

# HPCI-PPD553A 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

H P C I－PPD553Aシリーズのボードは  
H P C I－PPD552  
H P C I－PPD553  
および  
H P C I－PPD552A  
H P C I－PPD553A です。

Windows版デバイスドライバ の種別は

Windows95, 98	ではh i p p d 5 5 3. v x d
WindowsNT	ではh i p p d 5 5 3. s y s
Windows2000	ではh p 5 5 3 w 2 k. s y s
WindowsXP	ではh p 5 5 3 w x p. s y s
WindowsVista, 7 (32ビット)	ではh p 5 5 3 w d m. s y s
WindowsVista, 7 (64ビット)	ではh p 5 5 3 x 6 4. s y s

です。

Windows版ドライバI/Fライブラリ として  
h i p p d 5 5 3. d l lを上記OS共通で使します。

---

本書及びプログラムの全部又は一部の無断転載、コピーを禁止します。  
本シリーズ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。  
本シリーズ製品の内容についてお気づきの点がございましたら、お手数ながら当社までご連絡下さい。

Windows 7, Windows Vista, Windows XP Home Edition, Windows XP Professional, Windows 2000 Professional, Windows 98, Windows 95, VisualC++, Visual Basic, Visual Basic.NETはMicrosoft Corporation の米国及びその他の国における登録商標です。その他、記載されている会社名、製品名は、各社の商標又は登録商標です。

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

第2.6版 2011年01月27日発行  
不許複製・転載

## 保証範囲

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

## 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N C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N C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

 <b>警 告</b>	
	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

 <b>警 告</b>	
	<p>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。</p> <p>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。</p>

### 5. 対象開発ツール

 <b>注 意</b>	
	<p>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。</p> <ul style="list-style-type: none"><li>・ Microsoft Visual C++ (6.0)</li><li>・ Microsoft Visual Basic (6.0 / .NET)</li></ul> <p>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。</p>

### 6. サンプルプログラムの実行

 <b>注 意</b>	
	<p>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。</p> <p>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。</p> <p>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。</p>

## 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あります。

プログラム作成・試運転時共、十分な注意をお願いいたします。

# 目 次

1. はじめに.....	1
1. 1 添付ソフトウェアの構成.....	1
1. 1. 1 32ビット版OS.....	1
1. 1. 2 64ビット版OS.....	2
1. 2 添付ディスクの内容.....	3
2. デバイスドライバのインストール.....	4
2. 1 Windows 9 Xへのインストール.....	4
2. 2 Windows NT 4.0へのインストール.....	4
2. 3 Windows 2000へのインストール.....	4
2. 4 Windows XPへのインストール.....	5
2. 5 Windows Vista(32ビット), Windows 7(32ビット)へのインストール.....	5
2. 6 Windows Vista(64ビット), Windows 7(64ビット)へのインストール.....	5
2. 7 アンインストール.....	5
2. 7. 1 Windows 9x, Windows NT 4.0, Windows 2000, Windows XP の場合.....	5
2. 7. 2 Windows Vista, Windows 7 の場合.....	5
3. ボードを複数枚使用する場合.....	6
3. 1 ボードのスロット番号の確認.....	6
3. 2 スロット番号の確認方法.....	6
3. 3 ボードID (ボードアドレス) の使用.....	6
4. ドライバI/F用DLLの使用法.....	7
4. 1 概 要.....	7
4. 2 関数一覧.....	7
4. 3 準 備.....	8
4. 4 制御概念.....	9
4. 4. 1 ボード(デバイス)認識用のデータ構造体.....	9
4. 4. 2 ボードアクセスの準備手順.....	10
4. 4. 3 デバイスドライバの異常報告.....	11
5. 関数詳細.....	12
6. サンプルプログラム.....	33
6. 1 サンプルプログラムの操作.....	33
6. 1. 1 ボード(デバイス)の各種操作.....	34
6. 1. 2 ボード上の各軸操作.....	34
6. 1. 3 軸動作の設定条件.....	37
6. 2 サンプルプログラムの変更について.....	38
6. 2. 1 プロジェクトファイル.....	38
6. 2. 2 サンプルプログラムを構成する関数.....	39
7. 「動かしてみる」プログラム.....	40
7. 1 「動かしてみる」の動作確認画面.....	40
7. 1. 1 デバイス情報の表示.....	40
7. 1. 2 個々の軸表示と動作指令.....	40
7. 2 「動かしてみる」の設定画面.....	44
7. 2. 1 ボード選択とデバイス情報.....	44
7. 2. 2 変更可能な軸動作条件.....	45
附A PCL5014レジスタの抜粋.....	47
1. PCL5014・レジスタの使用分類.....	47
2. 個別レジスタ.....	48
2. 1 環境レジスタ1 (R6).....	48
2. 2 環境レジスタ2 (R7).....	49
2. 3 環境レジスタ3 (R8).....	50
2. 4 カウンタモニタ (R12).....	50
2. 5 コマンドモニタ1 (R13).....	50
2. 6 コマンドモニタ2 (R14).....	50
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.....	51



## 図 表 目 次

表 1. 1-1	32ビット版OS添付ソフトウェア関連図.....	1
表 1. 1-2	64ビット版OS添付ソフトウェア関連図.....	2
図 3. 1-1	ボードを複数枚使用する場合.....	6
図 3. 3-1	HPCI-PPD553ボードIDの設定.....	6
表 4. 2-1	関数一覧.....	7
表 4. 4-1	異常内容一覧.....	11
表 5. 1-1	書込レジスタ不使用のコマンドデータ : SW = 0.....	18
表 5. 1-2	書込レジスタとコマンドデータ : SW = 1.....	18
表 5. 1-3	動作モードデータ.....	19
表 5. 1-4	制御モードデータ.....	20
表 5. 1-5	ステータス1.....	21
表 5. 1-6	ステータス2.....	22
表 5. 1-7	割込ステータス.....	23
表 5. 1-8	レジスタ読込コマンド」参照.....	25
表 6. 1-1	軸制御のレジスタ初期設定.....	37

## 1. はじめに

本書は、H P C I－P P D 5 5 3 A シリーズボード（PPD552, PPD553, PPD552A, PPD553A）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**P P D 5 5 x**と記します。）

この添付ソフトウェアはWindows 7, Windows Vista, Windows X P Professional・Home Edition(以降**X P**と記します), Windows 2 0 0 0 (以降**2 K**と記します), Windows N T 4. 0, 及びWindows 9 5・9 8 (以降**9 X**と記します)において、**P P D 5 5 x**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ズ・マニュアルを参照してください。

### 1. 1 添付ソフトウェアの構成

#### 1. 1. 1 3 2ビット版OS

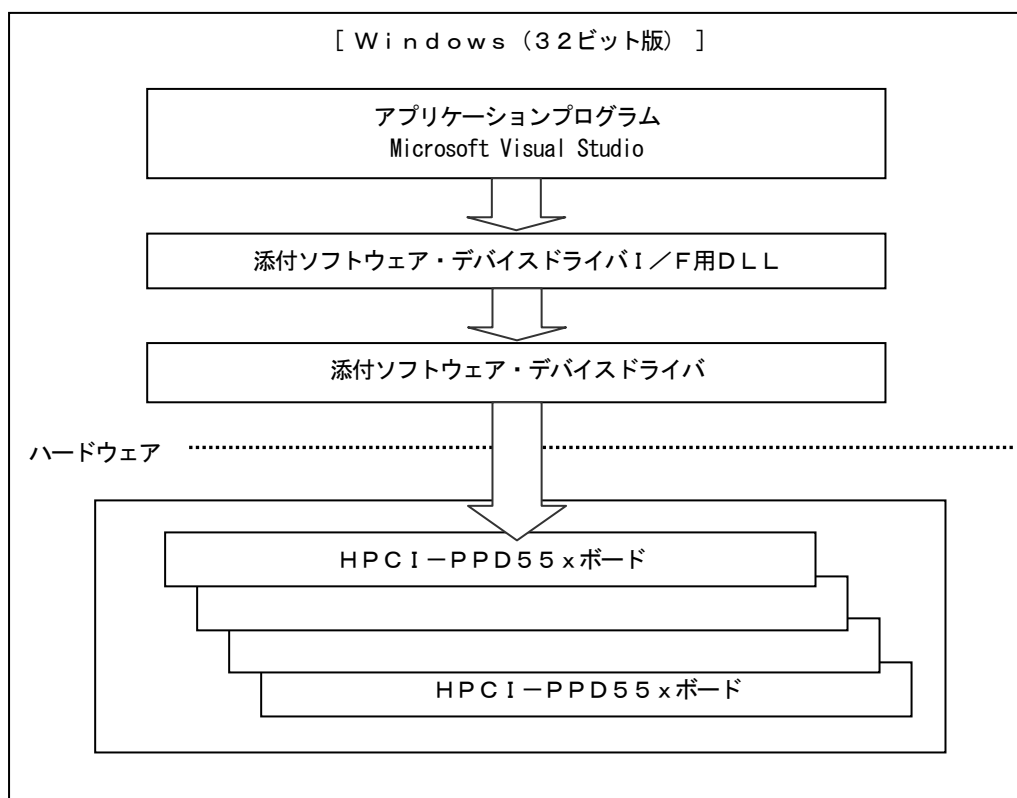


表1. 1－1 3 2ビット版OS添付ソフトウェア関連図

## 1. 1. 2 64ビット版OS

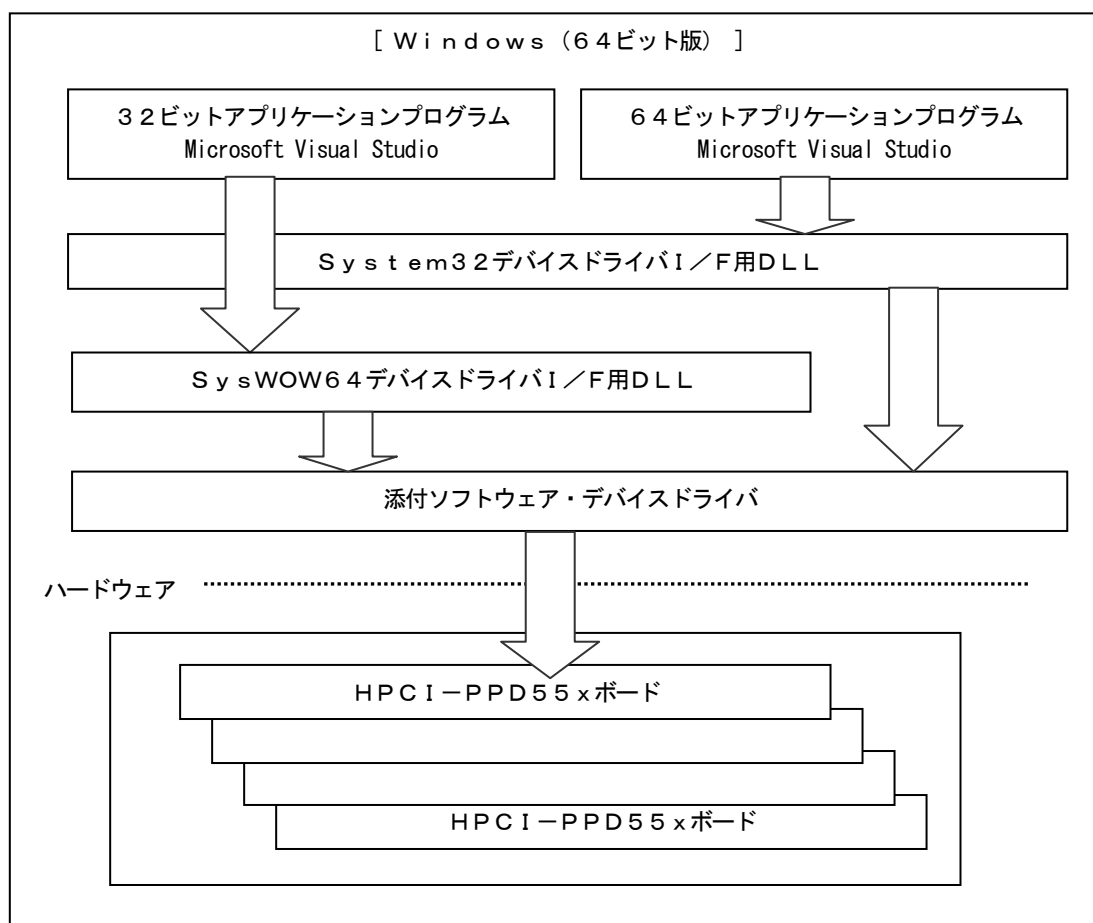


表1. 1-2 64ビット版OS添付ソフトウェア関連図

## 1. 2 添付ディスクの内容

hippd553.dll	....	ドライバ I/F 用 DLL
p553uins.exe	....	アンインストール用実行プログラム (Win9X, WinNT, Win2K 共通)
History.txt	....	添付ディスクの更新履歴
Win9x	hippd553.vxd	.... Windows9X 仮想デバイスドライバ
	hippd553.inf	.... Windows9X 用インストール情報 (VxD インストール用)
WinNt	hippd553.sys	.... WindowsNT4.0 デバイスドライバ
	p553inst.inf	.... インストール情報ファイル
	p553inst.bat	.... インストール用バッチファイル
Win2k	hp553w2k.sys	.... Windows2000 デバイスドライバ
	hp553w2k.inf	.... インストール情報ファイル
WinXP	hp553wXP.sys	.... WindowsXP デバイスドライバ
	hp553wXP.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	.... ドライバパッケージインストーラ用 xml ファイル
	hp553wdm.cat	.... セキュリティカタログファイル
	hp553wdm.sys	.... WindowsVista, 7 (32 ビット版) デバイスドライバ
	hp553wdm.inf	.... インストール情報ファイル
	hippd553.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	.... ドライバパッケージインストーラ用 xml ファイル
	hp553x64.cat	.... セキュリティカタログファイル
	hp553x64.sys	.... WindowsVista, 7 (64 ビット版) デバイスドライバ
	hp553x64.inf	.... インストール情報ファイル
	hippd553.dll	.... ドライバ I/F 用 DLL (32 ビット版)
	hippd553.x64	.... ドライバ I/F 用 DLL (64 ビット版)
document	PPD553_U_2_10J.pdf	.... ユーザーズマニュアル
	PPD553_W_2_60J.pdf	.... ソフトウェアマニュアル
Include	Vc_x86	hippd553.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)
		hippd553.h .... ヘッダーファイル (C アプリケーション構築用)
	Vc_x64	hippd553.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)
		hippd553.h .... ヘッダーファイル (C アプリケーション構築用)
	Vb6	hippd553.bas .... DLL 関数定義標準モジュール (VB 6.0 アプリケーション構築用)
	Vb_net	hippd553.vb .... DLL 関数定義ファイル (VB .NET アプリケーション構築用)
Sample	Vc	spp55300.exe .... サンプル実行ファイル (Visual C)
		spp55300.dsw .... Visual C 用 サンプル プロジェクトワークスペース
		spp55300.dsp .... サンプル プロジェクトファイル
		spp55300.opt .... サンプル opt ファイル
		spp55300.c .... Visual C 用 サンプル ソースプログラム
		spp55300.h .... サンプル ヘッダーファイル
		spp55300.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)
		bred.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)
		hippd553.xxx .... C 言語用 ドライバ添付ファイル 3 種 (xxx = lib, h, dll)
	Vcpp	spp55301.exe .... サンプル実行ファイル (VC++ (C++ コーディング) 版)
		spp55301.dsw .... VC++ (C++) 用 サンプル プロジェクトワークスペース
		spp55301.dsp .... サンプル プロジェクトファイル
		spp55301.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
		spp55301dlg.cpp .... VC++ (C++) 用 サンプル ソースプログラム
		spp55301dlg.h .... サンプル ヘッダーファイル
		hippd553.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)
		その他のファイル .... "readme.txt" ファイルを参照下さい。
	Vb6	spp55302.exe .... サンプル実行ファイル (Visual Basic 6.0)
		spp55302.vbp .... VB 用 サンプル プロジェクトファイル
		spp55302.frm .... サンプル フォームモジュールファイル
		spp55302Module.bas .... サンプル 標準モジュールファイル
		hippd553.bas .... ドライバ I/F 用 DLL 関数定義標準モジュールファイル
		hippd553.dll .... ドライバ I/F 用 DLL
	Vb_net	spp55303.sln .... ソリューションファイル
		spp55303.suo .... ソリューションユーザーオプション
		spp55303.vbproj .... プロジェクトファイル
		spp55303.vbproj.user .... プロジェクトユーザーオプション
		spp55303.resx .... フォームモジュールリソーステンプレート
		spp55303.vb .... フォームモジュールリソースファイル
		spp55303Module.vb .... 標準モジュールファイル
		AssemblyInfo.vb .... アセンブリ情報ファイル
		hippd553.vb .... ドライバ I/F 用 DLL 関数定義ファイル
	obj	Release
		spp55303.exe .... サンプル実行ファイル (Visual Basic .NET)
Test	hippd553.dll	.... ドライバ I/F 用 DLL
	tp55300.exe	.... 動かしてみる実行ファイル

## 2. デバイスドライバのインストール

### 2. 1 Windows 9 X 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、HPCI-PPD55x 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 Windows 9 X が起動すると、HPCI-PPD55x 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 Windows NT 4.0 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¥p553inst. 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¥p553inst. 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、Windows NT 起動時にHPCI-PPD55x 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-PPD553 (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。  
HPCI-PPD55x デバイスを再開させる場合も、コントロールパネルの「デバイス」操作を行います。  
「Hivertec HPCI-PPD553 (PCI)」を選択し、「開始」ボタンを押します。

### 2. 3 Windows 2000 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、HPCI-PPD55x 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 Windows 2000 が起動すると、HPCI-PPD55x 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-PPD553 (Win2k) Driver Disk 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¥hp553w2k. 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 Windows X Pへのインストール

※1. Windows X P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、H P C IーPPD55 x ボードをパソコンのP C I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 Windows X Pが起動すると、H P C IーPPD55 x 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 H i v e r t e c H P C IーPPD553 (WinXP) を選択します。
- ⑤ 「Windows 2000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 PPD55 x ボードをパソコンのP C I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、H P C IーPPD55 x をパソコンのP C I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 PPD55 x ボードをパソコンのP C I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、H P C IーPPD55 x をパソコンのP C I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

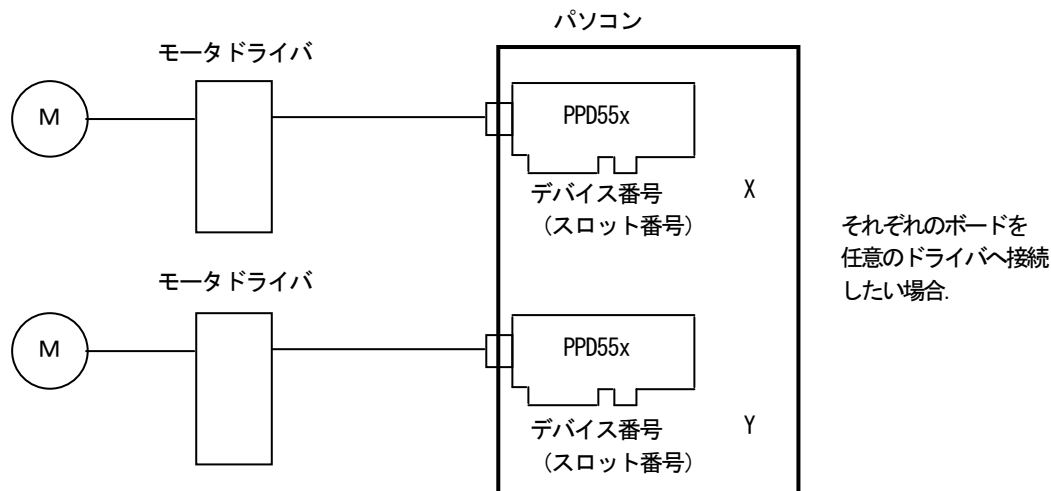
添付のCDをCDドライブに挿入します。  
エクスプローラを起動し、CDドライブ: ¥P553uins. exe を実行します。

### 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 H i v e r t e c H P C IーPPD55 x」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。



#### 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。

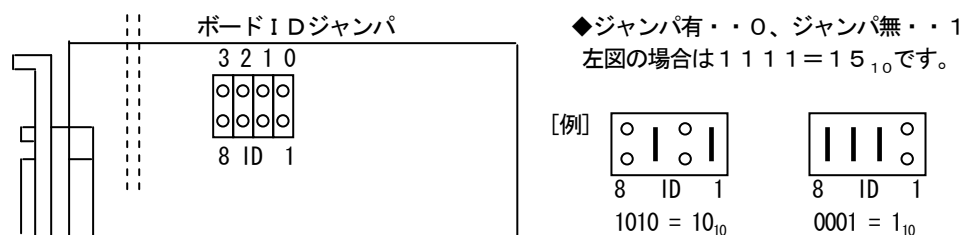
¥test¥tpp55300.exe . . . 2軸・3軸共通です。

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

PPD55x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。

- ①ボードID使用可能ボード・・・PPD552A～PPD553A
- ②ボードID固定(15)ボード・・・PPD552～PPD553



## 4 ドライバ I/F 用 DLL の使用方法

### 4. 1 概 要

Windows 用 DLL は、Windows V i s t a、Windows X P、Windows 2 K、Windows N T、Windows 9 X に  
おいて、H P C I - P P D 5 5 x 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 H P C I - P P D 5 5 x 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 Windows 9 X 用に hppd553.vxd、Windows N T 用に hppd553.sys、Windows 2 K 用に  
hp553w2k.sys、Windows X P 用に hp553wxp.sys、Windows V i s t a 用に hp553wdm.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 18 種類の関数が含まれます。

No	関 数 名 称	機 能	記載ページ
1	hppd553_GetDeviceCount()	ボード枚数の取得	11
2	hppd553_GetDeviceInfo()	デバイス情報の取得	12
3	hppd553_OpenDeviceID()	デバイスのオープン (ボード ID 参照)	13
4	hppd553_OpenDevice()	デバイスのオープン (ボード ID 無視)	14
5	hppd553_CloseDevice()	デバイスのクローズ	15
6	hppd553_CmdWrite()	コマンドバッファ書込	16
7	hppd553_DoutWrite()	動作モードバッファ書込	18
8	hppd553_SeiyoWrite()	制御モードバッファ書込	19
9	hppd553_status1Read()	ステータス 1 読込	20
10	hppd553_status2Read()	ステータス 2 読込	21
11	hppd553_intstatusRead()	割込ステータス読込	22
12	hppd553_regRead()	レジスタ読込	23
13	hppd553_ioswRead() hppd553_ioswWrite()	汎用入出力数切替ポート読込 汎用入出力数切替ポート書込	25 25
14	hppd553_inpReadB() hppd553_inpReadW()	汎用入力ポートバイト読込 汎用入力ポートワード読込	26 26
15	hppd553_outpReadB() hppd553_outpWriteB() hppd553_outpReadW() hppd553_outpWriteW()	汎用出力ポートバイト読込 汎用出力ポートバイト書込 汎用出力ポートワード読込 汎用出力ポートワード書込	27 27 27 27
16	hppd553_ElStRead() hppd553_ElStWrite()	E L S 極性選択ポート読込 E L S 極性選択ポート書込	29 29
17	hppd553_CmpcRead() hppd553_CmpcWrite()	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書込	30 30
18	hppd553_diRead() hppd553_diWrite()	汎用入力による同時スタート・ストップ 読込 汎用入力による同時スタート・ストップ 書込	31 31

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。

- ・hippd553.lib   ・インポートライブラリ
- ・hippd553.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 関数プロトタイプ宣言  
// _____  
#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。

<pre>(例) #define CPLUS 1 ..... #if CPLUS ..... #endif</pre>	<p>“1”でC++ 用（ファイル拡張子:.cpp） “0”でC 用（ファイル拡張子:.c）</p>
---	--

### [ VB6によるアプリケーションの構築 ]

標準モジュール「hippd553.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。

### [ VB.NETによるアプリケーションの構築 ]

標準モジュール「hippd553.vb」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 添付ソフトウェアは、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いません。
- ② ボードIDを使用する為、制御するPPD555x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 HPCI-PPD55x ボードを制御することができます。  
ある1つの HPCI-PPD55x 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
```

[ 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
```

- (注) 1. 管理番号はWindows 9 X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 HPCI-PPD55x のデバイス情報をまず取得します。

- ◆ hppd553\_GetDeviceCount() . . . ボード枚数の確認
- ◆ hppd553\_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 HPCI-PPD55x 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 HPCI-PPD55x 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。

- ◆ hppd553\_OpenDeviceID() . . . ボードのオープン処理 [ボード ID 参照]
- ◆ hppd553\_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 HPCI-PPD55x 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。

- ◆ hppd553\_ElStWrite() . . . ELS 極性の設定（極限リミットセンサ）
- ◆ hppd553\_CmpWrite() . . . コンパレータ同時スタート設定
- ◆ hppd553\_diWrite() . . . 汎用入力による同時スタート・ストップ 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。

- ◆ hppd553\_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、この異常内容を読み、異常内容に対応した処理を行います。

##### （1）異常内容の読み・・・GetLastError () 関数の起動

[ C言語: VC++ ]	[ VB6 ]	[ VB.NET ]
DWORD dErrorCode;	dErrorCode As Long;	dErrorCode As Integer
dErrorCode = GetLastError ();	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度に多数のウィンドウを開いた。
- 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	VB 6. 0, VB. NET
0x00000000	&H0
0x20001000	&H20001000

(1)	<b>hpd553_GetDeviceCount()</b> <b>ボード枚数の取得</b>
機能	現在パソコンに装着されている HPCI-PPD55x ボードの枚数を取得します。 PPD552~PPD553, PPD552A~PPD553A ボードの全枚数です。
書式	[ C 言語: VC++ ] DWORD WINAPI hpd553_GetDeviceCount( WORD dummy ); [ VB 6 ] Declare Function hpd553_GetDeviceCount Lib "hpd553.DLL" _ (ByVal dummy As Integer) As Long [ VB. NET ] Declare Function hpd553_GetDeviceCount Lib "hpd553.DLL" _ (ByVal dummy As short) As Integer
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	HPCI-PPD55x 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) .....HPCI-PPD55x 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 = hpd553_GetDeviceCount(1); [ VB 6 ] Dim count As Long count = hpd553_GetDeviceCount(1) [ VB. NET ] Dim count As Integer count = hpd553_GetDeviceCount(1)

(2)	<b>hppd553_GetDeviceInfo()</b> <b>デバイス情報の取得</b>
機能	現在パソコンに装着されている HPCI-PPD55x 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	<pre> [ C言語: VC++ ] DWORD WINAPI hppd553_GetDeviceInfo (DWORD* pcnDeviceNumber,                                      PHPCDEVICEINFO pHpcDeviceInfo);  [ VB6 ] Declare Function hppd553_GetDeviceInfo Lib "hppd553.DLL" _     (ByRef pcnDeviceNumber As Long, pHpcDeviceInfo As HPCDEVICEINFO) As Long  [ VB.NET ] Declare Function hppd553_GetDeviceInfo Lib "hppd553.DLL" _     (ByRef pcnDeviceNumber As Integer, pHpcDeviceInfo As HPCDEVICEINFO) As Integer </pre>
引数	<p>◆ 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。</p> <p>◆ 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。</p>
戻り値	<p>処理結果</p> <p>1 : 成功    .. 指令と戻りの枚数値を確認して下さい。</p> <p>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。</p>
呼び出し例	<p>パソコンに HPCI-PPD55x が2枚装着されていることを想定します。</p> <pre> [ 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 HPCI-PPD55x のデバイス情報が                                 //セットされるべきエリア  ret = hppd553_GetDeviceInfo(     &amp;count,                //count のアドレスを渡す。     &amp;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 = hppd553_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。  [ VB.NET ] Dim ret      As Integer Dim count    As Integer Dim HpcDeviceInfo(2) As HPCDEVICEINFO count = 2 ret = hppd553_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。 </pre>

(3)	<b>hppd553_OpenDeviceID()</b> <b>デバイスのオープン (ボードID参照)</b>
機能	渡した情報を持つ HPCI-PPD55x 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 HPCI-PPD55x にアクセスするためのハンドルとなります。
書式	<p>[ C 言語: VC++ ]</p> <pre>DWORD WINAPI hppd553_OpenDeviceID( PHPCDEVICEINFO pHpcDeviceInfo );</pre> <p>[ VB 6 ]</p> <pre>Declare Function hppd553_OpenDeviceID Lib "hppd553.DLL" _     (pHpcDeviceInfo As HPCDEVICEINFO) As Long</pre> <p>[ VB. NET ]</p> <pre>Declare Function hppd553_OpenDeviceID Lib "hppd553.DLL" _     (pHpcDeviceInfo As HPCDEVICEINFO) As Integer</pre>
引数	<p>◆ PHPCDEVICEINFO pHpcDeviceInfo</p> <p>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。</p>
戻り値	<p>デバイス ID 値</p> <p>INVALID_HANDLE_VALUE (=-1) : オープン失敗</p> <p>「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。</p> <p>上記以外 : オープン成功</p> <ul style="list-style-type: none"> <li>●上位ワード: ボードID (ボード上のジャンパ設定値: 0 ~ 15)</li> <li>●下位ワード: デバイスID (オープン順の番号 : 1 ~ )</li> </ul> <p>【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。</p> <p>VC++ [ hDeviceID &amp;= 0xffff0000; ]</p> <p>VB [ hDeviceID = hDeviceID And &amp;HFFF0000 ]</p>
呼び出し例	<p>パソコンに HPCI-PPD55x 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 hppd553_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。</p> <p>[ C 言語: VC++ ]</p> <pre>DWORD hDeviceID[2];            //デバイス ID 取得エリア hDeviceID[0] = hppd553_OpenDeviceID(     &amp;HpcDeviceInfo[0] ); // 1 番目のデバイス情報 hDeviceID[1] = hppd553_OpenDeviceID(     &amp;HpcDeviceInfo[1] ); // 2 番目のデバイス情報</pre> <p>[ VB 6 ]</p> <pre>Dim hDeviceID(2) As Long hDeviceID(0) = hppd553_OpenDeviceID(     HpcDeviceInfo(0) )       ' 1 番目のデバイス情報 hDeviceID(1) = hppd553_OpenDeviceID(     HpcDeviceInfo(1) )       ' 2 番目のデバイス情報</pre> <p>[ VB. NET ]</p> <pre>Dim hDeviceID(2) As Integer hDeviceID(0) = hppd553_OpenDeviceID(     HpcDeviceInfo(0) )       ' 1 番目のデバイス情報 hDeviceID(1) = hppd553_OpenDeviceID(     HpcDeviceInfo(1) )       ' 2 番目のデバイス情報</pre>

(4)	<b>hppd553_OpenDevice()</b> <b>デバイスのオープン (ボードID無視)</b>
機能	渡した情報を持つ HPCI-PPD55x 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 HPCI-PPD55x にアクセスするためのハンドルとなります。
書式	<p>[ C言語: VC++ ]  DWORD WINAPI hppd553_OpenDevice( PHPCDEVICEINFO pHpcDeviceInfo );</p> <p>[ VB6 ]  Declare Function hppd553_OpenDevice Lib "hppd553.DLL" _  (pHpcDeviceInfo As HPCDEVICEINFO) As Long</p> <p>[ V B . N E T ]  Declare Function hppd553_OpenDevice Lib "hppd553.DLL" _  (pHpcDeviceInfo As HPCDEVICEINFO) As Integer</p>
引数	<p>◆ 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。</p>
戻り値	<p>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</p> <p>●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 ~ )</p>
呼び出し例	<p>パソコンに HPCI-PPD55x 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 hppd553_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。</p> <p>[ C言語: VC++ ]  DWORD hDeviceID[2];                    //デバイス ID 取得エリア</p> <p>hDeviceID[0] = hppd553_OpenDevice(  &amp;HpcDeviceInfo[0] );    // 1 番目のデバイス情報  hDeviceID[1] = hppd553_OpenDevice(  &amp;HpcDeviceInfo[1] );    // 2 番目のデバイス情報</p> <p>[ VB6 ]  Dim hDeviceID(2) As Long</p> <p>hDeviceID(0) = hppd553_OpenDevice(  HpcDeviceInfo(0) )        ' 1 番目のデバイス情報  hDeviceID(1) = hppd553_OpenDevice(  HpcDeviceInfo(1) )        ' 2 番目のデバイス情報</p> <p>[ V B . N E T ]  Dim hDeviceID(2) As Integer</p> <p>hDeviceID(0) = hppd553_OpenDevice(  HpcDeviceInfo(0) )        ' 1 番目のデバイス情報  hDeviceID(1) = hppd553_OpenDevice(  HpcDeviceInfo(1) )        ' 2 番目のデバイス情報</p>



(5)	<b>hppd553_CloseDevice0      デバイスのクローズ</b>
機能	渡したデバイス ID で指定された HPCI-PPD55x 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	<pre> [ C 言語: VC++ ] DWORD WINAPI hppd553_CloseDevice( DWORD hDevID );  [ VB 6 ] Declare Function hppd553_CloseDevice Lib "hppd553.DLL" _     (ByVal hDevID As Long) As Long  [ VB. NET ] Declare Function hppd553_CloseDevice Lib "hppd553.DLL" _     "(ByVal hDevID As Long) As Integer </pre>
引数	◆ DWORD hDevID    ・ ・ クローズするボードのデバイス ID.
戻り値	<p>処理結果</p> <p>1 : 成功</p> <p>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.</p>
呼び出し例	<pre> [ C 言語: VC++ ] DWORD ret; //関数の戻り値 ret = hppd553_CloseDevice( hDeviceID ); //デバイス ID  [ VB 6 ] Dim ret As Long ret = hppd553_CloseDevice( hDeviceID ) ' デバイス ID  [ VB. NET ] Dim ret As Integer ret = hppd553_CloseDevice( hDeviceID ) ' デバイス ID </pre>

(6)	<b>hppd553_CmdWrite0 コマンドバッファ書込</b>
機能	<p>①デバイス ID で指定された HPCI-PPD55x 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。</p> <p>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。</p>
書式	<p>[ C言語: VC++ ]</p> <pre>DWORD WINAPI hppd553_CmdWrite( DWORD hDevID, WORD AxisNo, BYTE byData,                                WORD sw, DWORD dwReg );</pre> <p>[ VB6 ]</p> <pre>Declare Function hppd553_CmdWrite Lib "hppd553.DLL" _     ( ByVal hDevID As Long, ByVal AxisNo As Integer, _       ByVal byData As Byte, ByVal sw As Integer, _       ByVal dwReg As Long) As Long</pre> <p>[ VB.NET ]</p> <pre>Declare Function hppd553_CmdWrite Lib "hppd553.DLL" _     ( ByVal hDevID As Integer, ByVal AxisNo As short, _       ByVal byData As Byte, ByVal sw As short, _       ByVal dwReg As Integer) As Integer</pre>
引数	<p>◆ DWORD hDevID・・・対象デバイスのデバイス ID.</p> <p>◆ WORD AxisNo・・・軸指定 [ 0 : X軸, 1 : Y軸, 2 : Z軸 ]</p> <p>◆ BYTE byData・・・コマンドデータ</p> <p>「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</p> <p>◆ WORD sw</p> <p>= 0 : レジスタ書込をしない。 ≠ 0 : レジスタ書込を行う。</p> <p>◆ DWORD dwReg・・・レジスタへの書込データ</p>
戻り値	<p>処理結果</p> <p>1 : 成功</p> <p>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。</p>
呼び出し例	<p>[ C言語: VC++ ]</p> <pre>DWORD ret; //関数の戻り値 ret = hppd553_CmdWrite( hDeviceID, //デバイス ID                         0, //X軸を指定                         0x60, //（これは「ソフトリセット」コマンド）                         0, //レジスタ書込ではない。                         0 );</pre> <p>[ VB6 ]</p> <pre>Dim ret As Long '関数の戻り値 ret = hppd553_CmdWrite( hDeviceID, 'デバイス ID                         0, 'X軸を指定                         &amp;h60, '（これは「ソフトリセット」コマンド）                         0, 'レジスタ書込ではない。                         0 )</pre> <p>[ VB.NET ]</p> <pre>Dim ret As Integer '関数の戻り値 ret = hppd553_CmdWrite( hDeviceID, 'デバイス ID                         0, 'X軸を指定                         &amp;h60, '（これは「ソフトリセット」コマンド）                         0, 'レジスタ書込ではない。                         0 )</pre>

コマンドデータ	コマンド内容	コマンドデータ	コマンド内容
0x58	SVON ON	0x00	瞬時にR 1速度に変更
0x48	SVON OFF	0x01	瞬時にR 2速度に変更
0x59	SVRESET ON	0x02	減速してR 1速度に変更
0x49	SVRESET OFF	0x03	加速してR 2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	ソフトウェア 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	コマンドデータ	レジスタ内容	ビット数
PRO	0xc0	送り量	4
PR 1	0xc1	FL (PR 1 × 倍率 (PR 4)) 速度	2
PR 2	0xc2	FH (PR 2 × 倍率 (PR 4)) 速度	2
PR 3	0xc3	加減速レート	2
PR 4	0xc4	倍率	2
PR 5	0xc5	減速点	3
R 6	0xc6	環境レジスタ 1	4
R 7	0xc7	環境レジスタ 2	4
R 8	0xc8	環境レジスタ 3	2
R 9	0xc9	アップ/ダウンカウント値	4
R 1 0	0xca	コンパレータ 1 データ	4
R 1 1	0xcb	コンパレータ 2 データ	4
R 1	0xd1	動作中のFL (R 1 × 倍率) 速度変更	2
R 2	0xd2	動作中のFH (R 2 × 倍率) 速度変更	2
R 3	0xd3	動作中の加減速レート変更	2
R 5	0xd4	動作中のスローダウンポイント変更	3
PR 1 5	0xd8	減速レート	2
PR 1 6	0xd9	S字区間	2
R 1 5	0xda	動作中の減速レート変更	2
R 1 6	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)	hppd553_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 HPCI-PPD55x 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 hppd553_DousaWrite(DWORD hDevID, WORD AxisNo, BYTE byData ); [ VB 6 ] Declare Function hppd553_DousaWrite Lib "hppd553.DLL" _ ( ByVal hDevID As Long, ByVal AxisNo As Integer, ByVal byData As Byte) As Long [ VB. NET ] Declare Function hppd553_DousaWrite Lib "hppd553.DLL" _ ( ByVal hDevID As Integer, ByVal AxisNo As short, ByVal byData As Byte) As Integer
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 ] ◆ 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 = hppd553_DousaWrite( hDeviceID, //デバイス ID 0, //X軸を指定 0x20 ); // (動作モードデータ参照) [ VB 6 ] Dim ret As Long '関数の戻り値 ret = hppd553_DousaWrite( hDeviceID, 'デバイス ID 0, 'X軸を指定 &H20 ) ' (動作モードデータ参照) [ VB. NET ] Dim ret As Integer '関数の戻り値 ret = hppd553_DousaWrite( hDeviceID, 'デバイス ID 0, 'X軸を指定 &H20 ) ' (動作モードデータ参照)

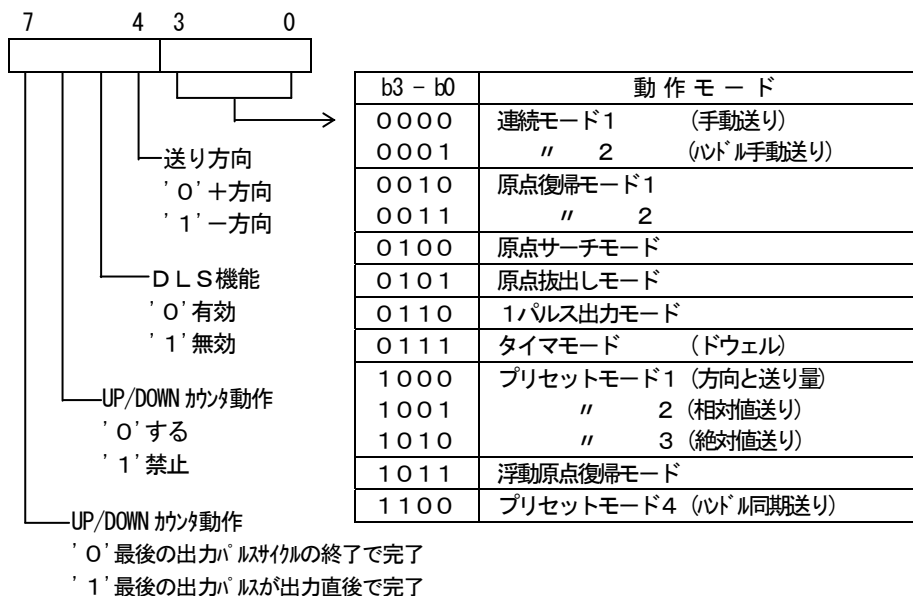


表 5. 1-3 動作モードデータ

(8)	<b>hppd553_SeigoWrite0 制御モードバッファ書込</b>
機能	デバイス ID で指定された HPCI-PPD55x 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 hppd553_SeigoWrite(DWORD hDevID, WORD AxisNo, BYTE byData ); [ VB 6 ] Declare Function hppd553_SeigoWrite Lib "hppd553.DLL" _ ( ByVal hDevID As Long, ByVal AxisNo As Integer, ByVal byData As Byte) As Long [ VB. NET ] Declare Function hppd553_SeigoWrite Lib "hppd553.DLL" _ ( ByVal hDevID As Integer, ByVal AxisNo As Short, ByVal byData As Byte) As Integer
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 ] ◆ 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 = hppd553_SeigoWrite( hDeviceID, //デバイス ID 1, //Y 軸を指定 0x04 ); // (制御モードデータ参照) [ VB 6 ] Dim ret As Long '関数の戻り値 ret = hppd553_SeigoWrite( hDeviceID, 'デバイス ID 1, 'Y 軸を指定 &H4 ) ' (制御モードデータ参照) [ VB. NET ] Dim ret As Integer '関数の戻り値 ret = hppd553_SeigoWrite( hDeviceID, 'デバイス ID 1, 'Y 軸を指定 &H4 ) ' (制御モードデータ参照)

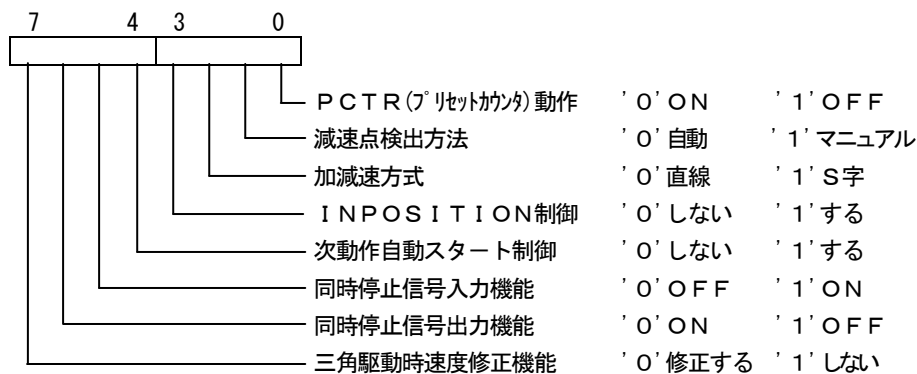


表 5. 1-4 制御モードデータ

(9)	<b>hppd553_status1Read0 ステータス1読み</b>
機能	デバイス ID で指定された HPCI-PPD55x の, AxisNo で指定された軸のステータス1を読み、指定した1バイトのエリアに格納します。
書式	<p>[ C言語: VC++ ]</p> <p>DWORD WINAPI hppd553_status1Read(DWORD hDevID, WORD AxisNo, BYTE* bySts1 );</p> <p>[ VB6 ]</p> <p>Declare Function hppd553_status1Read Lib "hppd553.DLL" _</p> <p>( ByVal hDevID As Long, ByVal AxisNo As Integer, ByRef bySts1 As Byte) As Long</p> <p>[ VB.NET ]</p> <p>Declare Function hppd553_status1Read Lib "hppd553.DLL" _</p> <p>( ByVal hDevID As Integer, ByVal AxisNo As Short, ByRef bySts1 As Byte) As Integer</p>
引数	<p>◆ DWORD hDevID・・・対象デバイスのデバイス ID.</p> <p>◆ WORD AxisNo・・・軸指定 [ 0: X軸, 1: Y軸, 2: Z軸 ]</p> <p>◆ BYTE* bySts1・・・読み出されたステータス1の格納先 (「表5. 1ー5 ステータス1」参照)</p>
戻り値	<p>処理結果</p> <p>1: 成功</p> <p>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。</p>
呼び出し例	<p>[ C言語: VC++ ]</p> <p>DWORD ret; //関数の戻り値</p> <p>BYTE sts1;</p> <p>ret = hppd553_status1Read( hDeviceID, //デバイス ID</p> <p>1, //Y軸を指定</p> <p>&amp;sts1 ); //格納先のアドレス</p> <p>[ VB6 ]</p> <p>Dim ret As Long '関数の戻り値</p> <p>Dim sts1 As Byte</p> <p>ret = hppd553_status1Read( hDeviceID, 'デバイス ID</p> <p>1, 'Y軸を指定</p> <p>sts1 ) '格納先のアドレス</p> <p>[ VB.NET ]</p> <p>Dim ret As Integer '関数の戻り値</p> <p>Dim sts1 As Byte</p> <p>ret = hppd553_status1Read( hDeviceID, 'デバイス ID</p> <p>1, 'Y軸を指定</p> <p>sts1 ) '格納先のアドレス</p>

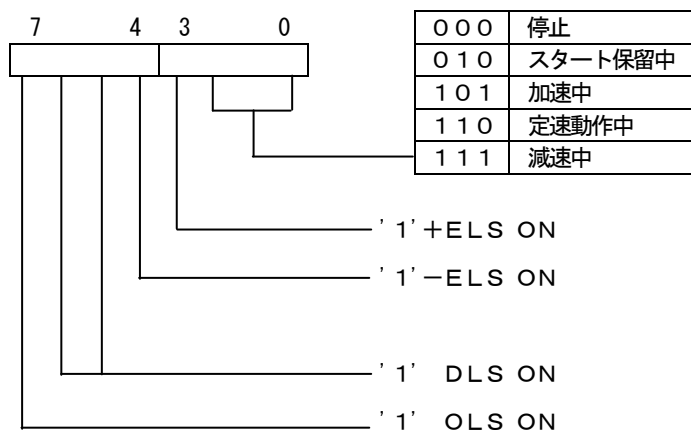


表5. 1ー5 ステータス1

(10)	<b>hppd553_status2Read0 ステータス2読み</b>
機能	デバイス ID で指定された HPCI-PPD55x の, AxisNo で指定された軸のステータス 2 を読み、指定した 1 バイトのエリアに格納します。
書式	<pre> [ C 言語: VC++ ] DWORD WINAPI hppd553_status2Read( DWORD hDevID, WORD AxisNo, BYTE* bySts2 );  [ VB 6 ] Declare Function hppd553_status2Read Lib "hppd553.DLL" _     ( ByVal hDevID As Long, ByVal AxisNo As Integer, ByRef bySts2 As Byte) As Long  [ VB. NET ] Declare Function hppd553_status2Read Lib "hppd553.DLL" _     ( ByVal hDevID As Integer, ByVal AxisNo As Short, ByRef bySts2 As Byte) As Integer </pre>
引数	<ul style="list-style-type: none"> <li>◆ DWORD hDevID・・・対象デバイスのデバイス ID.</li> <li>◆ WORD AxisNo・・・軸指定 [ 0: X軸, 1: Y軸, 2: Z軸 ]</li> <li>◆ 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)</li> </ul>
戻り値	<p>処理結果</p> <p>1: 成功</p> <p>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。</p>
呼び出し例	<pre> [ C 言語: VC++ ] DWORD ret; //関数の戻り値 BYTE sts2; ret = hppd553_status1Read( hDeviceID, //デバイス ID                            1,          //Y 軸を指定                            &amp;sts2 );   //格納先のアドレス  [ VB 6 ] Dim ret As Long ' 関数の戻り値 Dim sts2 As Byte ret = hppd553_status1Read( hDeviceID, ' デバイス ID                            1,          ' Y 軸を指定                            sts2 )      ' 格納先のアドレス  [ VB. NET ] Dim ret As Integer ' 関数の戻り値 Dim sts2 As Byte ret = hppd553_status1Read( hDeviceID, ' デバイス ID                            1,          ' Y 軸を指定                            sts2 )      ' 格納先のアドレス </pre>

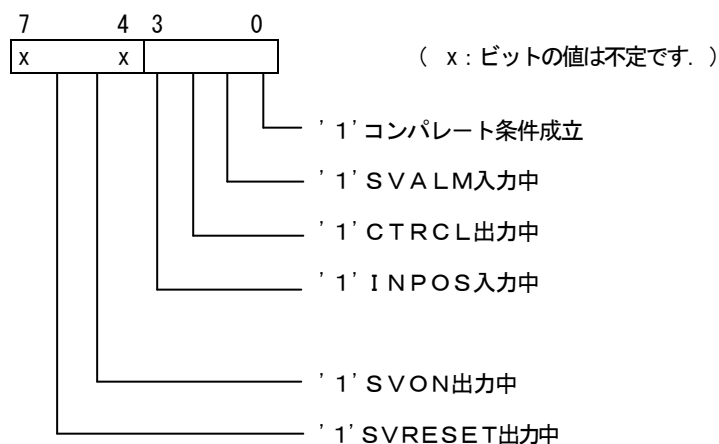


表 5. 1-6 ステータス 2

(11)	hppd553_intstatusRead() 割込ステータス読込
機能	デバイス ID で指定された HPCI-PPD55x の、AxisNo で指定された軸の割込ステータスを読み、指定した1バイトのエリアに格納します。
書式	<pre> [ C言語: VC++ ] DWORD WINAPI hppd553_intstatusRead(DWORD hDevID, WORD AxisNo, BYTE* byIntsts); [ VB6 ] Declare Function hppd553_intstatusRead Lib "hppd553.DLL" _     ( ByVal hDevID As Long, ByVal AxisNo As Integer, ByRef byIntsts As Byte) As Long [ VB.NET ] Declare Function hppd553_intstatusRead Lib "hppd553.DLL" _     ( ByVal hDevID As Integer, ByVal AxisNo As Short, ByRef byIntsts As Byte) As Integer </pre>
引数	<ul style="list-style-type: none"> <li>◆ DWORD hDevID・・・対象デバイスのデバイス ID.</li> <li>◆ WORD AxisNo・・・軸指定 [ 0: X軸, 1: Y軸, 2: Z軸 ]</li> <li>◆ 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)</li> </ul>
戻り値	<p>処理結果</p> <p>1: 成功</p> <p>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。</p>
呼び出し例	<pre> [ C言語: VC++ ] DWORD ret; //関数の戻り値 BYTE intsts; ret = hppd553_intstatusRead( hDeviceID, //デバイス ID                              1, //Y軸を指定                              &amp;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 = hppd553_intstatusRead( hDeviceID, 'デバイス ID                              1, 'Y軸を指定                              intsts ) '格納先のアドレス  [ VB.NET ] Dim ret As Integer '関数の戻り値 Dim intsts As Byte ret = hppd553_intstatusRead( hDeviceID, 'デバイス ID                              1, 'Y軸を指定                              intsts ) '格納先のアドレス </pre>

ステータス	要 因	割込有効 (マスクビット)
0x00	割込要因なし	
0x01	減速停止コマンド (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 2, フリリセットモード 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)	hppd553_regRead() レジスタ読込
機能	<p>デバイス ID で指定された HPCI-PPD55x 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.</p> <p>byDataに「レジスタ読込コマンドの値」をセットします.</p> <p>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.</p>
書式	<pre>[ C言語: VC++ ] DWORD WINAPI hppd553_regRead( DWORD hDevID, WORD AxisNo, BYTE byData, DWORD* dwReg );  [ VB6 ] Declare Function hppd553_regRead Lib "hppd553.DLL" _     ( ByVal hDevID As Long, ByVal AxisNo As Integer,       ByVal byData As Byte, ByRef dwReg As Long) As Long  [ VB. NET ] Declare Function hppd553_regRead Lib "hppd553.DLL" _     ( ByVal hDevID As Integer, ByVal AxisNo As Short,       ByVal byData As Byte, ByRef dwReg As Integer) As Integer</pre>
引数	<ul style="list-style-type: none"> <li>◆ DWORD hDevID・・・対象デバイスのデバイス ID.</li> <li>◆ WORD AxisNo・・・軸指定 [ 0: X軸, 1: Y軸, 2: Z軸 ]</li> <li>◆ BYTE byData・・・レジスタ読込コマンドの値 (「表5. 1ー8 レジスタ読込コマンド」参照)</li> <li>◆ 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</li> </ul>
戻り値	<p>処理結果</p> <p>1: 成功</p> <p>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.</p>
呼び出し例	<pre>[ C言語: VC++ ] DWORD ret; //関数の戻り値 DWORD regstr; ret = hppd553_regRead( hDeviceID, //デバイス ID                       0,           //X軸を指定                       0x80,        //R0 の内容を読む                       &amp;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 = hppd553_regRead( hDeviceID, _ ' デバイス ID                       0, _           ' X軸を指定                       &amp;h80, _        ' R0 の内容を読む                       regstr )       ' 格納先のアドレス  [ VB. NET ] Dim ret As Integer ' 関数の戻り値 Dim regstr As Integer ret = hppd553_regRead( hDeviceID, _ ' デバイス ID                       0, _           ' X軸を指定                       &amp;h80, _        ' R0 の内容を読む                       regstr )       ' 格納先のアドレス</pre>

読出レジスタ	レジスタ読込コマンド	レジスタ内容
R0	0 x 8 0	送り量
R1	0 x 8 1	FL (R1 x 倍率) 速度
R2	0 x 8 2	FH (R2 x 倍率) 速度
R3	0 x 8 3	加減速レート
R4	0 x 8 4	倍率
R5	0 x 8 5	減速点
R6	0 x 8 6	環境レジスタ 1
R7	0 x 8 7	環境レジスタ 2
R8	0 x 8 8	環境レジスタ 3
R9	0 x 8 9	アップ/ダウンカウンタ
R10	0 x 8 a	コンパレータ 1 データ
R11	0 x 8 b	コンパレータ 2 データ
R12	0 x 8 c	カウンタモニタ
R13	0 x 8 d	コマンドモニタ 1
R14	0 x 8 e	コマンドモニタ 2
PR0	0 x 9 0	次動作用プリセット量
PR1	0 x 9 1	次動作用 FL (PR1 x 倍率) 速度
PR2	0 x 9 2	次動作用 FH (PR2 x 倍率) 速度
PR3	0 x 9 3	次動作用加減速レート
PR4	0 x 9 4	次動作用倍率
PCTR	0 x 9 5	プリセットカウンタ カウント値
SCTR	0 x 9 6	減速点カウンタ カウント値
PR5	0 x 9 7	次動作用スローダウンポインタ
PR15	0 x 9 8	次動作用減速レート
PR16	0 x 9 9	次動作用 S 字区間
R15	0 x 9 a	減速レート
R16	0 x 9 b	S 字区間

表 5. 1-8 レジスタ読込コマンド」参照

(13)	<div>hppd553_ioswRead()</div> <div>汎用入出力数切替ポート読込</div> <div>hppd553_ioswWrite()</div> <div>汎用入出力数切替ポート書込</div>												
機能	<div>デバイス ID で指定された HPCI-PPD55x の,</div> <div>汎用入出力数切替ポート読込・・汎用入出力数切替ポートを読み、指定エリアに格納します。</div> <div>汎用入出力数切替ポート書込・・汎用入出力数切替ポートに指定データを書込みます。</div>												
書式	<div>[ C言語: VC++ ]</div> <div>DWORD WINAPI hppd553_ioswRead ( DWORD hDevID, BYTE* byIoSw );</div> <div>DWORD WINAPI hppd553_ioswWrite( DWORD hDevID, BYTE byData );</div> <div>[ VB6 ]</div> <div>Declare Function hppd553_ioswRead Lib "hppd553.DLL" _</div> <div>( ByVal hDevID As Long, ByRef byIoSw As Byte ) As Long</div> <div>Declare Function hppd553_ioswWrite Lib "hppd553.DLL" _</div> <div>( ByVal hDevID As Long, ByVal byData As Byte ) As Long</div> <div>[ VB.NET ]</div> <div>Declare Function hppd553_ioswRead Lib "hppd553.DLL" _</div> <div>( ByVal hDevID As Integer, ByRef byIoSw As Byte ) As Integer</div> <div>Declare Function hppd553_ioswWrite Lib "hppd553.DLL" _</div> <div>( ByVal hDevID As Integer, ByVal byData As Byte ) As Integer</div>												
引数	<div>◆ DWORD hDevID ・・対象デバイスのデバイス ID.</div> <div>◆ BYTE* byIoSw ・・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</div> <div>◆ BYTE byData ・・汎用入出力数切替ポートへの書込データ</div> <div><table><tr><td>7</td><td>4</td><td>3</td><td>0</td></tr><tr><td>x x x</td><td>x x x x</td><td></td><td></td></tr><tr><td>0 0 0</td><td>0 0 0 0</td><td></td><td></td></tr></table><div>読込 ( x : ビットの値は不定です. )</div><div>書込</div><div>0:16IN/8OUT, 1:12IN/12OUT</div></div>	7	4	3	0	x x x	x x x x			0 0 0	0 0 0 0		
7	4	3	0										
x x x	x x x x												
0 0 0	0 0 0 0												
戻り値	<div>処理結果</div> <div>1 : 成功</div> <div>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。</div>												
呼び出し例	<div>[ C言語: VC++ ]</div> <div>DWORD ret; //関数の戻り値</div> <div>BYTE IoSw;</div> <div>ret = hppd553_ioswRead ( hDeviceID, //デバイス ID</div> <div>&amp;IoSw ); //汎用入出力数切替ポートの読出し値の格納先</div> <div>ret = hppd553_ioswWrite( hDeviceID, //デバイス ID</div> <div>0x80 ); //12IN/12OUT</div> <div>[ VB6 ]</div> <div>Dim ret As Long</div> <div>Dim IoSw As Byte</div> <div>ret = hppd553_ioswRead ( hDeviceID, ' デバイス ID</div> <div>IoSw ) ' 汎用入出力数切替ポートの読出し値</div> <div>ret = hppd553_ioswWrite( hDeviceID, ' デバイス ID</div> <div>&amp;H80 ) ' 12IN/12OUT</div> <div>[ VB.NET ]</div> <div>Dim ret As Integer</div> <div>Dim IoSw As Byte</div> <div>ret = hppd553_ioswRead ( hDeviceID, ' デバイス ID</div> <div>IoSw ) ' 汎用入出力数切替ポートの読出し値</div> <div>ret = hppd553_ioswWrite( hDeviceID, ' デバイス ID</div> <div>&amp;H80 ) ' 12IN/12OUT</div>												

(14)	<b>hppd553_inpReadB()</b> 汎用入力ポートバイト読込 <b>hppd553_inpReadW()</b> 汎用入力ポートワード読込
機能	デバイス ID で指定された HPCI-PPD55x の、 <b>汎用入力ポートバイト読込</b> ・・・指定した入力ポートを読み込み、指定エリアに格納します。 <b>汎用入力ポートワード読込</b> ・・・入力ポート 1・2 を読み込み、指定エリアに格納します。
書式	<pre> [ C言語: VC++ ] DWORD WINAPI hppd553_inpReadB( DWORD hDevID, WORD PortNo, BYTE* byInp ); DWORD WINAPI hppd553_inpReadW( DWORD hDevID, WORD* wInp );  [ VB6 ] Declare Function hppd553_inpReadB Lib "hppd553.DLL"     ( ByVal hDevID As Long, ByVal PortNo As Integer, ByRef byInp As Byte ) As Long Declare Function hppd553_inpReadW Lib "hppd553.DLL"     ( ByVal hDevID As Long, ByRef wInp As Integer ) As Long  [ VB.NET ] Declare Function hppd553_inpReadB Lib "hppd553.DLL"     ( ByVal hDevID As Integer, ByVal PortNo As Short, ByRef byInp As Byte ) As Integer Declare Function hppd553_inpReadW Lib "hppd553.DLL"     ( ByVal hDevID As Integer, ByRef wInp As Short ) As Integer </pre>
引数	<ul style="list-style-type: none"> <li>◆ <b>DWORD hDevID</b>・・・対象デバイスのデバイス ID。</li> <li>◆ <b>WORD PortNo</b>・・・ポート番号指定 [ 0 : 入力ポート 1, 1 : 入力ポート 2 ]</li> <li>◆ <b>BYTE* byInp</b>・・・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</li> <li>◆ <b>WORD* wInp</b>・・・読込んだデータが格納される 2 バイトエリアのアドレス</li> </ul> <div style="text-align: center;"> <p>16 IN / 8 OUT 12 IN / 12 OUT</p> <p>読込</p> <p>ワード読込</p> </div> <p>(注) ① 枠内の数字は汎用入力信号 (DI) 1～16 です。          ② 無効となるビットの入力値は不定です。</p>
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	<pre> [ C言語: VC++ ] DWORD ret;          //関数の戻り値 BYTE  byInp; WORD  wInp; ret = hppd553_inpReadB( hDeviceID, //デバイス ID                         0,          //入力ポート 1                         &amp;byInp ); //格納先のアドレス ret = hppd553_inpReadW( hDeviceID, //デバイス ID                         &amp;wInp );    //格納先のアドレス  [ VB6 ] Dim ret As Long      '関数の戻り値 Dim byInp As Byte Dim wInp As Integer ret = hppd553_inpReadB( hDeviceID, _ 'デバイス ID                         0, _          '入力ポート 1                         byInp )      '格納先のアドレス ret = hppd553_inpReadW( hDeviceID, _ 'デバイス ID                         wInp )        '格納先のアドレス  [ VB.NET ] Dim ret As Integer   '関数の戻り値 Dim byInp As Byte Dim wInp As short ret = hppd553_inpReadB( hDeviceID, _ 'デバイス ID                         0, _          '入力ポート 1                         byInp )      '格納先のアドレス ret = hppd553_inpReadW( hDeviceID, _ 'デバイス ID                         wInp )        '格納先のアドレス </pre>

(15)	hppd553_outpReadB()      汎用出力ポートバイト読込 hppd553_outpWriteB()      汎用出力ポートバイト書込 hppd553_outpReadW()      汎用出力ポートワード読込 hppd553_outpWriteW()      汎用出力ポートワード書込
機能	デバイス ID で指定された HPCI-PPD55x の、 汎用出力ポートバイト読込・・指定した出力ポートを読み込み、指定エリアに格納します。 汎用出力ポートバイト書込・・指定した出力ポートに指定データを書込みます。 汎用出力ポートワード読込・・出力ポート 1・2 を読み込み、指定エリアに格納します。 汎用出力ポートワード書込・・出力ポート 1・2 に指定データを書込みます。
書式	<pre>[ C言語: VC++ ] DWORD WINAPI hppd553_outpReadB ( DWORD hDevID, WORD PortNo, BYTE* byOutp ); DWORD WINAPI hppd553_outpWriteB( DWORD hDevID, WORD PortNo, BYTE byData ); DWORD WINAPI hppd553_outpReadW ( DWORD hDevID, WORD* wOutp ); DWORD WINAPI hppd553_outpWriteW( DWORD hDevID, WORD wData );  [ VB 6 ] Declare Function hppd553_outpReadB Lib "hppd553.DLL"     ( ByVal hDevID As Long, ByVal PortNo As Integer, ByRef byOutp As Byte ) As Long Declare Function hppd553_outpWriteB Lib "hppd553.DLL"     ( ByVal hDevID As Long, ByVal PortNo As Integer, ByVal byData As Byte ) As Long Declare Function hppd553_outpReadW Lib "hppd553.DLL"     ( ByVal hDevID As Long, ByRef wOutp As Integer ) As Long Declare Function hppd553_outpWriteW Lib "hppd553.DLL"     ( ByVal hDevID As Long, ByVal wData As Integer ) As Long  [ VB. NET ] Declare Function hppd553_outpReadB Lib "hppd553.DLL"     ( ByVal hDevID As Integer, ByVal PortNo As Short, ByRef byOutp As Byte ) As Integer Declare Function hppd553_outpWriteB Lib "hppd553.DLL"     ( ByVal hDevID As Integer, ByVal PortNo As Short, ByVal byData As Byte ) As Integer Declare Function hppd553_outpReadW Lib "hppd553.DLL"     ( ByVal hDevID As Integer, ByRef wOutp As Short ) As Integer Declare Function hppd553_outpWriteW Lib "hppd553.DLL"     ( ByVal hDevID As Integer, ByVal wData As Short ) As Integer</pre>
引数	<p>◆ DWORD hDevID・・対象デバイスのデバイス ID.          ◆ WORD PortNo・・ポート番号指定 [ 0 : 出力ポート 1, 1 : 出力ポート 2 ]          ◆ BYTE* byOutp・・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          ◆ WORD* wOutp・・読込んだデータが格納される 2 バイトエリアのアドレス          ◆ BYTE byData・・出力ポートへの書込データ          ◆ WORD wData・・出力ポートへの書込データ</p> <div style="text-align: center;"> <p>PortNo = 1 (バイト 2)      PortNo = 0 (バイト 1)</p> <p>ワード読込・書込</p> </div> <p>(注) ① 枠内の数字は汎用出力信号 (D0) 1 ~ 12 です。          ② 無効となる出力ポートのビットの値は 0 とし、入力の場合は不定です。</p>
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	<pre> [ C言語: VC++ ] DWORD    ret;           //関数の戻り値 BYTE      byOutp; WORD      wOutp; ret = hppd553_outpReadB ( hDeviceID,    //デバイス ID                           0,             //出力ポート 1                           &amp;byOutp );    //格納先のアドレス ret = hppd553_outpWriteB( hDeviceID,    //デバイス ID                           0,             //出力ポート 1                           0x01 );        //OUT1 のみ ON する ret = hppd553_outpReadW ( hDeviceID,    //デバイス ID                           &amp;wOutp );      //格納先のアドレス ret = hppd553_outpWriteW( hDeviceID,    //デバイス ID                           0x0001 );     //出力ポート 1 の OUT1 のみ ON する  [ VB 6 ] Dim ret      As Long           ' 関数の戻り値 Dim byOutp   As Byte Dim wOutp    As Integer ret = hppd553_outpReadB ( hDeviceID, _ ' デバイス ID                           0, _         ' 出力ポート 1                           byOutp )     ' 格納先のアドレス ret = hppd553_outpWriteB( hDeviceID, _ ' デバイス ID                           0, _         ' 出力ポート 1                           &amp;H1 )        ' OUT1 のみ ON する ret = hppd553_outpReadW ( hDeviceID, _ ' デバイス ID                           wOutp )      ' 格納先のアドレス ret = hppd553_outpWriteW( hDeviceID, _ ' デバイス ID                           &amp;H1 )        ' 出力ポート 1 の OUT1 のみ ON する  [ VB. NET ] Dim ret      As Integer        ' 関数の戻り値 Dim byOutp   As Byte Dim wOutp    As Short ret = hppd553_outpReadB ( hDeviceID, _ ' デバイス ID                           0, _         ' 出力ポート 1                           byOutp )     ' 格納先のアドレス ret = hppd553_outpWriteB( hDeviceID, _ ' デバイス ID                           0, _         ' 出力ポート 1                           &amp;H1 )        ' OUT1 のみ ON する ret = hppd553_outpReadW ( hDeviceID, _ ' デバイス ID                           wOutp )      ' 格納先のアドレス ret = hppd553_outpWriteW( hDeviceID, _ ' デバイス ID                           &amp;H1 )        ' 出力ポート 1 の OUT1 のみ ON する </pre>
-------	---

(16)	<div>hppd553_ElstRead()</div> <div>hppd553_ElstWrite()</div> <div>E L S極性選択ポート読み</div> <div>E L S極性選択ポート書き</div>
機能	<div>デバイス ID で指定された HPCI-PPD55x の,</div> <div>E L S極性選択ポート読み・・ E L S 極性選択ポートを読み、指定エリアに格納します。</div> <div>E L S極性選択ポート書き・・ E L S 極性選択ポートに指定データを書込みます。</div>
書式	<div>[ C言語: VC++ ]</div> <div>DWORD WINAPI hppd553_ElstRead ( DWORD hDevID, BYTE* byElsData );</div> <div>DWORD WINAPI hppd553_ElstWrite( DWORD hDevID, BYTE byData );</div> <div>[ VB6 ]</div> <div>Declare Function hppd553_ElstRead Lib "hippd553.DLL"</div> <div>( ByVal hDevID As Long, ByRef byElsData As Byte ) As Long</div> <div>Declare Function hppd553_ElstWrite Lib "hippd553.DLL"</div> <div>( ByVal hDevID As Long, ByVal byData As Byte ) As Long</div> <div>[ VB. NET ]</div> <div>Declare Function hppd553_ElstRead Lib "hippd553.DLL"</div> <div>( ByVal hDevID As Integer, ByRef byElsData As Byte ) As Integer</div> <div>Declare Function hppd553_ElstWrite Lib "hippd553.DLL"</div> <div>( ByVal hDevID As Integer, ByVal byData As Byte ) As Integer</div>
引数	<div>◆ DWORD hDevID ・・対象デバイスのデバイス ID.</div> <div>◆ 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</div> <div>◆ BYTE byData ・・ E L S 極性選択ポートへの書き込データ</div> <div><div><div><div>7</div><div>4</div><div>3</div><div>0</div></div><div><div><div>x</div><div>x</div><div>x</div><div>x</div></div><div><div>x</div><div></div><div></div><div></div></div></div><div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div></div><div></div><div></div></div></div></div><div><div>読み ( x : ビットの値は不定です. )</div><div>書き</div><div>X 軸</div><div>Y 軸</div><div>Z 軸</div></div><div><div>0: ±ELS B 接 (デフォルト)</div><div>1: ±ELS A 接</div></div></div>
戻り値	<div>処理結果</div> <div>1 : 成功</div> <div>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。</div>
呼び出し例	<div>[ C言語: VC++ ]</div> <div>DWORD ret;</div> <div>BYTE ElsData;</div> <div>ret = hppd553_ElstRead ( hDeviceID,</div> <div>&amp;ElsData );</div> <div>ret = hppd553_ElstWrite( hDeviceID,</div> <div>0x01 );</div> <div>//関数の戻り値</div> <div>//デバイス ID</div> <div>//格納先のアドレス</div> <div>//デバイス ID</div> <div>//X 軸 A 接, Y ・ Z 軸 B 接</div> <div>[ VB6 ]</div> <div>Dim ret As Long</div> <div>Dim ElsData As Byte</div> <div>ret = hppd553_ElstWrite( hDeviceID, _</div> <div>&amp;H1 )</div> <div>ret = hppd553_ElstRead ( hDeviceID, _</div> <div>ElsData )</div> <div>' 関数の戻り値</div> <div>' デバイス ID</div> <div>' X 軸 A 接, Y ・ Z 軸 B 接</div> <div>' デバイス ID</div> <div>' 格納先のアドレス</div> <div>[ VB. NET ]</div> <div>Dim ret As Integer</div> <div>Dim ElsData As Byte</div> <div>ret = hppd553_ElstWrite( hDeviceID, _</div> <div>&amp;H1 )</div> <div>ret = hppd553_ElstRead ( hDeviceID, _</div> <div>ElsData )</div> <div>' 関数の戻り値</div> <div>' デバイス ID</div> <div>' X 軸 A 接, Y ・ Z 軸 B 接</div> <div>' デバイス ID</div> <div>' 格納先のアドレス</div>

(17)	hppd553_CmpcRead () hppd553_CmpcWrite ()	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書込
機能	デバイス ID で指定された HPCI-PPD55x 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み込み、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。	
書式	[ C言語: VC++ ] DWORD WINAPI hppd553_CmpcRead ( DWORD hDevID, BYTE* byCmpData ); DWORD WINAPI hppd553_CmpcWrite( DWORD hDevID, BYTE byData ); [ VB6 ] Declare Function hppd553_CmpcRead Lib "hippd553.DLL" ( ByVal hDevID As Long, ByRef byCmpData As Byte ) As Long Declare Function hppd553_CmpcWrite Lib "hippd553.DLL" ( ByVal hDevID As Long, ByVal byData As Byte ) As Long [ VB.NET ] Declare Function hppd553_CmpcRead Lib "hippd553.DLL" ( ByVal hDevID As Integer, ByRef byCmpData As Byte ) As Integer Declare Function hppd553_CmpcWrite Lib "hippd553.DLL" ( ByVal hDevID As Integer, ByVal byData As Byte ) As Integer	
引数	◆ 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 <div><div><div><div>7</div><div>4</div><div>3</div><div>0</div></div><div><div><div>x</div><div>x</div><div>x</div><div>x</div></div><div><div>x</div><div>x</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div></div><div>読込 ( x : ビットの値は不定です. ) 書込</div><div><div><div>X軸</div><div>Y軸</div><div>Z軸</div></div><div>0: 出力不可 (デフォルト) 1: 出力可</div></div></div>	
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 CmpData; ret = hppd553_CmpcRead ( hDeviceID,        //デバイス ID &CmpData );        //格納先のアドレス ret = hppd553_CmpcWrite( hDeviceID,        //デバイス ID 0x01 );            //X軸のみ出力可  [ VB6 ] Dim ret                                    ' 関数の戻り値 Dim CmpData                                ' デバイス ID ret = hppd553_CmpcRead ( hDeviceID,        ' 格納先のアドレス CmpData ) ret = hppd553_CmpcWrite( hDeviceID,        ' デバイス ID &H1 )            ' X軸のみ出力可  [ VB.NET ] Dim ret                                    ' 関数の戻り値 Dim CmpData                                ' デバイス ID ret = hppd553_CmpcRead ( hDeviceID,        ' 格納先のアドレス CmpData ) ret = hppd553_CmpcWrite( hDeviceID,        ' デバイス ID &H1 )            ' X軸のみ出力可	



(18)	hppd553_diRead() 汎用入力による同時スタート・ストップ読込 hppd553_diWrite() 汎用入力による同時スタート・ストップ書込																		
機能	デバイス ID で指定された HPCI-PPD55x の、 汎用入力 IN1 による同時スタート・IN2 による同時ストップ読込 ・ 汎用入力による同時スタート・ストップを読込み、指定エリアに格納します。 汎用入力 IN1 による同時スタート・IN2 による同時ストップ書込 ・ 汎用入力による同時スタート・ストップに指定データを書込みます。																		
書式	[ C 言語: VC++ ] DWORD WINAPI hppd553_diRead ( DWORD hDevID, BYTE* byDi ); DWORD WINAPI hppd553_diWrite( DWORD hDevID, BYTE byData ); [ VB6 ] Declare Function hppd553_diRead Lib "hppd553.DLL" ( ByVal hDevID As Long, ByRef byDi As Byte ) As Long Declare Function hppd553_diWrite Lib "hppd553.DLL" ( ByVal hDevID As Long, ByVal byData As Byte ) As Long [ VB. NET ] Declare Function hppd553_diRead Lib "hppd553.DLL" ( ByVal hDevID As Integer, ByRef byDi As Byte ) As Integer Declare Function hppd553_diWrite Lib "hppd553.DLL" ( ByVal hDevID As Integer, ByVal byData As Byte ) As Integer																		
引数	◆ DWORD hDevID ・ 対象デバイスのデバイス ID. ◆ BYTE* byDi ・ 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 ・ 汎用入力による同時スタート・ストップへの書込データ <div><div><div><div>7</div><div>4</div><div>3</div><div>2</div><div>1</div><div>0</div></div><table><tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td><td>IN2</td><td>IN1</td><td></td></tr><tr><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>IN1</td></tr></table><div>読込 ( x : ビットの値は不定です. ) 書込</div><div>同時スタート 0 : 出力不可、1 : 出力可 同時ストップ 0 : 出力不可、1 : 出力可</div></div></div>	x	x	x	x	x	x	x	x	x	IN2	IN1		O	O	O	O	O	IN1
x	x	x	x	x	x														
x	x	x	IN2	IN1															
O	O	O	O	O	IN1														
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 byDi; ret = hppd553_diRead ( hDeviceID, //デバイス ID &byDi ); //格納先のアドレス ret = hppd553_diWrite( hDeviceID, //デバイス ID 0x01 ); //IN1 による同時スタート設定  [ VB6 ] Dim ret As Long '関数の戻り値 Dim byDi As Byte ret = hppd553_diRead ( hDeviceID, 'デバイス ID byDi ) '格納先のアドレス ret = hppd553_diWrite( hDeviceID, 'デバイス ID &H1 ) 'IN1 による同時スタート設定  [ VB. NET ] Dim ret As Integer '関数の戻り値 Dim byDi As Byte ret = hppd553_diRead ( hDeviceID, 'デバイス ID byDi ) '格納先のアドレス ret = hppd553_diWrite( hDeviceID, 'デバイス ID &H1 ) 'IN1 による同時スタート設定																		

## 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 | VC++ . . . C コーディング  | 【 spp55300.exe 】 |
| ② | VC++ . . . C++コーディング | 【 spp55301.exe 】 |
| ③ | VB 6. 0              | 【 spp55302.exe 】 |
| ④ | VB . NET             | 【 spp55303.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（spp55300.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。

### 《 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 】

※ HPCI-PPD55x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

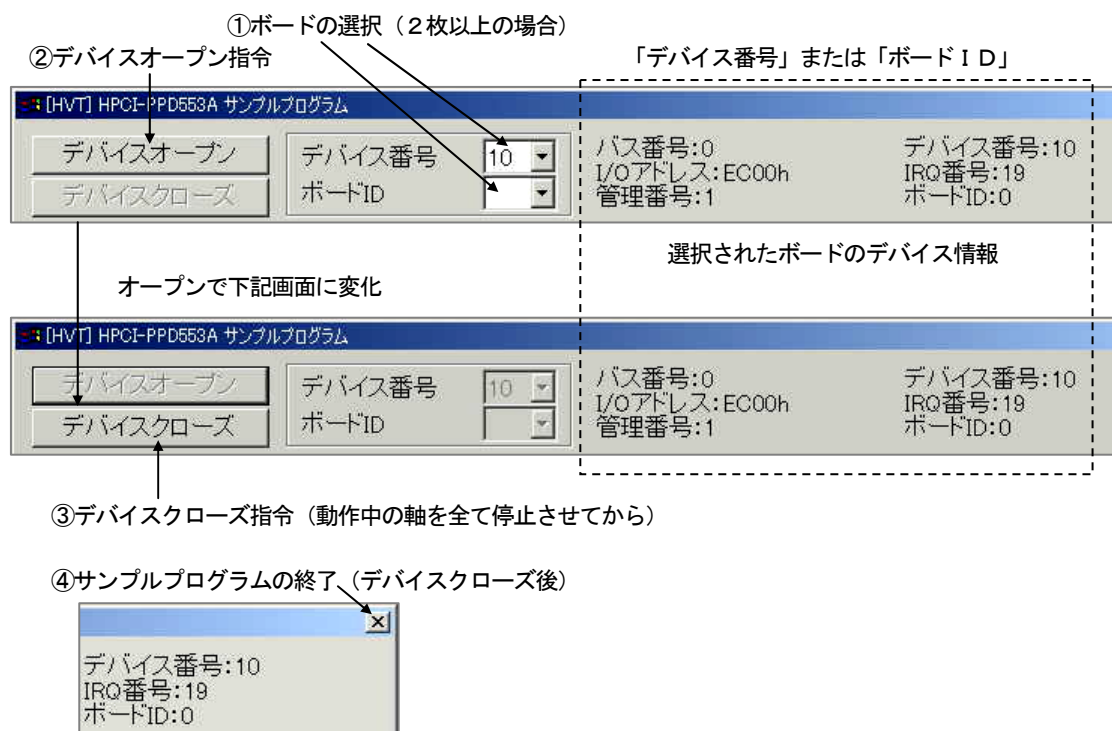
### 6. 1 サンプルプログラムの操作

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. 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。



## 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。

[HVT] HPCI-PPD553A サンプルプログラム

デバイスオープン デバイスクローズ

デバイス番号: 10 ボードID: 10

バス番号: 0 I/Oアドレス: EC00h 管理番号: 1

デバイス番号: 10 IRQ番号: 19 ボードID: 0

X軸 フリセット量: 1500

位置 指令

速度

初期化

sts1: F8h sts2: 80h ints: 00h

ELS B接 DLS B接 OLS B接 SVON

INPOS SVALM

+高速フリセット送り -高速フリセット送り +高速連続送り -高速連続送り

定速原点復帰 高速原点復帰 減速停止 即停止

Y軸 フリセット量: 1500

位置 指令

速度

初期化

sts1: F8h sts2: 80h ints: 00h

ELS B接 DLS B接 OLS B接 SVON

INPOS SVALM

+高速フリセット送り -高速フリセット送り +高速連続送り -高速連続送り

定速原点復帰 高速原点復帰 減速停止 即停止

Z軸 フリセット量: 1500

位置 指令

速度

初期化

sts1: F8h sts2: 80h ints: 00h

ELS B接 DLS B接 OLS B接 SVON

INPOS SVALM

+高速フリセット送り -高速フリセット送り +高速連続送り -高速連続送り

定速原点復帰 高速原点復帰 減速停止 即停止

入力ポート

16 12 9 8 1

出力ポート

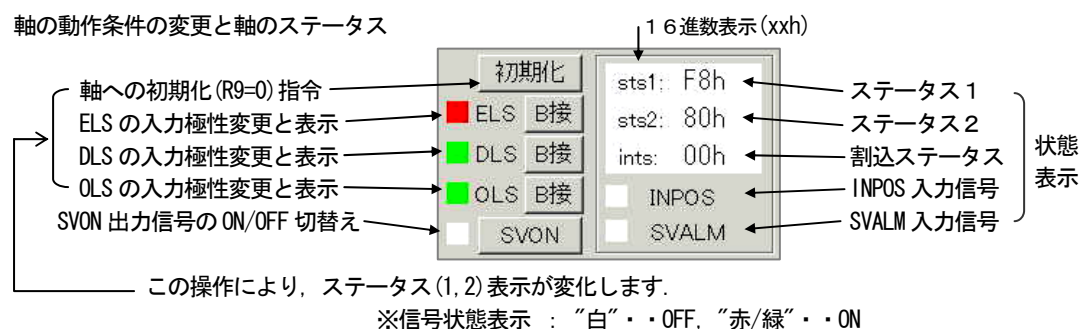
入出力点数切替

8 1

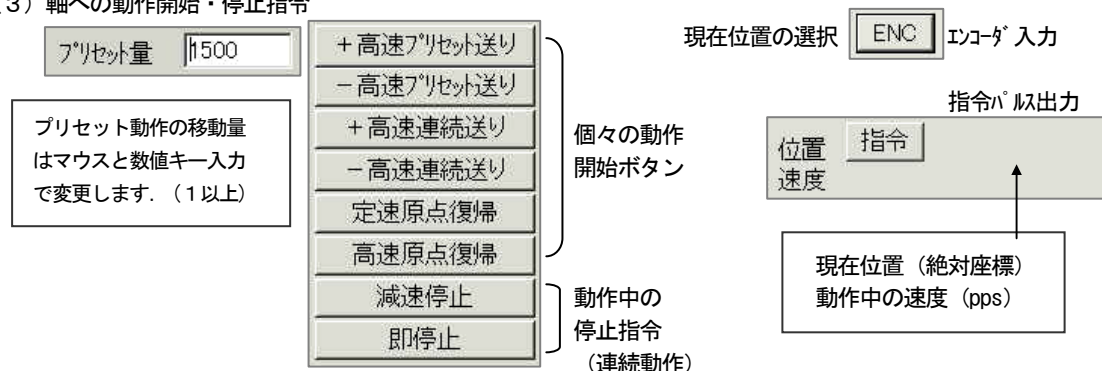
ボード上の軸数の多少に関わらず、個々の軸に対する操作は同一です。  
 なお、サンプルプログラムでは各軸の初期化は一部ソースプログラムで固定されています。  
 その為に、初期化の条件を変更して動作させたい場合には、ソースプログラム変更の必要があります。

(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。

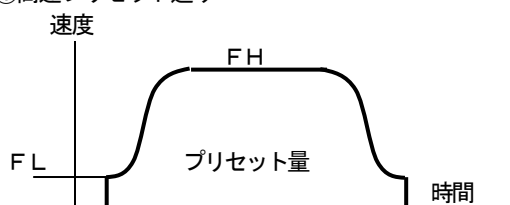
(2) 軸の動作条件の変更と軸のステータス



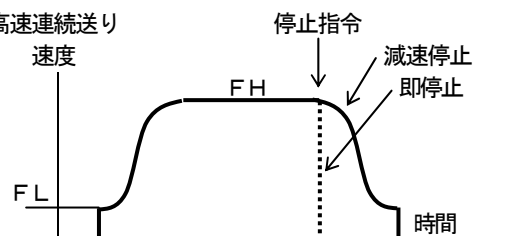
(3) 軸への動作開始・停止指令



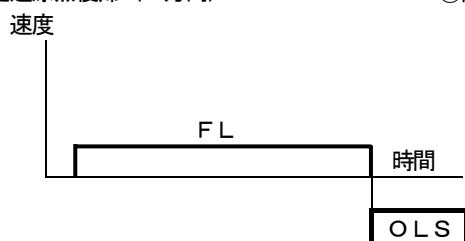
①高速プリセット送り



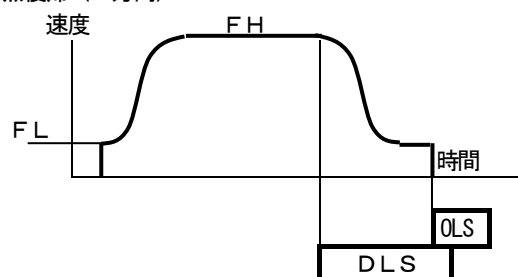
②高速連続送り



③定速原点復帰 (一方向)

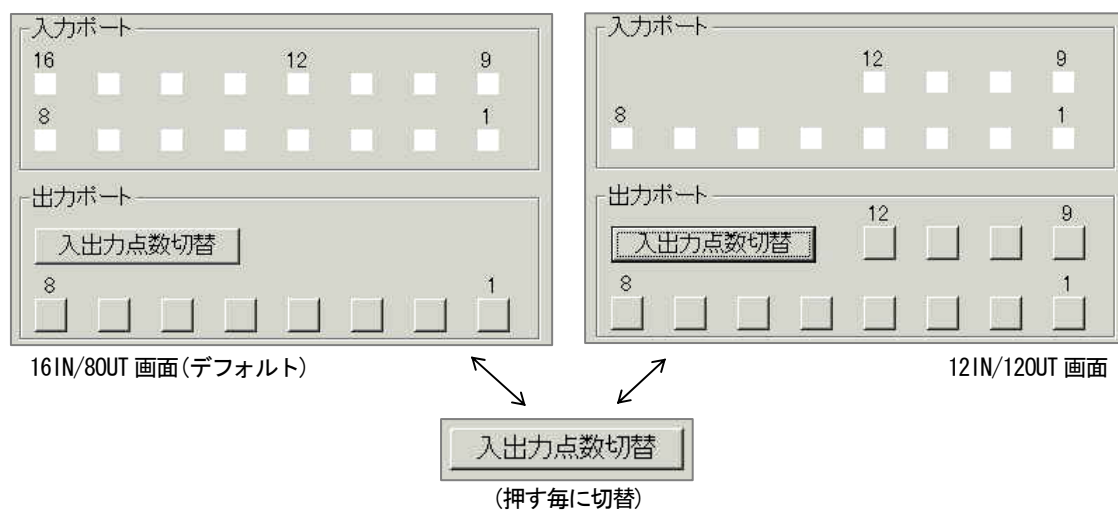


④高速原点復帰 (一方向)



(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。

(4) 16IN/8OUT と 12IN/12OUT の入出力点数切替



### 6. 1. 3 軸動作の設定条件

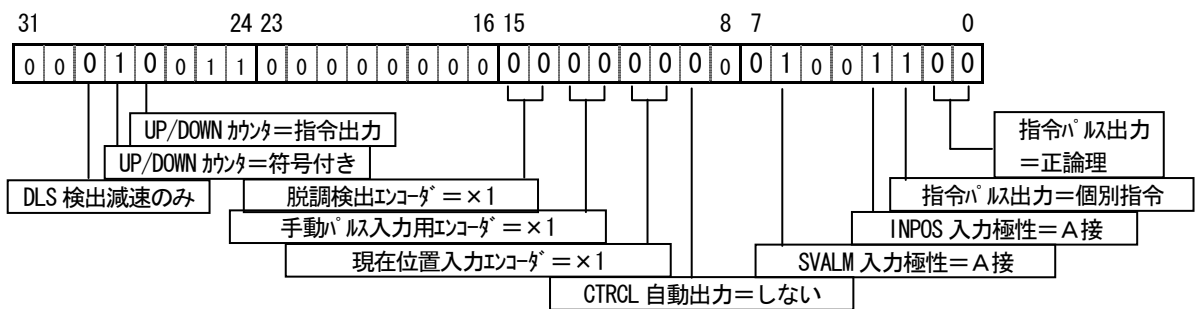
(1) 軸制御のレジスタ初期設定（デフォルト値）・・デバイスオープン時

PCLレジスタ	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） [ $(2457600 * 1\text{sec}) / (\text{FH} - \text{FL}) - 1$ ]
PR 4	299	速度倍率 = 1pps [ $= 300 / (\text{PR4} + 1)$ ]
PR 5	0	減速点（制御モードで“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

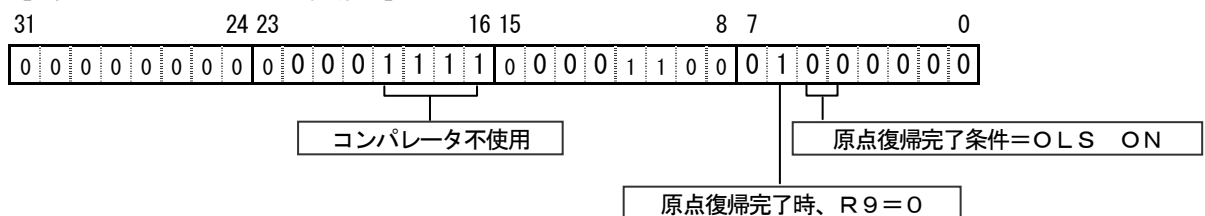
表 6. 1-1 軸制御のレジスタ初期設定

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。

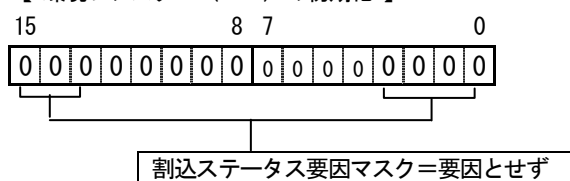
【 環境レジスタ 1（R 6）の初期化 】



【 環境レジスタ 2（R 7）の初期化 】



【 環境レジスタ 3（R 8）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	7	0	bit	制御内容	設定値
4	次動作自動スタート制御	0 しない	0	0	0	プリセットカウンタ動作	0 ON
5	同時停止信号入力機能	0 OFF	0	0	1	減速点検出方式	0 自動
6	同時停止信号出力機能	0 ON	0	0	2	加減速方式	1 S字
7	三角駆動時速度修正機能	0 修正	0	0	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 / .NET )**

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			
	VC++ (C)	VC++ (C++)	VB6	VB. NET
実行可能ファイル	spp55300.exe	spp55301.exe	spp55302.exe	spp55303.exe
プロジェクトファイル (ソリューションファイル)	spp55300.dsw spp55300.dsp  spp55300.opt	spp55301.dsw spp55300.dsp  spp55300.opt	spp55302.vbp	spp55303.sln spp55303.suo spp55303.vbproj spp55303.vbproj.user
ドライバ I/F ファイル	hippd553.dll hippd553.lib hippd553.h	hippd553.dll hippd553.lib hippd553.h	hippd553.dll	hippd553.dll
ソースプログラム	上記以外	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、spp55302.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、spp55300.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、spp55301.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) VB. NET 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、spp55303.sln を選択して下さい。

Microsoft Visual Basic .NET がインストールされている必要があります。

#### (5) 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()	デバイス静粛終了後の処理
	5	FormUnload()	ウィンドウを閉じる
	6	OnIdleho_Click()	デバイス静粛選択 & 表示(コンボボックスのクリック)
	7	BtnOpen2_Click()	デバイスオープンボタンのクリック
	8	BtnClose1_Click()	デバイスクローズボタンのクリック(クローズ直前)
	9	BtnClose3_Click()	デバイスクローズボタンのクリック(クローズ直後)
	10	BtnEls2_Click()	ELS(A接/B接)ボタンのクリックでELS入力極性の反転
	11	BtnDls2_Click()	DLS(A接/B接)ボタンのクリックでDLS入力極性の反転
	12	BtnOls2_Click()	OLS(A接/B接)ボタンのクリックでOLS入力極性の反転
	13	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流入変更
	14	BtnIoSw1_Click()	入出力点数切替ボタンのクリックでIN/OUT の表示切替
	15	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	16	FormLoad2()	デバイス静粛取得
	17	BtnOpen1_Click()	デバイスオープンボタンのクリックでポート初期化
	18	Initial()	ポート上の軸初期化 (1軸分)
	19	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
	20	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
	21	BtnNPre_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
	22	BtnPRen_Click()	+高速連続送りボタンのクリックで所定動作開始
	23	BtnNRen_Click()	-高速連続送りボタンのクリックで所定動作開始
	24	BtnTGen_Click()	定速原点復帰ボタンのクリックで所定動作開始
	25	BtnKGen_Click()	高速原点復帰ボタンのクリックで所定動作開始
	26	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
	27	BtnSStop_Click()	即停止ボタンのクリックで軸に即停止指令を発行
	28	BtnIni_Click()	初期化ボタンのクリックで「アップダウンカウンタ(R9)」クリア
	29	BtnSVOn_Click()	SVON(ON/OFF)ボタンのクリックで出力信号の反転指令
	30	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
	31	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
	32	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
	33	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
	34	BtnIoSw2_Click()	入出力点数切替ボタンのクリックで16IN/8OUT と12IN/12OUT の切替
	35	BtnOut_Click()	出力ポート(ON/OFF)ボタンのクリックで出力信号の反転指令
	36	TmrSokudo()	現在速度 & 現在位置表示 (動作1軸:0.1秒毎)
	37	TmrSts()	ステータス読入 (動作1軸:0.1秒毎)
	38	TmrInOut()	入出力ポート表示 (動作1軸:0.1秒毎)

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

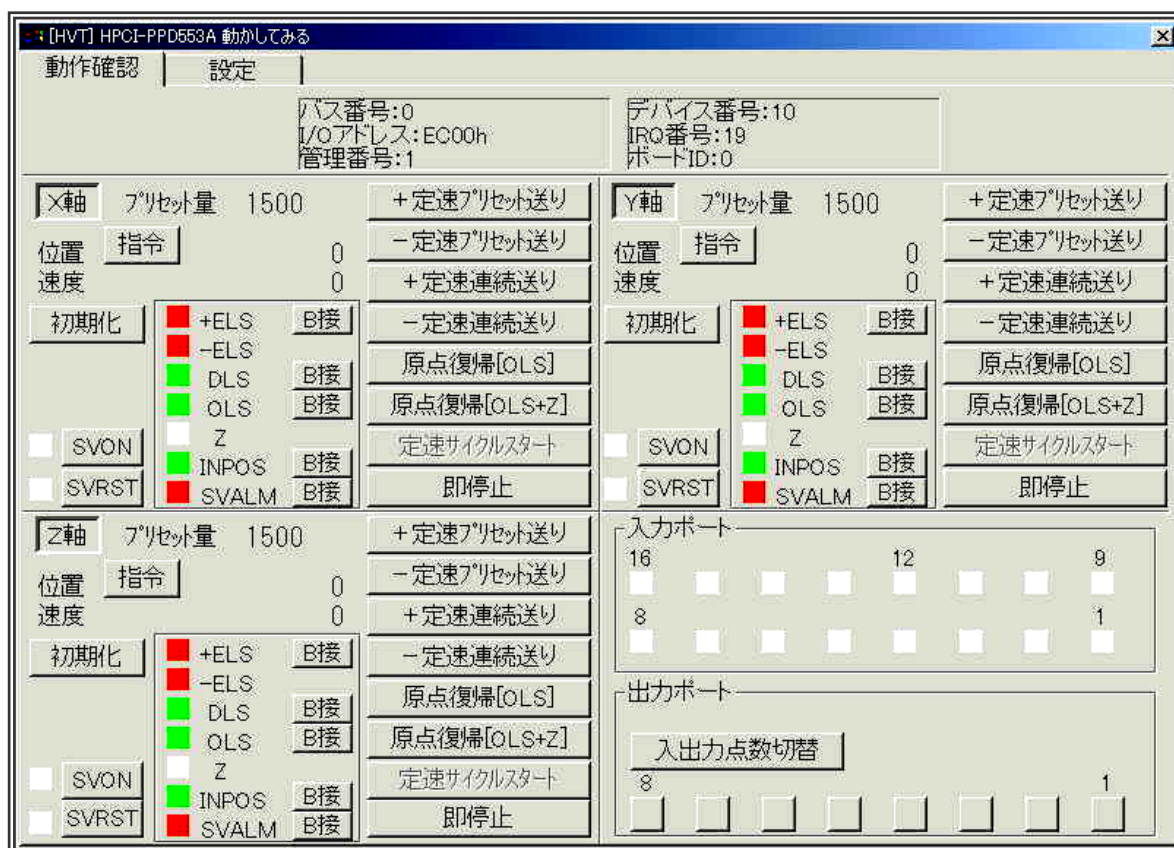


## 7. 「動かしてみる」プログラム

「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¥tpp55300.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。



#### 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。

バス番号:0	デバイス番号:10
I/Oアドレス:EC00h	IRQ番号:19
管理番号:1	ポートID:0

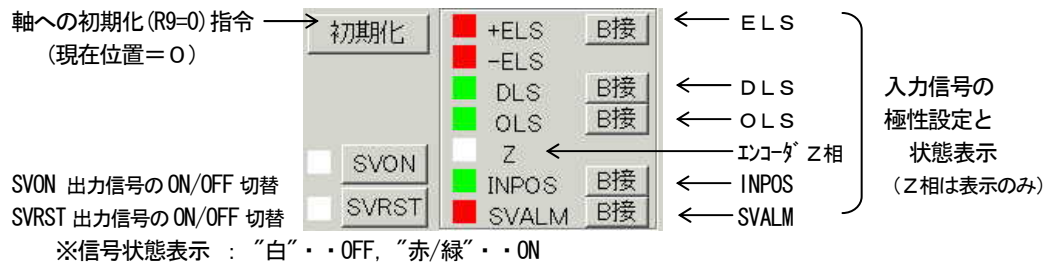
- (注) 1. 管理番号      Windows 9 X では“- 1”です。  
2. ボードID      ボード上のジャンパ設定値です。  
                     ジャンパの無いボードでは“1 5”となります。

#### 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。

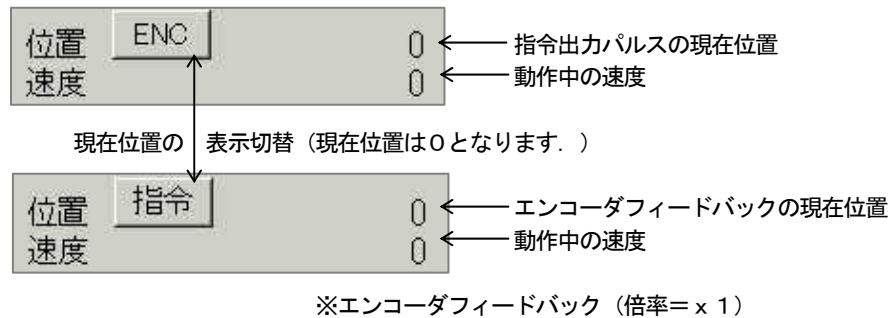
(2) 軸の動作条件の変更と軸のステータス



(3) 軸の現在位置・動作速度表示

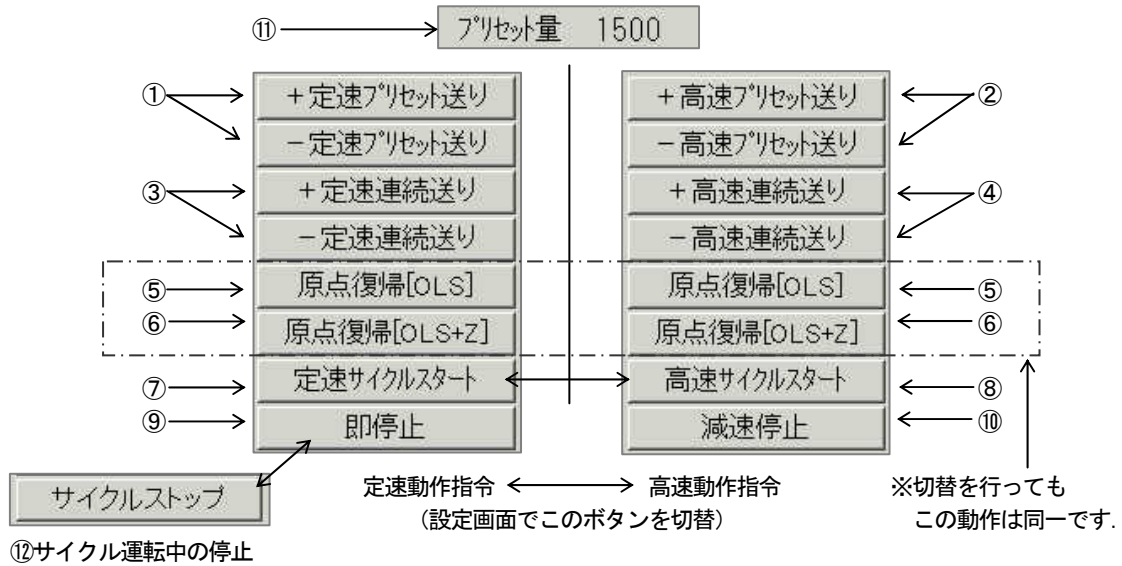
各軸の現在位置および動作中の速度は0.1秒毎に更新されます。

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。

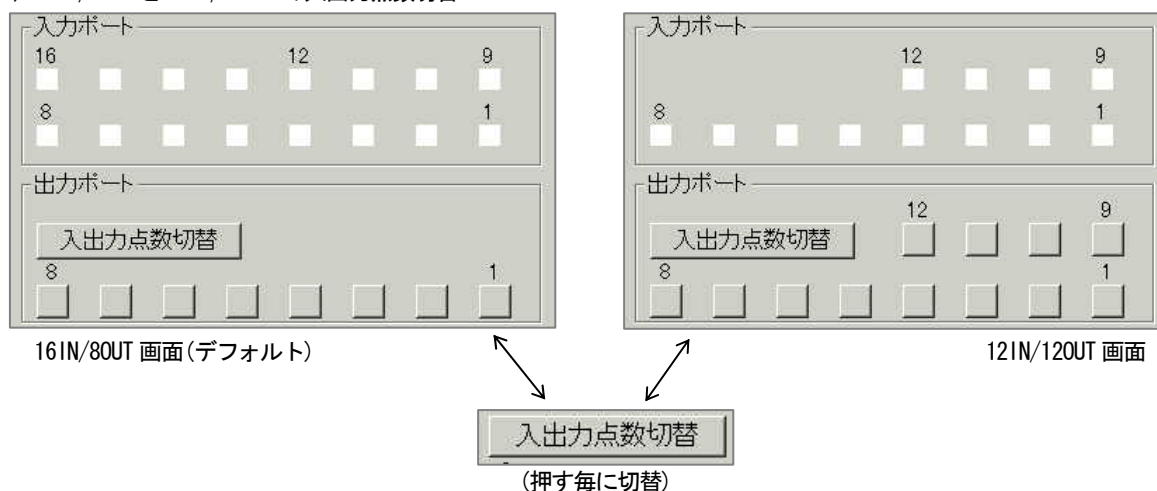


(4) 軸への動作開始・停止指令

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。

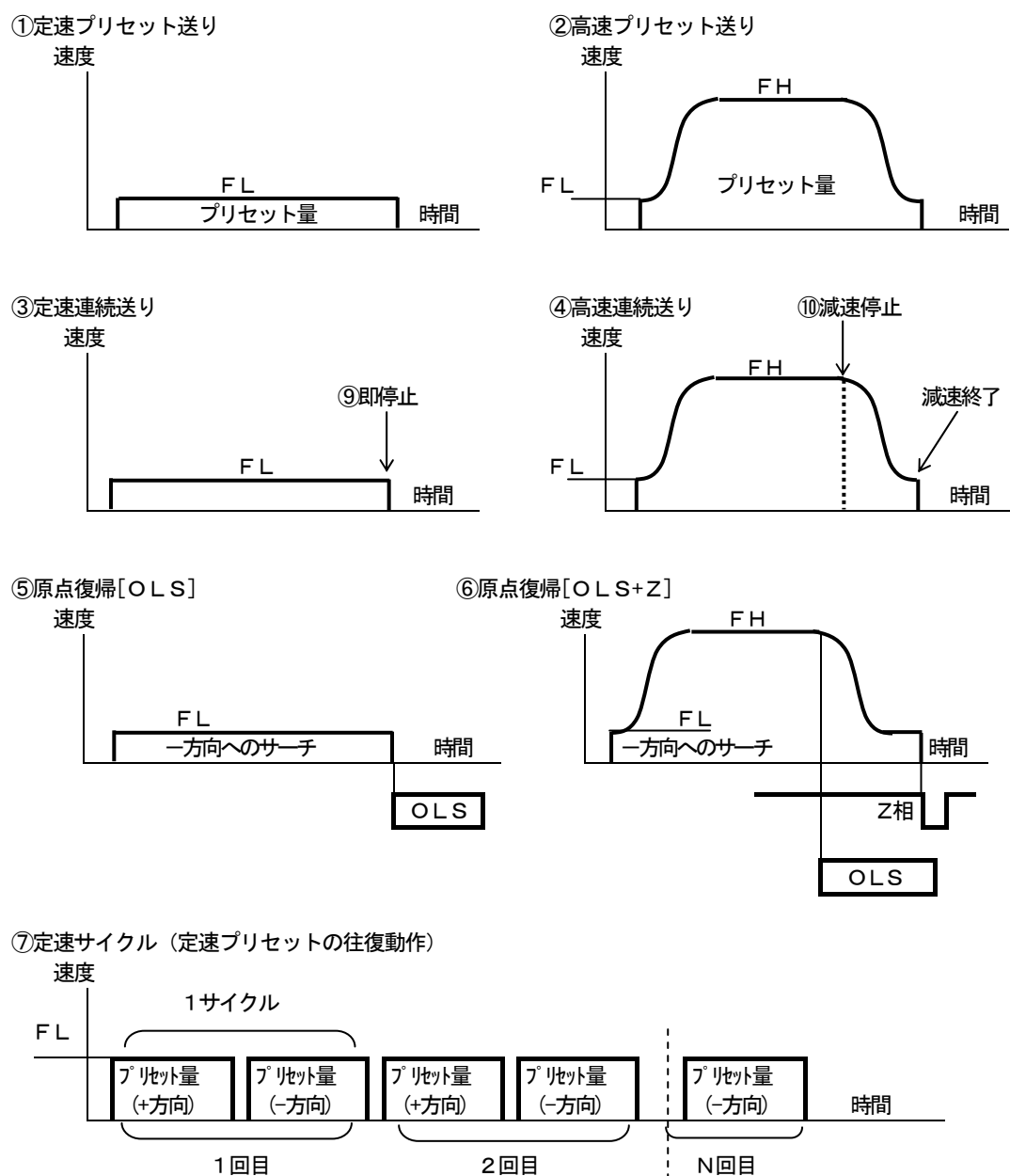


(5) 16IN/8OUT と 12IN/12OUT の入出力点数切替

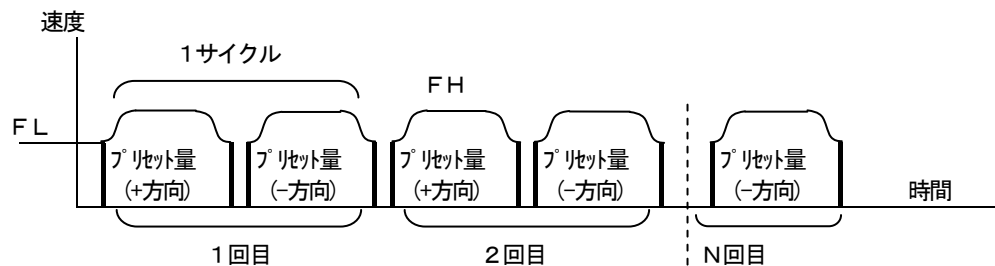


(6) 軸への指令と動作内容

「動かしてみる」プログラムの動作内容は下図の通りです。



⑧高速サイクル（高速プリセットの往復動作）



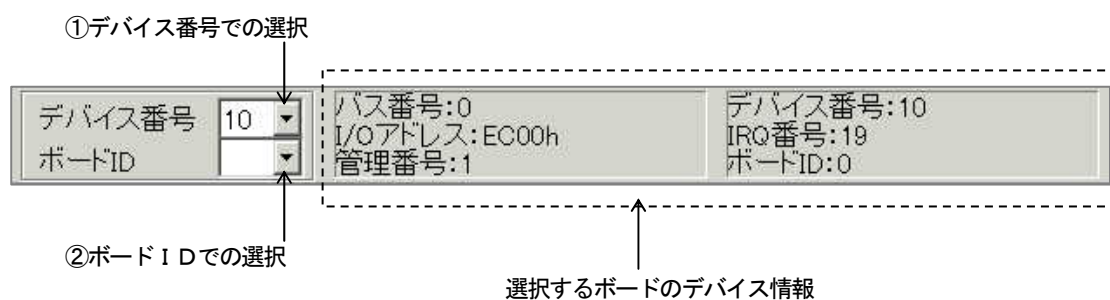
- (注) 1. 加減速はS字を使用しています。  
 2. 原点復帰の方向は“-”方向に固定されています。  
 3. 原点復帰[OLS]は定速であり，原点信号=OLSです。  
 4. 原点復帰[OLS+Z]は高速であり，原点信号=OLS減速+Z相1回です。  
 5. 動作に関する要素（プリセット量・速度等）は「設定画面」で変更出来ます。  
 6. “サイクル”動作は「設定画面」で回数指定を行います。  
 7. “サイクル”動作では「次動作自動スタート」機能で連続動作となります。

## 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。

### 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

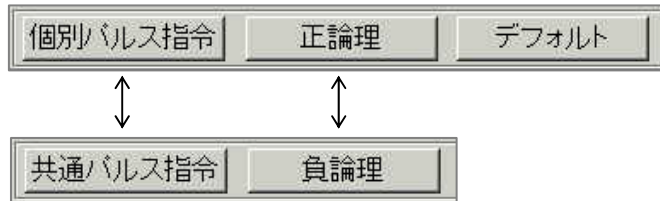
動作可能な全ての軸について、個々に動作条件が設定出来ます。

### (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。

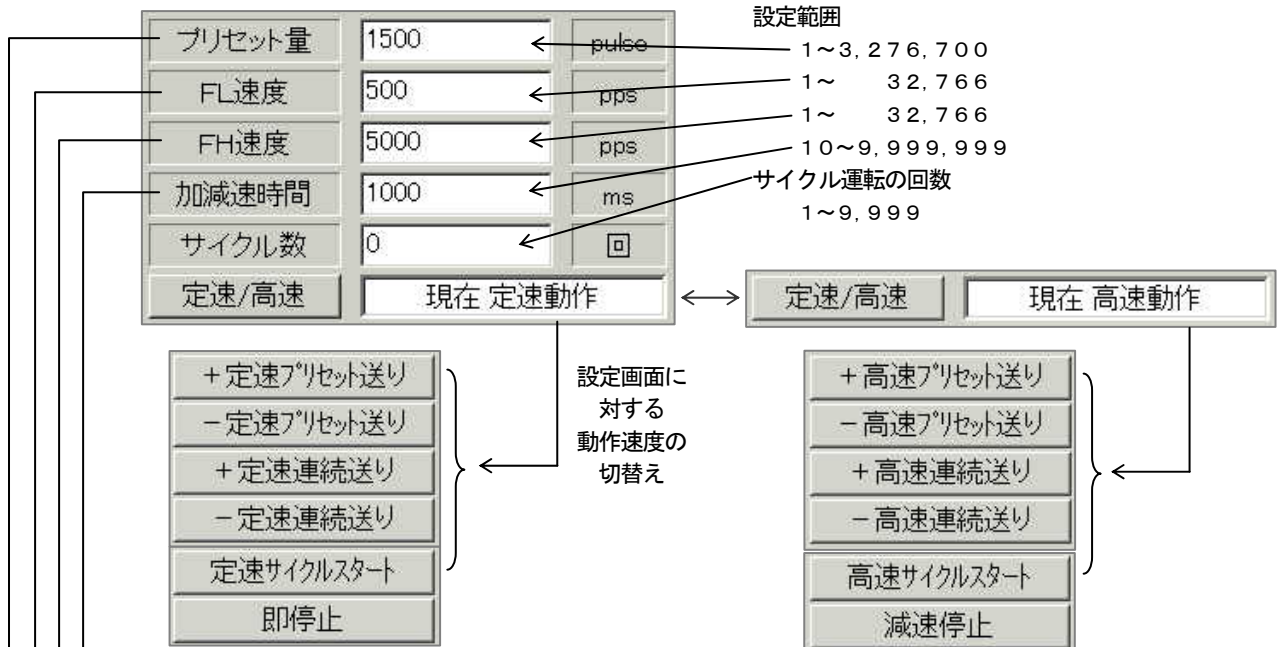
#### 【 指令パルス出力の設定 】

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 】



PCLレジスタ	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 (制御モードで“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で加減速=減速)
R16	0	S字加減速・S字区間

※ 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。

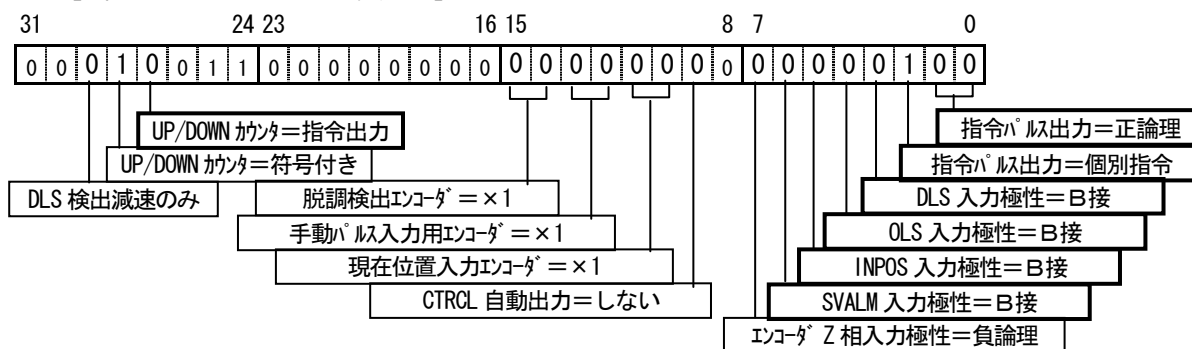


## (3) 環境レジスタの設定内容

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。

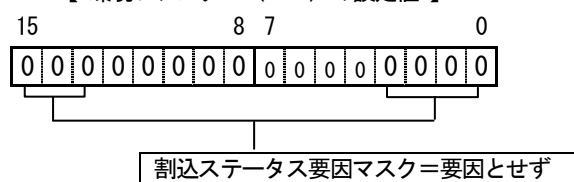
### 【環境レジスタ1 (R6) の設定値】



### 【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	7	0	bit	制御内容	設定値
4	次動作自動スタート制御	0 する	0	0	0	リセットカウンタ動作	0 ON
5	同時停止信号入力機能	0 OFF	1	0	1	減速点検出方式	0 自動
6	同時停止信号出力機能	0 ON	0	1	2	加減速方式	1 S字
7	三角駆動時速度修正機能	0 修正	0	0	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

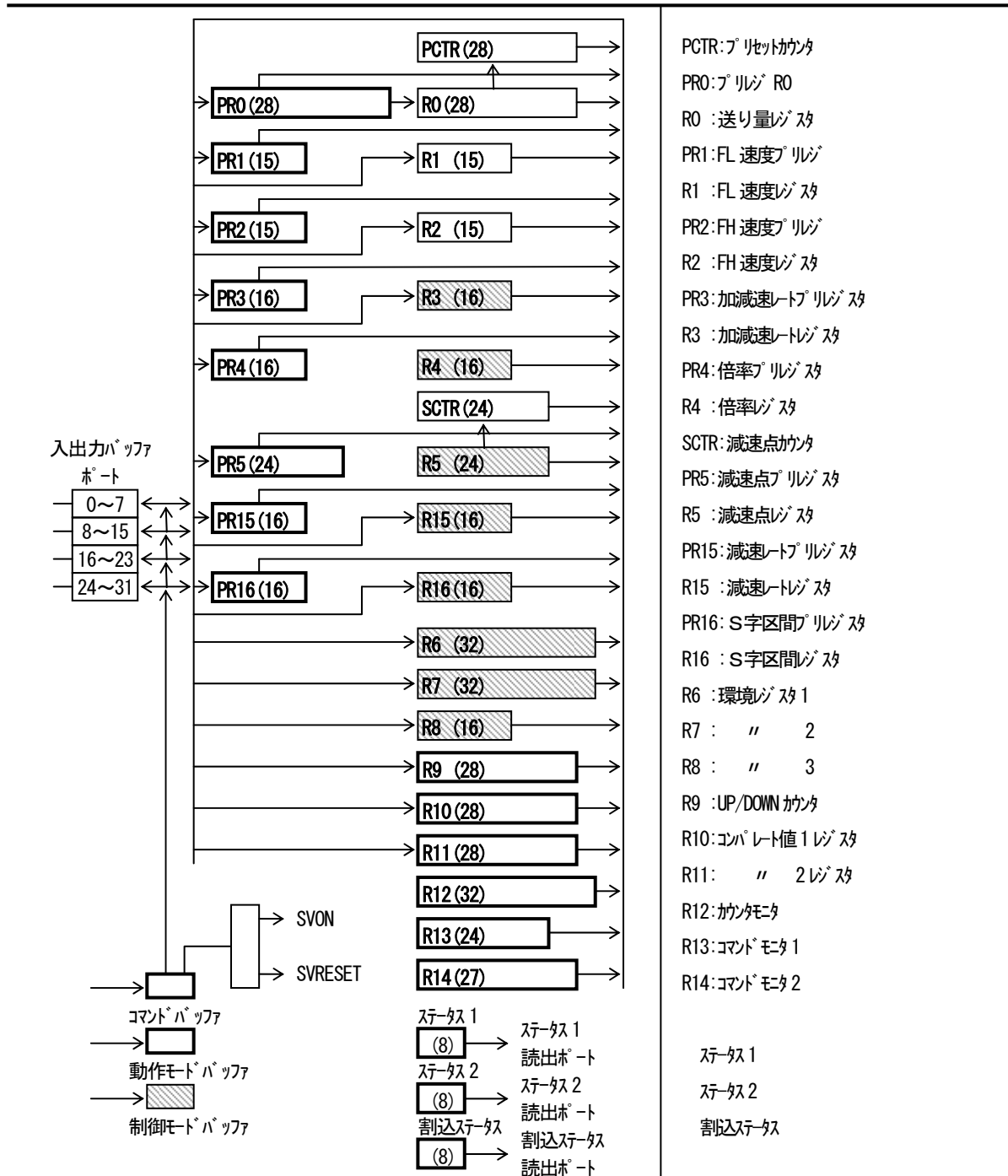
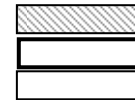
### 1. PCL5014・レジスタの使用分類

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ブリードの使用大別は、使用上次のように考えます。

(図中枠線区別)

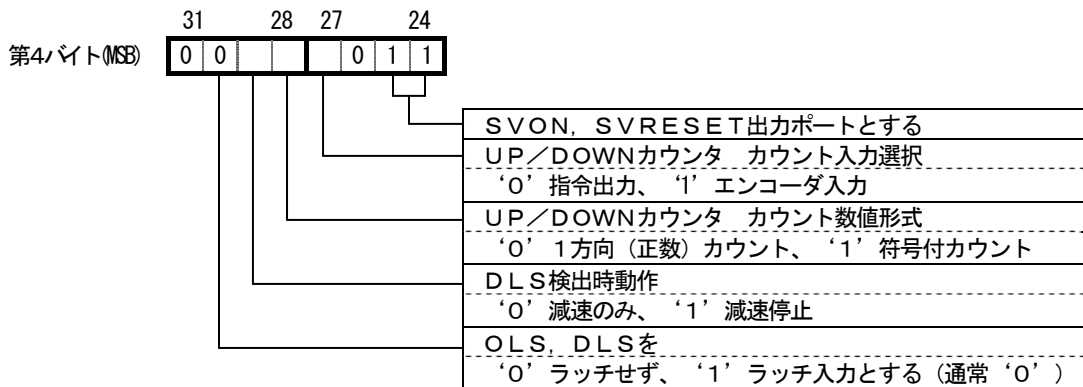
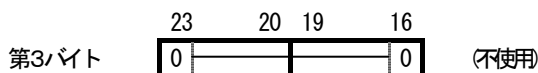
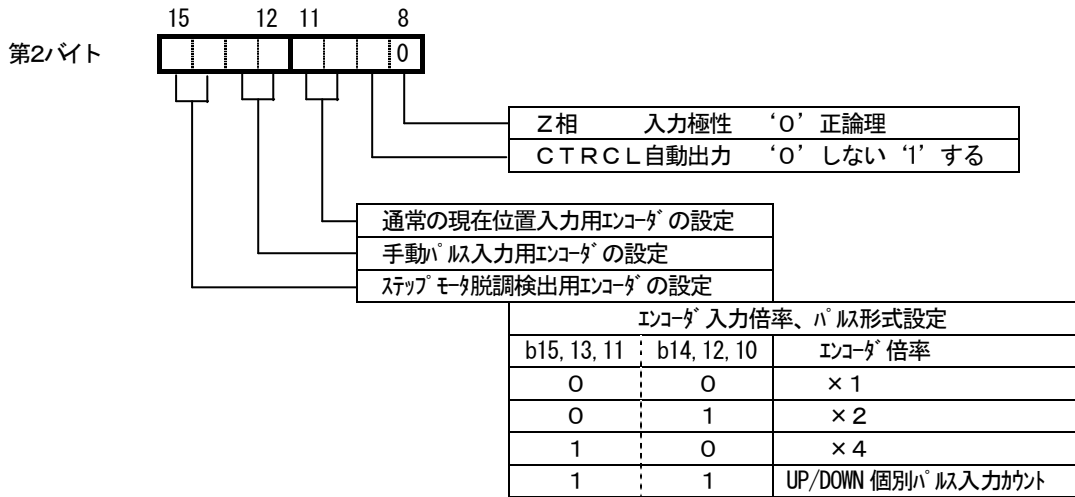
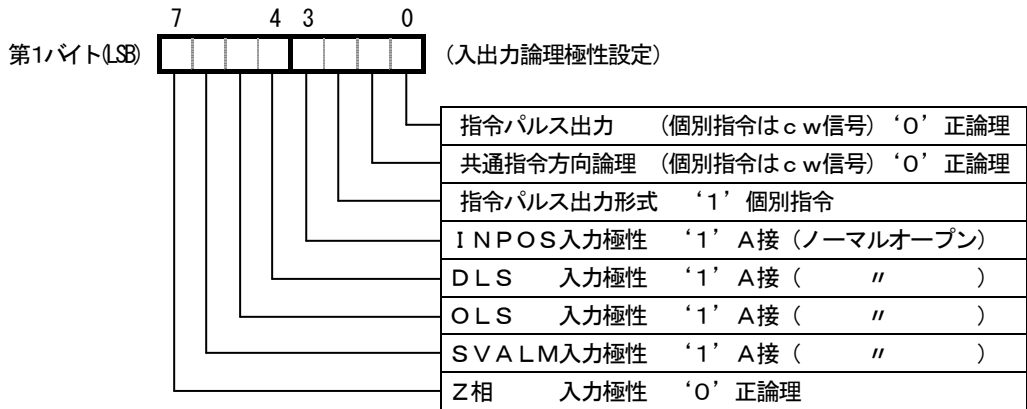
- ① 原則として初期化時に一度設定すればよいレジスタ類
- ② 実行時に設定するレジスタ又は必要に応じて読出して使用するレジスタ類
- ③ その他レジスタ類



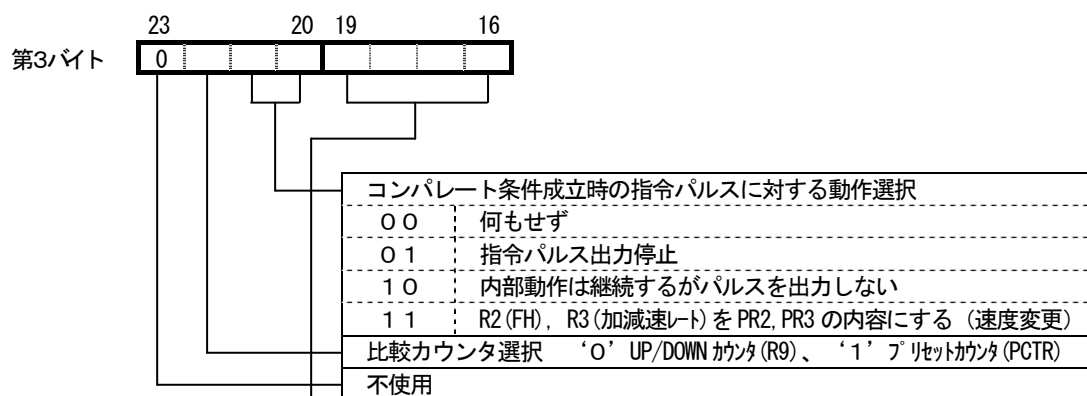
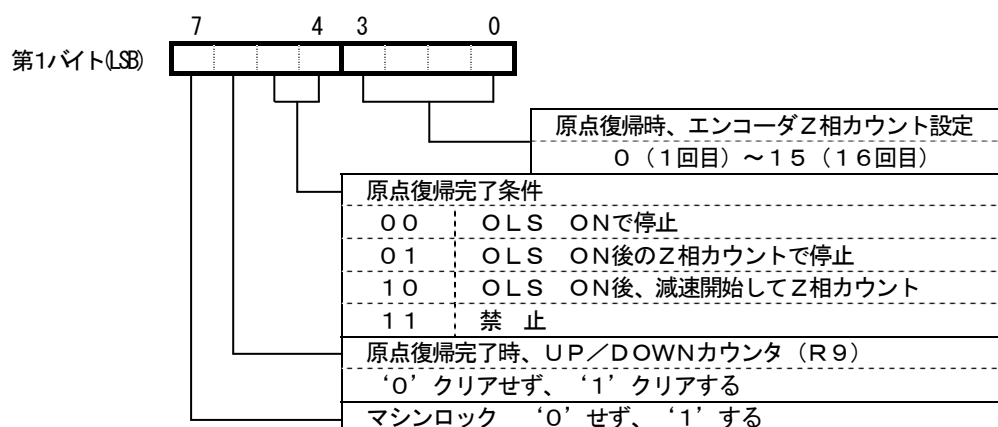


## 2. 個別レジスタ

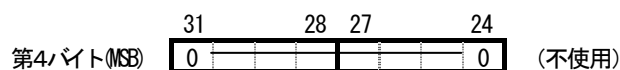
### 2. 1 環境レジスタ 1 (R6)



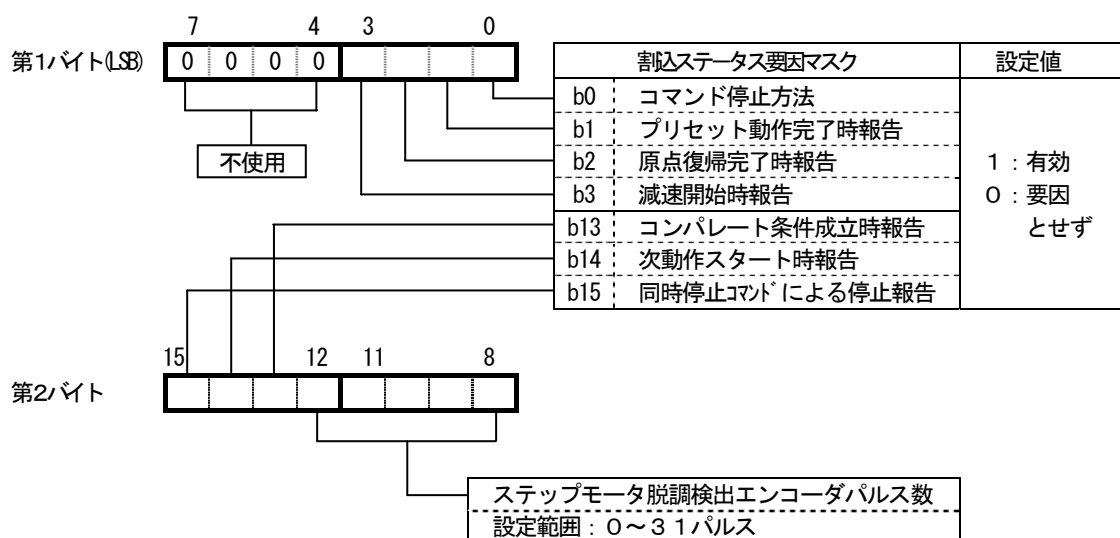
## 2. 2 環境レジスタ 2 (R7)



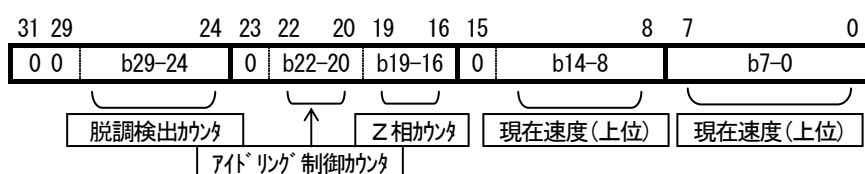
コンパレート条件選択		
b19-b16	比較条件	結果
0 0 0 0	R 1 0 > (カウンタ)	(1) ステータス 2 b0 = '1' で条件成立が確認できる。 (2) '=' 条件は、一致からカウンタ変化で直ちに変わる。この場合は割込ステータスを利用する (3) コンパレータ不使用は「1 1 1 1」とする
0 0 0 1	R 1 0 = (カウンタ)	
0 0 1 0	R 1 0 < (カウンタ)	
0 1 0 0	R 1 1 > (カウンタ)	
0 1 0 1	R 1 1 = (カウンタ)	
0 1 1 0	R 1 1 < (カウンタ)	
1 0 0 0	R 1 0 > (カウンタ) 又は R 1 1 < (カウンタ)	
1 0 0 1	R 1 0 < (カウンタ) 又は R 1 1 > (カウンタ)	
1 0 1 0	R 1 0 = (カウンタ) 又は R 1 1 = (カウンタ)	
1 0 1 1	R 1 0 > (カウンタ) 又は R 1 1 > (カウンタ)	
1 1 0 0	R 1 0 < (カウンタ) 又は R 1 1 < (カウンタ)	
1 1 0 1	R 1 0 < (カウンタ) < R 1 1	
1 1 1 0	R 1 0 > (カウンタ) > R 1 1	



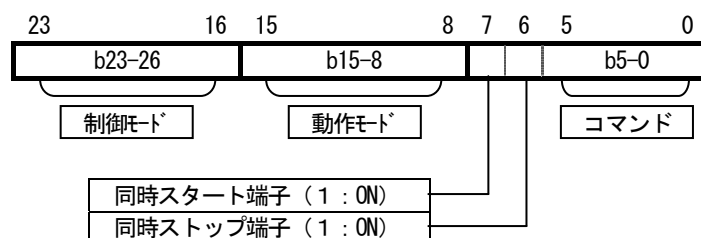
## 2. 3 環境レジスタ3 (R8)



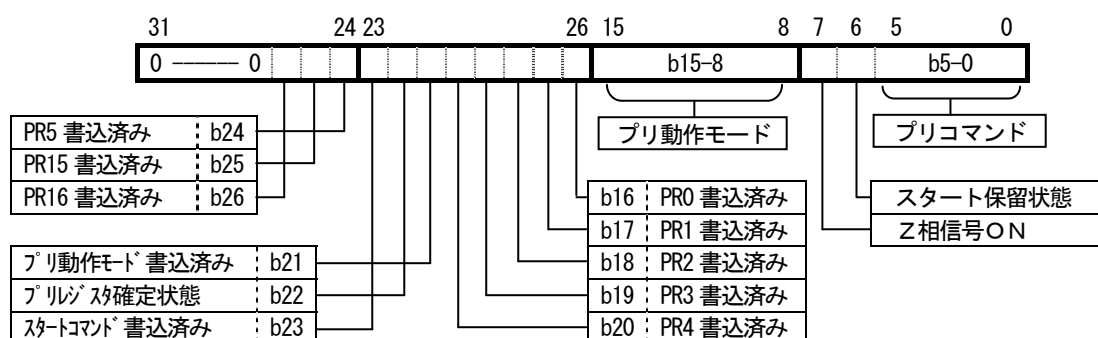
## 2. 4 カウンタモニタ (R12)



## 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。