではライブラリ関数およびドライバ関数の使用方法を説明します。

その為に、個々のボードの持つ詳細な機能については「ユーザーズマニュアル<個別ボード編>」を参照し動作軸制御の詳細な使用方法については「ユーザーズマニュアル<共通編>」を参照して下さい。

(1) モータコントロール L S I

CPDボードは日本パルスモータ社製モータコントロール L S I 「**PCL（4軸、2軸）**」を搭載しています。

主要な動作機能はこれらの L S I が行います。

制御軸が8軸のボード（HPC1-CPD578, HPC1-CPD508, HCPC1-CPD738, HPC-CPD278）については、この L S I が2個の組み合わせとなります。

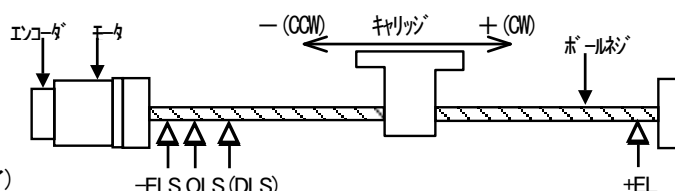
制御軸が12軸のボード（HPC1-CPD5212M）については、この L S I が3個の組み合わせとなります。

(2) 用語略称・詳細は「ユーザーズマニュアル<共通編>」を参照

このマニュアルで使用するモーション

コントロール関係の略語は次の通りです。

ELS：極限センサ
 OLS：原点センサ
 DLS：減速センサ
 SLS：ソフトリミット
 SVALM：サーボアラーム
 INPOS：インポジション（位置決め完了）
 SVCTRCL：偏差カウンタクリア出力
 ±DR：方向スイッチ信号
 LATCH：カウンタラッチ入力
 PCS：位置決め管理開始信号
 FH：動作速度
 FL：ベース速度
 FA：補助速度



(3) ボード種別とライブラリ関数の対応

前ページにボード種別とライブラリ関数の一覧表が記載されています。

ドライバ関数は「3. 1 ドライバ関数の種類（P66）」に記載され、全てのボードで使用可能です。

1. 2 準備

1. 2. 1 ライブラリ関数の使用

- (1) Visual C++ (5. 0以上) によるアプリケーションの構築
次のファイルをプロジェクトへ追加します。

■プロジェクト追加ファイル

- ◇ライブラリ関数用・・**cp530l1a.c** ・・VC++アプリケーション用ライブラリ関数ソースファイル
ライブラリ関数のコードがC言語で記述されています。
- ◇ドライバ関数用 ・・**hicpd530.lib** ・・ドライバ関数インポートライブラリ

■インクルードファイル

- ◇ライブラリ関数用・・**cp530l1a.h** ・・VC++アプリケーション用ライブラリ関数ヘッダーファイル
“cp530l1a.c” ファイルの中で “#include” されています。
- ◇ドライバ関数用 ・・**hicpd530.h** ・・ドライバ関数結合用ヘッダーファイル
“cp530l1a.h” ファイルの中で “#include” されています。

- (2) Visual Basic (5. 0/6. 0) によるアプリケーションの構築
次のファイルをプロジェクトへ追加します。

- ◇ライブラリ関数用・・**cp530l1a.bas** ・・VBアプリケーション用ライブラリ標準モジュールファイル
このファイルにライブラリ関数の関数定義と関数コードが記述されています。
- ◇ドライバ関数用 ・・**hicpd530.bas** ・・ドライバI/F用DLL関数定義標準モジュールファイル
このファイルに外部関数宣言 (Declare)、及びユーザー定義型宣言が記述されています。

- (3) DOS版：C言語によるアプリケーションの構築
次のファイルをプロジェクトへ追加します。

■プロジェクト追加ファイル

- ◇ライブラリ関数用・・**cp530l1a.c** ・・Cアプリケーション用ライブラリ関数ソースファイル
ライブラリ関数のコードがC言語で記述されています。
- ◇ドライバ関数用・・リンクファイルとして、次のメモリモデル対応ドライバ関数ライブラリ本体
 - Licpd530.lib** ・・ラージモデル [コード：64KB 以上、データ：64KB 以上]
 - Cicpd530.lib** ・・コンパクトモデル [コード：64KB 未満、データ：64KB 以上]
 - Micpd530.lib** ・・ミディアムモデル [コード：64KB 以上、データ：64KB 未満]
 - Sicpd530.lib** ・・スモールモデル [コード：64KB 未満、データ：64KB 未満]

■インクルードファイル

- ◇ライブラリ関数用・・**cp530l1a.h** ・・C言語アプリケーション用ライブラリ関数ヘッダーファイル
(CPD578ボードでは、ライブラリ関数ファイル内に統合)
- ◇ドライバ関数用 ・・**hicpd530.h** ・・ドライバ関数結合用ヘッダーファイル
hcupdtype.h ・・デバイス情報構造体定義ファイル
“hicpd530.h” ファイルの中で “#include” されています。

(注) 個々のボード毎に多少の変更がありますので、具体的な内容は各ボードの個別編を参照して下さい。

1. 2. 2 ドライバ関数の使用

- (1) Visual C++ (5. 0以上) によるアプリケーションの構築
次のファイルをプロジェクトへ追加します。

■プロジェクト追加ファイル

◇ドライバ関数用 ・・hicpd530.lib ・・ドライバ関数インポートライブラリ

■インクルードファイル

◇ドライバ関数用 ・・hicpd530.h ・・ドライバ関数結合用ヘッダーファイル

- (2) Visual Basic (5. 0/6. 0) によるアプリケーションの構築
次のファイルをプロジェクトへ追加します。

◇ドライバ関数用 ・・hicpd530.bas ・・ドライバI/F用DLL関数定義標準モジュールファイル
このファイルに外部関数宣言 (Declare) , 及びユーザー定義型宣言が記述されています。

- (3) DOS版: C言語によるアプリケーションの構築
DOS版では、ライブラリ関数, 及びドライバ関数は「C言語」で作成されています。
次のファイルをプロジェクトへ追加します。

■プロジェクト追加ファイル (※)

◇ドライバ関数用 ・・リンクファイルとして, 次のメモリモデル対応ドライバ関数ライブラリ本体

Licpd530.lib ・・ラージモデル [コード: 64KB 以上, データ: 64KB 以上]

Cicpd530.lib ・・コンパクトモデル [コード: 64KB 未満, データ: 64KB 以上]

Micpd530.lib ・・ミディアムモデル [コード: 64KB 以上, データ: 64KB 未満]

Sicpd530.lib ・・スモールモデル [コード: 64KB 未満, データ: 64KB 未満]

■インクルードファイル (※)

◇ドライバ関数用 ・・hicpd530.h ・・ドライバ関数結合用ヘッダーファイル

hcpdtype.h ・・デバイス情報構造体定義ファイル

“hicpd530.h” ファイルの中で “#include” されています。

※個々のボード毎に多少の変更はあります。(個別編を参照)

1. 2. 3 C++アプリケーションでの使用

Windows版では、ライブラリ関数, 及びドライバ関数は「C言語」で作成されています。
これらの関数をC++アプリケーションで使用できるように, ドライバ関数・ライブラリ関数の各ヘッダーファイル内で以下のように「ライブラリ関数・ドライバ関数はC言語」として明示的な定義をしています。

```
/*-----*/
//関数プロトタイプ宣言
//-----*/
#ifdef __cplusplus
extern "C"
{
#ifdef
/*-----*/
| 個々の関数のプロトタイプ宣言 |
/*-----*/
#endif
}
#endif
```

- (1) #ifdef __cplusplus ・・ Visual C++用の定義です
#endif C++コーディングでは明示的にライブラリ関数が” Cモジュール” として解釈されます。
- (2) extern "C" ・・ Cモジュールで定義されている関数を表します。

1. 3 関数の戻り値

ライブラリおよびドライバの諸関数を使用する時、関数の戻り値が異常値（'0'以外）であった場合には、異常内容に対応した処理を行います。

通常、この異常が発生した場合にはアプリケーションプログラムの続行は困難であり、プログラム内容の再検討が必要となります。

No	戻り値			異常内容と確認項目
	記号表記	16進数表記		
		VC++ DOS	VB VB.NET	
1	NO_ERROR	0x000	&H0	正常 異常は発生していません
2	NOT_FOUND	0x001	&H1	デバイスドライバが存在しない ◎デバイスドライバがインストールされていない ◎デバイスドライバが所定のフォルダに格納されていない
3	ALREADY_OPENED	0x002	&H2	既にオープン済みのデバイスをオープン ◎オープン済みデバイスに更にオープン指令 ◇オープンしたデバイスはクローズするまで使用 （多重のオープンは禁止） ◎ボード2枚以上使用する場合、オープンするデバイス情報の更新を確認します。
4	INSUFFICIENT_MEMORY	0x004	&H4	デバイス情報格納メモリが不足 ◎アプリケーション用のメモリ不足 ◇パソコン主記憶メモリの不足 ◎システムリソース（OS用メモリ）の不足 ◇多数のアプリケーション起動 ◇1度に多数のウィンドウを開いた
5	INVALID_HANDLE	0x008	&H8	無効なデバイスハンドルを指定 ◎デバイスオープンで得られた“デバイスハンドル”の不使用 ◎このデバイスは既にクローズされている
6	NOT_READY	0x010	&H10	デバイスの入出力ポートが使用できない ◎システムが不安定になっている可能性がありますので、 弊社サポートまでお問い合わせください
7	ILLEGAL_DEVICE	0x020	&H20	ボード固有情報が不正 ◎ポートの読み出しができない状態です。 弊社サポートまでお問い合わせください
8	ILLEGAL_ADDRESS	0x040	&H40	不正なベースアドレス（HPCボード） ◎指定したベースアドレスの確認
9	ILLEGAL_PARAM	0x100	&H100	関数の引数の値が異常 ◇速度倍率設定値の範囲は2～4095。 ◇その他引数の設定値を確認（マニュアル照合）

表 1. 3-1 関数の戻り値

なお、サーボ装置・メカセンサに起因する異常はこの異常報告に含まれません。

個々の要因毎に、異常発生内容を明確にすると共に、適切な処置が求められます。

■サーボ関係サーボアラーム (SVALM)、位置決め完了信号 (INPOS)

■メカセンサ極限センサ (ELS)、原点センサ (OLS)、減速センサ (DLS)、非常停止入力 (EMG)

2. ライブラリ関数

ライブラリ関数は、ソースプログラムで提供され、アプリケーションと同時にビルドします。

(1) ライブラリ関数一覧 (※運用度の記号説明◎：必ず使用, ○：通常使用)

No	関数名		機 能	運用度	記載頁		
					VC++	VB	DOS
1	デバイス 関係	hcxxxx_GetDevInfo	デバイス個数、デバイス情報の取得	◎	10	32	54
2		hcxxxx_DevOpen	デバイスのオープン、レジスタの初期化	◎	10	32	54
3		hcxxxx_DevClose	デバイスのクローズ	◎	11	33	54
4	初期設定	hcxxxx_SetOrgMode	原点復帰モードの設定	○	11	33	54
5		hcxxxx_SetEls	ELSの設定	○	13	35	54
6		hcxxxx_SetOls	OLSの設定	○	14	36	55
7		hcxxxx_SetSvAlm	SVALMの設定	○	14	36	55
8		hcxxxx_SetEz	エンコーダZ相の設定		14	36	55
9		hcxxxx_SetDisSel	DLS/PCS入力選択、設定		15	37	55
10		hcxxxx_SetInpos	INPOSの設定		15	37	55
11		hcxxxx_SetSvCtrCl	偏差カウンタクリア出力の設定		15	37	56
12		hcxxxx_SetSls	ソフトリミットの設定		16	38	56
13		hcxxxx_SetCmdPulse	指令パルスの出力形式選択		16	38	56
14		hcxxxx_SetAccProfile	S字/直線加減速の切替		17	39	56
15		hcxxxx_SetAutoDec	減速開始点の設定方式の自動/手動切替		17	39	56
16		hcxxxx_mSetDis	汎用入力/DLS/PCS/±DR入力選択、設定		18	40	57
17		hcxxxx_mSetLtc	汎用入力/LATCH入力選択、設定		18	40	57
18		hcxxxx_mSetInpos	INPOS入力/汎用入力選択、設定		19	41	57
19		hcxxxx_mSetSvCtrCl	SVCTRL出力/汎用出力選択、設定		19	41	57
20	状態読出し	hcxxxx_ReadMainSts	メインステータスの読出し	◎	20	42	58
21		hcxxxx_ReadErrorSts	エラーステータスの読出し	◎			58
22		hcxxxx_ReadEventSts	イベントステータスの読出し	◎			58
23		hcxxxx_ReadSubSts	サブステータスの読出し				58
24		hcxxxx_ReadExSts	拡張ステータスの読出し				58
25		hcxxxx_ReadIpSts	補間ステータスの読出し		20	42	59
26		hcxxxx_ReadSpd	現在速度の読出し	○	21	43	59
27		hcxxxx_ReadCtr	カウンタの読出し	○	21	43	59
28	動作設定	hcxxxx_SetFLSpd	ベース速度の設定	○	22	44	59
29		hcxxxx_SetAuxSpd	補助速度の設定				59
30		hcxxxx_SetAccRate	加速レートの設定	○			60
31		hcxxxx_SetDecRate	減速レートの設定		22	44	60
32		hcxxxx_SetMult	倍率設定値（MG）の設定	○	23	45	60
33		hcxxxx_SetEventMask	イベントマスクの設定		23	45	60
34	hcxxxx_SetDecPoint	減速開始点の設定		24	46	60	
35	運用設定	hcxxxx_WritOpeMode	動作モードの設定	◎	25	47	61
36		hcxxxx_WritFHSpd	動作速度の設定	○	26	48	61
37		hcxxxx_WritPos	移動量の設定	○	26	48	61
38		hcxxxx_WritLine	直線補間の移動量の設定		26	48	61
39		hcxxxx_WritCircl	円弧補間の移動量の設定		27	49	62
40		hcxxxx_WritCtr	カウンタプリセット		27	49	62
41	動作制御指令	hcxxxx_DecStop	減速停止	○	28	50	62
42		hcxxxx_QuickStop	即停止				62
43		hcxxxx_EmgStop	非常停止				63
44		hcxxxx_SyDecStop	同時減速停止		28	50	—
45		hcxxxx_SyQuickStop	同時即停止				—
46		hcxxxx_AccStart	加速スタート	○			63
47		hcxxxx_CnstStartFH	FH定速スタート		29	51	63
48		hcxxxx_CnstStartFL	FL定速スタート				63
49		hcxxxx_CnstStartByDec	FH定速スタート後減速停止				63
50		hcxxxx_SvOn	サーボオン		30	52	64
51		hcxxxx_SvOff	サーボオフ				64
52		hcxxxx_SvResetOn	サーボリセットオン				64
53		hcxxxx_SvResetOff	サーボリセットオフ				64
54		hcxxxx_PMOon	パルスモータ励磁オン				64
55		hcxxxx_PMOff	パルスモータ励磁オフ				65
56	汎用DIO	hcxxxx_mGetOptInp	オプション設定の汎用入力ポート確認		—	—	65
57	加減速レート計算	hcxxxx_CalAccRate	加減速レートの計算		31	53	65

表2. 1-1 ライブラリ関数

(2) ライブラリ関数説明のタイトル部

ライブラリ関数のタイトル部は次の記述となっています。

関数名		関数機能										運用度	
		※運用度の記号説明◎：必ず使用，○：通常使用											
(1) hcpxxx_GetDevInfo()		ボード枚数の取得，デバイス情報の取得										◎	
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	×	×	○	○	○	○	×	×
HPCI					HPC		HUSB		HCPCI		HPC104		HP104D
使用可能なボード種別													

(3) ボードの種類と使用可能軸の関係

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。

ボード種類	軸名称上段: CPD5212M以外 下段: CPD5212M											
	X	Y	Z	U	V	W	A	B	—	—	—	—
	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
HPCI-CPD532, HPC104-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

(4) 円弧補間動作と使用可能軸の関係

円弧補間は2軸を使用しますが, 下表に示す軸の組み合わせを使用して下さい。

ボード種類	軸組合せ上段: CPD5212M以外 下段: CPD5212M															
	XY	XZ	XU	YZ	YU	ZU	VW	VA	VB	WA	WB	AB	—	—	—	—
	X1-Y1	X1-Z1	X1-U1	Y1-Z1	Y1-U1	Z1-U1	X2-Y2	X2-Z2	X2-U2	Y2-Z2	Y2-U2	Z2-U2	X3-Y3	X3-Z3	X3-U3	Z3-U3
HPCI-CPD532, HPC104-CPD132	○	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
HPCI-CPD534, HPC-CPD234, HUSB-CPD434, HUSB-CPD434v2, HCPCI-CPD734, HP104D-CPD364	○	○	○	○	○	○	×	×	×	×	×	×	×	×	×	×
HPCI-CPD578, HPCI-CPD508, HCPCI-CPD738, HPC-CPD278	○	○	○	○	○	○	○	○	○	○	○	○	×	×	×	×
HPCI-CPD5212M	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

※ ○: 組み合わせ可, ×: 組み合わせ不可

(5) ライブラリ関数の戻り値

関数の起動を行った結果は「戻り値」に実行結果が反映されます。

この戻り値はライブラリ関数 (およびドライバ関数) で共通です。

戻り値が0の場合は正常終了ですが, '0'以外の場合には何らかの異常が発生しています。

「1. 3関数の戻り値 (P7)」を参照して下さい。

2. 1 VC++用ライブラリ関数

■デバイス関係

(1) hcpxxx_GetDevInfo() ボード枚数の取得, デバイス情報の取得													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	×	×	○	○	○	○	×	×
機能	現在パソコンに装着されているCPDボードの枚数,及びデバイス情報(※1)を取得します。												
書式	DWORD hcpxxx_GetDevInfo(DWORD* HpcDevNumber, HPCDEVICEINFO* HpcDevInfo);												
引数	◆DWORD* HpcDevNumber ・ ・ CPDボードの枚数 ◆HPCDEVICEINFO* HpcDevInfo ・ ・ CPDボードのデバイス情報格納の構造体												
呼出例	DWORD count; DWORD ret; HPCDEVICEINFO HpcDevInfo[0]; //ボードのデバイス情報が格納されているエリアの先頭アドレス ret = hcpxxx_GetDevInfo(&count, &HpcDevInfo[0]);												
備 考	※ 1. デバイス情報格納の構造体名及び構造体メンバは, ドライバの種類により異なる場合があります。詳細は「ユーザーズマニュアル<個別ボード編>」を参照して下さい。(以下同様)												

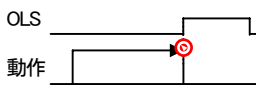
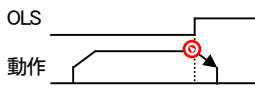

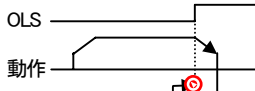
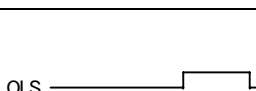
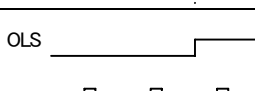
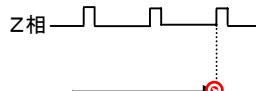
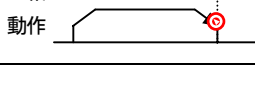

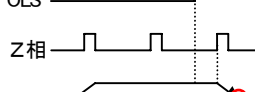
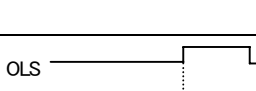
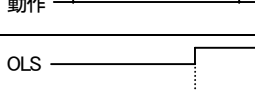
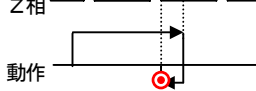
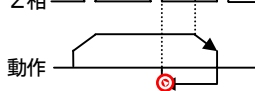
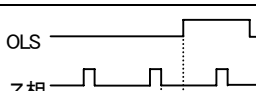
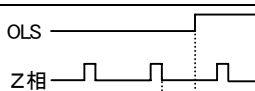
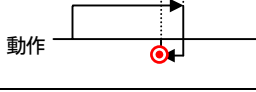
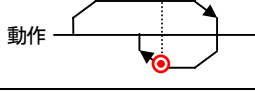
(2) hcpxxx_DevOpen() デバイスのオープン, レジスタとオプションポートの初期化													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	渡したデバイス情報を持つCPDボードをオープンし,他と識別するためのデバイスハンドルを取得します. 以降このデバイスハンドルは,このCPDボードにアクセスするために使用します. またオープンしたボードのレジスタとオプションポートの初期化をします.												
書式	DWORD hcpxxx_DevOpen(DWORD* hDevID, HPCDEVICEINFO* HpcDevInfo);												
引数	◆DWORD* hDevID ・ ・ デバイスハンドル ◆HPCDEVICEINFO* HpcDevInfo ・ ・ オープンするデバイスの情報がセットされたエリアのアドレス												
呼出例	パソコンにCPDボードが2枚装着されていることを想定します. デバイス情報格納エリアとしてHPCDEVICEINFO型の配列(※1) HpcDevInfo[2]を準備し,この中には既に hcpxxx_GetDevInfo 関数により全ボードのデバイス情報が入っているものとします. DWORD ret; //関数の戻り値 DWORD hDevID[2]; //デバイスハンドル取得エリア ret = hcpxxx_DevOpen(hDevID[0], &HpcDevInfo[0]); // 1 番目のデバイス情報 ret = hcpxxx_DevOpen(hDevID[1], &HpcDevInfo[1]); // 2 番目のデバイス情報												
備 考	[オプションポートの初期値 (※2)] ELS極性選択ポート : 0 x 0 0 (全軸B接) DLS／PCS切替ポート : 0 x f f (全軸PCS入力)												
	レジスタの初期値 (※2)												
	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: 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				CPD364			
						0 x 0 0 2 0 f d 5 5				CPD578, CPD278, CPD5212M			
						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	イベントマスク設定				0 x 0 0 0 0 1				正常停止時			
	本表以外のレジスタは初期値0 (レジスタの詳細はユーザーズマニュアル<共通編>を参照して下さい) ※1. ユーザーズマニュアル<個別ボード編>を参照 ※2. レジスタの初期値・オプションポートの初期値は関数内で直接与えています。そのため, この値を変更する場合はソースプログラム変更を行います.												

(3) hcpxxx_DevClose() デバイスのクローズ													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	渡したデバイスハンドルで指定されたC P Dボードをクローズします。 以降、このデバイスハンドルは無効となります。												
書式	DWORD hcpxxx_DevClose(DWORD hDevID);												
引数	◆DWORD hDevID ・ ・ クローズするボードのデバイスハンドル												
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_DevClose(hDevID);												
備 考	デバイスクローズの前にC P Dボードに対する終了処理を行ってください。												

■初期設定

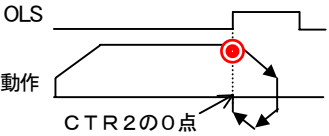
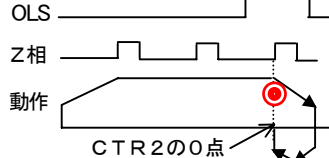
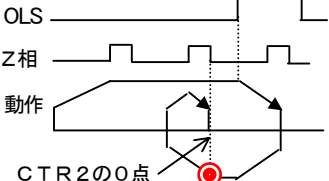
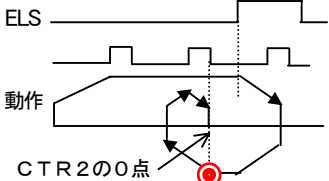
(4) hcpxxx_SetOrgMode() 原点復帰モードの設定														○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の原点復帰モードを設定します。													
書式	DWORD hcpxxx_SetOrgMode(DWORD hDevID, WORD axis, WORDmode);													
引数	◆DWORD hDevID ・ ・ 対象デバイスのデバイスハンドル ◆WORD axis ・ ・ 軸指定 ◆WORD mode ・ ・ 原点復帰モード[0:ORGmode0, 1:ORGmode1, ～, 12:ORGmode12](次頁表参照) hcpxxx_DevOpen()での mode の初期値は2 (原点復帰モード2, O L S onで減速, エンコーダZ相完了)													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
		引数	0	1	2	3	4	5	6	7	8	9	10	11
呼出例	DWORD ret; //関数の戻り値 //Y軸を指定, ORGmode1 (OLS 検出後拔出し再突入原点完了) ret = hcpxxx_SetOrgMode(hDevID, 1, 1);													

【 原点復帰モード (ORGmode) 】

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 (加速スタート)	
0	OLS原点 ORGmode=0		OLS検出で即停止 (完了)		OLS検出で減速停止 (完了)
1	OLS反転原点 ORGmode=1		OLS検出で即停止 補助速度で反転 OLS抜け出し後反転 OLS検出で即停止 (完了)		OLS検出で減速停止 補助速度で反転 OLS抜け出し後反転 OLS検出で即停止 (完了)
2	OLS+ENC原点 ORGmode=2		OLS検出後のn回目 のZ相で即停止 (完了)		OLS検出で減速開始 n回目のZ相で 即停止 (完了)
3	OLS+ENC原点 ORGmode=3				OLS検出後の n回目のZ相で 減速停止 (完了)
4	OLS反転+ ENC原点 ORGmode=4		OLS検出で即停止 補助速度で反転 n回目のZ相で 即停止 (完了)		OLS検出で減速停止 補助速度で反転 n回目のZ相で 即停止 (完了)
5	OLS反転+ ENC原点 ORGmode=5		OLS検出で即停止 ベ-ス速度で反転 n回目のZ相で 即停止 (完了)		OLS検出で減速停止 加速しながら反転 n回目のZ相で 減速停止 (完了)
6	ELS兼用センサ原点 ORGmode=6		ELS検出で即停止 補助速度で反転 ELS抜け出しで即停止 (完了)		ELS検出で停止 (停止方法はRBNW1に よる) 補助速度で反転 ELS抜け出しで即停止 (完了)
7	ELS反転+ ENC原点 ORGmode=7		ELS検出で即停止 補助速度で反転 n回目のZ相で 即停止 (完了)		ELS検出で停止 (停止方法はRBNW1に よる) 補助速度で反転 n回目のZ相で 即停止 (完了)
8	ELS反転+ ENC原点 ORGmode=8		ELS検出で即停止 ベ-ス速度で反転 n回目のZ相で 即停止 (完了)		ELS検出で停止 (停止方法はRBNW1に よる) 加速しながら反転 n回目のZ相で 減速停止 (完了)

(次頁へ続く)

(前頁からの続き)

No	原点復帰方法名	高速原点復帰 (加速スタート)	
9	OLS反転+ CTR2原点 ORGmode=9		OLS検出で減速。同時にCTR2クリア 停止後にCTR2の0点まで加減速動作。 ※ CTR2カウント入力は指令パルス(推奨) (REN3b9-b8=01)
10	ENC反転+ CTR2原点 ORGmode=10		OLS検出後のn回目Z相で減速。同時にCTR2クリア 停止後にCTR2の0点まで加減速動作。 ※ CTR2カウント入力はエンコーダ(推奨) (REN3b9-b8=00)
11	ENC反転+ CTR2原点 ORGmode=11		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CTR2の0点まで加減速動作。 ※ CTR2カウント入力はエンコーダ(推奨) (REN3b9-b8=00)
12	ELS反転+ CTR2原点 ORGmode=12		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CTR2の0点まで加減速動作。 ※ CTR2カウント入力はエンコーダ(推奨) (REN3b9-b8=00)

- 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。
 3. 最小限CTR2はクリアを行う設定が必要です。(REN3 bit21 = '1')
 4. HPCI-CPD508, HPCI-CPD5212Mはエンコーダ入力の使用に制限があります。

(5) hcpxxx_SetEls() ELSの設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
機能	○	○	○	○	○	○	○	○	○	○	○	○	○
書式	DWORD hcpxxx_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												
	◆WORD pol ・・極性 [0: B接, 1: A接] (hcpxxx_DevOpen()での初期値は0)												
軸指定	◆WORD stop ・・停止方法 [0: 即停止, 1: 減速停止] (hcpxxx_DevOpen()での初期値は0)												
	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
引数	0	1	2	3	4	5	6	7	8	9	10	11	
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetEls(hDevID, 1, 0, 0); //Y軸を指定, B接, 即停止												

(6) hcpxxx_SetOls() O L Sの設定													O
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のO L Sの入力極性を設定します。												
書式	DWORD hcpxxx_SetOls(DWORD hDevID, WORD axis, WORD pol);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD pol ・・極性 [0: B 接, 1: A 接] (hcpxxx_DevOpen() での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetOls(hDevID, 1, 0); //Y 軸を指定, B 接												

(7) hcpxxx_SetSvalm() S V A L Mの設定													O
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のS V A L Mの入力極性と入力停止方法を設定します。												
書式	DWORD hcpxxx_SetSvalm(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 ◆WORD pol ・・極性 [0: B 接, 1: A 接] (hcpxxx_DevOpen() での初期値は0) ◆WORD stop ・・停止方法 [0: 即停止, 1: 減速停止] (hcpxxx_DevOpen() での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetSvalm(hDevID, 1, 0, 0); //Y 軸を指定, B 接, 即停止												
備 考	HPC1-CPD508 の SVALM 信号は X-U, V-B にまとめられているため, X-U 軸, V-B 軸の入力極性は一致させる必要があります。												

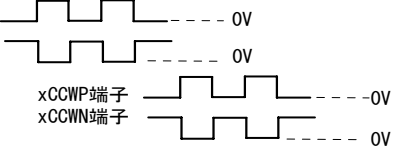
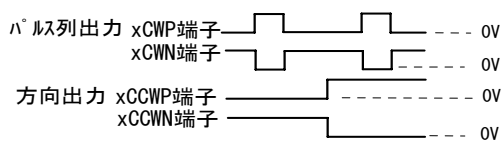
(8) hcpxxx_SetEz() エンコーダZ相の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のエンコーダZ相の入力処理を設定します。												
書式	DWORD hcpxxx_SetEz(DWORD hDevID, WORD axis, WORD zcount, WORD pol);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD zcount ・・原点復帰時のZ相回数[0: 1 回目, ..., 15: 16 回目] (hcpxxx_DevOpen() での初期値は0) ◆WORD pol ・・極性[0: 立下がりエッジ, 1: 立上がりエッジ] (hcpxxx_DevOpen() での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetEz(hDevID, 1, 2, 0); //Y 軸を指定, 3 回目, 立下がりエッジ												

(9) hcpxxx_SetDlsSel() DLS/PCSの選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	×	○	○	○	○	○	○	○	×
機能	デバイスハンドルで指定されたCPDボードの指定された軸のDLS/PCS入力及びその極性を選択, DLS入力時の動作を設定します.												
書式	DWORD hcpxxx_SetDlsSel(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 ◆WORD para ・・DLS/PCS入力の切替 [0: DLS入力, 1: PCS入力, 2: 両方共不使用] (hcpxxx_DevOpen()での初期値は1) ◆WORD pol ・・入力極性 [0: B接, 1: A接] (hcpxxx_DevOpen()での初期値は0) ◆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]												
	軸指定	軸名称	X	Y	Z	U	V	W	A	B	—	—	—
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetDlsSel(hDevID, 1, 0, 1, 0, 0); //Y軸, DLS使用, A接, 減速のみ, ラッチしない												

(10) hcpxxx_SetInpos() INPOSの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	○	○	○	○	○	○	○	○	×
機能	デバイスハンドルで指定されたCPDボードの指定された軸のINPOSの入力信号処理方法を設定します.												
書式	DWORD hcpxxx_SetInpos(DWORD hDevID, WORD axis, WORD enable, WORD pol);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD enable ・・使用/不使用 [0: 不使用, 1: 使用] (hcpxxx_DevOpen()での初期値は0) ◆WORD pol ・・極性 [0: B接, 1: A接] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetInpos(hDevID, 1, 1, 1); //Y軸を指定, INPOS使用, A接												

(11) hcpxxx_SetSvCtrCl() 偏差カウンタクリア出力の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	○	○	○	○	○	○	○	○	×
機能	デバイスハンドルで指定されたCPDボードの指定された軸の偏差カウンタクリア出力の設定をします.												
書式	DWORD hcpxxx_SetSvCtrCl(DWORD hDevID, WORD axis, WORD enable);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD enable ・・自動出力設定 [0: 不使用, 1: 原点完了時, 2: 異常停止時, 3: 原点完了及び異常停止時] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetSvCtrCl(hDevID, 1, 1); //Y軸を指定, 原点完了時出力												

(1 2) hcpxxx_SetSls() ソフトリミットの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のソフトリミット処理を設定します。												
書式	DWORD hcpxxx_SetSls(DWORD hDevID, WORD axis, long ps/s, long ms/s, WORD enable, 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 ◆long ps/s ・・+ S L S [パルス数] ◆long ms/s ・・- S L S [パルス数] ◆WORD enable ・・使用/不使用 [0:不使用, 1:使用] (hcpxxx_DevOpen() での初期値は0) ◆WORD stop ・・停止方法 [0:即停止, 1:減速停止]												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
	引数	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	DWORD ret; //関数の戻り値 //Y軸, +SLS = 100000, -SLS = -50000, ソフトリミット使用, 即停止 ret = hcpxxx_SetSls(hDevID, 1, 100000, -50000, 1, 0);												
備 考	1. ソフトリミット使用時は①ps/s は必ず ms/s より大きくして下さい。 2. ソフトリミット不使用時は①ms/s = ps/s = 0 とし, ②enable = 0 とします。 3. スタートコマンド書込み時にS L S がON状態の場合, S L S 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。												

(1 3) hcpxxx_SetCmdPulse() 指令パルスの出力形式の設定														○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の指令パルスの出力形式を設定します。													
書式	DWORD hcpxxx_SetCmdPulse(DWORD hDevID, WORD axis, WORD cmdpls);													
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD cmdpls ・・指令パルスの出力形式[0:個別指令方式, 1:共通指令方式](※1) (hcpxxx_DevOpen() での初期値は0)													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
	引数		0	1	2	3	4	5	6	7	8	9	10	11
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetCmdPulse(hDevID, 0, 1); //X軸を指定, 共通指令方式													
備考	※1. 指令パルスの出力形式													
	0 : 個別指令方式							1 : 共通指令方式						
	<div>xCWP端子</div> <div>xCWN端子</div> <div>xCCWP端子</div> <div>xCCWN端子</div> 							<div>パルス列出力</div> <div>xCWP端子</div> <div>xCWN端子</div> <div>方向出力</div> <div>xCCWP端子</div> <div>xCCWN端子</div> 						

(1 4) hcpxxx_SetAccProfile() S字/直線加減速の切替														
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のS字／直線加減速の設定をします。													
書式	DWORD hcpxxx_SetAccProfile(DWORD hDevID, WORD axis, WORD pr);													
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD pr ・・加減速形式 [0:直線, 1: S字] (hcpxxx_DevOpen() での初期値は0)													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
		引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetAccProfile(hDevID, 1, 1); //Y軸を指定, S字加減速に設定													

(1 5) hcpxxx_SetAutoDec() 減速開始点計算方式の自動計算/手動計算切替													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の減速開始点計算方式を手動計算か自動計算か設定します。												
書式	DWORD hcpxxx_SetAutoDec(DWORD hDevID, WORD axis, WORD para);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD para ・・減速開始点の設定方式 [0:自動計算設定, 1:手動計算設定] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetAutoDec(hDevID, 1, 1); //Y軸の減速開始点の設定を手動計算設定												

(1 6) hcpxxx_mSetDls()													汎用入力／±DR／PCS／DLS入力選択, 設定		O
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364		
	x	x	x	x	x	x	x	x	x	x	x	x	O		
機能	デバイスハンドルで指定されたC P Dボードの汎用入力/±DR/PCS/DLS 入力を選択し, ±DR, PCS, DLS の極性, DLS 選択時に DLS 入力時の動作を設定します.														
書式	DWORD hcpxxx_mSetDls(DWORD hDev, WORD para, WORD pol, WORD motion, WORD latch);														
引数	◆DWORD hDev ・ ・ 対象デバイスのデバイスハンドル ◆WORD para ・ ・ 汎用入力/±DR/PCS/DLS 入力選択 (詳細は個別編参照) ◆WORD pol ・ ・ 入力極性[0x001:X DLS=A 接,0x002:Y DLS=A 接,0x004:Z DLS=A 接,0x008:U DLS=A 接,0x010:XPCS=A 接,0x020:YPCS=A 接,0x040:ZPCS=A 接,0x080:UPCS=A 接,0x100:XDR =A 接,0x200:YDR =A 接,0x400:ZDR =A 接] O R したデータを設定(hcpxxx_DevOpen()での初期値はO : B接) para で入力選択されていない部分の設定は無効です. (B 接になります) ※汎用入力はA 接固定です. ◆WORD motion ・ ・ D L S 入力時の動作 [0x01:X 軸減速停止,0x02:Y 軸減速停止,0x04:Z 軸減速停止,0x08:U 軸減速停止] O R したデータを設定(hcpxxx_DevOpen()での初期値はO : 各軸減速のみ) ◆WORD latch ・ ・ D L S 入力のラッチ [0x01:X 軸ラッチ,0x02:Y 軸ラッチ,0x04:Z 軸ラッチ,0x08:U 軸ラッチ] O R したデータを設定(hcpxxx_DevOpen()での初期値はO : 各軸ラッチしない)														
	para					ビット / 機能 対応									
						7	6	5	4	3	2	1	0		
						US1	US0	ZS1	ZS0	YS1	YS0	XS1	XS0		
	J 4 コネクタ入力信号ピン番号					9 8 (9 9) ピン		9 7 ピン		4 8 (4 9) ピン		4 7 ピン			
	設 定 値				xS1 xS0	U 軸 : US1, US0		Z 軸 : ZS1, ZS0		Y 軸 : YS1, YS0		X 軸 : XS1, XS0			
	用 途 区 分	汎用入力信号 (初期値)				0 0	I N 3 0		I N 2 9		I N 2 6		I N 2 5		
		P C L 外部 J O G 起動信号				0 1	-Z D R (-Y D R)		+Z D R		-X D R (+Y D R)		+X D R		
		P C L 位置決め管理開始信号				1 0	U P C S		Z P C S		Y P C S		X P C S		
		P C L 減速センサ				1 1	U D L S		Z D L S		Y D L S		X D L S		
呼出例	DWORD ret; //関数の戻り値 //X 軸±DR 使用, Z, U 軸は DLS 使用, 減速のみ, DLS 入力ラッチ ret = hcpxxx_mSetDls(hDevID, 0xf3, 0, 0x0, 0xc);														

(18) hcpxxx_mSetInpos()													
I N P O S 入力／汎用入力の選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	x	x	x	x	x	x	x	x	x	x	x	x	○
機能	デバイスハンドルで指定された C P D ボードの指定された軸の汎用入力 / I N P O S 入力及びその極性を選択します。												
書式	DWORD hcpxxx_mSetInpos(DWORD hDevID, WORD axis, WORD para, WORD pol);												
引数	<p>◆DWORD hDevID ・・対象デバイスのデバイスハンドル</p> <p>◆WORD axis ・・軸指定 [0:X 軸, 1:Y 軸, 2:Z 軸, 3:U 軸]</p> <p>◆WORD para ・・INPOS 入力/汎用入力の切替 [0: I N P O S, 1: 汎用入力] (hcpxxx_DevOpen() での初期値は 0)</p> <p>◆WORD pol ・・INPOS 入力極性 [0: B 接, 1: A 接] (hcpxxx_DevOpen() での初期値は 0) ※汎用入力使用時は A 接固定です。</p>												
呼出例	<pre> DWORD ret; //関数の戻り値 ret = hcpxxx_mSetInpos(hDevID, 1, 0, 1); //Y 軸, INPOS 使用, A 接 </pre>												

(1 9) hcpxxx_mSetSvCtrl ()														汎用出力／SVCTRCL出力選択, 設定				
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364					
	x	x	x	x	x	x	x	x	x	x	x	x	○					
機能	デバイスハンドルで指定されたCPDボードの指定された軸の SVCTRCL 出力または汎用出力を選択します。 SVCTRCL 出力選択時の自動出力設定をします。																	
書式	DWORD hcpxxx_mSetSvCtrl(DWORD hDevID, WORD axis, WORD para, WORD enable);																	
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 ・ ・ SVCTRCL 出力/汎用出力の切替 [0 : SVCTRCL 出力, 1 : 汎用出力] (hcpxxx_DevOpen())での初期値は0) ◆WORD enable ・ ・ 自動出力設定 [0:不使用, 1:原点完了時, 2:異常停止時, 3:原点完了及び異常停止時] (hcpxxx_DevOpen())での初期値は0)																	
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_mSetSvCtrl(hDevID, 2, 0, 1); //Z 軸, SVCTRCL 原点復帰完了時自動出力																	

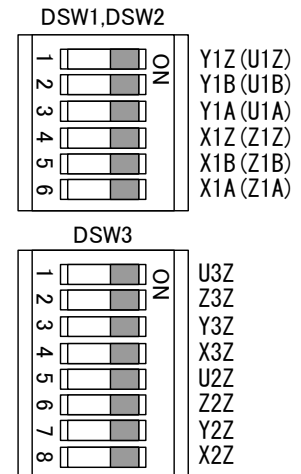
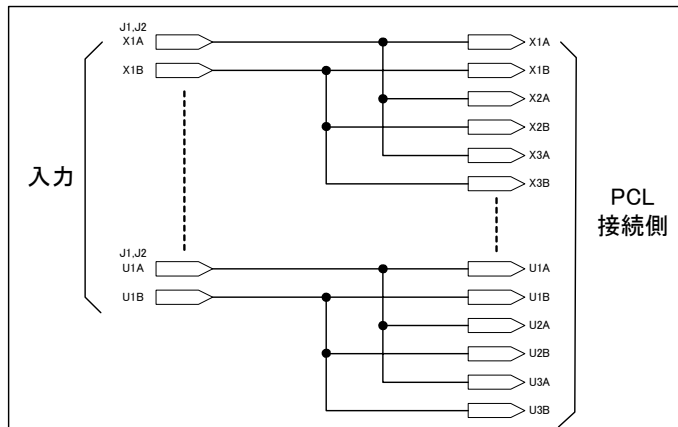
■状態読出し

(2 0) hcpxxx_ReadMainSts ()		メインステータスの読出し												◎
(2 1) hcpxxx_ReadErrorSts ()		エラーステータスの読出し												◎
(2 2) hcpxxx_ReadEventSts ()		イベントステータスの読出し												◎
(2 3) hcpxxx_ReadSubSts ()		サブステータスの読出し												
(2 4) hcpxxx_ReadExSts ()		拡張ステータスの読出し												
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定された C P D ボードの指定された軸のメインステータス（エラーステータス、イベントステータス、サブステータス、拡張ステータス）を読出し、指定したエリアに格納します。ステータスデータについては、ユーザーズマニュアル<共通編>を参照して下さい。													
書式	DWORD hcpxxx_ReadMainSts (DWORD hDevID, WORD axis, WORD* status); DWORD hcpxxx_ReadErrorSts (DWORD hDevID, WORD axis, DWORD* status); DWORD hcpxxx_ReadEventSts (DWORD hDevID, WORD axis, DWORD* status); DWORD hcpxxx_ReadSubSts (DWORD hDevID, WORD axis, WORD* status); DWORD hcpxxx_ReadExSts (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 ◆WORD* status ・ ・ メイン・サブステータスが格納されるエリアのアドレス ◆DWORD* status ・ ・ エラー・イベント・拡張ステータスが格納されるエリアのアドレス													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
		引数	0	1	2	3	4	5	6	7	8	9	10	11
呼出例	DWORD ret; //関数の戻り値 WORD ssts; //サブステータス ret = hcpxxx_ReadSubSts(hDevID, 1, &ssts); // Y 軸を指定、格納先のアドレス													

(2 5) hcpxxx_ReadIpSts()												補間ステータスの読出し		
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	×	×	×	○	×	○	○	○	×	○	○	
機能	デバイスハンドルで指定されたC P Dボードの補間ステータスを読出し、指定したエリアに格納します。ステータスデータについては、ユーザーズマニュアル<共通編>を参照して下さい。													
書式	DWORD hcpxxx_ReadIpSts(DWORD hDevID, DWORD* status);													
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆DWORD* status ・・読出されたデータが格納されるエリアのアドレス													
呼出例	DWORD ret; //関数の戻り値 DWORD ipsts; //補間ステータス ret = hcpxxx_ReadIpSts(hDevID, &ipsts); //格納先のアドレス													

(2 6) hcpxxx_ReadSpd ()														現在速度の読出														
対応ボード	532		534		578		508		5212M		234		278		434		434v2		734		738		132		364			
	○		○		○		○		○		○		○		○		○		○		○		○		○			
機能	デバイスハンドルで指定された C P D ボードの指定された軸の現在速度を読出し、指定したエリアに格納します。補間動作中は補間代表軸を読み出してください。																											
書式	DWORD hcpxxx_ReadSpd(DWORD hDevID, WORD* axis, WORD* 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																											
	◆WORD* speed ・ ・ 読出されたデータが格納されるエリアのアドレス																											
	軸指定	CPD5212M 以外軸名称		X	Y	Z	U	V	W	A	B	—	—	—	—													
CPD5212M 軸名称		X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3															
引数		0	1	2	3	4	5	6	7	8	9	10	11															
呼出例	DWORD ret; //関数の戻り値 WORD speed; //速度 ret = hcpxxx_ReadSpd(hDevID, 0, &speed); //X軸を指定、格納先のアドレス																											

(2 7) hcpxxx_ReadCtr () カウンタの読出													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の、指定されたカウンタを讀出し、指定したエリアに格納します。												
書式	DWORD hcpxxx_ReadCtr(DWORD hDevID, WORD* axis, WORD selctr, long* count);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆WORD selctr ・・カウンタ選択[1:カウンタ 1, 2:カウンタ 2, 3:カウンタ 3, 4:カウンタ 4] カウンタ 1, カウンタ 4 ・・指令パルス出力 カウンタ 2 ・・エンコーダ入力 カウンタ 3 ・・偏差カウンタ (脱調検出用) ◆long* count ・・読出されたデータが格納されるエリアのアドレス												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X 1	Y 1	Z 1	U 1	X 2	Y 2	Z 2	U 2	X 3	Y 3	Z 3
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	DWORD ret; //関数の戻り値 long* count; //カウンタ 1 (指令パルス出力) の値 // X 軸を指定, カウンタ 1 を指定, 格納先アドレス ret = hcpxxx_ReadCtr(hDevID, 0, 1, &count);												
備 考	HPCI-CPD508 はカウンタ 2, カウンタ 3 の指定はできません。 HPCI-CPD5212M は X1 軸～U1 軸のエンコーダ入力、X2 軸～U2 軸, X3 軸～U3 軸にも接続されています。詳しくは個別ボード編を参照してください。 [HPCI-CPD5212M エンコーダ入力A/B相接続]												



■動作設定

(28) hcpxxx_SetFLSpd() ベース速度の設定													○
(29) hcpxxx_SetAuxSpd() 補助速度の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の ベース速度・・・高速動作時の立ち上がりの速度（本速度から加速，本速度まで減速して停止） 補助速度・・・一部の原点復帰において，原点突入速度等に使用を設定します。 指令速度（pps）を速度倍率で除算した値を設定。												
書式	DWORD hcpxxx_SetFLSpd (DWORD hDevID, WORD axis, DWORD speed); DWORD hcpxxx_SetAuxSpd(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 ◆DWORD speed ・・ベース速度 (FL), 補助速度 (FA) レジスタ値 [1~65535, FL (FA) < FH (動作速度)] (hcpxxx_DevOpen() での初期値は 200)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetFLSpd(hDevID, 1, 500); //Y軸を指定，F L (ベース速度)=500												

(30) hcpxxx_SetAccRate() 加速レートの設定													○
(31) hcpxxx_SetDecRate() 減速レートの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の加速レート（減速レート）を設定します。												
書式	DWORD hcpxxx_SetAccRate(DWORD hDevID, WORD axis, DWORD rate); DWORD hcpxxx_SetDecRate(DWORD hDevID, WORD axis, DWORD rate);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆DWORD rate ・・加速レート [1~65535], 減速レート [0~65535] (※1) (hcpxxx_DevOpen() での初期値は 1364. FH=2000, FL=200, 直線加減速で加減速時間約 0.5 秒)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetAccRate(hDevID, 1, 500); //Y軸を指定，加速レートレジスタ=500												
備考	※1. 加速レート（減速レート）と加減速時間の関係 FH, FL (RFH, RFL レジスタ値), RUR (加速／減速レートレジスタ値) (1) 直線加速 加速時間[sec] = $\frac{(RFH - RFL) \times (RUR + 1) \times 4}{19,660,800}$ (2) 直線部分のないS字加速 加速時間[sec] = $\frac{(RFH - PRFL) \times (RUR + 1) \times 8}{19,660,800}$ 減速レート = 0 の時は減速レート = 加速レートとなります。												

(32) hcpxxx_SetMult() 速度倍率設定値 (MG) の設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の速度倍率設定値 (MG)(※1)を設定します。												
書式	DWORD hcpxxx_SetMult(DWORD hDevID, WORD axis, DWORD rmg);												
引数	◆DWORD hDevID ・・対象デバイスのデバイスハンドル ◆WORD axis ・・軸指定 ◆DWORD rmg ・・倍率設定値[2~4095] (hcpxxx_DevOpen()での初期値は初期値は299:速度倍率1倍)												
	軸指定	CPD5212M以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetMult(hDevID, 1, 299); //Y軸を指定, 速度倍率1倍												
備考	※1. 速度と速度倍率の関係及び速度倍率設定値と速度倍率の関係 RFx は速度レジスタ(RFH, RFL, RFA)の値 $\text{速度[PPS]} = RFx \times \text{速度倍率} = \frac{RFx \times 300}{RMG + 1} \quad \text{速度倍率設定値(MG)} = \frac{300}{\text{速度倍率}} - 1$												
	速度倍率設定例												
	設定値	倍率	出力速度範囲(pps)				設定値	倍率	出力速度範囲(pps)				
	2999 (0x0bb7)	0.1	0.1 ~ 6,553.5				59 (0x003b)	5	5 ~ 327,675				
	1499 (0x05db)	0.2	0.2 ~ 13,107				29 (0x001d)	10	10 ~ 655,350				
	599 (0x0257)	0.5	0.5 ~ 32,767.5				14 (0x000e)	20	20 ~ 1,310,700				
	299 (0x012b)	1	1 ~ 65,535				5 (0x0005)	50	50 ~ 3,276,750				
	149 (0x0095)	2	2 ~ 131,070				2 (0x0002)	100	100 ~ 6,553,500				

(3 3) hcpxxx_SetEventMask () イベントマスクの設定														
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のイベントマスクを設定します。													
書式	DWORD hcpxxx_SetEventMask(DWORD hDevID, WORD axis, DWORD mask);													
引数	◆DWORD hDevID ・ ・ 対象デバイスのデバイスハンドル													
	◆WORD axis ・ ・ 軸指定													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X 1	Y 1	Z 1	U 1	X 2	Y 2	Z 2	U 2	X 3	Y 3	Z 3	U 3
		引数	0	1	2	3	4	5	6	7	8	9	1 0	1 1
	◆DWORD mask ・ ・ イベントマスクデータ													
	下表の値をORしたデータを与えることで2種類以上のイベント報告指定となり ます。 (hcpxxx_DevOpen())での初期値は 0 x 00001 : 正常停止時)													
	mask (HEX)	イベントステータス報告のイベント						mask (HEX)	イベントステータス報告のイベント					
	0x00001	正常停止時						0x00400	コンパレータ 3 条件成立時					
	0x00002	次動作継続スタート時						0x00800	コンパレータ 4 条件成立時					
	0x00004	動作用プリレジスタフルから空気ができた時						0x01000	コンパレータ 5 条件成立時					
	0x00008	コンパレータ 5 用プリレジスタフルから空気ができた時						0x02000	CLR 入力によるカウント値のクリア時					
	0x00010	加速開始時						0x04000	LATCH 入力によるカウント値のラッチ時					
	0x00020	加速終了時						0x08000	OLS 入力によるカウント値のラッチ時					
	0x00040	減速開始時						0x10000	DLS 入力ON時					
	0x00080	減速終了時						0x20000	±DR 入力変化時					
0x00100	コンパレータ 1 条件成立時						0x40000	CSTA 信号入力 ON 時						
0x00200	コンパレータ 2 条件成立時													
呼出例	DWORD ret; //関数の戻り値 // Y 軸, 正常停止, コンパレータ 5 条件成立時にイベント報告 ret = hcpxxx_SetEventMask(hDevID, 1, 0x1001);													

(3 4) hcpxxx_SetDecPoint () 減速開始点の設定														
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の減速開始点を手動計算設定とした時の減速開始点を設定します。													
書式	DWORD hcpxxx_SetDecPoint (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 ◆long dstnc ・・減速開始点 (hcpxxx_DevOpen () での初期値は 0)													
	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—	
	CPD5212M 軸名称	X 1	Y 1	Z 1	U 1	X 2	Y 2	Z 2	U 2	X 3	Y 3	Z 3	U 3	
	引数	0	1	2	3	4	5	6	7	8	9	1 0	1 1	
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SetDecPoint (hDevID, 1, 300); // Y 軸を指定, 減速開始点													
備 考	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。													
	減速開始点の最適値は次式のようになります。													
	■直線減速	$\text{最適値[パルス]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 32,768}$												
	■直線部分のないS字減速	$\text{最適値[パルス]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1) \times 2}{(\text{RMG} + 1) \times 32,768}$												

■運用設定

(35) hcpxxx_WritOpeMode() 動作モードの設定													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の動作モードを設定します。												
書式	DWORD hcpxxx_WritOpeMode(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												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
	◆DWORD mask ・・動作モード (hcpxxx_DevOpen()での初期値は0 : +方向連続動作)												
	mode (Hex)	動作モード							備考			記事	
	0 x 0 0	コマンド制御による+方向連続動作											
	0 x 0 8	コマンド制御による一方向連続動作											
	0 x 0 1	パルス入力による連続動作										※ 1	
	0 x 0 2	±D R入力による連続動作										※ 2	
	0 x 1 0	+方向原点復帰動作											
	0 x 1 8	一方向原点復帰動作											
	0 x 1 2	+方向原点抜け出し動作											
	0 x 1 a	一方向原点抜け出し動作											
	0 x 1 5	+方向原点サーチ動作							hcpxxx_WritPos () 関数で引出量を設定				
	0 x 1 d	一方向原点サーチ動作											
	0 x 2 0	+E L又は+S L位置まで動作											
	0 x 2 8	－E L又はS L位置まで動作											
	0 x 2 2	+E L又は+S L抜け出し動作											
	0 x 2 a	－E L又はS L抜け出し動作											
	0 x 2 4	+方向にE Zカウント分だけ動作											
	0 x 2 c	一方向にE Zカウント分だけ動作											
	0 x 4 1	位置決め動作							hcpxxx_WritPos()関数で移動量を設定				
	0 x 4 2	P C S位置決め動作 (ライブラリ関数作成動作)											
	0 x 4 4	指令位置0点復帰動作											
	0 x 4 5	機械位置0点復帰動作										※ 1	
	0 x 4 6	+方向1パルス動作											
	0 x 4 e	一方向1パルス動作											
	0 x 4 7	タイマー動作							hcpxxx_WritPos()関数で移動量を設定				
	0 x 5 1	パルス入力による位置決め動作										※ 1	
	0 x 5 4	パルス入力による指令位置0点復帰動作										※ 1	
	0 x 5 5	パルス入力による機械位置0点復帰動作										※ 1	
	0 x 5 6	±D R入力による位置決め動作							hcpxxx_WritPos()関数で移動量を設定			※ 2	
	0 x 6 0	連続直線補間動作											
	0 x 6 1	直線補間動作							hcpxxx_WritLine()関数で移動量を設定				
	0 x 6 4	C W方向円弧補間動作 (時計回り)							hcpxxx_WritCirc()関数で移動量を設定				
	0 x 6 5	C C W方向円弧補間動作 (反時計回り)											
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_WritOpeMode(hDevID, 1, 0x41); //Y軸を指定, 位置決め動作												
備 考	※ 1. HPCI-CPD508 のボードでは,これらの動作はできません。 ※ 2. HP104D-CPD364 のみの機能です。												

(3 6) hcpxxx_WritFHSpd() 動作速度の設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の動作速度レジスタ（速度を倍率で除算した値）を設定します。												
書式	DWORD hcpxxx_WritFHSpd(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 ◆DWORD speed ・・F H（動作速度）レジスタ値[1~65535] (hcpxxx_DevOpen()での初期値は2000)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_WritFHSpd(hDevID, 1, 5000); //Y軸を指定, F H=5000												

(3 7) hcpxxx_WritPos() 位置決め移動量の設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の位置決め動作の移動量を設定します。												
書式	DWORD hcpxxx_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 ◆long dstnc ・・移動量[-134, 217, 728~134, 217, 727(28ビット)] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_WritPos(hDevID, 1, 15000); //Y軸を指定, +方向15000パルス												

(3 8) hcpxxx_WritLine() 直線補間の移動量の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の直線補間の移動量を設定します。												
書式	DWORD hcpxxx_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 ◆long dstnc ・・移動量[-134, 217, 728~134, 217, 727(28ビット)] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_WritLine(hDevID, 0, -30000); //補間代表軸: X軸を指定, 一方向30000パルス ret = hcpxxx_WritLine(hDevID, 1, -15000); //従軸: Y軸を指定, 一方向15000パルス												

(39) hcpxxx_WritCircI()																		
円弧補間の移動量の設定																		
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364					
	○	○	○	○	○	○	○	○	○	○	○	○	○					
機能	デバイスハンドルで指定されたCPDボードの指定された軸の円弧補間の移動量を設定します。 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。 データの設定方法については「ユーザーズマニュアル<共通編>」を参照して下さい。																	
書式	DWORD hcpxxx_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 ・ ・ 軸指定 ◆long dstnc1 ・ ・ 終点位置1 [-134, 217, 728~134, 217, 727(28ビット)] ◆long dstnc2 ・ ・ 終点位置2 [-134, 217, 728~134, 217, 727(28ビット)] ◆long center1 ・ ・ 中心位置1 [-134, 217, 728~134, 217, 727(28ビット)] ◆long center2 ・ ・ 中心位置2 [-134, 217, 728~134, 217, 727(28ビット)] (hcpxxx_DevOpen()での位置データの初期値は0)																	
	軸指定	CPD5212M 以外軸名称	X-Y	X-Z	X-U	Y-Z	Y-U	Z-U	V-W	V-A	V-B	W-A	W-B	A-B	—	—	—	—
		CPD5212M 軸名称	X1-Y1	X1-Z1	X1-U1	Y1-Z1	Y1-U1	Z1-U1	X2-Y2	X2-Z2	X2-U2	Y2-Z2	Y2-U2	Z2-U2	X3-Y3	X3-Z3	X3-U3	Y3-Z3
	引数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_WritCircI(hDevID, 0, 0, 0, 1000, 0); //XY軸円弧(半径1000パルスの真円) // 終点(X,Y)=(0,0), 中心(X,Y)=(1000,0)																	

(40) hcpxxx_WritCtr()													
カウンタプリセット													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の指定されたカウンタをプリセットします。												
書式	DWORD hcpxxx_WritCtr(DWORD <i>hDevID</i> , WORD <i>axis</i> , long <i>preset</i> , WORD <i>selctr</i>);												
引数	<p>◆DWORD <i>hDevID</i> ・・対象デバイスのデバイスハンドル</p> <p>◆WORD <i>axis</i> ・・軸指定</p> <p>◆long <i>preset</i> ・・プリセット値[-134, 217, 728~134, 217, 727(28ビット)]</p> <p>◆WORD <i>selctr</i> ・・カウンタ選択[1:カウンタ1, 2:カウンタ2, 3:カウンタ3, 4:カウンタ4]</p> <p style="padding-left: 150px;">カウンタ1, カウンタ4 ・・指令パルス出力</p> <p style="padding-left: 150px;">カウンタ2 ・・エンコーダ入力</p> <p style="padding-left: 150px;">カウンタ3 ・・偏差カウンタ(脱調検出用)</p> <p style="text-align: center;">(hcpxxx_DevOpen())での初期値は0)</p>												
軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
	引数	0	1	2	3	4	5	6	7	8	9	10	11
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_WritCtr(hDevID, 0, 5000, 1); //X軸, プリセット値=5000, カウンタ1												

■動作制御指令

(41) hcpxxx_DecStop()	減速停止	○
(42) hcpxxx_QuickStop()	即停止	
(43) hcpxxx_EmgStop()	非常停止	
対応ボード	532 534 578 508 5212M 234 278 434 434v2 734 738 132 364	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸を減速停止(即停止,非常停止)します。	
書式	DWORD hcpxxx_DecStop (DWORD hDevID, WORD axis); DWORD hcpxxx_QuickStop(DWORD hDevID, WORD axis); DWORD hcpxxx_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・・・軸指定はORしたデータになります。	
	軸指定	CPD5212M以外軸名称 X Y Z U V W A B — — — — CPD5212M軸名称 X1 Y1 Z1 U1 X2 Y2 Z2 U2 X3 Y3 Z3 U3 引数 (Hex) 0x001 0x002 0x004 0x008 0x010 0x020 0x040 0x080 0x100 0x200 0x400 0x800
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_EmgStop(hDevID, 0x07); //X, Y, Z軸非常停止	
備考	5軸以上(HPCI-CPD578/508, 5212M, HPCI-CPD738)で2つのモータコントロールL S Iにまたがる軸を指定した場合、停止するタイミングは異なります。それ以外は同時に停止します。	

(44) hcpxxx_SyDecStop()	同時減速停止	
(45) hcpxxx_SyQuickStop()	同時即停止	
対応ボード	532 534 578 508 5212M 234 278 434 434v2 734 738 132 364	x x ○ ○ ○ x ○ x x x ○ x x
機能	デバイスハンドルで指定されたC P Dボードの指定された軸を同時に減速停止(即停止)します。	
書式	DWORD hcpxxx_SyDecStop (DWORD hDevID, WORD axis); DWORD hcpxxx_SyQuickStop(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・・・軸指定はORしたデータになります。	
	軸指定	CPD5212M以外軸名称 X Y Z U V W A B — — — — CPD5212M軸名称 X1 Y1 Z1 U1 X2 Y2 Z2 U2 X3 Y3 Z3 U3 引数 (Hex) 0x001 0x002 0x004 0x008 0x010 0x020 0x040 0x080 0x100 0x200 0x400 0x800
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SyDecStop(hDevID, 0x07); //X, Y, Z軸同時減速停止	
備考	この関数の実行完了時にエラーステータスのビット8（同時停止入力による停止）が立ちますので、関数実行完了後に、指定した軸のエラーステータスレジスタを読み込む必要があります。	

(46) hcpxxx_AccStart()	加速スタート	○
(47) hcpxxx_CnstStartFH()	F H定速スタート	
(48) hcpxxx_CnstStartFL()	F L定速スタート	
(49) hcpxxx_CnstStartByDec()	F H定速スタート後減速停止	
対応ボード	532 534 578 508 5212M 234 278 434 434v2 734 738 132 364	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸を加速(F L 定速, F H定速スタート後減速停止) スタートします。	
書式	DWORD hcpxxx_AccStart (DWORD hDevID, WORD axis); DWORD hcpxxx_CnstStartFH (DWORD hDevID, WORD axis); DWORD hcpxxx_CnstStartFL (DWORD hDevID, WORD axis); DWORD hcpxxx_CnstStartByDec(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・・・軸指定はORしたデータになります。	
	軸指定	CPD5212M以外軸名称 X Y Z U V W A B — — — — CPD5212M軸名称 X1 Y1 Z1 U1 X2 Y2 Z2 U2 X3 Y3 Z3 U3 引数 (Hex) 0x001 0x002 0x004 0x008 0x010 0x020 0x040 0x080 0x100 0x200 0x400 0x800
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_AccStart(hDevID, 0x07); //X, Y, Z軸加速スタート	
備考	F H定速スタート後減速停止の場合、減速開始点が手動計算設定に固定されますので、適当な減速開始点を計算し設定して下さい。	

(5 0) hcpxxx_SvOn()				サーボオン										○
(5 1) hcpxxx_SvOff()				サーボオフ										
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
(5 2) hcpxxx_SvResetOn()				サーボリセットオン										
(5 3) hcpxxx_SvResetOff()				サーボリセットオフ										
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	x	○	○	○	○	○	○	○	○	○	
(5 4) hcpxxx_PMOOn()				パルスモータ励磁オン										
(5 5) hcpxxx_PMOOff()				パルスモータ励磁オフ										
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたCPDボードの指定された軸のサーボ（サーボリセット、パルスモータ励磁）をオン／オフします。													
書式	DWORD hcpxxx_SvOn (DWORD <i>hDevID</i> , WORD <i>axis</i>); DWORD hcpxxx_SvOff (DWORD <i>hDevID</i> , WORD <i>axis</i>); DWORD hcpxxx_SvResetOn (DWORD <i>hDevID</i> , WORD <i>axis</i>); DWORD hcpxxx_SvResetOff (DWORD <i>hDevID</i> , WORD <i>axis</i>); DWORD hcpxxx_PMOOn (DWORD <i>hDevID</i> , WORD <i>axis</i>); DWORD hcpxxx_PMOOff (DWORD <i>hDevID</i> , WORD <i>axis</i>);													
引数	◆DWORD <i>hDevID</i> ・ ・ 対象デバイスのデバイスハンドル ◆WORD <i>axis</i> ・ ・ 軸指定 ・ ・ 軸指定はORしたデータになります。													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B				
		CPD5212M 軸名称	X 1	Y 1	Z 1	U 1	—	—	—	—				
		引数 (Hex)	0x001	0x002	0x004	0x008	0x010	0x020	0x040	0x080				
呼出例	DWORD ret; //関数の戻り値 ret = hcpxxx_SvOn(<i>hDevID</i> , 0x07); // X,Y,Z 軸サーボオン													
備 考	HPCI-CPD508 は S V R S T 端子がありません。 HPCI-CPD5212M は S V O N 端子が X 1 ～ U 1 軸までの 4 端子です。													

■加減速レート計算

(57) hcpxxx_CalAccRate()		加減速レートの計算											
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	加減速時間, FH, FL等より加減速レートを計算し, 指定したエリアに格納します。												
書式	DWORD hcpxxx_CalAccRate(DWORD* <i>rate</i> , DWORD <i>time</i> , DWORD <i>fh</i> , DWORD <i>fl</i> , WORD <i>pro</i> , WORD <i>s</i>);												
引数	◆DWORD* <i>rate</i> ・・計算結果が格納されるエリアのアドレス(加減速レート[1~65535])												
	◆DWORD <i>time</i> ・・加減速時間(msec)												
	◆DWORD <i>fh</i> ・・FH レジスタ値[1~65535]												
	◆DWORD <i>fl</i> ・・FL レジスタ値[1~65535]												
	◆WORD <i>pro</i> ・・加減速形式 [0:直線, 1:S 字]												
	◆WORD <i>s</i> ・・常に'0' にして下さい。												
呼出例	DWORD ret; //関数の戻り値 DWORD rate; //加減速レート //加減速時間 200ms, RFH=10000, RFL=500, 直線加減速 ret = hcpxxx_CalAccRate(&rate, 200, 10000, 500, 0, 0);												

2. 2 VB用ライブラリ関数

■デバイス関係

(1) hcpxxx_GetDevInfo() ボード枚数の取得, デバイス情報の取得													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	×	×	○	○	○	○	×	×
機能	現在パソコンに装着されているCPDボードの枚数,及びデバイス情報(※1)を取得します。												
書式	Public Function hcpxxx_GetDevInfo (ByRef HpcDevNumber As Long, _ HpcDevInfo As HPCDEVICEINFO) As Long												
引数	◆Dim HpcDevNumber As Long ・ ・ CPDボードの枚数 ◆Dim HpcDevInfo As HPCDEVICEINFO ・ ・ CPDボードのデバイス情報												
呼出例	Dim count As Long Dim ret As Long Dim hInfo(16) As HPCDEVICEINFO 'ボードのデバイス情報が格納されているエリアの先頭アドレス ret = hcpxxx_GetDevInfo(count, hInfo(0))												
備 考	※ 1. デバイス情報格納の構造体名及び構造体メンバは, ドライバの種類により異なる場合があります。詳細は「ユーザーズマニュアル<個別ボード編>」を参照して下さい。 (以下同様)												

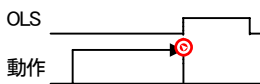
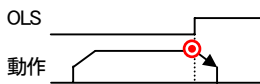
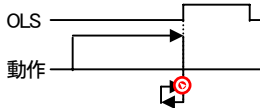
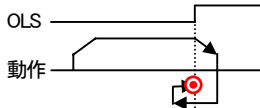
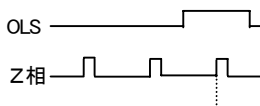
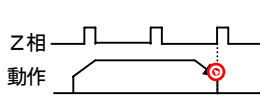
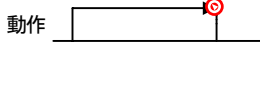
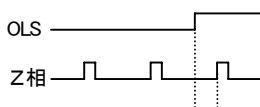
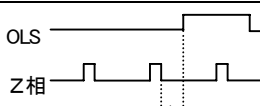
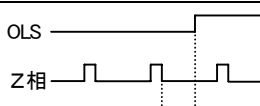
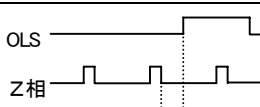
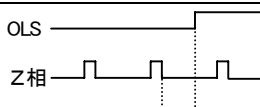
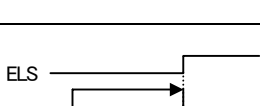
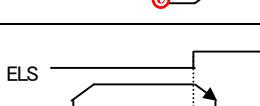
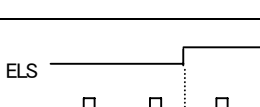

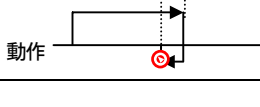
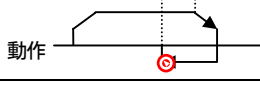
(2) hcpxxx_DevOpen() デバイスのオープン, レジスタとオプションポートの初期化													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	渡したデバイス情報を持つCPDボードをオープンし,他と識別するためのデバイスハンドルを取得します。以降このデバイスハンドルは,このCPDボードにアクセスするために使用します。 またオープンしたボードのレジスタとオプションポートの初期化をします。												
書式	Public Functionhcpxxx_DevOpen (ByRef hDevID As Long, hDevInfo As HPCDEVICEINFO) As Long												
引数	◆Dim hDevID As Long ・ ・ デバイスハンドル ◆Dim hDevInfo As HPCDEVICEINFO ・ ・ オープンするデバイス情報がセットされたエリアのアドレス												
呼出例	パソコンにCPDボードが2枚装着されていることを想定します。 デバイス情報格納エリアとしてHPCDEVICEINFO型(※1)の配列 HpcDevInfo(2)を準備し,この中には既に hcpxxx_GetDevInfo 関数により全ボードのデバイス情報が入っているものとします。 Dim ret As Long' 関数の戻り値 Dim hDevID(2) As Long' デバイスハンドル取得エリア ret = hcpxxx_DevOpen(hDevID(0), HpcDevInfo(0)) ' 1 番目のデバイス情報 ret = hcpxxx_DevOpen(hDevID(1), HpcDevInfo(1)) ' 2 番目のデバイス情報												
備 考	[オプションポートの初期値(※2)] ELS極性選択ポート : 0 x 0 0 (全軸B接) DLS／PCS切替ポート : &H f f (全軸PCS入力)												
	レジスタの初期値(※2)												
	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: 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				&H8 0 0 8 0 0 0							
	RENV1	環境設定 1				&H2 0 4 3 4 0 0 4							
	RENV2	環境設定 2				&H2 0 f d 6 5				CPD364			
						&H2 0 f d 5 5				CPD578, CPD278, CPD5212M			
						&H2 0 f 5 5 5				上記以外			
	RENV3	環境設定 3				&H f 0 0 0 0 2				原点復帰モード2(OLS+ENC)等			
	RIRQ	イベントマスク設定				&H 1				正常停止時			
本表以外のレジスタは初期値0 (レジスタの詳細はユーザーズマニュアル<共通編>を参照して下さい) ※1. ユーザーズマニュアル<個別ボード編>を参照 ※2. レジスタの初期値・オプションポートの初期値は関数内で直接与えています。そのため, この値を変更する場合はソースプログラム変更を行います。													

(3) hcpxxx_DevClose() デバイスのクローズ													◎
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	×	×	○	○	○	○	×	×
機能	デバイスハンドルで指定されたCPDボードをクローズします。 以降、このデバイスハンドルは無効となります。												
書式	Public Function hcpxxx_DevClose(ByVal <i>hDevID</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・ ・ クローズするボードのデバイスハンドル												
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_DevClose(hDevID)												
備考	デバイスクローズの前にCPDボードに対する終了処理を行ってください。												

■初期設定

(4) hcpxxx_SetOrgMode()													原点復帰モードの設定			○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364			
	○	○	○	○	○	○	○	○	○	○	○	○	○			
機能	デバイスハンドルで指定されたCPDボードの指定された軸の原点復帰モードを設定します。															
書式	Public Function hcpxxx_SetOrgMode _ (ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, ByVal <i>mode</i> As Integer) As Long															
引数	◆Dim <i>hDevID</i> As Long ・・ 対象デバイスのデバイスハンドル															
	◆Dim <i>axis</i> As Integer ・・ 軸指定															
	◆Dim <i>mode</i> As Integer ・・ 原点復帰モード [0:ORGmode0, 1:ORGmode1, ～, 12:ORGmode12] (次頁表)															
	(hcpxxx_DevOpen())での初期値は 2 : 原点復帰モード2, 0LSon で 減速, エンコーダZ相完了)															
軸指定	CPD5212M 以外軸名称			X	Y	Z	U	V	W	A	B	—	—	—	—	
	CPD5212M 軸名称			X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3	
	引数			0	1	2	3	4	5	6	7	8	9	10	11	
呼出例	Dim ret As Long '関数の戻り値 'Y軸を指定, Orgmode1(OLS 検出後拔出し再突入原点完了) ret = hcpxxx_SetOrgMode(hDevID, 1, 1)															


[原点復帰モード (ORGmode)]

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 (加速スタート)	
0	OLS原点 ORGmode=0		OLS検出で即停止 (完了)		OLS検出で減速停止 (完了)
1	OLS反転原点 ORGmode=1		OLS検出で即停止 補助速度で反転 OLS抜け出し後反転 OLS検出で即停止 (完了)		OLS検出で減速停止 補助速度で反転 OLS抜け出し後反転 OLS検出で即停止 (完了)
2	OLS+ENC原点 ORGmode=2		OLS検出後のn回目のZ相で即停止 (完了)		OLS検出で減速開始 n回目のZ相で 即停止 (完了)
3	OLS+ENC原点 ORGmode=3				OLS検出後の n回目のZ相で 減速停止 (完了)
4	OLS反転+ ENC原点 ORGmode=4		OLS検出で即停止 補助速度で反転 n回目のZ相で 即停止 (完了)		OLS検出で減速停止 補助速度で反転 n回目のZ相で 即停止 (完了)
5	OLS反転+ ENC原点 ORGmode=5		OLS検出で即停止 ベ-λ速度で反転 n回目のZ相で 即停止 (完了)		OLS検出で減速停止 加速しながら反転 n回目のZ相で 減速停止 (完了)
6	ELS兼用センサ原点 ORGmode=6		ELS検出で即停止 補助速度で反転 ELS抜け出しで即停止 (完了)		ELS検出で停止 (停止方法はRBN1による) 補助速度で反転 ELS抜け出しで即停止 (完了)
7	ELS反転+ ENC原点 ORGmode=7		ELS検出で即停止 補助速度で反転 n回目のZ相で 即停止 (完了)		ELS検出で停止 (停止方法はRBN1による) 補助速度で反転 n回目のZ相で 即停止 (完了)
8	ELS反転+ ENC原点 ORGmode=8		ELS検出で即停止 ベ-λ速度で反転 n回目のZ相で 即停止 (完了)		ELS検出で停止 (停止方法はRBN1による) 加速しながら反転 n回目のZ相で 減速停止 (完了)

(次頁へ続く)

(前頁からの続き)

No	原点復帰方法名	高速原点復帰（加速スタート）	
9	OLS反転+ CTR2原点 ORGmode=9		<p>OLS検出で減速、同時にCTR2クリア 停止後にCTR2の0点まで加減速動作。 ※ CTR2カウント入力は指令パルス(推奨) (REN3b9-b8=01)</p>
10	ENC反転+ CTR2原点 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CTR2の0点まで加減速動作。 ※ CTR2カウント入力はエンコーダ(推奨) (REN3b9-b8=00)</p>
11	ENC反転+ CTR2原点 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CTR2の0点まで加減速動作。 ※ CTR2カウント入力はエンコーダ(推奨) (REN3b9-b8=00)</p>
12	ELS反転+ CTR2原点 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CTR2の0点まで加減速動作。 ※ CTR2カウント入力はエンコーダ(推奨) (REN3b9-b8=00)</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。
3. 最小限CTR2はクリアを行う設定が必要です。（REN3 bit21 = '1'）
4. HPC1-CPD508、HPC1-CPD512M はエンコーダ入力の使用に制限があります。

(5) hcpxxx_SetEls()												E L S の設定				O
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364			
	○	○	○	○	○	○	○	○	○	○	○	○	○			
機能	デバイスハンドルで指定された C P D ボードの指定された軸の E L S の入力極性と入力時停止方法を設定します.															
書式	Public Function hcpxxx_SetEls(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal pol As Integer, ByVal stop As Integer) As Long															
引数	◆Dim hDevID As Long ・ 対象デバイスのデバイスハンドル ◆Dim axis As Integer ・ 軸指定 ◆Dim pol As Integer ・ 極性 [0: B 接, 1: A 接] (hcpxxx_DevOpen())での初期値は 0) ◆Dim stop As Integer ・ 停止方法 [0: 即停止, 1: 減速停止] (hcpxxx_DevOpen())での初期値は 0)															
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—		
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3		
		引数	0	1	2	3	4	5	6	7	8	9	10	11		
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetEls(hDevID, 1, 0, 0) 'Y 軸を指定, B 接, 即停止															

(6) hcpxxx_SetOls() OLSの設定													O
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸のOLSの入力極性を設定します。												
書式	Public Function hcpxxx_SetOls (ByVal hDevID As Long, ByVal axis As Integer, ByVal pol As Integer) As Long												
引数	◆Dim hDevID As Long ・・ 対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・ 軸指定 ◆Dim pol As Integer ・・ 極性 [0: B接, 1: A接] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetOls(hDevID, 1, 0) ' Y軸を指定, B接												

(7) hcpxxx_SetSvalm() SVALMの設定													O
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸のELSの入力極性と入力時停止方法を設定します。												
書式	Public Function hcpxxx_SetSvalm(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal pol As Integer, ByVal stop As Integer) As Long												
引数	◆Dim hDevID As Long ・・ 対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・ 軸指定 ◆Dim pol As Integer ・・ 極性 [0: B接, 1: A接] (hcpxxx_DevOpen()での初期値は0) ◆Dim stop As Integer ・・ 停止方法 [0: 即停止, 1: 減速停止] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetSvalm(hDevID, 1, 0, 0) ' Y軸を指定, B接, 即停止												
備考	HPCI-CPD508のSVALM信号はX-U, V-Bにまとめられているため, X-U軸, V-B軸の入力極性は一致させる必要があります。												









(8) hcpxxx_SetEz() エンコーダZ相の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸のエンコーダZ相の入力処理を設定します。												
書式	Public Function hcpxxx_SetEz(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal zcount As Integer, ByVal pol As Integer) As Long												
引数	◆Dim hDevID As Long ・・ 対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・ 軸指定 ◆Dim zcount As Integer ・・ 原点復帰時のZ相回数 [0: 1回目, ~, 15: 16回目] ◆Dim pol As Integer ・・ 極性 [0: 立下がりエッジ, 1: 立上がりエッジ] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
	引数		0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetEz(hDevID, 1, 2, 0) ' Y軸を指定, 3回目, 立下がりエッジ												

(9) hcpxxx_SetDlsSel () DLS/PCSの選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	×	○	○	○	○	○	○	○	×
機能	デバイスハンドルで指定されたCPDボードの指定された軸のDLS/PCS入力及びその極性を選択, DLS入力時の動作を設定します.												
書式	Public Function hcpxxx_SetEz(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>zcount</i> As Integer, ByVal <i>pol</i> As Integer) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・ 対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・ 軸指定 ◆Dim <i>para</i> As Integer ・・ DLS/PCS入力の切替 [0: DLS, 1: PCS, 2: 両方とも不使用] (hcpxxx_DevOpen())での初期値は2) ◆Dim <i>pol</i> As Integer ・・ 入力極性 [0: B接, 1: A接] ◆Dim <i>motion</i> As Integer ・・ DLS入力時の動作 [0:減速のみ, 1:減速停止] ◆Dim <i>latch</i> As Integer ・・ DLS入力のラッチ [0:ラッチしない, 1:ラッチする]												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetDlsSel(hDevID, 1, 0, 1, 0, 0) ' Y軸, DLS使用, A接, 減速のみ, ラッチしない												

(10) hcpxxx_SetInpos () INPOSの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	○	○	○	○	○	○	○	○	×
機能	デバイスハンドルで指定されたCPDボードの指定された軸のINPOSの入力信号処理方法を設定します.												
書式	Public Function hcpxxx_SetInpos(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>enable</i> As Integer, ByVal <i>pol</i> As Integer) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・ 対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・ 軸指定 ◆Dim <i>enable</i> As Integer ・・ 使用/不使用 [0:不使用, 1:使用] (初期値は0) ◆Dim <i>pol</i> As Integer ・・ 極性 [0: B接, 1: A接] (初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetInpos(hDevID, 1, 1, 1) ' Y軸を指定, 使用, A接												

(11) hcpxxx_SetSvCtrCl () 偏差カウンタクリア出力の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	○	○	○	○	○	○	○	○	×
機能	デバイスハンドルで指定されたCPDボードの指定された軸の偏差カウンタクリア出力の設定をします.												
書式	Public Function hcpxxx_SetSvCtrCl(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>enable</i> As Integer) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・ 対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・ 軸指定 ◆Dim <i>enable</i> As Integer ・・ 自動出力設定 [0: 不使用, 1: 原点完了時, 2: 異常停止時, 3: 原点完了及び異常停止時] (初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetSvCtrCl(hDevID, 1, 1) ' Y軸を指定, 原点完了時												

(12) hcpxxx_SetSls()		ソフトリミットの設定											
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸のソフトリミット処理を設定します。												
書式	DWORD hcpxxx_SetSls(DWORD hDevID, WORD axis, long ps/s, long ms/s, WORD enable, WORD stop);												
引数	◆Dim hDevID As Long ・ ・ 対象デバイスのデバイスハンドル												
	◆Dim axis As Integer ・ ・ 軸指定												
	◆Dim ps/s As Long ・ ・ + S L S [パルス数]												
	◆Dim ms/s As Long ・ ・ - S L S [パルス数]												
軸指定	◆Dim enable As Integer ・ ・ 使用/不使用 [0:不使用, 1:使用] (hcpxxx_DevOpen())での初期値は0)												
	◆Dim stop As Integer ・ ・ 停止方法 [0:即停止, 1:減速停止]												
	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
	引数	0	1	2	3	4	5	6	7	8	9	10	11
呼出例	Dim ret As Long ‘関数の戻り値 ’ Y軸, +SLS = 100000, -SLS = -50000, ソフトリミット使用, 即停止 ret = hcpxxx_SetSls(hDevID, 1, 100000, -50000, 1, 0)												
備 考	1. ソフトリミット使用時は①ps/sは必ずms/sより大きくして下さい。 2. ソフトリミット不使用時は①ms/s = ps/s = 0とし, ②enable = 0とします。 3. スタートコマンド書込み時に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。												

(1 3) hcpxxx_SetCmdPulse() 指令パルスの出力形式の設定														○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の指令パルスの出力形式を設定します。													
書式	DWORD hcpxxx_SetCmdPulse(DWORD hDevID, WORD axis, WORD cmdpls);													
引数	◆Dim hDevID As Long ・ 対象デバイスのデバイスハンドル ◆Dim axis As Integer ・ 軸指定 ◆Dim cmdpls As Integer ・ 指令パルスの出力形式 [0:個別指令方式, 1:共通指令方式] (※ 1) (hcpxxx_DevOpen() での初期値は 0)													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
	引数	0	1	2	3	4	5	6	7	8	9	10	11	
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetCmdPulse(hDevID, 0, 1) ' X軸を指定, 共通指令方式													
備考	※ 1. 指令パルスの出力形式													
	0 : 個別指令方式							1 : 共通指令方式						
	<div>xCWP端子 </div> <div>xCWN端子 </div> <div>xCCWP端子 </div> <div>xCCWN端子 </div>							<div>パルス列出力 xCWP端子 </div> <div>xCWN端子 </div> <div>方向出力 xCCWP端子 </div> <div>xCCWN端子 </div>						

(1 4) hcpxxx_SetAccProfile() S字/直線加減速の切替														
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたC P Dボードの指定された軸のS字／直線加減速の設定をします。													
書式	DWORD hcpxxx_SetAccProfile(DWORD <i>hDevID</i> , WORD <i>axis</i> , WORD <i>pr</i>);													
引数	◆Dim <i>hDevID</i> As Long ・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・軸指定 ◆Dim <i>pr</i> As Integer ・加減速形式 [0:直線, 1:S字] (hcpxxx_DevOpen()での初期値は0)													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
	引数		0	1	2	3	4	5	6	7	8	9	10	11
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetAccProfile(hDevID, 1, 1) ' Y軸を指定, S字加減速に設定													

(15) hcpxxx_SetAutoDec() 減速開始点計算方式の自動計算/手動計算切替													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の減速開始点計算方式を手動計算か自動計算か設定します。												
書式	DWORD hcpxxx_SetAutoDec(DWORD hDevID, WORD axis, WORD para);												
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ◆Dim para As Integer ・・減速開始点の設定方式 [0:自動計算設定, 1:手動計算設定] (hcpxxx_DevOpen() での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetAutoDec(hDevID, 1, 1) ' Y軸を指定, 減速開始点の設定方式をマニュアル設定												

(16) hcpxxx_mSetDls()													汎用入力／±DR／PCS／DLS入力選択, 設定		○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364		
	x	x	x	x	x	x	x	x	x	x	x	x	○		
機能	デバイスハンドルで指定されたC P Dボードの汎用入力/±DR/PCS/DLS 入力を選択し, ±DR, PCS, DLS の極性, DLS 選択時に DLS 入力時の動作を設定します.														
書式	DWORD hcpxxx_mSetDls(DWORD hDev, WORD para, WORD pol, WORD motion, WORD latch);														
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル														
	◆Dim para As Integer ・・汎用入力/±DR/PCS/DLS 入力選択 (詳細は個別編参照)														
	◆Dim pol As Integer ・・入力極性														
	[&H1:XDLS=A 接, &H2:YDLS=A 接, &H4:ZDLS=A 接, &H8:UDLS=A 接, &H10:XPCS=A 接, &H20:YPCS=A 接, &H40:ZPCS=A 接, &H80:UPCS=A 接, &H100:XDR=A 接,&H200:YDR=A 接,&H400:ZDR=A 接]														
	O R したデータを設定(初期値は 0 : 全軸 B 接)														
	para で入力選択されていない部分の設定は無効です.														
	(B 接になります. 汎用入力は A 接固定です.)														
	◆Dim motion As Integer ・・D L S 入力時の動作 [&H1 : X 軸減速停止, &H2 : Y 軸減速停止, &H4 : Z 軸減速停止, &H8 : U 軸減速停止]														
	O R したデータを設定														
	(hcpxxx_DevOpen())での初期値は 0 : 各軸減速のみ)														
呼出例	◆Dim latch As Integer ・・D L S 入力のラッチ														
	[&H1 : X 軸ラッチ, &H2 : Y 軸ラッチ, &H4 : Z 軸ラッチ, &H8 : U 軸ラッチ]														
	O R したデータを設定														
	(hcpxxx_DevOpen())での初期値は 0 : 全軸ラッチなし)														
	para					ビット / 機能 対応									
						7	6	5	4	3	2	1	0		
						US1	US0	ZS1	ZS0	YS1	YS0	XS1	XS0		
	J 4 コネクタ入力信号ピン番号					9 8 (9 9) ピン		9 7 ピン		4 8 (4 9) ピン		4 7 ピン			
	設 定 値					xS1 xS0	U 軸 : US1, US0		Z 軸 : ZS1, ZS0		Y 軸 : YS1, YS0		X 軸 : XS1, XS0		
	用	汎用入力信号 (初期値)				0 0	I N 3 0		I N 2 9		I N 2 6		I N 2 5		
途	P C L 外部 J O G 起動信号				0 1	- Z D R (- Y D R)		+ Z D R		- X D R (+ Y D R)		+ X D R			
区	P C L 位置決め管理開始信号				1 0	U P C S		Z P C S		Y P C S		X P C S			
分	P C L 減速センサ				1 1	U D L S		Z D L S		Y D L S		X D L S			
Dim ret As Long ' 関数の戻り値															
' X 軸±DR 使用, Z, U 軸は DLS 使用, 減速のみ, DLS 入力ラッチ															
ret = hcpxxx_mSetDls(hDevID, &Hf3, 0, &H0, &HC)															

(17) hcpxxx_mSetLtc() 汎用入力/LATCH入力の選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	x	x	x	x	x	x	x	x	x	x	x	x	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の汎用入力/LATCH 入力及びその極性を選択します。												
書式	Public Function hcpxxx_mSetLtc(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>para</i> As Integer, ByVal <i>pol</i> As Integer) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸] ◆Dim <i>para</i> As Integer ・・汎用入力/LATCH 入力の切替 [0: 汎用入力, 1: LATCH 入力,] (hcpxxx_DevOpen()での初期値は0) ◆Dim <i>pol</i> As Integer ・・LATCH 入力極性 [0: B接, 1: A接] (初期値は0) ※汎用入力使用時はA接固定です												
呼出例	Dim ret As Long; //関数の戻り値 ret = hcpxxx_mSetLtc(hDevID, 1, 1, 1); //Y軸, LATCH 入力使用, A接												

(18) hcpxxx_mSetInpos() INPOS入力/汎用入力の選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	x	x	x	x	x	x	x	x	x	x	x	x	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の汎用入力/INPOS入力及びその極性を選択します。												
書式	Public Function hcpxxx_mSetInpos(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>para</i> As Integer, ByVal <i>pol</i> As Integer) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸] ◆Dim <i>para</i> As Integer ・・汎用入力/INPOS 入力の切替 [0: INPOS 入力, 1: 汎用入力] (hcpxxx_DevOpen()での初期値は0) ◆Dim <i>pol</i> As Integer ・・INPOS 入力極性 [0: B接, 1: A接] (hcpxxx_DevOpen()での初期値は0) ※汎用入力使用時はA接固定です。												
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_mSetInpos(hDevID, 1, 0, 1) ' Y軸, INPOS 使用, A接												

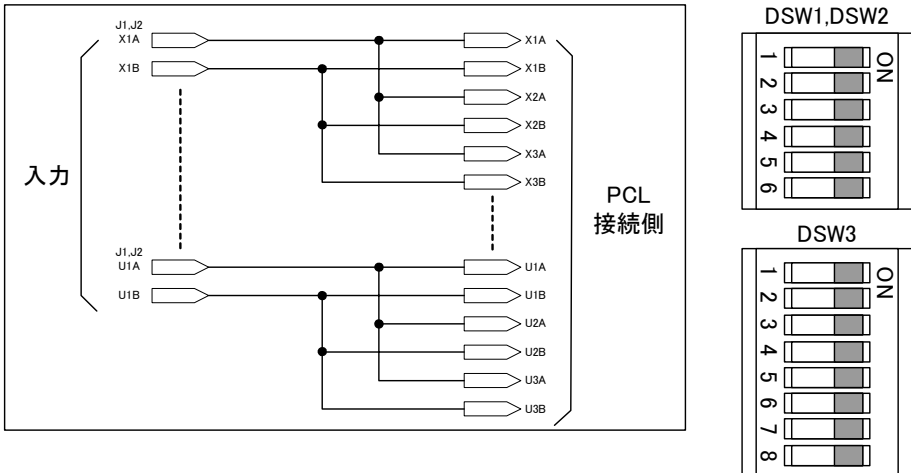
(19) hcpxxx_mSetSvCtrl() 汎用出力/SVCTRCL出力選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	x	x	x	x	x	x	x	x	x	x	x	x	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の SVCTRCL 出力または汎用出力を選択します。SVCTRCL 出力選択時の自動出力設定をします。												
書式	Public Function hcpxxx_mSetSvCtrl(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>para</i> As Integer, ByVal <i>enable</i> As Integer) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 [0:X軸, 1:Y軸, 2:Z軸, 3:U軸] ◆Dim <i>para</i> As Integer ・・SVCTRCL 出力/汎用出力の切替 [0: SVCTRCL 出力, 1: 汎用出力] (hcpxxx_DevOpen()での初期値は0) ◆Dim <i>enable</i> As Integer ・・自動出力設定 [0: 不使用, 1: 原点完了時, 2: 異常停止時, 3: 原点完了及び異常停止時] (hcpxxx_DevOpen()での初期値は0)												
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_mSetSvCtrl(hDevID, 2, 0, 1); ' Z軸, SVCTRCL 原点復帰完了時自動出力												

■状態読出し

(2 0) hcpxxx_ReadMainSts ()		メインステータスの読出し											◎	
(2 1) hcpxxx_ReadErrorSts ()		エラーステータスの読出し											◎	
(2 2) hcpxxx_ReadEventSts ()		イベントステータスの読出し											◎	
(2 3) hcpxxx_ReadSubSts ()		サブステータスの読出し												
(2 4) hcpxxx_ReadExSts ()		拡張ステータスの読出し												
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定された C P D ボードの指定された軸のメインステータス（エラーステータス、イベントステータス、サブステータス、拡張ステータス）を読出し、指定したエリアに格納します。ステータスデータについては、ユーザーズマニュアル<共通編>を参照して下さい。													
書式	Public Function hcpxxx_ReadMainSts _ (ByVal hDevID As Long, ByVal axis As Integer, ByRef status As Integer) As Long Public Function hcpxxx_ReadErrorSts _ (ByVal hDevID As Long, ByVal axis As Integer, ByRef status As Long) As Long Public Function hcpxxx_ReadEventSts _ (ByVal hDevID As Long, ByVal axis As Integer, ByRef status As Long) As Long Public Function hcpxxx_ReadSubSts _ (ByVal hDevID As Long, ByVal axis As Integer, ByRef status As Integer) As Long Public Function hcpxxx_ReadExSts _ (ByVal hDevID As Long, ByVal axis As Integer, ByRef status As Long) As Long													
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ◆Dim status As Integer ・・メイン・サブステータスが格納されるエリアのアドレス ◆Dim status As Long ・・エラー・イベント・拡張ステータスが格納されるエリアのアドレス													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
	引数		0	1	2	3	4	5	6	7	8	9	10	11
呼出例	Dim ret As Long	'関数の戻り値												
	Dim ssts As Integer	'サブステータス												
	ret = hcpxxx_ReadSubSts(hDevID, 1, ssts) ' Y 軸を指定 サブステータス													

(2 5) hcpxxx_ReadIpSts()													補間ステータスの読出し		
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364		
	○	○	×	×	×	○	×	○	○	○	×	○	○		
機能	デバイスハンドルで指定されたCPDボードの補間ステータスを読出し、指定したエリアに格納します。ステータスデータについては、ユーザーズマニュアル<共通編>を参照して下さい。														
書式	Public Function hcpxxx_ReadIpSts(ByVal hDevID As Long, ByRef status As Long) As Long														
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim status As Long ・・読出されたデータが格納されるエリアのアドレス														
呼出例	Dim ret As Long ' 関数の戻り値 Dim ipsts As Long ' 補間ステータス ret = hcpxxx_ReadIpSts(hDevID, ipsts) ' デバイスハンドル、格納先のアドレス														

(2 6) hcpxxx_ReadSpd() 現在速度の読出													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の現在速度を読出し,指定したエリアに格納します。補間動作中は補間代表軸を読み出してください。												
書式	Public Function hcpxxx_ReadSpd(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByRef <i>speed</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 ◆Dim <i>speed</i> As Long ・・読出されたデータが格納されるエリアのアドレス												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
呼出例	Dim ret As Long '関数の戻り値 Dim speed As Long '速度 ret = hcpxxx_ReadSpd(hDevID, 0, speed) ' X 軸を指定, 格納先のアドレス												

(2 7) hcpxxx_ReadCtr() カウンタの読出													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたC P Dボードの指定された軸の,指定されたカウンタを読出し,指定したエリアに格納します。												
書式	Public Function hcpxxx_ReadCtr(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>selctr</i> As Integer, ByRef <i>count</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 ◆Dim <i>selctr</i> As Integer ・・カウンタ選択 [1:カウンタ1, 2:カウンタ2, 3:カウンタ3, 4:カウンタ4] カウンタ1, カウンタ4 ・・指令パルス出力 カウンタ2 ・・エンコーダ入力 カウンタ3 ・・偏差カウンタ (脱調検出用) ◆Dim <i>count</i> As Long ・・読出されたデータが格納されるエリアのアドレス												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
	CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
呼出例	Dim ret As Long; //関数の戻り値 long* count; //カウンタ1 (指令パルス出力) の値 // X 軸を指定, カウンタ1 を指定, 格納先アドレス ret = hcpxxx_ReadCtr(hDevID, 0, 1, &count);												
備考	<p>HPCI-CPD508 はカウンタ2, カウンタ3の指定はできません。 HPCI-CPD5212M は X1 軸～U1 軸のエンコーダ入力が, X2 軸～U2 軸, X3 軸～U3 軸にも接続されています。詳しくは個別ボード編を参照してください。 [HPCI-CPD5212M エンコーダ入力A/B相接続]</p> 												

■動作設定

(28) hcpxxx_SetFLSpd() ベース速度の設定													○
(29) hcpxxx_SetAuxSpd() 補助速度の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の ベース速度・・・高速動作時の立ち上がりの速度（本速度から加速，本速度まで減速して停止） 補助速度・・・一部の原点復帰において，原点突入速度等に使用を設定します。 指令速度（pps）を速度倍率で除算した値を設定。												
書式	Public Function hcpxxx_SetFLSpd (ByVal hDevID As Long, ByVal axis As Integer, _ ByVal speed As Long) As Long Public Function hcpxxx_SetAuxSpd(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal speed As Long) As Long												
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ◆Dim speed As Long ・・ベース速度(FL), 補助速度(FA) レジスタ値 [1~65535, FL(FA)<FH(動作速度)] (hcpxxx_DevOpen()での初期値は200)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_SetFLSpd(hDevID, 1, 500) ' Y軸を指定, FL=500*倍率												

(30) hcpxxx_SetAccRate() 加速レートの設定													○
(31) hcpxxx_SetDecRate() 減速レートの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の加速レート（減速レート）を設定します。												
書式	Public Function hcpxxx_SetAccRate(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal rate As Long) As Long Public Function hcpxxx_SetDecRate(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal rate As Long) As Long												
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ◆Dim rate As Long ・・加速レート[1~65535], 減速レート[0~65535](※1) (hcpxxx_DevOpen()での初期値は1364. FH=2000, FL=200, 直線加減速で加減速時間約0.5秒)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_SetAccRate(hDevID, 1, 500) ' Y軸を指定, 加速レート=500												
備考	※1. 加速レート（減速レート）と加減速時間の関係 FH, FL (RFH, RFLレジスタ値), RUR (加速／減速レートレジスタ値) (1)直線加速 加速時間[sec] = $\frac{(RFH-RFL) \times (RUR+1) \times 4}{19,660,800}$ (2)直線部分のないS字加速 加速時間[sec] = $\frac{(RFH-PRFL) \times (RUR+1) \times 8}{19,660,800}$ 減速レート=0の時は減速レート=加速レートとなります。												

(32) hcpxxx_SetMult() 速度倍率設定値 (MG) の設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の速度倍率設定値 (MG)(※1)を設定します。												
書式	Public Function hcpxxx_SetMult(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal rmg As Long) As Long												
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ◆Dim rmg As Long ・・速度倍率設定値[2~4095] (hcpxxx_DevOpen())での初期値は初期値は299:速度倍率1倍)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetMult(hDevID, 1, 299) 'Y軸を指定, 倍率1倍												
備考	※1. 速度と速度倍率の関係及び速度倍率設定値と速度倍率の関係 RFxは速度レジスタ(RFH, RFL, RFA)の値 $\text{速度[PPS]} = \text{RFx} \times \text{速度倍率} = \frac{\text{RFx} \times 300}{\text{RMG} + 1} \quad \text{速度倍率設定値 (MG)} = \frac{300}{\text{速度倍率}} - 1$												
	速度倍率設定例												
	設定値	倍率	出力速度範囲(pps)				設定値	倍率	出力速度範囲(pps)				
	2999 (0x0bb7)	0.1	0.1~ 6,553.5				59 (0x003b)	5	5~ 327.675				
	1499 (0x05db)	0.2	0.2~ 13,107				29 (0x001d)	10	10~ 655.350				
	599 (0x0257)	0.5	0.5~ 32,767.5				14 (0x000e)	20	20~1,310.700				
	299 (0x012b)	1	1 ~ 65,535				5 (0x0005)	50	50~3,276.750				
	149 (0x0095)	2	2 ~131,070				2 (0x0002)	100	100~6,553,500				

(33) hcpxxx_SetEventMask() イベントマスクの設定														
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたCPDボードの指定された軸のイベントマスクを設定します。													
書式	Public Function hcpxxx_SetEventMask(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>mask</i> As Long) As Long													
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
		引数	0	1	2	3	4	5	6	7	8	9	10	11
	◆Dim <i>mask</i> As Long ・・イベントマスクデータ 下表の値をORしたデータを与えることで2種類以上のイベント報告指定となります。(hcpxxx_DevOpen())での初期値は1：正常停止時)													
	mask (HEX)	イベントステータス報告のイベント						mask (HEX)	イベントステータス報告のイベント					
	0x00001	正常停止時						0x00400	コンパレータ3条件成立時					
	0x00002	次動作継続スタート時						0x00800	コンパレータ4条件成立時					
	0x00004	動作用プリレジスタフルから空きができた時						0x01000	コンパレータ5条件成立時					
	0x00008	コンパレータ5用プリレジスタフルから空きができた時						0x02000	CLR入力によるカウン値のクリア時					
	0x00010	加速開始時						0x04000	LATCH入力によるカウント値のラッチ時					
	0x00020	加速終了時						0x08000	QLS入力によるカウント値のラッチ時					
0x00040	減速開始時						0x10000	DLS入力ON時						
0x00080	減速終了時						0x20000	±DR入力変化時						
0x00100	コンパレータ1条件成立時						0x40000	CSTA信号入力ON時						
0x00200	コンパレータ2条件成立時													
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetEventMask(<i>hDevID</i> , 1, &H1001) 'Y軸を指定, 正常停止, コンパレータ5条件成立時													

(34) hcpxxx_SetDecPoint() 減速開始点の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の減速開始点を手動計算設定とした時の減速開始点を設定します。												
書式	Public Function hcpxxx_SetDecPoint(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>dstnc</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 ◆◆Dim <i>dstnc</i> As Long ・・減速開始点 (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SetDecPoint(hDevID, 1, 300) 'Y軸を指定, 減速開始点=300												
備考	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。 減速開始点の最適値は次式のようになります。 ■直線減速 $\text{最適値[パルス]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 32,768}$ ■直線部分のないS字減速 $\text{最適値[パルス]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1) \times 2}{(\text{RMG} + 1) \times 32,768}$												

■運用設定

(35) hcpxxx_WritOpeMode() 動作モードの設定													◎	
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたCPDボードの指定された軸の動作モードを設定します。													
書式	DWORD hcpxxx_WritOpeMode(DWORD hDevID, WORD axis, WORD mode);													
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
		引数	0	1	2	3	4	5	6	7	8	9	10	11
	◆DWORD mask ・・動作モード (hcpxxx_DevOpen()での初期値は0 : +方向連続動作)													
	mode (Hex)	動作モード							備考				記事	
	&H0	コマンド制御による+方向連続動作												
	&H8	コマンド制御による-方向連続動作												
	&H1	パルス入力による連続動作											※1	
	&H2	±DR入力による連続動作											※2	
	&H10	+方向原点復帰動作												
	&H18	-方向原点復帰動作												
	&H12	+方向原点抜け出し動作												
	&H1a	-方向原点抜け出し動作												
	&H15	+方向原点サーチ動作							hcpxxx_WritPos () 関数で引出量を設定					
	&H1d	-方向原点サーチ動作												
	&H20	+EL又は+SL位置まで動作												
	&H28	-EL又はSL位置まで動作												
	&H22	+EL又は+SL抜け出し動作												
	&H2a	-EL又はSL抜け出し動作												
	&H24	+方向にEZカウント分だけ動作												
	&H2c	-方向にEZカウント分だけ動作												
	&H41	位置決め動作							hcpxxx_WritPos()関数で移動量を設定					
	&H42	PCS位置決め動作 (ライブラリ関数作成動作)												
	&H44	指令位置0点復帰動作												
	&H45	機械位置0点復帰動作											※1	
	&H46	+方向1パルス動作												
	&H4e	-方向1パルス動作												
	&H47	タイマー動作							hcpxxx_WritPos()関数で移動量を設定					
	&H51	パルス入力による位置決め動作											※1	
	&H54	パルス入力による指令位置0点復帰動作											※1	
	&H55	パルス入力による機械位置0点復帰動作											※1	
	&H56	±DR入力による位置決め動作							hcpxxx_WritPos()関数で移動量を設定				※2	
	&H60	連続直線補間動作												
	&H61	直線補間動作							hcpxxx_WritLine()関数で移動量を設定					
	&H64	CW方向円弧補間動作 (時計回り)							hcpxxx_WritCirc()関数で移動量を設定					
	&H65	CCW方向円弧補間動作 (反時計回り)												
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_WritOpeMode(hDevID, 1, &H41) ' Y軸を指定, 位置決め動作													
備 考	※1. HPCI-CPD508 のボードでは,これらの動作はできません. ※2. HP104D-CPD364 のみの機能です.													

(36) hcpxxx_WritFHSpd() 動作速度の設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の動作速度レジスタ（速度を倍率で除算した値）を設定します。												
書式	Public Function hcpxxx_WritFHSpd(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>speed</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 ◆Dim <i>speed</i> As Long ・・FH（動作速度）レジスタ値[1~65535] (hcpxxx_DevOpen()での初期値は2000)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_WritFHSpd(hDevID, 1, 5000) 'Y軸を指定, FH=5000*速度倍率												

(37) hcpxxx_WritPos() 位置決め移動量の設定													○
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の位置決め動作の移動量を設定します。												
書式	Public Function hcpxxx_WritPos(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>dstnc</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 ◆Dim <i>dstnc</i> As Long ・・移動量[-134, 217, 728~134, 217, 727(28ビット)] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_WritPos(hDevID, 1, 15000) 'Y軸を指定, +方向15000パルス												

(38) hcpxxx_WritLine() 直線補間の移動量の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	デバイスハンドルで指定されたCPDボードの指定された軸の直線補間の移動量を設定します。												
書式	Public Function hcpxxx_WritLine(ByVal <i>hDevID</i> As Long, ByVal <i>axis</i> As Integer, _ ByVal <i>dstnc</i> As Long) As Long												
引数	◆Dim <i>hDevID</i> As Long ・・対象デバイスのデバイスハンドル ◆Dim <i>axis</i> As Integer ・・軸指定 ◆Dim <i>dstnc</i> As Long ・・移動量[-134, 217, 728~134, 217, 727(28ビット)] (hcpxxx_DevOpen()での初期値は0)												
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3
		引数	0	1	2	3	4	5	6	7	8	9	10
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_WritLine(hDevID, 0, -30000) '補間代表軸: X軸を指定, 一方向30000パルス ret = hcpxxx_WritLine(hDevID, 1, -15000) '従軸: Y軸を指定, 一方向15000パルス												

(39) hcpxxx_WritCircI()														円弧補間の移動量の設定							
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364								
	○	○	○	○	○	○	○	○	○	○	○	○	○								
機能	デバイスハンドルで指定されたCPDボードの指定された軸の円弧補間の移動量を設定します。 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。 データの設定方法については「ユーザーズマニュアル<共通編>」を参照して下さい。																				
書式	Public Function hcpxxx_WritCircI(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal dstnc1 As Long, ByVal dstnc2 As Long, _ ByVal center1 As Long, ByVal center2 As Long) As Long																				
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・円弧補間軸指定 ◆Dim dstnc1 As Long ・・終点位置1 [-134, 217, 728~134, 217, 727 (28ビット)] ◆Dim dstnc2 As Long ・・終点位置2 [-134, 217, 728~134, 217, 727 (28ビット)] ◆Dim center1 As Long ・・中心位置1 [-134, 217, 728~134, 217, 727 (28ビット)] ◆Dim center2 As Long ・・中心位置2 [-134, 217, 728~134, 217, 727 (28ビット)] (hcpxxx_DevOpen()での位置データの初期値は0)																				
	軸指定	CPD5212M 以外軸名称	X-Y	X-Z	X-U	Y-Z	Y-U	Z-U	V-W	V-A	V-B	W-A	W-B	A-B	—	—	—	—	—	—	
		CPD5212M 軸名称	X1-Y1	X1-Z1	X1-U1	Y1-Z1	Y1-U1	Z1-U1	X2-Y2	X2-Z2	X2-U2	Y2-Z2	Y2-U2	Z2-U2	X3-Y3	X3-Z3	X3-U3	Y3-Z3	Y3-U3	Z3-U3	
	引数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
呼出例	Dim ret As Long '関数の戻り値 '半径 1000 パルスの真円 ret = hcpxxx_WritCircI(hDevID, 0, 0, 1000, 0) 'X-Y軸円弧, 終点(X,Y)=(0,0), 中心(X,Y)=(1000,0)																				

(40) hcpxxx_WritCtr() カウンタプリセット														
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364	
	○	○	○	○	○	○	○	○	○	○	○	○	○	
機能	デバイスハンドルで指定されたCPDボードの指定された軸の指定されたカウンタをプリセットします。													
書式	Public Function hcpxxx_WritCtr(ByVal hDevID As Long, ByVal axis As Integer, _ ByVal preset As Long, ByVal selctr As Integer) As Long													
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ◆Dim preset As Long ・・プリセット値 [-134, 217, 728~134, 217, 727 (28ビット)] ◆Dim selctr As Integer ・・カウンタ選択 [1:カウンタ1, 2:カウンタ2, 3:カウンタ3, 4:カウンタ4] カウンタ1, カウンタ4 ・・指令パルス出力 カウンタ2 ・・エンコーダ入力 カウンタ3 ・・偏差カウンタ (脱調検出用)													
	軸指定	CPD5212M 以外軸名称	X	Y	Z	U	V	W	A	B	—	—	—	—
		CPD5212M 軸名称	X1	Y1	Z1	U1	X2	Y2	Z2	U2	X3	Y3	Z3	U3
		引数	0	1	2	3	4	5	6	7	8	9	10	11
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_WritCtr(hDevID, 0, 5000, 1) 'X軸, プリセット値=5000, カウンタ1													

■動作制御指令

(41) hcpxxx_DecStop()	減速停止	○
(42) hcpxxx_QuickStop()	即停止	
(43) hcpxxx_EmgStop()	非常停止	
対応ボード	532 534 578 508 5212M 234 278 434 434v2 734 738 132 364	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
機能	デバイスハンドルで指定されたCPDボードの指定された軸を減速停止(即停止,非常停止)します。	
書式	Public Function hcpxxx_DecStop (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_QuickStop(ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_EmgStop (ByVal hDevID As Long, ByVal axis As Integer) As Long	
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ・・軸指定はORしたデータになります。	
軸指定	CPD5212M 以外軸名称	X Y Z U V W A B — — — —
軸指定	CPD5212M 軸名称	X1 Y1 Z1 U1 X2 Y2 Z2 U2 X3 Y3 Z3 U3
引数 (Hex)		0x001 0x002 0x004 0x008 0x010 0x020 0x040 0x080 0x100 0x200 0x400 0x800
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_EmgStop(hDevID, &H7) ' X, Y, Z 軸非常停止	
備考	5 軸以上 (HPCI-CPD578/508, 5212M, HPCI-CPD738) で 2 つのモータコントロール L S I にまたがる軸を指定した場合、停止するタイミングは異なります。それ以外は同時に停止します。	

(44) hcpxxx_SyDecStop()	同時減速停止	
(45) hcpxxx_SyQuickStop()	同時即停止	
対応ボード	532 534 578 508 5212M 234 278 434 434v2 734 738 132 364	x x ○ ○ ○ x ○ x x x ○ x x
機能	デバイスハンドルで指定されたCPDボードの指定された軸を同時に減速停止(即停止)します。	
書式	Public Function hcpxxx_SyDecStop (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_SyQuickStop(ByVal hDevID As Long, ByVal axis As Integer) As Long	
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ・・軸指定はORしたデータになります。	
軸指定	CPD5212M 以外軸名称	X Y Z U V W A B — — — —
軸指定	CPD5212M 軸名称	X1 Y1 Z1 U1 X2 Y2 Z2 U2 X3 Y3 Z3 U3
引数 (Hex)		0x001 0x002 0x004 0x008 0x010 0x020 0x040 0x080 0x100 0x200 0x400 0x800
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_SyDecStop(hDevID, &H7) ' X, Y, Z 軸同時減速停止	
備考	この関数の実行完了時にエラーステータスのビット 8 (同時停止入力による停止) が立ちますので、関数実行完了後に、指定した軸のエラーステータスレジスタを読み込む必要があります。	

(46) hcpxxx_AccStart()	加速スタート	○
(47) hcpxxx_CnstStartFH()	FH定速スタート	
(48) hcpxxx_CnstStartFL()	FL定速スタート	
(49) hcpxxx_CnstStartByDec()	FH定速スタート後減速停止	
対応ボード	532 534 578 508 5212M 234 278 434 434v2 734 738 132 364	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
機能	デバイスハンドルで指定されたCPDボードの指定された軸を加速(FL 定速, FH 定速スタート後減速停止) スタートします。	
書式	Public Function hcpxxx_AccStart (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_CnstStartFH (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_CnstStartFL (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_CnstStartByDec(ByVal hDevID As Long, ByVal axis As Integer) As Long	
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定 ・・軸指定はORしたデータになります。	
軸指定	CPD5212M 以外軸名称	X Y Z U V W A B — — — —
軸指定	CPD5212M 軸名称	X1 Y1 Z1 U1 X2 Y2 Z2 U2 X3 Y3 Z3 U3
引数 (Hex)		0x001 0x002 0x004 0x008 0x010 0x020 0x040 0x080 0x100 0x200 0x400 0x800
呼出例	Dim ret As Long ' 関数の戻り値 ret = hcpxxx_AccStart(hDevID, &H7) ' X, Y, Z 軸加速スタート	
備考	FH 定速スタート後減速停止の場合、減速開始点が手動計算設定に固定されますので、適当な減速開始点を計算し設定して下さい。	

(5 0) hcpxxx_SvOn()													サーボオン		○
(5 1) hcpxxx_SvOff()													サーボオフ		
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364		
	○	○	○	○	○	○	○	○	○	○	○	○	○		
(5 2) hcpxxx_SvResetOn()													サーボリセットオン		
(5 3) hcpxxx_SvResetOff()													サーボリセットオフ		
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364		
	○	○	○	×	○	○	○	○	○	○	○	○	○		
(5 4) hcpxxx_PMOOn()													パルスモータ励磁オン		
(5 5) hcpxxx_PMOOff()													パルスモータ励磁オフ		
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364		
	○	○	○	○	○	○	○	○	○	○	○	○	○		
機能	デバイスハンドルで指定されたCPDボードの指定された軸のサーボ（サーボリセット、パルスモータ励磁）をオン／オフします.														
書式	Public Function hcpxxx_SvOn (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_SvOff (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_SvResetOn (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_SvResetOff (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_PMOOn (ByVal hDevID As Long, ByVal axis As Integer) As Long Public Function hcpxxx_PMOOff (ByVal hDevID As Long, ByVal axis As Integer) As Long														
引数	◆Dim hDevID As Long ・・対象デバイスのデバイスハンドル ◆Dim axis As Integer ・・軸指定・・軸指定はORしたデータになります.														
	軸指定	CPD5212M 以外軸名称			X	Y	Z	U	V	W	A	B			
		CPD5212M 軸名称			X1	Y1	Z1	U1	—	—	—	—			
		引数 (Hex)			0x001	0x002	0x004	0x008	0x010	0x020	0x040	0x080			
呼出例	Dim ret As Long '関数の戻り値 ret = hcpxxx_SvOn(hDevID, &H7) ' X, Y, Z 軸サーボオン X, Y, Z 軸サーボオン														
備考	HPCI-CPD508 はSV RST 端子がありません. HPCI-CPD5212M はSV ON 端子がX 1 ～U 1 軸までの4 端子です.														

■加減速レート計算

(57) hcpxxx_CalAccRate()		加減速レートの計算											
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	○	○	○	○	○	○
機能	加減速時間、FH、FL等より加減速レートを計算し、指定したエリアに格納します。												
書式	Public Function hcpxxx_CalAccRate (ByRef rate As Long, ByVal time As Long, ByVal fh As Long, _ ByVal fl As Long, ByVal pro As Long, ByVal s As Integer) As Long												
引数	◆Dim rate As Long ・・計算結果が格納されるエリアのアドレス、(加減速レート[1～65535])												
	◆Dim time As Long ・・加減速時間 (msec)												
	◆Dim fh As Long ・・FH レジスタ値 [1～65535]												
	◆Dim fl As Long ・・FL レジスタ値 [1～65535]												
	◆Dim pro As Integer ・・加減速形式 [0:直線, 1:S字]												
	◆Dim s As Integer ・・常に'0'にして下さい。												
呼出例	Dim ret As Long '関数の戻り値												
	Dim rate As Long '加減速レート ret = hcpxxx_CalAccRate(&rate, 200, 10000, 500, 0, 0) '加減速時間 200ms, RfH=10000, RfL=500, 直線加減速												

2. 3 DOS版ライブラリ関数

DOS版のライブラリ関数はWindows版VC++と互換性を持たせていますが、DOSでの制御は「高度な使用方法」を前提としていますから、このライブラリ関数はアプリケーション作成の＜参考用＞として記載してあります。使用する場合には、十分な注意を払って下さい。

■デバイス関係

(1) hcpxxx_GetDevInfo() ボード枚数の取得, デバイス情報の取得													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	×	×	×	×	×	×	×	×
機能	現在パソコンに装着されているCPDボードの枚数, 及びデバイス情報を取得します。												
書式	short hcpxxx_GetDevInfo(WORD* DevCnt, HPCDEVINFO* DevInfo);												
引数	WORD* DevCnt; /*ボード枚数の格納アドレス*/ HPCDEVINFO* DevInfo; /*デバイス情報の格納アドレス*/												
(2) hcpxxx_DevOpen() デバイスのオープン, レジスタとオプションポートの初期化													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	渡したデバイス情報を持つCPDボードをオープンし, 他と識別するためのデバイスハンドルを取得します。以降このデバイスハンドルは, このCPDボードにアクセスするために使用します。またオープンしたボードのレジスタとオプションポートの初期化をします。												
書式	short hcpxxx_DevOpen(DWORD* hDevID, HPCDEVINFO* DevInfo);												
引数	DWORD* hDevID; /*取得デバイスハンドル格納アドレス*/ HPCDEVINFO* DevInfo; /*オープンするデバイス情報の格納アドレス*/												
(3) hcpxxx_DevClose() デバイスのクローズ													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたCPDボードをクローズします。以降, このデバイスハンドルは無効となります。												
書式	short hcpxxx_DevClose(DWORD hDevID);												
引数	DWORD hDevID; /*クローズするボードのデバイスハンドル*/												

■初期設定

(4) hcpxxx_SetOrgMode() 原点復帰モードの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたCPDボード・軸の原点復帰方式を設定します。 具体的な内容は「 原点復帰モード (ORGmode) (P12) 」の図に示してあります。												
書式	short hcpxxx_SetOrgMode(DWORD hDevID, short axis, short mode);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis: /*軸指定 (0-11)*/ short mode: /*原点復帰モード [0-12]*/												

(5) hcpxxx_SetEls() ELSの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたCPDボード・軸のELS入力極性と停止方法を設定します。												
書式	short hcpxxx_SetEls(DWORD hDevID, short axis, WORD pol, WORD stop);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis: /*軸指定 (0-11)*/ 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] */												

(6) hcpxxx_SetOls() O L S の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のO L S入力極性と停止方法を設定します。												
書式	short hcpxxx_SetOls(DWORD hDevID, short axis, WORD pol);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD pol; /*極性[0: B接, 1: A接]*/												
(7) hcpxxx_SetSvAlm() S V A L M の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のS V A L M入力極性と停止方法を設定します。												
書式	short hcpxxx_SetSvAlm(DWORD hDevID, short axis, WORD pol, WORD stop);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ 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] */												
(8) hcpxxx_SetEz() エンコーダZ相の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のエンコーダZ相使用時の検出回数と入力極性を設定します。												
書式	short hcpxxx_SetEz(DWORD hDevID, short axis, WORD count, WORD pol);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD count; /*原点復帰時のZ相回数[0:1回目, ~15:16回目]*/ WORD pol; /*極性[0: 立下がりエッジ, 1: 立上がりエッジ]*/												
(9) hcpxxx_SetDlsSel() D L S / P C S の選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	×
機能	指定されたC P Dボード・軸のD L S / P C S入力選択及びその極性を選択, D L S選択時の入力時動作を設定します。												
書式	short hcpxxx_SetDlsSel(DWORD hDevID, short axis, short enable, WORD pol, WORD motion, WORD latch);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ short para; /*DLS/PCS入力の切替 [0: D L S, 1: P C S, 2:両方不使用] */ WORD pol; /*入力極性 [0: B接, 1: A接] */ WORD motion; /*D L S入力時の動作 [0:減速のみ, 1:減速停止] */ WORD latch; /*D L S入力のラッチ [0:ラッチしない, 1:ラッチする] */												
(10) hcpxxx_SetInpos() I N P O S の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	×
機能	指定されたC P Dボード・軸のI N P O S使用可否と入力極性を設定します。												
書式	short hcpxxx_SetInpos(DWORD hDevID, short axis, WORD enable, WORD pol);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD enable; /*使用/不使用[0:不使用, 1:使用]*/ WORD pol; /*極性[0: B接, 1: A接]*/												
(11) hcpxxx_SetSvCtrCl() 偏差カウンタクリア出力の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	×
機能	指定されたC P Dボード・軸の偏差カウンタクリア出力を設定します。												
書式	short hcpxxx_SetSvCtrCl(DWORD hDevID, short axis, short enable);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ short enable; /*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*/												

(1 2) hcpxxx_SetSls() ソフトリミットの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のソフトリミット使用可否と停止方法を設定します。												
書式	short hcpxxx_SetSls(DWORD hDevID, short axis, long psIs, long msIs, WORD enable, WORD stop);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ long psIs; /*+ S L S [パルス数]*/ long msIs; /*- S L S [パルス数]*/ WORD enable; /*カウンタの指定 [0:不使用, 1:使用]*/ WORD stop; /*停止方法 [0:即停止, 1:減速停止]*/												
(1 3) hcpxxx_SetCmdPulse() 指令パルスの出力形式の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の偏差カウンタクリア出力を設定します。												
書式	short hcpxxx_SetCmdPulse(DWORD hDevID, short axis, short cmdpls);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ short cmdpls; /*指令パルスの出力形式[0:個別指令, 1:共通指令]*/												
(1 4) hcpxxx_SetAccProfile() S字/直線加減速の切替													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の加減速形式を設定します。												
書式	short hcpxxx_SetAccProfile(DWORD hDevID, short axis, WORD accpro);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD accpro; /*加減速形式 [0:直線, 1:S字]*/												
(1 5) hcpxxx_SetAutoDec() 減速開始点の設定方式の自動/手動切替													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の減速開始点の設定方式を設定します。												
書式	short hcpxxx_SetAutoDec(DWORD hDevID, short axis, WORD para);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD para; /*減速開始点の設定方式 [0:自動設定, 1:マニュアル設定]*/												
(1 6) hcpxxx_mSetDls() 汎用入力／±DR／PCS／DLS入力選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	×	×	×	×	×	×	×	×	×	×	×	×	○
機能	指定されたC P Dボード・汎用入力／±DR／PCS／DLS入力選択と極性選択, DLS選択時の停止方法を設定												
書式	short hcpxxx_mSetDls(short hDevID, short para, short pol, short motion, short latch);												
引数	short hDevID; /*デバイスハンドル*/ short para; /*汎用入力／±DR／PCS／DLS入力選択*/ short pol; /*入力極性*/ short motion; /*DLS入力時の動作*/ short latch; /*DLS入力時のラッチ*/												
(1 7) hcpxxx_mSetLtc() 汎用入力／LATCH入力の選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	×	×	×	×	×	×	×	×	×	×	×	×	○
機能	指定されたC P Dボード・軸の汎用入力／LATCH入力と極性を設定します。												
書式	short hcpxxx_mSetLtc(short hDevID, short axis, short para, short pol);												
引数	short hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-3)*/ short para; /*汎用入力(0)/LATCH(1)入力の切替*/ short pol; /*LATCH入力極性 [0:B接, 1:A接]												

(1 8) hcpxxx_mSetInpos() I N P O S入力／汎用入力の選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	x	x	x	x	x	x	x	x	x	x	x	x	○
機能	指定されたCPDボード・軸のI N P O S入力／汎用入力と極性を設定します。												
書式	short hcpxxx_mSetInpos(short hDevID, short axis, short para, short pol);												
引数	short hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-3)*/ short para; /*汎用入力(1)/INPOS(0)入力の切替*/ short pol; /*INPOS入力極性 [0: B接, 1: A接]												
(1 9) hcpxxx_mSetSvCtrCl() 汎用出力／SVCTRCL出力選択, 設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	x	x	x	x	x	x	x	x	x	x	x	x	○
機能	指定されたCPDボード・軸のI N P O S入力／汎用入力と極性を設定します。												
書式	short hcpxxx_mSetSvCtrCl(short hDevID, short axis, short para, short enbl);												
引数	short hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-3)*/ short para; /*SVCTRCL出力(0)/汎用出力(1)切替*/ short enbl; /*自動出力設定*/ /* [0: 不使用, 1: 原点完了時, 2: 異常停止時, 3: 原点完了及び異常停止時] */												

■状態読出し

(2 0) hcpxxx_ReadMainSts() メインステータスの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	x	x	x	x	○	○
機能	指定されたC P Dボード・軸のメインステータスを読出し, 指定したエリアに格納します。												
書式	short hcpxxx_ReadMainSts(DWORD hDevID, short axis, WORD* msts);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD* msts; /*読出されたデータが格納されるエリアのアドレス*/												
(2 1) hcpxxx_ReadErrorSts() エラーステータスの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	x	x	x	x	○	○
機能	指定されたC P Dボード・軸のエラーステータスを読出し, 指定したエリアに格納します。												
書式	short hcpxxx_ReadErrorSts(DWORD hDevID, short axis, DWORD* ests);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD* ests; /*読出されたデータが格納されるエリアのアドレス*/												
(2 2) hcpxxx_ReadEventSts() イベントステータスの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	x	x	x	x	○	○
機能	指定されたC P Dボード・軸のイベントステータスを読出し, 指定したエリアに格納します。												
書式	short hcpxxx_ReadEventSts(DWORD hDevID, short axis, DWORD* ists);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD* ists; /*読出されたデータが格納されるエリアのアドレス*/												
(2 3) hcpxxx_ReadSubSts() サブステータスの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	x	x	x	x	○	○
機能	指定されたC P Dボード・軸のサブステータスを読出し, 指定したエリアに格納します。												
書式	short hcpxxx_ReadSubSts(DWORD hDevID, short axis, WORD* ssts);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD* ssts; /*読出されたデータが格納されるエリアのアドレス*/												

(2 4) hcpxxx_ReadExSts () 拡張ステータスの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定された C P D ボード・軸の拡張ステータスを読出し、指定したエリアに格納します。												
書式	short hcpxxx_ReadExSts(DWORD hDevID, short axis, DWORD* exsts);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定 (0-11)*/ DWORD* exsts; /*読出されたデータが格納されるエリアのアドレス*/												
(2 5) hcpxxx_ReadIpSts () 補間ステータスの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	×	×	×	○	○	×	×	×	×	○	○
機能	指定された C P D ボード・軸の補間ステータスを読出し、指定したエリアに格納します。												
書式	short hcpxxx_ReadIpSts(DWORD hDevID, DWORD* ipsts);												
引数	DWORD hDevID; /*デバイスハンドル*/ DWORD* ipsts; /*読出されたデータが格納されるエリアのアドレス*/												
(2 6) hcpxxx_ReadSpd () 現在速度の読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定された C P D ボード・軸の速度データを読出し、指定したエリアに格納します。												
書式	short hcpxxx_ReadSpd(DWORD hDevID, short axis, WORD* spd);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定 (0-11)*/ WORD* spd; /*読出されたデータが格納されるエリアのアドレス*/												
(2 7) hcpxxx_ReadCtr () カウンタの読出し													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定された C P D ボード・軸・カウンタ値を読出し、指定したエリアに格納します。												
書式	short hcpxxx_ReadCtr(DWORD hDevID, short axis, WORD selctr, long* count);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定 (0-11)*/ WORD* selctr; /*カウンタ選択 [1:カウンタ 1, 2:カウンタ 2, 3:カウンタ 3, 4:カウンタ 4]*/ long* count; /*読出されたデータが格納されるエリアのアドレス*/												

■動作設定

(2 8) hcpxxx_SetFLSpd () ベース速度の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のベース速度を設定します。												
書式	short hcpxxx_SetFLSpd (DWORD hDevID, short axis, DWORD spd) ;												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定 (0-11)*/ DWORD spd; /*ベース速度 (FL) データ*/												

(2 9) hcpxxx_SetAuxSpd () 補助速度の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の補助速度を設定します。												
書式	short hcpxxx_SetAuxSpd (DWORD hDevID, short axis, DWORD spd) ;												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定 (0-11)*/ DWORD spd; /*補助速度 (FA) データ*/												

(3 0) hcpxxx_SetAccRate() 加速レートの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の加速レートを設定します。												
書式	short hcpxxx_SetAccRate(DWORD hDevID, short axis, DWORD rate);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD rate; /*加速レート[1~65535]*/												
(3 1) hcpxxx_SetDecRate() 減速レートの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の減速レートを設定します。												
書式	short hcpxxx_SetDecRate(DWORD hDevID, short axis, DWORD rate);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD rate; /*減速レート[1~65535]*/												
(3 2) hcpxxx_SetMult() 倍率設定値(MG)の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の速度の倍率設定値を設定します。												
書式	short hcpxxx_SetMult(DWORD hDevID, short axis, DWORD rmg);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD rmg; /*倍率設定値[2~4095]*/												
(3 3) hcpxxx_SetEventMask() イベントマスクの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のイベントマスクを設定します。												
書式	short hcpxxx_SetEventMask(DWORD hDevID, short axis, DWORD mask);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD mask; /*イベントマスクデータ*/												
(3 4) hcpxxx_SetDecPoint() 減速開始点の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の減速開始点を手動設定とした時の減速開始点を設定します。												
書式	short hcpxxx_SetDecPoint(DWORD hDevID, short axis, long dstnc);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ long dstnc; /*減速開始点(残り移動量)*/												

■運用設定

(3 5) hcpxxx_WritOpeMode() 動作モードの設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の動作モードを設定します。												
書式	short hcpxxx_WritOpeMode(DWORD hDevID, short axis, WORD mode);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ WORD mode; /*動作モード*/												

(3 6) hcpxxx_WritFHSpd() 動作速度の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の動作速度を設定します。												
書式	short hcpxxx_WritFHSpd(DWORD hDevID, short axis, DWORD spd);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ DWORD rmg; /*F H (動作速度) レジスタ値*/												
(3 7) hcpxxx_WritPos() 位置決め移動量の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の位置決め移動量を設定します。												
書式	short hcpxxx_WritPos(DWORD hDevID, short axis, long dstnc);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ long dstnc; /*移動量*/												
(3 8) hcpxxx_WritLine() 直線補間の移動量の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の直線補間の移動量を設定します。												
書式	short hcpxxx_WritLine(DWORD hDevID, short axis, long dstnc);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ long dstnc; /*移動量*/												
(3 9) hcpxxx_WritCircI() 円弧補間の移動量の設定													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸の円弧補間のデータを設定します。												
書式	short hcpxxx_WritCircI(DWORD hDevID, short axis, long dstnc1, long dstnc2, long center1, long center2)												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*円弧補間軸*/ /*[0:X, X1-Y, Y1 軸, 1:X, X1-Z, Z1 軸, 2:X, X1-U, U1 軸, 3:Y, Y1-Z, Z1 軸, 4:Y, Y1-U, U1 軸, 5:Z, Z1-U, U1 軸, 6:V, X2-W, Y2 軸, 7:V, X2-A, Z2 軸, 8:V, X2-B, U2 軸, 9:W, Y2-A, Z2 軸, 10:W, Y2-B, U2 軸, 11:A, Z2-B, U2 軸, 12:X3-Y3 軸, 13:X3-Z3 軸, 14:X3-U3 軸, 15:Y3-Z3 軸, 16:Y3-U3 軸, 17:Z3-U3 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*/												
(4 0) hcpxxx_WritCtr() カウンタプリセット													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボード・軸のカウンタをプリセットします。												
書式	short hcpxxx_WritCtr(DWORD hDevID, short axis, long preset, WORD selctr);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(0-11)*/ long preset; /*プリセット値*/ WORD selctr; /*カウンタ選択[1:カウンタ 1, 2:カウンタ 2, 3:カウンタ 3, 4:カウンタ 4]*/												

■動作制御指令

(4 1) hcpxxx_DecStop() 減速停止													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して減速停止指令を行います。(4軸単位)												
書式	short hcpxxx_DecStop(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(4 2) hcpxxx_QuickStop() 即停止													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して即停止指令を行います。(4軸単位)												
書式	short hcpxxx_QuickStop(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(4 3) hcpxxx_EmgStop() 非常停止													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して非常停止指令を行います。(4軸単位)												
書式	short hcpxxx_EmgStop(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(4 6) hcpxxx_AccStart() 加速スタート													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して同時に加速スタートを行います。												
書式	short hcpxxx_AccStart(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(4 7) hcpxxx_CnstStartFH() F H定速スタート													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して同時にF H定速スタートを行います。												
書式	short hcpxxx_CnstStartFH(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(4 8) hcpxxx_CnstStartFL() F L定速スタート													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して同時にF L定速スタートを行います。												
書式	short hcpxxx_CnstStartFL(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												

(4 9) hcpxxx_CnstStartByDec() F H定速スタート後減速停止													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対して同時にF H定速スタート後減速停止を行います。												
書式	short hcpxxx_ConstStartByDec(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(5 0) hcpxxx_SvOn() サーボオン													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対してサーボオンを行います。												
書式	short hcpxxx_SvOn(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(5 1) hcpxxx_SvOff() サーボオフ													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対してサーボオフを行います。												
書式	short hcpxxx_SvOff(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(5 2) hcpxxx_SvResetOn() サーボリセットオン													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対してサーボリセットオンを行います。												
書式	short hcpxxx_SvResetOn(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(5 3) hcpxxx_SvResetOff() サーボリセットオフ													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	×	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対してサーボリセットオフを行います。												
書式	short hcpxxx_SvResetOff(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												
(5 4) hcpxxx_PMOOn() パルスモータ励磁オン													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	指定されたC P Dボードの複数軸に対してパルスモータ励磁オンを行います。												
書式	short hcpxxx_PMOOn(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												

(55) hcpxxx_PMOff() パルスモータ励磁オフ													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	×	×	×	×	×	×	×	×	×	×	×	×	○
機能	指定されたC P Dボードの複数軸に対してパルスモータ励磁オフを行います。												
書式	short hcpxxx_PMOff(DWORD hDevID, short axis);												
引数	DWORD hDevID; /*デバイスハンドル*/ short axis; /*軸指定(b11:U3 b10:Z3, b9:Y3, b8:X3, b7:B, U2, b6:A, Z2, b5:W, Y2, b4:V, X2 b3:U, U1, b2:Z, Z1, b1:Y, Y1, b0:X, X1) */												

■汎用D I O

(56) hcpxxx_mGetOptInp() オプションポート設定の汎用入力ポートの確認													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	×	×	×	×	×	×	×	×	×	×	×	×	○
機能	指定されたC P Dボード・オプションポート設定の汎用入力ポートの確認を行います。												
書式	short hcpxxx_mGetOptInp(short hDevID, short* optinp);												
引数	DWORD hDevID; /*デバイスハンドル*/ short* optinp; /*汎用入力ポートのビット情報の格納エリアのアドレス*/												

■加減速レート計算

(57) hcpxxx_CalAccRate() 加減速レートの計算													
対応ボード	532	534	578	508	5212M	234	278	434	434v2	734	738	132	364
	○	○	○	○	○	○	○	×	×	×	×	○	○
機能	加減速時間, F H, F L等より加減速レートを計算し, 指定したエリアに格納します。												
書式	short hcpxxx_CalAccRate(DWORD* rate, DWORD dwtime, DWORD fh, DWORD fl, WORD pro, DWORD s)												
引数	DWORD rate; /*計算結果が格納されるエリアのアドレス*/ DWORD dwtime; /*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] */ DWORD s; /*通常は0, 直線部分ありのS字加減速ではS字範囲速度値*/ ※s=(FH-FL)/2の場合は直線部分なし												

3. ドライバ関数

ドライバ関数には、WinXP、Win2K、WinNT、Win98、DOSにおいて、CPDボードの制御を行うための関数が含まれます。

各関数は“VC++(5.0以上)”, “VB(5.0/6.0/.NET)”, “DOS版C言語”から外部関数として起動されます。

3. 1 ドライバ関数の種類

ドライバ関数は次表に示す13種類です。

No	関 数 名	ボード種別 (※1)							機 能
		HPC1-		HPC-	HUSB-	HPC1-	HPC104-	HP104D	
		CPD534 CPD532 CPD578 CPD508	CPD 5212M	CPD234 CPD278 (※2)	CPD434 CPD434v2 (※3)	CPD738 CPD734 (※4)	CPD132 (※2)	CPD364 (※2)	
	関数名“xxx”	530	52c	230	430	730	130	360	
1	cpxxx_GetDeviceCount()	○	○	×	○	○	×	×	ボード枚数の取得
2	cpxxx_GetDeviceInfo()	○	○	×	○	○	×	×	デバイス情報の取得
3	cpxxx_OpenDevice()	○	○	○	○	○	○	○	デバイスのオープン
4	cpxxx_CloseDevice()	○	○	○	○	○	○	○	デバイスのクローズ
5	cpxxx_rMstslW()	○	○	○	○	○	○	○	メインステータスの読出し
6	cpxxx_rSstslW()	○	○	○	○	○	○	○	サブステータスの読出し
7	cpxxx_wCmdW()	○	○	○	○	○	○	○	制御コマンドの書込
8	cpxxx_rReg()	○	○	○	○	○	○	○	レジスタの読出し
	cpxxx_wReg()	○	○	○	○	○	○	○	レジスタへ書込
9	cpxxx_rPortB()	○	○	○	○	○	○	○	オプションポートの読出し
	cpxxx_wPortB()	○	○	○	○	○	○	○	オプションポートへ書込
	cpxxx_rPortW()	×	○	○	×	×	×	○	オプションポートの読出し
	cpxxx_wPortW()	×	○	○	×	×	×	○	オプションポートへ書込
10	cpxxx_rBufDW()	○	○	○	○	○	○	○	入出力バッファの読出し
	cpxxx_wBufDW()	○	○	○	○	○	○	○	入出力バッファへ書込
11	cpxxx_SetIntCall() ※5	○	○	×	×	×	×	×	割込処理関数の登録・削除
12	cpxxx_GetDevVerNo() ※5	○	○	×	×	×	×	×	バージョン番号の取得
13	cpxxx_GetBoardCode()	○	×	×	×	×	×	×	ボード固有コードの取得

※1. ボード種別とドライバ関数の対応・○：対応，×：非対応（具体的には個々の関数を参照下さい。）

※2. 対応するOSは“WinVista”を除きます。

※3. 対応するOSは“WinVista”，“WinXP”，“Win2K”と“Win98”です。

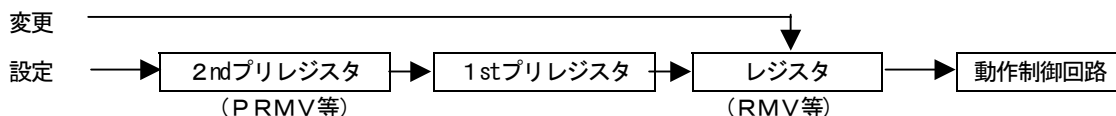
※4. 対応するOSは“WinVista”，“Win95”と“DOS”を除きます。

※5. この2種類のドライバ関数はDOS版専用です。

(1) プリレジスタ

RMV, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RUS, RDS, RCMP5レジスタとスタートコマンドにはプリレジスタがあります。

プリレジスタとは、動作中に次の動作用データをセットしておくレジスタで、PCLのプリレジスタは下図のような二段階構成になっています。



動作用データは「プリレジスタ（2ndプリレジスタ）」に書込みます。

速度変更、位置のオーバライド等、現在の動作状態を変更する場合には「レジスタ」に新データを書込みます。

(2) ドライバ関数の引数データ型、16進数表記の対応

言語	VC++言語	VB5.0/6.0	VB.NET	DOS版C言語
データ型	BYTE	Byte	Byte	unsigned char
	WORD	Integer	Short	unsigned short
	DWORD	Long	Integer	unsigned long
データ例	0x0000	&H0	&H0	0x0000
	0x1000	&H1000	&H1000	0x1000

(3) ライブラリ関数の戻り値

関数の起動を行った結果は「戻り値」に実行結果が反映されます。

この戻り値は（ライブラリ関数および）ドライバ関数で共通です。

戻り値が0の場合は正常終了ですが、'0'以外の場合には何らかの異常が発生しています。

「1. 3関数の戻り値（P7）」を参照して下さい。

3. 2 ドライバ関数の詳細

(1) WinVB関数書式の“libname”は使用されるボードのドライバI/F用DLLのファイル名で置き換えてください。

■ cpxxx_rMstsW(), cpxxx_rSstsW(), cpxxx_wCmdW(), cpxxx_rReg(), cpxxx_wReg() の関数の軸指定は
2 軸ボード (HPCI-CPD532, HPCI104-CPD132) では

0 : X 軸, 1 : Y 軸

4 軸ボード (HPCI-CPD534, HPC-CPD234, HP104D-CPD364, HUSB-CPD434, HCPCI-CPD734) では

0 : X 軸, 1 : Y 軸, 2 : Z 軸, 3 : U 軸

8 軸ボード (HPCI-CPD578, HPCI-CPD508, HPC-CPD278, HCPCI-CPD738) では

0 : X 軸, 1 : Y 軸, 2 : Z 軸, 3 : U 軸,

4 : V 軸, 5 : W 軸, 6 : A 軸, 7 : B 軸

12 軸ボード (HPCI-CPD5212M) では

0 : X1 軸, 1 : Y1 軸, 2 : Z1 軸, 3 : U1 軸,

4 : X2 軸, 5 : Y2 軸, 6 : Z2 軸, 7 : U2 軸

8 : X3 軸, 9 : Y3 軸, 10 : Z3 軸, 11 : U3 軸

となります。

(1) cpxxx_GetDeviceCount() ボード枚数の取得							
関数名 (xxx)	530	52C	230	434	730	130	360
	○	○	×	○	○	×	×
機能 現在パソコンに装着されているCPDボードの枚数を取得します。							
Win VC++	書式	DWORD WINAPI cpxxx_GetDeviceCount(DWORD* count);					
	引数	DWORD* count : 取得したCPDボード枚数の格納エリア					
	呼出例	DWORD* count; //CPDボードの枚数 DWORD ret; //関数の戻り値 ret = cpxxx_GetDeviceCount(&count);					
Win VB	書式	Declare Function cpxxx_GetDeviceCount Lib "libname" (ByRef count As Long) As Long					
	引数	ByRef count As Long : 取得したCPDボード枚数の格納エリア					
	呼出例	Dim count As Long ' CPDボードの枚数 Dim ret As Long ' 関数の戻り値 ret = cpxxx_GetDeviceCount(count)					
関数名 (xxx)	530	52C	230	434	730	130	360
	○	○	×	×	×	×	×
DOS	書式	short cpxxx_GetDeviceCount(short* count);					
	引数	short* count : 取得したCPDボード枚数の格納エリア					
	呼出例	short ans, count; ans = cpxxx_GetDeviceCount(&count);					

(2) cpxxx_GetDeviceInfo() デバイス情報の取得								
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	×	○	○	×	×
機能		現在パソコンに装着されている指定枚数CPDボードのデバイス情報を取得します。 この結果、デバイス情報構造体の配列にデバイス情報が格納されます。 このデバイス情報は、デバイスオープン時に利用します。						
Win VC++	書式	DWORD WINAPI cpxxx_GetDeviceInfo(DWORD*count, HPCDEVICEINFO* HpcDevInfo);						
	引数	DWORD* count ・ ・ CPDボードの枚数 (デバイス情報取得ボード枚数) HPCDEVICEINFO* HpcDevInfo ・ ・ 取得するデバイス情報がセットされるエリア先頭アドレス						
	呼出例	DWORD ret;						

(3) cpxxx_OpenDevice()		デバイスのオープン						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
機能		渡した情報を持つ C P D ボードをオープンし、他と識別するためのデバイスハンドルを取得します。以降このデバイスハンドルは、この C P D ボードにアクセスするためのハンドルとなります。						
Win VC++	書式	DWORD WINAPI cpxxx_OpenDevice(DWORD* hDevID, HPCDEVICEINFO* HpcDevInfo);						
	引数	DWORD hDevID; ・ ・ 取得するデバイスハンドルの格納エリア HPCDEVICEINFO* HpcDevInfo; ・ ・ オープンするボードのデバイス情報格納アドレス						
	呼出例	DWORD ret; //関数の戻り値 DWORD hDevID[2]; //デバイスハンドル取得エリア ret = cpxxx_OpenDevice(&hDevID[0], &HpcDevInfo[0]); // 1番目のデバイス情報 ret = cpxxx_OpenDevice(&hDevID[1], &HpcDevInfo[1]); // 2番目のデバイス情報						
Win VB	書式	Declare Function cpxxx_OpenDevice Lib "libname" (ByVal hDevID As Long, _ HpcDevInfo As HPCDEVICEINFO) As Long						
	引数	ByRef hDevID As Long ・ ・ 取得するデバイスハンドルの格納エリア HpcDevInfo As HPCDEVICEINFO ・ ・ オープンするボードのデバイス情報格納アドレス						
	呼出例	Dim ret As Long '関数の戻り値 Dim hDevID(2) As Long 'デバイスハンドル取得エリア ret = cpxxx_OpenDevice(hDevID(0), HpcDevInfo(0)) ' 1番目のデバイス情報 ret = cpxxx_OpenDevice(hDevID(1), HpcDevInfo(1)) ' 2番目のデバイス情報						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
DOS	書式	short cpxxx_OpenDevice(DWORD* hDevID, HPCDEVICEINFO* hpcDevInfo);						
	引数	DWORD* hDevID; ・ ・ 取得するデバイスハンドルの格納エリア HPCDEVICEINFO* hpcDevInfo; ・ ・ オープンするボードのデバイス情報格納アドレス						
	呼出例	short ans; DWORD hDevID[2]; /*2枚分のデバイスハンドル格納エリア*/ HPCDEVICEINFO* hpcDevInfo[2]; /*2枚分のデバイス情報格納エリア*/ ans = cpxxx_OpenDevice(&hDevID[0], hpcDevInfo[0]); /* 1枚目ボード*/ ans = cpxxx_OpenDevice(&hDevID[1], hpcDevInfo[1]); /* 2枚目ボード*/						

(4) cpxxx_CloseDevice()		デバイスのクローズ						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
機能		渡したデバイスハンドルを持つCPDボードをクローズします。 以降このデバイスハンドルは、無効となり、このCPDボードにアクセスはできません。						
Win VC++	書式	DWORD WINAPI cpxxx_CloseDevice(DWORD hDevID);						
	引数	DWORD hDevID・・・クローズするボードのデバイスハンドル						
	呼出例	DWORD ret; //関数の戻り値 ret = cpxxx_CloseDevice(hDevID);						
Win VB	書式	Declare Function cpxxx_CloseDevice Lib"libname" (ByVal hDevID As Long) As Long						
	引数	ByVal hDevID As Long・・・クローズするボードのデバイスハンドル						
	呼出例	Dim ret As Long ' 関数の戻り値 ret = cpxxx_CloseDevice(hDevID)						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
DOS	書式	short cpxxx_CloseDevice(DWORD hDevID);						
	引数	DWORD hDevID・・・クローズするボードのデバイスハンドル						
	呼出例	short ans; ans = cpxxx_CloseDevice(hDevID[0]); /*1 枚目ボード*/						
備考		デバイスクローズの前に必ずパルス停止などの終了処理を行ってください。						

(5) cpxxx_rMstsW()		メインステータスの読出し						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
機能		デバイスハンドルの指定されたボードの指定軸メインステータスを讀出し、指定エリアに格納						
Win VC++	書式	DWORD WINAPI cpxxx_rMstsW(DWORD hDevID, WORD axis, WORD* wMsts);						
	引数	DWORD hDevID ・・・対象デバイスのデバイスハンドル WORD axis ・・・軸指定 [0-11] WORD* wMsts ・・・讀出されたデータが格納されるエリアのアドレス						
	呼出例	DWORD ret; //関数の戻り値 WORD msts; //メインステータス ret = cpxxx_rMstsW(hDevID, 1, &msts); //Y軸						
Win VB	書式	Declare Function cpxxx_rMstsW Lib "libname" (ByVal hDevID As Long, _ ByVal axis As Integer, ByRef wMsts As Integer) As Long						
	引数	ByVal hDevID As Long ・・・対象デバイスのデバイスハンドル ByVal axis As Integer ・・・軸指定 [0-11] ByRef wMsts As Integer ・・・讀出されたデータが格納されるエリアのアドレス						
	呼出例	Dim ret As Long '関数の戻り値 Dim msts As Integer 'メインステータス ret = cpxxx_rMstsW(hDevID, 1, msts) 'Y軸						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
DOS	書式	short cpxxx_rMstsW(DWORD hDevID, short axis, WORD* wMsts);						
	引数	DWORD hDevID ・・・対象デバイスのデバイスハンドル short axis ・・・軸指定 [0-11] WORD* wMsts ・・・讀出されたデータが格納されるエリアのアドレス						
	呼出例	short ans; WORD msts; //メインステータス ans = cpxxx_rMstsW(hDevID, 0, &msts); /*X軸*/						
備考		ステータスの監視は常に対象となる軸のMSTSに対して行います。MSTSポートから読み出します。 スタートコマンドの発行後はMSTSを常にポーリングします。移動終了はb 5またはb 4のみを監視し、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 'となり、イベントステータスの処理をします。(※1) 異常終了ではb 4が' 1 'となり、エラーステータスの処理をします。						

【 メインステータスの内容 】

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SPDF	SPRF	0	SCMPx					SSCx		SINT	SERR	SEND	0	SRUN	SSCM
			5	4	3	2	1	1	0						
ビット	名称	説明													運用
0	SSCM	'1': スタート指令が書込まれた。													△
1	SRUN	'1': RUN中													△
3	SEND	'1': 停止状態 ※2													◎
4	SERR	'1': エラー報告あり (エラーステータス (REST) 読出しで'0') ※3													◎
5	SINT	'1': イベント報告あり (イベントステータス (RIST) 読出しで'0') ※3													◎
6, 7	SSCx	実行中または停止中のシーケンス番号 (PRMD設定値)													△
8~12	SCMPx	'1': CMP x比較条件成立時													△
14	SPRF	'1': 次動作プリレジスタが満杯 ('0': 書込可能)													△
15	SPDF	'1': CMP 5用プリレジスタが満杯 ('0': 書込可能)													△

運用欄・・・◎: 基本的運用, △: アプリで任意運用

- ※1. 正常終了でのイベント報告は”イベントマスク設定: 自動停止” [b0: ISEN=1] とします。
- ※2. b3 (SEND) は状態を示しているビットです。電源投入直後は'0'であり、即 (減速) 停止指令の実行または一度移動実行後の終了状態は'1'となります。移動中は'0'を示します。
通常停止中(='1')か、動作中(='0')かを確認したいときに使用します。
- ※3. b4 (SERR), b5 (SINT) はCPUへの割込要因となります。
この設定を行う為には、各軸の環境設定1 (RENV1: b29: INTM=0) とボードのオプションポート: ボード割込マスク設定 (BINTM=1) を行います。

(6) cpxxx_rSstsW()		サブステータスの読出し						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
機能		デバイスハンドルで指定されたボードの指定軸サブステータスを読出し、指定エリアに格納						
Win VC++	書式	DWORD WINAPI cpxxx_rMstsW(DWORD hDevID, WORD axis, WORD* wSsts);						
	引数	DWORD hDevID・・・対象デバイスのデバイスハンドル WORD axis・・・軸指定 [0-11] WORD* wSsts・・・読出されたデータが格納されるエリアのアドレス						
	呼出例	DWORD ret; //関数の戻り値 WORD ssts; //サブステータス ret = cpxxx_rSstsW(hDevID, 1, &ssts); //Y軸						
Win VB	書式	Declare Function cpxxx_rSstsW Lib "libname" (ByVal hDevID As Long, _ ByVal axis As Integer, ByRef wSsts As Integer) As Long						
	引数	ByVal hDevID As Long・・・対象デバイスのデバイスハンドル ByVal axis As Integer・・・軸指定 [0-11] ByRef wSsts As Integer・・・読出されたデータが格納されるエリアのアドレス						
	呼出例	Dim ret As Long '関数の戻り値 Dim ssts As Integer 'サブステータス ret = cpxxx_rSstsW(hDevID, 1, ssts) 'Y軸						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
DOS	書式	short cpxxx_rSMstsW(DWORD hDevID, short axis, WORD* wSsts);						
	引数	DWORD hDevID・・・対象デバイスのデバイスハンドル short axis・・・軸指定 [0-11] WORD* wSsts・・・読出されたデータが格納されるエリアのアドレス						
	呼出例	short ans; WORD ssts; ans = cpxxx_rSstsW(hDevID, 0, &ssts); /*X軸*/						

【 サブステータスの内容 】

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SDLS	SOLS	SMEL	SPEL	SALM	SFC	SFD	SFU	0	0	0	0	0	0	SVRST	SVON
ビット	名 称	説 明												運 用	
0	SVON	'1': "SVON"出力中												△	
1	SVRST	'1': "SVRST"出力中												△	
8	SFU	'1': 加速中												×	
9	SFD	'1': 減速中												×	
10	SFC	'1': 定速動作中												×	
11	SALM	'1': SVALM (サーボアラーム信号ON中)												△	
12	SPEL	'1': +ELS検出中												△	
13	SMEL	'1': -ELS検出中												△	
14	SOLS	'1': OLS検出中												△	
15	SDLS	'1': DLS検出中												△	

運用欄・・・△：アプリで任意運用，×：普通は不使用

- (注) 1. ビット0 (SVON)，ビット1 (SVRST) はサーボ出力指令のモニタビットです。
2. ビット12 (SALM)が'1'の場合，動作方向のELS検出中 (SPEL, SMEL) の軸に対する動作指令は異常終了となります。
3. 両方向のELSが共に検出中 (SPEL=SMEL='1') となる場合は，オプションポートのELS極性の設定 (A接/B接) が逆となっている事が考えられます。

(7) cpxxx_wCmdW()		制御コマンド書込み						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
機能		デバイスハンドルで指定されたボードの指定軸コマンドバッファへ制御コマンドデータを書込む						
Win VC++	書式	DWORD WINAPI cpxxx_wCmdW(DWORD hDevID, WORD axis, WORD wCmd);						
	引数	DWORD hDevID・・・対象デバイスのデバイスハンドル WORD axis・・・軸指定 [0-11] WORD wCmd・・・コマンドデータ						
	呼出例	DWORD ret;//関数の戻り値 ret = cpxxx_wCmdW(hDevID, 1, 0x4a); //Y軸減速停止						
Win VB	書式	Declare Function cpxxx_wCmdW Lib "libname"(ByVal hDevID As Long, _ ByVal axis As Integer, ByVal wCmd As Integer) As Long						
	引数	ByVal hDevID As Long・・・対象デバイスのデバイスハンドル ByVal axis As Integer・・・軸指定 [0-11] ByRef wCmd As Integer・・・コマンドデータ						
	呼出例	Dim ret As Long '関数の戻り値 ret = cpxxx_wCmdW(hDevID, 1, &H4A); 'Y軸減速停止						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
DOS	書式	short cpxxx_wCmdW(DWORD hDevID, short axis, WORD cmd);						
	引数	DWORD hDevID・・・対象デバイスのデバイスハンドル short axis・・・軸指定 [0-11] WORD cmd・・・制御コマンドデータ						
	呼出例	short ans; ans = cpxxx_wCmdW(hDevID, 1, 0x4a); /*Y軸減速停止*/						

【 コマンドデータについて 】

(1) コマンドデータの内容

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	コマンド実行軸 (SELx)				コマンドコード (code)							

	b11	b10	b9	b8	書き込む軸
	コマンド実行軸 (SELx)				
2軸	0	0	Y	X	X
4軸	U	Z	Y	X	X
8軸	U	Z	Y	X	X
	B	A	W	V	V
12軸	U1	Z1	Y1	X1	X1
	U2	Z2	Y2	X2	X2
	U3	Z3	Y3	X3	X3

(2) 実行軸の指定 (SELx)

個々の軸毎にコマンドコードを書込む場合はこの4ビットは'0'、2軸以上に同一コマンドコードを書込む場合は、X(X1)軸(axis=0)、V(X2)軸(axis=4)、X3軸(axis=8)に対して、SELxのビットで書込む軸を指定。

(3) コマンドコード一覧表 (詳細はユーザーズマニュアル<共通編>)

code	コマンド内容	code	コマンド内容
0x05	非常停止	0x29	ラッチ入力代行
0x06	CSTA出力 (同時スタート)	0x2a	自軸のみ、CSTA入力と同じ
0x07	CSTP出力 (同時ストップ)	0x2b	動作用ブリレジスタのシフト
0x10	SVONOFF	0x2c	RCMP5用ブリレジスタのシフト
0x18	SVNON	0x40	直ちにFL速度へ変更
0x11	SVRESETOFF	0x41	直ちにFH速度へ変更
0x19	SVRESETON	0x42	減速してFL速度へ変更
0x04	ソフトウェアリセット	0x43	加速してFH速度へ変更
0x20	カウンタ1リセット	0x49	即停止
0x21	カウンタ2リセット	0x4a	減速停止
0x22	カウンタ3リセット	0x50	FL定速スタート
0x23	カウンタ4リセット	0x51	FH定速スタート
0x24	偏差カウンタクリア信号の出力	0x52	FH定速継続スタート後減速停止
0x25	偏差カウンタクリア信号のリセット	0x53	高速スタート
0x26	動作用ブリレジスタのキャンセル	0x54	残量FL定速スタート
0x27	RCMP5用ブリレジスタのキャンセル	0x55	残量FH定速スタート
0x28	位置決め管理開始 (PCS入力代行)	0x57	残量高速スタート

(8) cpxxx_rReg() cpxxx_wReg()		レジスタ読出し レジスタ書込み						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
機能		デバイスハンドルで指定されたボードの、 読出・・指定軸の指定レジスタ内容を読出し、指定エリアに格納します。 書込・・指定軸の指定レジスタへ、データを書込みます。						
Win VC++	書式	DWORD WINAPI cpxxx_rReg(DWORD hDevID, WORD axis, BYTE byCmd, DWORD* dwReg); DWORD WINAPI cpxxx_wReg(DWORD hDevID, WORD axis, BYTE byCmd, DWORD dwReg);						
	引数	DWORD hDevID: ・・対象デバイスのデバイスハンドル WORD axis: ・・軸指定 [0-11] BYTE byCmd: ・・レジスタ読出／書込コマンド DWORD* dwReg: ・・レジスタ読出：読出データの格納エリアアドレス DWORD dwReg: ・・書込：レジスタ書込データ						
	呼出例	DWORD ret; //関数の戻り値 DWORD regstr: //レジスタのデータ ret = cpxxx_rReg(hDevID, 0, 0xc0, ®str); //X軸 PRMV 読出 ret = cpxxx_wReg(hDevID, 0, 0x80, 10000); //X軸 PRMV=10000 書込						
Win VB	書式	Declare Function cpxxx_rReg Lib "libname"(ByVal hDevID As Long, _ ByVal axis As Integer, ByVal byCmd As Byte, ByRef dwReg As Long) As Long Declare Function cpxxx_wReg Lib "libname"(ByVal hDevID As Long, _ ByVal axis As Integer, ByVal byCmd As Byte, ByVal dwReg As Long) As Long						
	引数	ByVal hDevID As Long ・・対象デバイスのデバイスハンドル ByVal axis As Integer ・・軸指定 [0-11] ByRef byCmd As Byte ・・レジスタ読出／書込コマンド ByRef dwReg As Long ・・レジスタ読出：読出データの格納エリアアドレス ByVal dwReg As Long ・・書込：レジスタ書込データ						
	呼出例	Dim ret As Long ' 関数の戻り値 Dim regstr As Long ' レジスタのデータ ret = cpxxx_rReg(hDevID, 0, &HC0, regstr) ' X軸 PRMV 読出 ret = cpxxx_wReg(hDevID, 0, &H80, 10000) ' X軸 PRMV=10000 書込						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
DOS	書式	short cpxxx_rReg(DWORD hDevID, short axis, BYTE cmd, DWORD* data); short cpxxx_wReg(DWORD hDevID, short axis, BYTE cmd, DWORD data);						
	引数	DWORD hDevID: ・・対象デバイスのデバイスハンドル short axis: ・・軸指定 [0-11] BYTE cmd: ・・レジスタ読出／書込コマンド DWORD* data: ・・レジスタ読出：読出データの格納エリアアドレス DWORD data: ・・書込：レジスタ書込データ						
	呼出例	short ans; DWORD data; ans = cpxxx_rReg(hDevID, 0, 0xc0, &data); /*X軸プリミティブ移動量の読出*/ ans = cpxxx_wReg(hDevID, 0, 0x80, data); /*X軸プリミティブ移動量の書込*/						
備 考		次ページにレジスタ読出／書込コマンド一覧						

【 レジスタ・プリレジスタ：読出コマンド・書込コマンド 】

No	内容	レジスタ			プリレジスタ		
		名称	コマンド		名称	コマンド	
			読出	書込		読出	書込
1	移動量、目標位置	RMV	0xd0	0x90	PRMV	0xc0	0x80
2	初速度	RFL	0xd1	0x91	PRFL	0xc1	0x81
3	動作速度	RFH	0xd2	0x92	PRFH	0xc2	0x82
4	加速レート	RUR	0xd3	0x93	PRUR	0xc3	0x83
5	減速レート	RDR	0xd4	0x94	PRDR	0xc4	0x84
6	速度倍率	RMG	0xd5	0x95	PRMG	0xc5	0x85
7	減速開始点	RDP	0xd6	0x96	PRDP	0xc6	0x86
8	動作モード	RMD	0xd7	0x97	PRMD	0xc7	0x87
9	円弧補間中心位置	RIP	0xd8	0x98	PRIP	0xc8	0x88
10	加速時S字区間	RUS	0xd9	0x99	PRUS	0xc9	0x89
11	減速時S字区間	RDS	0xda	0x9a	PRDS	0xca	0x8a
12	補助速度	RFA	0xdb	0x9b			
13	環境設定 1	RENV1	0xdc	0x9c			
14	環境設定 2	RENV2	0xdd	0x9d			
15	環境設定 3	RENV3	0xde	0x9e			
16	環境設定 4	RENV4	0xdf	0x9f			
17	環境設定 5	RENV5	0xe0	0xa0			
18	環境設定 6	RENV6	0xe1	0xa1			
19	環境設定 7	RENV7	0xe2	0xa2			
20	カウンタ 1（指令パルス出力）※ 1	RCUN1	0xe3	0xa3			
21	カウンタ 2（エンコーダ入力）※ 1	RCUN2	0xe4	0xa4			
22	カウンタ 3（偏差カウンタ）	RCUN3	0xe5	0xa5			
23	カウンタ 4（汎用カウンタ）	RCUN4	0xe6	0xa6			
24	コンパレータ 1 用データ	RCMP1	0xe7	0xa7			
25	コンパレータ 2 用データ	RCMP2	0xe8	0xa8			
26	コンパレータ 3 用データ	RCMP3	0xe9	0xa9			
27	コンパレータ 4 用データ	RCMP4	0xea	0xaa			
28	コンパレータ 5 用データ	RCMP5	0xeb	0xab	PRCP5	0xcb	0x8b
29	イベントマスク設定※ 3	RIRQ	0xec	0xac			
30	カウンタ 1 ラッチデータ	RLTC1	0xed				
31	カウンタ 2 ラッチデータ	RLTC2	0xee				
32	カウンタ 3 ラッチデータ	RLTC3	0xef				
33	カウンタ 4 ラッチデータ	RLTC4	0xf0				
34	拡張ステータス	RSTS	0xf1				
35	エラーステータス※ 5	REST	0xf2				
36	イベントステータス※ 4	RIST	0xf3				
37	位置決めカウンタ	RPLS	0xf4				
38	E Z カウンタ、速度モニタ※ 2	RSPD	0xf5				
39	スローダウンポイント	RSDC	0xf6				
40	円弧補間歩進数	RCI	0xfc	0xbc	PRCI	0xcc	0x8c
41	円弧補間歩進カウンタ	RCIC	0xfd				
42	補間ステータス	RIPS	0xff				

※ 1. カウンタ 1＝指令現在位置、カウンタ 2＝エンコーダフィードバック現在位置

※ 2. 動作速度のモニタ（下位 16 ビット）・・ [速度倍率を乗算して実速度]

※ 3. メインステータス：SINT＝1 とするイベントを設定

※ 4. 動作終了時のイベント発生内容確認ステータス（メインステータス：SINT＝1）

※ 5. 動作終了時のエラー終了内容の確認ステータス（メインステータス：SERR＝1）

(9) cpxxx_rPortB() cpxxx_wPortB()							
オプションポートバイト読出し				オプションポートバイト書込み			
関数名 (xxx)	530	52C	230	434	730	130	360
	○	○	○	○	○	○	○
cpxxx_rPortW() cpxxx_wPortW()							
オプションポートワード読出し				オプションポートワード書込み			
関数名 (xxx)	530	52C	230	434	730	130	360
	x	○	○	x	x	x	○
機 能		デバイスハンドルの指定されたボードの、 読込・・・指定オプションポート内容を読込み、指定エリアに格納します。 書込・・・指定オプションポートへ、データを書込みます。					
Win VC++	書 式	DWORD WINAPI cpxxx_rPortB(DWORD hDevID, BYTE port, BYTE* byData); DWORD WINAPI cpxxx_wPortB(DWORD hDevID, BYTE port, BYTE byData); DWORD WINAPI cpxxx_rPortW(DWORD hDevID, BYTE port, WORD* wData); DWORD WINAPI cpxxx_wPortW(DWORD hDevID, BYTE port, WORD wData);					
	引 数	DWORD hDevID・・・ボードのデバイスハンドル BYTE port・・・読込オプションポート/書込オプションポート BYTE* byData・・・ポート読込：読込データの格納エリアアドレス BYTE byData・・・書込：オプションポート書込データ WORD* wData・・・ポート読込：読込データの格納エリアアドレス WORD wData・・・書込：オプションポート書込データ					
	呼出例	DWORD ret; //関数の戻り値 BYTE byData; ret = cp530_rPortB(hDevID, 0x80, &byData); // 各軸E L S極性設定を読込 ret = cp530_wPortB(hDevID, 0x80, 0x01); // X軸のみA接に設定					
Win VB	書 式	Declare Function cpxxx_rPortB Lib "libname" _ (ByVal hDevID As Long, ByVal port As Byte, ByRef byData As Byte) As Long Declare Function cpxxx_wPortB Lib "libname" _ (ByVal hDevID As Long, ByVal port As Byte, ByVal byData As Byte) As Long Declare Function cpxxx_rPortW Lib "libname" _ (ByVal hDevID As Long, ByVal port As Byte, ByRef nData As Integer) As Long Declare Function cpxxx_wPortW Lib "libname" _ (ByVal hDevID As Long, ByVal port As Byte, ByVal nData As Integer) As Long					
	引 数	ByVal hDevID As Long・・・ボードのデバイスハンドル ByVal port As Byte・・・読込オプションポート/書込オプションポート ByRef byData As Byte・・・ポート読込：読込データの格納エリアアドレス ByVal byData As Byte・・・書込：オプションポート書込データ ByRef wData As Integer・・・ポート読込：読込データの格納エリアアドレス ByVal wData As Integer・・・書込：オプションポート書込データ					
	呼出例	Dim ret As Long '関数の戻り値 Dim byData As Byte ret = cp530_rPortB(hDevID, &H80, byData) '各軸E L S極性設定を読込 ret = cp530_wPortB(hDevID, &H80, &H1) 'X軸のみA接に設定					
関数名 (xxx)	530	52C	230	434	730	130	360
	○	○	○	x	x	x	○
DOS	書 式	short cpxxx_rPortB(DWORD hDevID, BYTE port, BYTE* data); short cpxxx_wPortB(DWORD hDevID, BYTE port, BYTE data);					
	引 数	DWORD hDevID;・・・ボードのデバイスハンドル BYTE port;・・・読込オプションポート/書込オプションポート BYTE* data;・・・ポートバイト読込：読込データの格納エリアアドレス BYTE data;・・・バイト書込：オプションポート書込データ					
	呼出例	short ans; BYTE data; ans = cp530_rPortB(hDevID, 0x80, &data); /* E L S 極性設定の読込 */ ans = cp530_wPortB(hDevID, 0x80, data); /* E L S 極性設定値の書込 */					
備考		オプションポート(p o r t)の値・・・個々のボード<個別編>を参照して下さい。					

(1 0) cpxxx_rBufDW () cpxxx_wBufDW ()		入出力バッファ読出し 入出力バッファ書込み						
機能		デバイスハンドルで指定されたボードの、 読出・・・指定軸の入出力バッファを読出し、指定エリアに格納します。 書込・・・指定軸の入出力バッファにデータを書込みます。						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	○	○	○	○
Win V C++	書式	DWORD WINAPI cpxxx_rBufDW(DWORD hDevID, WORD axis, DWORD* dwData); DWORD WINAPI cpxxx_wBufDW(DWORD hDevID, WORD axis, DWORD dwData);						
	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, 1:Y, 2:Z, 3:U, 4:V, 5:W, 6:A, 7:B] DWORD* dwData;・・・レジスタ読出：読出データの格納エリアアドレス DWORD dwData;・・・書込：レジスタ書込データ						
	呼出例	DWORD ret; //関数の戻り値 DWORD dwData; //入出力バッファデータ ret = cpxxx_rBufDW(hDevID, 0x01, &dwData); //Y軸入出力バッファから読出 ret = cpxxx_wBufDW(hDevID, 0x01, 10000); //Y軸入出力バッファへの書込						
Win V B	書式	Declare Function cpxxx_rBufDW Lib "libname" _ (ByVal hDevID As Long, ByVal axis As Integer, ByRef dwData As Long) As Long Declare Function cpxxx_wBufDW Lib "libname" _ (ByVal hDevID As Long, ByVal axis As Integer, ByVal dwData As Long) As Long						
	引数	ByVal hDevID As Long・・・対象デバイスのデバイスハンドル ByVal axis As Integer・・・軸指定 [0:X, 1:Y, 2:Z, 3:U, 4:V, 5:W, 6:A, 7:B] ByRef dwReg As Long・・・レジスタ読出：読出データの格納エリアアドレス ByVal dwReg As Long・・・書込：レジスタ書込データ						
	呼出例	Dim ret As Long '関数の戻り値 Dim dwData As Long '入出力バッファデータ ret = cpxxx_rBufDW(hDevID, &H1, dwData) 'Y軸入出力バッファから読出 ret = cpxxx_wBufDW(hDevID, &H1, 10000) 'Y軸入出力バッファへの書込						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	○	×	×	○	○
D O S	書式	short cpxxx_rBufDW(DWORD hDevID, short axis, DWORD* data); short cpxxx_wBufDW(DWORD hDevID, short axis, DWORD data);						
	引数	DWORD hDevID;・・・対象デバイスのデバイスハンドル short axis;・・・軸指定 [0:X, 1:Y, 2:Z, 3:U, 4:V, 5:W, 6:A, 7:B] DWORD* data;・・・レジスタ読出：読出データの格納エリアアドレス DWORD data;・・・書込：レジスタ書込データ						
	呼出例	short ans; DWORD data; ans = cpxxx_rBufDW(hDevID, 0, &data); /*X軸入出力バッファから読出*/ ans = cpxxx_wBufDW(hDevID, 0, data); /*X軸入出力バッファへの書込*/						
備考		<p>■レジスタ読出/書込関数との相違</p> <p>◆レジスタ読出/書込関数・・・PCLの指定入出力バッファを経由して目的レジスタを対象</p> <p>◆入出力バッファ操作関数・・・PCLの指定入出力バッファとの読出/書込みです。</p> <p>■レジスタ読出関数 (cpxxx_rBufDW ()) の応用</p> <p>複数軸のレジスタデータを同じタイミングで一括読出しを行います。(同一PCL内の軸)</p> <p>◇cpxxx_wCmdW () 関数の“コマンド”で複数軸の読出したいプリレジスタを指定。</p> <p>◎軸指定 (axis) はX軸 (0) , 制御コマンドデータ (cmd) 中のコマンド実行軸 (SELx) に 2軸以上設定, コマンドコード (code) に読出コマンドを設定</p> <p>◇コマンド実行軸 (SELx) で指定した全ての軸の入出力バッファを読出</p> <p>■レジスタ書込関数 (cpxxx_wBufDW ()) の応用</p> <p>複数軸へのレジスタデータを同じタイミングで一括書込を行います。(同一PCL内の軸)</p> <p>◇書込みを行う全ての入出力バッファに所定データを書込</p> <p>◇cpxxx_wCmdW () 関数の“コマンド”で複数軸の読出したいプリレジスタ・レジスタを指定。</p> <p>◎軸指定 (axis) はX軸 (0) , 制御コマンドデータ (cmd) 中のコマンド実行軸 (SELx) に 2軸以上設定, コマンドコード (code) に書込コマンドを設定</p>						

(1 1) cpxxx_SetIntCall()		割込処理関数の登録・削除						
機能		デバイス ID で指定された C P D ボードからの割込が発生した場合の処理関数を登録または削除						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	○	×	×	×	×	×
DOS	書式	short cpxxx_SetIntCall(DWORD hDevID, PINTPROC fnIntCall);						
	引数	DWORD hDevID: ・ ・ ボードのデバイスハンドル PINTPROC fnIntCall; ・ ・ 登録 : 割込が発生した時の割込処理モジュールのアドレス 削除 : NULL (=0)						
	呼出例	short ans; PINTPROC int_module; ans = hppdxxx_SetIntCall (hDevID, int_module); /*割込処理モジュール*/						
備考		<p>《ご注意》 ・ ・ ・ 割込を使用する場合には、次の点に留意して下さい。</p> <p>(1) 初期化時の指令関数の順序</p> <p>① オプションポート・ボード割込マスクの設定。(デバッグ時)</p> <p>② 全てのボード初期化を実行する。(ソフトウェアリセット(0x04)コマンドの指令)</p> <p>③ 各軸の動作条件設定では、環境設定 1 : 割込アンマスク : b 2 9=0)とします。</p> <p>④ cpxxx_SetIntCall () 関数で割込処理モジュールをドライバに通知します。</p> <p>(2) 割込処理モジュール (アプリケーションプログラム)</p> <p>① ボード単位で処理を行います。(全搭載軸の一括処理)</p> <p>② モジュール内では”CPU割込禁止”を保持します。(モジュールは割込禁止で起動。)</p> <p>③ モジュール本体およびモジュール内で起動するモジュールでは”スタックチェック”禁止です。(関数用スタックエリア=1KB)</p> <p>(禁止としない時、DOSから”スタックオーバーフロー”が報告される事があります。)</p> <p>④ ドライバ関数実行中に割込処理モジュール内でドライバ関数を起動する時、異常終了「戻り値=0 x 4 0 (ILLEGAL_ACCESS) (P7)」となる場合があります。</p> <p>(I / Oポートのリエントラント処理ができない為)</p> <p>⑤ cpxxx_rMstsW () 関数で全軸の割込要因を検出します。</p> <p>◎ 割込要因が「エラーステータス (SERR:b4=1)」である場合、 cpxxx_rReg () 関数でエラーステータス (REST) を読みます。</p> <p>◎ 割込要因が「イベントステータス (SINT:b5=1)」である場合、 cpxxx_rReg () 関数でイベントステータス (RIST) を読みます。</p> <p>⑥ 作成方法はサンプルプログラムを参照して下さい。</p> <p>(3) 終了時に忘れてはならないこと。</p> <p>① オプションポート・ボード割込マスクの設定。</p> <p>② cpxxx_SetIntCall () 関数で割込処理モジュールをドライバから消去します。</p>						

(1 2) cpxxx_GetDevVerNo()		バージョン番号の取得							
機能		現在パソコンにインストールされているデバイスドライバと、アプリケーションにリンクしたドライバ I/F ライブラリのバージョン番号を取得します。							
関数名 (xxx)		530	52C	230	434	730	130	360	
		○	○	×	×	×	×	×	
DOS	書式	short cpxxx_GetDevVerNo(U_LONG* verno);							
	引数	WORD* verno; ・ ・ バージョン番号が格納されます。							
	呼出例	short ans; WORD* verno; ans = cpxxx_GetDevVerNo(&verno);							
備考		【 バージョン番号の構成 】							
		15	12	11	8	7	4	3	0
		(1 ~)		(0 ~)		(1 ~)		(0 ~)	
		バージョン番号		リリース番号		バージョン番号		リリース番号	
		ドライバ I/F ライブラリ				デバイスドライバ			

(1 3) cpxxx_GetBoardCode ()		ボー ド固有コードの取得						
機能		H P C I ボードに対して、指定したボード固有コードの取得を行います。 これにより、H P C I - C P D 5 7 8 ボードとその他のボードの区分が可能となります。						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	×	×	×	×	×	×
Win V C++	書式	DWORD WINAPI cp530_GetBoardCode (DWORD hDevID, WORD* b_code);						
	引数	DWORD hDevID; ・ ・ 対象デバイスのデバイスハンドル WORD* b_code; ・ ・ 読出データ (ボード固有コード) の格納エリアアドレス						
	呼出例	DWORD ret; //関数の戻り値 WORD* b_code ; //ボード固有コード ret = cp530_GetBoardCode (hDevID, &b_code);						
Win V B	書式	Declare Function cp530_GetBoardCode Lib "libname" _ (ByVal hDevID As Long, ByRef b_data As Integer) As Long						
	引数	ByVal hDevID As Long ・ ・ 対象デバイスのデバイスハンドル ByRef b_code As Integer ・ ・ 読出データ (ボード固有コード) の格納エリアアドレス						
	呼出例	Dim ret As Long '関数の戻り値 Dim b_code As Integer 'ボード固有コード ret = cp530_GetBoardCode (hDevID, b_code)						
関数名 (xxx)		530	52C	230	434	730	130	360
		○	×	×	×	×	×	×
D O S	書式	short cp530_GetBoardCode (DWORD hDevID, short* b_code);						
	引数	DWORD hDevID; ・ ・ ボードのデバイスハンドル short* verno; ・ ・ ボード固有コードが格納されます。						
	呼出例	short ans; short b_code; ans = cp530_GetBoardCode (hDevID, &b_code);						
備考		H P C I ボード種別とボード固有コード H P C I - C P D 5 7 8 ・ ・ ・ 5 7 8 a h H P C I - C P D 5 3 4 , 5 3 2 , 5 0 8 ・ ・ ・ 5 2 5 4 h						

4. アプリケーション作成上の注意

4. 1 マルチスレッド使用時の注意点

Windowsではマルチスレッドがサポートされていますが、マルチスレッドを使用し複数のスレッドからボードにアクセスする場合は同期が必要になる場合があります。

CPDボードに搭載されているパルスコントローラ「PCL6045（以下PCL）」を制御する時には、PCL内部のレジスタのデータを読み出し、または書込みますがこの時に同期が必要になります。以下に解説します。

【 ポートアドレス 】

1軸分のポートアドレスを以下に記します。（16ビットアクセス時）

アドレス	読み込み（INP）		書込み（OUT）	
	呼称	内 容	呼称	内 容
+ 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)

【 レジスタ読み出し、書込み手順 】

レジスタからの読み出し、またはレジスタへの書込み手順は以下のようになります。

■レジスタ読み出し

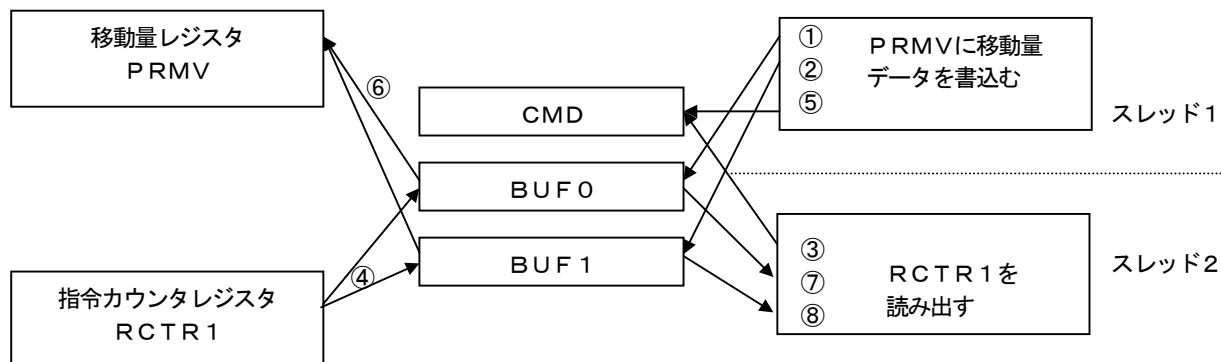
- (1) レジスタ読み出しコマンドをCMDに書込みます。
- (2) レジスタの内容がBUF0、BUF1に読み出されます。
- (3) BUF0、BUF1を読みします。

■レジスタ書込み

- (1) BUF0、BUF1にデータを書込みます。
- (2) レジスタ書込みコマンドをCMDに書込みます。
- (3) レジスタにBUF0、BUF1のデータが書込まれます。

この時、同期を行わないと、どのように不具合が発生するか考えてみます。

例えば下図のような順序の時は、



- (1) スレッド1がBUF0にデータを書く
- (2) スレッド1がBUF1にデータを書く
- (3) スレッド2がCMDにRCTR1読み出しコマンドを書く
- (4) RCTR1の内容がBUF0、BUF1にコピーされる
- (5) スレッド1がCMDにPRMV書込みコマンドを書く
- (6) PRMVにスレッド2で読み出したRCTR1のデータが書かれてしまう

となり、PRMVに正しいデータが書けません。

そのため、スレッド1とスレッド2の間で同期をとる必要があります。

【 スレッド間で同期が必要な関数グループ 】

No	関数名 “xxx”	グループ
1	hcpxxx_GetDevInfo	
2	hcpxxx_DevOpen	○
3	hcpxxx_DevClose	
4	hcpxxx_SetOrgMode	○
5	hcpxxx_SetEls	○
6	hcpxxx_SetOls	○
7	hcpxxx_SetSvAlm	○
8	hcpxxx_SetEz	○
9	hcpxxx_SetDlsSel	○
10	hcpxxx_SetInpos	○
11	hcpxxx_SetSvCtrCl	○
12	hcpxxx_SetSls	○
13	hcpxxx_SetOmdPulse	○
14	hcpxxx_SetAccProfile	○
15	hcpxxx_SetAutoDec	○
16	hcpxxx_mSetDls	○
17	hcpxxx_mSetLtc	○
18	hcpxxx_mSetInpos	○
19	hcpxxx_mSetSvCtrCl	○
20	hcpxxx_ReadMainSts	
21	hcpxxx_ReadErrorSts	○
22	hcpxxx_ReadEventSts	○
23	hcpxxx_ReadSubSts	
24	hcpxxx_ReadExSts	○
25	hcpxxx_ReadIpSts	○
26	hcpxxx_ReadSpd	○
27	hcpxxx_ReadCtr	○
28	hcpxxx_SetFLSpd	○
29	hcpxxx_SetAuxSpd	○
30	hcpxxx_SetAccRate	○
31	hcpxxx_SetDecRate	○
32	hcpxxx_SetMult	○
33	hcpxxx_SetEventMask	○
34	hcpxxx_SetDecPoint	○
35	hcpxxx_WritOpeMode	○
36	hcpxxx_WritFHSpd	○
37	hcpxxx_WritPos	○
38	hcpxxx_WritLine	○
39	hcpxxx_WritCircl	○
40	hcpxxx_WritCtr	○
41	hcpxxx_DecStop	
42	hcpxxx_QuickStop	
43	hcpxxx_EmgStop	
44	hcpxxx_SyDecStop	○
45	hcpxxx_SyQuickStop	○
46	hcpxxx_AccStart	
47	hcpxxx_CnstStartFH	
48	hcpxxx_CnstStartFL	
49	hcpxxx_CnstStartByDec	
50	hcpxxx_SvOn	
51	hcpxxx_SvOff	
52	hcpxxx_SvResetOn	
53	hcpxxx_SvResetOff	
54	hcpxxx_PMON	
55	hcpxxx_PMOFF	
56	hcpxxx_mGetOptInp	○
57	hcpxxx_CalAccRate	

次頁に続く

前頁からの続き

58	cpxxx_GetDeviceCount()	
59	cpxxx_GetDeviceInfo()	
60	cpxxx_OpenDevice()	
61	cpxxx_CloseDevice()	
62	cpxxx_rMstsW()	
63	cpxxx_rSstsW()	
64	cpxxx_wCmdW()	○
65	cpxxx_rReg()	○
	cpxxx_wReg()	○
66	cpxxx_rPortB()	
	cpxxx_wPortB()	
	cpxxx_rPortW()	
	cpxxx_wPortW()	
67	cpxxx_rBufDW()	○
	cpxxx_wBufDW()	○

○のグループの関数を複数のスレッドで呼ぶ場合は同期をとる必要があります。