ＰＣＩバス対応
汎用８軸モータコントロールボード

MC8080P　取扱説明書

初版　　２００２.０７.３０
改定　　２００６.０８.１０

NOVA　electronics　　株式会社 ノヴァ電子

はじめに

このたびは、ＭＣ８０８０Ｐをご検討いただきまして、ありがとうございます。

■ 安全にお使いいただくために

本製品を安全お使いいただくために､本書に記述されている内容を必ずお守りください｡なお注意事項をお守りいただかない場合、製品の故障､瑕疵担保責任､その他一切の保証をできかねる場合があります。

本製品を使用する前に必ず本書を熟読し理解した上でご使用ください。

■ 中身をお確かめ下さい

お買い求めになった製品の添付品が揃っているかどうか確認してください｡万一､添付品が足りない場合は､すぐにお買い求めの販売店にご連絡ください。

□ボード本体	1枚
□ＣＮ２用Ｉ／Ｏケーブル	1本
□ＣＮ３用コネクタ（５０ｐ）	1セット
□ＣＮ４用コネクタ（３０ｐ）	1セット

なお、取扱説明書､ソフトウェアについては資源節約の為、添付しておりません。追加でご必要の場合はお買い求めの販売店または弊社までご請求ください。また、取扱説明書､ソフトウェアは、弊社ホームページよりダウンロードできます。

URL: http://www.novaelec.co.jp/

■ マニュアルの併用

ＭＣ８０８０Ｐの回路構成は、２個の４軸モータコントロールＩＣ ＭＣＸ３０４をメインとし、ＰＣＩバスのバスインターフェイス回路と各軸のＩ／Ｏインターフェイス回路から成っています。本回路基板の基本的な機能はすべてＭＣＸ３０４に依存していますので、これら機能動作についてはＭＣＸ３０４の取扱説明書を併せてご参照ください。本書では、各軸入出力信号の電気的な仕様、WindowsＯＳ上でのインストール、および本ボードを制御するためのＡＰＩ関数について記述します。また、弊社ホームページには本ボードを制御するプログラムサンプルも用意しておりますので、ご利用ください。

■ 注意･危険

本製品は下記の環境で使用してください。

周囲温度	0～45℃
湿度(非結露)	20～90%
浮遊粉塵	特にひどくないこと
腐食性ガス	ないこと
供給電源	DC+5V(±5%)、外部電源：DC+24V

本製品を正しく使っていただくためにも定期的に点検を行ってください。

ケーブル接続	ボードのコネクタとケーブルが正しく接続されていること。
カードエッジ	汚れ､腐食などがないこと。
コネクタ接続部	汚れ､腐食などがないこと。
ＩＣ、ボード上	いちじるしいほこりや異物が付着していないこと。

■ 本製品の取り扱い

□本製品は静電気防止袋に入っています。本製品を取り扱う際には、人体、衣服の静電気を取り除き、基板の両端面をはさむように持つか、取付金具を持つようにしてください。

□コネクタの端子や実装部品の端子にはできるだけ触れないようにしてください。体が著しく帯電した状態でコネクタ端子や実装部品の端子に触れると、実装されているＣＭＯＳ－ＩＣを破壊する場合があります。特に冬季の乾燥した時期などは注意が必要です。

□衝撃､振動､磁気や静電気の加わる場所での保管や使用は行わないで下さい｡故障や誤動作の原因となります。

□本製品を改造しないで下さい｡改造した場合の故障、誤動作などについては一切の責任を負いません。

□供給電源が通電した状態で本製品や接続ケーブルの挿抜は行わないで下さい｡故障や誤動作の原因となります。

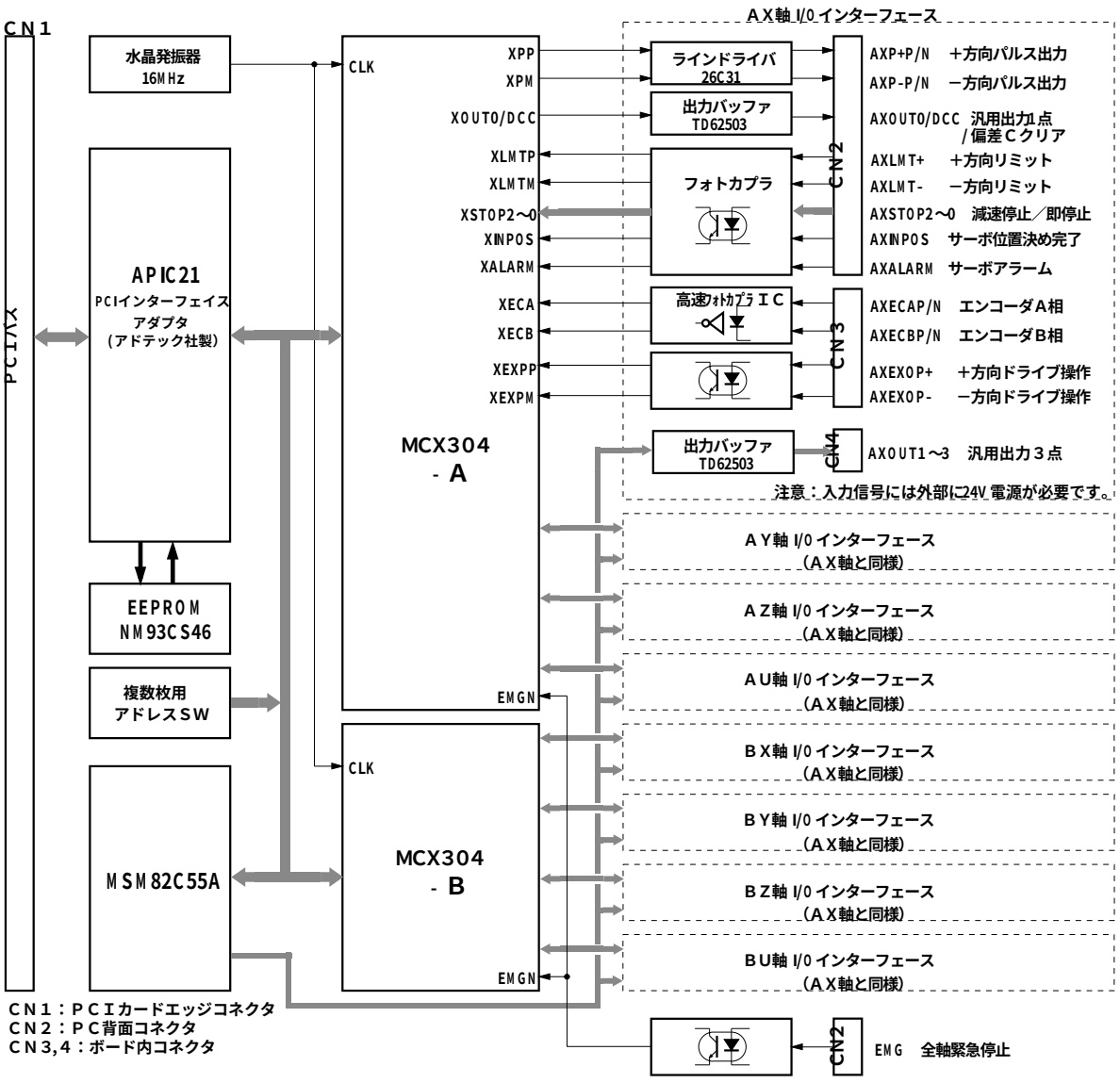
本書の記載内容は、２００６年８月現在のもので、今後、機能の向上などのため予告なしに変更する場合があります。

目　　次

１．概要

ＭＣ８０８０Ｐは、汎用４軸モータコントロールＩＣ ＭＣＸ３０４を２個搭載した、ＰＣＩバス対応の回路基板です。１ボードで８軸のサーボモータ、またはステッピングモータを各軸独立に位置決め制御または速度制御することができます。

下図にＭＣ８０８０Ｐの機能ブロック図を示します。ＭＣ８０８０Ｐは、２個のＭＣＸ３０４をメインに、ＰＣＩバスのインターフェイスと、ＡＸ，ＡＹ，ＡＺ、ＡＵ、ＢＸ，ＢＹ，ＢＺ、ＢＵ各軸のＩ／Ｏインターフェイス回路から構成されています。従って、本回路基板の基本的な機能はすべてＭＣＸ３０４に依存していますので、これら機能動作の詳細についてはＭＣＸ３０４の取扱説明書を併せてご参照ください。



ＭＣ８０８０Ｐ 回路ブロック図

1.1 ＭＣＸ３０４の持つ機能の制限

本ボードでは、MCX304の持つ次の汎用出力信号についてはサポートしていません。しかし、ボード上にはMSM82C55(沖電気製)を搭載していますので、各軸とも４点（MCX304-OUT0、8255-OUT1,2,3）の汎用出力を持っています。

■ OUT1信号出力

MCX304のnSTOP2/OUT1端子は、STOP2入力として使用していますので、OUT1出力としては使用できません。

■ OUT2,3信号出力

データバスを16ビットで行っていますので、D15～D8を使用します。従って、OUT2,3信号は使用できません。

1.2 ＰＣＩバスインターフェイス

■Ｉ／Ｏ占有アドレス

ＰＣＩバスのＩ／Ｏ占有アドレスは、ボード当たり１６使用します。本ボードのＩ／Ｏアドレス指定は、Windows 搭載のプラグアンドプレイ機能によって決定されます。

■データ長

データ長は１６ビットです。バイト単位のリード／ライトアクセスはできません。

■割り込み信号

ＰＣＩバスへの割り込みを使用する場合は、Windows 搭載のプラグアンドプレイ機能によって決定されるＩＲＱを使用します。

1.3 各軸Ｉ／Ｏインターフェイス

■ドライブパルス出力(nP+P/N,nP-P/N)

モータを駆動する＋方向／－方向のドライブパルス出力は、１PPSから最高４MPPSのデューティ５０％のパルスを出力します。各々の方向のドライブパルス出力信号は、AM26C31相当のラインドライバによる差動出力となっています。

■汎用出力(nOUT3～0)

各軸４本の汎用出力があります。出力バッファは、TD62503(東芝製)を使用し、出力耐圧35Vのオープンコレクタ出力です。nOUT0信号だけは、背面コネクタCN2に配置されています。nOUT0信号は、自動原点出し時にサーボモータドライバの偏差カウンタをクリアする信号(DCC)としても使用することができます。nOUT1,2,3信号はボード内コネクタCN4に配置されています。

■オーバランリミット入力(nLMT+,nLMT-)

＋方向、－方向のそれぞれの出力パルスを禁止する入力信号です。モード設定でアクティブ時に即停止／減速停止を選択することができます。この入力信号はフォトカプラで内部回路とは絶縁されています。外部からDC24Vの電源供給が必要です。

■減速停止／即停止入力(nSTOP2～0)

自動原点出し動作などにおいて、ドライブパルスを外部から減速停止または即停止させる入力信号です。有効／無効、アクティブ論理レベルをモード設定することができます。各軸３点用意されています。エンコーダＺ相信号はnSTOP2に入力します。ドライバー側の出力回路がオープンコレクタタイプでも、ラインドライバタイプでも接続できます。この入力信号はフォトカプラで内部回路とは絶縁されています。

■サーボモータ用入力(nINPOS,nALARM)

サーボモータドライバのINPOS（位置決め完了）信号、ALARM（アラーム）信号を入力します。汎用入力信号としても使用することができます。この入力信号はフォトカプラで内部回路とは絶縁されています。

■エンコーダ入力(nECAP/N,nECBP/N)

エンコーダからのＡ／Ｂ相信号を入力します。この信号入力はボード内コネクタCN3に配置されています。nECAP/N,nECBP/N信号は、エンコーダのＡ／Ｂ相信号のための入力で、MCX304内部の32ビット実位置カウンタをカウントアップ／ダウンします。この入力信号は高速フォトカプラＩＣで内部回路とは絶縁されています。差動出力のラインドライバとの接続が容易です。

■外部ドライブ操作入力(nEXOP+,nEXOP-)

外部から＋方向／－方向のドライブを起動する入力です。この信号入力はボード内コネクタCN3に配置されています。定量ドライブモードでは、入力信号のトリガ（立ち下がり）で指定ドライブパルスが出力されます。また、連続ドライブモードにすると、入力信号がLowレベルの間だけ、連続してドライブパルスを出し続けます。各軸のマニュアルジョグ送り等において、ＣＰＵの介在なしに応答性の速い軸送り動作が可能となります。この入力信号はフォトカプラで内部回路とは絶縁されています。

■緊急停止入力(EMG)

全軸のドライブを緊急停止させる入力信号です。ボード上のジャンパ選択でアクティブ論理レベルを設定することができます。この入力信号はフォトカプラで内部回路とは絶縁されています。

２．Ｉ／Ｏアドレス設定とリード／ライトレジスタ

ボードのＩ／Ｏポートアドレスは、ＰＣＩバスのプラグアンドプレイ機能(以下ＰｎＰ機能)によって決定されます。本ボードの２個のＭＣＸ３０４（－Ａ，Ｂ）内のリード／ライトレジスタ、およびＭＳＭ８２Ｃ５５のポートは、ＭＣ８０８０Ｐデバイスドライバが提供するＡＰＩ関数によってアクセスします。各レジスタ、ポートのＩ／Ｏアドレスは下表のようになります。ＭＣＸ３０４－ＡはＡＸ，ＡＹ，ＡＺ，ＡＵ軸を、ＭＣＸ３０４－ＢはＢＸ，ＢＹ，ＢＺ，ＢＵ軸を制御します。各レジスタの詳細は、ＭＣＸ３０４取扱説明書４章を参照してください。

I/Oアドレス	MCX304チップ	ライトレジスタ	リードレジスタ
00	MCX304-A	WR0　コマンドレジスタ	RR0　主ステータスレジスタ
01		XWR1　Ｘ軸モードレジスタ１ YWR1　Ｙ軸モードレジスタ１ ZWR1　Ｚ軸モードレジスタ１ UWR1　Ｕ軸モードレジスタ１	XRR1　Ｘ軸ステータスレジスタ１ YRR1　Ｙ軸ステータスレジスタ１ ZRR1　Ｚ軸ステータスレジスタ１ URR1　Ｕ軸ステータスレジスタ１
02		XWR2　Ｘ軸モードレジスタ２ YWR2　Ｙ軸モードレジスタ２ ZWR2　Ｚ軸モードレジスタ２ UWR2　Ｕ軸モードレジスタ２	XRR2　Ｘ軸ステータスレジスタ２ YRR2　Ｙ軸ステータスレジスタ２ ZRR2　Ｚ軸ステータスレジスタ２ URR2　Ｕ軸ステータスレジスタ２
03		XWR3　Ｘ軸モードレジスタ３ YWR3　Ｙ軸モードレジスタ３ ZWR3　Ｚ軸モードレジスタ３ UWR3　Ｕ軸モードレジスタ３	XRR3　Ｘ軸ステータスレジスタ３ YRR3　Ｙ軸ステータスレジスタ３ ZRR3　Ｚ軸ステータスレジスタ３ URR3　Ｕ軸ステータスレジスタ３
04		WR4　アウトプットレジスタ１	RR4　インプットレジスタ１
05		WR5　アウトプットレジスタ２	RR5　インプットレジスタ２
06		WR6　ライトデータレジスタ１	RR6　リードデータレジスタ１
07		WR7　ライトデータレジスタ２	RR7　リードデータレジスタ２
08	MCX304-B	WR0　コマンドレジスタ	RR0　主ステータスレジスタ
09		XWR1　Ｘ軸モードレジスタ１ YWR1　Ｙ軸モードレジスタ１ ZWR1　Ｚ軸モードレジスタ１ UWR1　Ｕ軸モードレジスタ１	XRR1　Ｘ軸ステータスレジスタ１ YRR1　Ｙ軸ステータスレジスタ１ ZRR1　Ｚ軸ステータスレジスタ１ URR1　Ｕ軸ステータスレジスタ１
0A		XWR2　Ｘ軸モードレジスタ２ YWR2　Ｙ軸モードレジスタ２ ZWR2　Ｚ軸モードレジスタ２ UWR2　Ｕ軸モードレジスタ２	XRR2　Ｘ軸ステータスレジスタ２ YRR2　Ｙ軸ステータスレジスタ２ ZRR2　Ｚ軸ステータスレジスタ２ URR2　Ｕ軸ステータスレジスタ２
0B		XWR3　Ｘ軸モードレジスタ３ YWR3　Ｙ軸モードレジスタ３ ZWR3　Ｚ軸モードレジスタ３ UWR3　Ｕ軸モードレジスタ３	XRR3　Ｘ軸ステータスレジスタ３ YRR3　Ｙ軸ステータスレジスタ３ ZRR3　Ｚ軸ステータスレジスタ３ URR3　Ｕ軸ステータスレジスタ３
0C		WR4　アウトプットレジスタ１	RR4　インプットレジスタ１
0D		WR5　アウトプットレジスタ２	RR5　インプットレジスタ２
0E		WR6　ライトデータレジスタ１	RR6　リードデータレジスタ１
0F		WR7　ライトデータレジスタ２	RR7　リードデータレジスタ２
10	MSM82C55	ポートＡ出力データ (D7～D0にセット)	ポートＡ出力データの読み出し (D7～D0)
11		ポートＢ出力データ (D7～D0にセット)	ポートＢ出力データの読み出し
12		ポートＣ出力データ (D7～D0にセット)	ポートＣ出力データの読み出し
13		コントロールレジスタ　注意１ (D7～D0にセット)	無　効

注意１:MSM8255のコントロールレジスタは、電源投入時にシステムがポートＡ，Ｂ，Ｃを出力モードにセットしますので、このレジスタにはデータを書き込まないでください。

３．入出力信号

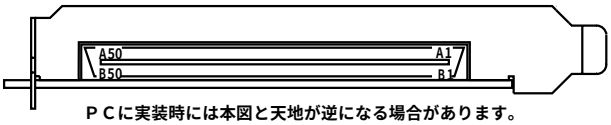
この章では、Ｉ／Ｏコネクタの各入出力信号について記述します。信号の説明、およびインターフェース回路では、各軸の信号名をｎ○○○○と記述していますが、この"ｎ"はＡＸ、ＡＹ、ＡＺ、ＡＵ，ＢＸ、ＢＹ、ＢＺ、ＢＵを表しています。

3.1 ＣＮ２コネクタ（背面コネクタ）

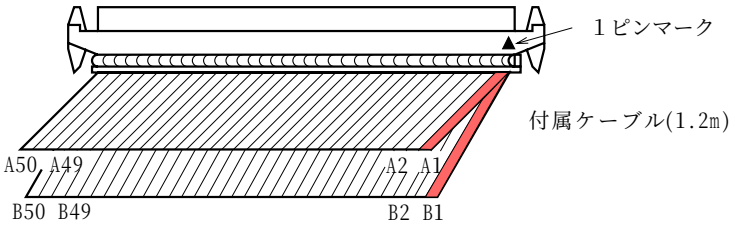
ＣＮ２コネクタには、外部電源(+24VDC)入力と、下表に示す各軸の信号を入出力しています。

コネクタ	信号の種類	信号名
CN2	＋方向/－方向のドライブパルス出力信号	nP+P/N，nP-P/N
	＋方向/－方向のオーバランリミット入力信号	nLMT+，nLMT-
	減速停止/即停止入力信号 3点	nSTOP0，nSTOP1，nSTOP2
	サーボモータ用インポジションとアラーム入力信号	nINPOS，nALARM
	サーボモータ用偏差カウンタクリア出力信号 兼 汎用出力 1点）	nOUT0/DCC
	全軸緊急停止入力信号	EMG

ＣＮ２コネクタ　ピン配置



ＰＣに実装時には本図と天地が逆になる場合があります。



付属ケーブルは、上図に示すように、コネクタの１ピンマーク（三角印）を右上にしたとき、上側ケーブルの右（赤線）から左に向かってA1，A2，… A49，A50、下側ケーブルの右（赤線）から左に向かってB1，B2，… B49，B50となります。

コネクタ型式：ボード側 FX2B-100P-1.27DS（ヒロセ），付属ケーブル側 FX2B-100S-1.27R（ヒロセ）

ＣＮ２コネクタ

ピン	信号名	入／出	内　　容	説明	ピン	信号名	入／出	内　　容	説明
A1	VEX		外部電源（+24V）	3.12	B1	EMG	入力	緊急停止（全軸共通）	3.11
A2	AXLMT+	入力	ＡＸ軸＋方向リミット	3.5	B2	BXLMT+	入力	ＢＸ軸＋方向リミット	3.5
A3	AXLMT-	入力	ＡＸ軸－方向リミット	3.5	B3	BXLMT-	入力	ＢＸ軸－方向リミット	3.5
A4	AXSTOP0	入力	ＡＸ軸減速停止／即停止	3.6	B4	BXSTOP0	入力	ＢＸ軸減速停止／即停止	3.6
A5	AXSTOP1	入力	ＡＸ軸減速停止／即停止	3.6	B5	BXSTOP1	入力	ＢＸ軸減速停止／即停止	3.6
A6	AYLMT+	入力	ＡＹ軸＋方向リミット	3.5	B6	BYLMT+	入力	ＢＹ軸＋方向リミット	3.5
A7	AYLMT-	入力	ＡＹ軸－方向リミット	3.5	B7	BYLMT-	入力	ＢＹ軸－方向リミット	3.5
A8	AYSTOP0	入力	ＡＹ軸減速停止／即停止	3.6	B8	BYSTOP0	入力	ＢＹ軸減速停止／即停止	3.6
A9	AYSTOP1	入力	ＡＹ軸減速停止／即停止	3.6	B9	BYSTOP1	入力	ＢＹ軸減速停止／即停止	3.6
A10	AZLMT+	入力	ＡＺ軸＋方向リミット	3.5	B10	BZLMT+	入力	ＢＺ軸＋方向リミット	3.5
A11	AZLMT-	入力	ＡＺ軸－方向リミット	3.5	B11	BZLMT-	入力	ＢＺ軸－方向リミット	3.5
A12	AZSTOP0	入力	ＡＺ軸減速停止／即停止	3.6	B12	BZSTOP0	入力	ＢＺ軸減速停止／即停止	3.6
A13	AZSTOP1	入力	ＡＺ軸減速停止／即停止	3.6	B13	BZSTOP1	入力	ＢＺ軸減速停止／即停止	3.6
A14	AULMT+	入力	ＡＵ軸＋方向リミット	3.5	B14	BULMT+	入力	ＢＵ軸＋方向リミット	3.5
A15	AULMT-	入力	ＡＵ軸－方向リミット	3.5	B15	BULMT-	入力	ＢＵ軸－方向リミット	3.5
A16	AUSTOP0	入力	ＡＵ軸減速停止／即停止	3.6	B16	BUSTOP0	入力	ＢＵ軸減速停止／即停止	3.6
A17	AUSTOP1	入力	ＡＵ軸減速停止／即停止	3.6	B17	BUSTOP1	入力	ＢＵ軸減速停止／即停止	3.6
A18	AXSTOP2	入力	ＡＸ軸エンコーダＺ相	3.7	B18	BXSTOP2	入力	ＢＸ軸エンコーダＺ相	3.7
A19	AXINPOS	入力	ＡＸ軸位置決め完了	3.8	B19	BXINPOS	入力	ＢＸ軸位置決め完了	3.8
A20	AXALARM	入力	ＡＸ軸アラーム	3.8	B20	BXALARM	入力	ＢＸ軸アラーム	3.8
A21	AYSTOP2	入力	ＡＹ軸エンコーダＺ相	3.7	B21	BYSTOP2	入力	ＢＹ軸エンコーダＺ相	3.7
A22	AYINPOS	入力	ＡＹ軸位置決め完了	3.8	B22	BYINPOS	入力	ＢＹ軸位置決め完了	3.8
A23	AYALARM	入力	ＡＹ軸アラーム	3.8	B23	BYALARM	入力	ＢＹ軸アラーム	3.8
A24	AZSTOP2	入力	ＡＺ軸エンコーダＺ相	3.7	B24	BZSTOP2	入力	ＢＺ軸エンコーダＺ相	3.7
A25	AZINPOS	入力	ＡＺ軸位置決め完了	3.8	B25	BZINPOS	入力	ＢＺ軸位置決め完了	3.8
A26	AZALARM	入力	ＡＺ軸アラーム	3.8	B26	BZALARM	入力	ＢＺ軸アラーム	3.8
A27	AUSTOP2	入力	ＡＵ軸エンコーダＺ相	3.7	B27	BUSTOP2	入力	ＢＵ軸エンコーダＺ相	3.7
A28	AUINPOS	入力	ＡＵ軸位置決め完了	3.8	B28	BUINPOS	入力	ＢＵ軸位置決め完了	3.8
A29	AUALARM	入力	ＡＵ軸アラーム	3.8	B29	BUALARM	入力	ＢＵ軸アラーム	3.8
A30	GND		内部回路ＧＮＤ		B30	GND		内部回路ＧＮＤ	
A31	AXOUT0/DCC	出力	ＡＸ軸汎用出力／ＤＣＣ注1	3.4	B31	BXOUT0/DCC	出力	ＢＸ軸汎用出力／ＤＣＣ注1	3.4
A32	AYOUT0/DCC	出力	ＡＹ軸汎用出力／ＤＣＣ	3.4	B32	BYOUT0/DCC	出力	ＢＹ軸汎用出力／ＤＣＣ	3.4
A33	AZOUT0/DCC	出力	ＡＺ軸汎用出力／ＤＣＣ	3.4	B33	BZOUT0/DCC	出力	ＢＺ軸汎用出力／ＤＣＣ	3.4
A34	AUOUT0/DCC	出力	ＡＵ軸汎用出力／ＤＣＣ	3.4	B34	BUOUT0/DCC	出力	ＢＵ軸汎用出力／ＤＣＣ	3.4
A35	AXP+P	出力	ＡＸ軸＋方向ドライブパルス	3.3	B35	BXP+P	出力	ＢＸ軸＋方向ドライブパルス	3.3
A36	AXP+N	出力	ＡＸ軸＋方向ドライブパルス	3.3	B36	BXP+N	出力	ＢＸ軸＋方向ドライブパルス	3.3
A37	AXP-P	出力	ＡＸ軸－方向ドライブパルス	3.3	B37	BXP-P	出力	ＢＸ軸－方向ドライブパルス	3.3
A38	AXP-N	出力	ＡＸ軸－方向ドライブパルス	3.3	B38	BXP-N	出力	ＢＸ軸－方向ドライブパルス	3.3
A39	AYP+P	出力	ＡＹ軸＋方向ドライブパルス	3.3	B39	BYP+P	出力	ＢＹ軸＋方向ドライブパルス	3.3
A40	AYP+N	出力	ＡＹ軸＋方向ドライブパルス	3.3	B40	BYP+N	出力	ＢＹ軸＋方向ドライブパルス	3.3
A41	AYP-P	出力	ＡＹ軸－方向ドライブパルス	3.3	B41	BYP-P	出力	ＢＹ軸－方向ドライブパルス	3.3
A42	AYP-N	出力	ＡＹ軸－方向ドライブパルス	3.3	B42	BYP-N	出力	ＢＹ軸－方向ドライブパルス	3.3
A43	AZP+P	出力	ＡＺ軸＋方向ドライブパルス	3.3	B43	BZP+P	出力	ＢＺ軸＋方向ドライブパルス	3.3
A44	AZP+N	出力	ＡＺ軸＋方向ドライブパルス	3.3	B44	BZP+N	出力	ＢＺ軸＋方向ドライブパルス	3.3
A45	AZP-P	出力	ＡＺ軸－方向ドライブパルス	3.3	B45	BZP-P	出力	ＢＺ軸－方向ドライブパルス	3.3
A46	AZP-N	出力	ＡＺ軸－方向ドライブパルス	3.3	B46	BZP-N	出力	ＢＺ軸－方向ドライブパルス	3.3
A47	AUP+P	出力	ＡＵ軸＋方向ドライブパルス	3.3	B47	BUP+P	出力	ＢＵ軸＋方向ドライブパルス	3.3
A48	AUP+N	出力	ＡＵ軸＋方向ドライブパルス	3.3	B48	BUP+N	出力	ＢＵ軸＋方向ドライブパルス	3.3
A49	AUP-P	出力	ＡＵ軸－方向ドライブパルス	3.3	B49	BUP-P	出力	ＢＵ軸－方向ドライブパルス	3.3
A50	AUP-N	出力	ＡＵ軸－方向ドライブパルス	3.3	B50	BUP-N	出力	ＢＵ軸－方向ドライブパルス	3.3

注１：ＤＣＣ(Deviation Counter Clear)：サーボモータドライバの偏差カウンタをクリアさせるための出力。

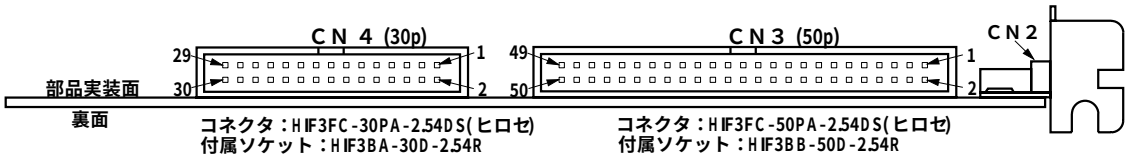
注意：ＣＮ２コネクタにケーブルを挿抜する場合は、まずパソコンの電源をＯＦＦの状態にし、ケーブルに供給している外部電源（ＤＣ＋２４Ｖ等）をＯＦＦにしてから挿抜して下さい。尚、ケーブルを挿入する場合はコネクタの向きに注意し、逆挿しにならないようにして下さい。パソコンや外部電源がＯＮのまま接続した場合、基板の内部回路等が破損する場合があります。

3.2 ＣＮ３,４コネクタ （ボード内コネクタ）

ボード内のＣＮ３、４コネクタには、以下に示す各軸の信号を入出力しています。

コネクタ	信号の種類	信号名
CN3	＋方向/－方向の外部ドライブ操作入力信号（汎用入力に使用可） エンコーダＡ／Ｂ相入力信号	nEXOP+,nEXOP- nECAP/N,nECBP/N
CN4	汎用出力信号（3点/軸）	nOUT1,2,3

ＣＮ３，４コネクタ　ピン配置



ＣＮ３コネクタ

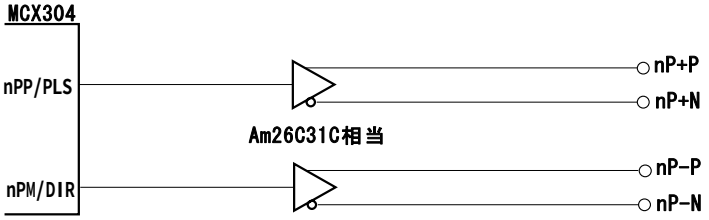
ピン	信号名	入／出	内　　容	説明	ピン	信号名	入／出	内　　容	説明
1	VEX		外部電源（+24V）	3.12	2	VEX		外部電源（+24V）	3.12
3	AXEXOP+	入力	ＡＸ軸＋方向ドライブ操作	3.10	4	AXEXOP-	入力	ＡＸ軸－方向ドライブ操作	3.10
5	AYEXOP+	入力	ＡＹ軸＋方向ドライブ操作	3.10	6	AYEXOP-	入力	ＡＹ軸－方向ドライブ操作	3.10
7	AZEXOP+	入力	ＡＺ軸＋方向ドライブ操作	3.10	8	AZEXOP-	入力	ＡＺ軸－方向ドライブ操作	3.10
9	AUEXOP+	入力	ＡＵ軸＋方向ドライブ操作	3.10	10	AUEXOP-	入力	ＡＵ軸－方向ドライブ操作	3.10
11	BXEXOP+	入力	ＢＸ軸＋方向ドライブ操作	3.10	12	BXEXOP-	入力	ＢＸ軸－方向ドライブ操作	3.10
13	BYEXOP+	入力	ＢＹ軸＋方向ドライブ操作	3.10	14	BYEXOP-	入力	ＢＹ軸－方向ドライブ操作	3.10
15	BZEXOP+	入力	ＢＺ軸＋方向ドライブ操作	3.10	16	BZEXOP-	入力	ＢＺ軸－方向ドライブ操作	3.10
17	BUEXOP+	入力	ＢＵ軸＋方向ドライブ操作	3.10	18	BUEXOP-	入力	ＢＵ軸－方向ドライブ操作	3.10
19	AXECAP	入力	ＡＸ軸エンコーダＡ相(正)	3.9	20	AXECAN	入力	ＡＸ軸エンコーダＡ相(反)	3.9
21	AXECBP	入力	ＡＸ軸エンコーダＢ相(正)	3.9	22	AXECBN	入力	ＡＸ軸エンコーダＢ相(反)	3.9
23	AYECAP	入力	ＡＹ軸エンコーダＡ相(正)	3.9	24	AYECAN	入力	ＡＹ軸エンコーダＡ相(反)	3.9
25	AYECBP	入力	ＡＹ軸エンコーダＢ相(正)	3.9	26	AYECBN	入力	ＡＹ軸エンコーダＢ相(反)	3.9
27	AZECAP	入力	ＡＺ軸エンコーダＡ相(正)	3.9	28	AZECAN	入力	ＡＺ軸エンコーダＡ相(反)	3.9
29	AZECBP	入力	ＡＺ軸エンコーダＢ相(正)	3.9	30	AZECBN	入力	ＡＺ軸エンコーダＢ相(反)	3.9
31	AUECAP	入力	ＡＵ軸エンコーダＡ相(正)	3.9	32	AUECAN	入力	ＡＵ軸エンコーダＡ相(反)	3.9
33	AUECBP	入力	ＡＵ軸エンコーダＢ相(正)	3.9	34	AUECBN	入力	ＡＵ軸エンコーダＢ相(反)	3.9
35	BXECAP	入力	ＢＸ軸エンコーダＡ相(正)	3.9	36	BXECAN	入力	ＢＸ軸エンコーダＡ相(反)	3.9
37	BXECBP	入力	ＢＸ軸エンコーダＢ相(正)	3.9	38	BXECBN	入力	ＢＸ軸エンコーダＢ相(反)	3.9
39	BYECAP	入力	ＢＹ軸エンコーダＡ相(正)	3.9	40	BYECAN	入力	ＢＹ軸エンコーダＡ相(反)	3.9
41	BYECBP	入力	ＢＹ軸エンコーダＢ相(正)	3.9	42	BYECBN	入力	ＢＹ軸エンコーダＢ相(反)	3.9
43	BZECAP	入力	ＢＺ軸エンコーダＡ相(正)	3.9	44	BZECAN	入力	ＢＺ軸エンコーダＡ相(反)	3.9
45	BZECBP	入力	ＢＺ軸エンコーダＢ相(正)	3.9	46	BZECBN	入力	ＢＺ軸エンコーダＢ相(反)	3.9
47	BUECAP	入力	ＢＵ軸エンコーダＡ相(正)	3.9	48	BUECAN	入力	ＢＵ軸エンコーダＡ相(反)	3.9
49	BUECBP	入力	ＢＵ軸エンコーダＢ相(正)	3.9	50	BUECBN	入力	ＢＵ軸エンコーダＢ相(反)	3.9

ＣＮ４コネクタ

ピン	信号名	入／出	内　　容	説明	ピン	信号名	入／出	内　　容	説明
1	VEX		外部電源（+24V）	3.12	2	VEX		外部電源（+24V）	3.12
3	AXOUT1/PA0	出力	ＡＸ軸汎用出力	3.4	4	AXOUT2/PA1	出力	ＡＸ軸汎用出力	3.4
5	AXOUT3/PA2	出力	ＡＸ軸汎用出力	3.4	6	AYOUT1/PA3	出力	ＡＹ軸汎用出力	3.4
7	AYOUT2/PA4	出力	ＡＹ軸汎用出力	3.4	8	AYOUT3/PA5	出力	ＡＹ軸汎用出力	3.4
9	AZOUT1/PA6	出力	ＡＺ軸汎用出力	3.4	10	AZOUT2/PA7	出力	ＡＺ軸汎用出力	3.4
11	AZOUT3/PB0	出力	ＡＺ軸汎用出力	3.4	12	AUOUT1/PB1	出力	ＡＵ軸汎用出力	3.4
13	AUOUT2/PB2	出力	ＡＵ軸汎用出力	3.4	14	AUOUT3/PB3	出力	ＡＵ軸汎用出力	3.4
15	BXOUT1/PB4	出力	ＢＸ軸汎用出力	3.4	16	BXOUT2/PB5	出力	ＢＸ軸汎用出力	3.4
17	BXOUT3/PB6	出力	ＢＸ軸汎用出力	3.4	18	BYOUT1/PB7	出力	ＢＹ軸汎用出力	3.4
19	BYOUT2/PC0	出力	ＢＹ軸汎用出力	3.4	20	BYOUT3/PC1	出力	ＢＹ軸汎用出力	3.4
21	BZOUT1/PC2	出力	ＢＺ軸汎用出力	3.4	22	BZOUT2/PC3	出力	ＢＺ軸汎用出力	3.4
23	BZOUT3/PC4	出力	ＢＺ軸汎用出力	3.4	24	BUOUT1/PC5	出力	ＢＵ軸汎用出力	3.4
25	BUOUT2/PC6	出力	ＢＵ軸汎用出力	3.4	26	BUOUT3/PC7	出力	ＢＵ軸汎用出力	3.4
27	GND		内部回路ＧＮＤ		28	GND		内部回路ＧＮＤ	
29	GND		内部回路ＧＮＤ		30	GND		内部回路ＧＮＤ	

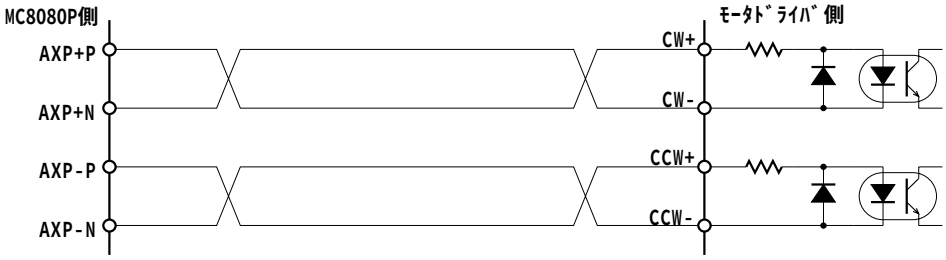
3.3 ドライブパルス出力信号（nP+P,nP+N,nP-P,nP-N）

ドライブパルス出力信号は、MCX304の＋方向／－方向のドライブパルス信号を差動出力のラインドライバ（AM26C31相当）を介して出力しています。nP+NはnP+Pの反転出力、nP-NはnP-Pの反転出力です。リセット時には、正出力（nP+P,nP-P）がLowレベル、反転出力（nP+N,nP-N）がHiレベルになっています。
ドライブパルス出力は、リセット直後は＋／－方向の独立２パルス方式になっていますが、モード設定によって方向・１パルス方式にすることもできます。MCX304取扱説明書2.6.2節、4.5節を参照してください。

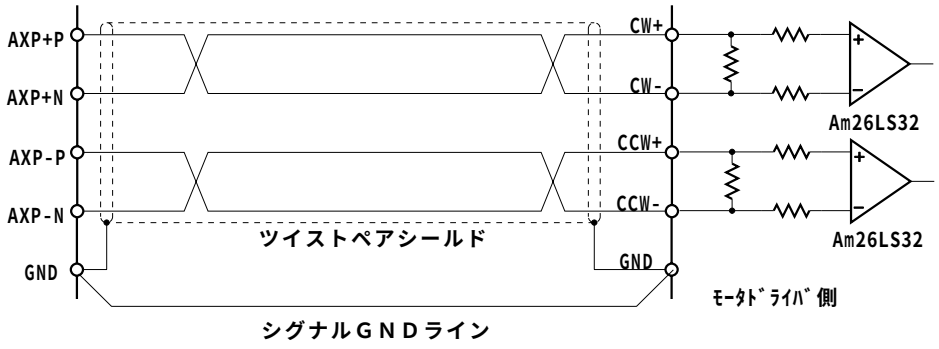


ドライブパルス出力信号回路

下図にフォトカプラ入力回路、およびラインレシーバ入力回路を持つモータドライバとの接続例を示します。



フォトカプラ入力回路のモータドライバとの接続例



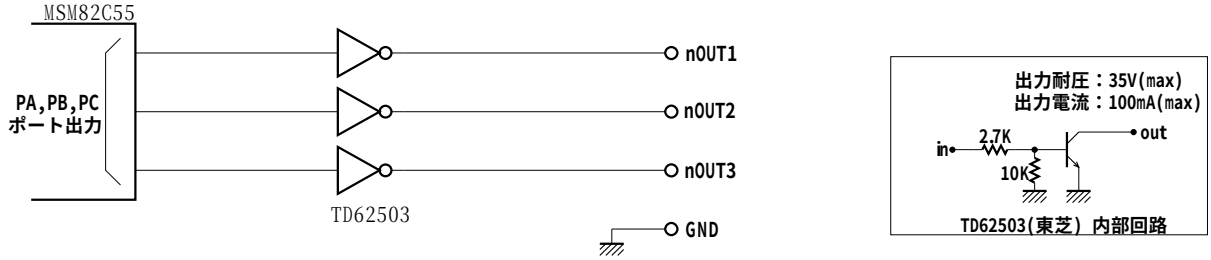
ラインレシーバ入力回路のモータドライバとの接続例

注1. ラインレシーバ入力回路を使用する場合の注意
上記のように、ラインレシーバ入力回路を使用する場合は、ラインドライバ側とモータドライバ側をシグナルＧＮＤラインで接続下さい。機器間のシグナルＧＮＤに電位差がありますと、ドライバ回路、モータドライバ回路の損傷に繋がることがあります。上図のようにシグナルＧＮＤを別途結んでご使用下さい。

3.4 汎用出力信号と偏差カウンタクリア出力信号（nOUT3,nOUT2,nOUT1,nOUT0/DCC）

汎用出力信号は、MCX304のnOUT0/DCC出力信号１点と、MSM82C55の出力信号からの３点の合計４点/軸を、バッファ（TD62503）を介して出力しています。nOUT0信号は偏差カウンタクリア出力(DCC)と兼用で、ＣＮ２コネクタから出力されています。また、他の汎用出力信号nOUT3,2,1はＣＮ４コネクタから出力されています。リセット時にはすべての出力信号（オープンコレクタ出力）がＯＦＦしています。

■ nOUT3,2,1出力



汎用出力信号回路

nOUT3～1汎用出力信号は、ＣＮ４に出力されています。本信号のＯＮ／ＯＦＦ制御は、MSM82C52（沖電気）の各ポートに書き込むことにより行います。ＰＣシステム起動時には各ポートとも出力にセットされ、ＯＦＦ状態になります。

nOUT3,2,1出力信号のＯＮ／ＯＦＦ制御は、プログラム上で次の(1)から(4)のように行います。

(1)現在設定されている出力データを読み込む。

InpMC8080P関数を使って、下の汎用出力信号/レジスタ・ビット対応表から、ＯＮ／ＯＦＦさせたい信号に対応するポートに現在の設定されている出力データを読み込みます。

```
mcb0pa = InpMC8080P(ボード番号, ポートアドレス);
                    ox10,0x11,0x12
                    0x0～0x9
```

例）　`mcb0pa = InpMC8080P(0x0, 0x10);`

(2)出力をＯＮする場合。

読みとった現在設定データに対して、ＯＮさせたい信号に対応するビットを１にします。

例）AXOUT1出力信号をＯＮする。　`mcb0pa = mcb0pa || 0x01;`　下表よりAXOUT1は0x10ポートアドレスのbit0。

(3)出力をＯＦＦする場合。

読みとった現在設定データに対して、ＯＦＦさせたい信号に対応するビットを０にします。

例）AXOUT1出力信号をＯＦＦする。 `mcb0pa = mcb0pa && 0xfe;`

(4)出力データをセットする。

OutpMC8080P(ボード番号, ポートアドレス, 出力データ)

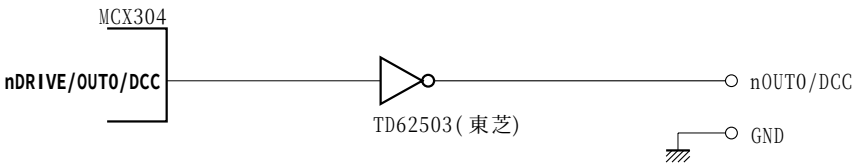
例）　`OutpMC8080P(0x0, 0x10, mcb0pa);`

汎用出力信号/レジスタ・ビット対応表

CN4ピン	信号名	ポートアドレス	ビット	CN4ピン	信号名	ポートアドレス	ビット
3	AXOUT1/PA0	0x10	0	4	AXOUT2/PA1	0x10	1
5	AXOUT3/PA2	0x10	2	6	AYOUT1/PA3	0x10	3
7	AYOUT2/PA4	0x10	4	8	AYOUT3/PA5	0x10	5
9	AZOUT1/PA6	0x10	6	10	AZOUT2/PA7	0x10	7
11	AZOUT3/PB0	0x11	0	12	AUOUT1/PB1	0x11	1
13	AUOUT2/PB2	0x11	2	14	AUOUT3/PB3	0x11	3
15	BXOUT1/PB4	0x11	4	16	AXOUT2/PB5	0x11	5
17	BXOUT3/PB6	0x11	6	18	AYOUT1/PB7	0x11	7
19	BYOUT2/PC0	0x12	0	20	AYOUT3/PC1	0x12	1
21	BZOUT1/PC2	0x12	2	22	AZOUT2/PC3	0x12	3
23	BZOUT3/PC4	0x12	4	24	AUOUT1/PC5	0x12	5
25	BUOUT2/PC6	0x12	6	26	AUOUT3/PC7	0x12	7

nOUT3,2,1出力信号のＯＮ／ＯＦＦ制御例は、サンプルプログラム内にも記載されていますので、参照してください。

■ nOUT0/DCC出力



nOUT0/DCC出力は、汎用出力信号(nOUT0)信号と偏差カウンタクリア出力(DCC)との兼用端子で、ＣＮ２に出力されています。リセット時にはＯＦＦ状態になっています。本信号の制御は、他の汎用出力とは異なり、MCX304のレジスタ書込によって行います。

汎用出力として使用する場合、

(1)nOUT0出力を有効にするために、MCX304/WR5のnOT0Eビットを１にセットする。

(2)ＯＮする：MCX304/WR4のnOUT0ビットを１セット。　ＯＦＦする：MCX304/WR4のnOUT0ビットを０セット。

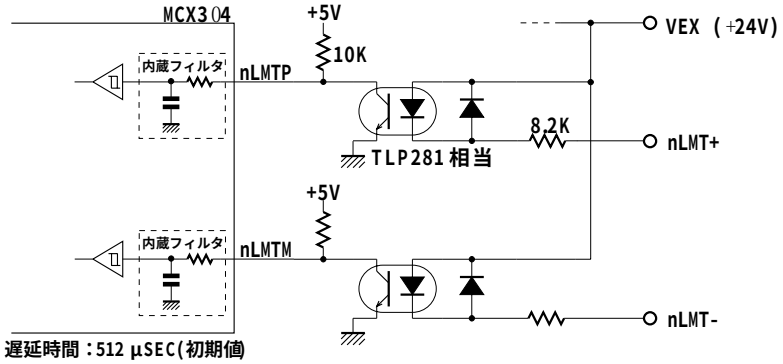
偏差カウンタクリア出力として使用する場合、

偏差カウンタクリア出力は、サーボモータドライバに対して、ドライバ内の偏差カウンタをクリアするための出力です。MCX304は、自動原点出しの中でこの偏差カウンタクリア信号を出力する機能を持っています。
偏差カウンタクリアの有効、論理レベル、パルス幅を設定する。詳細はMCX304マニュアル2.4.3節を、また自動原点出しについての詳細は2.4節をご参照ください。ボードでは、上図のようにTD62503をバッファーにしていますので、MCX304出力がアクティブHiのとき、ボード出力（オープンコレクタ出力）はＯＮします。

3.5 オーバランリミット入力信号（nLMT+,nLMT-）

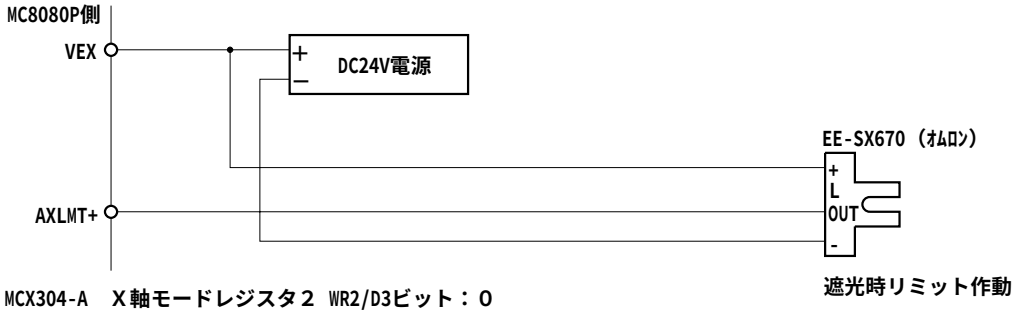
＋方向、－方向のそれぞれのドライブパルスを抑止する入力信号です。この入力信号はフォトカプラ回路を通してMCX304のリミット入力に接続されています。リセット直後は、MCX304はLowレベルでアクティブになりますので、信号端子（nLMT+,nLMT-）より電流が流出するときリミット機能が作動します。モード設定の詳細は、MCX304取扱説明書4.5節を参照してください。

この信号を動作させるには、外部からDC24Vの電源供給が必要です。この信号の内蔵フィルタは、ボードが電源オンされた初期時には信号遅延時間512μの設定になります。システムのノイズ環境によって、この信号遅延時間は変更することが可能です。詳細はMCX304マニュアル2.6.9節および4.6節を参照してください。



オーバランリミット入力信号回路

下図にオーバランリミット入力信号をフォトマイクロセンサに接続する例を示します。ＡＸ軸のモードレジスタ２（XWR2）のD3ビットを０（リセット時のモード）にすると、遮光時にリミット機能が作動します。



オーバランリミット入力信号とフォトマイクロセンサとの接続例

配線を長く引き回す場合は、シールド線を使用してください。

3.6 減速停止／即停止入力信号（nSTOP0,nSTOP1）

ドライブパルス出力を途中で減速停止または即停止させるための入力信号です。通常、nSTOP0信号は原点近傍信号として使用し、nSTOP1信号は原点信号として使用します。MCX304は原点出し用入力信号として各軸ともSTOP2～STOP0の３点持っていますが、本ボードでは、STOP2はエンコーダＺ相のためのインターフェイス回路が組まれています。STOP1信号は原点、STOP0信号は原点近傍の入力信号として使用します。

［有効/無効と論理設定］

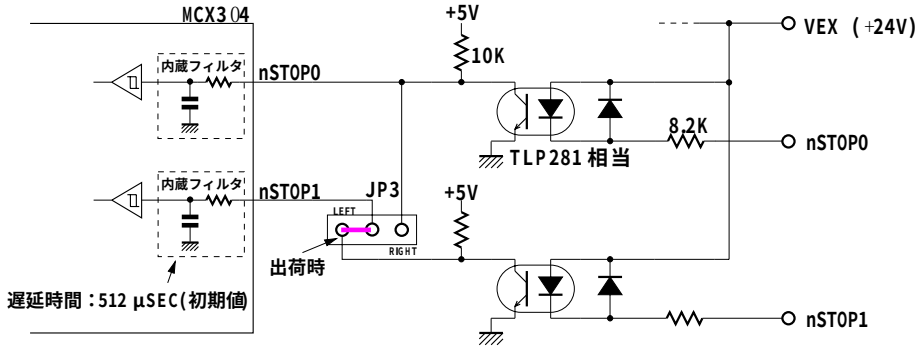
それぞれの信号は、有効／無効、論理レベルをモード設定することができます。有効にモード設定すると、ドライブの途中で本信号がアクティブになるとドライブパルス出力を停止します。加減速ドライブ中であれば減速停止、定速ドライブ中であれば即

停止します。リセット直後は、全信号が無効になっています。例えば、ＡＸ軸STOP0信号において、A-XWR1レジスタD1,D0ビットを1,0にセットし、Lowレベルで有効にすると、本ボードのAXSTOP0信号端子(CN2-A4)より電流が流出するとドライブが停止します。モード設定の詳細は、MCX304取扱説明書4.4節を参照してください。

［自動原点出し］

MCX304は、自動原点出し機能を持っています。詳細はMCX304取扱説明書2.4節を参照してください。

１つの信号だけで高速原点サーチ→低速追い込みを行う場合にはnSTOP0信号を使用し、下図のＪＰ３ジャンパーをRIGHT側に切り替えます。



減速停止／即停止入力信号回路

この信号を動作させるには、外部からDC24Vの電源供給が必要です。この信号の内蔵フィルタは、ボードが電源オンされた初期時には信号遅延時間512μの設定になります。システムのノイズ環境によって、この信号遅延時間は変更することが可能です。詳細はMCX304マニュアル2.6.9節および4.6節を参照してください。

この信号はインプットレジスタ１，２（RR4,5)で信号状態を常時読み出せますので汎用入力としても使用することができます。

3.7 エンコーダＺ相入力信号（nSTOP2）

nSTOP2入力信号はエンコーダ、またはサーボモータドライバのＺ相出力信号を接続して、ドライブパルス出力を途中で停止させるための入力です。

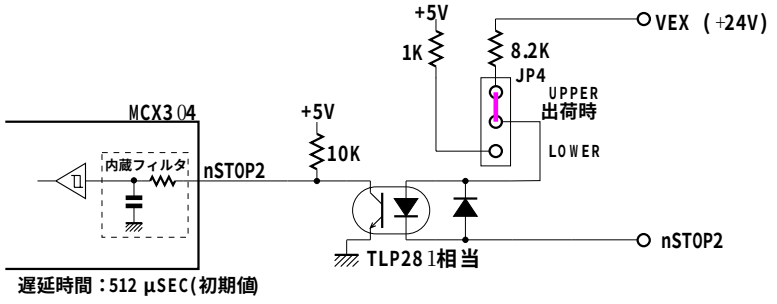
この信号を動作させるには、外部からDC24Vの電源供給が必要です。この信号の内蔵フィルタは、ボードが電源オンされた初期時には信号遅延時間512μの設定になります。システムのノイズ環境によって、この信号遅延時間は変更することが可能です。詳細はMCX304マニュアル2.6.9節および4.6節を参照してください。

［有効/無効と論理設定］

nSTOP2信号もnSTOP1,0信号と同様に有効／無効、論理レベルをモード設定することができます。例えば、WR1レジスタD5,D4ビットを1,0にセットし、Lowレベルで有効にすると、本ボードのnSTOP2信号端子より電流が流出するとドライブが停止します。モード設定の詳細は、MCX304取扱説明書4.4節を参照してください。

［ジャンパー設定］

本入力信号は、相手出力側がオープンコレクタ出力でもラインドライバ出力でも、ＪＰ４ジャンパーを切り替えることにより対応できます。相手出力側がオープンコレクタ出力のときは、ＪＰ４をUPPER側（出荷時の状態）にします。ラインドライバ出力のときは、ＪＰ４をLOWER側にし、nSTOP2信号をラインドライバ出力の片側に接続します。



エンコーダＺ相入力信号回路

［Ｚ相検出時の注意］

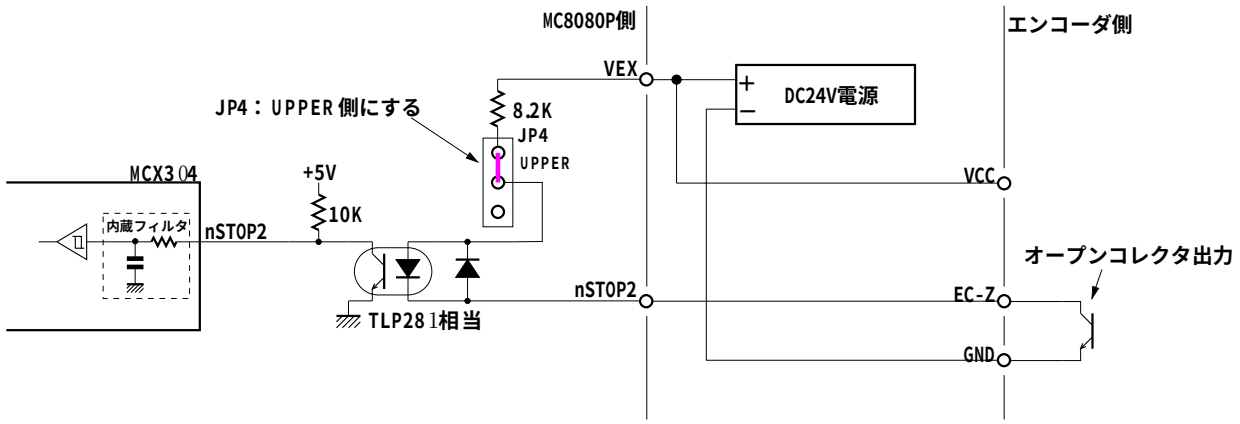
■ Z相検出のドライブ速度

本ボードは、電源オンされた初期時には、STOP2信号の内蔵フィルタの遅延時間が512μsecの設定になっています。さらにTLP281フォトカプラ(東芝)の遅延時間が100μsec程度ありますので、Ｚ相を検出するドライブ速度は、少なくともＺ相信号が１msec以上アクティブになるように設定する必要があります。ノイズ環境が良好の場合には、STOP2信号の内蔵フィルタを無効にすることによって、より高速で検出させることも可能です。

■ Z相検出の開始位置

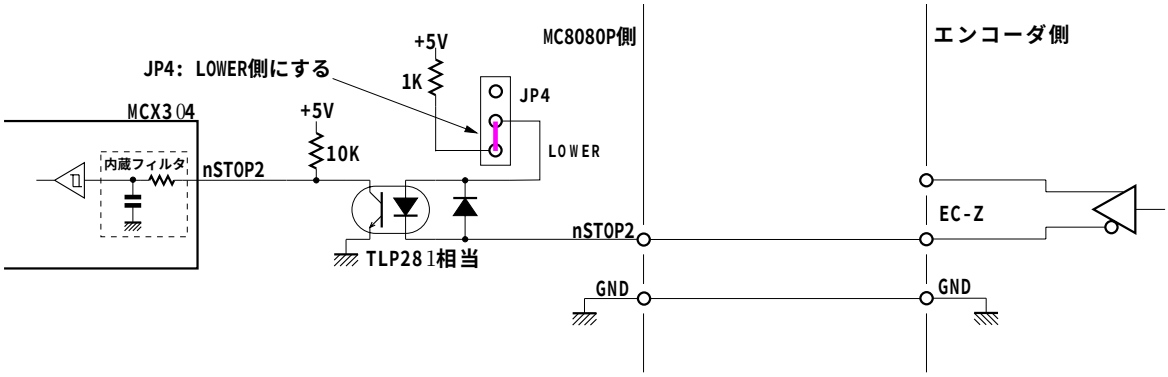
MCX304の自動原点出しでは、Z相（nSTOP2）信号が非アクティブ状態からアクティブに変化した時に検出ドライブを停止させます。従って、Z相検出の開始位置が安定してこの変化点から外れていなければなりません。通常は、この開始位置が、エンコーダのZ相位置の180°反対側になるように、機械的に調整します。

下図はnSTOP2入力信号とオープンコレクタ出力のエンコーダとの接続例です。オープンコレクタ出力がZ相検出時ONの場合には、MCX304の論理設定はWR1レジスタのD4(SP2-L)ビットを0（リセット時の状態）にセットします。



オープンコレクタのZ相出力との接続例

下図はnSTOP2入力信号とラインドライバ出力の片端子とエンコーダとの接続例です。出力がZ相検出時Lowレベルの場合には、MCX304の論理設定はWR1レジスタのD4(SP2-L)ビットを0（リセット時の状態）にセットします。



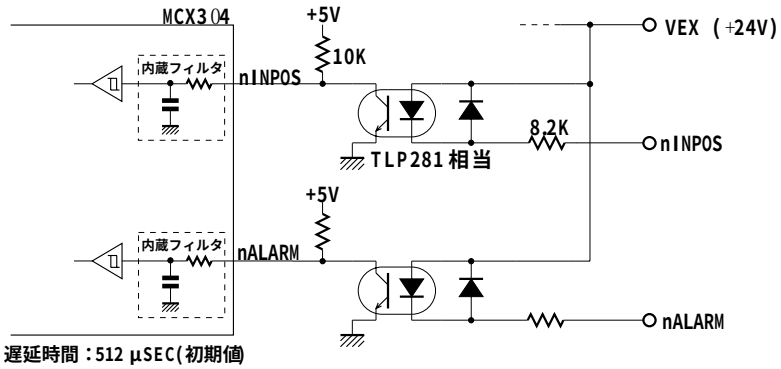
ラインドライバのZ相出力との接続例

3.8 サーボモータ用入力信号（nINPOS,nALARM）

nINPOS入力信号はサーボモータドライバのインポジション（位置決め完了）出力に対応する入力です。MCX304のモード設定で有効／無効、論理レベルを選択します。有効に設定すると、ドライブ終了後、この信号がアクティブになるのを待ってから主ステータスレジスタ(RR0)のn-DRVビットが0に戻ります。

nALARM入力信号はサーボモータドライバのアラーム出力に対応します。モード設定で有効／無効、論理レベルを選択します。有効に設定すると、nALARM入力信号を常に監視し、アクティブ状態の場合はステータスレジスタ2(nRR2)のALARMビットに1が立ちます。ドライブ中にアクティブレベルになると、ドライブは即停止されます。

リセット直後は、両信号とも無効になっています。nINPOS入力信号については、MCX304のモードレジスタ2(nWR2)のD15,14ビットを1,0にセットし、Lowレベルで有効にすると、本ボードのnINPOS信号端子より電流が流出する状態を待ってから、RR0レジスタのn-DRVビットが0に戻ります。また、nALARM入力信号については、nWR2レジスタのD13,12ビットを1,0にセットし、Lowレベルで有効にすると、本ボードのnALARM信号端子より電流が流出するときアラーム状態になります。詳細は、MCX304取扱説明書の2.6.5節、4.5節を参照してください。



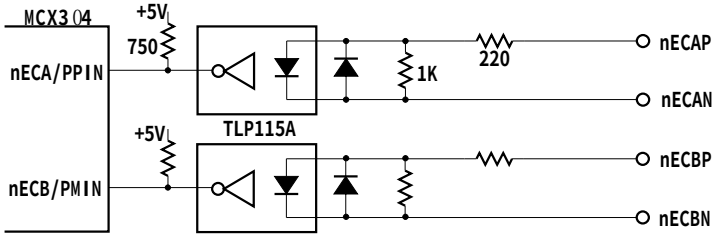
サーボモータ用入力信号回路

この信号を動作させるには、外部からDC24Vの電源供給が必要です。この信号の内蔵フィルタは、ボードが電源オンされた初期時には信号遅延時間512μの設定になります。システムのノイズ環境によって、この信号遅延時間は変更することが可能です。詳細はMCX304マニュアル2.6.9節および4.6節を参照してください。

また、サーボモータ用入力信号はインプットレジスタ１，２（RR4,5)で信号状態を常時読み出せますので汎用入力としても使用することができます。

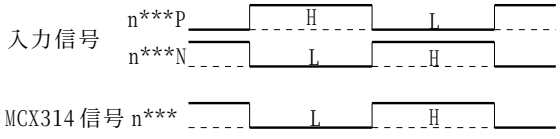
3.9 エンコーダＡ／Ｂ相入力信号（nECAP,nECAN,nECBP,nECBN）

nECAP/N,nECBP/N入力信号はエンコーダの２相出力信号、またはサーボモータドライバのエンコーダ２相出力信号を接続して、MCX304の実位置カウンタをカウントするための入力です。詳細は、MCX304取扱説明書の2.3.1節、2.6.3節、4.5節を参照してください。

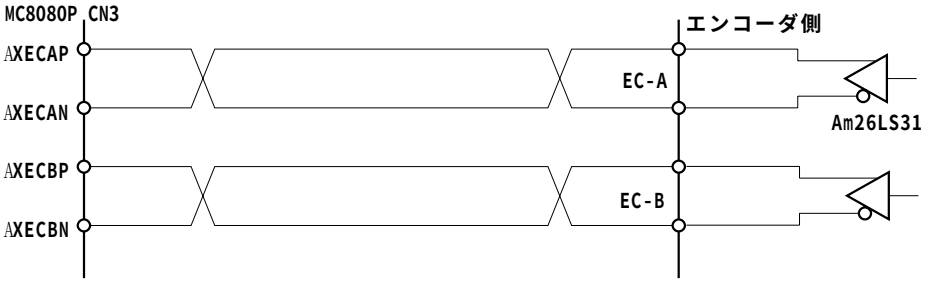


エンコーダＡ／Ｂ相入力信号回路

エンコーダA/B相入力信号回路は、上図に示すように、高速フォトカプラＩＣ TLP115A（東芝）を使用しています。各入力信号は差動出力のラインドライバとの直結が可能です。下図に示すように、n***P/N信号がH/Lのとき**MCX304**のn***信号がLowになり、L/HのときHiになります。入力から**MCX304**信号端子までの遅延時間は100nSEC以下ですので、２相パルス入力の場合であれば最高４MHzまでカウントできます。

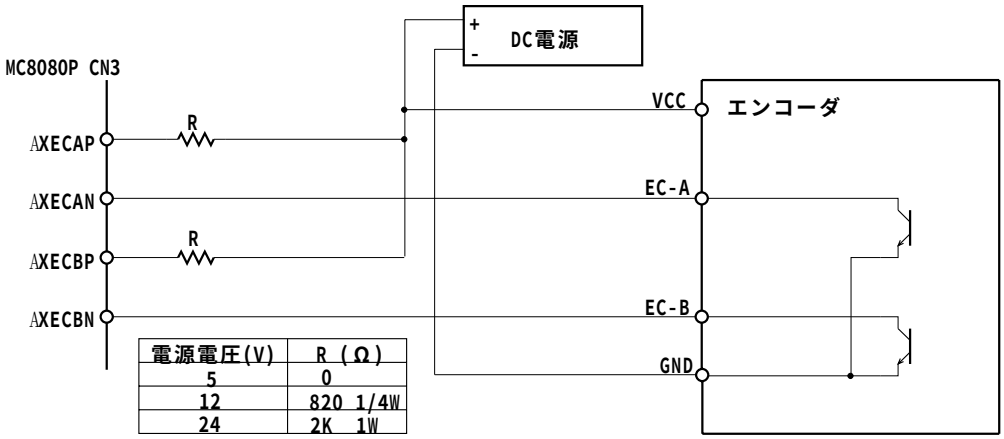


下図にエンコーダA/B相入力信号と差動出力のラインドライバとの接続例を示します。



差動出力のラインドライバとの接続例

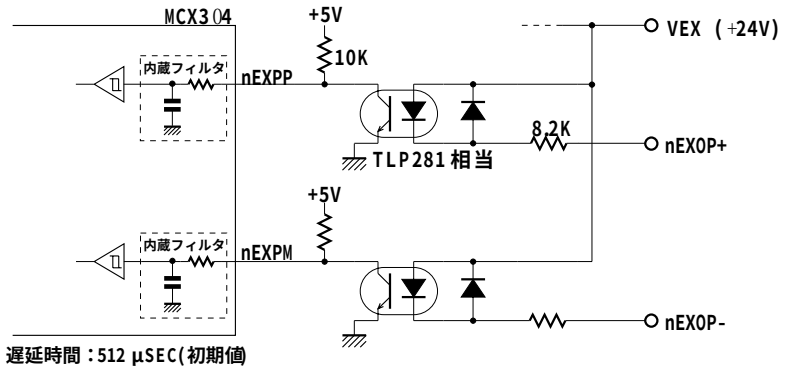
下図はエンコーダA/B相入力信号とオープンコレクタ出力のエンコーダとの接続例です。



オープンコレクタ出力との接続例

3.10 外部ドライブ操作信号（nEXOP+,nEXOP-）

外部から＋方向／－方向のドライブを起動する入力です。定量ドライブモードでは、入力信号のトリガ（立ち下がり）で指定ドライブパルスが出力されます。連続ドライブモードにすると、入力信号がLowレベルの間だけ、連続してドライブパルスを出し続けます。各軸のマニュアルジョグ送り等において、ＣＰＵの介在なしに軸送り動作が可能となります。外部ドライブ信号を有効にするには、MCX304のモード設定が必要です。詳細は、MCX304取扱説明書の2.6.1節、4.6節を参照してください。

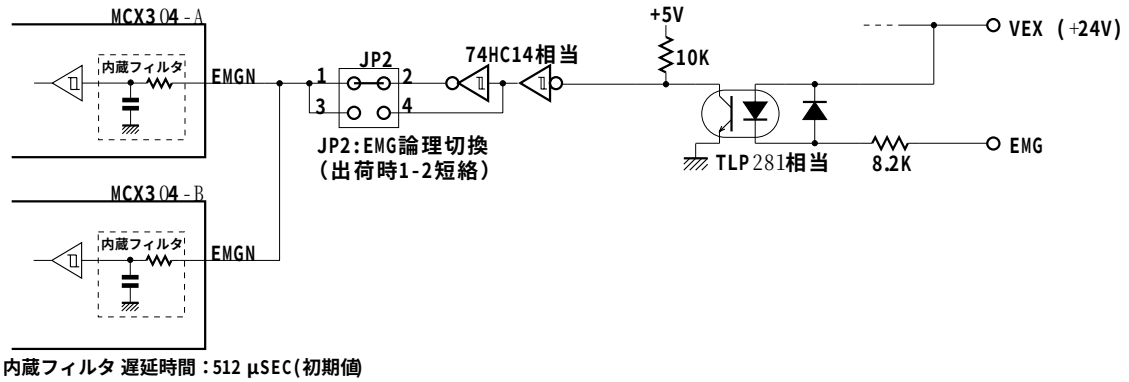


外部ドライブ操作信号回路

この信号を動作させるには、外部からDC24Vの電源供給が必要です。この信号の内蔵フィルタは、ボードが電源オンされた初期時には信号遅延時間512μの設定になります。システムのノイズ環境によって、この信号遅延時間は変更することが可能です。詳細はMCX304マニュアル2.6.9節および4.6節を参照してください。

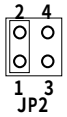
3.11 緊急停止入力信号（EMG）

緊急停止信号がアクティブレベルになると全軸のドライブパルス出力が停止します。アクティブレベルはボード内のＪＰ２ジャンパー端子で切り替えることができます。ドライブ中に緊急停止信号がアクティブになると、すべての軸のドライブは即停止し、主ステータスレジスタの全軸のエラービットに１が立ちます。MCX304の緊急停止については、MCX304取扱説明書の2.6.6節、4.12節を参照してください。



緊急停止入力信号回路

この信号を動作させるには、外部からDC24Vの電源供給が必要です。この信号の内蔵フィルタは、ボードが電源オンされた初期時には信号遅延時間512μの設定になります。システムのノイズ環境によって、この信号遅延時間は変更することが可能です。詳細はMCX304マニュアル2.6.9節および4.6節を参照してください。



左図はＪＰ２ジャンパーのピン配置を示しています。
1－2 間短絡：緊急停止信号(EMG)が外部電源のＧＮＤと短絡状態になるとアクティブレベルになります。
3－4 間短絡：緊急停止信号(EMG)がオープン状態になるとアクティブレベルになります。
出荷時は、1－2 間短絡になっています。

3.12 外部電源（VEX）

外部電源は、各軸のオーバランリミット入力信号（nLMT+,nLMT-）、減速停止／即停止入力信号（nSTOP0,nSTOP1,nSTOP2）、サーボモータ用入力信号（nINPOS,nALARM）、外部ドライブ操作信号（nEXOP+,nEXOP-）、および緊急停止入力信号（EMG）を動作させるために、外部から供給する電源です。ＤＣ２４Ｖの電源を供給してください。入力信号１点あたりの消費電流は、２．８ｍＡです。

3.13　ＰＣＩバスコネクタ

ピン	信号名	内　　容	信号出力時の駆動方法	入出力方向	
				マスタ	ターゲット
A1	TRST#	テストリセット		入力	入力
A2	+12V	電源		入力	入力
A3	TMS	テストモードセレクト		入力	入力
A4	TDI	テストデータ入力		入力	入力
A5	+5V	電源		入力	入力
A6	INTA#	割り込み要求Ａ	オープン･ドレイン	出力	出力
A7	INTC#	割り込み要求Ｃ	オープン･ドレイン	出力	出力
A8	+5V	電源		入力	入力
A9		予約			
A10	+5V	電源		入力	入力
A11		予約			
A12	GND	グランド		入力	入力
A13	GND	グランド		入力	入力
A14		予約			
A15	RST#	リセット		入力	入力
A16	+5V	電源		入力	入力
A17	GNT#	グラント	トライ･ステート	入力	
A18	GND	グランド		入力	入力
A19		予約			
A20	AD30	アドレスデータ30	トライ･ステート	入/出	入/出
A21	+3.3V	電源		入力	入力
A22	AD28	アドレスデータ28	トライ･ステート	入/出	入/出
A23	AD26	アドレスデータ26	トライ･ステート	入/出	入/出
A24	GND	グランド		入力	入力
A25	AD24	アドレスデータ24	トライ･ステート	入/出	入/出
A26	IDSEL	イニシャライゼーション デバイスセレクト		入力	入力
A27	+3.3V	電源		入力	入力
A28	AD22	アドレスデータ22	トライ･ステート	入/出	入/出
A29	AD20	アドレスデータ20	トライ･ステート	入/出	入/出
A30	GND	グランド		入力	入力
A31	AD18	アドレスデータ18	トライ･ステート	入/出	入/出
A32	AD16	アドレスデータ16	トライ･ステート	入/出	入/出
A33	+3.3V	電源		入力	入力
A34	FRAME#	サイクルフレーム	サスティンド･トライ･ステート	出力	入力
A35	GND	グランド		入力	入力
A36	TRDY#	ターゲットレディ	サスティンド･トライ･ステート	入力	出力
A37	GND	グランド		入力	入力
A38	STOP#	ストップ	サスティンド･トライ･ステート	入力	出力
A39	+3.3V	電源		入力	入力
A40	SDONE	スヌープ完了		入/出	入/出
A41	SB0#	スヌープバックオフ		入/出	入/出
A42	GND	グランド		入力	入力
A43	PAR	パリティ	トライ･ステート	入/出	入/出
A44	AD15	アドレスデータ15	トライ･ステート	入/出	入/出
A45	+3.3V	電源		入力	入力
A46	AD13	アドレスデータ13	トライ･ステート	入/出	入/出
A47	AD11	アドレスデータ11	トライ･ステート	入/出	入/出
A48	GND	グランド		入力	入力
A49	AD9	アドレスデータ9	トライ･ステート	入/出	入/出
A50		キーウェイ			
A51		キーウェイ			
A52	C/BE0#	バスコマンド･バイトイネーブル0	トライ･ステート	出力	入力
A53	+3.3V	電源		入力	入力
A54	AD6	アドレスデータ6	トライ･ステート	入/出	入/出
A55	AD4	アドレスデータ4	トライ･ステート	入/出	入/出
A56	GND	グランド		入力	入力
A57	AD2	アドレスデータ2	トライ･ステート	入/出	入/出
A58	AD0	アドレスデータ0	トライ･ステート	入/出	入/出
A59	+5V	電源		入力	入力
A60	REQ64#	64ビット転送要求	サスティンド･トライ･ステート	出力	入力
A61	+5V	電源		入力	入力
A62	+5V	電源		入力	入力

ピン	信号名	内容	信号出力時の駆動方法	入出力方向	
				マスタ	ターゲット
B1	-12V	電源		入力	入力
B2	TCK	テストクロック		入力	入力
B3	GND	グランド		入力	入力
B4	TDO	テストデータ出力		出力	出力
B5	+5V	電源		入力	入力
B6	+5V	電源		入力	入力
B7	INTB#	割り込み要求B	オープン･ドレイン	出力	出力
B8	INTD#	割り込み要求D	オープン･ドレイン	出力	出力
B9	PRSNT1#				
B10		予約			
B11	PRSNT2#				
B12	GND	グランド		入力	入力
B13	GND	グランド		入力	入力
B14		予約			
B15	GND	グランド		入力	入力
B16	CLK	クロック		入力	入力
B17	GND	グランド		入力	入力
B18	REQ#	リクエスト	トライ･ステート	出力	
B19	+5V	電源		入力	入力
B20	AD31	アドレスデータ31	トライ･ステート	入/出	入/出
B21	AD29	アドレスデータ29	トライ･ステート	入/出	入/出
B22	GND	グランド		入力	入力
B23	AD27	アドレスデータ27	トライ･ステート	入/出	入/出
B24	AD25	アドレスデータ25	トライ･ステート	入/出	入/出
B25	+3.3V	電源		入力	入力
B26	C/BE3#	バスコマンド･バイトイネーブル3	トライ･ステート	出力	入力
B27	AD23	アドレスデータ23	トライ･ステート	入/出	入/出
B28	GND	グランド		入力	入力
B29	AD21	アドレスデータ21	トライ･ステート	入/出	入/出
B30	AD19	アドレスデータ19	トライ･ステート	入/出	入/出
B31	+3.3V	電源		入力	入力
B32	AD17	アドレスデータ17	トライ･ステート	入/出	入/出
B33	C/BE2#	バスコマンド･バイトイネーブル2	トライ･ステート	出力	入力
B34	GND	グランド		入力	入力
B35	IRDY#	イニシエータ･レディ	サスティンド･トライ･ステート	出力	入力
B36	+3.3V	電源		入力	入力
B37	DEVSEL#	デバイス･セレクト	サスティンド･トライ･ステート	入力	出力
B38	GND	グランド		入力	入力
B39	LOCK#	ロック	サスティンド･トライ･ステート	出力	入力
B40	PERR#	パリティ･エラー	サスティンド･トライ･ステート	入/出	入力
B41	+3.3V	電源		入力	入力
B42	SERR#	システム･エラー	オープン･ドレイン	出力	出力
B43	+3.3V	電源		入力	入力
B44	C/BE1#	バスコマンド･バイトイネーブル1	トライ･ステート	出力	入力
B45	AD14	アドレスデータ14	トライ･ステート	入/出	入/出
B46	GND	グランド		入力	入力
B47	AD12	アドレスデータ12	トライ･ステート	入/出	入/出
B48	AD10	アドレスデータ10	トライ･ステート	入/出	入/出
B49	GND	グランド		入力	入力
B50		キーウェイ			
B51		キーウェイ			
B52	AD8	アドレスデータ8	トライ･ステート	入/出	入/出
B53	AD7	アドレスデータ7	トライ･ステート	入/出	入/出
B54	+3.3V	電源		入力	入力
B55	AD5	アドレスデータ5	トライ･ステート	入/出	入/出
B56	AD3	アドレスデータ3	トライ･ステート	入/出	入/出
B57	GND	グランド		入力	入力
B58	AD1	アドレスデータ1	トライ･ステート	入/出	入/出
B59	+5V	電源		入力	入力
B60	ACK64#	64ビット転送アクノリッジ	サスティンド･トライ･ステート	入力	出力
B61	+5V	電源		入力	入力
B62	+5V	電源		入力	入力

信号名に#の付いている信号は負論理を表します。

４．割り込み

本ボードでは、２個のMCX304で発生する割り込み信号を、ＰＣＩバスの4本の割り込み要求信号の内ＩＮＴＡ＃に接続しています。MCX304内で割り込みが発生すると本ボードの割り込み要求信号はHiレベルからLowレベルに変化します。割り込みを発生した軸のステータスレジスタ３（nRR3）を読み出すことにより、割り込み要求信号はLowからHiに戻ります。

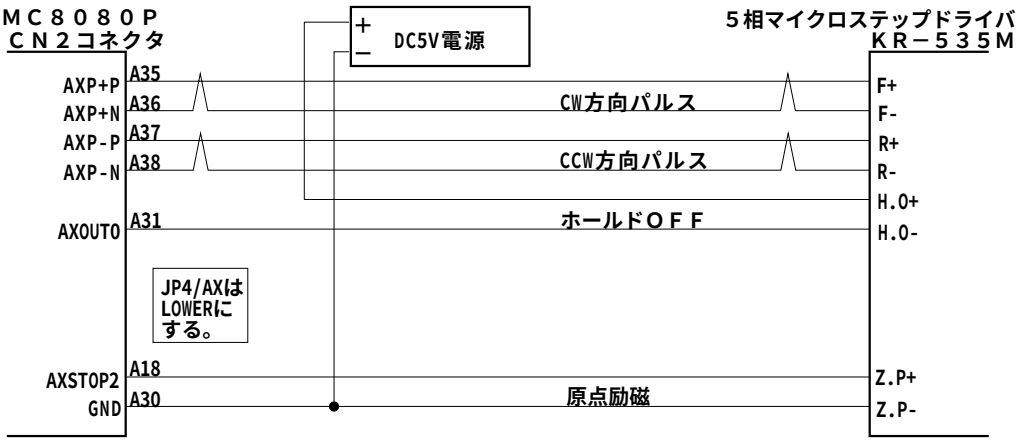
MCX304の割り込み発生機能については、MCX304取扱説明書2.5節（割り込み機能の説明）、4.4節（割り込み許可/禁止の設定）、4.13節（割り込み発生状態の読み出し）を参照してください。

【割り込みを使用する場合の注意】
本ボードからの割り込み信号出力は、PnP機能によってＩＲＱ番号が決定されます。またPnP機能とWindowsの機能によって同じ割り込み要求信号を他のデバイスと共用しますが、通常はWindowsによって管理されるため競合は発生しません。

５．モータドライバ接続例

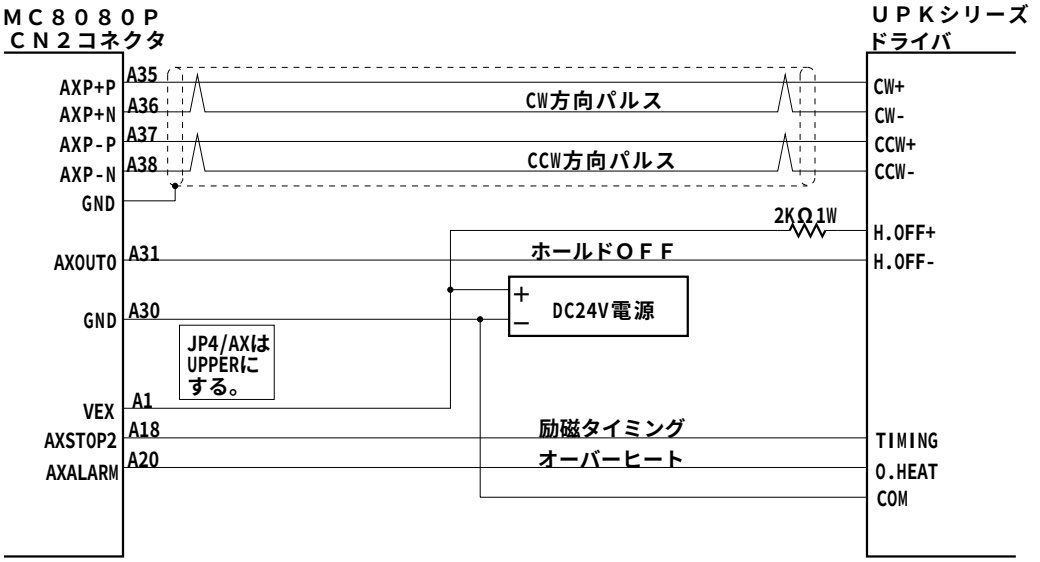
5.1 ステッピングモータドライバとの接続例

下図は、MC8080PのＡＸ軸とテクノドライブ製の５相マイクロステップドライバＫＲ５３５Ｍとの接続例を示しています。



注１：ホールドＯＦＦ、原点励磁信号は必要に応じて配線します。ホールドＯＦＦ信号は、MCX304のWR5レジスタのD0=1でXOUT0信号を有効にし、WR4レジスタのD0ビットに0,1を書き込むことによって制御します。原点励磁信号は、WR1レジスタのD4,5ビットをモード設定して、原点検出動作を行わせることができます。また、原点励磁信号は、RR4レジスタを通して直接信号レベルを読み出すことができます。

下図は、MC8080PのＡＸ軸とオリエンタルモータ製ＵＰＫシリーズのステッピングモータドライバとの接続例を示しています。

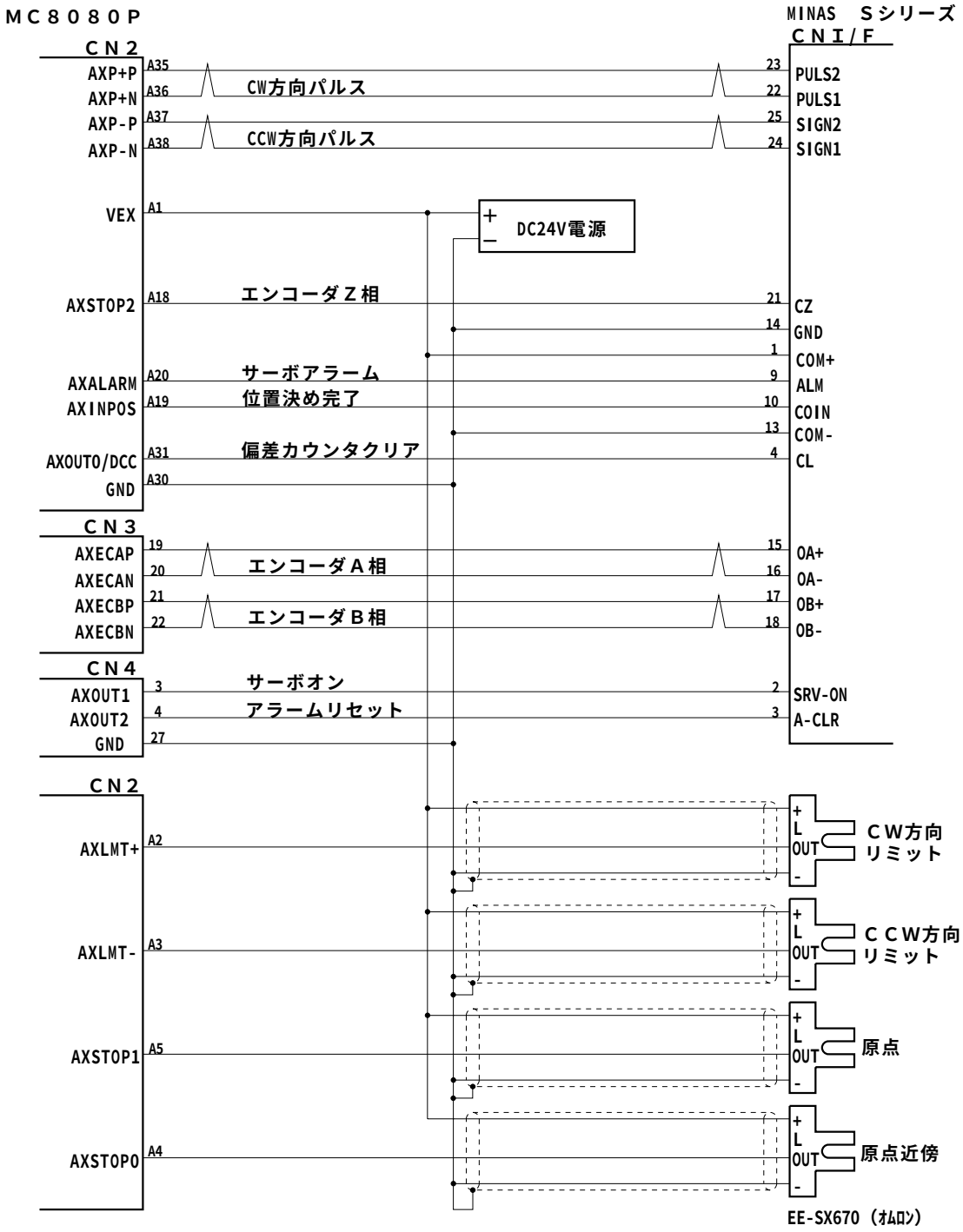


注１：ホールドＯＦＦ、励磁タイミング、オーバーヒート信号は必要に応じて配線します。ホールドＯＦＦ信号は、MCX304のWR5レジスタのD0=1でXOUT0信号を有効にし、WR4レジスタのD0ビットに0,1を書き込むことによって制御します。原点励磁信号は、WR1レジスタのD4,5ビットをモード設定して、原点検出動作を行わせることができます。オーバーヒート信号は、WR2レジスタのD12,13ビットをモード設定してアラーム機能を働かせることができます。また、励磁タイミング、オーバーヒート信号は、RR4レジスタを通して直接信号レベルを読み出すことができます。

注２：強いノイズ環境下、あるいはドライバまでの距離が長い場合は、上図のようにツイストペアシールド線を推奨します。

5.2 ＡＣサーボモータドライバとの接続例

下図は、MC8080PのＡＸ軸とMINAS Ｓシリーズ ＡＣサーボモータドライバとの接続例を示しています。



注１：ドライバの制御モード設定は位置制御モードに、指令パルス形態はＣＷ／ＣＣＷパルスモードにパラメータセットします。指令パルス形態をパルス／符号モードにすると、t6時間が不足しますので適当ではありません。

注２：エンコーダＡ／Ｂ相信号はMCX304内で実位置カウンタをカウントさせる場合に接続します。ＣＰＵ側で実位置データを必要としなければ接続する必要はありません。その他の信号も必要に応じて接続します。

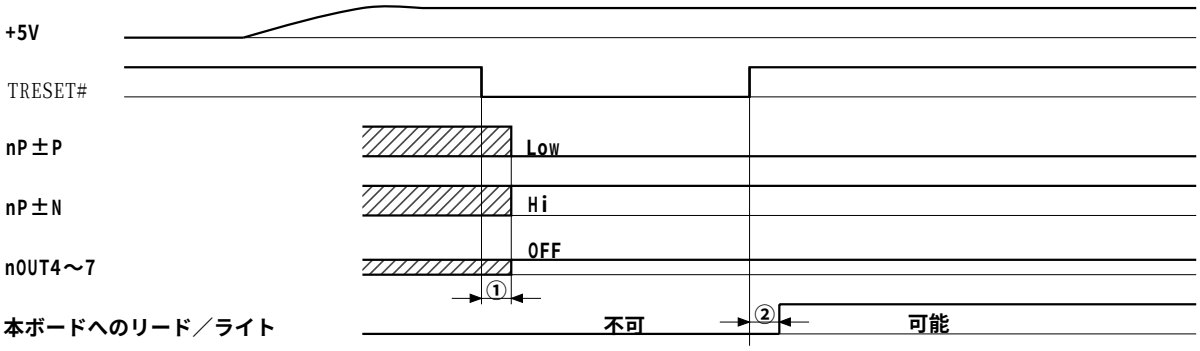
注３：エンコーダＺ相は、この例ではドライバ側のオープンコレクタ出力を使用していますのでは、ＪＰ４をUPPER側（出荷時の状態）にします。

注４：原点信号は、この例では原点近傍信号と原点信号をそれぞれ配線していますのでＪＰ３はLEFT側（出荷時の状態）にします。

注５：強いノイズ環境下、あるいはドライバまでの距離が長い場合は、上図のようにツイストペアシールド線を推奨します。

６．入出力信号タイミング

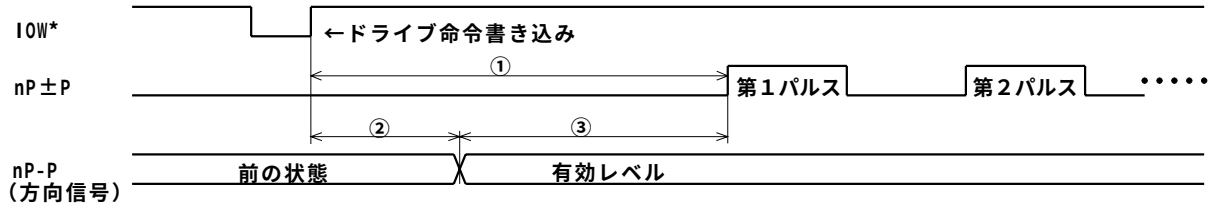
6.1 リセット時



① ドライブパルス出力信号（nP±P,nP±N）、および汎用出力信号（nOUT4～7）は、APIC21(アドテック社)のターゲットリセット信号（TRESET#）の↓から最大２５０ｎSEC以内に確定します。

② 本ボードへの書き込み／読み出しは、ターゲットリセット信号（TRESET#）の↑から５００ｎSEC後から可能になります。

6.2 独立ドライブ開始時

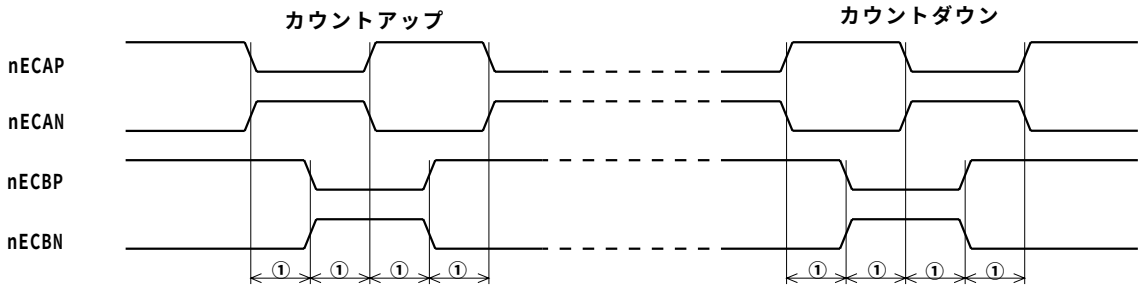


① ドライブ命令が書き込まれてから最大６５０ｎSEC以内に第１ドライブパルスが出力されます。

②③ ドライブ出力パルス方式を１パルス方式にしたときは、ドライブ命令書き込み後最大２７５ｎSEC以内に方向信号（nP-P）が有効レベルになり、方向信号が有効レベルになってから３７５ｎSEC後に第１ドライブパルスが出力されます。

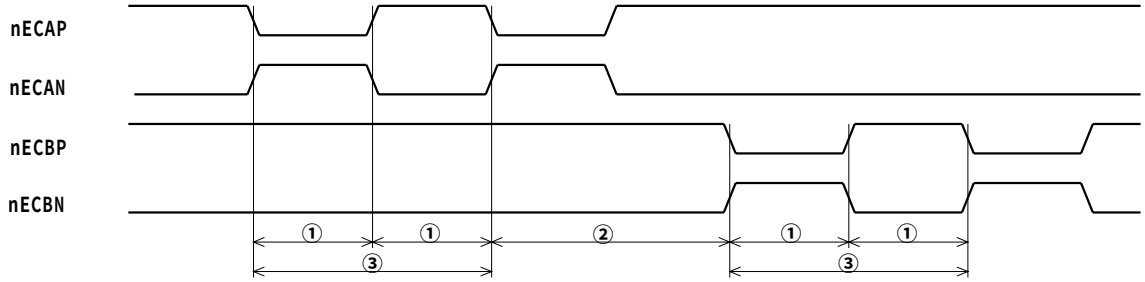
6.3 入力パルスタイミング

■ エンコーダ２相パルス入力時



①（EC-A,EC-B位相差時間）　：最小２００ｎSEC

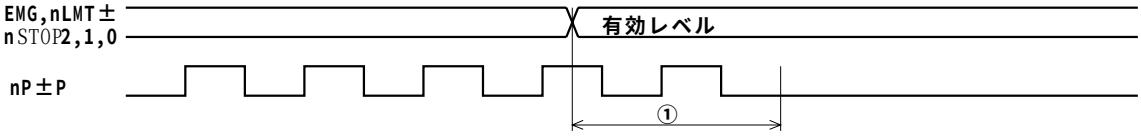
■ アップ/ダウンパルス入力時



①（UP/DOWNパルス幅）　：最小１３０ｎSEC　　②（UP⇔DOWNパルス間）：最小２６０ｎSEC
③（UP/DOWNパルス周期）：最小２６０ｎSEC

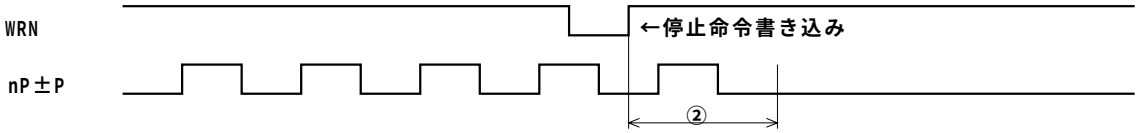
6.4　即停止タイミング

■ 外部信号による即停止



① ドライブ途中で外部停止信号が有効レベルになると、フォトカプラ遅延時間（最大100μsec）＋ＩＣ内蔵積分フィルタの遅延時間（初期値512μsec）＋１ドライブパルス後に停止します。

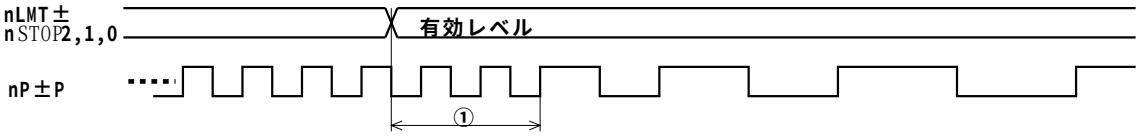
■ 命令による即停止



② ドライブ途中で停止命令が書き込まれると、最大１ドライブパルス後に停止します。

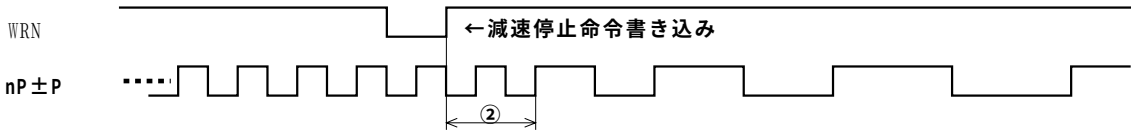
6.5　減速停止タイミング

■ 外部信号による減速停止



① ドライブ途中で外部減速停止信号が有効レベルになると、フォトカプラ遅延時間（最大100μsec）＋ＩＣ内蔵積分フィルタの遅延時間（初期値512μsec）＋２ドライブパルス後に減速を開始します。

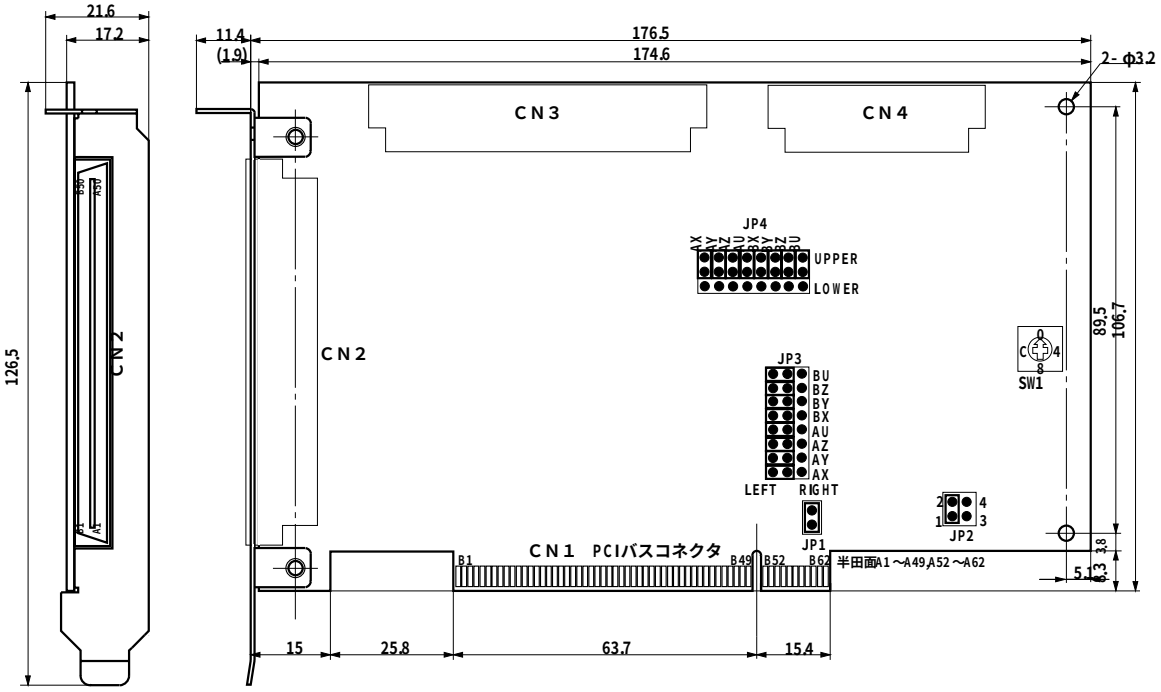
■ 命令による減速速止



② ドライブ途中で減速停止命令が書き込まれると、最大２ドライブパルス後に減速を開始します。

7．基板外形

単位：mm



JP1: 1-2短絡（出荷時の状態）のままにしておいてください。

JP2: 緊急停止信号(EMG)のアクティブ論理を選択します。
1-2短絡(出荷時)：信号をGNDと短絡するとアクティブになります。
3-4短絡： 信号オープンでアクティブになります。

JP3: 原点出し信号を選択します。3.6節参照。
LEFT(出荷時):STOP0を原点近傍、STOP1を原点信号として使用します。
RIGHT: STOP0だけを使用して、高速原点サーチ→低速追い込みを行います。

JP4: STOP2(エンコーダＺ相)入力回路を選択します。3.7節参照。
UPPER(出荷時)：相手側がオープンコレクタ出力。
LOWER： 相手側がラインドライバ出力。

SW1: ボードを複数枚使用するときのボード番号を設定するロータリスイッチです。
０～９の値を設定することができます。（出荷時：０）

８．インストール

この章では、本ボードのパソコンへの組込みとドライバのインストール及びアプリケーション開発のためのライブラリの使用方法について説明します。

8.1 パソコンへの本ボードの組込み

① パソコン本体の電源がＯＦＦであることを確認してから、外装カバー､スロットカバー等を外します。
② 空いている拡張スロットへ本製品を差し込みます｡基板のエッジコネクタをパソコンのＰＣＩバスコネクタに正しく挿入してください。
③ 取付金具をネジ止めしてください。この時キチンとねじを締めないと後で抜け落ちたりするなどして､ショートや故障､誤動作の原因となります。
④ パソコン本体の外装カバーを元通りに取り付けます。

注意：パソコンへの取り付け作業は必ずパソコンの電源を切断してから行ってください。さもないと回路素子を破壊する原因となります。

8.2 デバイスドライバのインストール

本デバイスドライバは　8.6.1　動作環境　で対応する全ての言語に共通です。
本デバイスドライバは本ボードを同時に１０枚まで認識します。

＜本ボードを複数枚使用する場合＞

本ボードを１つのシステム（ＰＣ）で複数枚ご使用になる時は、ＰＣＩバス上でそれぞれのボードを個別に認識させる為に、２枚以降のボードはボード番号をボード上のロータリスイッチで設定しなければなりません。ロータリスイッチ（ＳＷ１）の位置は、７章“基板外形”を参照してください。

8.2.1 Windows98/Me

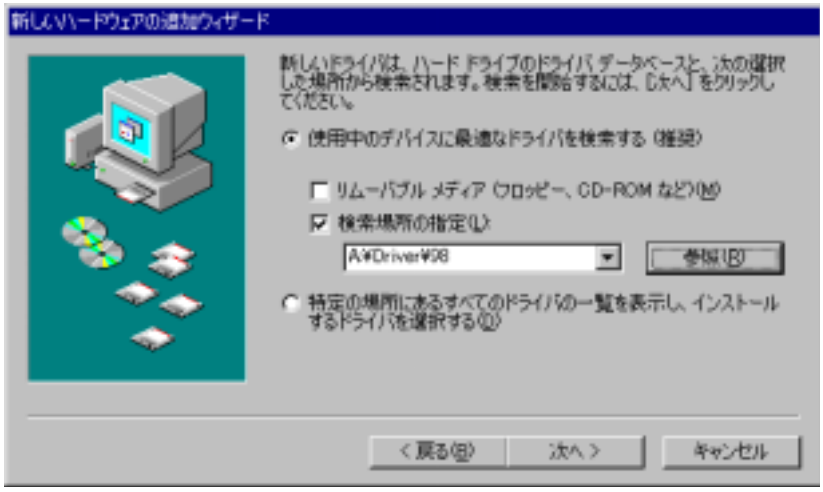
今回はWindows98を例にとって説明しますが、基本的にWindows Meも同様に操作してください。

① まず8.1によって本ボードが確実にパソコンに組み込まれているか確認してください。

② パソコン本体の電源をＯＮし、Windows98/Meを起動します。

③ デバイスドライバの自動検出を促すメッセージが出力されますので、「ドライバの場所を指定する（詳しい知識のある方向け）」にチェックをして[次へ]をクリックします。

④検索場所の指定にチェックしてテキストボックスへ A:\Driver\98 と入力して[次へ]をクリックします。

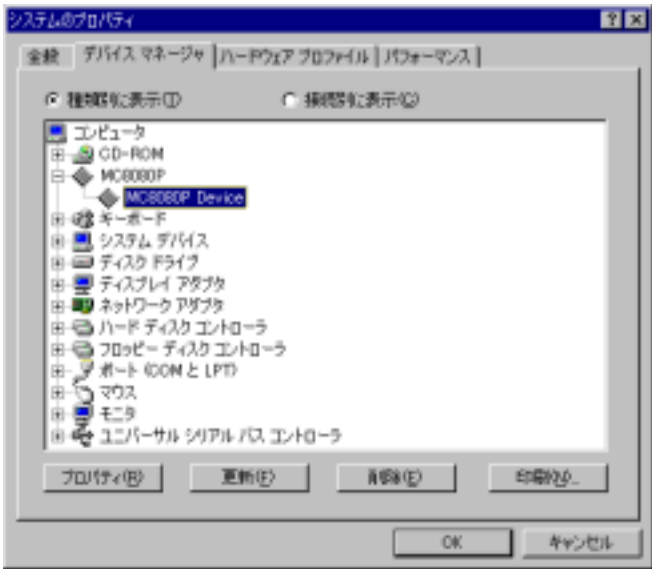


⑤ [MC8080P Device]が検出できたら[次へ]をクリックします。

⑥　デバイスドライバのコピーが完了すると、メッセージが出ますので[完了]をクリックします。

以上でデバイスドライバのインストールは完了です。デバイスドライバのインストール完了後はパソコンの起動時に手順③のようにハードウェアウィザードが起動することはありません。もしハードウェアウィザードが起動するような場合はインストールが正常に終了していない可能性がありますので、その場合は8.3の手順に従って一度本ボードを取り外した後に8.1の手順から再度インストールをやり直してください。

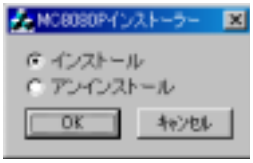
インストールを完了したらリソース(Ｉ／Ｏアドレス、割り込みレベル)の設定、競合の有無を[コントロールパネル]-[システム]-[デバイスマネージャ]タブで確認してくだい。



8.2.2 WindowsNT

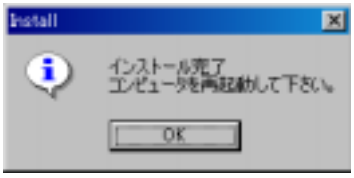
デバイスドライバーのインストールは必ずアドミニストレーター権限をもったユーザーログインで行ってください。アドミニストレーター権限以外でインストールをした場合正常にインストールされません。

① まず8.1によって本ボードが確実にパソコンに組み込まれているか確認してください。
② パソコン本体の電源をＯＮし、WindowsNTを起動します。
③ アドミニストレーター権限を持ったユーザーでログインしてください。
④ 次に Ａ：\Driver\NT フォルダにある install.exe を実行してください。[スタート]-[ファイル名を指定して実行]か、またはエクスプローラーから直接実行できます。
⑤ 下のように表示されますので「インストール」を選択して [OK]を押して次に進んでください。



⑥ もしこのときにWindowsNTのドライバフォルダが標準のインストール(通常は\WinNT\System32\Driver)でない場合は各自でMC8080P.SYS ファイルをシステムドライバフォルダへコピーしてください。

⑦ 次に下のように表示されたら [OK] を押してください。これでデバイスドライバーのインストールは終了です。



⑧ 最後にNTを再起動して下さい。

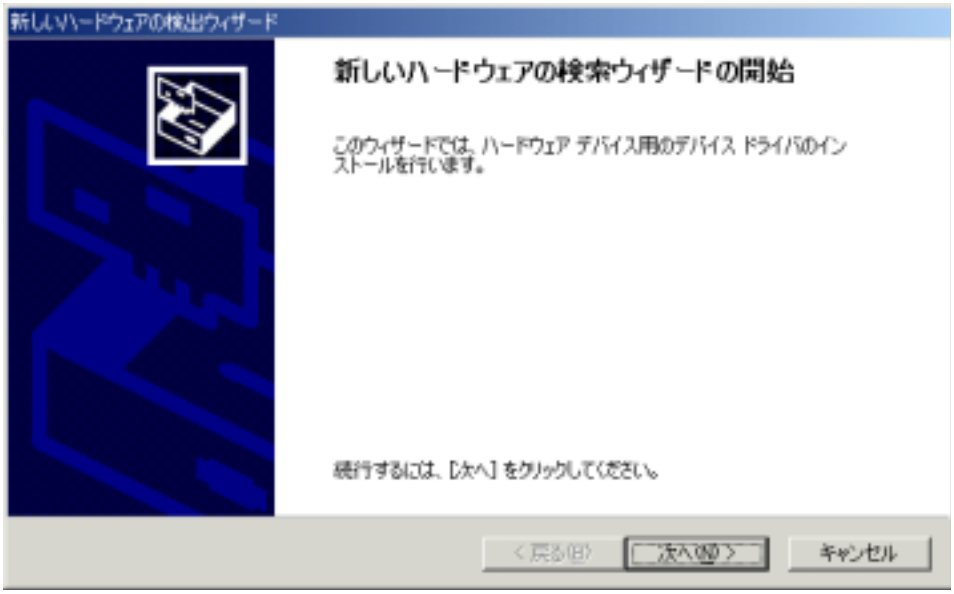
⑨再起動したら、NTの場合は[コントロールパネル]-[デバイス]でMC8080Pが開始状態になっていることを確認してください。なっていない場合はインストールが正常に終了していない可能性がありますので、その場合は8.3の手順に従って一度本ボードを取り外した後に8.1の手順から再度インストールをやり直してください。

8.2.3 Windows2000/XP

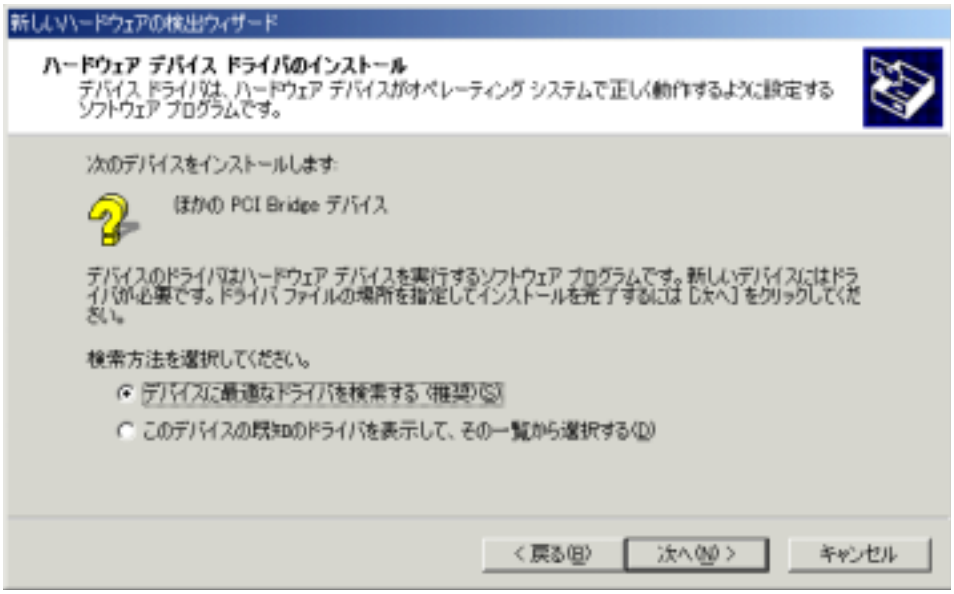
デバイスドライバーのインストールは必ずアドミニストレーター権限をもったユーザーログインで行ってください。アドミニストレーター権限以外でインストールをした場合正常にインストールされません。

Windows2000を例にとって説明しますが、基本的にWindowsXPも同様に操作してください。

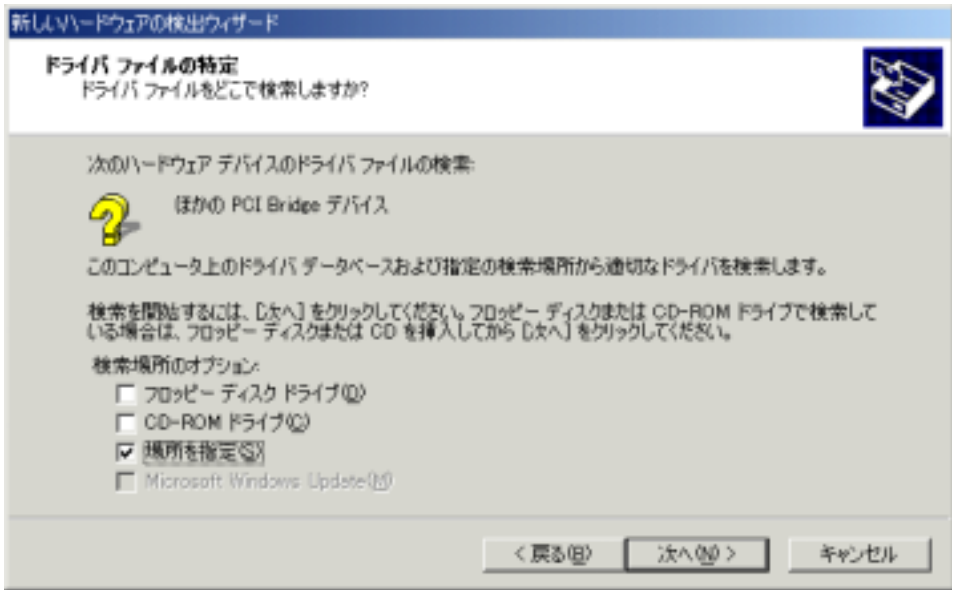
① まず8.1によって本ボードが確実にパソコンに組み込まれているか確認してください。
② パソコン本体の電源をＯＮし、Windows2000/XPを起動します。
⑨ アドミニストレーター権限を持ったユーザーでログインしてください。
④ すぐに「新しいハードウェアが発見されました」と表示されデバイスドライバのインストールが開始されます。
⑤ すぐに「新しいハードウェアの検出ウィザード」が表示され「新しいハードウェアの検索ウィザードの開始」が出ますので[次へ]をクリックします。



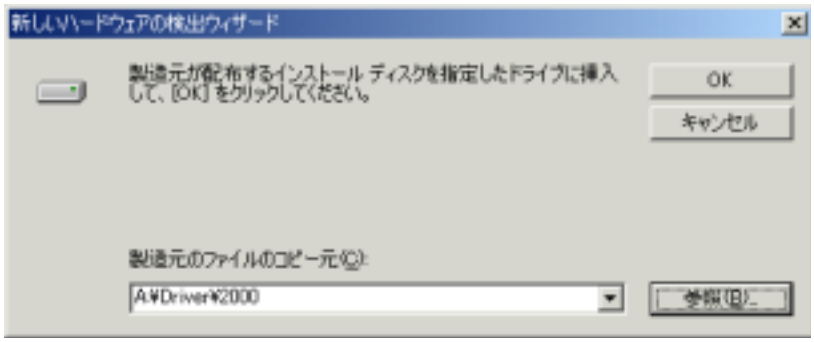
⑥ 「ハードウェアデバイスドライバのインストール」が表示され、デバイスドライバの自動検出を促す画面が出ますので、「デバイスに最適なドライバを検出する(推奨)」にチェックして[次へ]をクリックします。



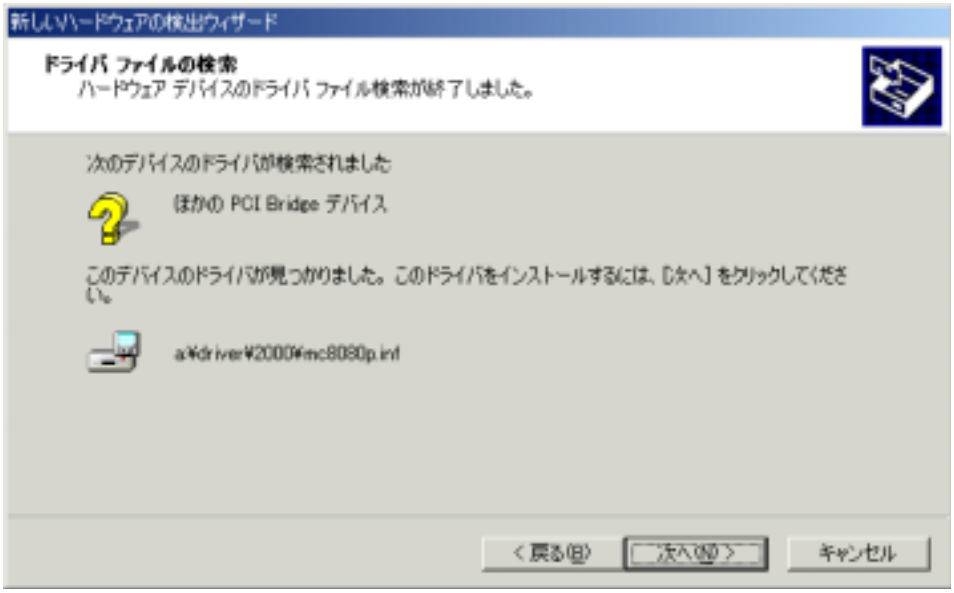
⑦　検索場所の指定が出ますので[場所を指定]にチェックして[次へ]　　をクリックします。



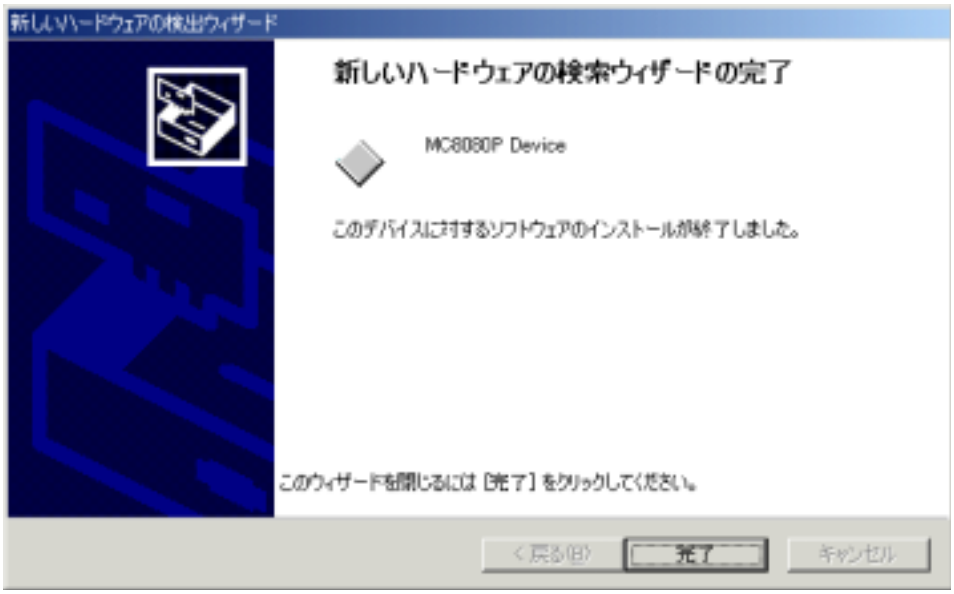
⑧　下のような画面が出ますので A:\Driver\2000 (XPの場合はA:\Driver\XP)と入力してＯＫを押してください。



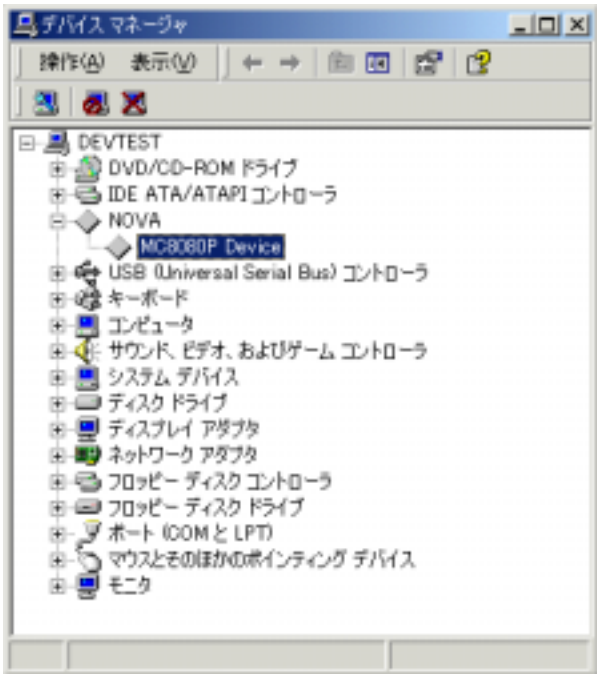
⑨　デバイスドライバの情報ファイルが発見されると確認の画面が出ますので、フロッピーの場所が正しいことを確認して、[次へ]をクリックします。

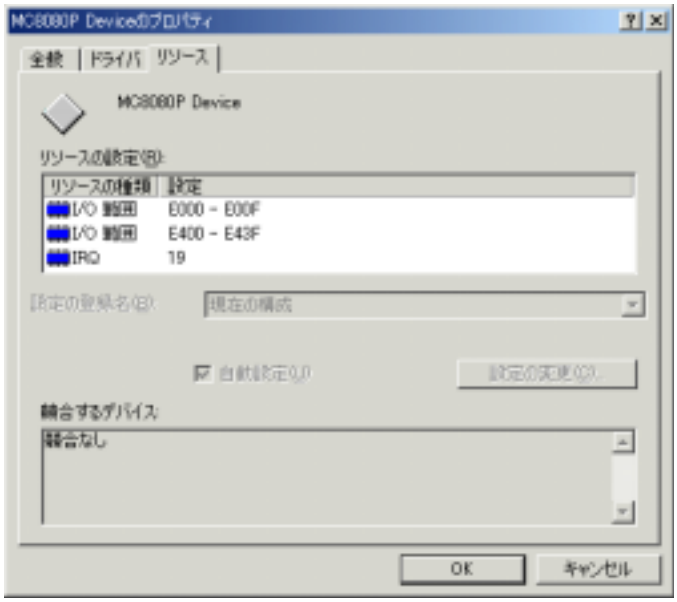


⑩　デバイスドライバのインストールが正常に完了すると、メッセージが出ますので、[完了]をクリックします。



以上でデバイスドライバのインストールは完了です。デバイスドライバのインストール完了後はパソコンの起動時に手順④のようにハードウェアウィザードが起動することはありません。もしハードウェアウィザードが起動するような場合はインストールが正常に終了していない可能性がありますので、その場合は8.3の手順に従って一度本ボードを取り外した後に8.1の手順から再度インストールをやり直してください。
インストールを完了したらリソース(Ｉ／Ｏアドレス、割り込みレベル)の設定、競合の有無を[コントロールパネル]-[システム]-[ハードウェア]-[デバイスマネージャ]の[プロパティ]で確認してください。

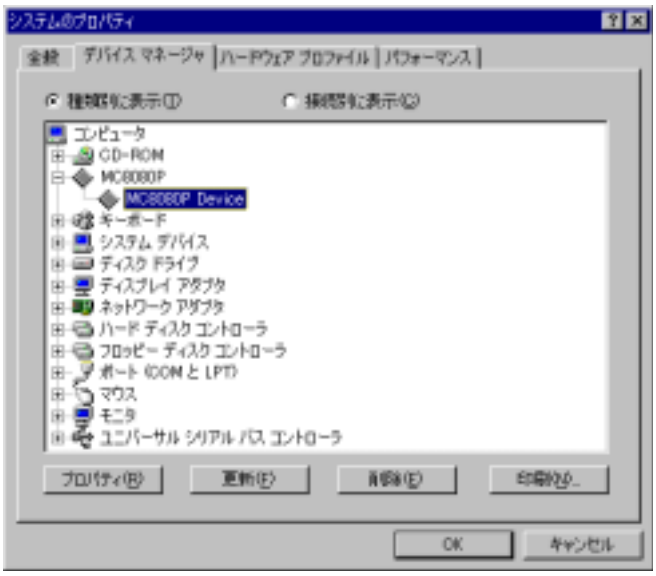




8.3 取り外し

8.3.1 Windows98/Me

① まず[コントロールパネル]-[システム]-[デバイスマネージャ]タブで本製品のデバイスドライバを削除してください。
② パソコン本体の電源がＯＦＦであることを確認してから、外装カバー、スロットカバー等を外します。
③ ボードを止めているビスを外します。
④ 本ボードを指先でつまんで軽く左右にゆするようにしながら引き出します。
⑤ パソコン本体の電源をＯＮし、Windows98/Meを起動します。
⑥ [コントロールパネル]-[システム]-[デバイスマネージャ]タブで本製品が削除されていることを確認してください。



8.3.2 WindowsNT

デバイスドライバーのアンインストールは必ずアドミニストレーター権限をもったユーザーログインで行ってください。アドミニストレーター権限以外でアンインストールをした場合正常にアンインストールされません。

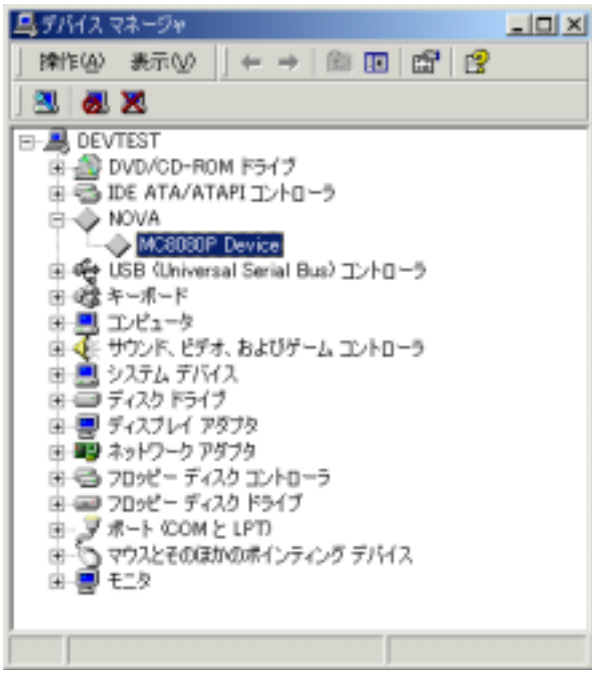
① まず、\Driver\NT フォルダにある install.exe を実行してください。[スタート]-[ファイル名を指定して実行]か、またはエクスプローラーから直接実行できます。
② 下のように表示されますので「アンインストール」を選択して[OK]を押して次に進んでください。

③ 次に下のように表示されたら[OK]を押してアンインストーラーを終了してください。

④ パソコン本体の電源がＯＦＦであることを確認してから、外装カバー、スロットカバー等を外します。
⑤ ボードを止めているビスを外します。
⑥ 本ボードを指先でつまんで軽く左右にゆするようにしながら引き出します。
⑦ パソコン本体の電源をＯＮし、WindowsNTを起動します。
⑧ [コントロールパネル]-[デバイス]でMC8080Pが削除されていることを確認してください。

8.3.3 Windows2000/XP

① まず[コントロールパネル]-[システム]-[ハードウェア]-[デバイスマネージャ]で本製品のデバイスドライバを削除してください。
② パソコン本体の電源がＯＦＦであることを確認してから、外装カバー、スロットカバー等を外します。
③ ボードを止めているビスを外します。
④ 本ボードを指先でつまんで軽く左右にゆするようにしながら引き出します。
⑤ パソコン本体の電源をＯＮし、Windows2000/XPを起動します。
⑥ [コントロールパネル]-[システム]-[デバイスマネージャ]タブで本製品が削除されていることを確認してください。



8.4 外部機器との接続時の注意

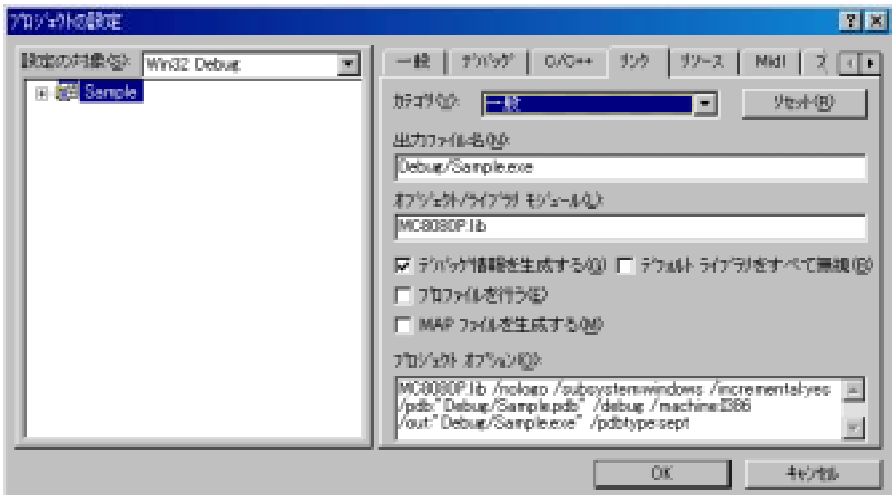
本製品を外部機器と接続して動作させる場合には以下の点に注意してください。
□出力信号は出力信号同士や他の機器の出力信号と接続しないで下さい。故障の原因となります。
□出力信号を外部電源と短絡すると故障の原因となります。
□誤動作時の安全確保のため必ずリミットスイッチ入力を本ボードと外部機器間で接続してください。
□モータを駆動する前に必ず配線などに間違いがないか十分に確認してください。
モータと装置を切り離した状態でモータの回転、リミットスイッチの動作を必ず確認してからご使用ください。
□サージ電圧の入力は本ボードの故障の原因となる場合があります。
□入／出力信号の接続
外部電源や入力／出力信号の接続において、極性を逆にしたり、定格範囲を越えた電圧／電流を印加すると、回路素子を破壊したり、動作の信頼性を低下させる原因となります。十分配線を確認の上、接続してください。
□Ｉ／Ｏケーブルの処置
付属のＣＮ２用ケーブルは１．２ｍの長さですが、A30～A50、およびB30～B50の信号はパソコン内部と同じ回路系の入出力信号線ですので、周囲からの電磁誘導ノイズを極力受けないよう十分配慮し、必要最小の長さでご使用ください。

8.5 ライブラリのセットアップ

\Lib98と\LibNT、\Lib2000XPに入っているファイルはファイル名は共通ですが､それぞれWindows98/Me、WindowsNT、Windous2000/XP専用です｡くれぐれも間違えないで下さい。

8.5.1 Microsoft Visual C++ (以下 VC++)でアプリケーションを開発する場合

① \Lib\VC6フォルダに入っている２つのファイルMC8080P.libとMC8080P_DLL.hを開発するアプリケーションのフォルダにコピーしてください。
② VC++の総合開発環境にてMC8080P_DLL.h をご使用のプロジェクトに追加登録してください。
③ 「プロジェクト->設定」で“ リンク ”タブを選択し“ オブジェクト／ライブラリモジュール ”にMC8080P.libを指定して下さい。



注意：この2つのファイルは VC++ 6.0 以降対応です。

8.5.2 Microsoft Visual Basic (以下 VB)でアプリケーションを開発する場合

特別なライブラリやヘッダファイルは必要としません｡以下を標準モジュールの中で定義して下さい｡(通常は拡張子が.BASのファイルです)
まれにデバッグ中など MC8080P.dll にリンクできないときは MC8080P.dll をカレントフォルダにコピーして使用してください

VB の中で、本ボードに関する割り込みを使用することはできません。

8.6 ソフトウェアの仕様

8.6.1 動作環境

対応ＯＳ　Windows98 WindowsNT Windows2000 WindowsXP

対応言語　Microsoft Visual C++ 6.0 以降
Microsoft Visual Basic 6.0 以降

8.6.2 プログラム構成

プログラム構成 種類	フォルダ	ファイル名	説　　明
デバイスドライバ	Driver\98	MC8080P.SYS	Windows98/Me用デバイスドライバ本体
		MC8080P.DLL	ダイナミックリンクライブラリ VC++,VB共通
	Driver\NT	MC8080P.SYS	WindowsNT用デバイスドライバ本体
		MC8080P.DLL	ダイナミックリンクライブラリ VC++,VB共通
	Driver\2000	MC8080P.SYS	Windows2000用デバイスドライバ本体
		MC8080P.DLL	ダイナミックリンクライブラリ VC++,VB共通
	Driver\XP	MC8080P.SYS	WindowsXP用デバイスドライバ本体
		MC8080P.DLL	ダイナミックリンクライブラリ VC++,VB共通
ライブラリ	Lib\VC6	MC8080P.LIB	MC8080P.DLLを使用するためのライブラリ VC++専用
		MC8080P.H	MC8080P.DLLを使用するためのヘッダ定義ファイル VC++専用
	Lib\VB6	MC8080P.BAS	MC8080P.DLLを使用するためのデクリア定義ファイル VB専用
VC++サンプルプログラム	Sample \MC8080P SMP VC6	MC8080P_SMP_VC6.cpp	アプリケーション用クラスの定義を行います。
		MC8080P_SMP_VC6Dlg.cpp	メインダイアログクラス定義 インプリメンテーション ファイル
		MC8080P.CPP	MC8080P制御関数定義ファイル
VBサンプルプログラム	Sample \MC8080P SMP VB6	MainWnd.frm	メインウインドウ定義ファイル
		MC8080P_DLL.BAS	MC8080P.DLL関数を使う為のデクリアファイル
		MC8080P.BAS	MC8080P制御関数定義ファイル

備考：VC++のMFC AppWizerdが自動的に作成するファイルに関しては説明を省略いたします、又ドライバセットアップ時にのみ使用されるファイルも省略致します。

8.6.3 ＡＰＩ

MC8080P.SYS MC8080P.DLL がアプリケーションに提供するＡＰＩ

■ＶＣ++ でプログラミングする場合

関数名	機能　及び　内容
OpenMC8080P	MC8080Pの使用を開始する。 入力パラメータ：int No // カード番号（カード上のロータリースイッチの設定値） 戻り値　　：BOOL オープンに成功するとTRUE、失敗するとFALSE 使用例　　： 　status = OpenMC8080P(0);　　// カード番号０をオープン
CloseMC8080P	MC8080Pの使用を終了する。 入力パラメータ：int No // カード番号（カード上のロータリースイッチの設定値） 戻り値　　：BOOL クローズに成功するとTRUE、失敗するとFALSE 使用例： 　status = CloseMC8080P(0);　　// カード番号０をクローズ
CloseAllMC8080P	全てのMC8080Pの使用を終了する。 入力パラメータ：なし 戻り値　　：なし 使用例： 　status = CloseAllMC8080();　　// 全てのカードをクローズ
OutpMC8080P	出力ポートに１ワード書き込む。 入力パラメータ：int No // カード番号（カード上のロータリースイッチの設定値） long Adr // 書き込むアドレス long Dat // 書き込むデータ 戻り値：　　なし 使用例： 　OutpMC8080P(0, MCX304A_R0, 0x8000);　// ボードのソフトリセット
InpMC8080P	入力ポートから１ワード読み出す。 入力パラメータ：int No // カード番号（カード上のロータリースイッチの設定値） long Adr // 読み出すアドレス 戻り値　　：long 入力ポートから読み込んだ１ワード 使用例： 　data = InpMC8080P(0, RR0);　　// リードレジスタ RR0 の読み出し 　リードレジスタ番号の指定はヘッダ定義ファイルに宣言されている RR0～RR7 を使用してください。
SetEventMC8080P	割込みを処理する関数をセットする。 入力パラメータ：int　No　//　カード番号　（カード上のロータリースイッチの設定値） Long　Adr　//　関数のアドレス。 戻り値　　：なし 使用例： 　OutpMC8080P(CARD_NO,　MCX304A_WR0,　0xFF);　　//　全軸指定 　OutpMC8080P(CARD_NO,　MCX304A_WR1,　0x8000);　//　停止時割込み発生 　SetEventMC8080P(CARD_NO,(LPTHREAD_START_ROUTINE)MC8080P_EventProc0);

ResetEventMC8080P	割込みを処理する関数を開放する。
	
	入力パラメータ：int No　// カード番号（カード上のロータリースイッチの設定値）
	戻り値　　：なし
	使用例：
	　　OutpMC8080P(CARD_NO, MCX304A_WR0, 0xff);　//全軸指定
	　　OutpMC8080P(CARD_NO, MCX304A_WR1, 0x0000); //割込み禁止
	　　ResetEventMC8080P(CARD_NO);

関数名	機能　及び　内容
ReadEventMC8080P	割込み発生直後の各軸ＲＲ３の値を取得 ＲＲ３は割込みをクリアする為にドライバ内で処理されてしまうので、割込み発生後直接読んでも発生要因を特定出来ません。 入力パラメータ：int No　　// カード番号（カード上のロータリースイッチの設定値） long* Rr3AX　// ＡＸ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3AY　// ＡＹ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3AZ　// ＡＺ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3AU　// ＡＵ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3BX　// ＢＸ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3BY　// ＢＹ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3BZ　// ＢＺ軸のＲＲ３を格納する為のバッファへのポインタ long* Rr3BU　// ＢＵ軸のＲＲ３を格納する為のバッファへのポインタ 戻り値　　：なし 使用例： long Rr3AX,Rr3AY,long Rr3AZ,long Rr3AU,long Rr3BX,long Rr3BY,long Rr3BZ,long Rr3BU; ReadEventMC8080P(CARD_NO, &Rr3AX,&Rr3AY,&Rr3AZ,&Rr3AU,&Rr3BX,&Rr3BY,&Rr3BZ,&Rr3BU); 注意：この関数を実行するとＲＲ３の値はリセットされます。

使用例で引数に使用している定数は次項の定数定義をご参照ください。

■ＶＣにて割込み処理する場合の注意

①複数枚ボード使用時、関数SetEventMC8080Pにて、割込みを処理する関数のアドレスは、ボード毎に異なるアドレスをセットして下さい。複数のボードの割込み関数のアドレスが同一アドレスに設定された場合、割込要因を認識する関数ReadEventMC8080Pにおいて、割込み発生以外の要因を読み込み、発生要因を特定できないためです。

（カード番号０の場合）

```
OutpMC8080P(0,  MCX304A_WR0,  0xFF);    //  全軸指定
OutpMC8080P(0,  MCX304A_WR1,  0x8000);  //  停止時割込発生
SetEventMC8080P(0,(LPTHREAD_START_ROUTINE)MC8080P_EventProc0);  //  関数のアドレス設定
・・・
```

（カード番号１の場合）

```
OutpMC8080P(1,  MCX304A_WR0,  0xFF);    //  全軸指定
OutpMC8080P(1,  MCX304A_WR1,  0x8000);  //  停止時割込発生
SetEventMC8080P(1,(LPTHREAD_START_ROUTINE)MC8080P_EventProc1);  //  関数のアドレス設定
・・・
```

②割込み処理する関数は、各ボード毎に別々に設けて下さい。各ボードの割込要因認識は、関数 ReadEventMC8080P で出来ます。

（カード番号０の割込みイベントの処理プロセス）

```
void MC8080P_EventProc0(void)
{
        long Rr3AX, Rr3AY, Rr3AZ,Rr3AU,Rr3BX,Rr3BY,Rr3BZ,Rr3BU;
             ReadEventMC8080P(0, &Rr3AX, &Rr3AY, &Rr3AZ, &Rr3AU, &Rr3BX, &Rr3BY, &Rr3BZ, &Rr3BU);
              ・・・・
}
```

（カード番号１の割込みイベントの処理プロセス）

```
void MC8080P_EventProc1(void)
{
        long Rr3AX, Rr3AY, Rr3AZ,Rr3AU,Rr3BX,Rr3BY,Rr3BZ,Rr3BU;
             ReadEventMC8080P(1, &Rr3AX, &Rr3AY, &Rr3AZ, &Rr3AU, &Rr3BX, &Rr3BY, &Rr3BZ, &Rr3BU);
              ・・・・
}
```

③割込み処理する関数を開放する場合は ResetEventMC8080P で行なって下さい。

■ＶＣ＋＋で使用する時の定数定義。

```
// アドレス定義
// MCX304 A
#define MCX304A_WR0             0x0000
#define MCX304A_WR1             0x0001
#define MCX304A_WR2             0x0002
#define MCX304A_WR3             0x0003
#define MCX304A_WR4             0x0004
#define MCX304A_WR5             0x0005
#define MCX304A_WR6             0x0006
#define MCX304A_WR7             0x0007
#define MCX304A_RR0             0x0000
#define MCX304A_RR1             0x0001
#define MCX304A_RR2             0x0002
#define MCX304A_RR3             0x0003
#define MCX304A_RR4             0x0004
#define MCX304A_RR5             0x0005
#define MCX304A_RR6             0x0006
#define MCX304A_RR7             0x0007

// MCX304 B
#define MCX304B_WR0             0x0008
#define MCX304B_WR1             0x0009
#define MCX304B_WR2             0x000A
#define MCX304B_WR3             0x000B
#define MCX304B_WR4             0x000C
#define MCX304B_WR5             0x000D
#define MCX304B_WR6             0x000E
#define MCX304B_WR7             0x000F
#define MCX304B_RR0             0x0008
#define MCX304B_RR1             0x0009
#define MCX304B_RR2             0x000A
#define MCX304B_RR3             0x000B
#define MCX304B_RR4             0x000C
#define MCX304B_RR5             0x000D
#define MCX304B_RR6             0x000E
#define MCX304B_RR7             0x000F

// 8255
#define MSM82C55_PA             0x0010
#define MSM82C55_PB             0x0011
#define MSM82C55_PC             0x0012
#define MSM82C55_CMD            0x0013

// カード番号ロータリースイッチ
#define CARD_NO_SW              0x0018

// 軸指定定数
#define AXIS_ALL                        0x0F
#define AXIS_X                          0x01
#define AXIS_Y                          0x02
#define AXIS_Z                          0x04
#define AXIS_U                          0x08
```

■ＶＣ＋＋で使用する時の関数定義。

```
__declspec(dllimport) BOOL __stdcall OpenMC8080P(int No);                                   // MC8080Pをオープン
__declspec(dllimport) long __stdcall InpMC8080P(int No, long Adr);                          // MC8080Pからポート入力
__declspec(dllimport) void __stdcall OutpMC8080P(int No, long Adr, long Dat);               // MC8080Pへポート出力
__declspec(dllimport) void __stdcall SetEventMC8080P(int No, LPTHREAD_START_ROUTINE UserThread);
                                                                    // MC8080Pへ割込みイベントの設定
__declspec(dllimport) void __stdcall ResetEventMC8080P(int No);             // MC8080Pへ割込みイベントの解除
__declspec(dllimport) BOOL __stdcall CloseMC8080P(int No);                  // MC8080Pをクローズ
__declspec(dllimport) void __stdcall CloseAllMC8080P(void);                 // MC8080Pを全てクローズ
__declspec(dllimport) void __stdcall ReadEventMC8080P(int No,
                                                    long* Rr3AX,
                                                    long* Rr3AY,
                                                    long* Rr3AZ,
                                                    long* Rr3AU,
                                                    long* Rr3BX,
                                                    long* Rr3BY,
                                                    long* Rr3BZ,
                                                    long* Rr3BU
                                                    );      // MC8080Pが割込みイベントを発生した時のRR3の値
```

注意：
各サービス関数を使用する前にOpenMC8080P()を必ず実行してください。
プログラム終了時にCloseMC8080P()又はCloseAllMC8080P()を実行してください。
SetEventMC8080P()を使用した時はCloseMC8080P()又はCloseAllMC8080P()の前でResetMC8080P()を実行して下さい。
OpenMC8080P関数実行前に各サービス関数を実行した場合の動作保証はできません。
接続していないMC8080Pのボード番号を誤って指定した場合も、各関数の動作の保証はできません。
割り込み処理関数を使用する場合はWindowsの性格上、割り込み発生からユーザー定義ルーチンへ制御が移行するまでの時間を保証することは出来ません。

■ＶＢでプログラミングする場合

関数名	機能　及び　内容
OpenMC8080P	MC8080Pの使用を開始する。 入力パラメータ：No As Variant ‘カード番号（カード上のロータリースイッチの設定値） 戻り値　　：Boolean オープンに成功するとTRUE、失敗するとFALSE 使用例　　： 　status = OpenMC8080P(0)　　‘ カード番号０をオープン
CloseMC8080P	MC8080Pの使用を終了する。 入力パラメータ：No As Variant ‘カード番号（カード上のロータリースイッチの設定値） 戻り値　　：Boolean クローズに成功するとTRUE、失敗するとFALSE 使用例： 　status = CloseMC8080P(0)　‘カード番号０をクローズ
CloseAllMC8080P	全てのMC8080Pの使用を終了する。 入力パラメータ：なし 戻り値　　：なし 使用例： 　status = CloseAllMC8080()　　‘全てのカードをクローズ
OutpMC8080P	出力ポートに１ワード書き込む。 入力パラメータ：No As Variant ‘カード番号（カード上のロータリースイッチの設定値） Adr As Long ‘書き込むアドレス Dat As Long ‘書き込むデータ 戻り値：　　なし 使用例： 　OutpMC8080P(0, MCX304A_R0, 0x8000)　‘ボードのソフトリセット
InpMC8080P	入力ポートから１ワード読み出す。 入力パラメータ：No As Variant ’カード番号（カード上のロータリースイッチの設定値） Adr As Long ‘読み出すアドレス 戻り値　　：Long 入力ポートから読み込んだ１ワード 使用例： 　data = InpMC8080P(0, RR0)　　‘ リードレジスタ RR0 の読み出し 　リードレジスタ番号の指定はヘッダ定義ファイルに宣言されている RR0～RR7 を使用してください。

使用例で引数に使用している定数は次項の定数定義をご参照ください。

■ＶＢで使用する時の定数定義。

```
        ' MC8080 制御関数

        ' アドレス定義
        ' MCX304 A
        Public Const MCX304A_WR0 = &H0
        Public Const MCX304A_WR1 = &H1
        Public Const MCX304A_WR2 = &H2
        Public Const MCX304A_WR3 = &H3
        Public Const MCX304A_WR4 = &H4
        Public Const MCX304A_WR5 = &H5
        Public Const MCX304A_WR6 = &H6
        Public Const MCX304A_WR7 = &H7
        Public Const MCX304A_RR0 = &H0
        Public Const MCX304A_RR1 = &H1
        Public Const MCX304A_RR2 = &H2
        Public Const MCX304A_RR3 = &H3
        Public Const MCX304A_RR4 = &H4
        Public Const MCX304A_RR5 = &H5
        Public Const MCX304A_RR6 = &H6
        Public Const MCX304A_RR7 = &H7

        ' MCX304 B
        Public Const MCX304B_WR0 = &H8
        Public Const MCX304B_WR1 = &H9
        Public Const MCX304B_WR2 = &HA
        Public Const MCX304B_WR3 = &HB
        Public Const MCX304B_WR4 = &HC
        Public Const MCX304B_WR5 = &HD
        Public Const MCX304B_WR6 = &HE
        Public Const MCX304B_WR7 = &HF
        Public Const MCX304B_RR0 = &H8
        Public Const MCX304B_RR1 = &H9
        Public Const MCX304B_RR2 = &HA
        Public Const MCX304B_RR3 = &HB
        Public Const MCX304B_RR4 = &HC
        Public Const MCX304B_RR5 = &HD
        Public Const MCX304B_RR6 = &HE
        Public Const MCX304B_RR7 = &HF

        ' 8255
        Public Const MSM82C55_PA = &H10
        Public Const MSM82C55_PB = &H11
        Public Const MSM82C55_PC = &H12
        Public Const MSM82C55_CMD = &H13

' カード番号ロータリースイッチ
        Public Const CARD_NO_SW = &H18

        ' 軸指定定数
        Public Const AXIS_ALL = &HF
        Public Const AXIS_X = &H1
        Public Const AXIS_Y = &H2
        Public Const AXIS_Z = &H4
        Public Const AXIS_U = &H8
```

■ＶＢで使用する時の関数定義。

```
' MC8080P.DLL 関数宣言

'=========================================
'MC8080Pをオープン
'=========================================
Declare Function OpenMC8080P Lib "MC8080P.DLL" ( _
                        ByVal No As Integer) _
                        As Boolean
'=========================================
'MC8080Pからポート入力
'=========================================
Declare Function InpMC8080P Lib "MC8080P.DLL" ( _
                        ByVal No As Integer, _
                        ByVal Adr As Long) _
                        As Long

'=========================================
'MC8080Pへポート出力
'=========================================
Declare Sub OutpMC8080P Lib "MC8080P.DLL" ( _
                        ByVal No As Integer, _
                        ByVal Adr As Long, _
                        ByVal Dat As Long)

'=========================================
'MC8080Pをクローズ
'=========================================
Declare Function CloseMC8080P Lib "MC8080P.DLL" ( _
                        ByVal No As Integer) _
                        As Boolean

'=========================================
'MC8080Pを全てクローズ
'=========================================
Declare Sub CloseAllMC8080P Lib "MC8080P.DLL" ()
```

注意：
各サービス関数を使用する前にOpenMC8080P()を必ず実行してください。
プログラム終了時にCloseMC8080P()又はCloseAllMC8080P()を実行してください。
OpenMC8080P関数実行前に各サービス関数を実行した場合の動作保証はできません。
接続していないMC8080Pのボード番号を誤って指定した場合も、各関数の動作の保証はできません。

8.7 ソフトウェアの内容

ソフトウェアの内容一覧です。ここではソフトウェアをＦＤに展開した例を示します。

```
\
+---LIB
|    +---VB6
|    |   +---MC8080P_DLL.bas             VB6用MC8080P.DLLデクリアファイル
|    +---VC6
|        +---MC8080P.lib                 VC6用ライブラリファイル
|        +---MC8080P_DLL.h               VC6用ライブラリインクルードファイル
+---Driver
|    +---98                              Windows98/Me用ドライバ
|    |   +---MC8080P.sys                 デバイスドライバ本体
|    |   +---MC8080P.inf                 デバイスドライバのインストール用プログラム
|    |   +---MC8080P.dll                 デバイスドライバを使用するためのダイナミックリンクライブラリ
|    |   +---MC8080P.cat                 セキュリティー保護ファイル
|    +---NT                              Windows NT用ドライバ
|    |   +---MC8080P.sys                 デバイスドライバ本体
|    |   +---Install.exe                 デバイスドライバのインストール用プログラム
|    |   +---MC8080P.dll                 デバイスドライバを使用するためのダイナミックリンクライブラリ
|    |   +---MC8080P.reg                 デバイスドライバを手動でレジストリに登録する時に使用
|    +---2000                            Windows2000用ドライバ
|    |   +---MC8080P.sys                 デバイスドライバ本体
|    |   +---MC8080P.inf                 デバイスドライバのインストール用プログラム
|    |   +---MC8080P.dll                 デバイスドライバを使用するためのダイナミックリンクライブラリ
|    |   +---MC8080P.cat                 セキュリティー保護ファイル
|    +---XP                              WindowsXP用ドライバ
|        +---MC8080P.sys                 デバイスドライバ本体
|        +---MC8080P.inf                 デバイスドライバのインストール用プログラム
|        +---MC8080P.dll                 デバイスドライバを使用するためのダイナミックリンクライブラリ
|    |   +---MC8080P.cat                 セキュリティー保護ファイル
+---Sample
     +---MC8080P SMP VB6                 VB6用のサンプルプログラム
     |   +---MC8080P_SMP_VB6.vbw         プロジェクトワークスペース
     |   +---MC8080P_SMP_VB6.vbp         プロジェクトファイル
     |   +---MC8080P_SMP_VB6.PDM         ＰＷＭファイル（プロジェクトで使用）
     |   +---MC8080P_DLL.bas             ＤＬＬ関数デクリアファイル
     |   +---MC8080P.bas                 MC8080P制御関数
     |   +---MainWnd.frm                 メインウインド定義ファイル
     +---MC8080P SMP VC6                 VC6用のサンプルプログラム
         +---MC8080P_SMP_VC6.APS         プロジェクトで使用
         +---MC8080P.CPP                 プロジェクトで使用
         +---MC8080P_SMP_VC6.cpp         アプリケーションクラスメンバ関数定義ファイル
         +---MC8080P_SMP_VC6Dlg.cpp      ダイアログクラスメンバ関数定義ファイル
         +---StdAfx.cpp                  ＭＦＣ関数定義ファイル
         +---MC8080P_SMP_VC6.clw         プロジェクトで使用
         +---MC8080P.H                   MC8080P制御関数定義
         +---MC8080P_DLL.h               ＤＬＬ関数インクルードファイル
         +---MC8080P_SMP_VC6.h           アプリケーションクラス定義ファイル
         +---MC8080P_SMP_VC6Dlg.h        ダイアログクラス定義ファイル
         +---Resource.h                  リソースインクルードファイル
         +---StdAfx.h                    ＭＦＣ関数インクルード
         +---MC8080P.lib                 ＤＬＬ関数ライブラリファイル
         +---MC8080P_SMP_VC6.ncb         プロジェクトで使用
         +---MC8080P_SMP_VC6.opt         プロジェクトで使用
         +---MC8080P_SMP_VC6.plg         プロジェクトで使用
         +---ReadMe.txt                  各ファイルの詳細説明ファイル
         +---MC8080P_SMP_VC6.dsp         プロジェクトファイル
         +---MC8080P_SMP_VC6.dsw         プロジェクトワークスペースファイル
         +---MC8080P_SMP_VC6.rc          リソース定義ファイル
         +---res
         |     +---MC8080P_SMP_VC6.ico   アイコン定義ファイル
         |     +---MC8080P_SMP_VC6.rc2   リソース定義ファイル２
         +---Release
               +---MC8080P_SMP_VC6.exe   サンプルプログラムの実行形式
```

8.8 ＭＣ８０８０Ｐの基本関数と自動原点サーチのサンプルプログラム例

8.8.1 ＶＣ＋＋サンプルプログラムにおけるＭＣ８０８０Ｐの基本関数

ＭＣ８０８０ＰのＶＣ＋＋用基本関数は、付属されているソフトウェアのＶＣ＋＋サンプルプログラムの中に、ソースレベルでプログラム例を提供しております。尚基本関数の詳細についてはＭＣＸ３０４取扱説明書を併せて参照ください。
ＭＣ８０８０ＰにはＭＣＸ３０４－ＡとＭＣＸ３０４－Ｂの２個が使用され、下記表の□内にはＭＣＸ３０４－Ａの場合は Ａ 、ＭＣＸ３０４－Ｂの場合は Ｂ が相当します。
尚本プログラム例はＦＤの「\Sample\MC8080P SMP VC6\MC8080P.CPP」の中にあります。

基本関数	機能内容
・ライトレジスタ０設定 MCX304□-wreg0(int wdata)	・ＷＲ０コマンドレジスタに軸指定＋命令コードをセットする。 ・入力パラメータ： int wdata……軸指定＋命令コード ・戻り値：　なし
・ライトレジスタ１設定 MCX304□_wreg1(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、モードレジスタ１（ＷＲ１）をセットする。 ・入力パラメータ： int axis……軸指定 int wdata……モードレジスタ１の内容 ・戻り値：　なし
・ライトレジスタ２設定 MCX304□_wreg2(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、モードレジスタ２（ＷＲ２）をセットする。 ・入力パラメータ： int axis……軸指定 int wdata……モードレジスタ２の内容 ・戻り値：　なし
・ライトレジスタ３設定 MCX304□_wreg3(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、モードレジスタ３（ＷＲ３）をセットする。 ・入力パラメータ： int axis……軸指定 int wdata……モードレジスタ３の内容 ・戻り値：　なし
・ライトレジスタ４設定 MCX304□_wreg4(int wdata)	・アウトプットレジスタ１（ＷＲ４）をセットする。 ・入力パラメータ： int wdata……アウトプットレジスタ１（ＷＲ４）の内容 ・戻り値：　なし
・ライトレジスタ５設定 MCX304□_wreg5(int wdata)	・アウトプットレジスタ２（ＷＲ５）をセットする。 ・入力パラメータ： int wdata……アウトプットレジスタ２（ＷＲ５）の内容 ・戻り値：　なし
・ライトレジスタ６設定 MCX304□_wreg6(int wdata)	・ライトデータレジスタ１（ＷＲ６）をセットする。 ・入力パラメータ： int wdata……ライトデータレジスタ１（ＷＲ６）の内容 ・戻り値：　なし
・リードレジスタ０読み出し MCX304□-rreg0(void)	・主ステタスレジスタからステタスデータを読み出す。 ・入力パラメータ：　なし ・戻り値： int……主ステタスレジスタ（ＲＲ０）の内容
・リードレジスタ１読み出し MCX304□_rreg1(int axis)	・軸を指定(ＷＲ０コマンド)し、ステタスレジスタ１（ＲＲ１）を読み出す。 ・入力パラメータ： int axis……軸指定 ・戻り値： int……ステタスレジスタ１(ＲＲ１)の内容
・リードレジスタ２読み出し MCX304□_rreg2(int axis)	・軸を指定(ＷＲ０コマンド)し、ステタスレジスタ２（ＲＲ２）を読み出す。 ・入力パラメータ： int axis……軸指定 ・戻り値： int……ステタスレジスタ２(ＲＲ２)の内容
・リードレジスタ３読み出し MCX304□_rreg3(int axis)	・軸を指定(ＷＲ０コマンド)し、ステタスレジスタ３（ＲＲ３）を読み出す。 ・入力パラメータ： int axis……軸指定 ・戻り値： int……ステタスレジスタ３(ＲＲ３)の内容

基本関数	機能内容
・リードレジスタ４読み出し MCX304□_rreg4(void)	・インプットレジスタ１（ＲＲ４）を読み出す。 ・入力パラメータ：　なし ・戻り値： 　　int・・・・・・・・・インプットレジスタ１（ＲＲ４)の内容
・リードレジスタ５読み出し MCX304□_rreg5(void)	・インプットレジスタ２（ＲＲ５）を読み出す。 ・入力パラメータ：　なし ・戻り値： 　　int・・・・・・・・・インプットレジスタ２（ＲＲ５)の内容
・命令書き込み MCX304□_command(int axis, int cmd)	・軸を指定(ＷＲ０コマンド)し、命令コードを（ＷＲ０）にセットする。 ・入力パラメータ： 　　int axis・・・・・・・・軸指定 　　int cmd・・・・・・・・命令コード ・戻り値：　　なし
・レンジ設定 MCX304□_range(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、レンジを（ＷＲ６，７）にセットする。 ・入力パラメータ： 　　int axis・・・・・・・・軸指定 　　long wdata・・・・・レンジの内容 ・戻り値：　　なし
・加加速度（Ｋ）設定 MCX304□_acac(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、加加速度を設定する。 ・入力パラメータ： 　　int axis・・・・・・・・軸指定 　　int wdata・・・・・・・加加速度の内容 ・戻り値：　　なし
・加速度（Ａ）設定 MCX304□_acc(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、加速度を設定する。 ・入力パラメータ： 　　int axis・・・・・・・・軸指定 　　int wdata・・・・・・・加速度の内容 ・戻り値：　　なし
・減速度（Ｄ）設定 MCX304□_dec(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、減速度を設定する。 ・入力パラメータ： 　　int axis・・・・・・・・軸指定 　　int wdata・・・・・・・減速度の内容 ・戻り値：　　なし
・初速度（ＳＶ）設定 MCX304□_startv(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、初速度を設定する。 ・入力パラメータ： 　　int axis・・・・・・軸指定 　　int wdata・・・・・初速度の内容 ・戻り値：　　なし
・ドライブ速度（Ｖ）設定 MCX304□_speed(int axis, int wdata)	・軸を指定(ＷＲ０コマンド)し、ドライブ速度を設定する。 ・入力パラメータ： 　　int axis・・・・・・軸指定 　　int wdata・・・・・ドライブ速度の内容 ・戻り値：　　なし
・出力パルス数／終点（Ｐ）設定 MCX304□_pulse(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、出力パルス数を設定する。 ・入力パラメータ： 　　int axis・・・・・・軸指定 　　long wdata・・・・出力パルス数の内容 ・戻り値：　　なし
・マニュアル減速点（ＤＰ）設定 MCX304□_decp(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、マニュアル減速点を設定する。 ・入力パラメータ： 　　int axis・・・・・・・軸指定 　　long wdata・・・・・・マニュアル減速点(ＤＰ) 　　の内容 ・戻り値：　　なし
・論理位置カウンタ（ＬＰ）設定 MCX304□_lp(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、論理位置カウンタを設定する。 ・入力パラメータ： 　　int axis・・・・・・・軸指定 　　long wdata・・・・論理位置カウンタ（ＬＰ） 　　の内容 ・戻り値：　　なし

基本関数	機能内容
・実位置カウンタ（ＥＰ）設定 MCX304□_lp(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、実位置カウンタを設定する。 ・入力パラメータ： int axis･･･････軸指定 long wdata････実位置カウンタ（ＥＰ）の内容 ・戻り値：　なし
・ＣＯＭＰ＋（ＣＰ）設定 MCX304□_compp(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、ＣＯＭＰ＋レジスタ（ＣＰ）を設定する。 ・入力パラメータ： int axis･･･････軸指定 long wdata･････ＣＯＭＰ＋レジスタの内容 ・戻り値：　なし
・ＣＯＭＰ－（ＣＭ）設定 MCX304□_compm(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、ＣＯＭＰ－レジスタ（ＣＭ）を設定する。 ・入力パラメータ： int axis･･･････軸指定 long wdata･････ＣＯＭＰ－レジスタの内容 ・戻り値：　なし
・加速カウンタオフセット（ＡＯ）設定 MCX304□_accofst(int axis, long wdata)	・軸を指定(ＷＲ０コマンド)し、加速カウンタオフセット（ＡＯ）を設定する。 ・入力パラメータ： int axis･･･････軸指定 long wdata･････加速カウンタオフセット（ＡＯ）の内容 ・戻り値：　なし
・論理位置カウンタ値（ＬＰ）読み出し MCX304□_readlp(int axis)	・軸を指定(ＷＲ０コマンド)し、論理位置カウンタ値（ＬＰ）を読み出す。 ・入力パラメータ： int axis･･･････軸指定 ・戻り値： long･･･････論理位置カウンタ値の内容
・実位置カウンタ値（ＥＰ）読み出し MCX304□_readep(int axis)	・軸を指定(ＷＲ０コマンド)し、実位置カウンタ　値（ＥＰ）を読み出す。 ・入力パラメータ： int axis･･･････軸指定 ・戻り値： long･･･････実位置カウンタ値の内容
８２５５基本関数 ・ポートＡデータ出力 MSM82C55_OutpA(int wdata)	・８２５５の汎用出力ポートＡをセットする。 ・入力パラメータ： int wdata･････ポートＡの内容 ・戻り値：　なし
８２５５基本関数 ・ポートＢデータ出力 MSM82C55_OutpB(int wdata)	・８２５５の汎用出力ポートＢをセットする。 ・入力パラメータ： int wdata･････ポートＢの内容 ・戻り値：　なし
８２５５基本関数 ・ポートＣデータ出力 MSM82C55_OutpC(int wdata)	・８２５５の汎用出力ポートＣをセットする。 ・入力パラメータ： int wdata･････ポートＣの内容 ・戻り値：　なし
・オープン及び初期設定 MC8080P_Open(void)	・ＭＣ８０８０Ｐボードのオープン処理及び初期設定を行う。 ・入力パラメータ：　なし ・戻り値：　なし
・クローズ MC8080P_Close(void)	・ＭＣ８０８０Ｐボードのクローズ処理を行う。 ・入力パラメータ：　なし ・戻り値：　なし
・割込みイベントの処理プロセス MC8080P_EventProc(void)	・ＭＣ８０８０Ｐボードの割込みイベントの処理プロセスを行う。 ・入力パラメータ：　なし ・戻り値：　なし
・割込みイベントの設定 MC8080P_SetEvent(void)	・ＭＣ８０８０Ｐボードの割込みイベントの設定を行う。 ・入力パラメータ：　なし ・戻り値：　なし
・割込みイベントの解除 MC8080P_ResetEvent(void)	・ＭＣ８０８０Ｐボードの割込みイベントの解除を行う。 ・入力パラメータ：　なし ・戻り値：　なし

8.8.2 ＶＣ＋＋サンプルプログラムにおける自動原点出し処理のプログラム例

ＶＣ＋＋サンプルプログラムの中に、自動原点出しのサンプルプログラムがソースレベルであります。
尚自動原点出しはプログラム例の中で３つのモードがあり、３つのモードの詳細は、ＭＣＸ３０４取扱説明書の中の自動原点出しの実例（Ｐ１９－Ｐ２２）を基本に作成されており、併せて参照ください。

モード１・・・・・原点近傍、原点、Ｚ相信号による原点出し（Ｐ１９参照）
モード２・・・・・原点信号のみの原点出し（Ｐ２０参照）
モード３・・・・・リミット信号を用いた原点出し（Ｐ２１参照）

ＭＣ８０８０ＰにはＭＣＸ３０４－ＡとＭＣＸ３０４－Ｂの２個が使用され、下記表の□内にはＭＣＸ３０４－Ａの場合は ａ 、ＭＣＸ３０４－Ｂの場合は ｂ が相当します。
尚本プログラム例はＦＤの「\Sample\MC8080P SMP VC6\MC8080P_SMP_VC6Dlg.cpp」の中にあります。

原点出しサブルーチン名	機能内容
・原点出しモード１ home_sh1md_□(int zaxis)	・軸を指定し、原点出し１モードによる自動原点出しを行う。 ・入力パラメータ： int zaxis・・・・・・・軸指定 ・戻り値：　なし
・原点出しモード２ home_sh2md_□(int zaxis)	・軸を指定し、原点出し２モードによる自動原点出しを行う。 ・入力パラメータ： int zaxis・・・・・・・軸指定 ・戻り値：　なし
・原点出しモード３ home_sh3md_□(int zaxis)	・軸を指定し、原点出し３モードによる自動原点出しを行う。 ・入力パラメータ： int zaxis・・・・・・・軸指定 ・戻り値：　なし

8.8.3 ＶＢサンプルプログラムにおけるＭＣ８０８０Ｐの基本関数

ＭＣ８０８０ＰのＶＢ用基本関数は、付属したソフトウェアのＶＢサンプルプログラムの中に、ソースレベルでプログラム例を提供しております。尚基本関数の詳細についてはＭＣＸ３０４取扱説明書を併せて参照ください。
ＭＣ８０８０ＰにはＭＣＸ３０４－ＡとＭＣＸ３０４－Ｂの２個が使用され、下記表の□内にはＭＣＸ３０４－Ａの場合は Ａ 、ＭＣＸ３０４－Ｂの場合は Ｂ が相当します。尚ＶＢにおいて、ＭＣ８０８０Ｐの割込みは使用できません。
尚本プログラム例はＦＤの「\Sample\MC8080P SMP VB6\MC8080P.bas」の中にあります。

基本関数	機能内容
・ライトレジスタ０設定 MCX304□-wreg0 (Wdata As Long)	・ＷＲ０コマンドレジスタに軸指定＋命令コードをセットする。 ・入力パラメータ： Long Wdata・・・・・・軸指定＋命令コード ・戻り値：　なし
・ライトレジスタ１設定 MCX304□_wreg1 (axis As Integer,Wdata As Long)	・軸を指定(ＷＲ０コマンド)し、モードレジスタ１（ＷＲ１）をセットする。 ・入力パラメータ： Intger axis・・・・・・・・軸指定 Long Wdata・・・・・・モードレジスタ１の内容 ・戻り値：　なし
・ライトレジスタ２設定 MCX304□_wreg2 (axis As Integer,Wdata As Long)	・軸を指定(ＷＲ０コマンド)し、モードレジスタ２（ＷＲ２）をセットする。 ・入力パラメータ： Integer axis・・・・・・・・軸指定 Long Wdata・・・・・・・モードレジスタ２の内容 ・戻り値：　なし
・ライトレジスタ３設定 MCX304□_wreg3 (axis As Integer,Wdata As Long)	・軸を指定(ＷＲ０コマンド)し、モードレジスタ３（ＷＲ３）をセットする。 ・入力パラメータ： Integer axis・・・・・・・・軸指定 Long Wdata・・・・・・・モードレジスタ３の内容 ・戻り値：　なし
・ライトレジスタ４設定 MCX304□_wreg4 (Wdata As Long)	・アウトプットレジスタ１（ＷＲ４）をセットする。 ・入力パラメータ： Long Wdata・・・・・・アウトプットレジスタ１（ＷＲ４）の内容 ・戻り値：　なし

基本関数	機能内容
・ライトレジスタ５設定 MCX304□_wreg5 (Wdata As Long)	・アウトプットレジスタ２（ＷＲ５）をセットする。 ・入力パラメータ： Long Wdata……アウトプットレジスタ２（ＷＲ５）の内容 ・戻り値：　なし
・ライトレジスタ６設定 MCX304□_wreg6 (Wdata As Integer)	・ライトデータレジスタ１（ＷＲ６）をセットする。 ・入力パラメータ： Long Wdata……ライトデータレジスタ１（ＷＲ６）の内容 ・戻り値：　なし
・リードレジスタ０読み出し MCX304□-rreg0()As Long	・主ステタスレジスタからステタスデータを読み出す。 ・入力パラメータ：　なし ・戻り値： Long……主ステタスレジスタ（ＲＲ０）の内容
・リードレジスタ１読み出し MCX304□_rreg1 (axis As Integer)	・軸を指定(ＷＲ０コマンド)し、ステタスレジスタ１（ＲＲ１）を読み出す。 ・入力パラメータ： Integer axis……軸指定 ・戻り値： Long……ステタスレジスタ１(ＲＲ１)の内容
・リードレジスタ２読み出し MCX304□_rreg2 (axis As Integer)	・軸を指定(ＷＲ０コマンド)し、ステタスレジスタ２（ＲＲ２）を読み出す。 ・入力パラメータ： Integer axis……軸指定 ・戻り値： Long……ステタスレジスタ２(ＲＲ２)の内容
・リードレジスタ３読み出し MCX304□_rreg3 (axis As Integer)	・軸を指定(ＷＲ０コマンド)し、ステタスレジスタ３（ＲＲ３）を読み出す。 ・入力パラメータ： Integer axis……軸指定 ・戻り値： Long……ステタスレジスタ３(ＲＲ３)の内容
・リードレジスタ４読み出し MCX304□_rreg4()As Long	・インプットレジスタ１（ＲＲ４）を読み出す。 ・入力パラメータ：　なし ・戻り値： Long……インプットレジスタ１（ＲＲ４)の内容
・リードレジスタ５読み出し MCX304□_rreg5()As long	・インプットレジスタ２（ＲＲ５）を読み出す。 ・入力パラメータ：　なし ・戻り値： Long……インプットレジスタ２（ＲＲ５)の内容
・命令書き込み MCX304□_command (axis As Integer,cmd As Integer)	・軸を指定(ＷＲ０コマンド)し、命令コードを（ＷＲ０）にセットする。 ・入力パラメータ： Integer axis……軸指定 Integer cmd……命令コード ・戻り値：　なし
・レンジ設定 MCX304□_range (axis As Integer,Wdata As Long)	・軸を指定(ＷＲ０コマンド)し、レンジを設定する。 ・入力パラメータ： Integer axis……軸指定 Long Wdata……レンジの内容 ・戻り値：　なし
・加加速度（Ｋ）設定 MCX304□_acac (axis As Integer,Wdata As Long)	・軸を指定(ＷＲ０コマンド)し、加加速度を設定する。 ・入力パラメータ： Integer axis……軸指定 Long Wdata……加加速度の内容 ・戻り値：　なし
・加速度（Ａ）設定 MCX304□_acc (axis As Integer,Wdata As Long)	・軸を指定(ＷＲ０コマンド)し、加速度を設定する。 ・入力パラメータ： Integer axis……軸指定 Long Wdata……加速度の内容 ・戻り値：　なし

基本関数	機能内容
・減速度（D）設定 MCX304□_dec (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、減速度を設定する。 ・入力パラメータ： Integer axis・・・・・・・・軸指定 Long Wdata・・・・・・・減速度の内容 ・戻り値：　なし
・初速度（SV）設定 MCX304□_startv (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、初速度を設定する。 ・入力パラメータ： Integer axis・・・・・・軸指定 Long Wdata・・・・・初速度の内容 ・戻り値：　なし
・ドライブ速度（V）設定 MCX304□_speed (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、ドライブ速度を設定する。 ・入力パラメータ： Integer axis・・・・・・軸指定 Long Wdata・・・・・ドライブ速度の内容 ・戻り値：　なし
・出力パルス数／終点（P）設定 MCX304□_pulse (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、出力パルス数を設定する。 ・入力パラメータ： Integer axis・・・・・・・軸指定 Long Wdata・・・・・・出力パルス数の内容 ・戻り値：　なし
・マニュアル減速点（DP）設定 MCX304□_decp (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、マニュアル減速点を設定する。 ・入力パラメータ： Integer axis・・・・・・・軸指定 Long Wdata・・・・・・マニュアル減速点(DP)の内容 ・戻り値：　なし
・論理位置カウンタ（LP）設定 MCX304□_lp (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、論理位置カウンタを設定する。 ・入力パラメータ： Integer axis・・・・・・・軸指定 Long Wdata・・・・・・論理位置カウンタ（LP）の内容 ・戻り値：　なし
・実位置カウンタ（EP）設定 MCX304□_lp (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、実位置カウンタを設定する。 ・入力パラメータ： Integer axis・・・・・・・軸指定 Long wdata・・・・・・・実位置カウンタ（EP）の内容 ・戻り値：　なし
・COMP＋（CP）設定 MCX304□_compp (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、COMP＋レジスタ（CP）を設定する。 ・入力パラメータ： Integer axis・・・・・・軸指定 Long Wdata・・・・・COMP＋レジスタの内容 ・戻り値：　なし
・COMP－（CM）設定 MCX304□_compm (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、COMP－レジスタ（CM）を設定する。 ・入力パラメータ： Integer axis・・・・・・軸指定 Long Wdata・・・・・COMP－レジスタの内容 ・戻り値：　なし
・論理位置カウンタ値（LP）読み出し MCX304□_readlp (axis As Integer)As Long	・軸を指定(WR0コマンド)し、論理位置カウンタ値（LP）を読み出す。 ・入力パラメータ： Integer axis・・・・・軸指定 ・戻り値： Long・・・・・・・・・・論理位置カウンタ値の内容
・実位置カウンタ値（EP）読み出し MCX304□_readep (axis As Integer)As Long	・軸を指定(WR0コマンド)し、実位置カウンタ値（EP）を読み出す。 ・入力パラメータ： Integer axis・・・・・・軸指定 ・戻り値： Long・・・・・・・・・・・実位置カウンタ値の内容
・加速カウンタオフセット（AO）設定 MCX304□_accofst (axis As Integer,Wdata As Long)	・軸を指定(WR0コマンド)し、加速カウンタオフセット（AO）を設定する。 ・入力パラメータ： Integer axis・・・・・・・・軸指定 Long Wdata・・・・・・・加速カウンタオフセット（AO）の内容 ・戻り値：　なし

基本関数	機能内容
８２５５基本関数 ・ポートＡデータ出力 MSM82C55_OutpA (Wdata As Long)	・８２５５の汎用出力ポートＡをセットする。 ・入力パラメータ： Long Wdata・・・・・ポートＡの内容 ・戻り値：　なし
８２５５基本関数 ・ポートＢデータ出力 MSM82C55_OutpB (Wdata As Long)	・８２５５の汎用出力ポートＢをセットする。 ・入力パラメータ： Long Wdata・・・・・ポートＢの内容 ・戻り値：　なし
８２５５基本関数 ・ポートＣデータ出力 MSM82C55_OutpC (Wdata As Long)	・８２５５の汎用出力ポートＣをセットする。 ・入力パラメータ： Long Wdata・・・・・ポートＣの内容 ・戻り値：　なし
・オープン及び初期設定 MC8080P_Open()	・ＭＣ８０８０Ｐボードのオープン処理及び初期設定を行う。 ・入力パラメータ：　なし ・戻り値：　なし
・クローズ MC8080P_Close(void)	・ＭＣ８０８０Ｐボードのクローズ処理を行う。 ・入力パラメータ：　なし ・戻り値：　なし

8.8.4　ＶＢサンプルプログラムにおける自動原点出し処理のプログラム例

ＶＢサンプルプログラムの中に、自動原点出しのサンプルプログラムがソースレベルであります。
尚自動原点出しはプログラム例の中で３つのモードがあり、３つのモードの詳細は、ＭＣＸ３０４取扱説明書の自動原点出しの実例（Ｐ１９－Ｐ２２）を基本に作成されており、併せて参照ください。

モード１・・・・・原点近傍、原点、Ｚ相信号による原点出し（Ｐ１９参照）
モード２・・・・・原点信号のみの原点出し（Ｐ２０参照）
モード３・・・・・リミット信号を用いた原点出し（Ｐ２１参照）

ＭＣ８０８０ＰにはＭＣＸ３０４－ＡとＭＣＸ３０４－Ｂのが使用され、下記表の□内にはＭＣＸ３０４－Ａの場合は ａ 、ＭＣＸ３０４－Ｂの場合は ｂ が相当します。
尚本プログラム例はＦＤの「\Sample\MC8080P SMP VB6\MainWnd.frm」の中にあります。

原点出しサブルーチン名	機能内容
・原点出しモード１ home_sh1md_□ (zaxis As Integer)	・軸を指定し、原点出し１モードによる自動原点出しを行う。 ・入力パラメータ： Integer zaxis・・・・・・・軸指定 ・戻り値：　なし
・原点出しモード２ home_sh2md_□ (zaxis As Integer)	・軸を指定し、原点出し２モードによる自動原点出しを行う。 ・入力パラメータ： Integer zaxis・・・・・・・軸指定 ・戻り値：　なし
・原点出しモード３ home_sh3md_□ (zaxis As Integer)	・軸を指定し、原点出し３モードによる自動原点出しを行う。 ・入力パラメータ： Integer zaxis・・・・・・・軸指定 ・戻り値：　なし

8.9　デバイスドライバを再インストールする場合

デバイスドライバのバージョンを新しくする等で、デバイスドライバを再インストールする時は、以下の手順で行ってください。
①現在のデバイスドライバを削除してください。削除は８．３項「取り外し」を参照に行ってください。
②新しくインストールするデバイスドライバにてインストールしてください。８．２項以降参照
注意：デバイスドライバを取り外しても、ＭＣ８０８０Ｐのドライバソフトは、システムの中に残っています。再インストールするドライバのバージョンが変更になっている場合は上書きされますが、同一バージョンの場合は上書きされず、更新されません。同一バージョンのドライバソフトを更新したい場合は
ＭＣ８０８０Ｐ．ＳＹＳ
ＭＣ８０８０Ｐ．ＤＬＬ
のソフトをシステムがら削除して、再インストールしてください。

9．仕様まとめ

● 制御軸　　８軸 （独立・同時制御）

ＰＣＩバスインターフェイス

● データビット幅　　１６Bit
● I/O占有アドレス　　６４　アドレスはPnP機能によって任意に決定
● 割り込み　　IRQ　PnP機能によって任意に接続

各軸共通仕様

■ ドライブパルス出力

● 出力回路：　　差動ラインドライバ（ＡＭ２６Ｃ３１）出力
● 出力速度範囲：　　1 PPS　～ ４ MPPS
● 出力速度精度：　　±0.1%以下（設定値に対して）
● 速度倍率：　　1　～ 500
● Ｓ字用加加速度：　　954　～ 62.5×10^{6}PPS/SEC^2 (倍率=1の時)
（加減速度の増減率）　　477×10^{3}　～ 31.25×10^{9}PPS/SEC^2(倍率=500の時)
● 加/減速度：　　125　～ 1×10^{6}PPS/SEC　(倍率=1の時)
62.5×10^{3}　～ 500×10^{6}PPS/SEC　(倍率=500の時)
● 初速度：　　1　～ 8,000PPS　(倍率=1の時)
500PPS　～ 4×10^{6}PPS　(倍率=500の時)
● ドライブ速度：　　1　～ 8,000PPS　(倍率=1の時)
500PPS　～ 4×10^{6}PPS　(倍率=500の時)
● 出力パルス数：　　0　～ 268,435,455（定量ドライブ）

● 速度カーブ：　　定速／直線加減速／放物線Ｓ字加減速ドライブ
● 定量ドライブの減速モード：　　自動減速（非対称台形も可能）／マニュアル減速

● ドライブ中の出力パルス数、ドライブ速度の変更可能
● 独立２パルス／１パルス・方向　方式選択可能。
● パルスの論理レベル選択可能。

■ エンコーダＡ相／Ｂ相入力

● 入力回路：　高速フォトカプラ入力。差動ラインドライバとの接続可能。
● ２相パルス／アップダウンパルス入力選択可能。
● ２相パルス　１，２，４逓倍選択可能。

■ 位置カウンタ

● 論理位置カウンタ（出力パルス用）カウント範囲：　-2,147,483,648 ～ +2,147,483,647
● 実位置カウンタ　（入力パルス用）カウント範囲：　-2,147,483,648 ～ +2,147,483,647
常時書き込み、読み出し可能

■ コンペアレジスタ

● COMP+レジスタ位置比較範囲：　-1,073,741,824 ～ +1,073,741,824
● COMP-レジスタ位置比較範囲：　-1,073,741,824 ～ +1,073,741,824
● ソフトウェアリミットとして動作可能。

■ 自動原点出し

● ステップ１（高速原点近傍サーチ)→ステップ２（低速原点サーチ）→ステップ３（低速エンコーダＺ相サーチ）→ステップ４（高速オフセット移動）を順次自動実行。
各ステップの有効／無効、検出方向選択可能。
● 偏差カウンタクリア出力：クリアパルス幅10μ～20msec，論理レベル選択可能。

■ 割り込み機能（補間を除く）

● 割り込み発生要因：　１ドライブパルス出力、位置カウンタ≧COMP-変化時、位置カウンタ＜COMP-変化時、位置カウンタ＜COMP+変化時、位置カウンタ≧COMP+変化時、加減速ドライブ中の定速開始時、加減速ドライブ中の定速終了時、ドライブ終了時。
いずれの要因に対しても有効／無効選択可能。

■ 外部信号によるドライブ操作

● EXPP、EXPM信号により、＋／－方向の定量／連続ドライブが可能。
● 入力回路：　フォトカプラ＋ＩＣ内蔵フィルタ。

■ **外部減速停止／即停止信号**

- STOP0～2　各軸３点（STOP0:原点近傍、STOP1:原点、STOP2:エンコーダＺ相入力用）
- 入力回路：　フォトカプラ＋ＩＣ内蔵フィルタ。

いずれの信号も有効／無効、論理レベルの選択可能。汎用入力としても使用可能。

■ **サーボモータ用入力信号**

- ALARM（アラーム）、INPOS（位置決め完了）。
- 入力回路：　フォトカプラ＋ＩＣ内蔵フィルタ

いずれの信号も有効／無効、論理レベルの選択可能。

■ **サーボモータ用出力信号**

- DCC（偏差カウンタクリア）OUT0信号と端子兼用。
- 出力回路：TD62503出力（オープンコレクタ出力）

■ **汎用出力信号**

- OUT0～3　各軸４点（合計：４×８軸＝32点）
- 出力回路：TD62503出力（オープンコレクタ出力）

■ **オーバランリミット信号入力**

- ＋方向、－方向各１点。
- 入力回路：　フォトカプラ＋ＩＣ内蔵フィルタ。

論理レベル選択可能。　アクティブ時、即停止／減速停止選択可能。

■ **緊急停止信号入力**

- 全軸でEMGN１点。全軸のドライブパルスを即停止。基板上のジャンパーで論理レベル選択可能。
- 入力回路：　フォトカプラ＋ＩＣ内蔵フィルタ。

その他

- **動作温度範囲：**　０℃　～＋４５℃（結露しないこと）
- **電源電圧：**　＋５Ｖ　±５％（消費電流　７００ｍＡmax）
- **外部電源電圧：**　＋２４Ｖ
- **基板外形寸法：**　174.6×106.7mm　（コネクタ、金具部は含まず）
- **I/0コネクタ型式：**　CN2:FX2B-100PA-1.27DS（ヒロセ）
  CN3:HIF3FC-50PA-2.54R（ヒロセ）
  CN4:HIF3FC-30PA-2.54R（ヒロセ）
- **付属品：**　CN2:FX2B-100SA-1.27R　（ヒロセ）１.２mケーブル付き
  CN3:HIF3BB-50D-2.54R　（ヒロセ）コネクタ単体
  CN4:HIF3BA-30D-2.54R　（ヒロセ）コネクタ単体