HPCI-PPD534A シリーズ
HPCI-PPD516A シリーズ
添付ソフトウェアマニュアル

株式会社ハイバーテック
http://www.hivertec.co.jp/

この説明書では

ＨＰＣＩ－ＰＰＤ５３４Ａシリーズのボードは
ＨＰＣＩ－ＰＰＤ５３２
ＨＰＣＩ－ＰＰＤ５３３
ＨＰＣＩ－ＰＰＤ５３４
および
ＨＰＣＩ－ＰＰＤ５３２Ａ
ＨＰＣＩ－ＰＰＤ５３３Ａ
ＨＰＣＩ－ＰＰＤ５３４Ａ　です.

ＨＰＣＩ－ＰＰＤ５１６Ａシリーズのボードは
ＨＰＣＩ－ＰＰＤ５１４Ａ
および
ＨＰＣＩ－ＰＰＤ５１６Ａ　です.

Ｗｉｎｄｏｗｓ版デバイスドライバ　の種別は

Ｗｉｎｄｏｗｓ９５，９８	ではｈｉｐｐｄ５３０．ｖｘｄ	
ＷｉｎｄｏｗｓＮＴ	ではｈｉｐｐｄ５３０．ｓｙｓ	
Ｗｉｎｄｏｗｓ２０００	ではｈｐ５３０ｗ２ｋ．ｓｙｓ	です.
ＷｉｎｄｏｗｓＸＰ	ではｈｐ５３０ｗｘｐ．ｓｙｓ	です.
ＷｉｎｄｏｗｓＶｉｓｔａ，７（３２ビット）	ではｈｐ５３０ｗｄｍ．ｓｙｓ	
ＷｉｎｄｏｗｓＶｉｓｔａ，７（６４ビット）	ではｈｐ５３０ｘ６４．ｓｙｓ	です.

Ｗｉｎｄｏｗｓ版ドライバＩ/Ｆライブラリ　として
ｈｉｐｐｄ５３０．ｄｌｌを上記ＯＳ共通で使用します.

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

第４．３版　２０１１年　１月２７日発行

保証範囲

１．本シリーズ製品の保証期間は，お買い上げ頂いた日より３年間です．保証期間中に弊社の判断により欠陥が判明した場合には，本シリーズ製品を弊社に引き取り，修理または交換を行います．

２．保証期間内外に関わらず，弊社製品の使用，供給（納期）または故障に起因する，お客様及び第三者が被った，直接，間接，２次的な損害あるいは，遺失利益の損害に付いて，弊社は本シリーズ製品の販売価格以上の責任を負わないものとしますので，予めご了承下さい．

免責事項

１．本マニュアルに記載された内容に沿わない，製品の取付，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，予めご了承下さい．

２．本シリーズ製品は，一般電子機器用（工作機械・計測機器・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．
ゆえに本シリーズ製品の安全性，品質および性能に関しては，本マニュアル（またはカタログ）に記載してあること以外は明示的にも黙示的にも一切保証するものではありませんので，予めご了承下さい．

３．保証期間内外に関わらず，お客様が行った弊社の承認しない製品の改造または，修理が原因で生じた損害に対しましては，一切の責任を負いかねますので，予めご了承下さい．

４．本マニュアルに記載された内容について，弊社もしくは，第三者の特許権，著作権，商標権，その他の知的所有権の権利に対する保証または実施権の許諾を行うものではありません．
また本マニュアルに記載された情報を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合，弊社は，その責任を負いかねますので，予めご了承下さい．

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，添付ソフトウェアをご使用して頂く場合の取扱い，留意点に付いて記入してありますので，必ずご一読の上ご利用をお願い致します.
尚，本マニュアルは，本書が添付された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．本シリーズ製品の知識，安全の情報及び注意事項の全てに付いて習熟してからご使用下さい.
本ユーザーズマニュアルでは，安全注意事項のランクを「警告」，「注意」として区分してあります.

⚠ **警 告**	この表示を無視して，誤った取扱いをすると，人が死亡または重傷を負う可能性が想定される内容を示しています.
⚠ **注 意**	この表示を無視して，誤った取扱いをすると，人が傷害を負う可能性または物的損害が想定される内容を示しています.

１．対象ユーザー

⚠ **注 意**

本シリーズ製品およびマニュアルは，以下の様な，ユーザーを対象としています.

⚠
・拡張用ボードの増設および配線に付いて基本的な知識を有している方.
・制御用電子機器およびパソコン等に付いて基本的な知識を有している方.
・ＯＳの操作およびソフトウェア開発環境等に付いて基本的な知識を有している方.

２．適合ＯＳ

⚠ **注 意**

⚠ 本製品は Windows 7, Windows Vista, Windows XP Professional, Windows XP HomeEdition, Windows NT4.0, Windows2000 Professional, Windows95, 及び Windows98 においてボードの制御を行う為のソフトウェアです. 上記以外のＯＳでのご使用については，弊社営業までお問合せ下さい.

３．ハードウェアの設定・取付け・接続

警　告

ボードの取付け，配線に際しては，ユーザーズマニュアルを良くお読み いただき，ユーザーズマニュアルの内容にしたがって実施願います．

４．「動かしてみる」プログラムの実行

警　告

本添付ソフト中の「動かしてみる」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確認するとともに，ボードの機能・動作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」プログラムを定常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．

５．対象開発ツール

注　意

本添付ソフト中のサンプルプログラムは，以下の開発ツールを対象にしています．
・Microsoft Visual Studio
上記以外の開発ツールでのご利用については，弊社営業までお問合せ下さい．

６．サンプルプログラムの実行

注 意

本添付ソフト中のサンプルプログラムは，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，サンプルプログラムを定常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．

7．ユーザープログラムの作成

警 告

ユーザープログラムの作成にあたっては，モータや装置の特性を考慮し，必要なインターロック・安全対策処理等を十分盛り込んだ設計として下さい.
プログラムコード・データのわずかな違いにより予想外の動作をして，機器や人体に損傷を与える恐れが有ります.
プログラム作成・試運転時共，十分な注意をお願いいたします.

目　次

１．はじめに

本書は，ＨＰＣＩ－ＰＰＤ５３４Ａシリーズボード（ PPD532,PPD533,PPD534, PPD532A, PPD533A,PPD534A ）及びＨＰＣＩ－ＰＰＤ５１６Ａシリーズボード（PPD514A,PPD516A）の添付ソフトウェアに関して解説を行うものです．（上記のボード総称として以降は**ＰＰＤ５３ｘ**と記します．）

この添付ソフトウェアはWindows ７，Windows Ｖista，WindowsＸＰ Ｐrofessional・HomeＥdition(以降**ＸＰ**と記します)，Windows２０００(以降**２Ｋ**と記します)，WindowsＮＴ４．０，及びWindows９５･９８(以降**９Ｘ**と記します)において，**ＰＰＤ５３ｘ**ボードの制御を行う為に使用します．

ボードの制御に関する説明については，個々の関数内及び”附Ａ”で関連項目を取り上げていますが，詳細な説明が必要な場合には，ボード添付のユーザーズ・マニュアルを参照してください．

１．１　添付ソフトウエアの構成

１．１．１　３２ビット版ＯＳ

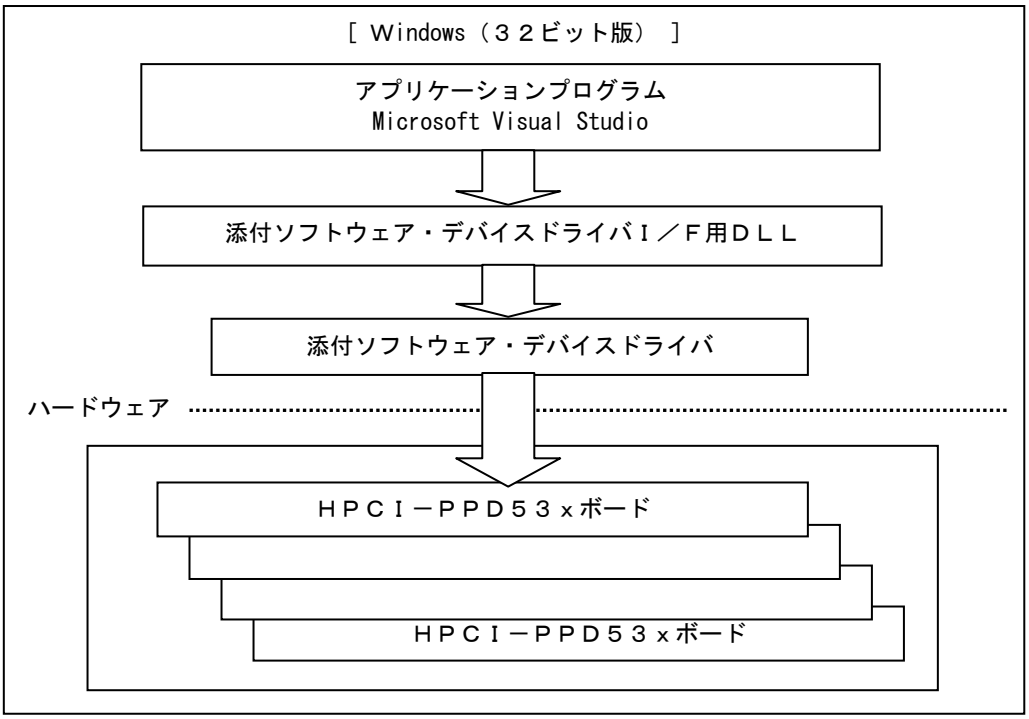


表１．１－１　３２ビット版ＯＳ添付ソフトウェア関連図

１．１．２　６４ビット版ＯＳ

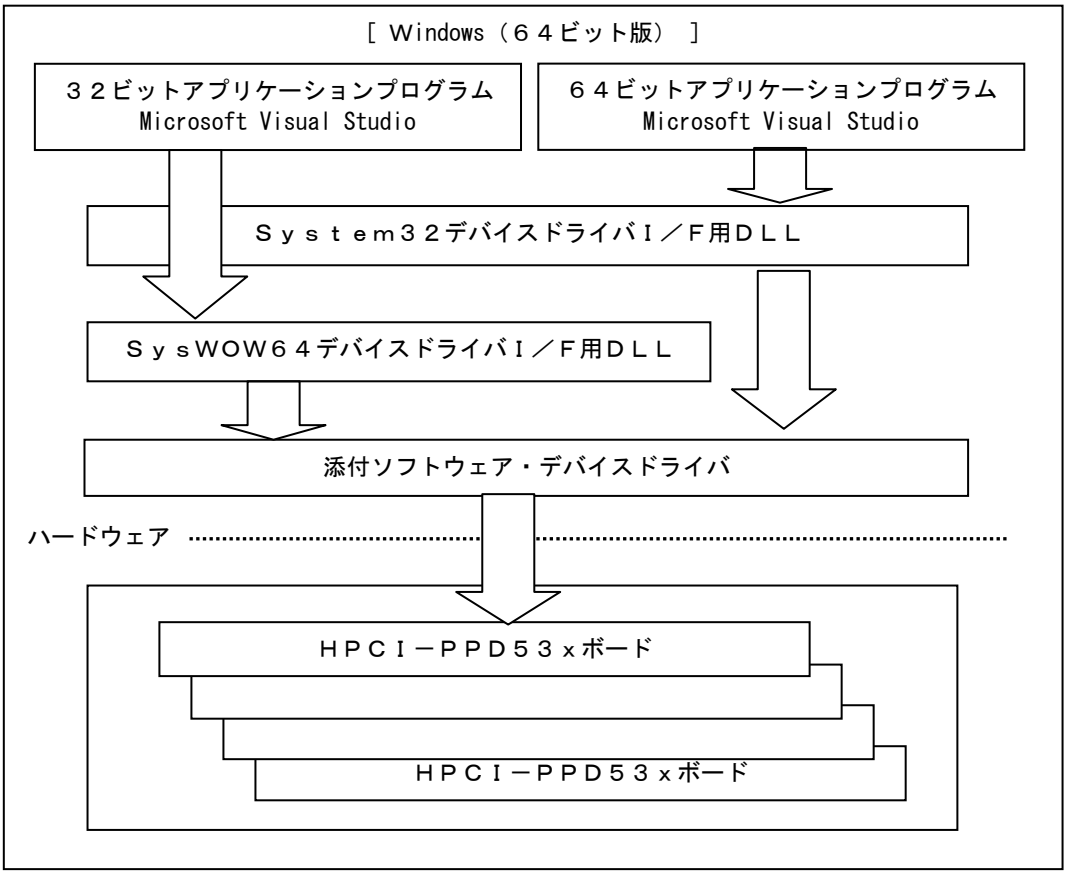


表１．１－２　６４ビット版ＯＳ添付ソフトウェア関連図

１．２　添付ディスクの内容

```
¥
├──document─────PPDxxxA_U_y_yyJ.pdf   ····ユーザーズマニュアル４種(xxx = ボード名、y_yy = バージョン)
│               └──PPD530_W_4_30J.pdf    ····添付ソフトウェアマニュアル
│
├──Win9x────hippd530.vxd     ···· Windows9X仮想デバイスドライバ
│           └──hippd530.inf     ···· Windows9X用インストール情報 (VxDインストール用)
├──WinNt────hippd530.sys     ···· WindowsNT4.0デバイスドライバ
│           ├──p530inst.inf     ···· インストール情報ファイル
│           └──p530inst.bat     ···· インストール用バッチファイル
├──Win2k────hp530w2k.sys     ···· Windows2000デバイスドライバ
│           └──hp530w2k.inf     ···· インストール情報ファイル
├──WinXp────hp530wxp.sys     ···· WindowsXPデバイスドライバ
│           └──Hp530wxp.inf     ···· インストール情報ファイル
├──Win_x86──────dpinst.exe     ···· ドライバパッケージインストーラ
│               ├──dpinst.xml    ···· ライバパッケージインストーラ用xmlファイル
│               ├──hp530wdm.cat  ···· セキュリティカタログファイル
│               ├──hp530wdm.sys  ···· WindowsVista,7(32ビット版)デバイスドライバ
│               ├──hp530wdm.inf  ···· インストール情報ファイル
│               └──hippd530.dll  ···· ドライバI/F用DLL
├──Win_x64──────dpinst.exe     ···· ドライバパッケージインストーラ
│               ├──dpinst.xml    ···· ライバパッケージインストーラ用xmlファイル
│               ├──hp530x64.cat  ···· セキュリティカタログファイル
│               ├──hp530x64.sys  ···· WindowsVista,7(64ビット版)デバイスドライバ
│               ├──hp530x64.inf  ···· インストール情報ファイル
│               └──hippd530.dll  ···· ドライバI/F用DLL(32ビット版)
│               └──hippd530.x64  ···· ドライバI/F用DLL(64ビット版)
│
├──Include──────Vb──────spp53002.txt    ···· 外部関数宣言テキスト( VB 6.0アプリケーション構築用 )
│               ├──Vc_x86──hippd530.lib    ···· (32ビット版)インポートライブラリ (Cアプリケーション構築用)
│               │        └──hippd530.h      ···· ヘッダーファイル    (Cアプリケーション構築用)
│               └──Vc_x64──hippd530.lib    ···· (64ビット版)インポートライブラリ (Cアプリケーション構築用)
│                        └──hippd530.h      ···· ヘッダーファイル    (Cアプリケーション構築用)
│
├──Sample─────ppd530──Vc─────spp53400.exe    ···· PPD53x用サンプル実行ファイル (Visual C)
│              │       │      ├──spp53400.dsw    ···· Visual C用 サンプル プロジェクトワークスペース
│              │       │      ├──spp53400.dsp    ····            サンプル プロジェクトファイル
│              │       │      ├──spp53400.opt    ····            サンプル optファイル
│              │       │      ├──spp53400.c      ···· Visual C用 サンプル ソースプログラム
│              │       │      ├──spp53400.h      ····            サンプル ヘッダーファイル
│              │       │      ├──spp53400.rc     ····            サンプル リソースファイル
│              │       │      ├──resource.h      ····            サンプル リソースヘッダーファイル
│              │       │      ├──bgreen.bmp      ····            サンプル ビットマップ（緑）
│              │       │      ├──bredD.bmp       ····            サンプル ビットマップ（赤）
│              │       │      ├──bwhite.bmp      ····            サンプル ビットマップ（（白）
│              │       │      └──hippd530.xxx    ···· C言語用 ドライバ添付ファイル３種(xxx = lib,h,dll)
│              │       ├──Vcpp─ spp53401.exe    ···· PPD53x用サンプル実行ファイル(VC++(C++コーディング)版)
│              │       │      ├──spp53401.dsw    ···· VC++(C++)用サンプル プロジェクトワークスペース
│              │       │      ├──spp53401.dsp    ····            サンプル プロジェクトファイル
│              │       │      ├──spp53401.opt    ····            サンプル optファイル
│              │       │      ├──README.txt      ····            サンプル 構成ファイルの解説
│              │       │      ├──spp53401dlg.cpp ···· VC++(C++)用サンプル ソースプログラム
│              │       │      ├──spp53401dlg.h   ····            サンプル ヘッダーファイル
│              │       │      ├──hippd530.xxx    ···· C言語用ドライバ添付ファイル３種(xxx = lib,h,dll)
│              │       │      ├──その他のファイル    ···· "readme.txt"ファイルを参照下さい。
│              │       │      └──RES─────bgreen.bmp       ···· サンプル ビットマップファイル（緑）
│              │       │                 ├──bred.bmp        ····           ビットマップファイル（赤）
│              │       │                 ├──bwhite.bmp      ····           ビットマップファイル（白）
│              │       │                 ├──bgray.bmp       ····           ビットマップファイル（灰）
│              │       │                 └──その他のファイル ···· "readme.txt"ファイルを参照下さい。
│              │       └──Vb──── spp53402.exe  ···· PPD53x用サンプル実行ファイル (Visual Basic 6.0)
│              │              ├──spp53402.vbp  ···· VB用サンプル プロジェクトファイル
│              │              ├──spp53402.frm  ····      サンプル フォームモジュールファイル
│              │              ├──spp53402.bas  ····      サンプル 標準モジュールファイル
│              │              └──hippd530.dll  ···· ドライバI/F用DLL
│              │
.              .
.              .
.        （次頁へ続く）
.              .
.              .
```

```
.               .
.               .
|               |
|               └ppd510──Vc─────spp51600.exe    ···· PPD51x用サンプル実行ファイル (Visual C)
|                       |       ├──spp51600.dsw    ···· Visual C用 サンプル プロジェクトワークスペース
|                       |       ├──spp51600.dsp    ····            サンプル プロジェクトファイル
|                       |       ├──spp51600.opt    ····            サンプル optファイル
|                       |       ├──spp51600.c      ···· Visual C用 サンプル ソースプログラム
|                       |       ├──spp51600.h      ····            サンプル ヘッダーファイル
|                       |       ├──spp51600.rc     ····            サンプル リソースファイル
|                       |       ├──resource.h      ····            サンプル リソースヘッダーファイル
|                       |       ├──bgreen.bmp      ····            サンプル ビットマップ（緑）
|                       |       ├──bredD.bmp       ····            サンプル ビットマップ（赤）
|                       |       ├──bwhite.bmp      ····            サンプル ビットマップ（（白）
|                       |       └──hippd530.xxx    ···· C言語用 ドライバ添付ファイル３種(xxx = lib,h,dll)
|                       ├──Vcpp─ spp51601.exe    ···· PPD51x用サンプル実行ファイル(VC++(C++コーディング)版)
|                       |       ├──spp51601.dsw    ···· VC++(C++)用サンプル プロジェクトワークスペース
|                       |       ├──spp51601.dsp    ····            サンプル プロジェクトファイル
|                       |       ├──spp51601.opt    ····            サンプル optファイル
|                       |       ├──README.txt      ····            サンプル 構成ファイルの解説
|                       |       ├──spp51601dlg.cpp ···· VC++(C++)用サンプル ソースプログラム
|                       |       ├──spp51601dlg.h   ····            サンプル ヘッダーファイル
|                       |       ├──hippd530.xxx    ···· C言語用ドライバ添付ファイル３種(xxx = lib,h,dll)
|                       |       ├──その他のファイル    ···· "readme.txt"ファイルを参照下さい。
|                       |       └──RES─────bgreen.bmp       ···· サンプル ビットマップファイル（緑）
|                       |                  ├──bred.bmp        ····           ビットマップファイル（赤）
|                       |                  ├──bwhite.bmp      ····           ビットマップファイル（白）
|                       |                  ├──bgray.bmp       ····           ビットマップファイル（灰）
|                       |                  └──その他のファイル ···· "readme.txt"ファイルを参照下さい。
|                       └──Vb─── spp51602.exe  ···· PPD51x用サンプル実行ファイル (Visual Basic 6.0)
|                               ├──spp51602.vbp  ···· VB用サンプル プロジェクトファイル
|                               ├──spp51602.frm  ····      サンプル フォームモジュールファイル
|                               ├──spp51602.bas  ····      サンプル 標準モジュールファイル
|                               └──hippd530.dll  ···· ドライバI/F用DLL
|
├──Test─────tpp53400.exe      ···· PPD53x用動かしてみる実行ファイル
|           └──tpp51600.exe      ···· PPD51x用動かしてみる実行ファイル
├──p530uins.exe         ···· アンインストール用実行プログラム (Win9X,WinNT,Win2K, WinXP 共通)
├──hippd530.dll         ···· ドライバI/F用DLL
└──History.txt          ···· 添付ディスクの更新履歴
```

２．デバイスドライバのインストール

２．１　Windows９Ｘへのインストール

① パソコンの電源がＯＦＦであることを確認した後，ＰＰＤ５３ｘボードをパソコンのＰＣＩバススロットに装着します．パソコンの電源をＯＮにしてWindows を起動します．
② Windows９Ｘが起動すると，ＰＰＤ５３ｘがシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．
③ システムがインストール元ディレクトリの指定を要求してきたら，添付のＣＤをＣＤドライブに挿入します．
④ □検索場所の指定　のチェックボックスを必ずチェックします．
(Windows95 では，場所の指定ボタンをクリックします．）
⑤ **ＣＤドライブ：¥ＷＩＮ９Ｘ**を指定してください．
後はシステムの指示に従ってインストールを完了させます．

２．２　WindowsＮＴ4.0 へのインストール

（1）デバイスドライバのインストール

① 添付のＣＤをＣＤドライブに挿入します．
ＮＴエクスプローラを起動し，ＣＤドライブ:¥WinNT¥p530inst.inf を選択します．
② 次にマウスの右ボタンをクリックします．表示されるポップアップメニューから「インストール」を選択します．
この操作によりデバイスドライバのインストールが開始されます．
後はシステムの指示に従ってインストールを完了させます．

コマンドプロンプトから，ＣＤドライブ:¥WinNT¥ p530inst.bat を実行させても同様にインストールが開始されます．

（2）デバイスの開始と停止

インストール完了後，デバイスドライバは「自動開始」に設定されており，WindowsＮＴ起動時にＰＰＤ５３ｘボードに対するサービスも開始されます．
何らかの理由により停止への変更が必要である場合は次の作業を行います．

① コントロールパネルから「デバイス」アイコンをダブルクリックし，デバイス一覧の中から
「Hivertec HPCI-PPD530(PCI)」を選択します．
② 「スタートアップ」ボタンを押すことにより「スタートアップの種類」ダイアログが表示されます．
「停止」を選択し，「ＯＫ」を押します．
ＰＰＤ５３ｘデバイスを再開始させる場合も，コントロールパネルの「デバイス」操作行います．
「Hivertec HPCI-PPD530(PCI)」を選択し，「開始」ボタンを押します．

２．３　Windows２０００へのインストール

① パソコンの電源がＯＦＦであることを確認した後，ＰＰＤ５３ｘボードをパソコンのＰＣＩバススロットに装着します．パソコンの電源をＯＮにして Windows を起動します．
② Windows２０００が起動すると，ＰＰＤ５３ｘがシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．
③ システムがインストール元ディレクトリの指定を要求してきたら，添付のＣＤをＣＤドライブに挿入します．
④ □検索の場所の指定　のチェックボックスを必ずチェックします．
⑤ 「Ｈｉｖｅｒｔｅｃ　ＨＰＣＩ－ＰＰＤ５３０（ＰＣＩ）ＤｒｉｖｅｒＤｉｓｋ上の一部のファイルが必要です．」という画面が表示された場合は参照をクリックし，ＣＤドライブ：¥WIN2k¥hp530w2k.sys を指定して下さい．後はシステムの指示に従ってインストールを完了させます．

２．４　WindowsＸＰへのインストール

※１．WindowsＸＰ上でWindows2000 用デバイスドライバを使用されていた場合
以下の手順の前にデバイスドライバをアンインストール（P530uins.exe を実行）
してください.

① パソコンの電源がＯＦＦであることを確認した後，ＰＰＤ５３ｘボードをパソコンのＰＣＩバススロットに装着します．パソコンの電源をＯＮにしてWindows を起動します.
② WindowsＸＰが起動すると，ＰＰＤ５３ｘがシステムにより検出され，自動的に必要なデバイスドライバのインストール画面が表示されます．添付のＣＤをＣＤドライブに挿入します.
③ ソフトウェアを自動的にインストールする（推奨）をチェックします.
④ Ｈｉｖｅｒｔｅｃ　ＨＰＣＩ－ＰＰＤ５３Ｘ（ＷｉｎＸＰ）を選択します.
⑤ 「Windows ﾛｺﾞﾃｽﾄに合格していません」との警告が表示されることがありますが，「続行」を選択してインストールを続けてください．後はシステムの指示に従ってインストールを完了させます.

２．５　Windows Vista(32 ビット)，Windows 7(32 ビット)へのインストール

① ＰＰＤ５３ｘボードをパソコンのＰＣＩバススロットに装着する前に,パソコンの電源をＯＮにしてWindows を起動します.
② ＣＤドライブ：￥ｗｉｎ７＿ｘ８６￥ｄｐｉｎｓｔ．ｅｘｅを起動します.
③ “ｄｐｉｎｓｔ．ｅｘｅ”が起動されたら、「次へ」をクリックして続行します.
④ インストーラ完了後，パソコンの電源をＯＦＦし，ＨＰＣＩ－ＰＰＤ５３ＸをパソコンのＰＣＩバススロットに装着します.
⑤ パソコンの電源をＯＮにしてWindows を起動します.
⑥ デバイスのインストールが自動的に行われ，再起動を促されますので再起動してインストールが完了します.

２．６　Windows Vista(64 ビット)，Windows 7(64 ビット)へのインストール

① ＰＰＤ５３ｘボードをパソコンのＰＣＩバススロットに装着する前に,パソコンの電源をＯＮにしてWindows を起動します.
② ＣＤドライブ：￥ｗｉｎ７＿ｘ６４￥ｄｐｉｎｓｔ．ｅｘｅを起動します.
③ “ｄｐｉｎｓｔ．ｅｘｅ”が起動されたら、「次へ」をクリックして続行します.
④ インストーラ完了後，パソコンの電源をＯＦＦし，ＨＰＣＩ－ＰＰＤ５３ＸをパソコンのＰＣＩバススロットに装着します.
⑤ パソコンの電源をＯＮにしてWindows を起動します.
⑥ デバイスのインストールが自動的に行われ，再起動を促されますので再起動してインストールが完了します.

２．７　アンインストール

２．７．１　Windows 9x, Windows NT4.0, Windows 2000, Windows XP の場合

添付のＣＤをＣＤドライブに挿入します.
エクスプローラを起動し，ＣＤドライブ：¥P530uins.exe を実行します.

２．７．２　Windows Vista, Windows 7 の場合

Ｗｉｎｄｏｗｓの「スタート」→「コントロールパネル」→「プログラムのアンインストール」
→「Ｗｉｎｄｏｗｓドライバパッケージ Ｈｉｖｅｒｔｅｃ ＨＰＣＩ－ＰＰＤ５３Ｘ」を右クリックして
アンインストールを行います.

３　ボードを複数枚使用する場合

ＰＰＤ５５ｘボードを同一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に対応させたい場合について説明します.

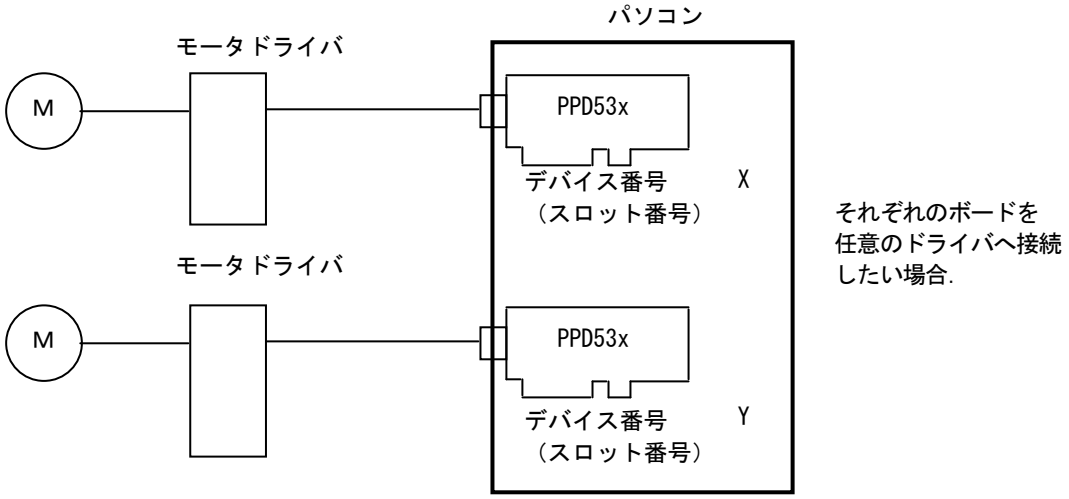


図３．１―１　ボードを複数枚使用する場合

３．１　ボードのスロット番号の確認

ＰＣＩバスのカードを装着するスロットには，それぞれのパソコンにより固定の番号が割り振られています.
その番号を知ることで，装着するスロットのボードと任意のモータドライバとの接続が可能になります.

３．２　スロット番号の確認方法

次の実行ファイルを起動することにより，現在スロットに装着されているデバイス番号（スロットの番号）を簡単に確認できます.

¥test¥tpp53400.exe　　・・　４軸以下のボードに添付（ PPD516A 以外）
¥test¥tpp51600.exe　　・・　５軸以上のボードに添付（ PPD516A 用　）

このプログラムは，ＤＬＬに対してデバイス情報の取得を行っています.
これにより，デバイス番号（スロット番号）の取得を行うことができます.
（詳細は，「５．関数詳細　(2)デバイス情報の取得」をご覧ください．）

このプログラムは，ＤＬＬに対してデバイス情報の取得を行っています.
これにより，デバイス番号（スロット番号）の取得を行うことができます.
（詳細は，「５．関数詳細　(2)デバイス情報の取得」をご覧ください．）

３．３　ボードＩＤ（ボードアドレス）の使用

ＰＰＤ５３ｘボードで次のボードはボード上のジャンパで設定したボードＩＤ（0～15）が使用できます.
この使用方法は後述します.

ボードＩＤ使用可能ボード・・ＰＰＤ５３２Ａ，ＰＰＤ５３３Ａ，ＰＰＤ５３４Ａ，ＰＰＤ５１４Ａ，ＰＰＤ５１６Ａ
ボードＩＤ固定(15)ボード・・ＰＰＤ５３２，ＰＰＤ５３３，ＰＰＤ５３４

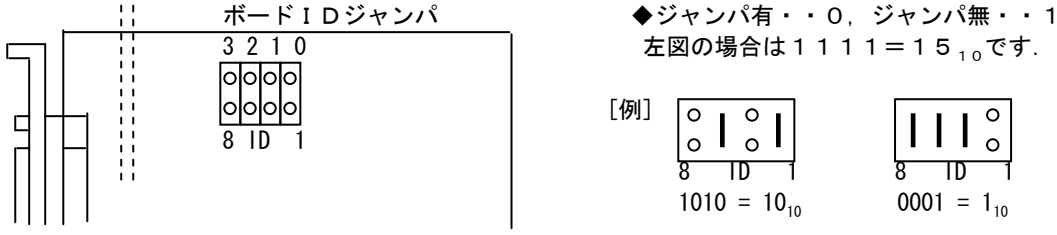


図３．３―１　ＰＰＤ５３Ｘボードの設定

４　ドライバＩ／Ｆ用ＤＬＬの使用方法

４．１　概　　要

Ｗｉｎｄｏｗｓ用ＤＬＬは，WindowsＶｉｓｔａ，WindowsＸＰ，Windows２Ｋ，WindowsＮＴ，Windows９Ｘにおいて，ＰＰＤ５３ｘボードの制御を行うための関数群です.

各関数は"Microsoft Visual C++(Ver6 以上)"（以下 VC++），
"Microsoft Visual Basic(Ver6.0)"　（以下 VB6）
"Microsoft Visual Basic .NET"　（以下 VB.NET）
から外部関数として起動されます.
アプリケーションプログラムがＤＬＬの関数を呼び出しを行い，ＤＬＬはＰＰＤ５３ｘのデバイスドライバにアクセスします.

デバイスドライバはWindows９Ｘ用に hippd530.vxd，WindowsＮＴ用に hippd530.sys，Windows２Ｋ用に hp530w2k.sys，WindowsＸＰ用に hp530wxp.sys，WindowsＶｉｓｔａ用に hp530wdm.sys が使用されます.

４．２　関数一覧

ドライバＩ／Ｆ用ＤＬＬは，次の１６種類の関数が含まれます.

No	関　数　名　称	機　　能	記載ﾍﾟｰｼﾞ
1	hppd530_GetDeviceCount()	ボード枚数の取得	１１
2	hppd530_GetDeviceInfo()	デバイス情報の取得	１２
3	hppd530_OpenDeviceID()	デバイスのオープン（ボードＩＤ参照）	１３
4	hppd530_OpenDevice()	デバイスのオープン（ボードＩＤ無視）	１４
5	hppd530_CloseDevice()	デバイスのクローズ	１５
6	hppd530_CmdWrite()	コマンドバッファ書込	１６
7	hppd530_DousaWrite()	動作モードバッファ書込	１８
8	hppd530_SeigyoWrite()	制御モードバッファ書込	１９
9	hppd530_status1Read()	ステータス１読込	２０
10	hppd530_status2Read()	ステータス２読込	２１
11	hppd530_intstatusRead()	割込ステータス読込	２２
12	hppd530_regRead()	レジスタ読込	２３
13	hppd530_ElstRead()	ＥＬＳ極性選択ポート読込	２９
14	hppd530_ElstWrite()	ＥＬＳ極性選択ポート書込	２９
15	hppd530_CmpcRead()	コンパレータ同時スタート読込	３０
16	hppd530_CmpcWrite()	コンパレータ同時スタート書込	３０

表４．２―１　関数一覧

４．３　準　　備

本ＤＬＬを使用する手順を説明します.

[ＶＣ++によるアプリケーションの構築　]

次のファイルをプロジェクトへ追加して下さい.

・hippd530.lib　・・　インポートライブラリ
・hippd530.h　　・・　Ｃアプリケーション構築用ヘッダーファイル

（注）１．ＤＬＬ関数はＣ言語で作成されています.
２．ヘッダーファイル中のＤＬＬ関数プロトタイプ宣言は次のように記述されています.

```
// ------------------------------------
// DLL 関数プロとタイプ宣言
// ------------------------------------
#ifdef __cplusplus
extern "C"
{
#endif
    ＤＬＬ関数のプロトタイプ宣言
#ifdef __cplusplus
}
#endif
```

これは，アプリケーションをC++コーディング（ファイル拡張子=ｃｐｐ）で作成する場合に備えての処理です.

３．「#ifdef　__cplusplus」の定義は"ＶＣ++"用です.
他言語で使用する場合には，明示的な宣言に変更できます.

```
（例）
     #define  CPLUS  1          "1"でC++ 用（ﾌｧｲﾙ拡張子：ｃｐｐ）
        ..........              "0"でＣ　 用（ﾌｧｲﾙ拡張子：ｃ　 ）
     #if      CPLUS
        ..........
     #endif
```

[ＶＢ６によるアプリケーションの構築]

標準モジュール「hippd530.bas」をプロジェクトに，追加してください.
このファイルに外部関数宣言（Declare 宣言）及びユーザー定義型宣言が記述されています.

※ 注意事項
①添付ソフトウエアは，Intel 互換ＣＰＵを搭載したマシン以外のプラットフォームには対応しておりません.
②ボードＩＤを使用する為，制御するＰＰＤ５３ｘボードは１６枚としています.

４．４　制御概念

ＤＬＬ及びデバイスドライバは複数のＰＰＤ５３Ｘボードを制御することができます.
ある１つのＰＰＤ５３Ｘ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，アクセスするための足がかりとなるデバイスＩＤ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．（ハードウェアリソースすなわち I/O ポートアドレスや IRQ 番号等は，Windows のシステム側によって確定されます．）

４．４．１　ボード（デバイス）認識用のデータ構造体

ボード認識のために次に示す HPCDEVICEINFO 型構造体を用意します.

[Ｃ言語：Windows: VC++]

```
typedef  struct  _HPCDEVICEINFO {
    DWORD      nBusNumber;          /* バス番号 */
    DWORD      nDeviceNumber;       /* デバイス番号（PCIｽﾛｯﾄ番号）*/
    DWORD      dwIoPortAddress;     /* I/O ポートアドレス */
    DWORD      dwIrqNo;             /* IRQ 番号 */
    DWORD      dwNumber;            /* 管理番号 */
    DWORD      dwBoardID;           /* ボードＩＤ(0～15) */
}  HPCDEVICEINFO,  *PHPCDEVICEINFO
```

[Windows：VB6]

```
Public  Type  HPCDEVICEINFO
    nBusNumber      As Long          'バス番号
    nDeviceNumber   As Long          'デバイス番号
    dwIoPortAddress As Long          'I/O ポートアドレス
    dwIrqNo         As Long          'IRQ 番号
    dwNumber        As Long          '管理番号
    dwBoardID       As Long          'ボードＩＤ(0～15)
End  Type
```

（注）１．管理番号はWindows９Ｘでは使用されません.
　　常に「INVALID_HPC_NUMBER(-1)」が格納されています.
　　２．ボードＩＤの使用できるボードと出来ないボード（１５固定）があります.
　　ご注意下さい.

４．４．２　ボードアクセスの準備手順

（１）使用する全ボードのデバイス情報の取得

"HPCDEVICEINFO"型構造体エリア（の配列）内に，全ＰＰＤ５３Ｘのデバイス情報をまず取得します.

◆ hppd530_GetDeviceCount()・・・ボード枚数の確認

◆ hppd530_GetDeviceInfo() ・・・全ボードのデバイス情報を取得

（２）ボード毎にデバイスオープン

ある１つのＰＰＤ５３Ｘのデバイス情報をデバイスオープン関数に渡します.

この結果そのＰＰＤ５３Ｘがオープンされ，デバイスオープン関数はこのボードにアクセスするためのデバイスＩＤ値を返してきます.

ボード枚数が２枚以上の場合には，個々のボード毎にこの処理を行います.

◆ hppd530_OpenDeviceID() ・・・ボードのオープン処理［ボードＩＤ参照］

◆ hppd530_OpenDevice() ・・・ボードのオープン処理［ボードＩＤ無視］

（３）ボードの信号処理方法の初期設定

以降は，この「デバイスＩＤをハンドル」として，そのＰＰＤ５３Ｘにアクセスすることができるようになります.

ボードの初期化を行う前に，次の関数でデバイス使用条件の設定を行います.

◆ hppd530_ElstWrite() ・・・ＥＬＳ極性の設定（極限リミットセンサ）

◆ hppd530_CmpcWrite() ・・・コンパレータ同時スタート設定

（４）各ボード・各軸の初期化

上記設定以降に，使用する全ボードの各軸の初期化を行います.

これにより，通常動作としての各軸パルス出力動作等が可能となります.

なお，汎用入出力をご使用になる場合，入出力数の設定と出力ポートへの初期出力を行います.

≪　お願い　≫

◆各軸をモータ動作可能状態に接続した時，次の確認を行って下さい.

・±ELS 信号の作動試験（モータ停止状態でセンサのみ作動させます．）

・サーボアラーム信号を接続した時の信号入力状態.

・原点信号（OLS･Ｚ相）の入力状態.

上記信号が正しく入力されない時，正常な動作が保証されません.

◆モータへの指令パルス出力で正常に作動しない時，次の確認が必要です.

・指令パルス出力設定は"サーボドライバ"入力と一致していますか.

・"サーボドライバ"入力信号にモータを停止させる要因がありますか.

（５）全ての処理が終了してアプリケーションを終了する場合には，オープンしたデバイスの「クローズ処理」を行って下さい.

◆ hppd530_CloseDevice() ・・・ボードのクローズ処理（１枚分）

４．４．３　デバイスドライバの異常報告

デバイスドライバの諸関数を使用する時，関数の戻り値が異常値（０）であった場合には，この異常内容を読込み，異常内容に対応した処理を行います.

（１）異常内容の読込・・・GetLastError() 関数の起動

```
[ Ｃ言語: VC++ ]                    [  VB6  ]
     DWORD   dErrorCode;                dErrorCode  As Long
     dErrorCode = GetLastError();       dErrorCode = GetLastError()
```

（２）異常内容の一覧

No	読 出 値 (値：Ｃ言語１６進数表記)	異　常　内　容
1	NO_ERROR (0x00000000)	異常なし（デバイスなし）
2	ERROR_FILE_NOT_FOUND (0x00000002)	デバイスドライバが存在しない
3	ERROR_NOT_ENOUGH_MEMORY (0x00000008)	デバイス情報格納メモリが不足
4	ERROR_HPC_ALREADY_OPENED(0x20000001)	既にオープン済のデバイスをオープン
5	ERROR_HPC_ILLEGAL_DEVICE(0x20001000)	HPCI ボードと認められない
6	ERROR_INVALID_PARAMETER (0x00000087)	指定デバイス情報に該当するボード無し
7	ERROR_HPC_INVALID_HANDLE(0x20000002)	無効なデバイス ID を指定
8	ERROR_NOT_READY (0x00000021)	デバイスの入出力ポートが使用できない

表４．４―１　異常内容一覧

（３）異常発生時の確認項目

① NO_ERROR ・・・・・・・・・・◎異常は発生していません.
② ERROR_FILE_NOT_FOUND ・・・◎デバイスドライバがインストールされていません.
　　・◎デバイスドライバが所定のフォルダに格納されていません.
③ ERROR_NOT_ENOUGH_MEMORY・・◎アプリケーション用のメモリ不足です.
　　◇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）の不足です.
　　◇多数のアプリケーション起動.
　　◇１度に多数のウインドウを開いた.
④ ERROR_HPC_ALREADY_OPENED・ ◎オープン済みデバイスに更にオープン指令を行いました.
　　◇オープンしたデバイスはクローズするまで使用できます.
　　（多重のオープンは禁止）
　　◎ボード２枚以上使用する場合，オープンするデバイス
　　情報の更新を確認して下さい.
⑤ ERROR_HPC_ILLEGAL_DEVICE ・◎デバイス情報は正常ですが，ボード機能が一致していません.
⑥ ERROR_INVALID_PARAMETER・・◎デバイスオープン関数で指定したデバイス情報の内容を確認して下さい.
⑦ ERROR_HPC_INVALID_HANDLE ・◎デバイスオープンで得られた"デバイスＩＤ"ではありません.
　　◎このデバイスは既にクローズされています.
⑧ ERROR_NOT_READY ・・・・・ ◎デバイス（ボード）内部の入出力ポートがありません.
　　（ＯＳが不安定になっている可能性が考えられます．）

５．関数詳細

関数説明文中で１６進数の表記はＣ言語記述で行っています.
ＶＢ でご利用の場合には, Ｃ言語１６進数表記を次のように読み替えて下さい.

Ｃ言語	ＶＢ６．０, ＶＢ．ＮＥＴ
0x00000000	&H0
0x20001000	&H20001000

（1）	**hppd530_GetDeviceCount()　　ボード枚数の取得**
機能	現在パソコンに装着されているＰＰＤ５３Ｘボードの枚数を取得します. PPD552～PPD530,PPD552A～PPD530Aﾎﾞｰﾄﾞの全枚数です.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_GetDeviceCount(WORD dummy); [ＶＢ６] Declare Function hppd530_GetDeviceCount Lib "hippd530.DLL" _ (ByVal dummy As Integer)　As Long
引数	◆ DWORD dummy　・・　実際には使用していません. （任意の値を設定）
戻り値	ＰＰＤ５３Ｘボードの枚数 枚数の値が１以上　・・　実装枚数との一致を確認して下さい. 枚数の値が０の時　・・　「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい. この結果は次のようになります. ◇ NO_ERROR　(0x00000000)・・・・・・ＰＰＤ５３Ｘボードは１枚もなし. ◇ ERROR_FILE_NOT_FOUND (0x00000002)・・・・・デバイスドライバが存在しない.
呼び出し例	[Ｃ言語: VC++] DWORD　count; count = hppd530_GetDeviceCount(1); [ＶＢ６] Dim count　　As Long count = hppd530_GetDeviceCount(1)

（2）	**hppd530_GetDeviceInfo()　　デバイス情報の取得**
機能	現在パソコンに装着されているＰＰＤ５３Ｘボードのデバイス情報を取得します. この結果, HPCDEVICEINFO 型の配列にデバイス情報が格納されます. この値は, デバイスオープン時に利用します.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_GetDeviceInfo (DWORD* pcnDeviceNumber, PHPCDEVICEINFO pHpcDeviceInfo); [ＶＢ６] Declare Function hppd530_GetDeviceInfo Lib "hippd530.DLL" _ (ByRef pcnDeviceNumber As Long,pHpcDeviceInfo As HPCDEVICEINFO) As Long
引数	◆ DWORD*　pcnDeviceNumber 情報を取得するボードの最大枚数が格納された DWORD 型エリアのアドレスを渡します. 関数の呼び出し後, 実際に情報を取得したボードの枚数が格納されます. ◆ PHPCDEVICEINFO　pHpcDeviceInfo 各ボードのデバイス情報がセットされるべきエリアのアドレス, すなわち HPCDEVICEINFO 型の配列の先頭アドレスを渡します.
戻り値	処理結果 １：成功　・・　指令と戻りの枚数値を確認して下さい. ０：失敗　・・　「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.

呼び出し例	パソコンにＰＰＤ５３Ｘが２枚装着されていることを想定します.
	
	[Ｃ言語: VC++]
	DWORD　　ret;　　//関数の戻り値
	DWORD　　count = 2;　　//最大枚数は２
	HPCDEVICEINFO　　HpcDeviceInfo[2];　　//２枚のＰＰＤ５３Ｘのデバイス情報が
	//セットされるべきエリア
	ret = hppd530_GetDeviceInfo(
	&count,　　//count のアドレスを渡す.
	&HpcDeviceInfo[0]);　　//配列の先頭アドレスを渡す.
	
	[ＶＢ６]
	Dim ret　　As Long
	Dim count　　As Long
	Dim HpcDeviceInfo(2)　As HPCDEVICEINFO
	count = 2
	ret = hppd530_GetDeviceInfo(
	count,　　'count のアドレスを渡す.
	HpcDeviceInfo(0))　　'配列の先頭アドレスを渡す.

（3）	**hppd530_OpenDeviceID()　　デバイスのオープン（ボードＩＤ参照）**
機能	渡した情報を持つＰＰＤ５３Ｘボードをオープンし，他と識別するためのデバイス ID を取得します． 以降このデバイス ID は，このＰＰＤ５３Ｘにアクセスするためのハンドルとなります．
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_OpenDeviceID(PHPCDEVICEINFO　pHpcDeviceInfo); [ＶＢ６] Declare Function hppd530_OpenDeviceID Lib "hippd530.DLL" _ (pHpcDeviceInfo As HPCDEVICEINFO) As Long
引数	◆ PHPCDEVICEINFO　pHpcDeviceInfo オープンするデバイスの情報がセットされたエリアのアドレスを渡します．
戻り値	デバイスＩＤ値 INVALID_HANDLE_VALUE（＝-1）：オープン失敗 「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい． 上記以外：オープン成功 ●上位ワード：ボ ー ドＩＤ（ボード上のジャンパ設定値：０～１５） ●下位ワード：デバイスＩＤ（オープン順の番号　　：１～　） 【ヒント】　上位ワードの"ボードＩＤ"をデバイスＩＤとする事ができます． この場合には，下位ワードを強制的に"０"とします． VC++ [hDeviceID &= 0xffff0000;] VB　[hDeviceID = hDeviceID And &HFFFF0000]
呼び出し例	パソコンにＰＰＤ５３Ｘが２枚装着されていることを想定します． デバイス情報格納エリアとして HPCDEVICEINFO 型の配列 HpcDeviceInfo[2]を準備し，この中には既に hppd530_GetDeviceInfo 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． [Ｃ言語: VC++] DWORD　　hDeviceID[2];　　//デバイス ID 取得エリア hDeviceID[0] = hppd530_OpenDeviceID(&HpcDeviceInfo[0]);　//１番目のデバイス情報 hDeviceID[1] = hppd530_OpenDeviceID(&HpcDeviceInfo[1]);　//２番目のデバイス情報 [ＶＢ６] Dim hDeviceID(2)　As Long hDeviceID(0) = hppd530_OpenDeviceID(HpcDeviceInfo(0))　　'１番目のデバイス情報 hDeviceID(1) = hppd530_OpenDeviceID(HpcDeviceInfo(1))　　'２番目のデバイス情報

（4）	**hppd530_OpenDevice()　　デバイスのオープン（ボードＩＤ無視）**
機能	渡した情報を持つＰＰＤ５３Ｘボードをオープンし，他と識別するためのデバイス ID を取得します. 以降このデバイス ID は，このＰＰＤ５３Ｘにアクセスするためのハンドルとなります.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_OpenDevice(PHPCDEVICEINFO　pHpcDeviceInfo); [ＶＢ６] Declare Function hppd530_OpenDevice Lib "hippd530.DLL" _ (pHpcDeviceInfo As HPCDEVICEINFO) As Long
引数	◆ PHPCDEVICEINFO　pHpcDeviceInfo オープンするデバイスの情報がセットされたエリアのアドレスを渡します.
戻り値	デバイスＩＤ値 INVALID_HANDLE_VALUE（＝-1）：オープン失敗 「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい. 上記以外：オープン成功 ●上位ワード：常に０ ●下位ワード：デバイスＩＤ（オープン順の番号　　：１～　　）
呼び出し例	パソコンにＰＰＤ５３Ｘが２枚装着されていることを想定します. デバイス情報格納エリアとして HPCDEVICEINFO 型の配列 HpcDeviceInfo[2]を準備し，この中には既に hppd530_GetDeviceInfo 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. [Ｃ言語: VC++] DWORD　　hDeviceID[2];　　//デバイス ID 取得エリア hDeviceID[0] = hppd530_OpenDevice(&HpcDeviceInfo[0]);　//１番目のデバイス情報 hDeviceID[1] = hppd530_OpenDevice(&HpcDeviceInfo[1]);　//２番目のデバイス情報 [ＶＢ６] Dim hDeviceID(2)　As Long hDeviceID(0) = hppd530_OpenDevice(HpcDeviceInfo(0))　　’１番目のデバイス情報 hDeviceID(1) = hppd530_OpenDevice(HpcDeviceInfo(1))　　’２番目のデバイス情報

（５）	**hppd530_CloseDevice()　　デバイスのクローズ**
機能	渡したデバイス ID で指定されたＰＰＤ５３Ｘボードをクローズします. 以降, このデバイス ID は無効となります.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_CloseDevice(DWORD　hDevID); [ＶＢ６] Declare Function hppd530_CloseDevice Lib "hippd530.DLL" _ (ByVal hDevID As Long) As Long
引数	◆ DWORD　hDevID　・・　クローズするボードのデバイス ID.
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.
呼び出し例	[Ｃ言語: VC++] DWORD ret;　//関数の戻り値 ret = hppd530_CloseDevice(hDeviceID);　//デバイス ID [ＶＢ６] Dim ret As Long ret = hppd530_CloseDevice(hDeviceID)　'デバイス ID

（6）	**hppd530_CmdWrite()　　コマンドバッファ書込**
機能	①デバイス ID で指定されたＰＰＤ５３Ｘの，AxisNo で指定された軸のコマンドバッファへコマンドデータを書込みます. ②PPD55x の各レジスタへの書込を行います．レジスタ書込コマンドの詳細については，ユーザーズマニュアルをご覧ください.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_CmdWrite(DWORD hDevID, WORD AxisNo, BYTE byData, WORD sw, DWORD dwReg); [ＶＢ６] Declare Function hppd530_CmdWrite Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, _ ByVal byData As Byte, ByVal sw As Integer, _ ByVal dwReg As Long) As Long
引数	◆ DWORD hDevID・・対象デバイスのデバイス ID. ◆ WORD AxisNo・・軸指定 [０：Ｘ軸, １：Ｙ軸, ２：Ｚ軸, ３：Ｕ軸, ４：Ｖ軸, ５：Ｗ軸] ◆ BYTE byData・・コマンドデータ 「表５．１－１　書込レジスタ不使用のコマンドデータ：　sw = ０」, 「表５．１－２　書込レジスタとコマンドデータ：　　sw = １」参照 ◆ WORD sw ＝０：レジスタ書込をしない. ≠０：レジスタ書込を行う. ◆ DWORD dwReg ・・ レジスタへの書込データ
戻り値	処理結果 １：成功 ０：失敗 ・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.

呼び出し例	[Ｃ言語: VC++]
	DWORD ret; //関数の戻り値
	ret = hppd530_CmdWrite(hDeviceID, //デバイス ID
	0, //Ｘ軸を指定
	0x60, //（これは「ｿﾌﾄﾘｾｯﾄ」ｺﾏﾝﾄﾞ）
	0, //ﾚｼﾞｽﾀ書込ではない.
	0);
	
	[ＶＢ６]
	Dim ret As Long '関数の戻り値
	ret = hppd530_CmdWrite(hDeviceID, 'デバイス ID
	0, 'Ｘ軸を指定
	&h60, '（これは「ｿﾌﾄﾘｾｯﾄ」ｺﾏﾝﾄﾞ）
	0, 'ﾚｼﾞｽﾀ書込ではない.
	0)

ｺﾏﾝﾄﾞﾃﾞｰﾀ	コマンド内容	ｺﾏﾝﾄﾞﾃﾞｰﾀ	コマンド内容
0x58	SVON　ON	0x00	瞬時にＲ１速度に変更
0x48	SVON　OFF	0x01	瞬時にＲ２速度に変更
0x59	SVRESET ON	0x02	減速してＲ１速度に変更
0x49	SVRESET OFF	0x03	加速してＲ２速度に変更
0x10	ＦＬ定速スタート	0x20	ＦＬ定速スタート保留
0x11	ＦＨ定速スタート	0x21	ＦＨ定速スタート保留
0x13	高速スタート	0x23	高速スタート保留
0x14	残量ＦＬスタート	0x24	残量ＦＬスタート保留
0x15	残量ＦＨスタート	0x25	残量ＦＨスタート保留
0x17	残量高速スタート	0x27	残量高速スタート保留
0x30	同時スタート	0x28	同時停止
0x09	即停止	0x0a	減速停止
0x60	ソフトウエア リセット	0x63	非常停止
0x61	UP/DOWN カウンタ リセット	0x66	プリレジスタ確定セット
0x62	脱調検出偏差ｶｳﾝﾀ ﾘｾｯﾄ	0x67	プリレジスタ確定リセット

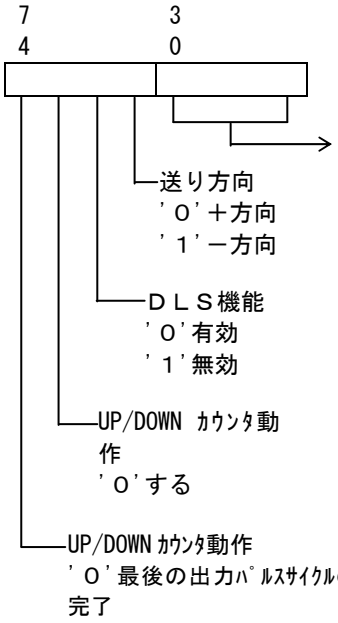
表５．１－１　書込レジスタ不使用のコマンドデータ：　sw = 0

書込レジスタ	ｺﾏﾝﾄﾞﾃﾞｰﾀ	レジスタ内容	ﾚｼﾞｽﾀﾊﾞｲﾄ数
ＰＲ０	0xc0	送り量	４
ＰＲ１	0xc1	ＦＬ（ＰＲ１×倍率（ＰＲ４））速度	２
ＰＲ２	0xc2	ＦＨ（ＰＲ２×倍率（ＰＲ４））速度	２
ＰＲ３	0xc3	加減速レート	２
ＰＲ４	0xc4	倍率	２
ＰＲ５	0xc5	減速点	３
Ｒ６	0xc6	環境レジスタ１	４
Ｒ７	0xc7	環境レジスタ２	４
Ｒ８	0xc8	環境レジスタ３	２
Ｒ９	0xc9	アップ／ダウンカウント値	４
Ｒ１０	0xca	コンパレータ１データ	４
Ｒ１１	0xcb	コンパレータ２データ	４
Ｒ１	0xd1	動作中のＦＬ（Ｒ１×倍率）速度変更	２
Ｒ２	0xd2	動作中のＦＨ（Ｒ２×倍率）速度変更	２
Ｒ３	0xd3	動作中の加減速レート変更	２
Ｒ５	0xd4	動作中のスローダウンポイント変更	３
ＰＲ１５	0xd8	減速レート	２
ＰＲ１６	0xd9	Ｓ字区間	２
Ｒ１５	0xda	動作中の減速レート変更	２
Ｒ１６	0xdb	動作中のＳ字区間変更	２

表５．１－２　書込レジスタとコマンドデータ：　sw = 1

（7）	**hppd530_DousaWrite()　　動作モードバッファ書込**
機能	デバイス ID で指定されたＰＰＤ５３Ｘの, AxisNo で指定された軸の動作モードバッファへ動作モードデータを書込みます.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_DousaWrite(DWORD hDevID, WORD AxisNo, BYTE byData); [ＶＢ６] Declare Function hppd530_DousaWrite Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, ByVal byData As Byte) As Long
引数	◆ DWORD　hDevID・・対象デバイスのデバイス ID ◆ WORD　AxisNo・・軸指定 [0：Ｘ軸, 1：Ｙ軸, 2：Ｚ軸, 3：Ｕ軸, 4：Ｖ軸, 5：Ｗ軸] ◆ BYTE　byData・・動作モードデータ　（次ページの「表５．１－３　動作モードデータ」参照）
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.

呼び出し例	[Ｃ言語: VC++]
	DWORD　　ret;　　//関数の戻り値
	ret = hppd530_DousaWrite(hDeviceID, //デバイス ID
	0,　　//Ｘ軸を指定
	0x20);　　//（動作モードデータ参照）
	[ＶＢ６]
	Dim ret　As Long　　'関数の戻り値
	ret = hppd530_DousaWrite(hDeviceID,　'デバイス ID
	0,　　'Ｘ軸を指定
	&H20)　　'（動作モードデータ参照）



b3 - b0	動 作 モ ー ド
００００	連続モード１　（手動送り）
０００１	〃　２　（ﾊﾝﾄﾞﾙ手動送り）
００１０	原点復帰モード１
００１１	〃　２
０１００	原点サーチモード
０１０１	原点抜出しモード
０１１０	１パルス出力モード
０１１１	タイマモード　（ドウェル）
１０００	プリセットモード１（方向と送り量）
１００１	〃　２（相対値送り）
１０１０	〃　３（絶対値送り）
１０１１	浮動原点復帰モード
１１００	プリセットモード４（ﾊﾝﾄﾞﾙ同期送り）

表５．１－３　動作モードデータ

（8）	**hppd530_SeigyoWrite()　　制御モードバッファ書込**
機能	デバイス ID で指定されたＰＰＤ５３Ｘの, AxisNo で指定された軸の制御モードバッファへ制御モードデータを書込みます.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_SeigyoWrite(DWORD hDevID, WORD AxisNo, BYTE byData); [ＶＢ６] Declare Function hppd530_SeigyoWrite Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, ByVal byData As Byte) As Long
引数	◆ DWORD　hDevID・・対象デバイスのデバイス ID. ◆ WORD　AxisNo・・軸指定 [0：Ｘ軸, 1：Ｙ軸, 2：Ｚ軸, 3：Ｕ軸, 4：Ｖ軸, 5：Ｗ軸] ◆ BYTE　byData・・制御モードデータ（「表５．１－４　制御モードデータ」参照）
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.

呼び出し例	[Ｃ言語: VC++]
	DWORD ret; //関数の戻り値
	ret = hppd530_SeigyoWrite(hDeviceID, //デバイス ID
	1, //Ｙ軸を指定
	0x04); //（制御モードデータ参照）
	[ＶＢ６]
	Dim ret As Long '関数の戻り値
	ret = hppd530_SeigyoWrite(hDeviceID, 'デバイス ID
	1, 'Ｙ軸を指定
	&H4) '（制御モードデータ参照）

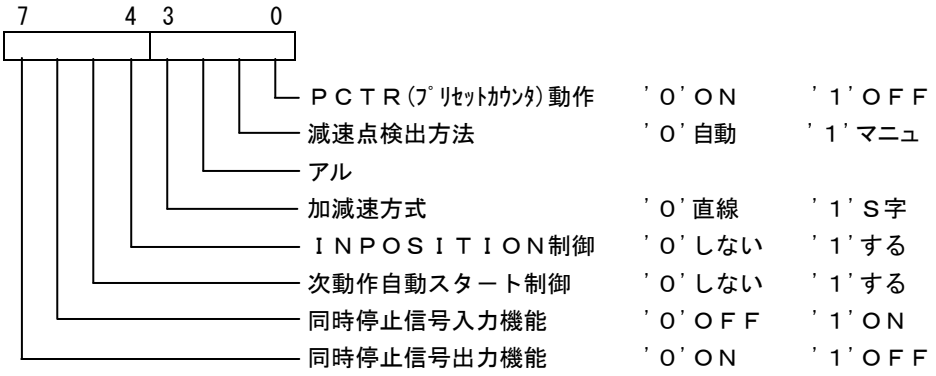
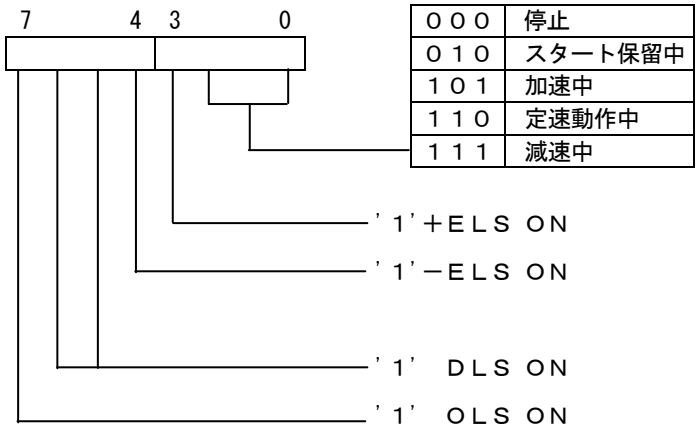


表５．１－４　制御モードデータ

（9）	**hppd530_status1Read()　　ステータス１読込**
機能	デバイス ID で指定されたＰＰＤ５３Ｘの, AxisNo で指定された軸のステータス１を読込み, 指定した１バイトのエリアに格納します.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_status1Read(DWORD hDevID, WORD AxisNo, BYTE* bySts1); [ＶＢ６] Declare Function hppd530_status1Read Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, ByRef bySts1 As Byte) As Long
引数	◆ DWORD hDevID・・対象デバイスのデバイス ID. ◆ WORD AxisNo・・軸指定 [0：Ｘ軸, 1：Ｙ軸, 2：Ｚ軸, 3：Ｕ軸, 4：Ｖ軸, 5：Ｗ軸] ◆ BYTE* bySts1・・読み出されたステータス１の格納先 （「表５．１－５　ステータス１」参照）
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.

呼び出し例	[Ｃ言語: VC++]
	DWORD ret; //関数の戻り値
	BYTE sts1;
	ret = hppd530_status1Read(hDeviceID, //デバイス ID
	1, //Ｙ軸を指定
	&sts1); //格納先のアドレス
	
	[ＶＢ６]
	Dim ret As Long '関数の戻り値
	Dim sts1 As Byte
	ret = hppd530_status1Read(hDeviceID, 'デバイス ID
	1, 'Ｙ軸を指定
	sts1) '格納先のアドレス



０００	停止
０１０	スタート保留中
１０１	加速中
１１０	定速動作中
１１１	減速中

表５．１－５　ステータス１

（１０）	**hppd530_status2Read()　　ステータス２読込**
機能	デバイス ID で指定されたＰＰＤ５３Ｘの, AxisNo で指定された軸のステータス２を読込み, 指定した１バイトのエリアに格納します.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_status2Read(DWORD hDevID, WORD AxisNo, BYTE* bySts2); [ＶＢ６] Declare Function hppd530_status2Read Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, ByRef bySts2 As Byte) As Long
引数	◆ DWORD　hDevID・・対象デバイスのデバイス ID. ◆ WORD　 AxisNo・・軸指定 [0：Ｘ軸, 1：Ｙ軸, 2：Ｚ軸, 3：Ｕ軸, 4：Ｖ軸, 5：Ｗ軸] ◆ BYTE*　bySts2・・読み出されたステータス１の格納先 （「表５．１－６　ステータス２」参照）
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.
呼び出し例	[Ｃ言語: VC++] DWORD ret;　//関数の戻り値 BYTE　sts2; ret = hppd530_status1Read(hDeviceID,　//デバイス ID 1,　　//Ｙ軸を指定 &sts2);　　//格納先のアドレス [ＶＢ６] Dim ret　As Long　　'関数の戻り値 Dim sts2　As Byte ret = hppd530_status1Read(hDeviceID,　　'デバイス ID 1,　　'Ｙ軸を指定 sts2)　　'格納先のアドレス

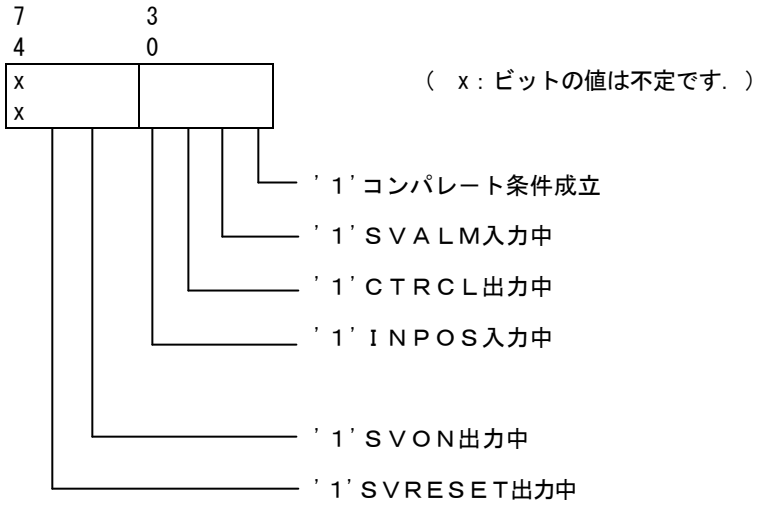


表５．１－６　ステータス２

（１１）	hppd530_intstatusRead()　　割込ステータス読込
機能	デバイス ID で指定されたＰＰＤ５３Ｘの, AxisNo で指定された軸の割込ステータスを読込み, 指定した１バイトのエリアに格納します.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_intstatusRead(DWORD hDevID, WORD AxisNo, BYTE* byIntsts); [ＶＢ６] Declare Function hppd530_intstatusRead Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, ByRef byIntsts As Byte) As Long
引数	◆ DWORD hDevID・・・対象デバイスのデバイス ID. ◆ WORD AxisNo・・・軸指定 [0：Ｘ軸, 1：Ｙ軸, 2：Ｚ軸, 3：Ｕ軸, 4：Ｖ軸, 5：Ｗ軸] ◆ BYTE* byIntsts・・読込んだ割込ステータスデータが格納される１バイトエリアのアドレス （「表５．１－７　割込ステータス」参照）
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.
呼び出し例	[Ｃ言語: VC++] DWORD ret;　//関数の戻り値 BYTE intsts; ret = hppd530_intstatusRead(hDeviceID,　//デバイス ID 1,　//Ｙ軸を指定 &intsts);　//格納先のアドレス [ＶＢ６] Dim ret As Long　'関数の戻り値 Dim intsts As Byte ret = hppd530_intstatusRead(hDeviceID,　'デバイス ID 1,　'Ｙ軸を指定 intsts)　'格納先のアドレス

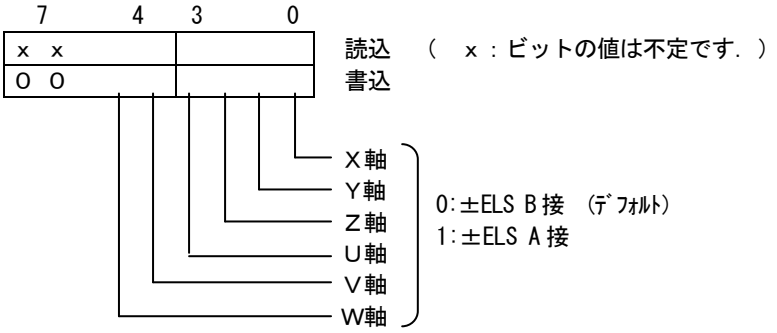
ステータス	要　　因	割込有効　（マスクﾋﾞｯﾄ）
０ｘ００	割込要因なし	
０ｘ０１	減速停止コマンﾄﾞ（０Ａｈ）書込みによる停止	Ｒ８　ｂ０＝’１’
０ｘ０２	位置決め動作完了による停止	Ｒ８　ｂ１＝’１’
０ｘ０３	原点復帰（原点サーチ）動作完了による停止	Ｒ８　ｂ２＝’１’
０ｘ０４	原点抜け出し動作完了による停止	Ｒ８　ｂ２＝’１’
０ｘ０５	即停止コマンド（０９ｈ）書込みによる停止	Ｒ８　ｂ０＝’１’
０ｘ０６	同時停止コマンドによる停止	制御モードﾞのｂ５＝’１’
０ｘ０７	－ＥＬＳ信号ＯＮによる停止	
０ｘ０８	＋ＥＬＳ信号ＯＮによる停止	
０ｘ０９	－ＤＬＳ信号ＯＮによる減速停止	Ｒ６　ｂ２９＝’１’
０ｘ０ａ	＋ＤＬＳ信号ＯＮによる減速停止	Ｒ６　ｂ２９＝’１’
０ｘ０ｂ	ＳＶＡＬＭ信号ＯＮによる停止	
０ｘ０ｃ	脱調検出による停止	Ｒ８　ｂ１２～ｂ８▫０
０ｘ１１	現在位置エンコーダ入力エラー	Ｒ６　ｂ２７＝’１’
０ｘ１２	手動パルスエンコーダ入力エラー	連続モードﾞ２，ﾌﾟリセットモードﾞ４
０ｘ１３	脱調検出エンコーダ入力エラー	Ｒ８　ｂ１２～ｂ８▫０
０ｘ１４	次動作スタート（プリレジスタ変更可）	Ｒ８　ｂ１４＝’１’
０ｘ１５	減速開始	Ｒ８　ｂ３＝’１’
０ｘ１６	コンパレート条件が偽→真に変化	Ｒ８　ｂ１３＝’１’

表５．１－７　割込ステータス

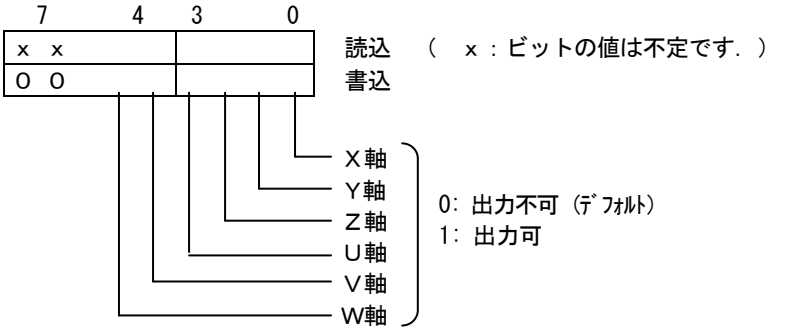
（１２）	**hppd530_regRead()　　レジスタ読込**
機能	デバイス ID で指定されたＰＰＤ５３Ｘの，AxisNo で指定された軸のレジスタ内容を読み込み，指定した４バイトのエリアに格納します. byData に「レジスタ読込コマンドの値」をセットします. レジスタ内容の詳細については，ユーザーズマニュアルをご覧ください.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_regRead(DWORD hDevID, WORD AxisNo, BYTE byData, DWORD* dwReg); [ＶＢ６] Declare Function hppd530_regRead Lib "hippd530.DLL" _ (ByVal hDevID As Long, ByVal AxisNo As Integer, ByVal byData As Byte, ByRef dwReg As Long) As Long
引数	◆ DWORD　hDevID・・対象デバイスのデバイス ID. ◆ WORD　 AxisNo・・軸指定 [0：Ｘ軸，1：Ｙ軸，2：Ｚ軸，3：Ｕ軸，4：Ｖ軸，5：Ｗ軸] ◆ BYTE　 byData・・レジスタ読込コマンドの値（「表５．１－８　レジスタ読込コマンド」参照） ◆ DWORD* dwReg ・・読込んだデータが格納される４バイトエリアのアドレス
戻り値	処理結果 １：成功 ０：失敗　・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.
呼び出し例	[Ｃ言語: VC++] DWORD　ret;　//関数の戻り値 DWORD　regstr; ret = hppd530_regRead(hDeviceID,　//デバイス ID 0,　//Ｘ軸を指定 0x80,　//R0 の内容を読む &regstr);　//格納先のアドレス [ＶＢ６] Dim ret　　As Long　　'関数の戻り値 Dim regstr　As Long ret = hppd530_regRead(hDeviceID, _　'デバイス ID 0, _　'Ｘ軸を指定 &h80, _　'R0 の内容を読む regstr)　'格納先のアドレス

読出レジスタ	ﾚｼﾞｽﾀ読込ｺﾏﾝﾄﾞ	レジスタ内容
Ｒ０	０ｘ８０	送り量
Ｒ１	０ｘ８１	ＦＬ（Ｒ１×倍率）速度
Ｒ２	０ｘ８２	ＦＨ（Ｒ２×倍率）速度
Ｒ３	０ｘ８３	加減速レート
Ｒ４	０ｘ８４	倍率
Ｒ５	０ｘ８５	減速点
Ｒ６	０ｘ８６	環境レジスタ１
Ｒ７	０ｘ８７	環境レジスタ２
Ｒ８	０ｘ８８	環境レジスタ３
Ｒ９	０ｘ８９	アップ／ダウンカウンタ
Ｒ１０	０ｘ８ａ	コンパレータ１データ
Ｒ１１	０ｘ８ｂ	コンパレータ２データ
Ｒ１２	０ｘ８ｃ	カウンタモニタ
Ｒ１３	０ｘ８ｄ	コマンドモニタ１
Ｒ１４	０ｘ８ｅ	コマンドモニタ２
ＰＲ０	０ｘ９０	次動作用プリセット量
ＰＲ１	０ｘ９１	次動作用ＦＬ（ＰＲ１×倍率）速度
ＰＲ２	０ｘ９２	次動作用ＦＨ（ＰＲ２×倍率）速度
ＰＲ３	０ｘ９３	次動作用加減速レート
ＰＲ４	０ｘ９４	次動作用倍率
ＰＣＴＲ	０ｘ９５	プリセットカウンタ　カウント値
ＳＣＴＲ	０ｘ９６	減速点カウンタ　カウント値
ＰＲ５	０ｘ９７	次動作用スローダウンポインタ
ＰＲ１５	０ｘ９８	次動作用減速レート
ＰＲ１６	０ｘ９９	次動作用Ｓ字区間
Ｒ１５	０ｘ９ａ	減速レート
Ｒ１６	０ｘ９ｂ	Ｓ字区間

表５．１－８　レジスタ読込コマンド」参照

（１６）	**hppd530_ElstRead()　　ＥＬＳ極性選択ポート読込** **hppd530_ElstWrite()　　ＥＬＳ極性選択ポート書込**
機能	デバイス ID で指定されたＰＰＤ５３Ｘの, **ＥＬＳ極性選択ポート読込**・・ＥＬＳ極性選択ポートを読込み, 指定エリアに格納します. **ＥＬＳ極性選択ポート書込**・・ＥＬＳ極性選択ポートに指定データを書込みます.
書式	[Ｃ言語: VC++] DWORD WINAPI hppd530_ElstRead (DWORD hDevID, BYTE* byElsData); DWORD WINAPI hppd530_ElstWrite(DWORD hDevID, BYTE byData); [ＶＢ６] Declare Function hppd530_ElstRead Lib "hippd530.DLL" (ByVal hDevID As Long, ByRef byElsData As Byte) As Long Declare Function hppd530_ElstWrite Lib "hippd530.DLL" (ByVal hDevID As Long, ByVal byData As Byte) As Long
引数	◆ DWORD hDevID ・・対象デバイスのデバイス ID. ◆ BYTE* byElsData・・読込んだデータが格納される１バイトエリアのアドレス ◆ BYTE byData ・・ＥＬＳ極性選択ポートへの書込データ 
戻り値	処理結果 １：成功 ０：失敗 ・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.

呼び出し例	[Ｃ言語: VC++]
	DWORD ret; //関数の戻り値
	BYTE ElsData;
	ret = hppd530_ElstRead (hDeviceID, //デバイス ID
	&ElsData); //格納先のアドレス
	ret = hppd530_ElstWrite(hDeviceID, //デバイス ID
	0x01); //Ｘ軸Ａ接, Ｙ・Ｚ軸Ｂ接
	
	[ＶＢ６]
	Dim ret As Long '関数の戻り値
	Dim ElsData As Byte
	ret = hppd530_ElstWrite(hDeviceID, _ 'デバイス ID
	&H1) 'Ｘ軸Ａ接, Ｙ・Ｚ軸Ｂ接
	ret = hppd530_ElstRead (hDeviceID, _ 'デバイス ID
	ElsData) '格納先のアドレス

<table>
<tr><td>（１７）</td><td>hppd530_CmpcRead()　　コンパレータ同時スタート読込
hppd530_CmpcWrite()　　コンパレータ同時スタート書込</td></tr>
<tr><td>機能</td><td>デバイス ID で指定されたＰＰＤ５３Ｘの,
コンパレータ同時スタート読込・・コンパレータ同時スタートを読込み, 指定エリアに格納します.
コンパレータ同時スタート書込・・コンパレータ同時スタートに指定データを書込みます.</td></tr>
<tr><td>書式</td><td>[Ｃ言語: VC++]
DWORD WINAPI hppd530_CmpcRead (DWORD hDevID, BYTE* byCmpData);
DWORD WINAPI hppd530_CmpcWrite(DWORD hDevID, BYTE byData);
[ＶＢ６]
Declare Function hppd530_CmpcRead Lib "hippd530.DLL"
(ByVal hDevID As Long, ByRef byCmpData As Byte) As Long
Declare Function hppd530_CmpcWrite Lib "hippd530.DLL"
(ByVal hDevID As Long, ByVal byData As Byte) As Long</td></tr>
<tr><td>引数</td><td>◆ DWORD hDevID ・・対象デバイスのデバイス ID.
◆ BYTE* byCmpData・・読込んだデータが格納される１バイトエリアのアドレス
◆ BYTE byData ・・コンパレータ同時スタートへの書込データ
 
</td></tr>
<tr><td>戻り値</td><td>処理結果
１：成功
０：失敗 ・・「４．４．３ デバイスドライバの異常報告(P10)」を参照して下さい.</td></tr>
<tr><td>呼び出し例</td><td>[Ｃ言語: VC++]
DWORD ret; //関数の戻り値
BYTE CmpData;
ret = hppd530_CmpcRead (hDeviceID, //デバイス ID
&CmpData); //格納先のアドレス
ret = hppd530_CmpcWrite(hDeviceID, //デバイス ID
0x01); //Ｘ軸のみ出力可

[ＶＢ６]
Dim ret As Long '関数の戻り値
Dim CmpData As Byte
ret = hppd530_CmpcRead (hDeviceID, _ 'デバイス ID
CmpData) '格納先のアドレス
ret = hppd530_CmpcWrite(hDeviceID, _ 'デバイス ID
&H1) 'Ｘ軸のみ出力可</td></tr>
</table>

６．サンプルプログラム

ドライバＩ／Ｆ用ＤＬＬの各関数の使用方法を解説する目的のサンプルプログラムを添付しています.
サンプルプログラムは次の３種類があり，ほぼ同一の画面表示と操作となっています.
以降のサンプルプログラム説明では，①の「Ｃコーディング」を用います.

① Ｖisual Ｃ/Ｃ++ (6.0)・・ Ｃ コーディング 【 sppα00.exe 】
② Ｖisual Ｃ/Ｃ++ (6.0)・・ Ｃ++コーディング 【 sppα01.exe 】
③ Ｖisual Ｂasic (6.0) 【 sppα02.exe 】

サンプルプログラムを使用する場合は，お客様のハードディスクにコピーして使用します.
個々のサンプル実行ファイル(sppα00.exe) は”マウスのダブルクリック”操作を行う事
で実行できます. 《 αは"534"又は"516"であり，ボード種別で異なります. 》

≪ ご注意 ≫

（１） ＶＢサンプルは開発ツールとして「Microsoft Visual Basic 6.0」がインストールされている必要があります.

（２） 実行開始時に次のエラーメッセージが表示される場合には，サンプルプログラムは動作しません.

【 エラーメッセージの表示 】
※ ＰＰＤ５３ｘボードが未搭載.
※ デバイスドライバがインストールされていない.

６．１ サンプルプログラムの操作

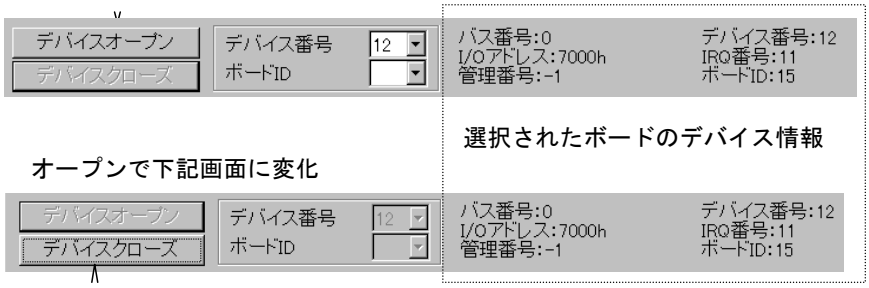
サンプルプログラムが起動され，１枚以上のボード（デバイス）が正常に認識される時，次の画面が表示されます.

（６軸画面）

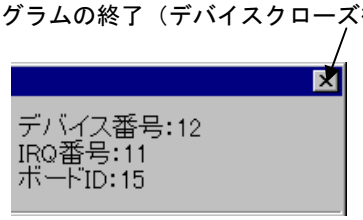
６．１．１　ボード（デバイス）の各種操作

サンプルプログラムでは、ボード上の動作軸操作開始・終了は次の手順に従います.

①ボードの選択（２枚以上の場合）
②デバイスオープン指令　　「デバイス番号」または「ボードＩＤ」

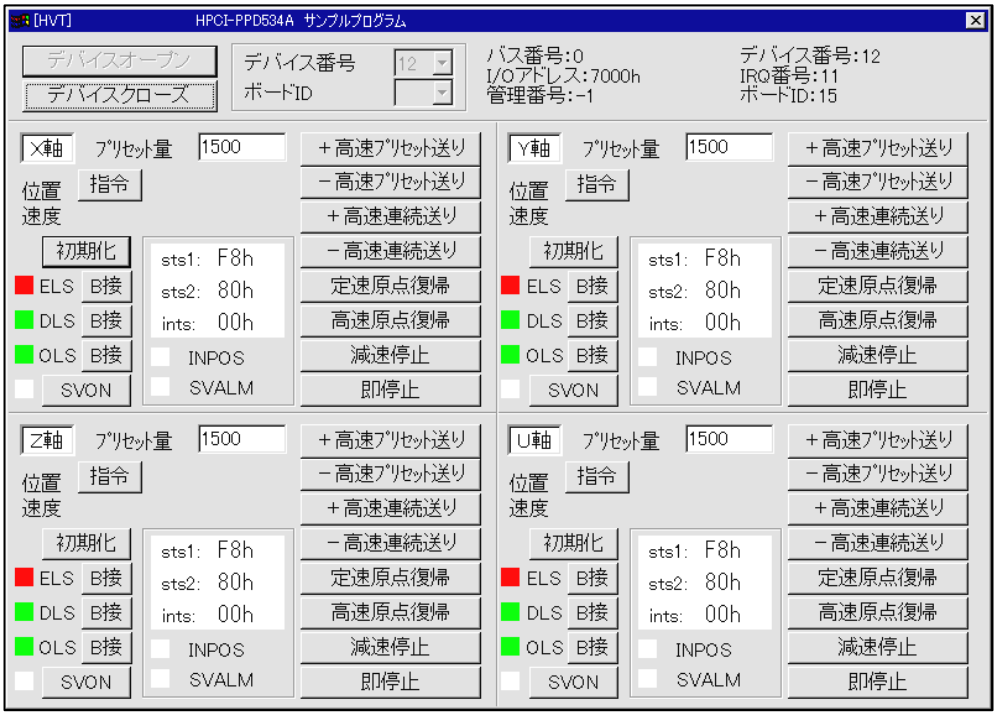


③デバイスクローズ指令（動作中の軸を全て停止させてから）
④サンプルプログラムの終了（デバイスクローズ後）



６．１．２　ボード上の各軸操作

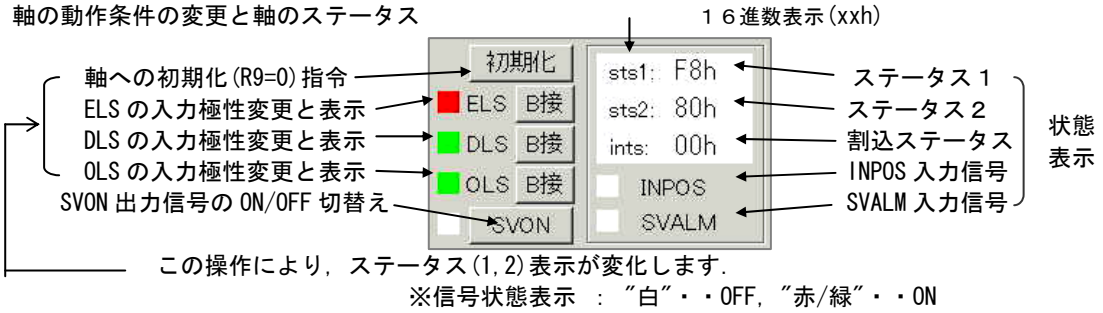
デバイスオープンを行いますと次の画面となります.　（４軸の場合）



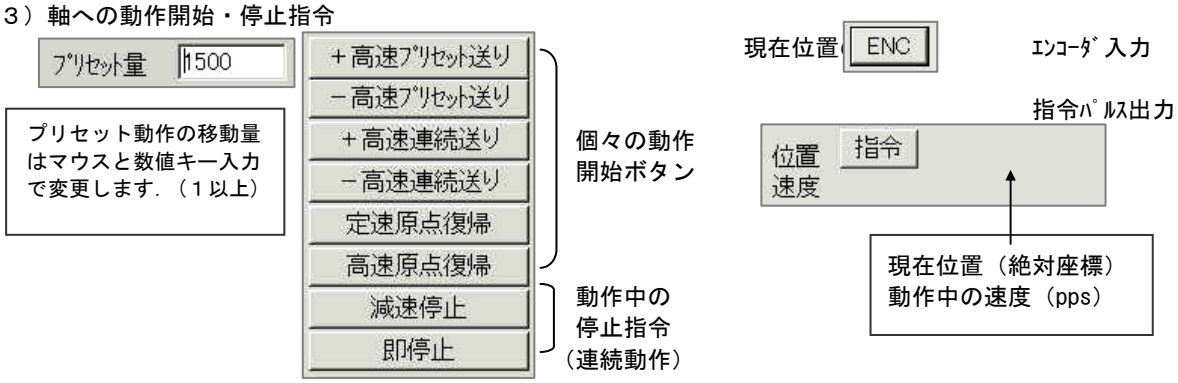
ボード上の軸数の多少に関わらず，個々の軸に対する操作は同一です.
なお，サンプルプログラムでは各軸の初期化は一部ソースプログラムで固定されています.
その為に，初期化の条件を変更して動作させたい場合には，ソースプログラム変更の必要があります.

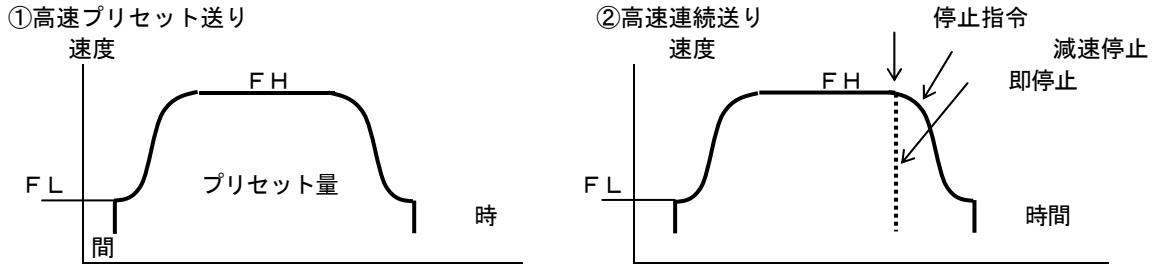
（１）軸数が２軸以下の場合・・有効軸数分のボタンが操作可能となります.

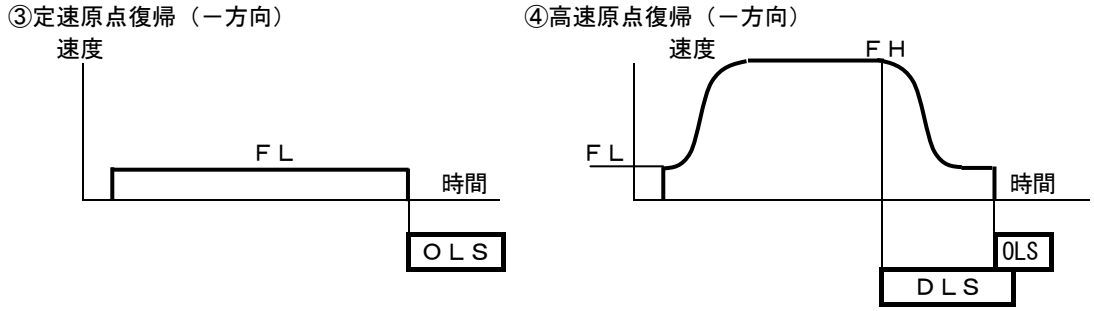
（２）軸の動作条件の変更と軸のステータス



（３）軸への動作開始・停止指令







（注）１．加減速はＳ字を使用しています.
２．高速原点復帰の減速用にＤＬＳを使用しています.
３．原点復帰の方向は″－″方向に固定されています.

６．１．３　軸動作の設定条件

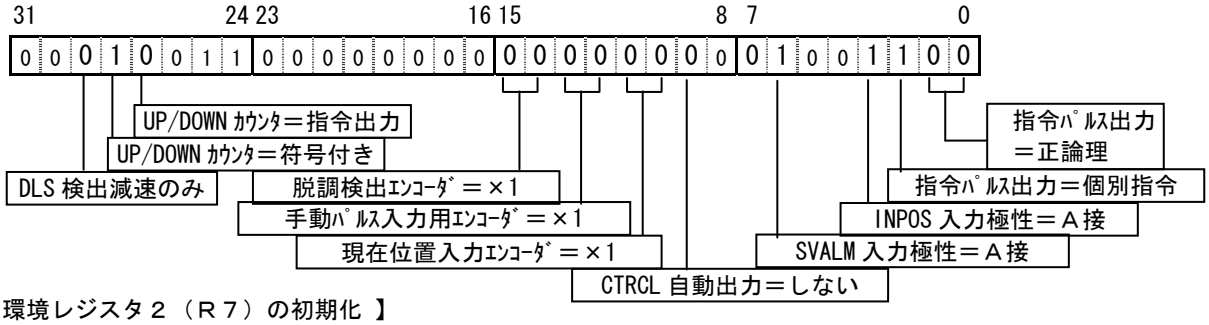
（１）軸制御のレジスタ初期設定（デフォルト値）・・デバイスオープン時

PCL ﾚｼﾞｽﾀ	設定値	備　　考
PR 0	1500	送り量（プリセット動作移動量）
PR 1	500	ＦＬ速度
PR 2	5000	ＦＨ速度
PR 3	545	加減速レート（１秒）[(2457600 * 1sec)/(FH - FL) - 1]
PR 4	299	速度倍率 = 1pps　　[= 300 / (PR4 + 1)]
PR 5	0	減速点　　（制御ﾓｰﾄﾞで"減速点検出=自動"を設定）
R 6	0x1300004c	環境レジスタ１
R 7	0x000f0c40	環境レジスタ２
R 8	0x0000	環境レジスタ３
R 9	0	アップダウンカウンタ（現在位置）
R10	0	コンパレータ１　　（不使用）
R11	0	コンパレータ２　　（不使用）
PR15	0	減速レート　　（設定値＝０で加速ﾚｰﾄ＝減速ﾚｰﾄ）
R16	0	Ｓ字加減速・Ｓ字区間

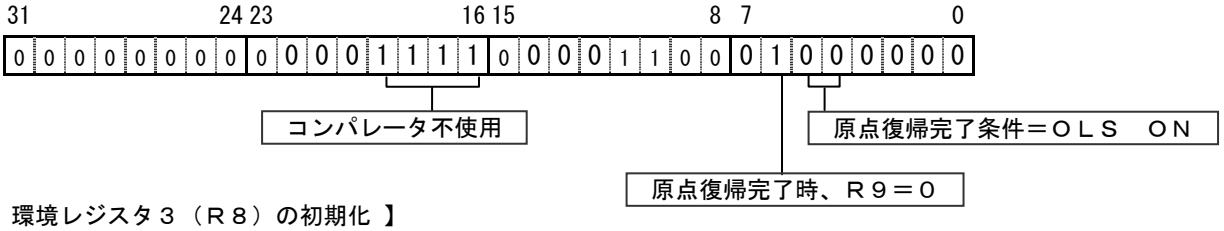
表６．１－１　軸制御のレジスタ初期設定

次に環境レジスタ１～３の詳細内容を記します．・・・・縮小文字"1, 0"は固定ビットデータです．

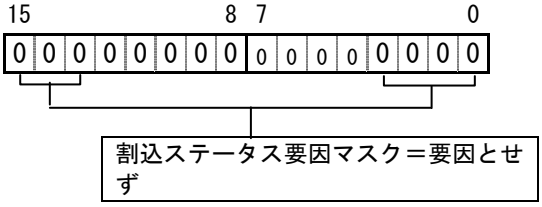
【 環境レジスタ１（Ｒ６）の初期化 】



【 環境レジスタ２（Ｒ７）の初期化 】

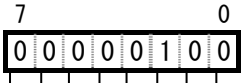


【 環境レジスタ３（Ｒ８）の初期化 】



（２）制御モードバッファ

制御モードバッファへの書込はデバイスオープン時に行われます．



bit	制御内容	設定値	
4	次動作自動ｽﾀｰﾄ制御	0	しない
5	同時停止信号入力機能	0	OFF
6	同時停止信号出力機能	0	ON
7	三角駆動時速度修正機能	0	修正

bit	制御内容	設定値	
0	ﾌﾟﾘｾｯﾄｶｳﾝﾀ動作	0	ON
1	減速点検出方式	0	自動
2	加減速方式	1	Ｓ字
3	INPOSITION 制御	0	しない

６．２ サンプルプログラムの変更について

４種類のサンプルプログラムは,
Microsoft Visual C++ 6.0 及び Microsoft Visual Basic（ 6.0 ）を使用しています.
サンプルプログラムを変更する場合には，上記開発環境がパソコンにインストールされている事が前提条件です.

個々のディレクトリ（ホルダー）には「実行可能ファイル(.exe)」を含めて，新規の実行可能ファイルを作成する為の全てのファイルが格納されています.

ファイル種類	開発環境・コーディング区分		
	Visual C/C++ (C)	Visual C/C++(C++)	Visual Basic
実行可能ﾌｧｲﾙ	sppα00.exe	sppα01.exe	sppα02.exe
ﾌﾟﾛｼﾞｪｸﾄﾌｧｲﾙ	sppα00.dsw sppα00.dsp sppα00.opt	sppα01.dsw sppα00.dsp sppα00.opt	sppα02.vbp
ﾄﾞﾗｲﾊﾞ I/Fﾌｧｲﾙ	hippd530.dll hippd530.lib hippd530.h	hippd530.dll hippd530.lib hippd530.h	hippd530.dll
ｿｰｽﾌﾟﾛｸﾞﾗﾑ	上記以外	上記以外	上記以外

６．２．１ プロジェクトファイル

（１）VB6 版サンプルプログラムを変更する場合
プロジェクトファイルは，spp53002.vbp を選択して下さい.
Microsoft Visual Basic 6.0 がインストールされている必要があります.

（２）VC++(C) 版サンプルプログラムを変更する場合
プロジェクトワークスペースは，spp53000.dsw を選択して下さい.
Microsoft Visual C++ 6.0 がインストールされている必要があります.

（３）VC++(C++)版サンプルプログラムを変更する場合
プロジェクトワークスペースは，spp53001.dsw を選択して下さい.
Microsoft Visual C++ 6.0 がインストールされている必要があります.

（４）開発環境のバージョンが異なる場合
Microsoft Visual Basic 5.0, Microsoft Visual C++ 5.0 等を使用している場合には，そのままプロジェクトファイルが使用出来ません.

この場合には，新規のプロジェクトを作成し，このプロジェクトにサンプルプログラムを構成する各種のファイルを追加して下さい.

なお，開発言語の種類により，サンプルプログラムで使用している機能が使用できない事があります.
ソースプログラムを削除してご使用下さい.

[例] Microsoft Visual C++ 5.0 で"VC++(C++)"サンプルをビルドする場合
stdafx.h ・・・ #include 〈afxdtctl.h〉 の１行をコメントとします.

６．２．２　サンプルプログラムを構成する関数

サンプルプログラムを構成する関数を大別すると次のようになります.

分類	No	関　数　名	関　数　の　処　理　（　ボタン対応　）
画面表示の処理関数	1	WinMain()	Windowsのメイン関数
	2	WndProc()	Windowsから呼び出されて、メッセージキューからメッセージの引き渡しを受ける
	3	FormLoad1()	ウィンドウを中央へ移動する、ハンドル取得
	4	FormLoad3()	ダイアログボックス位置取得
	5	FormUnload()	ウィンドウを閉じる
	6	FormMove()	ウィンドウが移動した時
	7	CmbJoho_Click()	デバイス情報選択 & 表示(コンボボックスのクリック)
	8	BtnOpen2_Click()	デバイスオープンボタンのクリック
	9	BtnClose1_Click()	デバイスクローズボタンのクリック(クローズ直前)
	10	BtnClose3_Click()	デバイスクローズボタンのクリック(クローズ直後)
	11	Dlg1Proc()	Ｘ軸～Ｕ軸のダイアログボックス
	12	DlgLoad()	ダイアログボックス開始(ハンドル取得)
	13	BtnDsbl()	動作確認時、ボタン使用不可
	14	BtnEls2_Click()	ＥＬＳ(Ａ接/Ｂ接)ボタンのクリックでＥＬＳ入力極性の反転
	15	BtnDls2_Click()	ＤＬＳ(Ａ接/Ｂ接)ボタンのクリックでＤＬＳ入力極性の反転
	16	BtnOls2_Click()	ＯＬＳ(Ａ接/Ｂ接)ボタンのクリックでＯＬＳ入力極性の反転
	17	BtnEnc2_Click()	エンコーダ入力/指令出力ボタンのクリックで現在位置読込変更
	18	ErrMsg()	エラーメッセージ出力（サンプルプログラムの終了）
ボード制御・軸制御の関数	19	FormLoad2()	デバイス情報取得
	20	BtnOpen1_Click()	デバイスオープンボタンのクリックでボード初期化
	21	Initial()	ボード上の軸初期化（１軸分）
	22	BtnClose2_Click()	デバイスクローズボタンのクリックで軸停止とデバイスクローズ
	23	BtnPPre_Click()	＋高速プリセット送りボタンのクリックで所定動作開始
	24	BtnMPre_Click()	－高速プリセット送りボタンのクリックで所定動作開始
	25	BtnPRen_Click()	＋高速連続送りボタンのクリックで所定動作開始
	26	BtnMRen_Click()	－高速連続送りボタンのクリックで所定動作開始
	27	BtnTGen_Click()	定速原点復帰ボタンのクリックで所定動作開始
	28	BtnKGen_Click()	高速原点復帰ボタンのクリックで所定動作開始
	29	BtnGStop_Click()	減速停止ボタンのクリックで軸に減速停止指令を発行
	30	BtnSStop_Click()	即停止ボタンのクリックで軸に即停止指令を発行
	31	BtnIni_Click()	初期化ボタンのクリックで「ｱｯﾌﾟﾀﾞｳﾝｶｳﾝﾀ(R9)」クリア
	32	BtnSvOn_Click()	ＳＶＯＮ(ＯＮ/ＯＦＦ)ボタンのクリックで出力信号の反転指令
	33	BtnEls1_Click()	ＥＬＳ(Ａ接/Ｂ接)ボタンのクリックで入力信号の反転指令
	34	BtnDls1_Click()	ＤＬＳ(Ａ接/Ｂ接)ボタンのクリックで入力信号の反転指令
	35	BtnOls1_Click()	ＯＬＳ(Ａ接/Ｂ接)ボタンのクリックで入力信号の反転指令
	36	BtnEnc1_Click()	エンコーダ入力/指令出力ボタンのクリックで現在位置入力切替
	37	TmrSokudo()	現在速度 & 現在位置表示　(動作中軸：0.1秒毎)
	38	TmrSts()	ステータス読込　　(動作中軸：0.1秒毎)

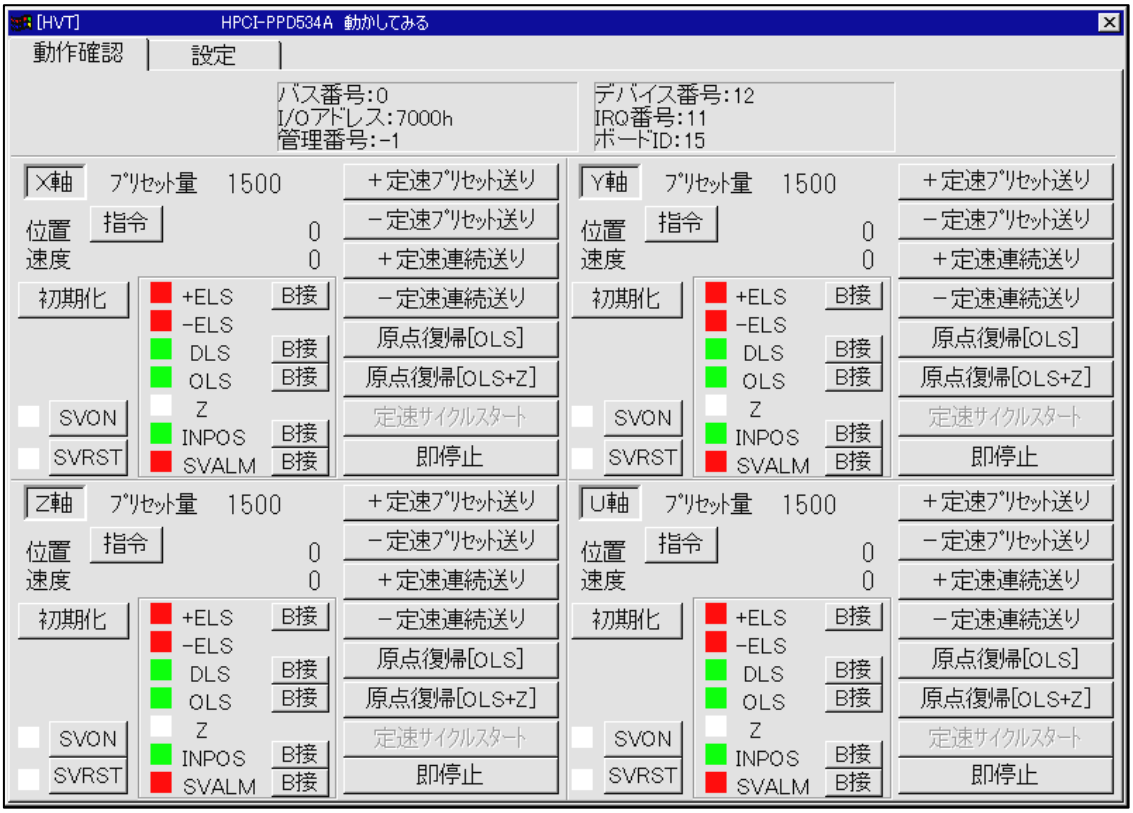
（注）１．上表は「Visual C　版」の場合です.
「Visual Basic 版」の場合もほぼ同様です.
「Visual C++版」の場合は極力同様の関数構成に努めています.

７．「動かしてみる」プログラム

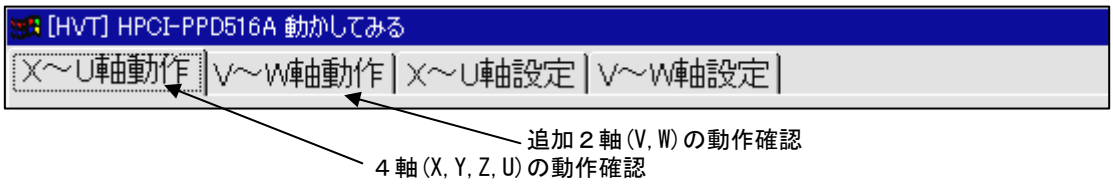
「動かしてみる」プログラムは、ボードをパソコンへ装着するだけで、最小限の動作をディスプレイ上で確認できるソフトです.
添付ソフトウェアフロッピーディスクの「¥test¥tp53400.exe」または「¥testtpp51600.exe」を実行してください.

７．１ 「動かしてみる」の動作確認画面

「動かしてみる」プログラム実行で次の画面が表示されます. （４軸の場合）



ボードに搭載された軸数が４軸ではない時、次のようになります。
●軸数が３軸以下の場合・・有効軸数分のボタンが操作可能となります。
●軸数が５軸以上の場合・・４軸までの画面と５軸以降の画面を切り替えて操作します。



７．１．１ デバイス情報の表示

現在選択されているボードのデバイス情報は下記部分に表示されます.

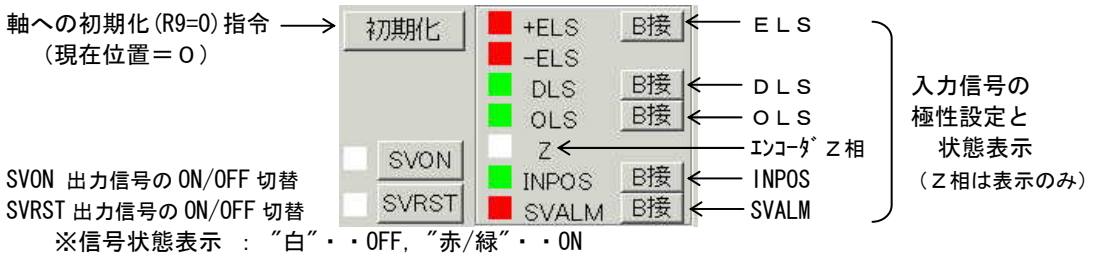
（注）１．管理番号　　Windows９Ｘでは"－１"です.
　　２．ボードＩＤ　　ボード上のジャンパ設定値です.
　　ジャンパの無いボードでは"１５"となります.

７．１．２　個々の軸表示と動作指令

ボード上の軸数の多少に関わらず，個々の軸に対する操作は同一です．
なお，サンプルプログラムでは各軸の初期化は一部ソースプログラムで固定されています．
その為に，初期化の条件を変更して動作させたい場合には，ソースプログラム変更の必要があります．

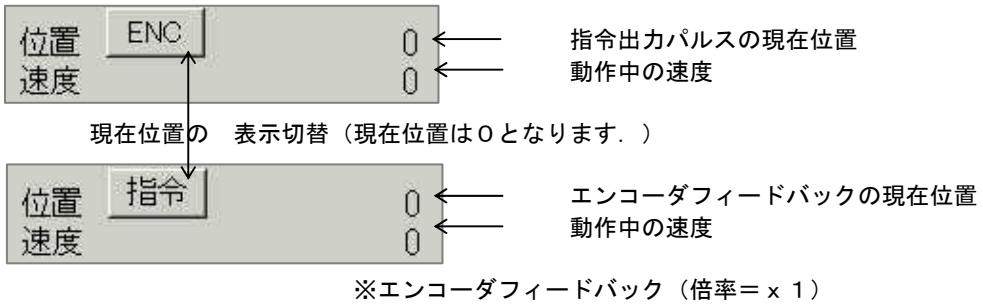
（１）軸数が３軸以下の場合・・有効軸数分のボタンが操作可能となります．

（２）軸の動作条件の変更と軸のステータス



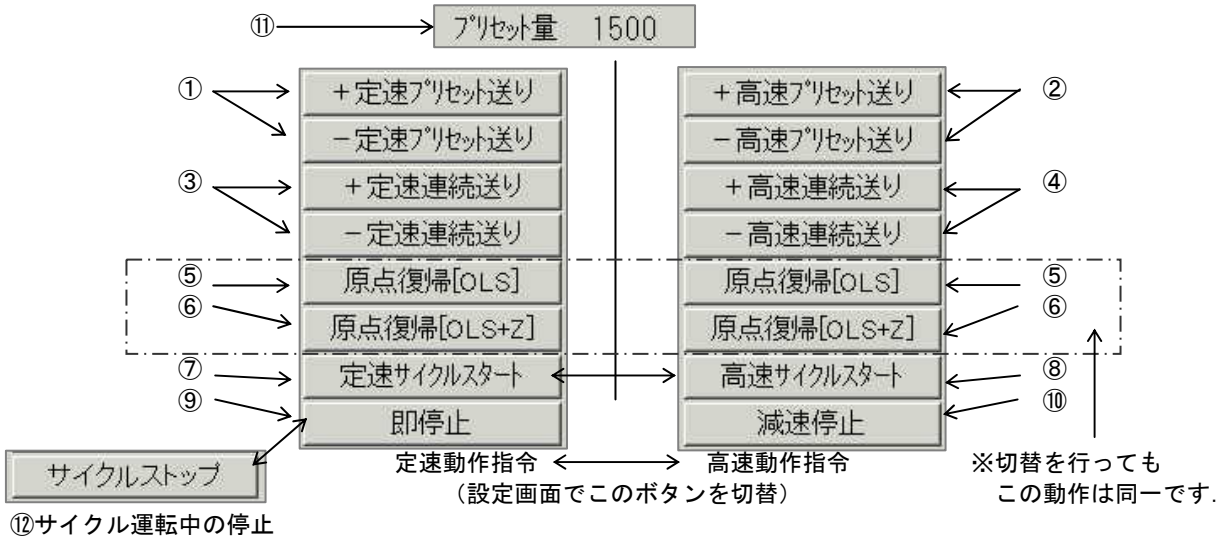
（３）軸の現在位置・動作速度表示

各軸の現在位置および動作中の速度は０．１秒毎に更新されます．
現在位置については，「指令出力パルス」の表示と「エンコーダフィードバック」の表示が選択できます．



（４）軸への動作開始・停止指令

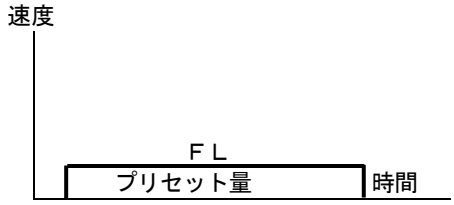
個々の軸に対する動作は，①～⑩あり，⑪プリセット量は①②⑦⑧の４種類の移動量となります．



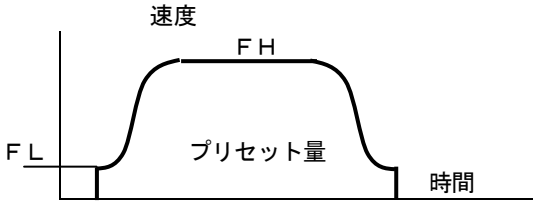
（５）軸への指令と動作内容

「動かしてみる」プログラムの動作内容は下図の通りです.

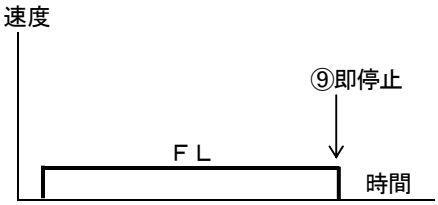
①定速プリセット送り



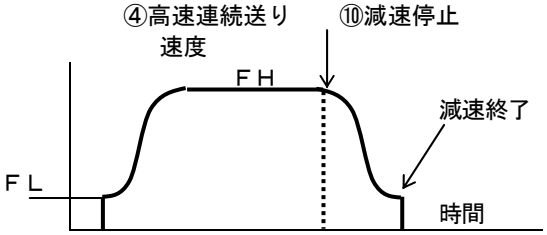
②高速プリセット送り



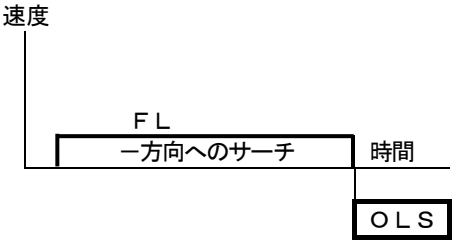
③定速連続送り



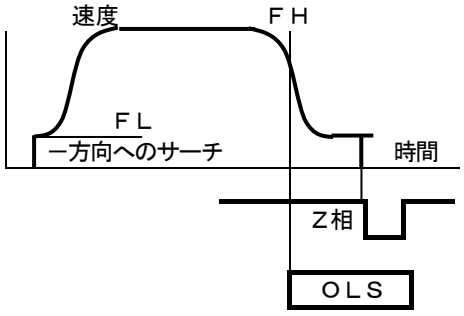
④高速連続送り



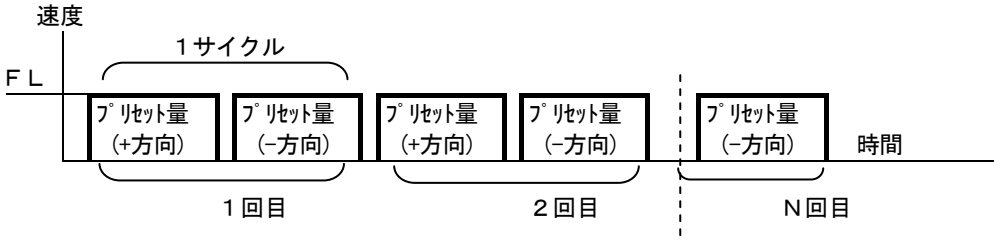
⑤原点復帰[ＯＬＳ]



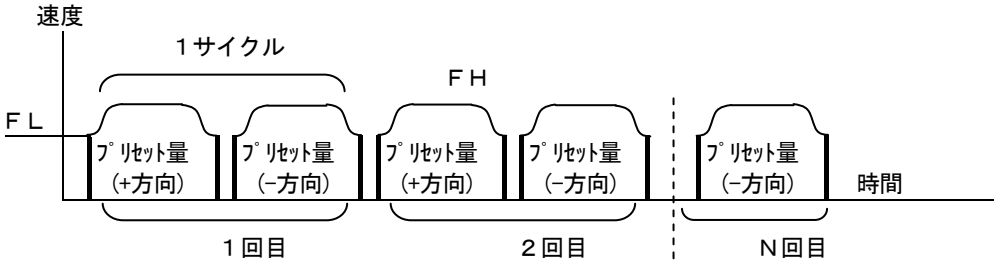
⑥原点復帰[ＯＬＳ+Ｚ]



⑦定速サイクル（定速プリセットの往復動作）



⑧高速サイクル（高速プリセットの往復動作）



（注）１．加減速はＳ字を使用しています.

２．原点復帰の方向は"－"方向に固定されています.

３．原点復帰[ＯＬＳ]は定速であり，原点信号＝ＯＬＳです.

４．原点復帰[ＯＬＳ＋Ｚ]は高速であり，原点信号＝ＯＬＳ減速＋Ｚ相１回です.

５．動作に関係する要素（プリセット量･速度等）は「設定画面」で変更出来ます.

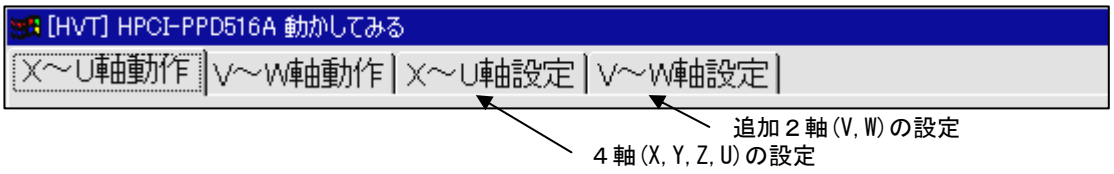
６．"サイクル"動作は「設定画面」で回数指定を行います.

７．"サイクル"動作では「次動作自動スタート」機能で連続動作となります.

７．２　「動かしてみる」の設定画面

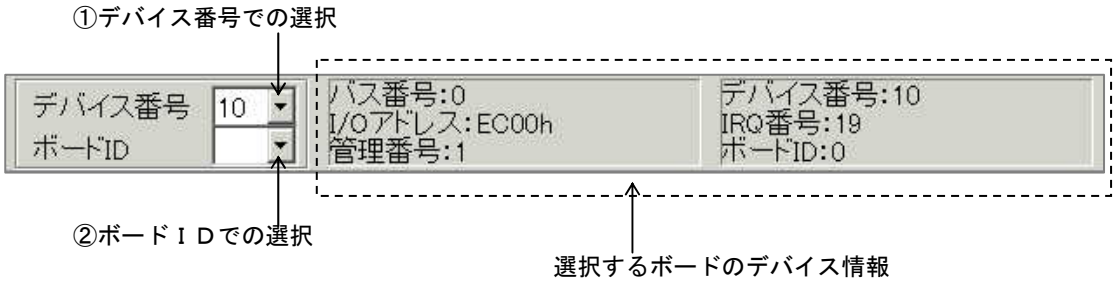
「動作確認」画面で全ての軸を停止させて"設定"を選択しますと下記画面が表示されます.

（注） ６軸の場合には次の部分が異なります.



７．２．１　ボード選択とデバイス情報

ボードが２枚以上装着されている場合に，「デバイス番号」または「ボードＩＤ」で「動かしてみる」ボードを指定します.



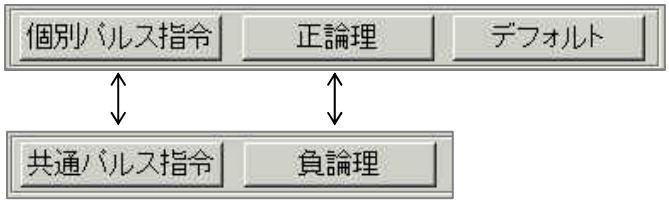
７．２．２　変更可能な軸動作条件

動作可能な全ての軸について，個々に動作条件が設定出来ます.

（１）設定項目とＰＣＬレジスタ

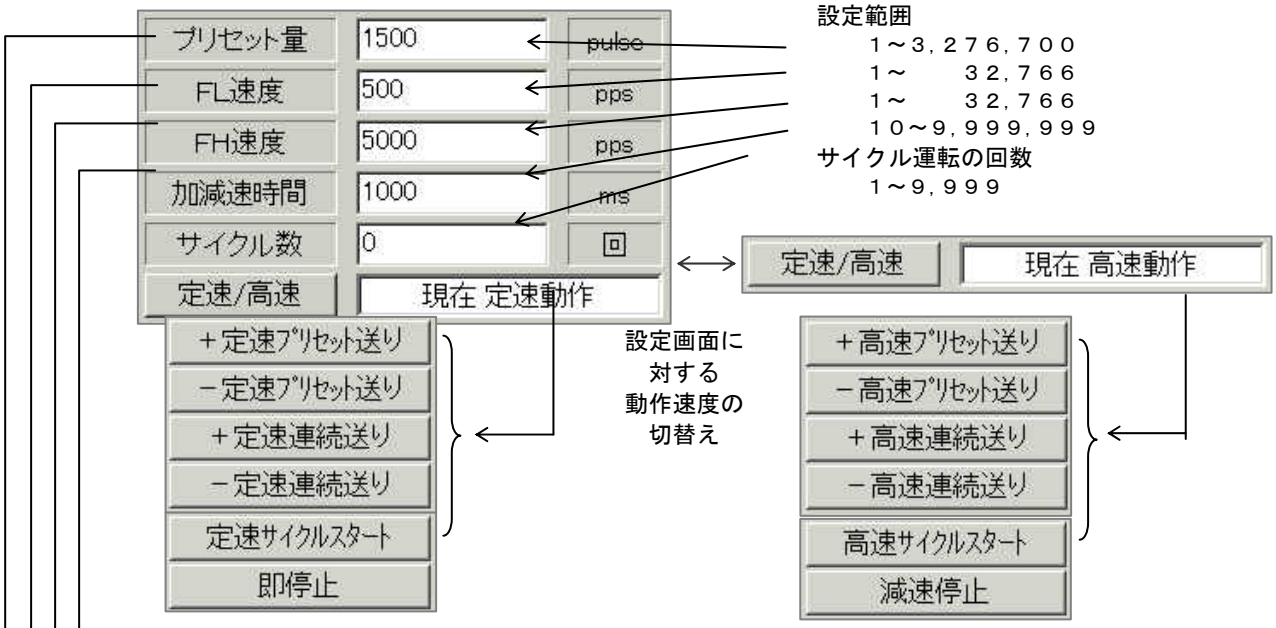
設定画面に於ける設定項目と，設定値とＰＣＬレジスタの関連は次の通りです.

【　指令パルス出力の設定　】　　モータ接続を行う場合，モータ用ドライバの接続方法に合わせて下さい.



※デフォルトボタン
″個別パルス指令″および
″正論理″に戻します.

【　移動量･速度･サイクル･動作モード　】



設定範囲
１～３，２７６，７００
１～　　３２，７６６
１～　　３２，７６６
１０～９，９９９，９９９
サイクル運転の回数
１～９，９９９

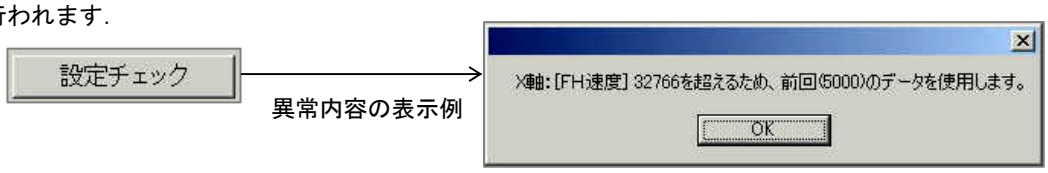
PCL ﾚｼﾞｽﾀ	設定値	備　　考
PR 0	1500	送り量（プリセット動作移動量）
PR 1	500	ＦＬ速度
PR 2	5000	ＦＨ速度
PR 3	545	加減速レート（１秒）[(2457600 * 1sec)/(FH - FL) - 1]
PR 4	299	速度倍率 = 1pps　　[= 300 / (PR4 + 1)]
PR 5	0	減速点　　（制御ﾓｰﾄﾞで″減速点検出=自動″を設定）
R 6	0x13000004	環境レジスタ１
R 7	0x000f0c40	環境レジスタ２
R 8	0x0000	環境レジスタ３
R 9	0	アップダウンカウンタ（現在位置）
R10	0	コンパレータ１　　（不使用）
R11	0	コンパレータ２　　（不使用）
PR15	0	減速レート　　（設定値＝０で加速ﾚｰﾄ＝減速ﾚｰﾄ）
R16	0	Ｓ字加減速・Ｓ字区間

個々の詳細内容は次ページを参照して下さい.

※　関連付けられたレジスタと環境レジスタ１以外はデフォルト値であり，デバイスオープン時に初期設定されます.

（２）設定値の確定操作

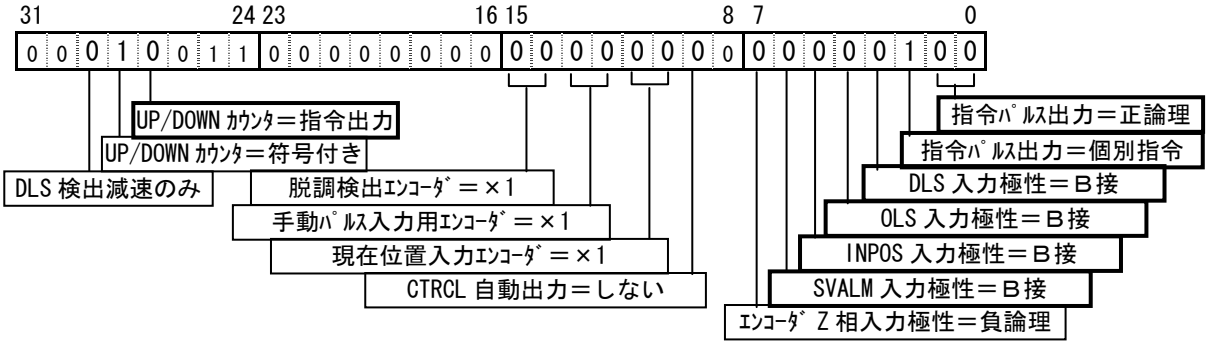
画面上で数値入力した値は「設定チェック」ボタン操作で確定されます.
設定値が範囲外である場合には，異常表示が行われます.
なお，このボタン操作を行わずに「動作確認」画面に戻る操作を行いますと，自動的に設定チェックが行われます.



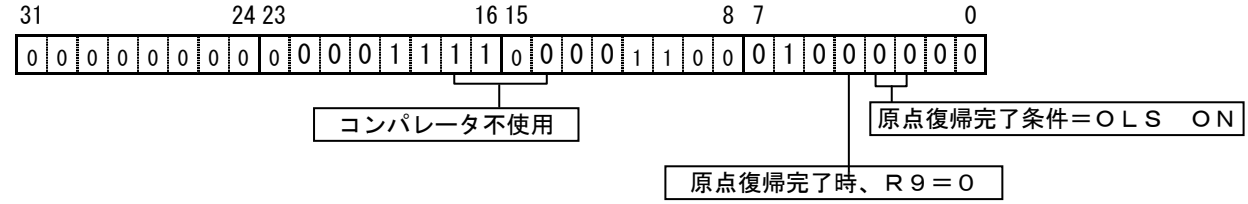
（３）環境レジスタの設定内容

３種類の環境レジスタの設定内容は次の通りです.
個々の項目で　　で表された内容は，画面上の操作で変更されます.

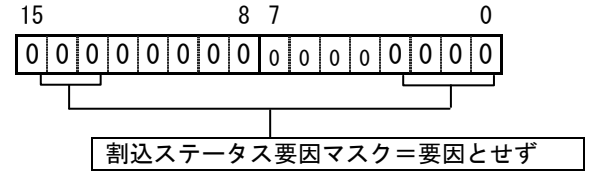
【 環境レジスタ１（Ｒ６）の設定値 】



【 環境レジスタ２（Ｒ７）の設定値 】

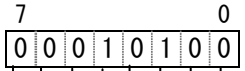


【 環境レジスタ３（Ｒ８）の設定値 】



（４）制御モードバッファ

制御モーＯドバッファへの書込はデバイスオープン時に行われます.



bit	制御内容	設定値	
4	次動作自動ｽﾀｰﾄ制御	0	する
5	同時停止信号入力機能	0	OFF
6	同時停止信号出力機能	0	ON
7	三角駆動時速度修正機能	0	修正

bit	制御内容	設定値	
0	ﾌﾟﾘｾｯﾄｶｳﾝﾀ動作	0	ON
1	減速点検出方式	0	自動
2	加減速方式	1	Ｓ字
3	INPOSITION 制御	0	しない

※"次動作自動スタート"は「サイクル動作」用の設定です.

附Ａ　ＰＣＬ５０１４レジスタの抜粋

１．　ＰＣＬ５０１４・レジスタの使用分類

図４．３－１に，レジスタ群を体系的に図示します．

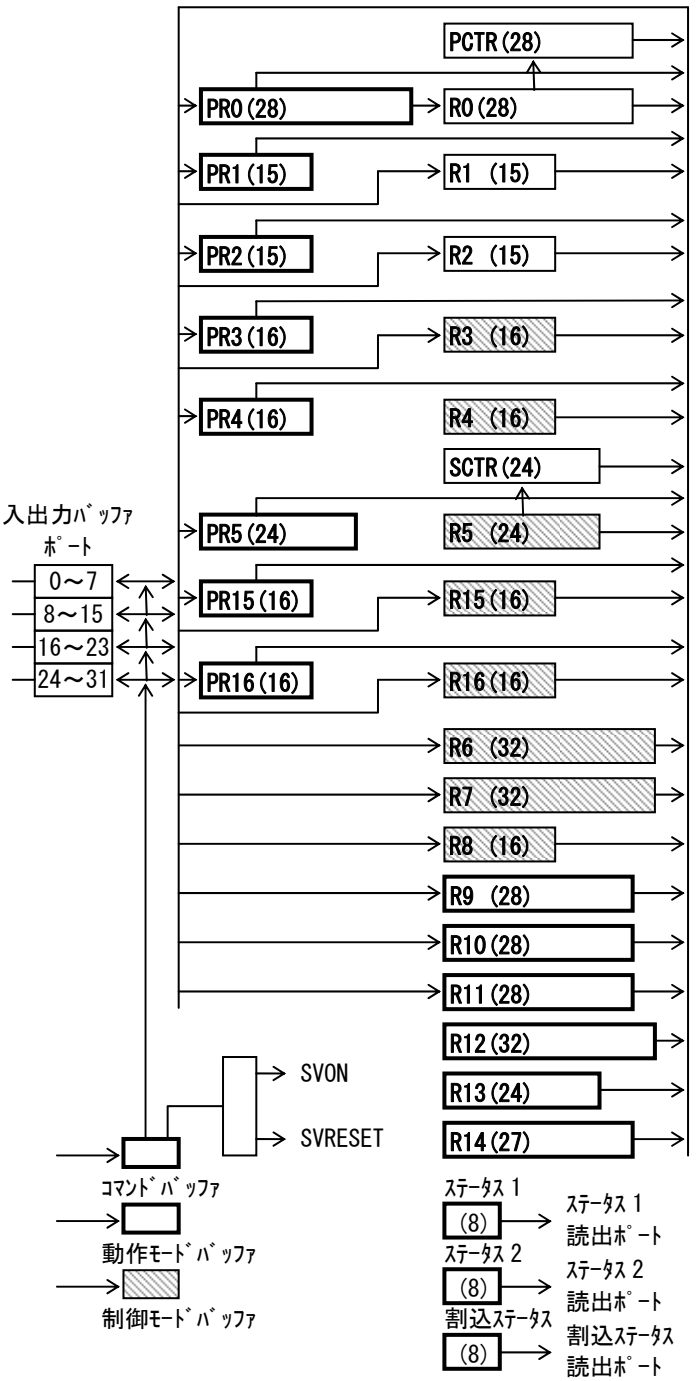
レジスタ群のセット及びリードの使用大別は，使用上次のように考えます．

① 原則として初期化時に一度設定すればよいレジスタ類

② 実行時に設定するレジスタ又は必要に応じて読出して使用するレジスタ類

③ その他レジスタ類

（図中枠線区別）



PCTR:ﾌﾟﾘｾｯﾄｶｳﾝﾀ

PR0:ﾌﾟﾘﾚｼﾞ R0

R0 :送り量ﾚｼﾞｽﾀ

PR1:FL 速度ﾌﾟﾘﾚｼﾞ

R1 :FL 速度ﾚｼﾞｽﾀ

PR2:FH 速度ﾌﾟﾘﾚｼﾞ

R2 :FH 速度ﾚｼﾞｽﾀ

PR3:加減速ﾚｰﾄﾌﾟﾘﾚｼﾞｽﾀ

R3 :加減速ﾚｰﾄﾚｼﾞｽﾀ

PR4:倍率ﾌﾟﾘﾚｼﾞｽﾀ

R4 :倍率ﾚｼﾞｽﾀ

SCTR:減速点ｶｳﾝﾀ

PR5:減速点ﾌﾟﾘﾚｼﾞｽﾀ

R5 :減速点ﾚｼﾞｽﾀ

PR15:減速ﾚｰﾄﾌﾟﾘﾚｼﾞｽﾀ

R15 :減速ﾚｰﾄﾚｼﾞｽﾀ

PR16:Ｓ字区間ﾌﾟﾘﾚｼﾞｽﾀ

R16 :Ｓ字区間ﾚｼﾞｽﾀ

R6 :環境ﾚｼﾞｽﾀ 1

R7 : 〃 2

R8 : 〃 3

R9 :UP/DOWN ｶｳﾝﾀ

R10:ｺﾝﾊﾟﾚｰﾄ値 1 ﾚｼﾞｽﾀ

R11: 〃 2 ﾚｼﾞｽﾀ

R12:ｶｳﾝﾀﾓﾆﾀ

R13:ｺﾏﾝﾄﾞﾓﾆﾀ 1

R14:ｺﾏﾝﾄﾞﾓﾆﾀ 2

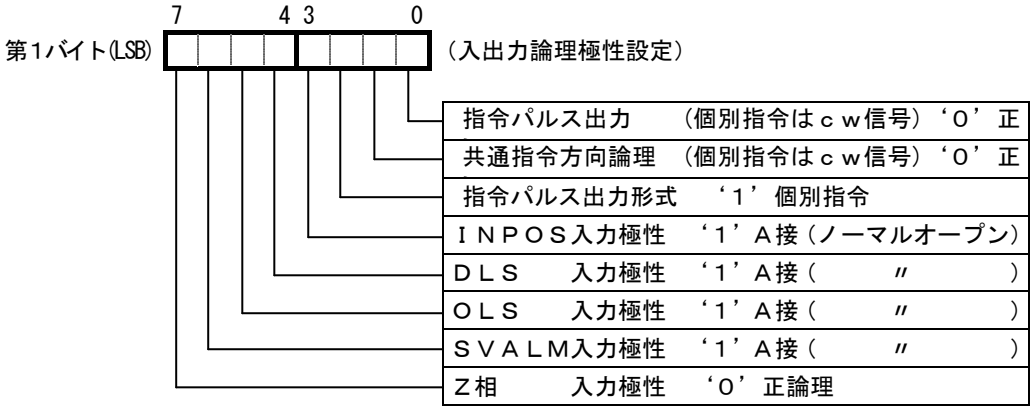
ｽﾃｰﾀｽ 1

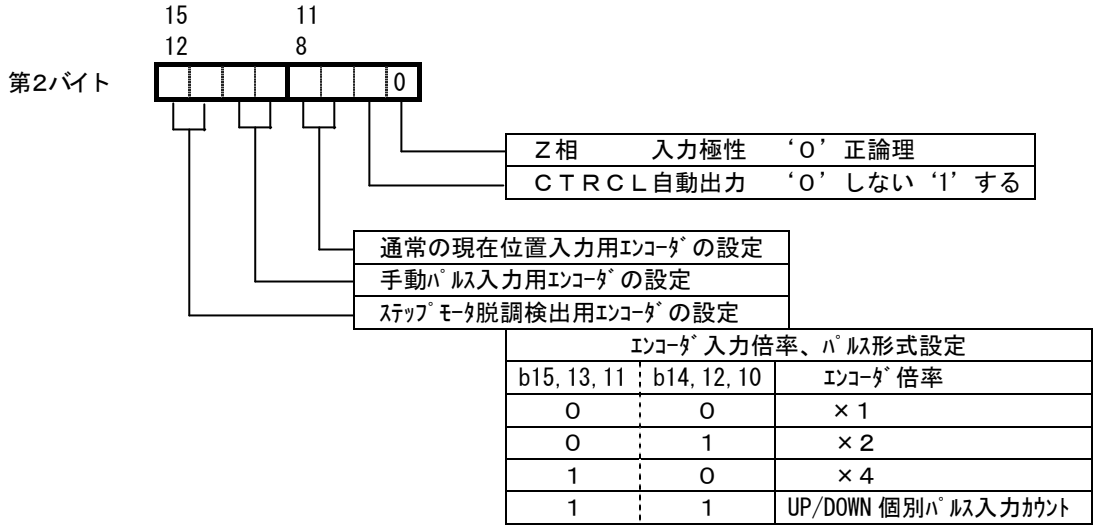
ｽﾃｰﾀｽ 2

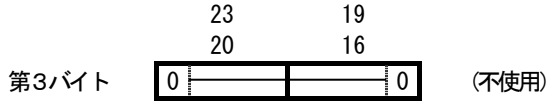
割込ｽﾃｰﾀｽ

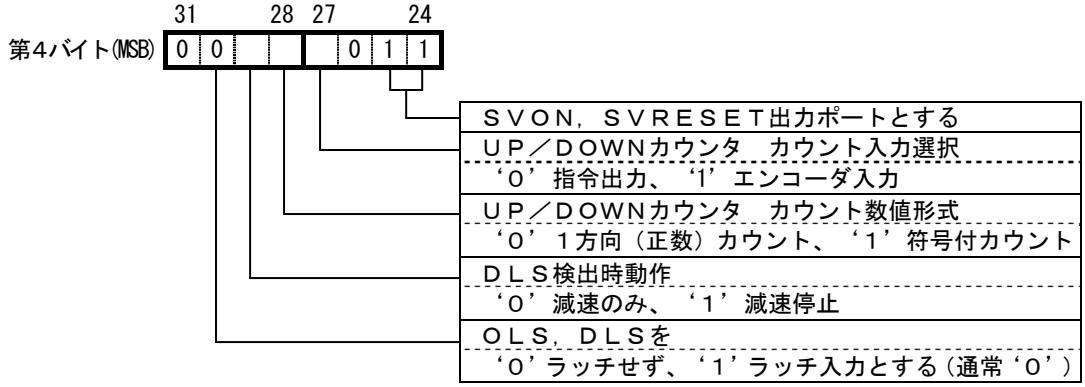
２．個別レジスタ

２．１　環境レジスタ１（Ｒ６）



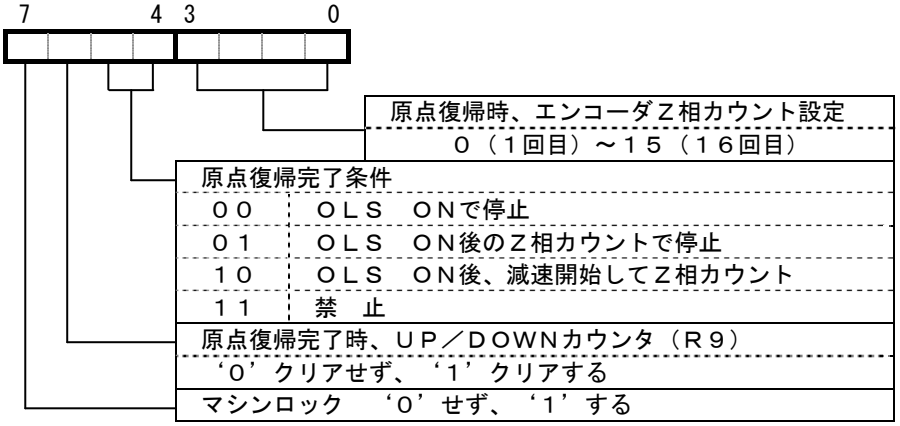






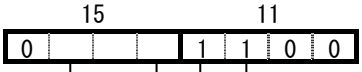
２．２　環境レジスタ２（Ｒ７）

第1バイト(LSB)



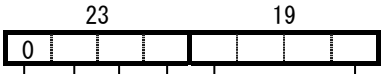
原点復帰時、エンコーダＺ相カウント設定	
０（１回目）～１５（１６回目）	
原点復帰完了条件	
００	ＯＬＳ　ＯＮで停止
０１	ＯＬＳ　ＯＮ後のＺ相カウントで停止
１０	ＯＬＳ　ＯＮ後、減速開始してＺ相カウント
１１	禁　止
原点復帰完了時、ＵＰ／ＤＯＷＮカウンタ（Ｒ９）	
‘０’クリアせず、‘１’クリアする	
マシンロック　‘０’せず、‘１’する	

第2バイト



ＣＴＲＣＬ出力後の次のｽﾀｰﾄｺﾏﾝﾄﾞ遅延有効
共通指令出力形式時、方向変化ﾊﾟﾙｽ出力遅延有効
アイドリングパルス数設定０～７

第3バイト



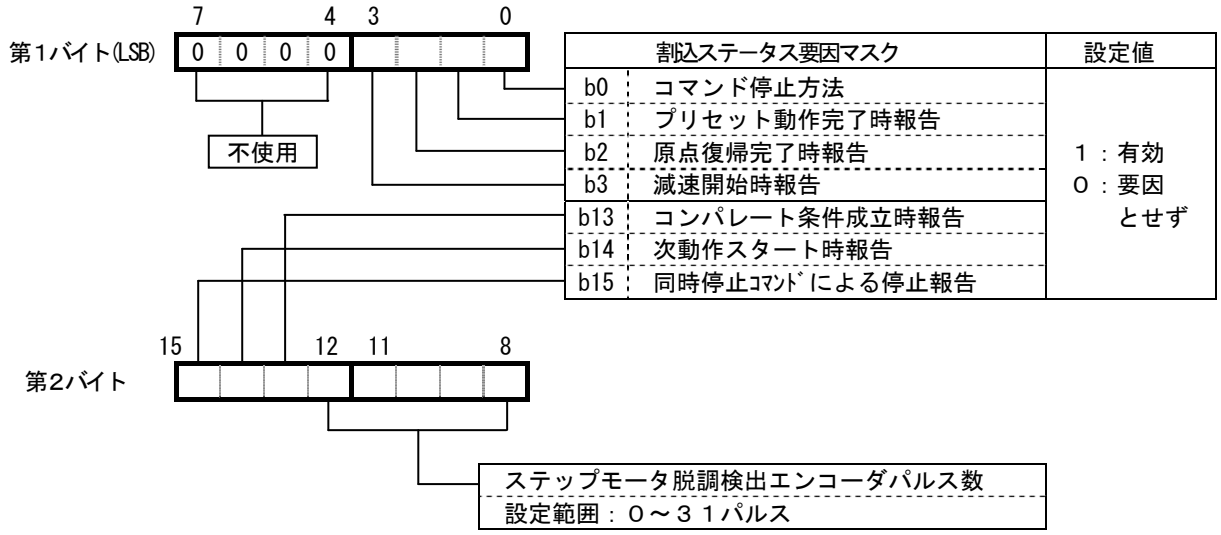
コンパレート条件成立時の指令パルスに対する動作選択	
００	何もせず
０１	指令パルス出力停止
１０	内部動作は継続するがパルスを出力しない
１１	R2(FH)，R3(加減速ﾚｰﾄ)を PR2,PR3 の内容にする（速度変更）
比較カウンタ選択　‘０’UP/DOWN ｶｳﾝﾀ(R9)、‘１’ﾌﾟﾘｾｯﾄｶｳﾝﾀ(PCTR)	
不使用	

コンパレート条件選択		
b19-b16	比較条件	結果
０００００	Ｒ１０＞(ｶｳﾝﾀ)	(1)ｽﾃｰﾀｽ 2 b0 = ‘1’ で条件成立が確認できる。 (2) ‘＝’条件は、一致からｶｳﾝﾀ変化で直ちに変わる。この場合は割込ｽﾃｰﾀｽを利用する (3)ｺﾝﾊﾟﾚｰﾀ不使用は「１１１１」とする
０００１	Ｒ１０＝(ｶｳﾝﾀ)	
００１０	Ｒ１０＜(ｶｳﾝﾀ)	
０１００	Ｒ１１＞(ｶｳﾝﾀ)	
０１０１	Ｒ１１＝(ｶｳﾝﾀ)	
０１１０	Ｒ１１＜(ｶｳﾝﾀ)	
１０００	Ｒ１０＞(ｶｳﾝﾀ) 又は Ｒ１１＜(ｶｳﾝﾀ)	
１００１	Ｒ１０＜(ｶｳﾝﾀ) 又は Ｒ１１＞(ｶｳﾝﾀ)	
１０１０	Ｒ１０＝(ｶｳﾝﾀ) 又は Ｒ１１＝(ｶｳﾝﾀ)	
１０１１	Ｒ１０＞(ｶｳﾝﾀ) 又は Ｒ１１＞(ｶｳﾝﾀ)	
１１００	Ｒ１０＜(ｶｳﾝﾀ) 又は Ｒ１１＜(ｶｳﾝﾀ)	
１１０１	Ｒ１０＜(ｶｳﾝﾀ)＜Ｒ１１	
１１１０	Ｒ１０＞(ｶｳﾝﾀ)＞Ｒ１１	

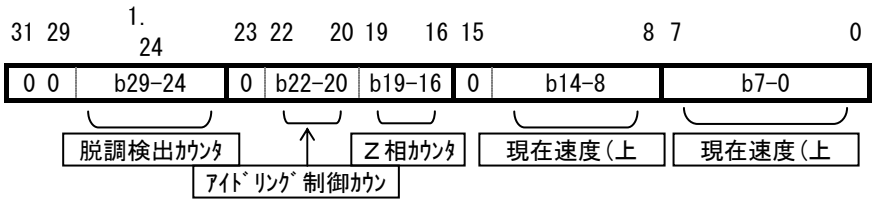
第4バイト(MSB)

（不使用）

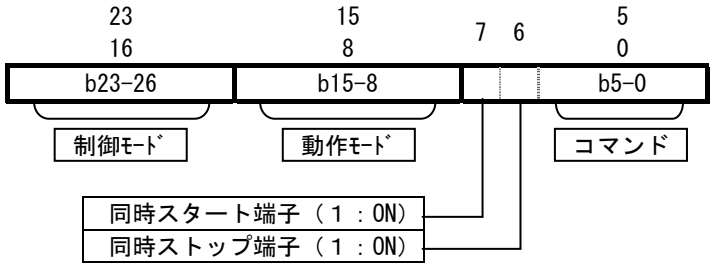
２．３　環境レジスタ３（Ｒ８）



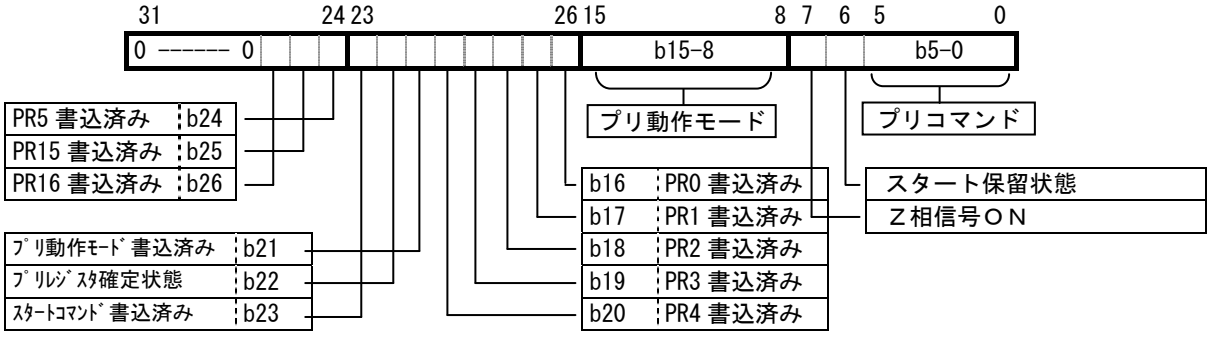
２．４　カウンタモニタ　（Ｒ１２）



２．５　コマンドモニタ１（Ｒ１３）



２．６　コマンドモニタ２（Ｒ１４）



附Ｂ　添付ソフトウェア第２．０版以降とそれ以前の主な相違点について

■”デバイスドライバ”にボードＩＤ（０～１５）がサポートされました.
　ボードＩＤは，ボード上の４ビット・ディップスイッチで決定される値であり，ソフトで
この値を指定したデバイス（ボード）指定が可能となりました.

◆デバイス情報に”ボードＩＤ”の項目が追加されました.

[Ｃ言語：Windows: VC++]

```
typedef  struct  _HPCDEVICEINFO {
    DWORD     nBusNumber;          /* バス番号 */
    DWORD     nDeviceNumber;       /* デバイス番号（PCIｽﾛｯﾄ番号）*/
    DWORD     dwIoPortAddress;     /* I/O ポートアドレス */
    DWORD     dwIrqNo;               /* IRQ 番号 */
    DWORD     dwNumber;            /* 管理番号 */
→   DWORD     dwBoardID;           /* ボードＩＤ(0～15) */
}  HPCDEVICEINFO,  *PHPCDEVICEINFO
```

[Windows：VB6]

```
Public  Type  HPCDEVICEINFO
    nBusNumber      As Long          'バス番号
    nDeviceNumber   As Long          'デバイス番号
    dwIoPortAddress As Long          'I/O ポートアドレス
    dwIrqNo         As Long          'IRQ 番号
    dwNumber        As Long          '管理番号
→   dwBoardID       As Long          'ボードＩＤ(0～15)
End  Type
```

[Windows：VB.NET]

```
Public Structure HPCDEVICEINFO
    Dim nBusNumber      As Integer     'バス番号
    Dim nDeviceNumber   As Integer     'デバイス番号
    Dim dwIoPortAddress As Integer     'I/O ポートアドレス
    Dim dwIrqNo         As Integer     'IRQ 番号
    Dim dwNumber        As Integer     '管理番号
→   Dim dwBoardID       As Integer     'ボード ID
End  Structure
```

■ドライバＩ/Ｆ　ＤＬＬ関数の引数と戻り値のデータ型

◆VC++
”ＢＯＯＬ”・”ＩＮＴ”の定義を”ＤＷＯＲＤ(unsigned long)”としました.
戻り値の値は従来通りです.

◆VB
”Ｂｏｏｌｅａｎ”の定義を”ｌｏｎｇ”としました.
戻り値の値は従来通りです.

■Ｃ言語用ヘッダーファイルの統一

◆ドライバＩ/Ｆ　ＤＬＬ結合用ヘッダーファイルを１つとしました.