

PCI Bus CPD ボードシリーズ

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・別冊〉

INtime HLS-CPD500/IT 導入版

NC ボード

多機能・高速 円弧・直線補間・位置決め



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

この説明書は次のボードに適応しています。

| | | |
|--------------------|--------|----------|
| PCI | HPCI- | CPD532 |
| | HPCI- | CPD534 |
| | HPCI- | CPD508 |
| | HPCI- | CPD578 |
| | HPCI- | CPD574N |
| | HPCI- | CPD578N |
| | HPCI- | CPD5212M |
| | HPCI- | CPD5016 |
| | HPCI- | CPD553 |
| Compact PCI | HCPCI- | CPD734 |
| | HCPCI- | CPD738 |
| PCI Express | HPCIe- | CPD674N |
| | HPCIe- | CPD678 N |

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

Windows は Microsoft Corporation の米国及びその他の国における登録商標です。
その他、記載されている会社名、製品名は、各社の商標又は登録商標です。

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

第 1.02 版 2022 年 11 月 07 日発行
不許複製・転載



本製品をご使用される前に「注意事項」を必ずご一読の上ご利用
をお願い致します。

目 次

| | |
|--|----|
| ■ 注意事項 | 0 |
| ■ 保証範囲 | 0 |
| ■ 免責事項 | 0 |
| ■ 安全にお使い頂くために | 0 |
| ■ 対象ユーザー | 1 |
| ■ 添付ソフトウェア適合 OS | 1 |
| ■ サンプルプログラム | 1 |
| ■ ユーザープログラム | 2 |
| ■ 試運転・調整 | 2 |
| ■ CPD シリーズのマニュアル構成 | 1 |
| ■ マニュアル更新履歴 | 1 |
| 1. はじめに | 1 |
| 1.1 ソフトウェアの構成 | 1 |
| 1.2 Setup.exe によりインストールされるファイル | 2 |
| 1.3 関数名 | 2 |
| 1.4 関数の戻り値 | 3 |
| 1.5 アプリケーション作成準備 | 4 |
| 1.5.1 Microsoft Visual C++ (2008 以上) | 4 |
| 1.5.2 ボードを複数枚使用する場合 | 4 |
| 1.5.3 ボードアクセス方法 | 5 |
| 2. ライブラリ関数 | 7 |
| 2.1 ライブラリ関数について | 7 |
| 2.1.1 軸の指定 | 7 |
| 2.1.2 ライブラリ関数 | 8 |
| 2.2 デバイス操作 | 10 |
| 2.2.1 hlt500_GetDevInfo() ボード枚数、デバイス情報の取得 | 10 |
| 2.2.2 hlt500_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 | 11 |
| 2.2.3 hlt500_DevClose() デバイスのクローズ | 12 |
| 2.3 初期設定 | 13 |
| 2.3.1 hlt500_SetOrgMode() 原点復帰モードの設定 | 13 |
| 2.3.2 hlt500_SetEls() ELS の設定 | 13 |
| 2.3.3 hlt500_SetOls() OLS の設定 | 14 |
| 2.3.4 hlt500_SetSvAlm () SVALM の設定 | 14 |
| 2.3.5 hlt500_SetEz() エンコーダ Z 相の設定 | 14 |
| 2.3.6 hlt500_SetDlsSel() DLS/PCS の入力選択、設定 | 15 |
| 2.3.7 hlt500_SetInpos () INPOS の設定 | 15 |
| 2.3.8 hlt500_SetSvCtrCl() 偏差カウンタクリア出力の設定 | 16 |
| 2.3.9 hlt500_SetSlis() ソフトリミットの設定 | 16 |
| 2.3.10 hlt500_SetCmdPulse() 指令パルスの出力形式の設定 | 17 |
| 2.3.11 hlt500_SetAccProfile() 加減速形式の設定 | 17 |
| 2.3.12 hlt500_SetAutoDec() 減速開始点計算方式の自動計算/手動計算切り替え | 17 |
| 2.4 状態読み出し | 18 |
| 2.4.1 hlt500_ReadMainSts() メインステータスの読み出し | 18 |
| 2.4.2 hlt500_ReadErrorSts() エラーステータスの読み出し | 19 |

| | | | |
|--------|-----------------------------|------------------|----|
| 2.4.3 | hlt500_ReadEventSts() | イベントステータスの読出し | 20 |
| 2.4.4 | hlt500_ReadSubSts() | サブステータスの読出し | 21 |
| 2.4.5 | hlt500_ReadExSts() | 拡張ステータスの読出し | 22 |
| 2.4.6 | hlt500_ReadSpd() | 指令速度の読出し | 23 |
| 2.4.7 | hlt500_ReadCtr() | カウンタの読出し | 23 |
| 2.5 | 動作設定 | | 24 |
| 2.5.1 | hlt500_SetFLSpd() | ベース速度の設定 | 24 |
| 2.5.2 | hlt500_SetAuxSpd() | 補助速度の設定 | 24 |
| 2.5.3 | hlt500_SetAccRate() | 加速レートの設定 | 25 |
| 2.5.4 | hlt500_SetDecRate() | 減速レートの設定 | 26 |
| 2.5.5 | hlt500_SetMult() | 速度倍率レジスタ値の設定 | 27 |
| 2.5.6 | hlt500_SetEventMask() | イベントマスクの設定 | 28 |
| 2.5.7 | hlt500_SetDecPoint() | 減速開始点の設定 | 29 |
| 2.6 | 運用設定 | | 30 |
| 2.6.1 | hlt500_WritOpeMode() | 動作モードの設定 | 30 |
| 2.6.2 | hlt500_WritFHSpd() | 動作速度の設定 | 31 |
| 2.6.3 | hlt500_WritPos() | 位置決め移動量の設定 | 31 |
| 2.6.4 | hlt500_WritLine() | 直線補間の移動量の設定 | 31 |
| 2.6.5 | hlt500_WritCircl() | 円弧補間の移動量の設定 | 32 |
| 2.6.6 | hlt500_WritCtr() | カウンタプリセット | 32 |
| 2.7 | 動作制御指令 | | 33 |
| 2.7.1 | hlt500_DecStop() | 減速停止 | 33 |
| 2.7.2 | hlt500_QuickStop() | 即停止 | 33 |
| 2.7.3 | hlt500_EmgStop() | 非常停止 | 33 |
| 2.7.4 | hlt500_AccStart() | 加速スタート | 34 |
| 2.7.5 | hlt500_CnstStartFH() | FH 定速スタート | 34 |
| 2.7.6 | hlt500_CnstStartFL() | FL 定速スタート | 34 |
| 2.7.7 | hlt500_CnstStartByDec() | FH定速スタート後減速停止 | 35 |
| 2.7.8 | hlt500_SvOn() | SVON オン | 35 |
| 2.7.9 | hlt500_SvOff() | SVON オフ | 35 |
| 2.7.10 | hlt500_SvResetOn() | SVRST オン | 36 |
| 2.7.11 | hlt500_SvResetOff() | SVRST オフ | 36 |
| 2.7.12 | hlt500_PMON() | パルスモータ励磁オン | 36 |
| 2.7.13 | hlt500_PMOFF() | パルスモータ励磁オフ | 37 |
| 2.8 | 加減速レートの計算 | | 38 |
| 2.8.1 | hlt500_CalAccRate() | 加減速レートの計算 | 38 |
| 2.9 | 内部情報の取得 | | 39 |
| 2.9.1 | hlt500_GetBoardCode() | ボード固有コードの取得 | 39 |
| 2.9.2 | hlt500_GetDevVerNo() | バージョン情報の取得 | 39 |
| 2.9.3 | hlt500_GetLastDetailError() | ローカルエラーコードの取得 | 39 |
| 3. | ドライバ関数 | | 40 |
| 3.1 | 関数の種類 | | 40 |
| 3.2 | プリレジスタ | | 41 |
| 3.3 | ドライバ関数の戻り値 | | 41 |
| 3.4 | 関数の詳細 | | 42 |
| 3.4.1 | lt500_ApiStartup() | ドライバ API 関数の活性化 | 42 |
| 3.4.2 | lt500_ApiCleanup() | ドライバ API 関数の資源解放 | 42 |
| 3.4.3 | lt500_GetDeviceCount() | ボード枚数の取得 | 42 |
| 3.4.4 | lt500_GetDeviceInfo() | デバイス情報の取得 | 43 |
| 3.4.5 | lt500_OpenDevice() | デバイスのオープン | 43 |
| 3.4.6 | lt500_CloseDevice() | デバイスのクローズ | 44 |
| 3.4.7 | lt500_rMstsW() | メインステータスの読出し | 45 |

| | | | |
|--------|--------------------------|-----------------------------|----|
| 3.4.8 | lt500_rSstsW() | サブステータスの読出し | 46 |
| 3.4.9 | lt500_wCmdW() | 制御コマンド書込み | 47 |
| 3.4.10 | lt500_rReg() | レジスタ読出し | 49 |
| 3.4.11 | lt500_wReg() | レジスタ書込み | 49 |
| 3.4.12 | lt500_rPortB() | オプションポートバイト読出し..... | 69 |
| 3.4.13 | lt500_wPortB() | オプションポートバイト書込み..... | 69 |
| 3.4.14 | lt500_rPortW() | オプションポートワード(2 バイト)読出し..... | 70 |
| 3.4.15 | lt500_wPortW() | オプションポートワード(2 バイト)書込み | 70 |
| 3.4.16 | lt500_rBufDW() | 入出力バッファ読出し | 71 |
| 3.4.17 | lt500_wBufDW() | 入出力バッファ書込み | 72 |
| 3.4.18 | lt500_SetIntCall() | 割込処理関数の登録..... | 73 |
| 3.4.19 | lt500_ResetIntCall() | 割込機能の削除..... | 73 |
| 3.4.20 | lt500_WaitInt() | 割込イベント待ち..... | 73 |
| 3.4.21 | lt500_GetIntData | 割込イベント要因の取得..... | 74 |
| 3.4.22 | lt500_GetDevVerNo | バージョン番号の取得 | 76 |
| 3.4.23 | lt500_GetBoardCode() | ボード固有コードの取得 | 76 |
| 3.4.24 | lt500_GetLastDetailError | 詳細エラーコードの取得 | 77 |

図 表 目 次

| | |
|--|----|
| 図 1.1-1 ソフトウェアの構成 | 1 |
| *1 64ビット Windows はフォルダー名が“C:¥Program Files(x86)になります. | 2 |
| 表 1.2-1 Setup.exe によりコピーされるファイル | 2 |
| 表 1.3-1 関数名 | 2 |
| 表 1.4-1 関数の戻り値 | 3 |
| 表 1.5-1 プロジェクトへ追加するファイル | 4 |
| 表 1.5-2 その他ビルドで使用するファイル | 4 |
| 図 1.5-1 ボードを複数枚使用 | 4 |
| 表 2.1-1 ライブラリ関数一覧表 | 9 |
| 表 3.1-1 ドライバ関数一覧 | 40 |
| 表 3.1-2 プリレジスタ | 41 |
| 表 3.4-1 割込処理開始手順 | 75 |
| 表 3.4-2 割込スレッド内処理手順 | 75 |
| 表 3.4-3 割込処理停止手順 | 75 |

■ 注意事項

■ 保証範囲

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



■ 免責事項

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

■ 安全にお使い頂くために

この度は、弊社 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、本製品をご使用して頂く場合の取扱い、留意点に付いて記入してありますので、必ずご一読の上ご利用をお願い致します。

尚、本書は、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、必ずこのユーザーズマニュアル及び付属書類を全て熟読し、内容を理解してから正しくご使用下さい。本製品の知識、安全の情報及び注意事項の全てに付いて習熟してからご使用下さい。 本ユーザーズマニュアルでは、安全注意事項のランクを「警告」、「注意」として区分してあります。 | |
|  警告 | この表示を無視して、誤った取扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示しています。 |
|  注意 | この表示を無視して、誤った取扱いをすると、人が傷害を負う可能性または物的損害が想定される内容を示しています。 |

■ 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。

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

■ 添付ソフトウェア適合 OS

注意



添付ソフトウェアは、I/Time および Windows8(32bit/64bit)、Windows7(32bit/64bit)の各エディション、Windows XP SP3(32bit)の何れかの Windows OS が動いているパソコン上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 および dRTOS 上でのご使用については非対応です。詳しくは弊社営業までお問合せ下さい。

■ サンプルプログラム

警告



本製品に添付される「サンプルプログラム」は、ボードが正しく設定・装着されているか、動作環境が正しく設定されているかを理解して頂くとともに、ボードの機能・動作を理解およびボードを制御する手順・制御プログラムの作成方法を理解して頂く為のものです。
故に使用される機器毎に固有な安全対策処理等を含んでいませんので、「動かしてみる」プログラムを定期的に機器運転に使用しないで下さい。



モータや装置を接続して動作させる場合は、モータや装置の特性を考慮した動作条件を設定願います。
特に試運転時は、十分に安全な値で実施し、徐々に所定の値に変更することをお勧めします。



サンプルプログラムを使用し装置を動作させる時、最初は速度の低いところで、また機械系に合った設定を行って動作を確認して下さい。機械系に合わない設定で動作を行うと思わぬ動きをすることがあります。

■ ユーザープログラム



本製品を使用し装置を動作させる時は、プログラムのデバッグを充分行ってから動作させて下さい。プログラムに間違いがあると、思わぬ動きをすることがあります。

■ 試運転・調整

警告



本シリーズ製品を使用し装置を動作させる時は、プログラムのデバッグを充分行ってから動作させてください。プログラムに間違いがあると、思わぬ動きをすることがあります。



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

■ CPD シリーズのマニュアル構成

CPD シリーズ製品のマニュアルは

- (1) CPD シリーズユーザーズマニュアル <導入編>
- (2) CPD シリーズユーザーズマニュアル <運用編>
- (3) 各製品ユーザーズマニュアル <ハードウェア編>
- (4) 各製品ユーザーズマニュアル <ソフトウェア編> (標準添付は Windows 版, DOS 版は別途)

の 4 部構成です。

各マニュアルの内容は以下の通りです。

CPD シリーズユーザーズマニュアル <導入編>

ー 全ての開発者向け

- CPD シリーズ概要
- インストール
- 動かしてみる
- 試運転
- 用語解説

別冊において以下 INtime HLS-CPD500 導入版

の差異部分を説明

- インストール
- サンプルプログラム(動かしてみるの代替)

CPD シリーズユーザーズマニュアル <運用編>

ー 主としてソフトウェア開発者向け

- 基本的な運用
- 特殊な運用

各製品ユーザーズマニュアル <ハードウェア編>

ー 主として配線担当者向け

- 製品仕様, 購入時オプション
- ブロック図
- 接続構成
- ボード上の設定
- 外部との接続
- アクセサリ(中継コネクタボード, 接続ケーブルなど)
- 各社サーボアンプとの接続例

各製品ユーザーズマニュアル <ソフトウェア編>

ー 主としてソフトウェア開発者向け

- ソフトウェア概要
- ライブラリ関数
- ドライバ関数
- サンプルプログラム
- ポート資料

別冊において以下 INtime HLS-CPD500 導入版

の差異部分を説明

- ソフトウェア概要
- ライブラリ関数
- ドライバ関数
- サンプルプログラム

■ マニュアル更新履歴

| 版数 | 日付 | 更新内容 | 備考 |
|--------|------------|--|----|
| 1.00 版 | 2014/06/23 | HLS-Cpd530/IT ソフトウェア編・別冊 初版 | |
| 1.01 版 | 2014/07/14 | PCI-Express MSI 対応の記述を追加 | |
| 1.02 版 | 2015/11/05 | 製品名称を HLS-CPD530/IT から HLS-CPD500/IT に変更 関数戻り値誤記修正, その他誤記訂正, 変更 | |
| 1.10 版 | 2022/11/7 | デバイスドライバ Ver1.31 以前と Ver1.40 以降のバージョン 統合によるインストール手順変更. | |

1. はじめに

本マニュアルは CPD シリーズボード(以下, CPD ボード)で使用される INtime 版 HLS-CPD500/IT パッケージソフトウェアの API 関数の説明書です. ソフトウェアの基本的な運用方法等につきましては, CPD シリーズユーザーズマニュアル<運用編 >なども併せてご覧ください.

1.1 ソフトウェアの構成

CPD ボードを INtime 上で制御するためのアプリケーション開発, 動作に必要な最小ソフトウェア構成は以下となります.

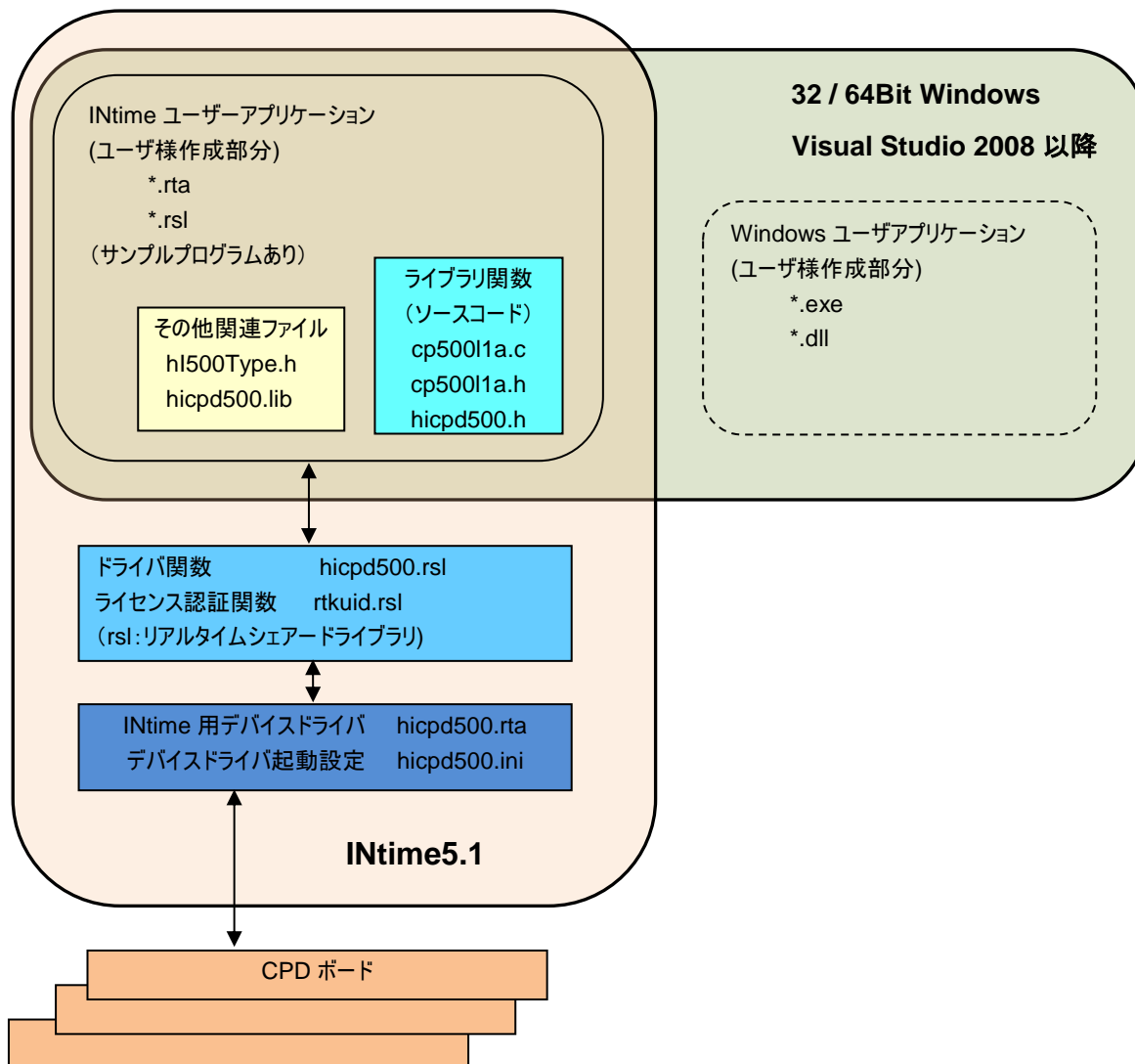


図 1.1-1 ソフトウェアの構成

1.2 Setup.exe によりインストールされるファイル

ソフトウェアパッケージ CD 内の Setup フォルダにある Setup.exe の実行により以下のファイルが必要な場所にコピーされます。(事前に INtime のインストールが必要です)

| No. | ファイル名 | コピー先パス | 内容 |
|-----|--------------|---|---|
| 1 | hicpd500.rta | C:\Program Files\INtime\Drivers\Cpd500 (*1) | PCI/PCI-e/CPCI BUS CPD530 シリーズ用 INtime デバイスドライバ |
| 2 | hicpd500.ini | C:\Program Files\INtime\Drivers\Cpd500 | デバイスドライバ起動設定 ini ファイル |
| 3 | hicpd500.rsl | C:\Program Files\INtime\bin | ドライバ関数 RSL ファイル |
| 4 | hicpd500.lib | C:\Program Files\INtime\rt\lib | ライブラリーリンク用ファイル |
| 5 | hicpd500.h | C:\Program Files\INtime\rt\include | ライブラリー関数外部宣言 |
| 6 | ht500Type.h | C:\Program Files\INtime\rt\include | システム定数, 構造体定義 |
| 7 | rtkuid.rsl | C:\Program Files\INtime\bin | ライセンス認証用 INtime RSL ファイル |

*1 64 ビット Windows はフォルダ名が“C:\Program Files(x86)”になります。

表 1.2-1 Setup.exe によりコピーされるファイル

サンプルプログラムとドキュメント関連ファイルにつきましては Setup.exe ではインストールされません。ソフトウェアパッケージ CD

1.3 関数名

各 API 関数は以下の命名規則によって関数名がつけられています。

| No. | 種別 | 関数名 | 提供形態 |
|-----|----------|------------|-------------------------|
| 1 | ドライバ関数 | lt500_xxxx | リアルタイムシェアードライブラリ(*.rsl) |
| 2 | ライブラリー関数 | ht500_xxxx | ソースファイル(*.c) |

表 1.3-1 関数名

1.4 関数の戻り値

以下はライブラリ関数およびドライバ関数が返す戻り値です。この戻り値は複数エラーが同時発生した場合は複数ビットがセットされて返されます。

関数の戻り値が異常値('0'以外)であった場合は、異常内容に対応した処理を行います。この異常が発生した場合は通常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見直しが必要となります。

発生したエラー内容の詳細を知りたい場合はライブラリ関数“hit500_GetLastDetailError”を実行して詳細なエラーコードを取得します。

| No | 記号表記 | 戻り値 | 異常内容と確認項目 |
|----|---------------------|--------|--|
| | | 16進数表記 | |
| 1 | NO_ERROR | 0000h | 正常 異常は発生していません |
| 2 | NOT_FOUND | 0001h | 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 | 0002h | 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枚以上を使用する場合、オープンするデバイス情報の更新を確認します。 |
| 4 | INSUFFICIENT_MEMORY | 0004h | 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 | 0008h | 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 | 0010h | 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 | 0020h | 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_PARAM | 0100h | 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 |
| 9 | OTHER_BOARD | 0200H | ボードの種類が判定できない ◎ライブラリが対応していないボードが存在します。ライブラリを最新に変えて確認して下さい。 |

表 1.4-1 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(サーボアラームやエンドリミットによる停止など)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。

1.5 アプリケーション作成準備

1.5.1 Microsoft Visual C++ (2008 以上)

INtime アプリケーションでプロジェクト作成後、次のファイルをプロジェクトへ追加します。

| No. | ファイル名 | 内容 |
|-----|------------|----------------------|
| 1 | cp500l1a.c | ライブラリ関数 C ソースコードファイル |
| 2 | cp500l1a.h | ライブラリ関数定義ヘッダファイル |

表 1.5-1 プロジェクトへ追加するファイル

上記ファイル以外に、以下のファイルがコンパイル、リンク時必要ですが、HLS-CPD500/IT のインストール時にこれらファイルは INtime フォルダへコピーされます。したがってこれらファイルをユーザーにてプロジェクト内にコピーする必要はありません。

| No. | ファイル名 | 内容 |
|-----|--------------|----------------------------------|
| 1 | hl500Type.h | デバイスドライバ定義ヘッダファイル |
| 2 | hicpd500.lib | ドライバ関数リンク用 LIB ファイル |
| 3 | hicpd500.h | ドライバ関数定義ヘッダファイル |
| 4 | hicpd500.rsl | ドライバ関数 RSL ファイル |
| 5 | rtkuid.rsl | ドライバ関数内で内部呼び出しを行っている関数の RSL ファイル |

表 1.5-2 その他ビルドで使用するファイル

ユーザ作成のソースコード中でドライバ関数またはライブラリ関数を使用する場合は以下ファイルを#include 宣言にてヘッダファイルをインクルードして下さい。

例. #include "hicpd500.h" ... ドライバ関数使用の場合
 #include "cp500l1a.h" ... ライブラリ関数使用の場合

1.5.2 ボードを複数枚使用する場合

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

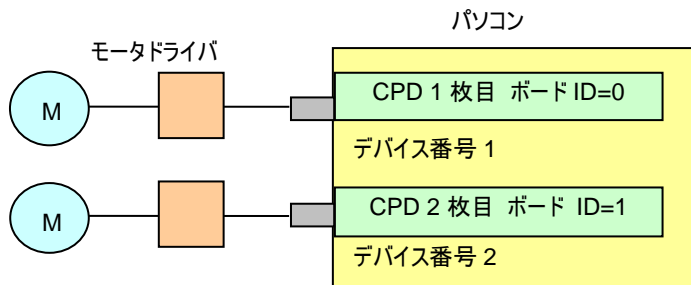


図 1.5-1 ボードを複数枚使用

(1) ボードの-slot番号とボード ID

PCI ではシステムがボードのアドレス管理をしています。

ボードが装着される slot にはシステム側で決めたデバイス番号が割振られます。

しかし、このデバイス番号はシステムによって割振られるため、ボードと slot の関係が外部から直接認識出来ません。

このために、CPD ボードには「ボード ID」が設けられています。これにより、ボードとソフトを対応させることが出来ます。

(2) ボード ID の使用

ボード ID は No.0~15 を設定出来ます。CPD530 シリーズのボード同士で重複する ID は使用できません。

これにより合計 16 枚までの CPD ボードを 1 台の PC で扱うことが可能です。

1.5.3 ボードアクセス方法

添付される API 関数は、複数の CPD ボードを制御することができます。あるひとつの CPD ボードにアクセスするためには、まずこのデバイスをオープンして、アクセスするために必要なデバイスハンドル値を取得する必要があります。

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

この情報をデバイス情報と呼びます。

(I/O ポートアドレスや IRQ 番号等のハードウェアリソースは、システム側によって確定されます。)

(1) デバイス情報構造体

| Ver1.40 以降で使用しているメンバー名 | Ver1.31 以前で使用していたメンバー名 | |
|---|------------------------|-------------------------|
| typedef struct { | | |
| WORD wloMapTypePcl; | WORD dwloMapTypePcl; | /* マッピングタイプ:PCL */ |
| WORD wloMapTypeOpt; | WORD dwloMapTypeOpt; | /* マッピングタイプ:オプションポート */ |
| BYTE* pbyloAdrPcl; | BYTE* dwloAdrPcl; | /* I/O アドレス:PCL */ |
| BYTE* pbyloAdrOpt; | BYTE* dwloAdrOpt; | /* I/O アドレス:オプションポート */ |
| WORD wBoardType; | WORD dwBoardType; | /* ボードタイプ */ |
| WORD nBusNumber; | WORD nBusNumber; | /* バス番号 */ |
| WORD nDevNumber; | WORD nDevNumber; | /* デバイス番号 */ |
| WORD wIrqNo; | WORD dwIrqNo; | /* IRQ 番号 */ |
| WORD wNumber; | WORD dwNumber; | /* 管理番号 */ |
| WORD wBoardID; | WORD dwBoardID; | /* ボード ID */ |
| WORD MsiOffset; | WORD wMsiOffset; | /* MSI オフセット */ |
| } HPC_DEVINFO, *PHPC_DEVINFO, *LPHPC_DEVINFO; | | |

上表に記載した構造体メンバー名は、従来のサンプルプログラム互換のためで使用しています。最新版サンプルプログラムでは、これら以外に下記の構造体が定義され使用できます(メンバー名をデータ型と一致する名称に修正したものです。構造体サイズは同じです)

```
typedef struct {
    WORD wloMapTypePcl; /* マッピングタイプ:PCL */
    WORD wloMapTypeOpt; /* マッピングタイプ:オプションポート */
    BYTE* pbyloAdrPcl; /* I/O アドレス:PCL */
    BYTE* pbyloAdrOpt; /* I/O アドレス:オプションポート */
    WORD wBoardType; /* ボードタイプ */
    WORD wBusNumber; /* バス番号 */
    WORD wDevNumber; /* デバイス番号 */
    WORD wIrqNo; /* IRQ 番号 */
    WORD wNumber; /* 管理番号 */
    WORD wBoardID; /* ボード ID */
    WORD wMsiOffset; /* MSI オフセット */
} NHPC_DEVINFO, *NPHPC_DEVINFO, *NLPHPC_DEVINFO;
```

これらの構造体はファイル“¥INtime¥rt¥include¥hl500Type.h”で定義しています。コメント部分の定義が必要な場合はコメントを外し、重複する他方の定義をコメントにするなど直接編集してご使用ください。

(2) 関数のリエントラント性および排他について

INtime はマルチスレッドで動作する事が基本となるので、HLS-CPD530/IT に添付されるドライバ関数、およびそれを使用しているライブラリ関数はリエントラント性を持ちます。ドライバ関数はセマフォによりボード毎の排他を行っています。また複数のスレッドにより異なるボードで同時にドライバ関数を使用した場合、デバイスドライバに先取りされたドライバ関数の処理がまず実行されます。それ以外の関数は待ち行列にキューイングされます。デバイスドライバは、実行中の関数処理が終了するとすぐにキューイングされた関数を実行、待ち行列内のドライバ関数を順番に処理します。これら処理によってデバイスに対する排他を行っています。

2. ライブラリ関数

2.1 ライブラリ関数について

ライブラリ関数は、Visual Studio2008 Vc/Vc++で作成されていて、ソースコード(cp500l1a.c ファイルおよび cp500l1a.h ファイル)の形で提供されます。ご使用時にはユーザー様が作成するアプリケーションに組み込んでビルドします。

このライブラリ関数は、次章の RSL(リアルタイムシェアードライブラリ)で提供されるドライバー関数を使用して作られている CPD ボードおよびモーション制御用の応用関数です。

2.1.1 軸の指定

初期設定、状態読み出し、動作設定、運用設定(円弧補間の軸設定を除く)の各関数の軸指定は

0 : X 軸(X1 軸), 1 : Y 軸(Y1 軸), 2 : Z 軸(Z1 軸), 3 : U 軸(U1 軸),
4 : V 軸(X2 軸), 5 : W 軸(Y2 軸), 6 : A 軸(Z2 軸), 7 : B 軸(U2 軸),
8 : X3 軸, 9 : Y3 軸, 10 : Z3 軸, 11 : U3 軸,
12 : X4 軸, 13 : Y4 軸, 14 : Z4 軸, 15 : U4 軸

となり、1回の関数呼び出しで1軸のみ操作・指定できます。

停止や起動など動作制御の軸指定は

01h: X1 軸, 02h: Y1 軸, 04h: Z1 軸, 08h: U1 軸,
10h: X2 軸, 20h: Y2 軸, 40h: Z2 軸, 80h: U2 軸,
100h: X3 軸, 200h: Y3 軸, 400h: Z3 軸, 800h: U3 軸,
1000h: X4 軸, 2000h: Y4 軸, 4000h: Z4 軸, 8000h: U4 軸

となり、1回の関数呼び出しで複数軸同時の操作および指定ができます。(複数軸は各軸指定値の OR)

2.1.2 ライブラリ関数

ライブラリ関数は、ソースコードで提供されます。

(1) 関数一覧

| No | 関 数 名 | 機 能 | 記載項 | |
|----|------------------------------|----------------------|---------------------|--------|
| 1 | デバイス操作 | hIt500_GetDevInfo | ボード枚数, デバイス情報の取得 | 2.2.1 |
| 2 | | hIt500_DevOpen | デバイスのオープン, レジスタの初期化 | 2.2.2 |
| 3 | | hIt500_DevClose | デバイスのクローズ | 2.2.3 |
| 4 | 初期設定 | hIt500_SetOrgMode | 原点復帰モードの設定 | 2.3.1 |
| 5 | | hIt500_SetEls | ELSの設定 | 2.3.2 |
| 6 | | hIt500_SetOls | OLSの設定 | 2.3.3 |
| 7 | | hIt500_SetSvAlm | SVALMの設定 | 2.3.4 |
| 8 | | hIt500_SetEz | エンコーダZ相の設定 | 2.3.5 |
| 9 | | hIt500_SetDls | DLS/PCS入力選択, 設定 | 2.3.6 |
| 10 | | hIt500_SetInpos | INPOSの設定 | 2.3.7 |
| 11 | | hIt500_SetSvCtrCl | 偏差カウンタクリア出力の設定 | 2.3.8 |
| 12 | | hIt500_SetSls | ソフトリミットの設定 | 2.3.9 |
| 13 | | hIt500_SetCmdPulse | 指令パルスの出力形式選択 | 2.3.10 |
| 14 | | hIt500_SetAccProfile | 加減速形式の設定 | 2.3.11 |
| 15 | hIt500_SetAutoDec | 減速点の設定方式の自動/手動切り替 | 2.3.12 | |
| 16 | 状態読み出し | hIt500_ReadMainSts | メインステータスの読出し | 2.4.1 |
| 17 | | hIt500_ReadErrorSts | エラーステータスの読出し | 2.4.2 |
| 18 | | hIt500_ReadEventSts | イベントステータスの読出し | 2.4.3 |
| 19 | | hIt500_ReadSubSts | サブステータスの読出し | 2.4.4 |
| 20 | | hIt500_ReadExSts | 拡張ステータスの読出し | 2.4.5 |
| 21 | | hIt500_ReadSpd | 指令速度の読出し | 2.4.6 |
| 22 | | hIt500_ReadCtr | カウンタの読出し | 2.4.7 |
| 23 | 動作設定 | hIt500_SetFLSpd | ベース速度の設定 | 2.5.1 |
| 24 | | hIt500_SetAuxSpd | 補助速度の設定 | 2.5.2 |
| 25 | | hIt500_SetAccRate | 加速レートの設定 | 2.5.3 |
| 26 | | hIt500_SetDecRate | 減速レートの設定 | 2.5.4 |
| 27 | | hIt500_SetMult | 倍率倍率レジスタ値の設定 | 2.5.5 |
| 28 | | hIt500_SetEventMask | イベントマスクの設定 | 2.5.6 |
| 29 | | hIt500_SetDecPoint | 減速開始点の設定 | 2.5.7 |
| 30 | 運用設定 (Ver1.40以降 で使用可能) | hIt500_WritOpeMode | 動作モードの設定 | 2.6.1 |
| 31 | | hIt500_WritFHSpd | 動作速度の設定 | 2.6.2 |
| 32 | | hIt500_WritPos | 移動量の設定 | 2.6.3 |
| 33 | | hIt500_WritLine | 直線補間の移動量の設定 | 2.6.4 |
| 34 | | hIt500_WritCircl | 円弧補間の移動量の設定 | 2.6.5 |
| 35 | | hIt500_WritCtr | カウンタプリセット | 2.6.6 |
| | A | | | |

| | | | | |
|----|--------|---------------------------|---------------|--------|
| 36 | 動作制御 | hIt500_DecStop | 減速停止 | 2.7.1 |
| 37 | | hIt500_QuickStop | 即停止 | 2.7.2 |
| 38 | | hIt500_EmgStop | 非常停止 | 2.7.3 |
| 39 | | hIt500_AccStart | 加速スタート | 2.7.4 |
| 40 | | hIt500_CnstStartFH | FH定速スタート | 2.7.5 |
| 41 | | hIt500_CnstStartFL | FL定速スタート | 2.7.6 |
| 42 | | hIt500_CnstStartByDec | FH定速スタート後減速停止 | 2.7.7 |
| 43 | | hIt500_SvOn | サーボオン | 2.7.8 |
| 44 | | hIt500_SvOff | サーボオフ | 2.7.9 |
| 45 | | hIt500_SvResetOn | サーボリセットオン | 2.7.10 |
| 46 | | hIt500_SvResetOff | サーボリセットオフ | 2.7.11 |
| 47 | | hIt500_PMOn | パルスモータ励磁オン | 2.7.12 |
| 48 | | hIt500_PMOff | パルスモータ励磁オフ | 2.7.13 |
| 49 | 演算処理 | hIt500_CalAccRate | 加減速レートの計算 | 2.8.1 |
| 50 | 内部情報取得 | hIt500_GetBoardCode | ボード固有コードの取得 | 2.9.1 |
| 51 | | hIt500_GetDevVerNo | バージョン情報の取得 | 2.9.2 |
| 52 | | hIt500_GetLastDetailError | ローカルエラーコードの取得 | 2.9.3 |

表 2.1-1 ライブラリ関数一覧表

(2) 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。詳細は「1.4 関数の戻り値」をご覧ください。

2.2 デバイス操作

2.2.1 hlt500_GetDevInfo() ボード枚数, デバイス情報の取得

| | |
|-------------|---|
| No. | 1 |
| 機能 | パソコンに装着され, INtime の管理下にある CPD シリーズボードの枚数, 及びデバイス情報を取得します. |
| 開発言語 | 書 式 |
| VC++ | short hlt500_GetDevInfo(WORD* DevCnt, HPC_DEVINFO* DevInfo); |
| 引 数 | 説 明 |
| DevCnt, | CPD シリーズボードの枚数 |
| DevInfo | CPD ボードのデバイス情報格納の構造体 |
| VC++ 記述例 | WORD count; short ret; HPC_DEVINFO HpcDevInfo[2]; //2 枚のボードのデバイス情報を格納するエリアの先頭アドレス ret = hlt500_GetDevInfo(&count, &HpcDevInfo[0]); |

2.2.2 hlt500_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

| | |
|-----|--|
| No. | 2 |
| 機能 | 指定したデバイス情報を持つ CPD をオープンし、他の CPD 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、指定した CPD にアクセスするために使用します。またオープンした CPD のレジスタとオプションポートの初期化をします。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_DevOpen(DWORD* hDeviceID, HPC_DEVINFO* DevInfo); |

| | |
|-----------|----------------------|
| 引数 | 説明 |
| hDeviceID | デバイスハンドルの格納先 |
| DevInfo | CPD ボードのデバイス情報格納の構造体 |

| | |
|-------------|--|
| VC++ 記述例 | <pre>// パソコンに CPD 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デバイス情報構造体の配列 DevInfo[2]を準備し、この中には // 既に hlt500_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。 short ret; //関数の戻り値 DWORD hDeviceID[2]; //デバイスハンドル取得エリア ret = hlt500_DevOpen(hDeviceID[0], &DevInfo[0]); //1 番目のデバイス情報 ret = hlt500_DevOpen(hDeviceID[1], &DevInfo[1]); //2 番目のデバイス情報</pre> |
|-------------|--|

| | | | | |
|--------------|-----------|-----------|-----------|---|
| 備考 | レジスタ | 内容 | 初期値 | 補足 |
| | PRFL,RFL | ベース速度 | 200 | 200pps |
| | PRFH,RFH | 動作速度 | 2000 | 2000pps |
| | PRUR,RUR | 加速レート | 1364 | 直線加(減)速時 200→2000pps(2000pps→200pps) 加速(減速)時間:約 0.5 秒 |
| | PRMG,RMG | 速度倍率 | 299 | 1 倍 |
| | RFA | 補助速度 | 200 | 200pps |
| | PRMD,RMD | 動作モード | 08008000h | |
| | RENV1 | 環境設定 1 | 20434004h | |
| | RENV2 | 環境設定 2 | 0020f555h | |
| | RENV3 | 環境設定 3 | 00f00002h | 原点復帰モード 2(OLS+Z 相)等 |
| | RIRQ | イベントマスク設定 | 1 | 正常停止時 |
| | その他 | | 0 | |
| | (次ページに続く) | | | |
| | (前ページより) | | | |
| | オプションポート | | 初期値 | 補足 |
| ELS 入力極性 | | 0 | 全軸 B 接 | |
| DLS/PCS 切り替え | | ffh | 全軸 PCS | |
| その他 | | 0 | | |

2.2.3 hlt500_DevClose() デバイスのクローズ

| | |
|--------|--|
| No. | 3 |
| 機能 | デバイスハンドルで指定された CPD 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。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_DevClose(DWORD hDevID); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| 備考 | デバイスクローズの前に CPD の終了処理を行ってください。 |

2.3 初期設定

2.3.1 hlt500_SetOrgMode() 原点復帰モードの設定

| | |
|-----|---------------------------------------|
| No. | 4 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の原点復帰モードを設定します。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_SetOrgMode (DWORD hDevID, WORD axis, short mode); |

| | |
|--------|--|
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| mode | <p>原点復帰モード</p> <p>0: ORGmode0 OLSoff→on で即停止(加減速動作時は減速停止)</p> <p>1: ORGmode1 OLSoff→on で即停止(加減速動作時は減速停止)後、補助速度定速で逆方向へ OLSoff まで動作し、その後補助速度で初めの方向へ動作し OLSoff→on で即停止</p> <p>2: ORGmode2 定速時は OLSoff→on 後の Z 相カウントアップで即停止 加減速動作時は OLSoff→on で減速、Z 相カウントアップで即停止</p> <p>3: ORGmode3 定速時は OLSoff→on 後の Z 相カウントアップで即停止 加減速動作時は OLSoff→on で減速、Z 相カウントアップで減速停止</p> <p>4: ORGmode4 OLSoff→on で即停止(加減速動作時は減速停止)後に補助速度定速で逆転、OLSon→off 後の Z 相カウントアップ時に即停止</p> <p>5: ORGmode5 OLSoff→on で即停止(加減速動作時は減速停止)後に逆転、OLSon→off 後の Z 相カウントアップ時に即停止(加減速動作時は減速停止)</p> <p>6: ORGmode6 ELSon で停止後、補助速度定速で逆転、ELSoFF で即停止</p> <p>7: ORGmode7 ELSon で停止後、補助速度定速で逆転、ELSoFF 後の Z 相カウントアップ時に即停止</p> <p>8: ORGmode8 ELSon で停止後に逆転、ELSoFF 後の Z 相カウントアップで即停止 (加減速動作時は減速停止)</p> <p>9: ORGmode9 ORGmode0 の動作後、機械位置(CTR2)0 点復帰</p> <p>10: ORGmode10 ORGmode3 の動作後、機械位置(CTR2)0 点復帰</p> <p>11: ORGmode11 ORGmode5 の動作後、機械位置(CTR2)0 点復帰</p> <p>12: ORGmode12 ORGmode8 の動作後、機械位置(CTR2)0 点復帰</p> |

| | |
|-------------|--|
| VC++ 記述例 | <pre>short 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 = hlt500_SetOrgMode(hDevID, 1, 1);</pre> |
|-------------|--|

2.3.2 hlt500_SetEls() ELS の設定

| | |
|-----|---|
| No. | 5 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の ELS 入力極性と入力時停止方法を設定します。 |

| | |
|------|---|
| 開発言語 | 書式 |
| VC++ | short hlt500_SetEls (DWORD hDevID, WORD axis, WORD pol, WORD stop); |

| | |
|--------|---------------------|
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| pol | 入力極性[0:B 接, 1:A 接] |
| stop | 停止方法[0:即停止, 1:減速停止] |

| | |
|-------------|---|
| VC++ 記述例 | <pre>short ret; //関数の戻り値 ret = hlt500_SetEls(hDevID, 1, 1, 0); //Y 軸を指定, A 接, 即停止</pre> |
|-------------|---|

2.3.3 hlt500_SetOls() OLS の設定

| | |
|-------------|--|
| No. | 6 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の OLS 入力極性を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetOls (DWORD hDevID, WORD axis, WORD pol); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| pol | 入力極性[0:B 接, 1:A 接] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetOls(hDevID, 1, 1); // Y 軸を指定, A 接 |

2.3.4 hlt500_SetSvAlm () SVALM の設定

| | |
|-------------|--|
| No. | 7 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の SVALM 入力極性と入力時停止方法を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetSvAlm (DWORD hDevID, WORD axis, WORD pol, WORD stop); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| pol | 入力極性[0:B 接, 1:A 接] |
| stop | 停止方法[0:即停止, 1:減速停止] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetSvAlm(hDevID, 1, 1, 0); //Y 軸を指定, A 接, 即停止 |
| 備考 | HPCI-CPD5212M/HPCI-CPD508/HPCI-CPD5016 の 3 種類のボードに関して HPCI-CPD5212M は 4 軸分 (X,Y,Z,U) の SVALM 信号の入力端子を持っていて、これらの軸に対して本関数を使用できますが、その他の軸については使用できません。 また HPCI-CPD508 は 2 軸分 (X,Y), HPCI-CPD5016 は 4 軸分 (X,Y,Z,U) を SVALM 信号の入力端子として使用可能で、SVALM 信号として設定した軸については本関数を使用する事が出来ます。 |

2.3.5 hlt500_SetEz() エンコーダ Z 相の設定

| | |
|-------------|---|
| No. | 8 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のエンコーダ Z 相の入力処理を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetEz (DWORD hDevID, WORD axis, WORD count, WORD pol); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| count | 原点復帰時の Z 相カウント回数 [0:1 回目,~,15:16 回目] |
| pol | 入力極性[0:A 接, 1:B 接] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetEz(hDevID, 1, 2, 1); //Y 軸を指定, 3 回目, A 接 |

2.3.6 hlt500_SetDisSel() DLS/PCS の入力選択, 設定

| | |
|-------------|--|
| No. | 9 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の 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。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetDisSel (DWORD hDevID, WORD axis, WORD enable, WORD pol, WORD motion, WORD latch); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| enable | DLS/PCS 入力の切り替え[0:DLS 入力,1:PCS 入力,2:両方共不使用] |
| pol | 入力極性[0:B 接, 1:A 接] |
| motion | DLS 入力時の動作[0:減速のみ,1:減速停止] |
| Latch | DLS 入力のラッチ[0:ラッチしない,1:ラッチする] |
| VC++ 記述例 | short ret; //関数の戻り値 //Y 軸,DLS 使用,A 接,減速のみ,ラッチしない ret = hlt500_SetDisSel(hDevID,1,0,1,0,0); |
| 備考 | HPCI-CPD508/HPCI-CPD5212M/HPCI-CPD5016 の 3 種類のボードに関して これらボードは, DLS 信号の入力端子を持たないため, 本関数の使用はできません。 入カソース カウンタ 1・・・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) カウンタ 4・・・汎用カウンタ(指令パルス, エンコーダ入力等) |

2.3.7 hlt500_SetInpos () INPOS の設定

| | |
|-------------|---|
| No. | 10 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の INPOS の入力信号処理方法を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetInpos (DWORD hDevID, WORD axis, WORD enable, WORD pol); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| enable | INPOS 制御[0:OFF,1:ON] |
| pol | 入力極性[0:B 接, 1:A 接] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetInpos(hDevID, 1, 1, 1); //Y 軸を指定, INPOS 制御 ON, A 接 |
| 備考 | HPCI-CPD508/HPCI-CPD5016 の 2 種類のボードに関して HPCI-CPD508 は 2 軸分(X,Y), HPCI-CPD5016 は 4 軸分(X,Y,Z,U)を INPOS 信号の入力端子として使用可能で, INPOS 信号として設定した軸については本関数を使用する事が出来ますが, その他の軸については本関数を使用できません。 |

2.3.8 hlt500_SetSvCtrCI() 偏差カウンタクリア出力の設定

| | |
|-------------|--|
| No. | 11 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の偏差カウンタクリア自動出力の設定をします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetSvCtrCI (DWORD hDevID, WORD axis, WORD enable); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| 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] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetSvCtrCI(hDevID, 1, 1); // Y 軸を指定, 原点完了時出力 |

2.3.9 hlt500_SetSls() ソフトリミットの設定

| | |
|-------------|--|
| No. | 12 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のソフトリミットの設定をします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetSls(DWORD hDevID, WORD axis, long psls, long msls, WORD enable, WORD stop); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| psls | +SLS[パルス数] |
| msls | -SLS[パルス数] |
| enable | 使用/不使用 [0:不使用,1:使用] |
| stop | 停止方法 [0:即停止,1:減速停止] |
| VC++ 記述例 | short 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 = hlt500_SetSls (hDevID, 1, 100000, -50000, 1, 0); |
| 備考 | 1.ソフトリミット使用時は+SLS は必ず -SLS より大きくして下さい。 2.ソフトリミット不使用時は msls = psls = enable = 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 になる方向へはスタートはできません (動きません)。逆方向へはスタートできます。 |

2.3.10 hlt500_SetCmdPulse() 指令パルスの出力形式の設定

| | |
|-------------|---|
| No. | 13 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の指令パルスの出力形式を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetCmdPulse (DWORD hDevID, WORD axis, WORD cmdpls); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| 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] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetCmdPulse(hDevID, 0, 1); //X 軸を指定, 共通指令方式 |

2.3.11 hlt500_SetAccProfile() 加減速形式の設定

| | |
|-------------|---|
| No. | 14 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の加減速形式の設定をします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetAccProfile (DWORD hDevID, WORD axis, WORD accpro); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| accpro | 加減速形式[0:直線,1:S字] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetAccProfile(hDevID, 1, 1); //Y 軸を指定, S字加減速に設定 |

2.3.12 hlt500_SetAutoDec() 減速開始点計算方式の自動計算/手動計算切り替え

| | |
|-------------|---|
| No. | 15 |
| 機能 | デバイスハンドルで指定された CPD 指定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。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetAutoDec (DWORD hDevID, WORD axis, WORD para); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| 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] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetAutoDec(hDevID, 1, 1); //Y 軸の減速開始点の設定を手動計算設定 |

2.4 状態読み出し

2.4.1 hlt500_ReadMainSts() メインステータスの読み出し

| | |
|-----|--|
| No. | 16 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のメインステータスを读出します。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_ReadMainSts (DWORD hDevID, WORD axis, WORD* msts); |

| | | | |
|--------|--|-------|--------------------------------------|
| 引数 | 説明 | | |
| hDevID | デバイスハンドル | | |
| axis | 軸指定 | | |
| msts | メインステータス | | |
| | ビット | 名称 | 説明 |
| | 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)读出して'0') |
| | 5 | SINT | '1':イベント報告あり(イベントステータス(RIST)读出して'0') |
| | 7,6 | SSCx | 実行中または停止中のシーケンス番号(PRMD.b17,16 設定値) |
| | 12-8 | SCMPx | '1':CMP5-1 比較条件成立時 |
| | 13 | SEOR | '1':位置のオーバーライド失敗時(本ステータス读出して'0') |
| | 14 | SPRF | '1':次動作用プリレジスタが満杯('0':書込可能) |
| | 15 | SPDF | '1':CMP5 用プリレジスタが満杯('0':書込可能) |
| | *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"[RIRQ.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. b13(SEOR)は HPCI-CPD574N, HPCI-CPD578N, HPCIe-CPD674N, HPCIe-CPD678N のみ対応しています. | | |

| | |
|-------------|--|
| VC++ 記述例 | <pre>short ret; //関数の戻り値 WORD msts; //メインステータス ret = hlt500_ReadMainSts(hDevID, 1, &msts); //Y 軸を指定</pre> |
|-------------|--|

2.4.2 hlt500_ReadErrorSts() エラーステータスの読出し

| | |
|-----|--|
| No. | 17 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のエラーステータスを読出します。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_ReadErrorSts (DWORD hDevID, WORD axis, DWORD* ests); |

| | | | |
|---|----------|------|--------------------------|
| 引数 | 説明 | | |
| hDevID | デバイスハンドル | | |
| axis | 軸指定 | | |
| ests | エラーステータス | | |
| | ビット | 名称 | 説明 |
| | 0 | ESC1 | 1: CMP1 条件成立で停止 (+SLS) |
| | 1 | ESC2 | 1: CMP2 条件成立で停止 (-SLS) |
| | 2 | ESC3 | 1: CMP3 条件成立で停止 |
| | 3 | ESC4 | 1: CMP4 条件成立で停止 |
| | 4 | ESC5 | 1: CMP5 条件成立で停止 |
| | 5 | ESPL | 1: +ELS 検出で停止 |
| | 6 | ESML | 1: -ELS 検出で停止 |
| | 7 | ESAL | 1: SVALM 検出で停止 |
| | 8 | ESSP | 1: STPon による停止 |
| | 9 | ESEM | 1: EMGon による停止 |
| | 10 | ESSD | 1: DLS 検出による停止 |
| | 12 | ESDT | 1: 動作データが不正で停止(注意 1) |
| | 13 | ESIP | 1: 補間動作中に補間他軸の停止による停止 |
| | 14 | ESPO | 1: パルス入力パルファオーバーフローによる停止 |
| | 15 | ESAO | 1: 補間データのレンジオーバによる停止 |
| | 16 | ESEE | 1: エンコーダ信号エラー(停止しない) |
| | 17 | ESPE | 1: パルス入力信号エラー(停止しない) |
| <p>以下の場合に ESDT=1 になります。</p> <p>1 軸だけ直線補間 1 モード(MOD=60h,61h,68h,69h)にしてスタートコマンドを書き込んだ時。</p> <p>1 軸だけ円弧補間モード(MOD=64h,65h,66h,67h,6Ch,6Dh)にしてスタートコマンドを書き込んだ時。</p> <p>円弧補間モードで RIP 設定(円弧中心座標)を(0,0)にしてスタートコマンドを書き込んだ時。</p> <p>3 軸または 4 軸を円弧補間モードにしてスタートコマンドを書き込んだ時。</p> <p>直線補間 2 モード(MOD=62h,63h,6Ah,6Bh), RIP=0 の状態でスタートコマンドを書き込んだ時。</p> <p>U 軸同期の円弧補間モード(MOD=66h,67h)でスタートコマンドを書き込んだ時に U 軸が動作しない時。</p> <p>または円弧補間動作中に U 軸が動作完了になった時。</p> | | | |

| | |
|-------------|---|
| VC++ 記述例 | <pre>short ret; //関数の戻り値 DWORD ests; //エラーステータス ret = hlt500_ReadErrorSts(hDevID, 1, &ests); //Y 軸を指定</pre> |
|-------------|---|

2.4.3 hlt500_ReadEventSts() イベントステータスの読出し

| | |
|-----|---|
| No. | 18 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のイベントステータスを読出します。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_ReadEventSts (DWORD hDevID, WORD axis, DWORD* ists); |

| | | | |
|--------|------------------------------------|------|-------------------------|
| 引数 | 説明 | | |
| hDevID | デバイスハンドル | | |
| axis | 軸指定 | | |
| ists | イベントステータス | | |
| | ビット | 名称 | 説明 |
| | 0 | IREN | 1:動作完了報告 |
| | 1 | IRN | 1:次動作継続スタート報告 |
| | 2 | IRNM | 1:動作用プリレジスタ書込み可能報告 |
| | 3 | IRND | 1:CMP5 用プリレジスタ書込み可能報告 |
| | 4 | IRUS | 1:加速開始報告 |
| | 5 | IRUE | 1:加速終了報告 |
| | 6 | IRDS | 1:減速開始報告 |
| | 7 | IRDE | 1:減速終了報告 |
| | 8 | IRC1 | 1:CMP1 比較条件成立報告(+SLS) |
| | 9 | IRC2 | 1:CMP2 比較条件成立報告(-SLS) |
| | 10 | IRC3 | 1:CMP3 比較条件成立報告(脱調検出用途) |
| | 11 | IRC4 | 1:CMP4 比較条件成立報告 |
| | 12 | IRC5 | 1:CMP5 比較条件成立報告 |
| | 15 | IROL | 1:OLSon によるカウンタ値ラッチ報告 |
| | 16 | IRSD | 1:DLS 信号 OFF→ON 報告 |
| | 17 | IRPD | 1:+DR 信号 OFF→ON 報告 |
| | 18 | IRMD | 1:-DR 信号 OFF→ON 報告 |
| | 19 | IRSA | 1:STA 信号 OFF→ON 報告 |
| | イベントステータスを使用するためにはイベントマスクの設定が必要です。 | | |

| | |
|-------------|---|
| VC++ 記述例 | short ret; //関数の戻り値 DWORD ists; //サブステータス ret = hlt500_ReadEventSts(hDevID, 1, &ists); //Y 軸を指定 |
|-------------|---|

2.4.4 hlt500_ReadSubSts() サブステータスの読出し

| | | | |
|-------------|--|------|---------------------------|
| No. | 19 | | |
| 機能 | デバイスハンドルで指定された CPD 指定軸のサブステータスを読出します。 | | |
| 開発言語 | 書 式 | | |
| VC++ | short hlt500_ReadSubSts (DWORD hDevID, WORD axis, WORD* ssts); | | |
| 引 数 | 説 明 | | |
| hDevID | デバイスハンドル | | |
| axis | 軸指定 | | |
| ssts | サブステータス | | |
| | ビット | 名 称 | 説 明 |
| | 0 | SVON | '1':"SVON"出力中 |
| | 1 | SVRS | '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 |
| VC++ 記述例 | <pre>short ret; //関数の戻り値 WORD ssts; //サブステータス ret = hlt500_ReadSubSts(hDevID, 1, &ssts); //Y 軸を指定</pre> | | |

2.4.5 hlt500_ReadExSts() 拡張ステータスの読出し

| | |
|-----|---------------------------------------|
| No. | 20 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の拡張ステータスを読出します。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_ReadExSts (DWORD hDevID, WORD axis, DWORD* exsts); |

| | | | | | |
|--------|----------|--|----------------------|-----------------|--------------------|
| 引数 | 説明 | | | | |
| hDevID | デバイスハンドル | | | | |
| axis | 軸指定 | | | | |
| exsts | 拡張ステータス | | | | |
| | ビット | 名称 | 説明 | | |
| | 3-0 | CND3-0 | 動作状態 | | |
| | | | 0000:停止中 | 1000:パルス入力待ち | |
| | | | 0001:DR 入力待ち | 1001:FA 定速動作中 | |
| | | | 0010:STA 入力待ち | 1010:FL 定速動作中 | |
| | | | 0011:条件付きスタート待ち状態 | 1011:加速中 | |
| | | | 0100:他軸の停止待ち | 1100:FH 定速動作中 | |
| | | | 0101:SVCTRCL タイマ完了待ち | 1101:減速中 | |
| | | | 0110:方向変化タイマ完了待ち | 1110:INPOSon 待ち | |
| | | | 0111:バックラッシュ/スリップ補正中 | 1111:その他 | |
| | | | 4 | SDIR | 動作方向 0:+方向, 1:-方向 |
| | | | 5 | SSTA | 1:同時スタート信号(STA) on |
| | 6 | SSTP | 1:同時停止信号(STP) on | | |
| | 7 | SEMG | 1:EMGon | | |
| | 8 | SPCS | 1:PCSon | | |
| | 9 | SERC | 1:サーボ偏差カウンタ・クリア信号 on | | |
| | 10 | SEZ | 1:エンコーダZ相信号 on | | |
| | 11 | SDRP | 1:+DRon | | |
| | 12 | SDRM | 1:-DRon | | |
| 15 | SDIN | 1:DLSon | | | |
| 16 | SINP | 1:INPOSon | | | |
| 19,18 | PFC1,0 | RCMP5 用プリレジスタの使用状態 00:未確定, 01:レジスタ確定, 10: 1st レジスタ確定, 11: 2nd レジスタ確定(レジスタフル) | | | |
| 21,20 | PFM1,0 | 動作用プリレジスタの使用状態 00:未確定, 01:レジスタ確定, 10: 1st レジスタ確定, 11: 2nd レジスタ確定(レジスタフル) | | | |

| | |
|-------------|--|
| VC++ 記述例 | short ret; //関数の戻り値 DWORD exsts; //拡張ステータス ret = hlt500_ReadExSts(hDevID, 1, &exsts); //Y 軸を指定 |
|-------------|--|

2.4.6 hlt500_ReadSpd() 指令速度の読出し

| | |
|-------------|--|
| No. | 21 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の指令速度を読出します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_ReadSpd (DWORD hDevID, WORD axis, WORD* spd); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| spd | 指令速度 (速度倍率を含まない ※) |
| VC++ 記述例 | short ret; //関数の戻り値 WORD spd; //速度 ret = hlt500_ReadSpd(hDevID, 0, &spd); //X軸を指定, 格納先 |
| 備考 | ※ 実際の指令速度[pps]は“読出したデータ×速度倍率”で算出します。 |

2.4.7 hlt500_ReadCtr() カウンタの読出し

| | |
|-------------|---|
| No. | 22 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の,指定されたカウンタを読出します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_ReadCtr(DWORD hDevID, WORD axis, WORD selctr, long* count); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| 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] |
| count | カウンタ値 |
| VC++ 記述例 | short ret; //関数の戻り値 long* ctr1; //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 ret = hlt500_ReadCtr(hDevID, 0, 1, &ctr1); |
| 備考 | 入カソース カウンタ 1・・・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) カウンタ 4・・・汎用カウンタ(指令パルス, エンコーダ入力等) |

2.5 動作設定

2.5.1 hlt500_SetFLSpd() ベース速度の設定

| | |
|-------------|---|
| No. | 23 |
| 機能 | デバイスハンドルで指定された CPD の指定軸のベース速度(pps)を速度倍率で除算した値を設定. |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetFLSpd (DWORD hDevID, WORD axis, DWORD spd); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| spd | ベース速度レジスタ値(RFL)[1~65535, RFL<RFH で設定] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetFLSpd(hDevID, 1, 500); //Y 軸を指定, RFL=500 |
| 備考 | ベース速度・加減速動作時の立ち上がりの速度(本速度から加速, 本速度まで減速して停止) |

2.5.2 hlt500_SetAuxSpd() 補助速度の設定

| | |
|-------------|--|
| No. | 24 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の補助速度(pps)を速度倍率で除算した値を設定. |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetAuxSpd(DWORD hDevID, WORD axis, DWORD spd); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| spd | 補助速度レジスタ値(RFA)[1~65535] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetAuxSpd(hDevID, 1, 500); //Y 軸を指定, RFA=500 |
| 備考 | 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されます. |

2.5.3 hlt500_SetAccRate() 加速レートの設定

| | |
|-------------|--|
| No. | 25 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の加速レート(RUR)を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetAccRate(DWORD hDevID, WORD axis, DWORD rate); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| rate | 加速レート(RUR)[1~65535] |
| VC++ 記述例 | <pre>short ret; //関数の戻り値 DWORD rur; //RUR // RUR 計算 RFH=5000, RFL=500, 加速時間=100msec(直線加速時) ret = hlt500_CalAccRate(&rur, 100, 5000, 500, 0, 0); //Y 軸に設定 ret = hlt500_SetAccRate(hDevID, 1, rur);</pre> |
| 備考 | <p>加速レート(RUR)と加速時間の関係 RFH:動作速度レジスタ RFL:ベース速度レジスタ RUR:加速レートレジスタ</p> <p>直線加速 $\text{加速時間[sec]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 4}{19,660,800}$</p> <p>直線部分のないS字加速 $\text{加速時間[sec]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 8}{19,660,800}$</p> |

2.5.4 hlt500_SetDecRate() 減速レートの設定

| | |
|-----|--|
| No. | 26 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の減速レート(RDR)を設定します。 加速時間と減速時間が異なる場合に RDR の設定を行います。 |

| | |
|------|---|
| 開発言語 | 書式 |
| VC++ | short hlt500_SetDecRate(DWORD hDevID, WORD axis, DWORD rate); |

| | |
|--------|--|
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| rate | 減速レート(RDR)[0~65535] 0を設定した場合は RUR=RDR として減速レートが決定されます。(加速時間=減速時間) |

| | |
|-------------|--|
| VC++ 記述例 | <pre>short ret; //関数の戻り値 DWORD rdr; //RDR // RDR 計算 RFH=5000, RFL=500, 減速時間=200msec(直線加速時) ret = hlt500_CalAccRate(&rdr, 200, 5000, 500, 0, 0); //Y 軸に設定 ret = hlt500_SetDecRate(hDevID, 1, rdr);</pre> |
|-------------|--|

| | |
|----|---|
| 備考 | <p>減速レート(RDR)と減速時間の関係 RFH:動作速度レジスタ RFL:ベース速度レジスタ RDR:減速レートレジスタ</p> <p>直線減速</p> $\text{加速時間[sec]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RDR} + 1) \times 4}{19,660,800}$ <p>直線部分のない S 字減速</p> $\text{加速時間[sec]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RDR} + 1) \times 8}{19,660,800}$ |
|----|---|

2.5.5 hlt500_SetMult() 速度倍率レジスタ値の設定

| | |
|-----|--|
| No. | 27 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の速度倍率レジスタ値(RMG)を設定します。 |

| | |
|------|---|
| 開発言語 | 書式 |
| VC++ | short hlt500_SetMult(DWORD hDevID, WORD axis, DWORD rmg); |

| | |
|--------|-------------------|
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| rmg | 速度倍率レジスタ値[2~4095] |

| | |
|-------------|---|
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetMult(hDevID, 1, 299); //Y 軸を指定, 速度倍率 1 倍 |
|-------------|---|

| | | | | |
|----|--|----------|-----------------|----------------|
| 備考 | 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 | | | |
| | $\text{速度[PPS]} = \text{RFx} \times \text{速度倍率} = \frac{\text{RFx} \times 300}{\text{RMG} + 1} \quad \text{速度倍率設定値 (RMG)} = \frac{300}{\text{速度倍率}} - 1$ | | | |
| | 設定例 | | | |
| | RMG(DEC) | RMG(HEX) | 速度倍率 | 出力速度範囲(pps) |
| | 2999 | bb7 | 0.1 | 0.1 ~ 6,553.5 |
| | 1499 | 5db | 0.2 | 0.2 ~ 13,107 |
| | 599 | 257 | 0.5 | 0.5 ~ 32,767.5 |
| | 299 | 12b | 1 | 1 ~ 65,535 |
| | 149 | 95 | 2 | 2 ~ 131,070 |
| | 59 | 3b | 5 | 5 ~ 327,675 |
| 29 | 1d | 10 | 10 ~ 655,350 | |
| 14 | e | 20 | 20 ~ 1,310,700 | |
| 11 | b | 25 | 25 ~ 1,638,375 | |
| 9 | 9 | 30 | 30 ~ 1,966,050 | |
| 5 | 5 | 50 | 50 ~ 3,276,750 | |
| 4 | 4 | 60 | 60 ~ 3,932,100 | |
| 3 | 3 | 75 | 75 ~ 4,915,125 | |
| 2 | 2 | 100 | 100 ~ 6,553,500 | |

2.5.6 hlt500_SetEventMask() イベントマスクの設定

| | |
|-------------|---|
| No. | 28 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のイベントマスクを設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SetEventMask(DWORD hDevID, WORD axis, DWORD mask); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| mask | <p>イベントマスクデータ</p> <p>00001h 正常停止時</p> <p>00002h 次動作継続スタート時</p> <p>00004h 動作用プリレジスタフルから空きができた時</p> <p>00008h 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</p> <p>00010h 加速開始時</p> <p>00020h 加速終了時</p> <p>00040h 減速開始時</p> <p>00080h 減速終了時</p> <p>00100h コンパレータ 1 条件成立時</p> <p>00200h コンパレータ 2 条件成立時</p> <p>00400h コンパレータ 3 条件成立時</p> <p>00800h コンパレータ 4 条件成立時</p> <p>01000h コンパレータ 5 条件成立時</p> <p>02000h CLR 入力によるカウント値のクリア時</p> <p>04000h LATCH 入力によるカウント値のラッチ時</p> <p>08000h OLS 入力によるカウント値のラッチ時</p> <p>10000h DLS 入力 ONN時</p> <p>20000h ±DR 入力変化時</p> <p>40000h CSTA 信号入力 ON 時</p> <p>上記データの OR したデータを与えることで複数のイベント報告指定。</p> |
| VC++ 記述例 | <pre>short 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 = hlt500_SetEventMask(hDevID, 1, 0x1001);</pre> |

2.5.7 hlt500_SetDecPoint() 減速開始点の設定

| | |
|-----|-------------------------------------|
| No. | 29 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の減速開始点を設定します。 |

| | |
|------|--|
| 開発言語 | 書式 |
| VC++ | short hlt500_SetDecPoint(DWORD hDevID, WORD axis, long dstnc); |

| | |
|--------|--------------|
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| dstnc | 減速開始点(pulse) |

| | |
|-------------|--|
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SetDecPoint(hDevID, 1, 300); //Y 軸を指定, 減速開始点 |
|-------------|--|

| | |
|----|---|
| 備考 | <p>残移動量が減速開始点以下になると減速を開始します。</p> <p>[減速開始点計算自動時] 自動で計算された減速開始点に対するオフセット値(単位: pulse)となります。 +の値を設定すると減速が早めに開始され, 減速後ベース速度で動作します。 -の値を設定すると減速が遅めに開始され, ベース速度に到達する前に動作完了となります。 0を設定すると通常の動作になります。</p> <p>[減速開始点計算手動時] 残移動量が設定した値(pulse)以下になると減速を開始します。 この場合に設定する値は次式の様になります。(ベース速度到達時に動作完了になる値)</p> <p>直線減速</p> $\text{最適値[パルス]} = \frac{(RFH^2 - RFL^2) \times (RDR + 1)}{(RMG + 1) \times 32,768}$ <p>直線部分のない S 字減速</p> $\text{最適値[パルス]} = \frac{(RFH^2 - RFL^2) \times (RDR + 1) \times 2}{(RMG + 1) \times 32,768}$ |
|----|---|

2.6 運用設定

2.6.1 hlt500_WritOpeMode() 動作モードの設定

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------------|---|----------------------|----------------------|----------------------|-----------------------|---------------|---------------|-----------------|-----------------|----------------|----------------|------------------------|------------------------|------------------------|------------------------|--------------------|--------------------|------------|---------------------------|------------------|----------------------|-----------------|-----------------|-----------|------------------------|---------------------------|---------------------------|-------------------------|--|--------------|------------|-----------------|------------------|
| No. | 30 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 機能 | デバイスハンドルで指定された CPD 指定軸の動作モードを設定します。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 開発言語 | 書式 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VC++ | short hlt500_WritOpeMode(DWORD hDevID, WORD axis, WORD mode); | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 引数 | 説明 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| hDevID | デバイスハンドル | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Axis | 軸指定 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| mode | <p>動作モード</p> <table border="0"> <tr> <td>00h:コマンド制御による＋方向連続送り</td> <td>08h:コマンド制御による－方向連続送り</td> </tr> <tr> <td>01h:パルス入力による連続送り(※1)</td> <td>02h:±DR 入力による連続送り(※2)</td> </tr> <tr> <td>10h:＋方向原点復帰動作</td> <td>18h:－方向原点復帰動作</td> </tr> <tr> <td>12h:＋方向原点抜け出し動作</td> <td>1ah:－方向原点抜け出し動作</td> </tr> <tr> <td>15h:＋方向原点サーチ動作</td> <td>1dh:－方向原点サーチ動作</td> </tr> <tr> <td>20h:＋ELS 又は＋SLS 位置まで動作</td> <td>28h:－ELS 又は－SLS 位置まで動作</td> </tr> <tr> <td>22h:＋ELS 又は＋SLS 抜け出し動作</td> <td>2ah:－ELS 又は－SLS 抜け出し動作</td> </tr> <tr> <td>24h:＋方向に Z 相カウント動作</td> <td>2ch:－方向に Z 相カウント動作</td> </tr> <tr> <td>41h:位置決め動作</td> <td>42h:PCS 位置決め動作(ライブラリ関数のみ)</td> </tr> <tr> <td>44h:指令位置 0 点復帰動作</td> <td>45h:機械位置 0 点復帰動作(※1)</td> </tr> <tr> <td>46h:＋方向 1 パルス動作</td> <td>4eh:－方向 1 パルス動作</td> </tr> <tr> <td>47h:タイマ動作</td> <td>51h:パルス入力による位置決め動作(※1)</td> </tr> <tr> <td>54h:パルス入力指令位置 0 点復帰動作(※1)</td> <td>55h:パルス入力機械位置 0 点復帰動作(※1)</td> </tr> <tr> <td>56h:±DR 入力による位置決め動作(※2)</td> <td></td> </tr> <tr> <td>60h:連続直線補間動作</td> <td>61h:直線補間動作</td> </tr> <tr> <td>64h:CW 方向円弧補間動作</td> <td>65h:CCW 方向円弧補間動作</td> </tr> </table> | 00h:コマンド制御による＋方向連続送り | 08h:コマンド制御による－方向連続送り | 01h:パルス入力による連続送り(※1) | 02h:±DR 入力による連続送り(※2) | 10h:＋方向原点復帰動作 | 18h:－方向原点復帰動作 | 12h:＋方向原点抜け出し動作 | 1ah:－方向原点抜け出し動作 | 15h:＋方向原点サーチ動作 | 1dh:－方向原点サーチ動作 | 20h:＋ELS 又は＋SLS 位置まで動作 | 28h:－ELS 又は－SLS 位置まで動作 | 22h:＋ELS 又は＋SLS 抜け出し動作 | 2ah:－ELS 又は－SLS 抜け出し動作 | 24h:＋方向に Z 相カウント動作 | 2ch:－方向に Z 相カウント動作 | 41h:位置決め動作 | 42h:PCS 位置決め動作(ライブラリ関数のみ) | 44h:指令位置 0 点復帰動作 | 45h:機械位置 0 点復帰動作(※1) | 46h:＋方向 1 パルス動作 | 4eh:－方向 1 パルス動作 | 47h:タイマ動作 | 51h:パルス入力による位置決め動作(※1) | 54h:パルス入力指令位置 0 点復帰動作(※1) | 55h:パルス入力機械位置 0 点復帰動作(※1) | 56h:±DR 入力による位置決め動作(※2) | | 60h:連続直線補間動作 | 61h:直線補間動作 | 64h:CW 方向円弧補間動作 | 65h:CCW 方向円弧補間動作 |
| 00h:コマンド制御による＋方向連続送り | 08h:コマンド制御による－方向連続送り | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01h:パルス入力による連続送り(※1) | 02h:±DR 入力による連続送り(※2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10h:＋方向原点復帰動作 | 18h:－方向原点復帰動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12h:＋方向原点抜け出し動作 | 1ah:－方向原点抜け出し動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15h:＋方向原点サーチ動作 | 1dh:－方向原点サーチ動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20h:＋ELS 又は＋SLS 位置まで動作 | 28h:－ELS 又は－SLS 位置まで動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22h:＋ELS 又は＋SLS 抜け出し動作 | 2ah:－ELS 又は－SLS 抜け出し動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24h:＋方向に Z 相カウント動作 | 2ch:－方向に Z 相カウント動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41h:位置決め動作 | 42h:PCS 位置決め動作(ライブラリ関数のみ) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 44h:指令位置 0 点復帰動作 | 45h:機械位置 0 点復帰動作(※1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 46h:＋方向 1 パルス動作 | 4eh:－方向 1 パルス動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 47h:タイマ動作 | 51h:パルス入力による位置決め動作(※1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 54h:パルス入力指令位置 0 点復帰動作(※1) | 55h:パルス入力機械位置 0 点復帰動作(※1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 56h:±DR 入力による位置決め動作(※2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 60h:連続直線補間動作 | 61h:直線補間動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 64h:CW 方向円弧補間動作 | 65h:CCW 方向円弧補間動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VC++ | short ret; //関数の戻り値 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 記述例 | ret = hlt500_WritOpeMode(hDevID, 1, 0x41); //Y 軸を指定, 位置決め動作 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

2.6.2 hlt500_WritFHSpd() 動作速度の設定

| | |
|-------------|--|
| No. | 31 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の動作速度(pps)に速度倍率で除算した値(RFH)を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_WritFHSpd(DWORD hDevID, WORD axis, DWORD spd); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| spd | 動作速度レジスタ値(RFH)[2~65535, RFL<RFH で設定] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_WritFHSpd(hDevID, 1, 5000); //Y 軸を指定, RFH=5000 |

2.6.3 hlt500_WritPos() 位置決め移動量の設定

| | |
|-------------|---|
| No. | 32 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の位置決め動作の移動量を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_WritPos(DWORD hDevID, WORD axis, long dstnc); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| dstnc | 移動量(pulse)[-134,217,728~134,217,727(28ビット)] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_WritPos(hDevID, 1, 15000); //Y 軸を指定, +方向 15000pulse |

2.6.4 hlt500_WritLine() 直線補間の移動量の設定

| | |
|-------------|--|
| No. | 33 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の直線補間の移動量を設定します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_WritLine(DWORD hDevID, WORD axis, long dstnc); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| dstnc | 移動量(pulse)[-134,217,728~134,217,727(28ビット)] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_WritLine(hDevID, 0, -30000); //補間代表軸: X 軸を指定, 一方向 30000pulse ret = hlt500_WritLine(hDevID, 1, -15000); //従軸: Y 軸を指定, 一方向 15000pulse |

2.6.5 hlt500_WritCircl() 円弧補間の移動量の設定

| | |
|-------------|---|
| No. | 34 |
| 機能 | デバイスハンドルで指定された CPD 指定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<運用編>」を参照して下さい。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_WritCircl(DWORD hDevID,WORD axis,long dstnc1,long dstnc2, long center1,long center2); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| dstnc1 | 終点位置 1[-134,217,728~134,217,727(28ビット)] |
| dstnc2 | 終点位置 2[-134,217,728~134,217,727(28ビット)] |
| center1 | 中心位置 1[-134,217,728~134,217,727(28ビット)] |
| center2 | 中心位置 2[-134,217,728~134,217,727(28ビット)] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_WritCircl(hDevID, 0, 0, 0, 1000, 0); //XY 軸円弧(半径 1000pulse の真円) //終点(X,Y)=(0,0),中心(X,Y)=(1000,0) |

2.6.6 hlt500_WritCtr() カウンタプリセット

| | |
|-------------|---|
| No. | 35 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の指定されたカウンタへプリセット(座標値の書込み)をします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_WritCtr(DWORD hDevID, WORD axis, long preset, WORD selctr); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| preset, | プリセット値[-134,217,728~134,217,727(28ビット)] |
| 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] |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_WritCtr(hDevID, 0, 5000, 1); //X 軸, プリセット値=5000, カウンタ 1 |

2.7 動作制御指令

2.7.1 hlt500_DecStop() 減速停止

| | |
|-------------|--|
| No. | 36 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸を減速停止します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_DecStop (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_DecStop(hDevID, 0x07); //XYZ 軸減速停止 |

2.7.2 hlt500_QuickStop() 即停止

| | |
|-------------|---|
| No. | 37 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸を即停止します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_QuickStop(DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_QuickStop(hDevID, 0x07); //XYZ 軸即停止 |

2.7.3 hlt500_EmgStop() 非常停止

| | |
|-------------|--|
| No. | 38 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸を非常停止します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_EmgStop (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_EmgStop(hDevID, 0x07); //XYZ 軸非常停止 |

2.7.4 hlt500_AccStart() 加速スタート

| | |
|-------------|---|
| No. | 39 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸を加速スタートします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_AccStart (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_AccStart(hDevID, 0x07); //XYZ 軸加速スタート |

2.7.5 hlt500_CnstStartFH () FH 定速スタート

| | |
|-------------|---|
| No. | 40 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸をFH定速スタートします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_CnstStartFH (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_CnstStartFH (hDevID, 0x07); //XYZ 軸 FH 定速スタート |

2.7.6 hlt500_CnstStartFL () FL 定速スタート

| | |
|-------------|---|
| No. | 41 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸をFL定速スタートします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_CnstStartFL (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_CnstStartFL (hDevID, 0x07); //XYZ 軸 FL 定速スタート |

2.7.7 hlt500_CnstStartByDec () FH定速スタート後減速停止

| | |
|-------------|--|
| No. | 42 |
| 機能 | デバイスハンドルで指定された CPD の指定された軸をFH定速スタート後減速停止します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_CnstStartByDec(DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_CnstStartByDec(hDevID, 0x07); //XYZ 軸FH定速スタート後減速停止 |

2.7.8 hlt500_SvOn() SVON オン

| | |
|-------------|--|
| No. | 43 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の SVON をオンします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SvOn (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SvOn(hDevID, 0x07); //XYZ 軸サーボオン |

2.7.9 hlt500_SvOff() SVON オフ

| | |
|-------------|--|
| No. | 44 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の SVON をオフします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SvOff (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SvOff(hDevID, 0x07); //XYZ 軸 SVON オフ |

2.7.10 hlt500_SvResetOn() SVRST オン

| | |
|-------------|---|
| No. | 45 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の SVRST をオンします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SvResetOn (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SvResetOn(hDevID, 0x07); //XYZ 軸 SVRST オン |

2.7.11 hlt500_SvResetOff() SVRST オフ

| | |
|-------------|--|
| No. | 46 |
| 機能 | デバイスハンドルで指定された CPD 指定軸の SVRST をオフします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_SvResetOff(DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_SvResetOff(hDevID, 0x07); //XYZ 軸 SVRST オフ |

2.7.12 hlt500_PMON() パルスモータ励磁オン

| | |
|-------------|---|
| No. | 47 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のパルスモータ励磁をオンします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_PMON (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_PMON(hDevID, 0x07); //XYZ 軸励磁オン |
| 備考 | SVON をパルスモータドライバのモーターフリー(出力電流オフ)信号に接続した場合に使用可能。 |

2.7.13 hlt500_PMOff() パルスモータ励磁オフ

| | |
|-------------|--|
| No. | 48 |
| 機能 | デバイスハンドルで指定された CPD 指定軸のパルスモータ励磁オフ(モーターフリー)にします。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_PMOff (DWORD hDevID, WORD axbit); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axbit | 軸指定(複数軸ビット指定) |
| VC++ 記述例 | short ret; //関数の戻り値 ret = hlt500_PMOff(hDevID, 0x07); //XYZ 軸パルスモータ励磁オフ |
| 備考 | SVON をパルスモータドライバのモーターフリー(出力電流オフ)信号に接続した場合に使用可能。 |

2.8 加減速レートの計算

2.8.1 hlt500_CalAccRate() 加減速レートの計算

| | |
|-------------|---|
| No. | 49 |
| 機能 | 加減速時間, RFH, RFL 等より加減速レート(RUR, RDR)を計算します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_CalAccRate(DWORD* r, DWORD t, DWORD fh, DWORD fl, WORD p, WORD s); |
| 引数 | 説明 |
| r | 計算結果(加減速レート[1~65535]) |
| t | 加減速時間(msec) |
| fh | FHレジスタ値[1~65535] |
| fl | FLレジスタ値[1~65535] |
| p | 加減速形式[0:直線,1:S字] |
| s | 加減速形式をS字にした場合のS字速度区間[1~32767], 0を指定した場合は直線部分なし |
| VC++ 記述例 | <pre>short 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 = hlt500_CalAccRate(&rate, 200, 10000, 500, 0, 0); //加速レート設定 ret = hlt500_SetAccRate(hDevID, 0, rate);</pre> |

2.9 内部情報の取得

2.9.1 hlt500_GetBoardCode() ボード固有コードの取得

| | |
|-------------|---|
| No. | 50 |
| 機能 | デバイスハンドルで指定された CPD のボード固有コードを取得します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_GetBoardCode(DWORD hDevID, short *devtyp); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| devtyp | デバイスコードを格納するアドレス |
| VC++ 記述例 | short ret; //関数の戻り値 short devtyp //ボードコード格納先 ret = hlt500_GetBoardCode(hDevID, &devtyp); |
| 備考 | 取得したコードの意味はドライバ関数“lt500_GetBoardCode”をご覧ください。 |

2.9.2 hlt500_GetDevVerNo() バージョン情報の取得

| | |
|-------------|---|
| No. | 51 |
| 機能 | デバイスドライバとドライバ関数のバージョンを取得します。 |
| 開発言語 | 書式 |
| VC++ | Short hlt500_GetDevVerNo(DWORD *verno); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| verno | バージョン番号を格納するアドレス |
| VC++ 記述例 | short ret; //関数の戻り値 DWORD verno //バージョン番号格納先 ret = hlt500_GetDevVerNo(&verno); |
| 備考 | データ書式などの詳細はドライバ関数“lt500_GetDevVerNo”をご覧ください。 |

2.9.3 hlt500_GetLastDetailError() ローカルエラーコードの取得

| | |
|-------------|---|
| No. | 52 |
| 機能 | 最後に発生したエラー詳細を取得します。 |
| 開発言語 | 書式 |
| VC++ | short hlt500_GetLastDetailError(WORD *derr) |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| verno | 詳細エラーコードを格納するアドレス |
| VC++ 記述例 | short ret; //関数の戻り値 WORD derr; //詳細エラーコード格納先 ret = hlt500_GetLastDetailError(&derr); |
| 備考 | 詳細エラーコードはドライバ関数“lt500_GetLastDetailError”をご覧ください。 |

3. ドライバ関数

ドライバ関数には、CPD の制御を行うための基本的な関数です。

各関数は RSL(リアルタイムシェアードライブラリ) hicpd500.rsl として提供され、上位アプリケーションからは外部関数として呼び出して使用します。

3.1 関数の種類

| No. | 関数名 | 機能 | 備考 |
|-----|--------------------------|---------------------|----------------|
| 1 | It500_ApiStartup | ドライバAPI関数の活性化 | |
| 2 | It500_ApiCleanup | ドライバAPI関数の資源解放 | |
| 3 | It500_GetDeviceCount() | ボード枚数の取得 | |
| 4 | It500_GetDeviceInfo() | デバイス情報の取得 | |
| 5 | It500_OpenDevice() | デバイスのオープン | |
| 6 | It500_CloseDevice() | デバイスのクローズ | |
| 7 | It500_rMstsW() | メインステータスの読出し | |
| 8 | It500_rSstsW() | サブステータスの読出し | |
| 9 | It500_wCmdW() | 制御コマンドの書込 | |
| 10 | It500_rReg() | レジスタの読出し | |
| 11 | It500_wReg() | レジスタへ書込 | |
| 12 | It500_rPortB() | オプションポートの読出し(1 バイト) | |
| 13 | It500_wPortB() | オプションポートへ書込(1 バイト) | |
| 14 | It500_rPortW() | オプションポートの読出し(2 バイト) | |
| 15 | It500_wPortW() | オプションポートへ書込(2 バイト) | |
| 16 | It500_rBufDW() | 入出力バッファの読出し | |
| 17 | It500_wBufDW() | 入出力バッファへ書込 | |
| 18 | It500_SetIntCall | 割込み機能の登録 | |
| 19 | It500_ResetIntCall | 割込み機能の削除 | |
| 20 | It500_WaitInt | 割込みイベント待ち | |
| 21 | It500_GetIntData | 割込みイベント要因の取得 | |
| 22 | It500_GetDevVerNo | バージョン番号の取得 | Ver1.40以降で使用可能 |
| 23 | It500_GetBoardCode | ボード固有コードの取得 | Ver1.40以降で使用可能 |
| 24 | It500_GetLastDetailError | 詳細エラーコードの取得 | Ver1.40以降で使用可能 |

表 3.1-1 ドライバ関数一覧

3.2 プリレジスタ

RMV, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RUS, RDS, RCMP5の各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、PCL6045のプリレジスタは下図のような2段構成になっています。プリレジスタへ動作用データを書き込むと、まず「1stプリレジスタ」に対して書き込みます。この時「1stプリレジスタ」にデータが残っている時は「2ndプリレジスタ」に書き込まれます。1つの動作が終了すると「レジスタ」が空になり、保留されている「1stプリレジスタ」のデータは「レジスタ」へ、「2ndプリレジスタ」のデータは「1stプリレジスタ」へデータがシフトされます。

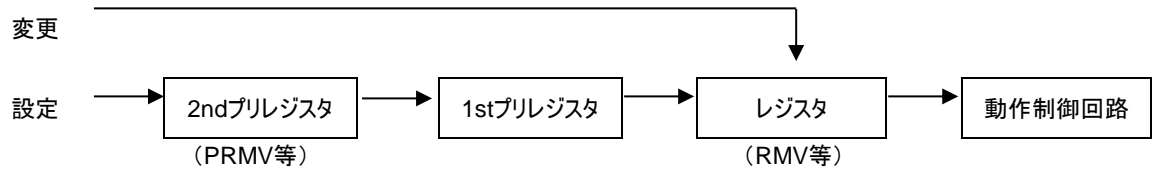


表 3.1-2 プリレジスタ

動作用データは「プリレジスタ」に書込みます。

速度変更、位置のオーバーライド等、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。

3.3 ドライバ関数の戻り値

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。

3.4 関数の詳細

3.4.1 It500_ApiStartup() ドライバ API 関数の活性化

| | |
|-------------|--------------------------------------|
| No. | 1 |
| 機能 | HLS-Cpd500 用 API 関数を活性化 (API の初期化処理) |
| 開発環境 | 書式 |
| VC++ | short It500_ApiStartup(void); |
| 引数 | 説明 |
| | なし |
| VC++ 記述例 | short ret; ret = ApiStartup(); |

3.4.2 It500_ApiCleanup() ドライバ API 関数の資源解放

| | |
|-------------|--|
| No. | 2 |
| 機能 | HLS-Cpd500 用 API 関数の利用を完了 (API の資源を解放) |
| 開発環境 | 書式 |
| VC++ | short It500_ApiCleanup(void); |
| 引数 | 説明 |
| | なし |
| VC++ 記述例 | short ret; ret = ApiCleanup(); |

3.4.3 It500_GetDeviceCount() ボード枚数の取得

| | |
|-------------|---|
| No. | 3 |
| 機能 | 現在パソコンに装着されている CPD シリーズボードの枚数を取得します。 |
| 開発環境 | 書式 |
| VC++ | short It500_GetDeviceCount(short *DevCnt) |
| 引数 | 説明 |
| DevCnt | CPD シリーズボードの枚数 |
| VC++ 記述例 | short ret; //関数の戻り値 short count; //CPD ボードの枚数 ret = It500_GetDeviceCount(&count); |

3.4.4 It500_GetDeviceInfo() デバイス情報の取得

| | |
|-------------|---|
| No. | 4 |
| 機能 | 現在パソコンに装着されている CPD シリーズボードの枚数と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。 |
| 開発環境 | 書式 |
| VC++ | short It500_GetDeviceInfo(short * DevCnt,, HPC_DEVINFO * DevInfo); |
| 引数 | 説明 |
| DevCnt, | CPD シリーズボードの枚数 (Windows 版同名関数では取得する上限枚数をセットして渡していますが, INtime 版ではボード枚数を返す変数としてのみ使用します) |
| DevInfo | CPD シリーズボードのデバイス情報格納の構造体 |
| VC++ 記述例 | short ret; //関数の戻り値 DWORD count; //CPD ボードの枚数 HPC_DEVINFO HpcDevInfo[2]; //この例では 2 枚の CPD のデバイス情報格納配列 ret = It500_GetDeviceInfo(&count, &HpcDevInfo[0]); |
| 備考 | デバイス情報を格納するエリアはボード枚数分を確保します。 |

3.4.5 It500_OpenDevice() デバイスのオープン

| | |
|-------------|--|
| No. | 5 |
| 機能 | 渡したデバイス情報を持つ CPD シリーズボードをオープンし, 他ボードと識別するためのデバイスハンドルを取得します。以降このデバイスハンドルは目的の CPD シリーズボードにアクセスするためのハンドルとなります。 |
| 開発環境 | 書式 |
| VC++ | short It500_OpenDevice(DWORD* hDeviceID, HPC_DEVINFO *DevInfo); |
| 引数 | 説明 |
| hDeviceID | デバイスハンドル |
| DevInfo | デバイス情報 |
| VC++ 記述例 | short ret; //関数の戻り値 DWORD count; //CPD ボードの枚数 HPC_DEVINFO HpcDevInfo [2]; //2 枚の CPD のデバイス情報格納配列 ret = It500_GetDeviceInfo(&count, &HpcDevInfo[0]); |

3.4.6 It500_CloseDevice() デバイスのクローズ

| | |
|-------------|---|
| No. | 6 |
| 機能 | デバイスハンドルで指定された CPD 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 に対するアクセスはできません。 |
| 開発環境 | 書式 |
| VC++ | short It500_CloseDevice(DWORD hDevID); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| VC++ 記述例 | short ret; //関数の戻り値 ret = It500_CloseDevice(hDevID); |
| 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してください。 |

3.4.7 It500_rMstsW() メインステータスの読出し

| | |
|-------------|--|
| No. | 7 |
| 機能 | デバイスハンドルで指定されたボードの指定軸メインステータスを読出します。 |
| 開発環境 | 書式 |
| VC++ | short It500_rMstsW(DWORD hDevID, WORD axis, WORD* wMsts); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| wMsts | メインステータス |
| VC++ 記述例 | short ret; //関数の戻り値 WORD wMsts; //メインステータス ret = It500_rMstsW(hDevID, 1, & wMsts); //Y 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。 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 b5(SINT)及び b4(SERR)を監視し、これらが'0'の場合は移動中です。 移動が終了すると正常終了では b5(SINT)が'1'となり、イベントステータスを読み出します。(*1) 異常終了では b4(SERR)が'1'となり、エラーステータスの処理をします。 |

(1) メインステータスの内容

| | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| SPDF | SPRF | SEOR | SCMP | SCMP | SCMP | SCMP | SCMP | SSC1 | SSC0 | SINT | SERR | SEND | SRUN | SSCM |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|

| bit | 名称 | 内容 | 備考 |
|-----|--------|---------------------|---|
| 0 | SSCM | 1 = スタートコマンド書き込み済み | スタートコマンド書き込み後、即動作が始まらない場合があります。 (条件付スタートのスタート保留状態など) |
| 1 | SRUN | 1 = RUN 中 | |
| 2 | --- | 予約 | |
| 3 | SEND | 1 = 停止中(電源投入直後は'0') | 本ビットの変化タイミングは、INPOS 制御設定により異なります。 |
| 4 | SERR | 1 = エラーステータスに要因有 | REST 読み出しで 1→0 |
| 5 | SINT | 1 = イベントステータスに要因有 | RIST 読み出しで 1→0 イベントマスク(RIRQ)の設定が必要。 |
| 7,6 | SSC1,0 | 実行中の RMD.b17,16 | 次動作連続実行中の動作確認などに使用 |
| 8 | SCMP1 | 1 = CMP1 比較条件成立状態 | 比較条件成立中の間のみ'1'となります。 (既に通過してしまった場合は'0'となります) 条件が成立したことを監視する場合はイベントステータス (RIST)を使用します。 |
| 9 | SCMP2 | 1 = CMP2 比較条件成立状態 | |
| 10 | SCMP3 | 1 = CMP3 比較条件成立状態 | |
| 11 | SCMP4 | 1 = CMP4 比較条件成立状態 | |
| 12 | SCMP5 | 1 = CMP5 比較条件成立状態 | |
| 13 | SEOR | 1 = 位置オーバーライド失敗 | 本ステータスリードでクリア。停止中に RMV を書込んでも"1". |
| 14 | SPRF | 1 = 動作用プリレジスタフル | |
| 15 | SPDF | 1 = コンパレータ用プリレジスタフル | |

*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"[RIRQ.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. b13(SEOR)は HPCI-CPD574N, HPCI-CPD578N, HPCIe-CPD674N, HPCIe-CPD678N のみ対応しています。

3.4.8 It500_rSstsW() サブステータスの読出し

| | |
|-----|-------------------------------------|
| No. | 8 |
| 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ます。 |

| | |
|------|---|
| 開発環境 | 書式 |
| VC++ | short It500_rSstsW(DWORD hDevID, WORD axis, WORD* wSsts); |

| | |
|--------|----------|
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| wSsts | サブステータス |

| | |
|-------------|--|
| VC++ 記述例 | short ret; //関数の戻り値 WORD wSsts; //サブステータス ret = It500_rSstsW(hDevID, 1, & wSsts); //Y 軸 |
|-------------|--|

| | |
|----|---|
| 備考 | b0(SVON), ビット 1(SVRS)はサーボ出力指令のモニタビットです。 b12(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。 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。 |
|----|---|

(1) 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 | SVRS | SVON |

| bit | 名称 | 説明 | 備考 |
|-----|-------|-----------------|---|
| 0 | SVON | '1' = SVON ON | 指令状態のモニタ |
| 1 | SVRST | '1' = SVRST ON | |
| 7-2 | --- | 不使用(予約) | |
| 8 | SFU | '1' = 加速中 | レベル検出 入力極性設定が反映される。 A 接設定時はカプラ電流 ON で 1 B 接設定時はカプラ電流 OFF で 1 |
| 9 | SFD | '1' = 減速中 | |
| 10 | SFC | '1' = 定速動作中 | |
| 11 | SALM | '1' = SVALM 検出中 | |
| 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 | |

3.4.9 It500_wCmdW() 制御コマンド書込み

| | |
|-------------|--|
| No. | 9 |
| 機能 | デバイスハンドルで指定されたボードの指定軸のコマンドバッファへ制御コマンドデータを書込みます。 |
| 開発環境 | 書式 |
| VC++ | short It500_wCmdW(DWORD hDevID, WORD axis, WORD wCmd); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| wCmd | コマンドデータ |
| VC++ 記述例 | short ret; //関数の戻り値 ret = It500_wCmdW(hDevID, 1, 0x4a); //Y 軸減速停止 |

(1) コマンドデータについて

(1-1) コマンドデータのビット構成

| bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|----|----|----|----|-----------------|--------|--------|--------|---------|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 実行 軸の 指定 (SELx) | | | | コ マ ン ド | | | | | | | |
| | | | | | U B | Z A | Y W | X V | | | | | | | | |

実行軸の指定(SELx)

個々の軸にコマンドを書込む場合はこの4ビットは全て"0"にします。

2~4軸に同一コマンドを書込む場合(同時にスタート、停止など)は、SELxのビットで書込む軸を指定し、軸指定したPCLの任意の軸に書き込みます。

(1-2) スタートコマンド

| No. | 名称 | コマンド (HEX) | 備考 |
|-----|-----------------|---------------|-------------------|
| 1 | 加速スタート | 53 | 加速→FH定速→減速 |
| 2 | FH定速スタート | 51 | |
| 3 | FH定速スタート後減速停止 | 52 | FH定速→減速 |
| 4 | FL定速スタート | 50 | |
| 5 | 残量FL定速スタート | 54 | |
| 6 | 残量FH定速スタート | 55 | |
| 7 | 残量加速スタート | 57 | |
| 8 | 残量FH定速スタート後減速停止 | 56 | |
| 9 | 同時スタート信号(STA)出力 | 06 | 約407nsecのワンショット出力 |

(1-3) 速度変更コマンド

| No. | 名称 | コマンド (HEX) | 備考 |
|-----|-------------|---------------|----|
| 1 | FL 定速瞬時速度変更 | 40 | |
| 2 | FH 定速瞬時速度変更 | 41 | |
| 3 | FL 速度まで減速 | 42 | |
| 4 | FH 速度まで加速 | 43 | |

(1-4) 停止コマンド

| No. | 名称 | コマンド (HEX) | 備考 |
|-----|---------------|---------------|----------------------------|
| 1 | 即停止 | 49 | |
| 2 | 減速停止 | 4A | ベース速度動作中は即停止. その他動作中は減速停止. |
| 3 | 非常停止 | 05 | エラー停止(EMG 入力による停止と同等) |
| 4 | 同時停止信号(STP)出力 | 07 | 約 407nsec のワンショット出力 |

(1-5) コントロールコマンド

| No. | 名称 | コマンド (HEX) | 備考 |
|-----|-------------------|---------------|---|
| 1 | ソフトウェアリセット | 04 | 任意の軸への書き込みで 4 軸全てのレジスタがリセット(POWON 状態)になります. オプションポートはリセットされません. |
| 2 | CTR1 リセット | 20 | |
| 3 | CTR2 リセット | 21 | |
| 4 | CTR3 リセット | 22 | |
| 5 | CTR4 リセット | 23 | |
| 6 | SVCTRCL 信号出力 | 24 | |
| 7 | SVCTRCL 信号リセット | 25 | SVCTRCL 出力幅レベル時の SVCTRCL 出力リセット |
| 8 | 動作用プリレジスタキャンセル | 26 | |
| 9 | CMP5 用プリレジスタキャンセル | 27 | |
| 10 | 動作用プリレジスタシフト | 2B | |
| 11 | CMP5 用プリレジスタシフト | 2C | |
| 12 | カウンタラッチ | 29 | CTR1~CTR4 を一斉にラッチ |
| 13 | PCS コマンド | 28 | PCS 信号の代わりに PCS を実行 |
| 14 | SVON 信号 ON | 18 | |
| 15 | SVON 信号 OFF | 10 | |
| 16 | SVRST 信号 ON | 19 | |
| 17 | SVRST 信号 OFF | 11 | |

3.4.10 It500_rReg() レジスタ読出し

| | |
|-------------|--|
| No. | 10 |
| 機能 | デバイスハンドルで指定されたボードの、指定軸の指定レジスタ内容を読出します。 |
| 開発環境 | 書式 |
| VC++ | short It500_rReg(DWORD hDevID, WORD axis, BYTE byCmd, DWORD* dwData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| byCmd | レジスタ読出コマンド |
| dwData | レジスタ読出データ |
| VC++ 記述例 | short ret; //関数の戻り値 DWORD dwData; //レジスタのデータ ret = It500_rReg(hDevID, 0, 0xc0, &dwData); //X 軸 PRMV 読出 |

3.4.11 It500_wReg() レジスタ書込み

| | |
|-------------|--|
| No. | 11 |
| 機能 | デバイスハンドルで指定されたボードの、指定軸の指定レジスタへ、データを書込みます。 |
| 開発環境 | 書式 |
| VC++ | short It500_wReg(DWORD hDevID, WORD axis, BYTE byCmd, DWORD dwData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| byCmd | レジスタ書込コマンド |
| dwData | レジスタ書込データ |
| VC++ 記述例 | short ret; //関数の戻り値 ret = It500_wReg(hDevID, 0, 0x80, 10000); //X 軸 PRMV=10000 書込 |

(1) レジスタ制御コマンド・内容

レジスタ・プリレジスタ:読出コマンド・書込コマンド一覧

| No | 内容 | レジスタ | | | プリレジスタ | | |
|----|------------------|-------|-----------|----|--------|-----------|----|
| | | 名称 | コマンド(HEX) | | 名称 | コマンド(HEX) | |
| | | | 読出 | 書込 | | 読出 | 書込 |
| 1 | 移動量 | RMV | d0 | 90 | PRMV | c0 | 80 |
| 2 | ベース速度設定 | RFL | d1 | 91 | PRFL | c1 | 81 |
| 3 | 動作速度設定 | RFH | d2 | 92 | PRFH | c2 | 82 |
| 4 | 加速レートを決定するパラメータ | RUR | d3 | 93 | PRUR | c3 | 83 |
| 5 | 減速レートを決定するパラメータ | RDR | d4 | 94 | PRDR | c4 | 84 |
| 6 | 速度倍率設定 | RMG | d5 | 95 | PRMG | c5 | 85 |
| 7 | 減速開始点 | RDP | d6 | 96 | PRDP | c6 | 86 |
| 8 | 動作モード | RMD | d7 | 97 | PRMD | c7 | 87 |
| 9 | 円弧補間中心位置 | RIP | d8 | 98 | PRIP | c8 | 88 |
| 10 | 加速時 S 字区間設定 | RUS | d9 | 99 | PRUS | c9 | 89 |
| 11 | 減速時 S 字区間設定 | RDS | da | 9a | PRDS | ca | 8a |
| 12 | 補助速度設定 | RFA | db | 9b | | | |
| 13 | 環境設定 1 | RENV1 | dc | 9c | | | |
| 14 | 環境設定 2 | RENV2 | dd | 9d | | | |
| 15 | 環境設定 3 | RENV3 | de | 9e | | | |
| 16 | 環境設定 4 | RENV4 | df | 9f | | | |
| 17 | 環境設定 5 | RENV5 | e0 | a0 | | | |
| 18 | 環境設定 6 | RENV6 | e1 | a1 | | | |
| 19 | 環境設定 7 | RENV7 | e2 | a2 | | | |
| 20 | カウンタ 1(指令パルス出力) | RCTR1 | e3 | a3 | | | |
| 21 | カウンタ 2(エンコーダ入力) | RCTR2 | e4 | a4 | | | |
| 22 | カウンタ 3(偏差カウンタ) | RCTR3 | e5 | a5 | | | |
| 23 | カウンタ 4(汎用カウンタ) | RCTR4 | e6 | a6 | | | |
| 24 | コンパレータ 1 用データ | RCMP1 | e7 | a7 | | | |
| 25 | コンパレータ 2 用データ | RCMP2 | e8 | a8 | | | |
| 26 | コンパレータ 3 用データ | RCMP3 | e9 | a9 | | | |
| 27 | コンパレータ 4 用データ | RCMP4 | ea | aa | | | |
| 28 | コンパレータ 5 用データ | RCMP5 | eb | ab | PRCP5 | cb | 8b |
| 29 | イベントマスク設定 | RIRQ | ec | ac | | | |
| 30 | カウンタ 1 ラッチデータ | RLTC1 | ed | | | | |
| 31 | カウンタ 2 ラッチデータ | RLTC2 | ee | | | | |
| 32 | カウンタ 3 ラッチデータ | RLTC3 | ef | | | | |
| 33 | カウンタ 4 ラッチデータ | RLTC4 | f0 | | | | |
| 34 | 拡張ステータス | RSTS | f1 | | | | |
| 35 | エラーステータス | REST | f2 | | | | |
| 36 | イベントステータス | RIST | f3 | | | | |
| 37 | 位置決めカウンタ | RPLS | f4 | | | | |
| 38 | Z 相カウンタ, 指令速度モニタ | RSPD | f5 | | | | |
| 39 | 減速開始点計算値 | RSDC | f6 | | | | |
| 40 | 円弧補間歩進数 | RCI | fc | bc | PRCI | cc | 8c |
| 41 | 円弧補間歩進カウンタ | RCIC | fd | | | | |
| 42 | 補間ステータス | RIPS | ff | | | | |

- *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算出]
- *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)

(2) レジスタ・プリレジスタデータ内容**(2-1) 動作用レジスタ****① RMV(PRMV)移動量レジスタ Read:d0(c0), Write:90(80) …(括弧内はプリレジスタ)**

補間動作, 位置決め動作において移動量を相対位置で設定します。

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

設定範囲 -134,217,728~+134,217,727

書込みはプリレジスタ PRMV に書込む。

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

② RFL(PRFL)ベース速度レジスタ Read:d1(c1), Write:91(81) …(括弧内はプリレジスタ)

ベース速度(加減速送りに於いて加速開始する速度または減速時に減速終了する速度)を設定。

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

設定範囲 1~65,535(0xffff) ただし, 速度(PPS)は速度倍率レジスタ(RMG)の速度倍率によります。

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

書込みはプリレジスタ PRFL に書込む。

*表記のビットは書込み時には無視され, 読出し時には0になる。

③ RFH(PRFH)動作速度レジスタ Read:d2(c2), Write:92(82) …(括弧内はプリレジスタ)

動作速度を設定するレジスタです。

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

設定範囲 1~65,535(0xffff) ただし, 速度(PPS)は速度倍率レジスタ(RMG)の速度倍率によります。

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

書込みはプリレジスタ PRFH に書込む。

*表記のビットは書込み時には無視され, 読出し時には0になる。

④ RUR(PRUR)加速レートレジスタ Read:d3(c3), Write:93(83) …(括弧内はプリレジスタ)

高速動作(加減速動作)の場合の加速特性を設定します。設定値と加速時間の関係は次式の様になります。

直線加速(RMD.b10 (acc 方法) = 0)

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

直線部分のないS字加速(RMD.b10 (acc 方法) = 1 かつ RUS(加速 S 字区間) = 0)

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

直線部分のあるS字加速(RMD.b10 (acc 方法) = 1 かつ RUS(加速 S 字区間) > 0)

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

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

設定範囲 1~65,535(0xffff)

書込みはプリレジスタ PRUR に書込む。

*表記のビットは書込み時には無視され, 読出し時には0になる。

⑤ RDR(PRDR)減速レートレジスタ Read:d4(c4), Write:94(84) …(括弧内はプリレジスタ)

高速動作(加減速動作)の場合の減速特性を設定します。"0"を設定した場合はRURの設定値を設定した時と同じ減速特性になります。設定値と減速時間の関係は次式の様になります。

直線減速(RMD.b10(acc方法)=0)

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

直線部分のないS字減速(RMD.b10(acc方法)=1かつRDS(減速S字区間)=0)

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

直線部分のあるS字減速(RMD.b10(acc方法)=1かつRDS(減速S字区間)>0)

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

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

設定範囲 0~65,535(0xffff)

書込みはプリレジスタ PRDR に書込む。

*表記のビットは書込み時には無視され、読出し時には0になる。

⑥ RMG(PRMG)速度倍率レジスタ Read:d5(c5), Write:95(85) …(括弧内はプリレジスタ)

速度倍率を設定します。

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

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

設定範囲 2~4,095(0xffff)

書込みはプリレジスタ PRMG に書込む。

*表記のビットは書込み時には無視され、読出し時には0になる。

⑦ RDP(PRDP)減速開始点レジスタ Read:d6(c6), Write:96(86) …(括弧内はプリレジスタ)

加減速動作を伴う位置決め時の減速開始点を設定するレジスタです。

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

手動減速開始点計算(RMD.b13='1')の場合は終点からの減速移動量(パルス数)を設定します。

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

書込みはプリレジスタ PRDP に書込む。

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

RMD.b13='0'自動設定値に対するオフセット量を示し、正数は早めに減速する値、負数は遅めの減速。

オフセットが必要ない場合は"0"に設定します。#表記のビットはビット23と同じ。

設定範囲は-8,388,608(0x800000)~8,388,607(0x7FFFFFF)の範囲で設定します。

RMD.b13='1'手動減速開始点計算設定。移動残量が設定値以下になると減速開始をする。

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

減速開始点の最適値は次式の様になります。

直線減速(RMD.b10(acc方法)=0)

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

直線部分のないS字減速(RMD.b10(acc方法)=1かつRDS(減速S字区間)=0)

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

直線部分のあるS字減速(RMD.b10(acc方法)=1かつRDS(減速S字区間)>0)

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

(位置決めカウンタ値) ≤ (RDP設定値) のタイミングで減速を開始します。

⑧ RMD(PRMD)動作モードレジスタ Read:d7(c7), Write:97(87) …(括弧内はプリレジスタ)

| | | | | | | | | | |
|------|-------|-------|----------|-------|-------|--------|---------|---------|--------|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 | | |
| 合成速度 | PCS有効 | 減速開始点 | CTR 1DIS | acc方法 | NPS E | DLSE | M O D | | |
| 31 | 28 | 27 | 24 | 23 | 20 | 19 | 16 | | |
| 0 | 0 | MSD C | MPIE | FH補正 | 他軸停止 | 同時ストップ | 停止指定軸設定 | 条件付スタート | SEQ No |

書き込みはプリレジスタ PRMD に書込む。

動作モード(MOD)

| No | 動作分類 | MOD(HEX) | 動作モード | 記 事 |
|----|------------|----------|--------------------|-------------|
| 1 | 連続動作 | 00 | +方向連続送り | |
| | | 08 | -方向連続送り | |
| 2 | 位置決め動作 | 41 | 位置決め動作(相対位置指定) | |
| | | 42 | 位置決め動作(指令位置指定) | ドライバ関数のみ設定可 |
| | | 44 | 指令位置0点復帰動作 | |
| | | 45 | 機械位置0点復帰動作 | |
| | | 46 | +方向1パルス動作 | |
| | | 4e | -方向1パルス動作 | |
| | | 47 | タイマ動作 | |
| 3 | 直線補間動作 | 60 | 直線補間連続送り | |
| | | 61 | 直線補間 | |
| | | 62 | 2個のPCL間の直線補間連続送り | |
| | | 63 | 2個のPCL間の直線補間 | |
| 4 | 円弧補間動作 | 64 | CW方向円弧補間 | |
| | | 65 | CCW方向円弧補間 | |
| | | 66 | U(B)軸同期CW方向円弧補間 | |
| | | 67 | U(B)軸同期CCW方向円弧補間 | |
| 5 | 原点復帰動作 | 10 | +方向原点復帰動作 | |
| | | 18 | -方向原点復帰動作 | |
| | | 12 | +方向原点拔出 | |
| | | 1a | -方向原点拔出 | |
| | | 15 | +方向原点サーチ | |
| | | 1d | -方向原点サーチ | |
| 6 | ELS, SLS動作 | 20 | +ELSまたは+SLS位置まで動作 | |
| | | 28 | -ELSまたは-SLS位置まで動作 | |
| | | 22 | +ELSまたは+SLS拔出し動作 | |
| | | 2a | -ELSまたは-SLS拔出し動作 | |
| 7 | Z相移動 | 24 | +方向にZ相カウント分動作 | |
| | | 2c | -方向にZ相カウント分動作 | |
| 8 | パルサ動作 | 01 | パルサ入力による連続動作 | 手動パルサ送り |
| | | 51 | パルサ入力による位置決め動作 | |
| | | 54 | パルサ入力による指令位置0点復帰動作 | |
| | | 55 | パルサ入力による機械位置0点復帰動作 | |
| 9 | JOG送り動作 | 02 | ±DR入力による連続動作 | JOG送り |
| | | 56 | ±DR入力による位置決め動作 | |

| bit | 名称 | 説明 |
|-------|---------------------|---|
| 7-0 | MOD | 動作モード(前ページ参照) |
| 8 | DLS 有効 | 0:DLS 無効, 1:DLS 有効 |
| 9 | INPOS 有効 | 0:INPOS 無効, 1:INPOS 有効 |
| 10 | 加減速方式 | 0:直線加減速, 1:S 字加減速 |
| 11 | CTR1 カウント禁止 | 1:指令位置カウンタ(CTR1) カウント禁止 |
| 12 | 動作完了タイミング | 0:完了タイミングをパルスの周期完了とします。 |
| 13 | 減速開始点計算方法 | 0:減速開始点自動計算, 1:減速開始点手動計算 |
| 14 | PCS 有効 | 0:PCS 無効, 1:PCS 有効 |
| 15 | 補間合成速度 | 0:補間時合成速度一定制御 OFF, 1:補間時合成速度一定制御 ON |
| 17,16 | SEQ No | MSTS.b7,6 に現在実行中の SEQ No が反映されます。 |
| 19-18 | スタート条件 | 00: スタートコマンド書込み後即スタート。 01: STA 入力によるスタート。 10: 他軸の条件一致によるスタート。 11: 指定軸の停止によるスタート。 |
| 23-20 | 軸停止によるスタート時の軸指定 | 指定軸の停止によるスタート(RMD.b19,18=11)の場合、軸を指定します。 bit20: X(V,X1,X2,X3,X4)軸の停止 bit21: Y(W,Y1,Y2,Y3,Y4)軸の停止 bit22: Z(A,Z1,Z2,Z3,Z4)軸の停止 bit23: U(B,U1,U2,U3,U4)軸の停止 <例> 0001: X(V,X1,X2,X3,X4)軸の停止で次動作スタート 1001: X(V,X1,X2,X3,X4)軸の停止かつ U(B,U1,U2,U3,U4)軸の停止で次動作スタート |
| 24 | STP 有効 | 1:STP 入力により停止します。 |
| 25 | 異常時 STP 自動出力 | 1:異常停止時に STP(同時停止信号)を自動出力します。 |
| 26 | FH 補正 | 0:ON, 1:OFF |
| 27 | MPIE (円弧自動終点引込み) | 1:円弧自動終点引込み |
| 28 | 予約 | |
| 29 | MSDC | 1:減速開始点自動計算時、減速開始点=加速パルス数とする。 0:補間動作を合成速度一定制御 ON 時のみ減速開始点=加速パルス数とする。 (位置決め動作の減速開始点計算は内部自動演算) |
| 31,30 | 予約 | |

⑨ RIP(PRI)円弧補間中心位置レジスタ Read:d8(c8), Write:98(88) …(括弧内はプリレジスタ)

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

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

RIP は、RMD の MOD(b7-0)が次の動作モードの時有効。

円弧補間(MOD=0x64, 0x65), 2個以上の PCL 間の直線連続送り(MOD=0x62),

2個以上の PCL 間の直線補間(MOD=0x63)

円弧補間の場合は円弧中心位置を相対値で設定する。

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

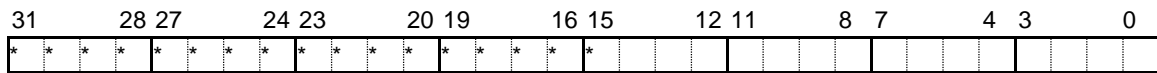
設定範囲 -134,217,727~+134,217,727

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

⑩ RUS(PRUS)加速 S 字区間レジスタ Read:d9(c9), Write:99(89) …(括弧内はプリレジスタ)

⑪ RDS(PRDS)減速 S 字区間レジスタ Read:da(ca), Write:9a(8a) …(括弧内はプリレジスタ)

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



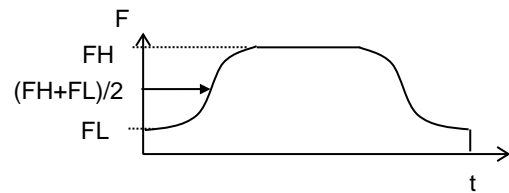
*表記のビットは、書き込み時には無視され、読出し時には0になる.

設定範囲 1~32,767

0を設定すると

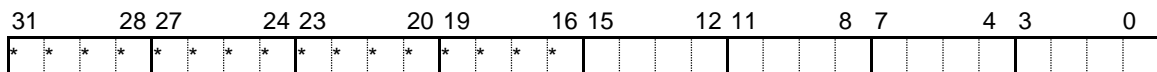
$$\frac{RFH - RFL}{2} \text{ の値が自動的に設定される.}$$

すなわち、RUS=RDS=0 の場合には完全に S 字となる.



⑫ RFA 補助速度レジスタ Read:db, Write:9b

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



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

*表記のビットは、書き込み時には無視され、読出し時には0になる.

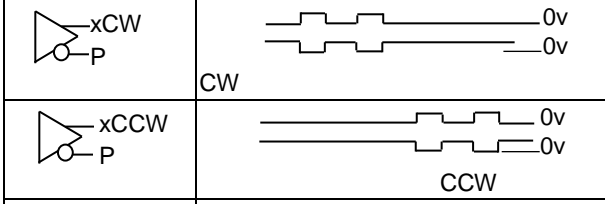
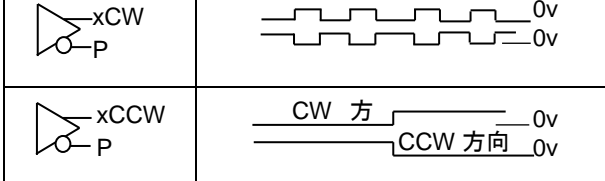
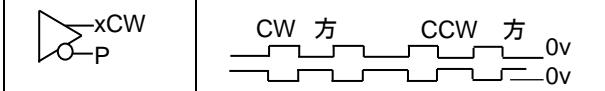
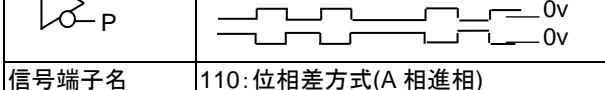
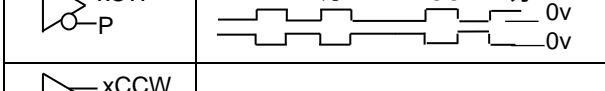
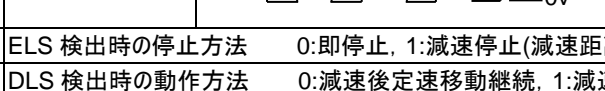
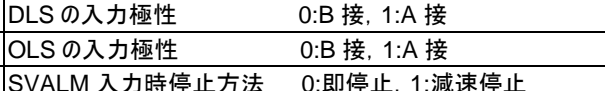
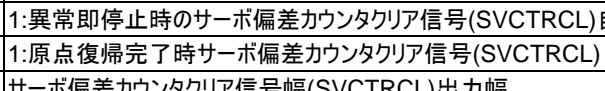
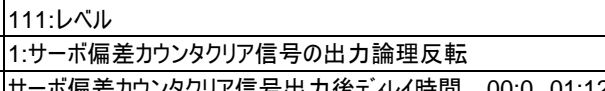

$$\text{速度} F = RFA \times \frac{300}{RMG + 1} \text{ [PPS]}$$

(2-2) 環境設定レジスタ

① RENV1 環境設定レジスタ 1 Read:dc, Write:9c

指令出力, マシンインターフェイス信号, サーボインターフェイス信号など入出力極性, パルス幅, 動作などを設定します.

| | | | | | | | | | | | | | | | |
|--------|------|------|------|--------|--------|------|------|-----|------|------|------|------|------|------|------|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 | | | | | | | | |
| ER CTL | EPW2 | EPW1 | EPW0 | ER CTO | ER CTE | ALML | ALMM | OLL | SDL | SDLT | SDM | ELM | PMD2 | PMD1 | PMD0 |
| 31 | 28 | 27 | 24 | 23 | 20 | 19 | 16 | | | | | | | | |
| 0 | 0 | INTM | DTMF | DRF | FLTR | DR | PCS | LTC | INPS | CLR1 | CLR0 | CSTP | CSTA | ETW1 | ETW0 |

| bit | 名称 | 説明 |
|-------|-----------|---|
| 2-0 | PMD2-0 | 信号端子名 100:個別パルス方式 指令出力  |
| | | 信号端子名 010:共通パルス方式 指令出力  |
| | | 信号端子名 パルス列  |
| | | 信号端子名 方向出力  |
| | | 信号端子名 101:位相差方式(B相進相) 指令出力  |
| | | 信号端子名 A相出力  |
| | | 信号端子名 B相出力  |
| | | 信号端子名 110:位相差方式(A相進相) 指令出力  |
| | | 信号端子名 A相出力  |
| | | 信号端子名 B相出力  |
| 3 | ELM | ELS 検出時の停止方法 0:即停止, 1:減速停止(減速距離に注意) |
| 4 | SDM | DLS 検出時の動作方法 0:減速後定速移動継続, 1:減速停止 |
| 5 | SDLT | DLS のラッチ機能 0:使用しない, 1:使用する |
| 6 | SDL | DLS の入力極性 0:B 接, 1:A 接 |
| 7 | OLL | OLS の入力極性 0:B 接, 1:A 接 |
| 8 | SVALMM | SVALM 入力時停止方法 0:即停止, 1:減速停止 |
| 9 | SVALML | SVALM 信号入力極性 0:B 接, 1:A 接 |
| 10 | ERCTE | 1:異常即停止時のサーボ偏差カウンタクリア信号(SVCTRCL)自動出力する |
| 11 | ERCTO | 1:原点復帰完了時サーボ偏差カウンタクリア信号(SVCTRCL)自動出力する |
| 14-12 | EPW2-EPW0 | サーボ偏差カウンタクリア信号幅(SVCTRCL)出力幅 000:12us, 001:0.1ms, 010:0.4ms, 011:1.6ms, 100:13ms, 101:52ms, 110:104ms, 111:レベル |
| 15 | SVCTRCL | 1:サーボ偏差カウンタクリア信号の出力論理反転 |
| 17-16 | ETW1-ETW0 | サーボ偏差カウンタクリア信号出力後デレイ時間 00:0, 01:12us, 10:1.6ms, 11:104ms |

| | | | | | | | | | | | | | | | |
|-----------|------|------|------|-----------|-----------|------|------|-----|------|------|------|------|------|------|------|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 | | | | | | | | |
| ER CTL | EPW2 | EPW1 | EPW0 | ER CTO | ER CTE | ALML | ALMM | OLL | SDL | SDLT | SDM | ELM | PMD2 | PMD1 | PMD0 |
| 31 | 28 | 27 | 24 | 23 | 20 | 19 | 16 | | | | | | | | |
| 0 | 0 | INTM | DTMF | DRF | FLTR | DR | PCS | LTC | INPS | CLR1 | CLR0 | CSTP | CSTA | ETW1 | ETW0 |

| bit | 名称 | 説明 |
|-----|-------|----------------------------------|
| 18 | CSTA | STA 入力仕様 0:レベル, 1:エッジ |
| 19 | CSTP | STP 入力時停止方法 0:即停止, 1:減速停止(注意 1) |
| 22 | INPOS | インポジション信号入力極性 0:B 接, 1:A 接 |
| 24 | PCS | PCS入力信号極性 0:B 接, 1:A 接 |
| 25 | DR | DR入力信号極性 0:B 接, 1:A 接 |
| 28 | DTMF | 方向変化タイマ ON/OFF 0:ON, 1:OFF |
| 29 | INTM | 0:この軸からの割込信号出力, 1:この軸からの割込信号出力禁止 |



注意

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

② RENV2 環境設定レジスタ 2 Read :dd, Write:9d

エンコーダ A/B 相カウント方法/方向, Z 相極性, パルス A/B 相カウント方法/方向を設定します。

| | | | | | | | | | | | | | | | |
|------|------|------|------|----|------|-----------|-----|------|------|---|---|---|---|---|---|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 | | | | | | | | |
| 1 | 1 | 1 | 1 | x | 1 | 0 | 1 | 0 | 1 | x | x | 0 | 1 | 0 | 1 |
| 31 | 28 | 27 | 24 | 23 | 20 | 19 | 16 | | | | | | | | |
| POFF | EOFF | SMAX | PMSK | 0 | PDIR | PULS R | EZL | EDIR | ENCM | 0 | 0 | 0 | 0 | | |

| bit | 名称 | 説明 |
|-------|---------------|---|
| 19-0 | ボード仕様 (固定) | 汎用入出力ポートの仕様設定. b19~b0 は必ず次のように初期設定する. HPCI-CPD574N, HPCI-CPD578N, HPCle-CPD674N, HPCle-CPD678N .. 0x0fd55, その他 .. 0x0f555 |
| 21-20 | ENCM | エンコーダ逡倍設定 00:1 逡倍, 01:2 逡倍, 10:4 逡倍, 11:up/down パルス |
| 22 | EDIR | 1:エンコーダ A 相, B 相カウント方向逆転 |
| 23 | EZL | 1:エンコーダ Z 相入力極性反転 |
| 25-24 | PULSR | パルス逡倍設定 00:1 逡倍, 01:2 逡倍, 10:4 逡倍, 11:up/down パルス |
| 26 | PDIR | 1:パルス A 相, B 相カウント方向逆転 |
| 28 | PMSK | 1:指令パルス出力禁止(CTR1 は動作します) |
| 29 | SMAX | 1:自軸を含めた指定軸停止によるスタート有効(RMD.b23-20) |
| 30 | EOFF | 1:エンコーダ入力をマスク |
| 31 | POFF | 1:パルス入力をマスク |

③ RENV3 環境設定レジスタ 3 Read :de, Write:9e

原点復帰方法指定, 原点復帰 Z 相カウント設定, CTR 動作方法の設定などを行います.

| | | | | | | | |
|-------------|-------------|-------------|---------|-------------|-------------|-------------|-------------|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| 0 | CTR4 G | CTR 4IP | CTR 3IP | CTR 2IP | Z 相 CT | ORG mode | |
| 31 | 28 | 27 | 24 | 23 | 20 | 19 | 16 |
| CTR4 DIS | CTR3 DIS | CTR2 DIS | 0 | CTR4 BS | CTR3 BS | CTR2 BS | CTR1 BS |
| | | | | CTR4 ORG | CTR3 ORG | CTR2 ORG | CTR1 ORG |
| | | | | 0 | 0 | 0 | 0 |

| bit | 名称 | 説明 |
|-------|----------|--|
| 3-0 | ORG mode | 0000:OLSon で完了 0001:OLSon 停止後 RFA 反転 OLSoff まで動作後 RFA 再反転 OLSon で完了 0010:OLSon 減速+Z 相カウントで完了 0011:OLSon 後の Z 相カウントで完了 0100:OLSon 停止後 RFA 反転+ Z 相カウントで完了 0101:OLSon 停止後加速反転+ Z 相カウントで完了 0110:ELSon 停止後 RFA 反転 ELSoff で完了 0111:ELSon 停止後 RFA 反転+ Z 相カウントで完了 1000:ELSon 停止後加速反転+ Z 相カウントで完了 1001:ORGmode0 の動作後 CTR2 零点復帰 1010:ORGmode3 の動作後 CTR2 零点復帰 1011:ORGmode5 の動作後 CTR2 零点復帰 1100:ORGmode8 の動作後 CTR2 零点復帰 |
| 7-4 | Z 相 CT | 原点復帰時のエンコーダZ相カウント値:0000(1回)~1111(16回) |
| 9-8 | CTR2IP | CTR2 入力元選択 00:ENC 入力, 01:指令パルス, 10:パルサ入力 |
| 11-10 | CTR3IP | CTR3 入力元選択 00:指令パルスと ENC 入力の偏差, 01:指令パルスとパルサ入力の偏差 |
| 13-12 | CTR4IP | CTR4 入力元選択 00:指令パルス, 01:ENC 入力, 10:パルサ入力, 11: 9.8304MHz CLK カウント |
| 14 | CTR4G | 1:動作中のみ CTR4 カウント |
| 23-20 | CTRxOR G | 1:原点復帰完了時 CTRx クリア |
| 27-24 | CTRxB S | 1:バックラッシュ動作, スリップ動作中も CTRx カウント |
| 31-29 | CTRxD IS | 1:CTRx カウントしない |

④ RENV4 環境設定レジスタ 4 Read :df, Write:9f

コンパレータ(CMP)は 5 組あります。CMP1~4 を設定します。(CMP5 は RENV5)

| | | | | | | | | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C2RM | CMP 2M | | CMP2 D | | CMP 2C | C1RM | CMP 1M | | CMP1 D | | CMP 1C | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CMP 4M | | CMP 4D | | CMP 4C | 0 | CMP 3M | | CMP3 D | | CMP 3C | | | | | |

| bit | 名称 | 説明 |
|-------|-------|---|
| 1-0 | CMP1C | CMP1 比較カウンタ 00:CTR1, 01:CTR2, 10:CTR3, 11:CTR4 |
| 4-2 | CMP1D | CMP1 条件 001:RCMP1=比較 CTR(UP/DOWN), 010: RCMP1=比較 CTR(UP 時) 011:RCMP1=比較 CTR(DOWN 時), 100: RCMP1>比較 CTR 101: RCMP1<比較 CTR, 110:+SLS(RCMP1<CTR1), その他:比較せず |
| 6-5 | CMP1M | CMP1 条件成立時の処理 00:処理なし, 01:即停止, 10:減速停止, 11:動作プリレジスタのデータに変更 |
| | C1RM | 1:RCMP1 と対で CTR1 を回転軸カウンタとして使用 |
| 9-8 | CMP2C | CMP2 比較カウンタ 00:CTR1, 01:CTR2, 10:CTR3, 11:CTR4 |
| 12-10 | CMP2D | CMP2 条件 001:RCMP2=比較 CTR(UP/DOWN), 010: RCMP2=比較 CTR(UP 時) 011:RCMP2=比較 CTR(DOWN 時), 100: RCMP2>比較 CTR 101: RCMP2<比較 CTR, 110:-SLS(RCMP2>CTR1), その他:比較せず |
| 14-13 | CMP2M | CMP2 条件成立時の処理 00:処理なし, 01:即停止, 10:減速停止, 11:動作プリレジスタのデータに変更 |
| 15 | C2RM | 1: RCMP2 と対で CTR2 を回転軸カウンタとして使用 |
| 17-16 | CMP3C | CMP3 比較カウンタ 00:CTR1, 01:CTR2, 10:CTR3, 11:CTR4 |
| 20-18 | CMP3D | CMP3 条件 001:RCMP3=比較 CTR(UP/DOWN), 010: RCMP3=比較 CTR(UP 時) 011:RCMP3=比較 CTR(DOWN 時), 100: RCMP3>比較 CTR 101: RCMP3<比較 CTR, 110:設定禁止, その他:比較せず |
| 22-21 | CMP3M | CMP3 条件成立時の処理 00:処理なし, 01:即停止, 10:減速停止, 11:動作プリレジスタのデータに変更 |
| 25-24 | CMP4C | CMP4 比較カウンタ 00:CTR1, 01:CTR2, 10:CTR3, 11:CTR4 |
| 29-26 | CMP3D | CMP4 条件 0001:RCMP4=比較 CTR(UP/DOWN) 1000:定ピッチ信号出力用として使用(UP/DOWN) 0010:RCMP4=比較 CTR(UP 時) 1001:定ピッチ信号出力用として使用(UP 時) 0011:RCMP4=比較 CTR(DOWN 時) 1010:定ピッチ信号出力用として使用(DOWN 時) 0100: RCMP4>比較 CTR その他:比較せず 0101: RCMP4<比較 CTR |
| 31-30 | CMP4M | CMP4 条件成立時の処理 00:処理なし, 01:即停止, 10:減速停止, 11:動作プリレジスタのデータに変更 |



注意

比較対象カウンタとして CTR3 を選択した場合はカウント値の絶対値(0~32,767)と比較データとの比較となります。
+SLS, -SLS に設定する場合は比較対象カウンタに CTR1 を選択します。

⑤ RENV5 環境設定レジスタ 5 Read :e0, Write:a0

CMP5 の設定, アイドリングパルス数, カウンタのラッチ条件, 他軸スタート条件などの設定を行います.

| | | | | | | | | | | | | | | | |
|----|--------|----|-----|------|------|------|------|--------|--------|------------|------------|--------|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | CTLCHF | CT | LCH | PDSM | IDLP | | | CMP 5M | CMP5 D | | | CMP5 C | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | CU4L | CU3L | CU2L | CU1L | 0 | 0 | TRIGRINsel | TRIGROTsel | | | | |

| bit | 名称 | 説明 |
|-------|------------|--|
| 2-0 | CMP5C | CMP5 比較対象 000:CTR1, 001:CTR2, 010:CTR3, 011:CTR4, 100:PCTR, 101:現在速度 |
| 5-3 | CMP5D | CMP5 条件 001:RCMP5=比較 CTR(UP/DOWN), 010: RCMP5=比較 CTR(UP 時) 011:RCMP5=比較 CTR(DOWN 時), 100: RCMP5>比較 CTR 101: RCMP5<比較 CTR, その他:比較せず |
| 7-6 | CMP5M | CMP5 条件成立時の処理 00:処理なし, 01:即停止, 10:減速停止, 11:動作プリレジスタのデータに変更 |
| 10-8 | IDLP | アイドリングパルス数(0~7) |
| 11 | PDSM | 1: PA/PB 及び±DR による連続動作時, 動作方向の ELS によるエラー割り込みが発生します. 再スタートにはスタートコマンドが必要です. (CPDxxxN のみ) |
| 13-12 | CTLCH | CTR ラッチ条件 01:OLSoft→on, 10:CMP4 条件成立時, 11:CMP5 条件成立時 |
| 14 | CTLCHF | 1:CTR3 の代わりに速度レジスタをラッチ |
| 19-16 | TRIGROTsel | 他軸をスタートさせるトリガ条件設定 0001:CMP1 条件成立時 1000:加速開始時 0010:CMP2 条件成立時 1001:加速終了時 0011:CMP3 条件成立時 1010:減速開始時 0100:CMP4 条件成立時 1011:減速終了時 0101:CMP5 条件成立時 その他:トリガ OFF |
| 21-20 | TRIGRINsel | 自軸がスタートするためのトリガ軸の選択 00:X(V)軸, 01:Y(W)軸, 10:Z(A)軸, 11:U(B)軸 |
| 27-24 | CUxL | 1:CTR _x のラッチ後 CTR _x クリア |

⑥ RENV6 環境設定レジスタ 6 Read :e1, Write:a1

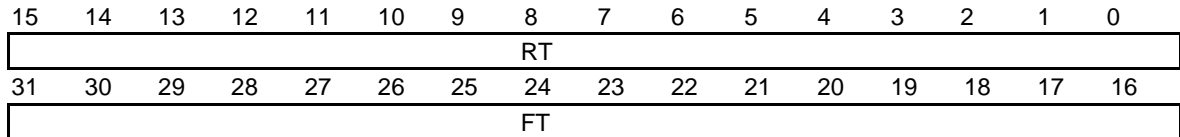
バックラッシュ/スリップ補正量, パルサ分周/逡倍比を設定します. .

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

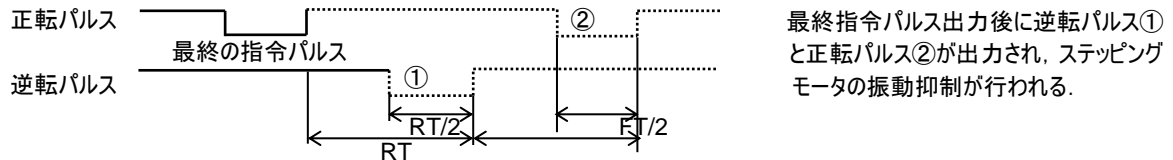
| bit | 名称 | 説明 |
|-------|---------|------------------------------------|
| 11-0 | BLSH | バックラッシュ/スリップ補正量 |
| 13-12 | BLSHsel | 00:補正 off, 01:バックラッシュ補正, 10:スリップ補正 |
| 26-16 | PD | パルサ分周比 (設定値) / 2,048 に分周 0:分周 OFF |
| 31-27 | PMG | パルサ逡倍比 (設定値+1)倍に逡倍 |

⑦ RENV7 環境設定レジスタ 7 Read :e2, Write:a2

ステッピングモータの停止時特性を改善する「制振機能」を設定します。



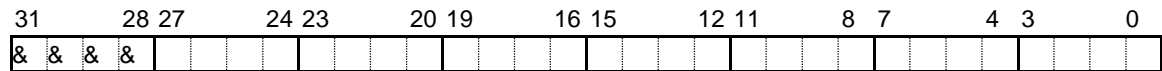
| bit | 名称 | 説明 |
|-------|----|--|
| 15-0 | RT | 最終出力パルス後の逆転パルス出カタイミンング RT×1.6us(1.6us~104ms) |
| 31-16 | FT | 上記逆転パルス出力後の正転パルス出カタイミンング FT×1.6us(1.6us~104ms) |



(2-3) カウンタレジスタ

① RCTR1 指令位置カウンタレジスタ(Read Only) Read:e3, Write:a3

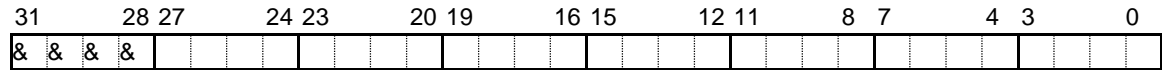
指令パルス用カウンタレジスタです。カウンタはアップダウン・カウンタです。
特定のタイミングでカウンタを読み出す場合は "ラッチ" を使用し、RLTC1 から読み出します。



設定範囲 -134,217,728~+134,217,727
&表記のビットは、書込み時には無視され、読み出し時には空欄表示の最上位ビットと同一になります。(符号拡張)

② RCTR2 指令位置カウンタレジスタ(Read Only) Read:e4, Write:a4

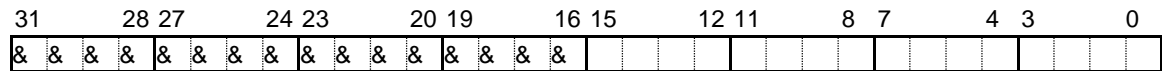
機械位置用カウンタレジスタです。カウンタはアップダウン・カウンタです。
エンコーダ信号、パルサ信号および指令パルスが選択できます。(RENV3.b9-b8)
特定のタイミングでカウンタを読み出す場合は "ラッチ" を使用し、RLTC2 から読み出します。



設定範囲 -134,217,728~+134,217,727
&表記のビットは、書込み時には無視され、読み出し時には空欄表示の最上位ビットと同一になります。(符号拡張)

③ RCTR3 脱調カウンタレジスタ(Read Only) Read:e5, Write:a5

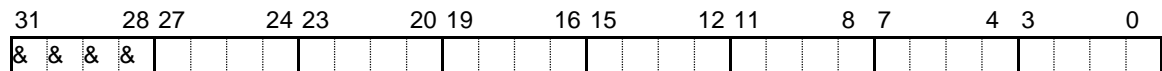
脱調検出用 偏差カウンタレジスタです。2つの信号の偏差(差分)をカウントします。
指令パルスとエンコーダ信号、指令パルスとパルサ信号の何れかを選択できます。(RENV3.b11-b10)
特定のタイミングでカウンタを読み出す場合は "ラッチ" を使用し、RLTC2 から読み出します。



設定範囲 -32,768~+32,767
&表記のビットは、書込み時には無視され、読み出し時には空欄表示の最上位ビットと同一になります。(符号拡張)

④ RCTR4 汎用カウンタレジスタ(Read Only) Read:e6, Write:a6

機械位置用カウンタレジスタです。カウンタはアップダウン・カウンタです。
指令パルス、エンコーダ信号、パルサ信号および 9.8304MHz クロック信号が選択入力できます(RENV3.b13-b12)
特定のタイミングでカウンタを読み出す場合は "ラッチ" を使用し、RLTC2 から読み出します。



設定範囲 -134,217,728~+134,217,727
&表記のビットは、書込み時には無視され、読み出し時には空欄表示の最上位ビットと同一になります。(符号拡張)

(2-4) コンパレータ比較レジスタ

- ① **RCMP1** コンパレータ 1 比較レジスタ Read:e7, Write:a7
 ② **RCMP2** コンパレータ 2 比較レジスタ Read:e8, Write:a8
 ③ **RCMP3** コンパレータ 3 比較レジスタ Read:e9, Write:a9
 ④ **RCMP4** コンパレータ 4 比較レジスタ Read:ea, Write:aa

コンパレータ 1~4 用比較データを設定します。

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

設定範囲 -134,217,728~+134,217,727

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

⑤ **RCMP5(PRCP5)** コンパレータ 5 比較レジスタ Read:eb(cb), Write:ab(8b) …(括弧内はプリレジスタ)

コンパレータ 5 用比較データを設定します。プリレジスタがあります。

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

設定範囲 -134,217,728~+134,217,727

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

⑥ イベントマスクレジスタ

⑦ **RIRQ** イベントマスクレジスタ(Read Only) Read:ec, Write:ac

RIST イベントステータスレジスタに対するイベント報告を設定するレジスタです。

各ビットを 1 にするとイベントステータス(RIST)に反映されます。

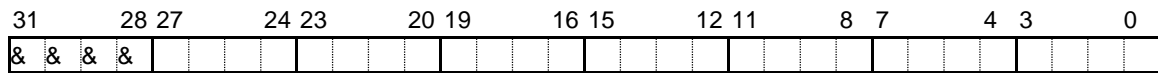
| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IROL | IRLT | IRCL | IRC5 | IRC4 | IRC3 | IRC2 | IRC1 | IRDE | IRDS | IRUE | IRUS | IRND | IRNM | IRN | IREN |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IRSA | IRDR | IRSD |

| bit | 名称 | 内容 |
|-------|------|---------------------|
| 0 | IREN | 動作完了報告 |
| 1 | IRN | 次動作継続スタート報告 |
| 2 | IRNM | 動作用プリレジスタフル→空き報告 |
| 3 | IRND | CMP5 用プリレジ書き込み可能報告 |
| 4 | IRUS | 加速開始報告 |
| 5 | IRUE | 加速終了報告 |
| 6 | IRDS | 減速開始報告 |
| 7 | IRDE | 減速終了報告 |
| 12-8 | IRCn | CMPn 比較条件成立報告 |
| 13 | IRCL | CLR 入力によるカウンタクリア報告 |
| 14 | IRLT | LTCH 入力によるカウンタラッチ報告 |
| 15 | IROL | OLS 信号入力時にカウンタラッチ報告 |
| 16 | IRSD | DLS 信号 OFF→ON |
| 17 | IRDR | ±DR 信号 OFF→ON |
| 18 | IRSA | STA 信号 OFF→ON |
| 19-31 | 未定義 | 常に'0'を設定してください |

(2-5) カウンタラッチレジスタ

① RLTC1 カウンタラッチレジスタ(Read Only) Read:ed

RCTR1のカウンタラッチレジスタです。

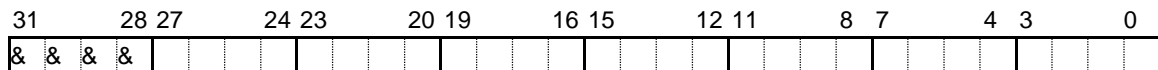


データ範囲 -134,217,728~+134,217,727

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

② RLTC2 カウンタラッチレジスタ(Read Only) Read:ee

RCTR2のカウンタラッチレジスタです。

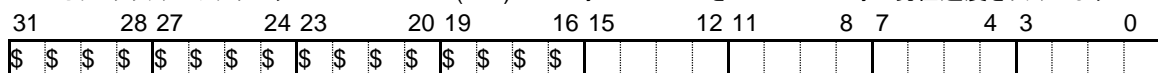


データ範囲 -134,217,728~+134,217,727

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

③ RLTC3 カウンタラッチレジスタ(Read Only) Read:0xef

RCTR3のカウンタラッチレジスタです。RENV5.LTFD(b14)=0の時はRCTR3をLTFD=1の時は現在速度をラッチします。



データ範囲

LTFD=0の時は -32,768 ~ +32,767

LTFD=1の時は 0 ~ 65535

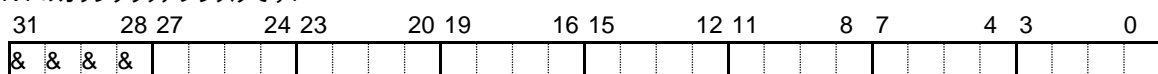
\$表示のビットは、

RENV5.LTFD(b14)=0の時は空欄表示の最上位ビットと同一になります(符号拡張)

RENV5.LTFD(b14)=1の時は "0" になります

④ RLTC4 カウンタラッチレジスタ(Read Only) Read:0xf0

RCTR4のカウンタラッチレジスタです。



データ範囲 -134,217,728~+134,217,727

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

(2-6) ステータスレジスタ

① RSTS 拡張ステータスレジスタ(Read Only) Read:f1

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|-----|------|------|------|------|------|------|------|------|------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SDIN | SLTC | SCLR | SDRM | SDRP | SEZ | SERC | SPCS | SEMG | SSTA | SSTP | SDIR | CND3 | CND2 | CND1 | CND0 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PFM1 | PFM0 | PFC1 | PFC0 | 0 | SINP |

| bit | 名称 | 内 容 |
|-------|--------|--|
| 0 | CND0 | 動作状態を表す |
| 1 | CND1 | 0000: 停止中 |
| 2 | CND2 | 0001: DR 入力待ち |
| 3 | CND3 | 0010: STA 入力待ち 0011: スタートコマンドが発行されていて、軸スタート条件成立待ち状態。 0100: 指定軸の停止によるスタート待ち 0101: サーボ偏差カウンタ・クリアタイム on 中 0110: 指令方向信号切替中(共通パルス指令時方向切替時間 0.2ms) 0111: バックラッシュ補正中(バックラッシュ補正中の時間は補正パルス量と補助速度による) 1000: パルス入力待ち 1001: FA(補助速度)定速で動作中 1010: FL 定速で動作中 1011: 加速中 1100: FH 定速で動作中 1101: 減速中 1110: INPOS 信号待ち 1111: その他 |
| 4 | SDIR | 動作方向(0: +方向, 1: -方向) |
| 5 | SSTA | '1'= 同時スタート信号 on 状態 |
| 6 | SSTP | '1'= 同時停止信号 on 状態 |
| 7 | SEMG | '1'= EMG 信号入力中 |
| 8 | SPCS | '1'= PCS 位置決めスタート信号入力中 |
| 9 | SERC | '1'= サーボ偏差カウンタクリア信号出力中 |
| 10 | SEZ | '1'= エンコーダ Z 相信号入力中 |
| 11 | SDRP | '1'= +DR 信号入力中 |
| 12 | SDRM | '1'= -DR 信号入力中 |
| 13 | SCLR | '1'= CLR 信号入力中 |
| 14 | SLTC | '1'= LTCH 信号入力中 |
| 15 | SDIN | '1'= DLS 信号入力中 |
| 16 | SINP | '1'= INPOS 信号入力中 |
| 17 | 未定義 | 常に'0' |
| 19,18 | PFC1,0 | RCMP5 用プリレジスタの使用状態 00:未確定, 01:レジスタ確定, 10:1st プリレジスタ確定, 11:2nd プリレジスタ確定(プリレジスタフル) |
| 21,20 | PFM1,0 | 動作用プリレジスタ(RCMP5 用以外)の使用状態 00:未確定, 01:レジスタ確定, 10:1st プリレジスタ確定, 11:2nd プリレジスタ確定(プリレジスタフル) |
| 31-22 | 未定義 | 常に'0' |

② REST エラーステータスレジスタ(Read Only) Read:f2

エラー報告の発生要因を確認できます。

エラー報告が発生した時に対応するビットが"1"になります。またこの時 MST.SERR(b4)=1 となります。

このレジスタは、読み出しによりリセットされ、MST.SERR(b4)=0 となります。エラーステータスはマスクすることは出来ません。

| | | | | | | | | | | | | | | | |
|------|------|------|------|----|------|------|------|------|------|------|------|------|------|------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ESAO | ESPO | ESIP | ESDT | 0 | ESSD | ESEM | ESSP | ESAL | ESML | ESPL | ESC5 | ESC4 | ESC3 | ESC2 | ESC1 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ESPE | ESEE |

| bit | 名称 | 内容 | 備考 |
|-------|------------|---|--|
| 0 | ESC1 | 1 = CMP1 条件成立で停止 (+SLS) | |
| 1 | ESC2 | 1 = CMP2 条件成立で停止 (-SLS) | |
| 2 | ESC3 | 1 = CMP3 条件成立で停止 | |
| 3 | ESC4 | 1 = CMP4 条件成立で停止 | |
| 4 | ESC5 | 1 = CMP5 条件成立で停止 | |
| 5 | ESPL | 1 = +ELS による停止 | 停止中に+ELS が ON しても 1 になりません。 |
| 6 | ESML | 1 = -ELS による停止 | 停止中に-ELS が ON しても 1 になりません。 |
| 7 | ESAL | 1 = サーボアラームによる停止 | 停止中に SVALM が ON しても 1 になりません。 |
| 8 | ESSP | 1 = STP 入力 ON による停止 (または同時ストップコマンド)による停止 | 次動作使用時はプリレジスタキャンセルコマンド (26h)を書き込んでください。 |
| 9 | ESEM | 1 = EMG 入力 ON による停止 | 非常停止コマンドによる停止または非常停止機能 を使用している場合 |
| 10 | ESSD | 1 = DLS 検出による減速停止 | 停止中に DLS が ON しても 1 になりません。 |
| 11 | 未定義 | 常に'0' | |
| 12 | ESDT | 1 = 動作データが不正で停止 | |
| 13 | ESIP | 1 = 補間他軸の異常停止による停止 | |
| 14 | ESPO | 1 = パルサ用バッファオーバーフローによる停止 | (パルサ入力 × パルサ逡倍) > 動作速度の状態が 続いた場合発生します。 |
| 15 | ESAO | 1 = 補間データのレンジオーバによる停止 | |
| 17,16 | ESPE ,ESEE | 1 = エンコーダ A/B 相が同時に変化 | 停止しません。 |
| 31-18 | 未定義 | 常に'0' | |

注意 1. 以下の場合に ESDT=1 になります。

1軸だけ直線補間1モード(MOD=60h,61h,68h,69h)にしてスタートコマンドを書き込んだ時。

1軸だけ円弧補間モード(MOD=64h,65h,66h,67h,6Ch,6Dh)にしてスタートコマンドを書き込んだ時。

円弧補間モードで RIP 設定(円弧中心座標)を(0,0)にしてスタートコマンドを書き込んだ時。

3軸または4軸を円弧補間モードにしてスタートコマンドを書き込んだ時。

直線補間2モード(MOD=62h,63h,6Ah,6Bh), RIP=0 の状態でスタートコマンドを書き込んだ時。

U軸同期の円弧補間モード(MOD=66h,67h)でスタートコマンドを書き込んだ時にU軸が動作しない時。

または円弧補間動作中にU軸が動作完了になった時。

③ RIST イベントステータス (Read Only) Read:f3

イベントマスク(RIRQ)の設定をすることでイベントステータスが有効となります。

イベント報告の発生要因を確認できます。

イベント報告が発生した時に対応するビットが"1"になります。またこの時 MSTS.SINT(b5)=1 となります。

このレジスタは、読み出しによりリセットされ、MSTS.SINT(b5)=0 となります。

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ISOL | ISLT | ISCL | ISC5 | ISC4 | ISC3 | ISC2 | ISC1 | ISDE | ISDS | ISUE | ISUS | ISND | ISN | ISN | ISEN |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ISSA | ISM | ISPD | ISSD |

| bit | 名称 | 内容 |
|-------|------|--|
| 0 | ISEN | 正常停止。 本ビットの変化タイミングは、INPOS 制御設定により異なります。 |
| 1 | ISN | 次動作継続スタート。 |
| 2 | ISNM | 動作用プリレジスタフル→空き |
| 3 | ISND | CMP5 用プリレジスタフル→空き |
| 4 | ISUS | 加速開始 |
| 5 | ISUE | 加速終了 |
| 6 | ISDS | 減速開始 |
| 7 | ISDE | 減速終了 |
| 12-8 | ISCn | CMPn 比較条件成立 |
| 13 | ISCL | CLR 入力によるカウンタクリア時 |
| 14 | ISLT | LTCH 入力によるカウンタラッチ時 |
| 15 | ISOL | OLS 入力によるカウンタラッチ時 |
| 16 | ISSD | DLS 信号 OFF→ON |
| 17 | ISPD | +DR 信号 OFF→ON |
| 18 | ISM | -DR 信号 OFF→ON |
| 19 | ISSA | STA 信号 OFF→ON |
| 31-20 | 未定義 | 常に'0' |

(2-7) その他読み出しレジスタ

① RPLS 移動残パルスレジスタ(Read Only) Read:f4

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

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

② RSPD EZ カウンタ・指令速度モニタレジスタ(Read Only) Read:f5

現在速度の他に原点復帰時エンコーダ Z 相カウント値, アイドリングカウント値が読めます。指令速度は読んだ値に速度倍率を掛けた結果が指令速度となります。

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|------|------|----|---|---|------|------|---|
| 31 | 28 | 27 | 24 | 23 | 22 | 20 | 19 | 16 | 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DLCT | Z CT | | | | CRNT | VELO | |

| ビット | 名称 | 説明 |
|-------|-----------|-----------------------------|
| 15-0 | CRNT VELO | この値に速度倍率を掛けた値が指令速度。停止時は "0" |
| 19-16 | ZCT | 原点復帰に使用されている Z 相カウント値 |
| 22-20 | IDL CT | アイドリングカウント値を読出す |

③ RSDC 減速開始点計算値レジスタ(Read Only) Read:0xf6

位置決め動作における減速開始点自動演算値を確認できます。

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

④ 円弧補間歩進数レジスタ

RCI(PRCI) 円弧補間歩進数レジスタ Read:fc(cc) , Write:bc(8c) ... (括弧内はプリレジスタ)

X(V), Y(W), Z(A)軸のみに存在し, U(B)軸にはありません。

円弧補間時に減速を行う場合に円弧補間に必要な歩進数を設定します。

0 以外の値を設定することにより自動減速開始点設定による減速が行えます。(ただし合成速度一定制御 ON 時は不可)

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|
| 31 | 30 | 28 | 27 | 24 | 23 | 20 | 19 | 16 | 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| * | | | | | | | | | | | | | | | | |

設定範囲 0 ~ 2,147,483,647

⑤ RCIC 円弧補間歩進数カウンタレジスタ(Read Only) Read:fd

円弧補間歩進カウント値を読み出すレジスタ。円弧補間スタート時に RCI の値になり円弧補間パルス出力毎にダウンカウントします。

ただし, カウント値=0 になるとダウンカウントしません。また, 円弧補間完了後のカウント値は, 次に円弧補間動作を開始するまで記憶されています。

X 軸からの読み出しで X~U 軸の円弧補間歩進数。V 軸からの読み出しで V~B 軸の円弧補間歩進数。

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|
| 31 | 30 | 28 | 27 | 24 | 23 | 20 | 19 | 16 | 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| 0 | | | | | | | | | | | | | | | | |

データ範囲 0 ~ 2,147,483,647

⑥ 補間ステータスレジスタ(参考)

RIPS 補間ステータスレジスタ(Read Only) Read:ff

補間設定状態と動作状態を読み出します。

X 軸からの読み出しで X~U 軸の補間ステータス。V 軸からの読み出しで V~B 軸の補間ステータス。

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|-----|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PFu | PFz | PFy | PFx | PSu | PSz | PSy | PSx | PEu | PEz | PEy | PEx | PLu | PLz | PLy | PLx |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SED1 | SED0 | SDM1 | SDM0 | IPCC | IPCW | IPE | IPL |

| ビット | 名称 | 説明 |
|-------|--------|--|
| 0 | IPLx | 1:X(V)軸が直線補間モード |
| 1 | IPLy | 1:Y(W)軸が直線補間モード |
| 2 | IPLz | 1:Z(A)軸が直線補間モード |
| 3 | IPLu | 1:U(B)軸が直線補間モード |
| 4 | IPEx | 1:X(V)軸が PCL 間の直線補間モード |
| 5 | IPEy | 1:Y(W)軸が PCL 間の直線補間モード |
| 6 | IPEz | 1:Z(A)軸が PCL 間の直線補間モード |
| 7 | IPEu | 1:U(B)軸が PCL 間の直線補間モード |
| 8 | IPSx | 1:X(V)軸が円弧補間モード |
| 9 | IPSy | 1:Y(W)軸が円弧補間モード |
| 10 | IPSz | 1:Z(A)軸が円弧補間モード |
| 11 | IPSu | 1:U(B)軸が円弧補間モード |
| 12 | IPFx | 1:X(V)軸が合成速度一定指定 |
| 13 | IPFy | 1:Y(W)軸が合成速度一定指定 |
| 14 | IPFz | 1:Z(A)軸が合成速度一定指定 |
| 15 | IPFu | 1:U(B)軸が合成速度一定指定 |
| 16 | IPL | 1:直線補間で動作中 |
| 17 | IPE | 1:PCL 間の直線補間で動作中 |
| 18 | IPCW | 1:CW 方向円弧補間動作中 |
| 19 | IPCC | 1:CCW 方向円弧補間動作中 |
| 21-20 | SDM1-0 | 円弧補間の現在象限 00:第 1 象限, 01:第 2 象限, 10:第 3 象限, 11:第 4 象限 |
| 23-22 | SED1-0 | 円弧補間の終点象限 00:第 1 象限, 01:第 2 象限, 10:第 3 象限, 11:第 4 象限 |

3.4.12 It500_rPortB() オプションポートバイト読出し

| | |
|-------------|--|
| No. | 12 |
| 機能 | デバイスハンドルで指定されたボードの指定オプションポートまたは入力ポート・出力ポートの内容を読出します。 |
| 開発環境 | 書式 |
| VC++ | short It500_rPortB(DWORD hDevID, BYTE byCmd, BYTE* byData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| byCmd | オプションポート読出しコマンド オプションポート詳細を参照 |
| byData | オプションポート読出データ データ内容についてはオプションポート詳細を参照 |
| VC++ 記述例 | short ret; //関数の戻り値 BYTE byData; ret = It500_rPortB(hDevID, 0x80, &byData); // 各軸 ELS 入力極性設定を読出し |

3.4.13 It500_wPortB() オプションポートバイト書込み

| | |
|-------------|---|
| No. | 13 |
| 機能 | デバイスハンドルで指定されたボードの指定オプションポートまたは出力ポートへデータを書込みます。 |
| 開発環境 | 書式 |
| VC++ | short It500_wPortB(DWORD hDevID, BYTE byCmd, BYTE byData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| byCmd | オプションポート書込コマンド コマンド内容についてはオプションポート詳細を参照 |
| byData | オプションポート書込データ データ内容についてはオプションポート詳細を参照 |
| VC++ 記述例 | short ret; //関数の戻り値 ret = It500_wPortB(hDevID, 0x80, 0x01); // X 軸 ELS のみ A 接に設定 |

3.4.14 It500_rPortW() オプションポートワード(2 バイト)読出し

| | |
|-------------|--|
| No. | 14 |
| 機能 | デバイスハンドルで指定されたボードの指定オプションポートまたは入力ポート・出力ポートの内容を読出します。 |
| 開発環境 | 書式 |
| VC++ | short It500_rPortW(DWORD hDevID, BYTE byCmd, WORD* wData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| byCmd | オプションポート読出しコマンド コマンド内容についてはオプションポート詳細を参照 |
| wData | オプションポート読出データ データ内容についてはオプションポート詳細を参照 |
| VC++ 記述例 | short ret; //関数の戻り値 WORD wData; ret = It500_rPortW(hDevID, 0x80, &wData); // 各軸 ELS 極性設定を読出し |

3.4.15 It500_wPortW() オプションポートワード(2 バイト)書込み

| | |
|-------------|---|
| No. | 15 |
| 機能 | デバイスハンドルで指定されたボードの指定オプションポートまたは出力ポートへデータを書込みます。 |
| 開発環境 | 書式 |
| VC++ | short It500_wPortW(DWORD hDevID, BYTE byCmd, WORD wData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| byCmd | オプションポート書込コマンド コマンド内容についてはオプションポート詳細を参照 |
| wData | オプションポート書込データ データ内容についてはオプションポート詳細を参照 |
| VC++ 記述例 | short ret; //関数の戻り値 ret = It500_wPortW(hDevID, 0x80, 0x0001); // X 軸 ELS のみ A 接に設定 |

3.4.16 It500_rBufDW() 入出力バッファ読出し

| | |
|-------------|---|
| No. | 16 |
| 機能 | デバイスハンドルで指定されたボードの指定軸の入出力バッファを読出します。 |
| 開発環境 | 書式 |
| VC++ | short It500_rBufDW(DWORD hDevID, WORD axis, DWORD* dwData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| dwData | 入出力バッファ読出データ |
| VC++ 記述例 | <pre>short ret; //関数の戻り値 DWORD dwData; //入出力バッファデータ ret = It500_wCmdW(hDevID, 0, 0x03c0); //XY 軸 PRMV 読み出し ret = It500_rBufDW(hDevID, 0, &dwData); //X 軸入出力バッファから読出 ret = It500_rBufDW(hDevID, 1, &dwData); //Y 軸入出力バッファから読出</pre> |
| 備考 | <p><レジスタ読出関数との相違> レジスタ読出関数・・・PCLの指定軸入出力バッファを経由して目的レジスタを対象 入出力バッファ操作関数・・・PCLの指定軸入出力バッファの読出です。</p> <p><レジスタ読出関数(It500_rBufDW())の応用> 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) It500_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。 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 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> |

3.4.17 It500_wBufDW() 入出力バッファ書込み

| | |
|-------------|---|
| No. | 17 |
| 機能 | デバイスハンドルで指定されたボードの指定軸の入出力バッファにデータを書込みます。 |
| 開発環境 | 書式 |
| VC++ | short It500_wBufDW(DWORD hDevID, WORD axis, DWORD dwData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| axis | 軸指定 |
| dwData | 入出力バッファ書込データ |
| VC++ 記述例 | <pre>short ret; //関数の戻り値 ret = It500_wBufDW(hDevID, 0, 10000); //X 軸入出力バッファへ書込 ret = It500_wBufDW(hDevID, 1, 20000); //Y 軸入出力バッファへ書込 ret = It500_wCmdW(hDevID, 0, 0x0380); //XY 軸 PRMV 書込</pre> |
| 備考 | <p><レジスタ書込関数との相違> レジスタ書込関数・・・PCLの指定軸入出力バッファを経由して目的レジスタを対象。 入出力バッファ操作関数・・・PCLの指定軸入出力バッファの書込みです。</p> <p><レジスタ書込関数(It500_wBufDW())の応用> 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) 書込みを行う全ての軸入出力バッファに所定データを書込 It500_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。 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> |

3.4.18 It500_SetIntCall() 割込処理関数の登録

| | |
|-------------|---|
| No. | 18 |
| 機能 | デバイス ID で指定されたCPDボードからの割込が発生した場合の処理関数を登録 |
| 開発環境 | 書式 |
| VC++ | short It500_SetIntCall(DWORD hDevID, PINTPROC fnIntCall); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| 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) |
| VC++ 記述例 | short ret; //関数の戻り値 PINTPROC int_module; // 割込処理モジュールアドレス ret = It500_GetBoardCode(hDevID, &bcode); |
| 備考 | 割込みイベントの受信は、It500_WaitInt()で行ないます |

3.4.19 It500_ResetIntCall() 割込機能の削除

| | |
|-------------|--|
| No. | 19 |
| 機能 | 割込み機能の削除をします |
| 開発環境 | 書式 |
| VC++ | short It500_ResetIntCall(DWORD hDevID); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| VC++ 記述例 | short ret; //関数の戻り値 ret = It500_ResetIntCall(hDevID); |
| 備考 | 割込みイベント受信用のハンドルは削除されます |

3.4.20 It500_WaitInt() 割込イベント待ち

| | |
|-------------|---|
| No. | 20 |
| 機能 | 割込み機能から、割込み発生時の情報取得を完了した事を受信します。 タイムアウト指定で指定時間内に割込みイベントが発生しない場合には、エラーを返します |
| 開発環境 | 書式 |
| VC++ | short It500_WaitInt(RTHANDLE semIntEvent, WORD timeout, WORD evRst, DWORD *intEvent); |
| 引数 | 説明 |
| semIntEvent | 割込みイベント受信用のハンドル |
| timeout | 10mSec 単位のタイムアウト指定 (0~FFFh, 永久待ち:FFFh,) |
| evRst | 1: 残りの割込みイベントを削除 |
| intEvent | 残りの割込みイベント数を取得(割込みが連続した時) |
| VC++ 記述例 | short ret; INTHANDLE semIntEvent; DWORD intEvent; ret = It500_WaitInt(semIntEvent, 500, 1, &intEvent); |

3.4.21 It500_GetIntData 割込イベント要因の取得

| | |
|-------------|--|
| No. | 21 |
| 機能 | <p>割込み発生時の割込要因と、全軸(12 軸)の下記情報を読み込みます。</p> <p>①メインステータス(割込要因の無い PCL は、0 を設定)</p> <p>②エラーステータスレジスタ(メインステータスで要因のない軸は、0 を設定)</p> <p>③イベントステータスレジスタ(メインステータスで要因のない軸は、0 を設定)</p> |
| 開発環境 | 書式 |
| VC++ | short It500_GetIntData(DWORD hDevID, WORD *rem, INTEVDATA *intEvData); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| VC++ 記述例 | <pre>short ret; WORD rem; INTEVDATA intEvData; ret = It500_GetIntData(hDevID, &rem, &intEvData);</pre> |
| 備考 | 割込みイベントの最大スタック数は 31 件となります |

① 割込処理開始手順

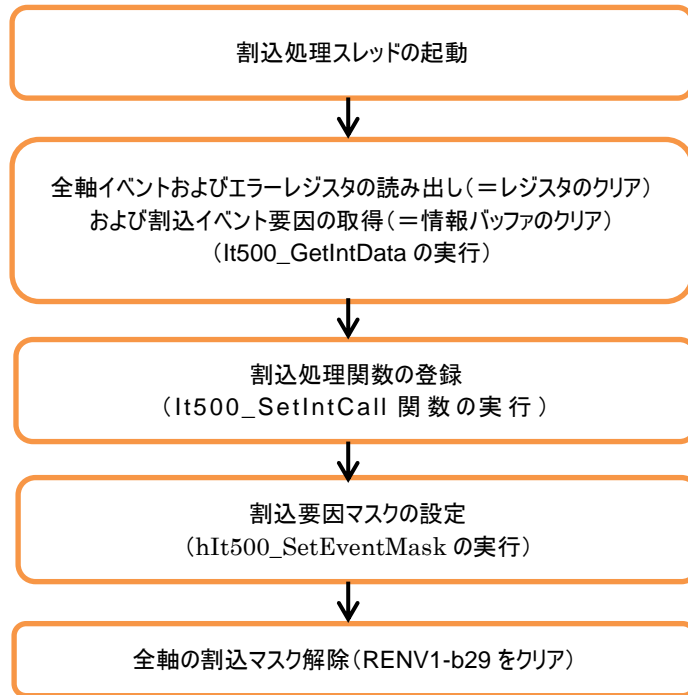


表 3.4-1 割込処理開始手順

② 割込スレッド内処理手順

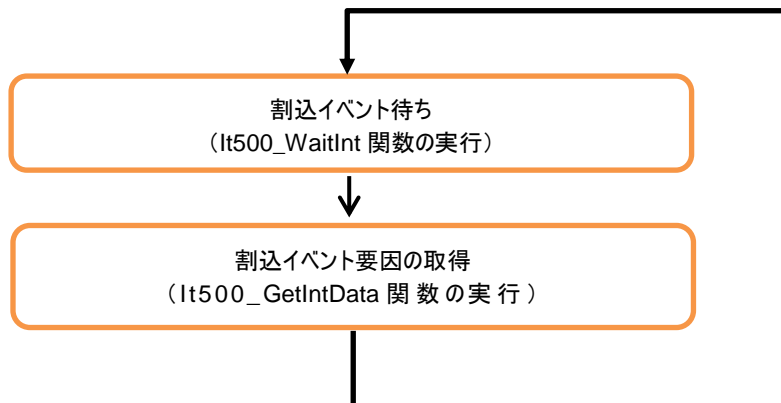


表 3.4-2 割込スレッド内処理手順

③ 割込処理停止手順

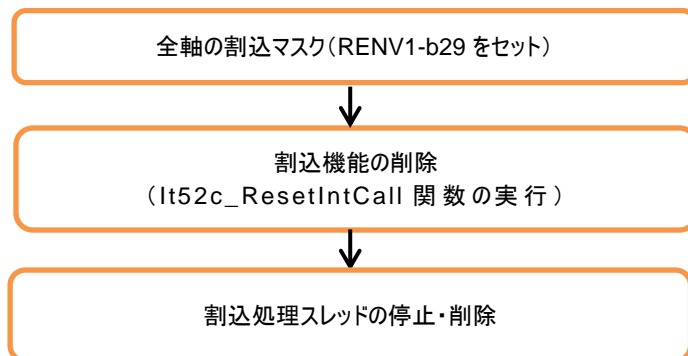


表 3.4-3 割込処理停止手順

3.4.22 It500_GetDevVerNo バージョン番号の取得

| | | | | | | | | | | | | | | | | |
|-------------------------|--|-----------------------|-----------|-------------|-----|---|---------------------|--|-----------------------|--|--|-----------|--|-------------|-----------|-------------|
| No. | 22 | | | | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | |
| 開発環境 | 書式 | | | | | | | | | | | | | | | |
| VC++ | short It500_GetDevVerNo(DWORD* verno); | | | | | | | | | | | | | | | |
| 引数 | 説明 | | | | | | | | | | | | | | | |
| Verno | バージョン番号 | | | | | | | | | | | | | | | |
| VC++ 記述例 | short ret; DWORD verno; ret = It500_GetDevVerNo(&verno); | | | | | | | | | | | | | | | |
| 備考 | <table border="1"> <tr> <td>31</td> <td>24 23</td> <td>16 15</td> <td>8 7</td> <td>0</td> </tr> <tr> <td colspan="2">ドライバ関数バージョン番号(x.xx)</td> <td colspan="3">デバイスドライババージョン番号(x.xx)</td> </tr> <tr> <td colspan="2">メジャーバージョン</td> <td colspan="1">マイナー以下バージョン</td> <td colspan="1">メジャーバージョン</td> <td colspan="1">マイナー以下バージョン</td> </tr> </table> | 31 | 24 23 | 16 15 | 8 7 | 0 | ドライバ関数バージョン番号(x.xx) | | デバイスドライババージョン番号(x.xx) | | | メジャーバージョン | | マイナー以下バージョン | メジャーバージョン | マイナー以下バージョン |
| | 31 | 24 23 | 16 15 | 8 7 | 0 | | | | | | | | | | | |
| ドライバ関数バージョン番号(x.xx) | | デバイスドライババージョン番号(x.xx) | | | | | | | | | | | | | | |
| メジャーバージョン | | マイナー以下バージョン | メジャーバージョン | マイナー以下バージョン | | | | | | | | | | | | |
| ※ ハイパーテックで管理の番号が設定されます。 | | | | | | | | | | | | | | | | |

3.4.23 It500_GetBoardCode() ボード固有コードの取得

| | |
|-------------|---|
| No. | 23 |
| 機能 | ボード固有コードの取得を行います。 |
| 開発環境 | 書式 |
| VC++ | short It500_GetBoardCode(DWORD hDevID, WORD* bcode); |
| 引数 | 説明 |
| hDevID | デバイスハンドル |
| bcode | ボード固有コード HPCI-CPD578, HPCI-CPD578N, HPCI-CPD574N, HPCIe-CPD674N, HPCIe-CPD678N … 578a(Hex) HPCI-CPD5016 … 5016(Hex) HPCI-CPD553 … 5530(Hex) 上記以外 … 5254(Hex) |
| VC++ 記述例 | short ret; //関数の戻り値 WORD* bcode; //ボード固有コード ret = It500_GetBoardCode(hDevID, &bcode); |

3.4.24 It500_GetLastDetailError 詳細エラーコードの取得

| | |
|-----|--------------------------|
| No. | 21 |
| 機能 | 最後に発生したエラーの詳細コードを読み出します。 |

| | |
|------|--|
| 開発環境 | 書式 |
| VC++ | short It500_GetLastDetailError(WORD *errcode); |

| 引数 | 説明 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------------|---|------------------------------|----------|----|-------------------|--|--|------------------------------|--|--|----------|--------|-------------------|--------------|--------|------------------|-----------|--------|-----------------|-----------|--------|---------------|----------------------|--|--|---------------|--------|--------------------|--------------|--------|----------------|--------------|--------|----------------|-------------|--------|-------------------|----------------|--------|---------------------|----------------|--------|---------------------|---------------|--------|-----------------|----------------------|--|--|----------|--------|-------------------------|------------|--------|---------------------------|-----------|--------|----------------------|-----------|--------|--------------------|-----------|--------|------------------|-----------|--------|------------------|--------------|--------|-----------------------|-------------|--------|------------------------------|----------|--------|--------------------|-----------|--------|-------------------------|------------|--------|-----------------|-------------|--------|-----------------|-----------|--------|--------------------|-----------|--------|---------------------|
| hDevID | デバイスハンドル | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pdVerNo | バージョン番号読み出しデータ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VC++ 記述例 | short ret; WORD errcode; ret = It500_GetLastDetailError(&errcode); | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 備考 | <p>詳細エラーの情報は次のエラーが発生するまで更新されません。 読み取り後もクリアされず残ります。</p> <p>-----</p> <p>詳細エラーコード一覧</p> <p>-----</p> <table border="1"> <thead> <tr> <th>記号</th> <th>詳細エラーコード</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td colspan="3">(hI500Type.h で定義)</td> </tr> <tr> <td colspan="3">/* 主に外部からの渡されるデータに起因するエラー */</td> </tr> <tr> <td>IHDLEERR</td> <td>0x1001</td> <td>/* 指定したハンドルが無効 */</td> </tr> <tr> <td>IEADDRESSERR</td> <td>0x1002</td> <td>/* アドレスの指定エラー */</td> </tr> <tr> <td>IESIZEERR</td> <td>0x1003</td> <td>/* サイズの指定エラー */</td> </tr> <tr> <td>IEAXISERR</td> <td>0x1004</td> <td>/* 軸指定が範囲外 */</td> </tr> <tr> <td colspan="3">/* 主に内部処理に起因するエラー */</td> </tr> <tr> <td>IMSG_FUNCTION</td> <td>0x2001</td> <td>/* 該当する処理が存在しない */</td> </tr> <tr> <td>ISEMAPHORE_C</td> <td>0x2002</td> <td>/* セマフォ作成失敗 */</td> </tr> <tr> <td>ISEMAPHORE_D</td> <td>0x2003</td> <td>/* セマフォ削除失敗 */</td> </tr> <tr> <td>IDELTHREAED</td> <td>0x2004</td> <td>/* 割り込みタスク削除失敗 */</td> </tr> <tr> <td>ICREATETHREAED</td> <td>0x2005</td> <td>/* 既に割り込みタスク生成済み */</td> </tr> <tr> <td>IINT_EVENT_NON</td> <td>0x2006</td> <td>/* 割り込みイベント情報が無い */</td> </tr> <tr> <td>ITASK_CONTROL</td> <td>0x2007</td> <td>/* スレッド生成に失敗 */</td> </tr> <tr> <td colspan="3">/* 主にデバイスに起因するエラー */</td> </tr> <tr> <td>IETIMOUT</td> <td>0x4001</td> <td>/* タイムアウト指定でタイムアウト発生 */</td> </tr> <tr> <td>IENONEDATA</td> <td>0x4002</td> <td>/* 情報が存在しない(全情報読み込み完了) */</td> </tr> <tr> <td>IEINITERR</td> <td>0x4003</td> <td>/* API の初期化がされてない */</td> </tr> <tr> <td>IENONEDRV</td> <td>0x4004</td> <td>/* ドライバが起動されてない */</td> </tr> <tr> <td>IEMSGSEND</td> <td>0x4005</td> <td>/* MSG 送信でエラー */</td> </tr> <tr> <td>IEMSGRECV</td> <td>0x4006</td> <td>/* MSG 受信でエラー */</td> </tr> <tr> <td>IRECVTIMEOUT</td> <td>0x4007</td> <td>/* MSG 受信でタイムアウト発生 */</td> </tr> <tr> <td>IEINTEVRECV</td> <td>0x4008</td> <td>/* 割り込みイベント受信でエラー(セマフォ削除) */</td> </tr> <tr> <td>IEMEMERR</td> <td>0x4009</td> <td>/* メッセージメモリ取得失敗 */</td> </tr> <tr> <td>IEGETMHDL</td> <td>0x4010</td> <td>/* メッセージメモリハンドルの取得失敗 */</td> </tr> <tr> <td>IERNCREATE</td> <td>0x4011</td> <td>/* リージョン取得失敗 */</td> </tr> <tr> <td>IERNRELEASE</td> <td>0x4012</td> <td>/* リージョン開放失敗 */</td> </tr> <tr> <td>IEMAILBOX</td> <td>0x4013</td> <td>/* メールボックスの取得失敗 */</td> </tr> <tr> <td>IESYSCALL</td> <td>0x4017</td> <td>/* システムコールでエラー発生 */</td> </tr> </tbody> </table> | 記号 | 詳細エラーコード | 意味 | (hI500Type.h で定義) | | | /* 主に外部からの渡されるデータに起因するエラー */ | | | IHDLEERR | 0x1001 | /* 指定したハンドルが無効 */ | IEADDRESSERR | 0x1002 | /* アドレスの指定エラー */ | IESIZEERR | 0x1003 | /* サイズの指定エラー */ | IEAXISERR | 0x1004 | /* 軸指定が範囲外 */ | /* 主に内部処理に起因するエラー */ | | | IMSG_FUNCTION | 0x2001 | /* 該当する処理が存在しない */ | ISEMAPHORE_C | 0x2002 | /* セマフォ作成失敗 */ | ISEMAPHORE_D | 0x2003 | /* セマフォ削除失敗 */ | IDELTHREAED | 0x2004 | /* 割り込みタスク削除失敗 */ | ICREATETHREAED | 0x2005 | /* 既に割り込みタスク生成済み */ | IINT_EVENT_NON | 0x2006 | /* 割り込みイベント情報が無い */ | ITASK_CONTROL | 0x2007 | /* スレッド生成に失敗 */ | /* 主にデバイスに起因するエラー */ | | | IETIMOUT | 0x4001 | /* タイムアウト指定でタイムアウト発生 */ | IENONEDATA | 0x4002 | /* 情報が存在しない(全情報読み込み完了) */ | IEINITERR | 0x4003 | /* API の初期化がされてない */ | IENONEDRV | 0x4004 | /* ドライバが起動されてない */ | IEMSGSEND | 0x4005 | /* MSG 送信でエラー */ | IEMSGRECV | 0x4006 | /* MSG 受信でエラー */ | IRECVTIMEOUT | 0x4007 | /* MSG 受信でタイムアウト発生 */ | IEINTEVRECV | 0x4008 | /* 割り込みイベント受信でエラー(セマフォ削除) */ | IEMEMERR | 0x4009 | /* メッセージメモリ取得失敗 */ | IEGETMHDL | 0x4010 | /* メッセージメモリハンドルの取得失敗 */ | IERNCREATE | 0x4011 | /* リージョン取得失敗 */ | IERNRELEASE | 0x4012 | /* リージョン開放失敗 */ | IEMAILBOX | 0x4013 | /* メールボックスの取得失敗 */ | IESYSCALL | 0x4017 | /* システムコールでエラー発生 */ |
| 記号 | 詳細エラーコード | 意味 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (hI500Type.h で定義) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| /* 主に外部からの渡されるデータに起因するエラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IHDLEERR | 0x1001 | /* 指定したハンドルが無効 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEADDRESSERR | 0x1002 | /* アドレスの指定エラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IESIZEERR | 0x1003 | /* サイズの指定エラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEAXISERR | 0x1004 | /* 軸指定が範囲外 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| /* 主に内部処理に起因するエラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IMSG_FUNCTION | 0x2001 | /* 該当する処理が存在しない */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ISEMAPHORE_C | 0x2002 | /* セマフォ作成失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ISEMAPHORE_D | 0x2003 | /* セマフォ削除失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IDELTHREAED | 0x2004 | /* 割り込みタスク削除失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ICREATETHREAED | 0x2005 | /* 既に割り込みタスク生成済み */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IINT_EVENT_NON | 0x2006 | /* 割り込みイベント情報が無い */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ITASK_CONTROL | 0x2007 | /* スレッド生成に失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| /* 主にデバイスに起因するエラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IETIMOUT | 0x4001 | /* タイムアウト指定でタイムアウト発生 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IENONEDATA | 0x4002 | /* 情報が存在しない(全情報読み込み完了) */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEINITERR | 0x4003 | /* API の初期化がされてない */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IENONEDRV | 0x4004 | /* ドライバが起動されてない */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEMSGSEND | 0x4005 | /* MSG 送信でエラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEMSGRECV | 0x4006 | /* MSG 受信でエラー */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IRECVTIMEOUT | 0x4007 | /* MSG 受信でタイムアウト発生 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEINTEVRECV | 0x4008 | /* 割り込みイベント受信でエラー(セマフォ削除) */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEMEMERR | 0x4009 | /* メッセージメモリ取得失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEGETMHDL | 0x4010 | /* メッセージメモリハンドルの取得失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IERNCREATE | 0x4011 | /* リージョン取得失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IERNRELEASE | 0x4012 | /* リージョン開放失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IEMAILBOX | 0x4013 | /* メールボックスの取得失敗 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IESYSCALL | 0x4017 | /* システムコールでエラー発生 */ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |