

# MPG-314 USER'S MANUAL

Last up date 2002/03/12

概略 .....	3
免責事項 .....	3
MPG-68K互換コマンド.....	3
互換性の注意 .....	3
互換（従来）のパルスコマンドとの関係.....	4
MPG-314専用コマンド(314ネイティブコマンド)概略 .....	6
制約.....	6
互換コマンドとネイティブコマンドの混在 .....	6
ビット定数.....	7
MCX-314の重要なレジスタとアクセス方法(参考) .....	8
コマンドリファレンス.....	9
互換コマンドから拡張されたコマンド .....	9
【ACCEL】加減速テーブルの作成 .....	9
【PRSET_ACCEL】アクセルパラメータの復旧 .....	9
【FEED】速度設定.....	10
【FEED】速度設定(微細設定).....	10
【HOUT】ポート出力/レジスタ設定.....	11
【HPT】入力ポート読み込み .....	11
【HOME】原点復帰 .....	12
【STOP】停止条件 .....	12
【RANGE】動作範囲制限 (MPG-68K互換) .....	13
【RANGE】動作範囲制限(MPG-314拡張).....	13
MPG-314純正コマンド .....	14
【RMVS】軸独立パルス発生 (相対座標移動) .....	14
【MOVS】軸独立 $\mu$ s発生 (絶対座標移動) .....	14
【RMVL】直線補間 (相対座標移動) .....	15
【MOVL】直線補間 (絶対座標移動) .....	15
【RMVT】連続補間 (相対座標移動) .....	16
【MOVT】連続補間(絶対座標移動).....	17
【RMVC】無限 $\mu$ s発生.....	19
【PLSC】一定速 $\mu$ s発生.....	19
【STPS】現在位置指定.....	20

【RR】レジスタの読み取り.....	21
【ERR_PAUSE】エラー発生時のタスク制御.....	22
【INCHK_314】保守コマンド.....	22
【INSET_314】入力ポート機能設定.....	22
<b>応用例.....</b>	<b>23</b>
一枚のMPG-314の軸をタスクごとに非同期で使用する場合.....	23
一枚のMPG-314の軸で三軸直線補間と残りの軸を他のタスクで制御.....	24
パルス発生 途中停止や速度変更.....	25
入力を検出してソフトで停止させる.....	25
条件により速度を変更する.....	25
原点復帰.....	25
パルス発生 途中停止 ハード.....	26
INPOSの有効 サーボオンエラーリセットなど.....	26
移動距離を規定しないパルス発生 (RMVC,PLSC).....	27
エラーの扱い方.....	28
<b>ハード リファレンス.....</b>	<b>29</b>
仕様.....	29
ハード構成.....	29
ボードアドレスの設定.....	29
ピンアサイン表.....	30
J1 エンコーダ入力.....	30
J3 パルス出力.....	30
J2 各軸リミット、アラーム入力.....	30
J4 標準入出力(原点センサ入力、インポジション入力、サーボオン出力 等).....	31
J5 電源.....	31
原点センサー接続例.....	32
LS接点、オープンコレクタ出力センサー.....	32
差動ドライバZ相インターフェース.....	33
2線式無接点センサー.....	34
S字加減速.....	35
アブソリュートエンコーダの接続.....	36
<b>付録.....</b>	<b>37</b>
MPG-314への加減速度の計算について.....	37
MPG-314 Q&A.....	39

## 概略

MPG-314は、制御用コンピュータMPC-684用のパルス発生ボードです。NOVA電子製MCX-314が搭載され4軸分の独立したPGと直線補間・円弧補間ドライバが一つシステムにまとめられたもので、多様かつ高度なパルス発生が可能です。従来のユーザにはMPG-68K/405ユーザが旧来のプログラムで最小限の修正でそのまま使用できるようになっています。

さらに、細かい制御が可能な専用コマンド(MPG314ネイティブコマンド)も用意されています。専用コマンドの中にはMCX-314に対するレジスタアクセスコマンドも含まれているため、MCX-314のすべての機能を引き出すことができます。MPG-314は同一システムに10枚まで使用することができます。ボードはPGコマンドで各タスクに割り当てて使用します。アドレスはボード前面のDSW1の値10倍して&H400を加えた値です。DSW1=2の場合は&H420です。割り当て方法は以下のとおりです。

PG &H400 : ボードアドレス&H400(DSW1=0)のMPG-314を現在のタスクで使用します。

PG &h410 15 : ボードアドレス&H410(DSW1=1)のMPG-314をタスク15で使用します。

## 免責事項

MPG-314を使用したプログラムは複雑かつ多様な制御に対応できますが、その多様性のため思わぬ誤作動を引き起こす可能性もあります。製作されたプログラムはすべて実行して動作を確認の上実用に供してください。コマンドの仕様書のみを根拠に、動作するであろうと机上で製作されたプログラムをそのまま適用しないでください。たとえば、エラー処理の回復プログラムはコーディングされただけで一回も実行されることなく実際の装置に組み込まれている場合があります大変危険です。また、コマンドには多様な引数入力についてバグ、弊害を発生させる可能性があります。このバグについてもユーザプログラムの試験実行によりバグ回避使用対応し、当方まで御指摘ください。当社ではバグの修正は責任を以って迅速に対応しますが、バグによる損害補償には応ずることができません。

## MPG-68K互換コマンド

MPG-314は従来のソフトを変更なく使用できるようにするために、ほとんどのMPG-68Kコマンドに対応しています。このためMPG-68K(405)からMPG-314に移行は簡単です。

また、追加機能としてXYUが動作中でもZ軸の制御が独立してできること、またパルス発生中でもHPT()/X(0)-U(0)のボード上のIO関数、コマンドが使えるように改善されています。

注意事項として、エラーフラグが立つような入力を加えると、エラーメッセージを表示して制御インタプリタが停止します。このため、エラー回復操作などをソフトで対応する必要がある場合は、互換コマンドを使用しないか(専用コマンドで対応)、エラー入力を接続しないでください。

## 互換性の注意

- アドレスが&H400,410,420.~&H490となっています。PG &HE0などのMPG-68K/405アドレスでは使用できません。
- MPG-314のPULSEコマンドは405互換です。このため、パルスデューティの決定はできません。
- GO,RMコマンドは同時4軸スタートのパルス発生ですが、直線補間にはなりません。MPG-314では4軸直線補間がハードの制約から不可能となっています。また、GO,RMを使用する場合は4軸のACCEL設定が同一である必要があります。
- XPLS,WPLSは使用できません。しかし、ネイティブコマンドでは同等の機能を実現できます。
- HOME,HOMZコマンドはS1側に近傍センサ、S2にZ相(C相)入力を想定しています。また、近傍センサのみで原点復帰を完結する場合は、S2側のみ使用してください。これ以外の複雑な原点復帰はMPG-314専用コマンドのHOMEを参照してください。
- MPG-405にありMPG-68Kに無いコマンド、TR,SETXやACCELの405専用入力方式は使用できません。

互換性には注意しておりますが細部に仕様の相違やバグの可能性もあります。ユーザで動作を検証の上実用に供してください。

## 互換（従来）のパルスコマンドとの関係

下表はMPG-68K/405のパルス発生に関するコマンド・関数です。

これらはMPCのパルス制御にとって基礎となるもので、MPG-314でもこれらを使用することができます。MPG-314専用コマンドもこれらの互換コマンドを踏まえた上で使用してください。例えば、単軸制御などMPG-314機能を使う場合には、専用コマンド・拡張ステートメントを使いますが、現在点の取得は表中のX～Zを用います。

### ■ conventional commands

ACCEL	.. 加減速テーブルの作成	PL1	..パレットイン
BSY	..パルス発生状態入力	PL2	..パレットイン
CLRPOS	..現在位置クリア	PL3	..パレットイン
CURPOS	..現在位置表示	PL4	..パレットイン
FEDD	..スピード設定	PULSE	..定速パルス発生
FEDH	..スピード設定	Q_PAUSE	..クイックポーズ
FEDT	..スピード設定	RANGE	..動作範囲の制限
FEDZ	..スピード設定	RM	..4軸相対座標移動 (*1)
FEED	..スピード設定	RMOV	..XYU軸相対座標移動
GO	..4軸同時パルス発生 (*1)	RMVZ	..Z軸相対座標移動
HOME	..原点復帰	SET	..インテグレーション量設定
HOMZ	..原点復帰	SETP	..点データ設定
JMPZ	..ゲートモーション移動	SETPOS	..現在位置変更
JUMP	..ゲートモーション移動	SHMZ	..原点復帰モード設定
LIMZ	..ゲートモーション規制	SHOM	..原点復帰モード設定
MOVE	..XYU絶対座標移動	STOP	..パルス発生停止
MOVZ	..Z絶対座標移動	TEACH	..ティーチモード
P	..点データ	U	..U軸点データ
PALET1	..パレット宣言	X	..X軸点データ
PALET2	..パレット宣言	Y	..Y軸点データ
PALET3	..パレット宣言	Z	..Z軸点データ
PALET4	..パレット宣言	NEWP	..点データ初期化
PG	..PG宣言	PLIST	..点データ表示
PGSEL	..PGボード選択	HOUT	..パルスボード汎用出力
		HPT	..パルスボード原点リセット入力

(\*1)MPG-314ではDDA4軸制御がハード的に不可能なため疑似的に4軸同時パルス発生をしています。

### 主なコマンドの概要

詳細は"MPG-68K ユーザーズマニュアル"をご覧ください。

#### ■ PG

タスクとMPGを引当てます。タスクで実行された命令は引当てられたMPGに対して発行されます。

PGは全てのパルス発生コマンドに先立ち実行しなければなりません。

例えば、タスク0で "PG &HE4 2" または、タスク2の中で "PG &HE4" とすれば、

タスク2で実行されるコマンドはボードアドレス&HE4に設定されたMPGに対して発行されます。

ボードアドレスはMPG-68K/405は &HE0 &HE4 --、MPG-314は &H400 &H410 -- となります。

#### ■ ACCEL

加減速テーブル（最大スピード、加減速領域パルス数、立ち上がりパルス数）を作成します。

汎用コマンドでは台形駆動です。作成されたデータはMPG側で保持しており、MOVE・RMOV等の移動命令に適用されます。

MPG-314に対しては軸選択、S字加減速パラメータが拡張されています。

## ■ FEED

ACCELコマンドで設定した加減速テーブルが富士山だとすると、その何合目まで登るかを決めるのがFEEDです。

"FEED 0" で頂上まで登ります。FEEDはACCEL設定後に実行します。

MPG-314では軸指定と速度指定の分解能が拡張されています。

## ■ MOVE

XYU絶対座標移動です。最も一般的で使用頻度の高い移動コマンドです。

"MOVE 1000 2000 3000" とすれば座標 X=1000 Y=2000 U=3000 へ移動します。

パラメータには変数・定数・ポイントデータを与えることができます。

絶対座標移動の314専用コマンドとしては "MOVL、MOVLS、MOVLT"があります。

## ■ RMOV

XYU軸相対座標移動で、MOVEと同じく使用頻度の高いコマンドです。

パラメータで与えられたパルス数だけ現在位置から移動します。

パラメータには変数・定数を与えることができます。

相対座標移動の314専用コマンドとしては "RMVL、RMVLS、RMVLT"があります。

## ■ P(n),X(n),Y(n),U(n),Z(n)

点データを設定・参照する関数です。P(n)は4軸分、X(n)~Z(n)は各軸の位置を返します。

nには1~10000の点番号または0を設定します。

0は現在位置を表すもので、"nowXpos=X(0)"などと軸ごとに現在位置を変数に読み込むことができます。

現在位置の設定は"SETPOS"です。

ティーチングポイントは"PLIST(省略形PLS)"で一覧表示します。

また、X(n)~Z(n)は配列変数としても使えます(nは1~10000)。

MPG-314に対してはn=-1でエンコーダカウンタ値を返します。

## ■ TEACH (省略形T)

FTMWでティーチングするコマンドです。キーボードのX,Y,U,Zキーでジョグ移動し、Pキーで教示します。

#T

PG[0,400] X= -150 Y= 150 U= 0 Z= 0 dx= 50 dy= 50 du= 50 dz= 50

点番号を指定して下さい P100

ティーチモードはQキーで終了します。

MPGやモータドライバの動作確認にも利用できます。

## ■ HOUT

MPG-68K/405では汎用出力の制御。

MPG-314では汎用出力、ドライバSON信号の制御のほか、レジスタ制御に用いられます。

## ■ HPT(n)

MPG-68K/405では原点センサポートの読み込み、

MPG-314では原点センサ、インポジション、アラーム入力を一括で読み込みます。

## MPG-314専用コマンド(314ネイティブコマンド)概略

MPG-314を用いて新しい装置を設計製作する場合は、ネイティブコマンドを使用してください。任意の軸の組み合わせで二軸および三軸直線補間可能です。また、単軸パルス発生も軸ごとに独立して制御できます。

ネイティブコマンドと互換コマンドのもっとも大きな相違は、ネイティブコマンドはコマンドの実行(MCX-314へのデータ設定)を終えるとただちにコマンドが抜け出すことです。このため、パルス発生中の条件停止や速度変更などをインタプリタ上で記述することができます。

```
10      PG &H410
20      ACCEL X_A 8000
30      STPS X_A 0
40      RMVS X_A 10000      /* MPG-314を設定するとただちにコマンドから出る
50      WAIT X(0)>5000      /*リアルタイムで現在位置の監視が可能
60      FEED X_A 200        /*途中で速度変更
70      TIME 1000
80      STOP X_A STP_D      /*途中停止
90      WAIT RR(X_A)==0     /*動作完了検出
100     CP
#run
  X= 7309 Y= 0 U= 0 Z= 0
#
```

### 制約

しかし以下のような使い方はハードの制約上できません。

- XY軸、ZU軸などのように二組の補間制御を同時に行うこと。同時でなければ可能です。
- 連続補間の途中で軸の組み合わせを変更すること。
- 絶対位置移動(MOVS,MOVLなど)は、軸が停止する前に実行できない。(現在位置が確定していないため)
- 移動コマンド(パルス発生)は、エラー発生時にパルス出力を停止しますが、インタプリタは実行継続となります。エラー発生時に実行をhaltするには、ERR\_PAUSEコマンドを使います。

### 互換コマンドとネイティブコマンドの混在

MPG互換モードコマンドと314ネイティブコマンド混在は可能です。MPG互換モードコマンドから、ネイティブコマンドに移る場合はインタプリタの制御を開放しないので矛盾しませんが、ネイティブコマンドから互換モードに移る場合は条件待ちが必要です。

```
move 1000 1000 0          /*このコマンドはパルス発生終了まで抜けてこない。
movl 1000 0 0 1000        /* MOVL実行問題なし

movl 10000 10000 0 0      /* パルス発生終了を待たずに次ぎの行へ
wait rr(0)&(X_A|Y_A)=0     /*この条件待ちを外すと次のmove 0 0 0がただちに実行されて誤作動する。
move 0 0 0
```

## ビット定数

MPC-684では、常用される機能のためコマンドを理解しやすいものにとりまとめていますが、細部の機能設定も可能なように、レジスタも直接アクセス可能としています。

また、レジスタ設定は多様なビット定数が必要となりますが、代表的なものを予約定数としてあらかじめ684に登録してあります。

次はレジスタ関連の関数・コマンドの使用例です。

- X軸のパルス発生終了を待つにはには WAIT RR(X\_A)=0
- X軸の発生中のパルスを停止しさせるには STOP X\_A STP\_D
- Y軸の速度を変更するには FEED Y\_A 128
- Z軸のパルス出力をパルス/方向信号方式にする HOUT MD\_DPLS (Z\_A;NOP,2)

NOP	ノーオペレーション 軸選択のレジスタ設定に使用
VOID	入力無効定数
CLR_ERR (HOUT)	Wr0 コマンドエラー解除
STP_I,STP_D (STOP)	Wr0 停止コマンド
X_A,Y_A,U_A,Z_A,ALL_A	Wr0軸選択定数 ALL_AはACCEL等で全軸
CW,CCW	円弧補間方向指定 連続補間に使用します
DS_DACL,EN_DACL	自動減速無効、有効 連続補間に使用します
IN0_ON,IN1_ON,IN2_ON,IN3_ON IN0_OFF,IN1_OFF,IN2_OFF,IN3_OFF (STOP,HOME)	wr1の設定パラメータです。 停止入力をON/OFF状態で有効にします。HOME X_A,STOP X_A のタイプのコマンドで使用します。
INP_ON,INP_OFF (INSET_314)	MCX-314/wr2の設定パラメータです。 インポジションを有効にしてON/OFFを決定します。
ALM_ON,ALM_OFF (INSET_314)	MCX-314/wr2の設定パラメータです。 アラーム入力を有効にしてON/OFFを指定します。
LMT_ON,LMT_OFF (INSET_314)	MCX-314/wr2の設定パラメータです。 リミット入力のON/OFF設定
SLMT_ON	MCX-314/wr2の設定パラメータです。 RANGEコマンドによるソフトリミットの有効化定数
MD_2PLS	wr2の設定パラメータです。CW/CCWパルス発生指定
MD_DPLS	wr2の設定パラメータです。DIR,パルス発生指定

注:()中は使用コマンド

## MCX-314の重要なレジスタとアクセス方法(参考)

レジスタの詳細は巻末の資料を参照してください。以下には概略を示します。

### ■ 読み取り用

#### RR0: パルス発生とエラーの監視

パルス発生中の監視とエラーの監視用のレジスタです。

関数RR(0)で参照することができます。下位4ビットがビジー、次の4ビットがエラーです。

以下のように使用します。エラーは下記のRR1の上位8ビットの反映です。

- パルス発生終了待ち WAIT RR(0)&X\_A=0 もしくは WAIT RR(X\_A)=0
- 他のタスクのRR0読み取り RR(&H410)

#### RR1: 軸毎のエラーステータス1

X,Y,U,Zそれぞれが持つステータスです。関数RR(軸定数,1)という形式で読み取ることができます。

このレジスタの上位8ビットはパルス発生がどのように終了したかを表します。

RR()関数で読み取りますが以下のような使用方法となります。

- X軸リミットエラーの参照 LMT=&h3000&RR(X\_A,1)

#### RR2: 軸毎のエラーステータス2

X,Y,U,Zそれぞれが持つエラー入力の参照レジスタです。ALAMやLMT信号を直接参照するにはこのレジスタを参照します。

#### RR3: 軸毎のエラーステータス3

X,Y,U,Zそれぞれが持つカウンタの比較情報です。MCX-314には軸ごとに比較用レジスタと比較器があります。これにより特定の場所で正確な信号を発生させることも可能です。

### ■ 設定用

#### WR0: コマンドレジスタ

軸選択とMCX-314に対するコマンドレジスタです。たとえば

HOUT X\_A;STP\_D

という記述では、X\_A;STP\_Dという演算によって&H0126という数値が生成されますがHOUTはこれをWR0への直接書き込みとして扱います。WR0は軸選択レジスタとしても使用されるため、以下のWR1,WR2への書き込み時にもダミーライトが必要となります。

#### WR1: モードレジスタ1

主にIN0~IN3のドライブ停止検出信号の設定として使用します。WR1への書き込みは次のようなフォーマットになります。

HOUT 2 (X\_A;NOP,1)

2はWR2に書き込む値でIN0がLOWで停止という条件です。()の中の二番目の引数X\_A;NOPでWR0に対して軸設定をします。、の後ろの番号がレジスタ番号です。レジスタ番号にボードアドレスを加えると任意のアドレスのMPG-314にアクセスすることができます。

#### WR2: モードレジスタ2

リミット検出やインポジション検出の設定を行うレジスタです。しかしながらこのレジスタにはパルス出力モードの設定ビットも含まれていますので注意して再設定してください。

HOUT INP\_ON+MD\_2PLS (X\_A;NOP,2)

ここではINP\_ON+MD\_2PLSをWR2に設定する値としています。MD\_2PLSはパルス出力モードをCW/CCW負論理方式(デフォルト)に設定するビット定数ですが、もしもこの値を加えておかないとパルス出力のモード設定フィルードがすべて0となり、パルス出力が正論理に変わってしまいます。(X\_A;NOP,2)はWR1の場合と同様の入力でX軸選択でWR2を選んでいきます。

#### WR3: モードレジスタ3

下位3ビットが加減速度のモード設定となります。初期値としてこのレジスタは4(自動減速、対称減速、S字加減速度有効)が設定されています。このほかにはOUT4~7の出力モード設定があり汎用出力として使用するか、位置信号の出力として使うかを決定できます。



## コマンドリファレンス

### 互換コマンドから拡張されたコマンド

#### 【ACCEL】加減速テーブルの作成

---

##### 書式

ACCEL [n] [s] max [long min]

n:軸選択予約定数 X\_A~Z\_A もしくは論理和

s:S字加減速パラメータ -1~100 (%です。台形領域の指定%をS字領域とします。)

max:最大スピード 1~4Mpps(円弧2M)

long:加減速領域パルス数

min:立ち上がりパルス数

##### 解説

ACCELコマンドはこれまでのmax,long,minパラメータのほかに軸選択引数、S字加減速の設定を追加することができます。軸設定の値はX\_A~Z\_Aの軸定数、S字は負の値で-1~100を設定します。

##### 例)

ACCEL U\_A 10000

軸指定はU、最大速度10Kpps

なお、軸指定をしないとACCELは全軸に対する加速度、速度設定となります。また、S字値は設定しなければ台形制御となります。また、設定されたパラメータを参照するには、引数無しでACCELを実行します。

ACCEL X\_A|U\_A -50 10000

XU軸を同時に設定します。また加速領域の50%をS字加減速とします。

##### 注意

minは1ppsから指定可能ですが微小距離移動等で著しく遅くなる場合があるのでむやみに10pps以下に設定しないで下さい。

#### 【PRSET\_ACCEL】アクセルパラメータの復旧

---

##### 書式

PRSET\_ACCEL n

n:軸指定予約定数 X\_A~Z\_A

##### 解説

ppsを指定する一部のコマンド(PLSC,HOME X\_A)ではACCELで設定したパラメータを破壊します。これを復旧するのがPRSET\_ACCELです。引数は軸指定定数のみです。

PRSET\_ACCELはIN0~IN3停止条件もクリアします。

##### 例)

PRSET\_ACCEL X\_A            X軸のパラメータ復旧

PRSET\_ACCEL Z\_A U\_A        Z,U軸のパラメータ復旧

## 【FEED】速度設定

---

### 書式

FEED n m

n:軸指定予約定数 X\_A~Z\_A もしくは 論理和

m:速度指定 0~255

### 解説

ACCELで設定された最高速に対して256段階で速度段階を指定。

### 例)

FEED Z\_A 128

Z軸の速度を最高速の半分にします。このコマンドはパルス発生中でも有効です。軸選択パラメータを指定しない場合は、FEEDは互換コマンドとなりXYUに対しての速度設定となります。

次のような連続設定も可です。

FEED Z\_A 0 X\_A 32

FEED Z\_A|X\_A 0

注意:S字を指定していると行き先の決まっているパルス発生では増速が無効になります。

## 【FEED】速度設定(微細設定)

---

### 書式

FEED n VOID|m

n:軸指定予約定数 X\_A~Z\_A

m:速度指定 1(最低)~8000(最大)

### 解説

速度指定に予約定数VOIDをorすると設定範囲は1~8000となり、MCX314の速度レジスタに直接書き込みます。ACCELの最高速度が8000以下の場合は指定値がそのままppsになります。ACCELの指定値が8000を超える場合は最高速に対して8000等分されたppsになります。

### 例)

ACCEL 50000 /\* 最高速8000超

FEED X\_A VOID|4000 /\*この場合 25000pps=50000/8000\*4000

## 【HOUT】ポート出力レジスタ設定

---

### ポート出力

書式

HOUT p

p:出力データ 1~255

解説

HOUTを1~255の範囲の引数で使用するとOP1~4,SON\_X~SON\_Zの平行設定となります。  
下位がOP1~4の設定で従来のコマンドと互換性があります。

### レジスタ設定

書式

HOUT n;cmnd

n:軸選択予約定数 X\_A~Z\_A

cmnd:コマンド予約定数

解説

レジスタWR0を設定します。

例)

HOUT X\_A;STP\_D

X軸に対して減速停止命令を実行します。(STOP X\_A STP\_Dと同じ)

HOUT Y\_A;CLR\_ERR

Y軸のエラービットをクリアします。

## 【HPT】入力ポート読み込み

---

書式

HPT(0)

解説

MPG-314の入力を読み込みます。

例)

#prx hpt(0)

00010000 (H7 H0 右側)

返す値を8桁のヘキサで表現すると以下の入力ポートの値がそれぞれの桁に反映されます。

H0: XS1, XS2(XIN0, XIN1), YS1, YS2(YIN0, YIN1) (コネクタJ4)

H1: US1, US2(UIN0, UIN1), ZS1, ZS2(ZIN0, ZIN1) (コネクタJ4)

H2: XIN2~ZIN2 (差動入力ポート) (コネクタJ1)

H3: XIN3~ZIN3 (コネクタJ2)

H4: XINP~ZINP (コネクタJ4)

H5: X\_ALM~Z\_ALM (コネクタJ2)

## 【HOME】原点復帰

---

### 定則低速原点復帰

書式

HOME n rate cond [n1 rate1 cond]  
n:軸指定予約定数 X\_A~Z\_A  
rate:原点復帰速度 (pps)  
cond:入力指定 INx\_ON~INx\_OFFの論理和

解説

原点復帰コマンドの最初の引数に軸指定定数を設定すると1軸ごとの原点復帰コマンドになります。引数にパルスレートとIN0-IN3のどの入力の信号を使用するかを決定する値を設定できます。条件はIN0\_ON~N3\_OFFが用意されており和をとることにより複数の信号により停止させることができます。なお、原点復帰はACCELで設定したパラメータを破壊するため、PRSET\_ACCELでパラメータ復帰してください。また、座標値のクリア、プリセットは行わないのでSTPSで別途実施します。

例)

HOME X\_A 1000 IN0\_ON Y\_A -1000 IN1\_ON  
X軸とY軸をそれぞれCW1kpps/CCW1kppsでIN0オンになるまでY軸はIN1がONになるまでパルス発生します。

### 高速加減速原点復帰

書式

HOME n cond ±F [n1 cond1 ±F1]  
n:軸指定予約定数 X\_A~Z\_A  
cond:入力指定 INx\_ON~INx\_OFFの論理和  
F:feed 1~255

解説

原点復帰コマンドの最初の引数に軸指定定数を設定し次に停止条件を指定すると1軸ごとの高速原点復帰コマンドになります。引数にパルスレートとIN0-IN3のどの入力の信号を使用するかを決定する値を設定できます。条件はIN0\_ON~N3\_OFFが用意されており論理和をとることにより複数の信号により停止させることができます。

注意

原点復帰時に指定した停止条件は原点復帰後も有効となるためSTOP n VOIDで条件をクリアして下さい。

## 【STOP】停止条件

---

書式

STOP n cond  
n:軸指定予約定数 X\_A~Z\_A  
cond:入力指定 INx\_ON~INx\_OFFの論理和 または STP\_I,STP\_D

解説

指定した軸のIN0-IN3の条件を規定します。  
INxを指定場合、STOP n VOIDで解除しないぎり有効になり続けます。

例)

STOP Z\_A IN0\_ON|IN3\_ON  
Z軸でIN0オン,IN3オンになったら停止するという条件を設定します。一度設定するとSTOP Z\_A VOIDを実行するまで条件は解除されません。  
また、condにSTP\_I,STP\_Dをセットすると発生中のパルスを途中停止します。軸指定nを軸定数の論理和とすれば指定軸を同時に停止させることができます。

STOP X\_A|Y\_A STP\_D          XY同時減速停止。

## 【RANGE】動作範囲制限 (MPG-68K互換)

---

### 書式

RANGE n

n=0,1,2,11,12

n=0 領域設定を解除します。

n=1 現在位置を最大領域として設定します。

n=2 現在位置を最少領域として設定します。

n=11 最大領域の値を現在位置に複写します。現在位置が失われるので注意

n=12 最少領域の値を現在位置に複写します。現在位置が失われるので注意

### 解説

RANGEコマンドはMPG-68K互換コマンドです。このコマンドにより自動的にSLMT\_ONが有効になります。次のプログラムの110行でSLMT\_ONをorしているのは、INSET\_314で入力条件を設定した時RANGE有効状態を保つためです。(INSET\_314は指定したパラメータ以外の設定はリセットします)

### 例)

```
10      SETP 1 1000 500 400 200
20      SETP 2 -1000 -500 -400 -200
30      PG &H410
40      ACCEL 8000
50      SETPOS X(1) Y(1) U(1) Z(1)
60      RANGE 1
70      SETPOS X(2) Y(2) U(2) Z(2)
80      RANGE 2
90      CLRPOS
100     CP
110     INSET_314 ALL_A INP_OFF|SLMT_ON
```

また、RANGEによって停止したかどうかは、RR()関数とSLMP@(正方向リミット)SLMM@(負方向リミット)を組み合わせて検出することができます。

### 例)

```
RR(X_A,SLMP@|SLMM@)
RR(Y_A,SLMP@|SLMM@)
RR(U_A,SLMP@|SLMM@)
RR(Z_A,SLMP@|SLMM@)
```

## 【RANGE】動作範囲制限(MPG-314拡張)

---

### 書式

RANGE n max min

n:軸選択予約定数 X\_A~Z\_A

max:動作範囲上限

min:動作範囲下限

### 解説

互換RANGEコマンドは暗黙でSLMT\_ON(ソフトリミット)を有効にしますが、拡張RANGEを使うためにはINSET\_314コマンドで明示的にSLMT\_ONを有効します。

### 例)

```
RANGE X_A 1000 -1000
INSET_314 X_A SLMT_ON
```

## MPG-314純正コマンド

### 【RMVS】軸独立パルス発生（相対座標移動）

---

書式

#### 単軸指定

RMVS n pls  
n:軸指定予約定数 X\_A~Z\_A  
pls:パルス数

#### 複数軸指定

RMVS x y u z  
x y u z:各軸パルス

解説

パルス発生コマンドです。速度加速度等はACCELの設定に従います。入力方法には単軸指定、複数軸指定の二通りあります。

RMVSは直線補間はしません。

例)

RMVS X\_A 1000                    X軸指定でCW1000パルス発生します。  
RMVS 1000 0 2000 0            X軸をCW1000パルス、U軸CW2000パルス発生します。

### 【MOVS】軸独立パルス発生（絶対座標移動）

---

書式

#### 単軸指定

MOVS n c  
n:軸指定予約定数 X\_A~Z\_A  
c:座標値

#### 複数軸指定

MOVS x y u x  
x y u x:各軸座標値。指定無=VOID予約定数

解説

パルス発生コマンドです。速度加速度等はACCELの設定に従います。入力方法には単軸指定、複数軸指定の二通りあります。複数軸指定でパルス発生しない軸にはVOID定数をセットします。絶対座標移動は現在位置との比較のため、直前のパルス発生が停止している必要があります。

MOVSは直線補間はしません。

例)

MOVS X\_A 10000                    X軸を10000の位置までパルス発生  
MOVS 10000 VOID VOID VOID      X軸を10000の位置までパルス発生。yuzはVOIDで無効。

実際には次のように移動させる軸の停止を確認して実行

```
WAIT RR(Y_A)==0  
MOVS Y_A 10000
```

## 【RMVL】直線補間（相対座標移動）

---

書式

RMVL x y u z

x y u z:各軸パルス (最大3軸まで)

解説

直線補間パルス発生コマンドです。RMVL x y u zとしますが、4軸直線補間はハード的に不可能なので0以外の値を指定できる軸は3軸までです。直線補間の場合の速度加速度等はACCELの設定に従いますが、各軸が異なる値を持つ場合は”主軸”の値が用いられます。主軸とは、X>Y>U>Zの優先順位で選択されます。たとえばYUZの組み合わせの場合はY軸が主軸となります。(一般的なDDA演算の主軸の意味とは異なります)。

例)

RMVL 1000 2000 0 0                   XY軸で直線補間パルス発生

RMVL 0 2000 2000 200               YUZ軸で直線補間パルス発生

## 【MOVL】直線補間（絶対座標移動）

---

書式

MOVL x y u z

x y u z:各軸座標時。パルス発生無=VOID

解説

直線補間パルス発生コマンドです。MOVL x y u zとしますが、現在位置との相対値を算出してパルス発生しますので、MOVSと同様現在位置が確定していることと、実際の直線補間軸が何軸になるか注意が必要です。パルス発生させない軸にはVOIDを入力しておきます。

たとえば現在位置が100 200 300 400でMOVL 200 300 400 500とするとそれぞれの軸で100パルスずつの出力が必要となり結果4軸直線補間となりエラーとなります。しかし、MOVL 200 300 400 400という入力の場合にはZ軸の移動量が0となり三軸直線補間となり実行可能です。相対値を計算してからのパルス発生はRMVLと同じ扱いとなります。

例)

MOVL 1000 2000 3000 VOID           /\* Z軸を除いた直線補間パルス発生

直線補間連続の場合は直前コマンドのパルス発生終了待ちを省略できます。

40 RMVL 10000 2000 0 0               /\* 相対座標移動の直線補間

50 MOVL 1000 2000 VOID VOID       /\* 絶対座標移動の直線補間

## 【RMVT】連続補間（相対座標移動）

### 直線補間

#### 書式

RMVT n p1 p2

n:軸指定予約定数。2軸の論理和

p1,p2:移動先の相対座標

### 円弧補間

#### 書式

RMVT n p1 p2 <CW,CCW> cnt1 cnt2

n:軸指定予約定数。2軸の論理和

p1,p2:移動先の相対座標

CW,CCW:円弧補間方向指定定数

cnt1,cnt2:円中心

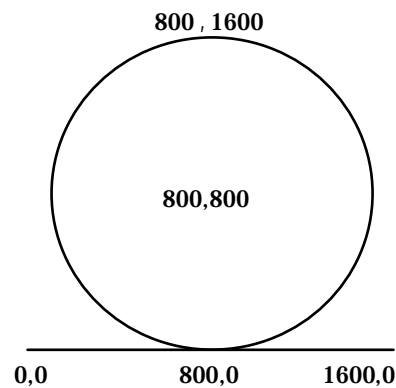
#### 解説

連続移動補間コマンドです。軌跡制御に使用します。

入力フォーマットはRMVT n pls1 pls2もしくはRMVT n pls1 pls2 <CW,CCW> cnt1 cnt2です。nは軸定数で二つの軸定数の和を使用します。pls1,pls2は行き先の相対座標です。CWもしくはCCW定数に続いて円中心を相対位置で与えると目的位置まで円弧移動します。連続補間の途中で軸定数を変更することはできません。

#### 例)

```
10 PG &H410
20 ACCEL 4000
25 HOUT X_A;DS_DACL
30 Aset=X_A|Y_A
40 RMVT Aset 800 0
50 RMVT Aset 0 1600 CCW 0 800
60 RMVT Aset 0 -1600 CCW 0 -800
70 RMVT Aset 800 0
80 HOUT X_A;EN_DACL
```



RMVTコマンドはパルス発生中に次のコマンドを設定することができます。このため、コマンドの切れ目でパルス出力が途絶えることはありません。25のステートメントは通常のパルス発生で自動減速を有効としているのを解除するものです。この設定を略すと最初のコマンドの行き先に到達する時に減速してしまいのこりのパルス発生がすべて最低速となります。

逆に、80のステートメントは減速機能を有効にして最後の目的位置に到達する時に減速停止させるものです。



## 【MOV T】連続補間(絶対座標移動)

### 書式

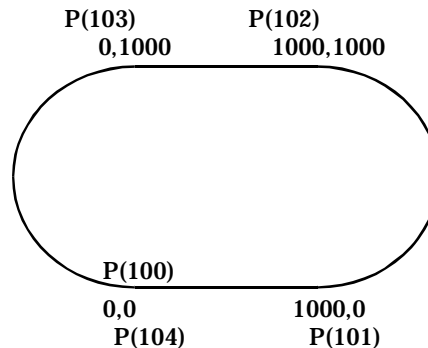
MOV T n point [<CW,CCW>]  
n:軸指定予約定数。2軸の論理和  
point:P(n)形式で点データを指定  
CWまたはCCW:円弧補間方向

### 解説

RMVTの座標位置指定連続補間です。RMVTではすべて相対位置指定となるために、実際のデータ入力が大変困難です。MOV Tはあらかじめ座標値を点データに指定しておくことにより所定の軌跡制御を実行することができます。座標値は移動する順序にしたがって連続で登録する必要があります。これは、MOV Tコマンドは指定された一つ前との点との差を計算し、その値をRMVTとして実行するためです。点データは通常X,Y,U,Zの値となりますが、MOV Tでは第一軸、第二軸、第一軸円弧中心、第二軸円弧中心の順に入力します。軸定数はかならず和で表記してください。

### 例)

```
10 SETP 100 0 0 0 0
20 SETP 101 1000 0 0 0
30 SETP 102 1000 1000 1000 500
40 SETP 103 0 1000 0 0
50 SETP 104 0 0 0 500
60 PG &H410
70 ACCEL 2000
80 STPS 0 VOID 0 VOID
90 HOUT X_A;DS_DACL
100 MOV T X_A|U_A P(101)
110 MOV T X_A|U_A P(102) CCW
120 MOV T X_A|U_A P(103)
130 MOV T X_A|U_A P(104) CCW
140 MOV T X_A|U_A P(101)
150 HOUT X_A;EN_DACL
#
```



たとえば、110の MOV T X\_A|U\_A P(102) CCWでは、点102を指定します。X\_Aの移動量としてはX(102)-X(101),U\_Aの移動量としてはY(102)-Y(101)が使用されます。CCWは円弧指定のためP(102)へ至る経路は円弧補間となり、その円弧中心はU(102)-X(101),Z(102)-Y(101)となります。ここではXU軸の円弧補間を含む軌跡制御ですが、座標のXYUZというラベルと適用される軸は無関係なことに留意してください。

以下は同様の動作を実際のプログラムに応用したものです。点3999を図面上の基準点として4000~4005の各点を設定しています。このプログラムではD(0)~D(6)で動作モードを設定しておいてサブルーチン\*UT1\_dis1で実際の動作を行います。

最初の MOV S 1000 1000 0 0は待機点で任意の点です。次のMOV Tで最初に点4000へ移動することになっているので点3999が図面上の基準点になります(MOV Tは1つ前の点からの相対移動になるためです)。点番号に相当する配列に0セットすると直線、1をセットするとCW円弧補間、2をセットするとCCW円弧補間、3で終端としています。このようにMOV Tでは図面上の点を直接扱うことができます。

また、画像処理で補正する場合には、あらかじめAFFIN変換により各点を回転補正した上で作業点のXY座標を修正すれば良いこととなります。

以下のプログラムではMOV TとMOV Tの間でいくつかのステートメントが実行されます。このため、点と点の間があまりに短いと、MOV Tの連続実行が間に合わなくなる場合がありますので、注意してください。その場合、位置制御そのものには矛盾が生じませんが、パルスが途切れるために機械系にストレスを与えます。

ここでは、HOUT X\_A;DS\_DACL,HOUT X\_A;EN\_DACLという特殊なコマンドが使われています。

それぞれの意味は、減速を無効、減速を有効にするというものです。

減速を無効にしておかないと通常のMOV S等と同様、最初のMOV Tの終点で減速してしまうために以後のパルス発生が低速になってしまいます。しかし、終端では逆に減速を有効にしておかないと急停止になってしまいますので最後のMOV Tのあとで減速を有効にします。MPG-314ではトレース移動を動的に決定できるために始点と終端をICに知らせることが必要です。

```

DIM D(100)
SETP 3999 0 0 0 0
SETP 4000 1000 0 0 0
SETP 4001 2000 0 0 0
SETP 4002 2000 1000 2000 500
SETP 4003 1000 1000 0 0
SETP 4004 0 1000 0
SETP 4005 0 0 0 500
SETP 4006 0 0 0 0
SETP 4007 0 0 0 0
  D(0)=0
  D(1)=0
  D(2)=2
  D(3)=0
  D(4)=0
  D(5)=2
  D(6)=3
  D(7)=3
PG &H410
ACCEL 2000
CLRPOS
MOVS 1000 1000 0 0
WAIT RR(ALL_A)==0
TIME 1000
CP
PRINT "START"
GOSUB *UT1_dis1
WAIT RR(ALL_A)==0
CP
END
*UT1_dis1
HOUT X_A;DS_DACL
FOR Nans3=4000 TO 4999
  PRINT Nans3
  Turn=D(Nans3-4000)
  SELECT_CASE Turn
    CASE 0 : MOVT X_A|Y_A P(Nans3)
    CASE 1 : MOVT X_A|Y_A P(Nans3) CW
    CASE 2 : MOVT X_A|Y_A P(Nans3) CCW
    CASE_ELSE : GOTO *UT1_dis2
  END_SELECT
NEXT Nans3
*UT1_dis2
HOUT X_A;EN_DACL
RETURN

```

## 【RMVC】無限パルス発生

---

### 書式

RMVC n m [n1,m1,n2,m2]  
n:軸指定予約定数  
m:方向指定 +1または-1

### 解説

移動量を指定しない無限パルス発生です。加減速はACCEL、速度はFEED指定に従います。nは軸指定、mは+1,-1で方向指定のみ。引数を続けて与えて連続で起動することもできます。通常ニアオリジンなどへ高速復帰する場合などに用います。

### 例)

X\_IN0をニアオリジンと想定して加減速付き高速移動でパルス発生します。

```
10 PG &H410
20 ACCEL 4000
30 HOUT 2 (X_A;NOP,1)
40 RMVC X_A -1
45 WAIT RR(X_A)==0
50 HOUT 0 (X_A;NOP,1)
#
```

## 【PLSC】一定速パルス発生

---

### 書式

PLSC n m [pls] [n1,m1,pls1]  
n:軸指定予約定数  
m:パルスレート(pps)  
pls:パルス量

### 解説

一定速パルス発生です。加減速はありません。引数の与え方で定量パルス発生にも、無限パルス発生にも使用できます。注意として指定パルスレート値によってMCX-314上のACCELパラメータが破壊されます。PLSC実行後に通常のパルス発生コマンドを使用する場合は、**PRSET\_ACCEL**コマンドを実行します。

移動量を規定しない無限パルス発生ではplsを省略します。この場合のCW/CCW方向規定はmを正の数とするか負の数にするかによって決定されます。

移動量を規定する場合はplsの値の正負によって方向が決定されます。

下の例では、X軸を一定速パルス発生してXS2(IN1)入力で停止させています。停止後、PRSET\_ACCELによってACCEL設定を回復し通常のRMVSを使用しています。PRSET\_ACCELがないとRMVSでのパルス発生も加減速無し的一定速となります。

### 例)

```
10 PG &H410
20 ACCEL 10000
40 STOP X_A IN1_ON
60 PLSC X_A -100
70 WAIT RR(X_A)==0
80 PRSET_ACCEL X_A
90 RMVS X_A 1000
```

## 【STPS】現在位置指定

---

書式

STPS n pos

n:軸指定予約定数

pos:指定する位置

STPS x y u z

x y u z:各軸の指定する位置

指定無=VOID

解説

STPSはSETPOSとよく似た働きをしますが機能が拡張されています。

例)

軸定数により選択した軸のみ現在値を設定

STPS U\_A 1000                      U軸を1000にする。

VOID定数以外の軸を設定

STPS VOID 1000 VOID 2000

X軸とU軸を保留にしてYZを設定する。

また、パルス発生中の軸に値を設定しようとすると停止を待ちます。

## 【RR】レジスタの読み取り

書式

### RR0を読み取る

RR(0)

### パルス終了確認

RR(n) n:軸指定予約定数。指定軸がパルス終了すると0を返す

### ボードを指定してRR0を読み取る

RR(adrs) adrs:MPG-314ボードアドレス

### 軸とレジスタを指定して読み取る (RR0以外の読み取り)

RR(n,a)

n:軸指定予約定数

a:レジスタ番号。オフセットアドレスを加えることによりMPGのアドレス指定が可能

解説

レジスタを読み込むことにより、パルス発生状態やエラー内容を知ることができます。

例)

```
WAIT RR(X_A)=0           X軸パルス発生終了待ち
A=RR(&H410)&&HF0         &H410のMPG-314のエラーステータスを読み取る
A=RR(Y_A,1)             Y軸のRR1を読み取る。(終了ステータスの参照)
A=RR(Y_A,1+&H410)      &H410のY軸のRR1を読み取る。(終了ステータスの参照)
```

RR()関数に状態定数(EMG@,ALM@,LMTP@,LMTM@,ALLI@,SLMP@,SLMM@)や停止条件定数(EMG\_,ALM\_,LMTP\_,LMTM\_,IN3\_,IN2\_,IN1\_,IN0\_,ALLE\_)をセットすることにより各入力状態の検出、またパルス発生が停止した原因を特定することができます。@の付いた定数グループは状態検出です。たとえばALM入力は論理設定でON/OFFいずれかの入力でアラーム受付となりますが、この条件が成立している時にRR(X\_A,ALM@)で検出することができます。

これに対してALM\_はパルス発生がアラーム入力によって停止したかどうかを判断するためのものです。RR(X\_A,ALM\_)が0であればアラーム入力によって停止していないということです。

RR(X\_A,ALM\_|LMTP@)のように状態検出は論理和をとって検出することができますが、状態入力グループと停止条件グループを混在させることはできません。

例)

```
PRX RR(X_A,ALM_|LMTP_|LMTM@)
PRX RR(X_A,ALM_|LMTP_|LMTM_|IN3_)
```

予約定数	状態監視グループ	EMG@	EMGがイネーブル検出
		ALM@	サーボアラーム出力ON検出
		LMTP@	リミット入力(+)検出
		LMTM@	リミット入力(-)検出
		SLMP@	ソフトリミット(+)検出
		SLMM@	ソフトリミット(-)検出
	停止条件グループ	EMG_	EMG入力によって停止
		ALM_	アラーム入力によって停止
		LMTP_	リミット入力によって停止
		LMTM_	リミット入力によって停止
		IN3_	STOP条件のIN3によって停止
		IN2_	STOP条件のIN2によって停止
		IN1_	STOP条件のIN1によって停止
		IN0_	STOP条件のIN0によって停止

## 【ERR\_PAUSE】エラー発生時のタスク制御

---

書式

ERR\_PAUSE n

n:軸指定予約定数。複数監視する場合は論理和をとる

解説

エラーフラグを検出するとタスクはポーズします。ただしメインタスク(タスク0)はエラーメッセージを出力して停止します。エラーのクリアはHOUT X\_A;CLR\_ERR &H400というように行います。

例) ERR\_PAUSE X\_A|Z\_A    XとZのエラーを監視する。

## 【INCHK\_314】保守コマンド

---

書式

INCHK\_314

解説

INCHK\_314はMPG-314の入力ポートをモニタする関数です。標準IOに対するIOコマンドと同等の機能を持ちます。

## 【INSET\_314】入力ポート機能設定

---

書式

INSET\_314 n cond

n:軸指定予約定数

cond:入力指定

解説

入力ポートの機能を設定します。

例)

INSET\_314 X\_A ALM\_ON|INP\_ON

X軸に対して入力設定します。アラームを有効にしてON状態をアラームとします。また、インポジションを有効にしてONでインポーズ成立とします。

INSET\_314 ALL\_A ALM\_ON|INP\_OFF

すべての軸に対して入力設定します。アラームはONで有効、INPOSはOFFで有効とします。

## 応用例

### 一枚のMPG-314の軸をタスクごとに非同期で使用する場合

MPG-68K/405では1枚のMPGに対して複数のタスクから同時にパルスコマンドを実行できませんがMPG-314では可能です。次のプログラムは1枚のMPGの各軸を別々のタスクで制御しています。

MPG-68K/405でこのような制御を行おうとすると軸数分のMPGが必要だったので省スペース、低コストになります。

```
PG &H410 1          /*複数のタスクを一枚のMPGに割り当てます。
PG &H410 2
PG &H410 3
PG &H410 4
FORK 1 *MPG_x      /*それぞれのタスクを起動します。
FORK 2 *MPG_y
FORK 3 *MPG_u
FORK 4 *MPG_z
END
*MPG_x
ACCEL X_A 8000     /*加速度を軸毎に選択します。
STPS X_A 0
DO
  GOSUB *PULSE X_A /*タスク間共通サブルーチンをコールしていますが、
LOOP              /*例文を簡略化するためです。
*MPG_y
ACCEL Y_A 6000
STPS Y_A 0
DO
  GOSUB *PULSE Y_A
LOOP
*MPG_u
ACCEL U_A 4000
STPS U_A 0
DO
  GOSUB *PULSE U_A
LOOP
*MPG_z
ACCEL Z_A 2000
STPS Z_A 0
DO
  GOSUB *PULSE Z_A
LOOP
*PULSE            /*タスク間で共通のルーチンです。
  _VAR axs!      /*そのためにローカル変数を用いています。
  FOR i!=0 TO 10 /*単軸制御は例のように軸選択定数を用いて制御するのが安全です。
    FEED axs! i!*20
    RMVS axs! 1000
  NEXT i!
  WAIT RR(axs!)==0
  MOVS axs! 0
  RETURN
```

## 一枚のMPG-314の軸で三軸直線補間と残りの軸を他のタスクで制御

以下のように直線補間と単軸駆動を組み合わせて非同期で使用できますが、XY,ZUように二軸直線補間を二組同時に使用することはできません。

```
PG &H410 1 /*二つのタスクを一つのMPGに割り当てます
PG &H410 2
FORK 1 *MPG_xyz
FORK 4 *MPG_z
END
*MPG_xyz
ACCEL X_A 8000 /*主軸(X>Y>Z>U)のACCELを設定
STPS 0 0 0 VOID /*他のタスクが使う軸はVOID変数で扱わないようにする。
DO
  FOR i=1 TO 10
    RMLV i*1000 i*100 i*-10 0 /* RMLV,MOVLは直線補間
  NEXT I
  WAIT RR(X_A)==0
  MOVL 0 0 0 VOID
LOOP
*MPG_z /* Z軸の単独制御
ACCEL Z_A 2000
STPS Z_A 0
DO
  GOSUB *PULSE Z_A
LOOP
*PULSE
_VAR axs!
FOR i!=0 TO 10
  FEED axs! i!*20
  RMVS axs! 1000
NEXT i!
WAIT RR(axs!)==0
FEED axs! 0
MOVS axs! 0
RETURN
```



## パルス発生 途中停止や速度変更

### 入力を検出してソフトで停止させる

MPG-314ネイティブコマンドは実行後すぐに次のステップに移行します。次はパルス発生しながらセンサがオンするのを待ちます。このようにパルス発生とその後の管理が同一タスクで行えるのでプログラムの見通しがよくなります。

```
*Y
RMVS Y_A 200000          /*相対座標移動
WAIT HPT(1)=1          /*センサ入力待ち
STOP Y_A STP_I         /*STP_Iで即停止、STP_Dで減速停止です。
WAIT HPT(1)=0
MOVS Y_A 0
WAIT RR(0)&Y_A=0
TIME 500
GOTO *Y
```

### 条件により速度を変更する

行き先の決まっているパルス発生では減速しかできません。RMVCでは加速変更も可能です。

```
PG &H410
ACCEL 8000
CLRPOS
RMVS X_A 200000
WAIT X(0)>10000
FEED X_A 128           /*速度を途中で変更する。
WAIT X(0)>20000
STOP X_A STP_D
```

### 原点復帰

原点復帰はHOMEコマンドを使用すると明示的になりますが、現在位置については放置されます。原点復帰が完了したら位置をクリア、あるいは設定する必要があります。

```
PG &H410
ACCEL 8000
HOME X_A IN0_ON -125 Z_A IN0_ON -200    /*速度指定原点復帰 IN0使用 FEED 125 CCW FEED
200 CCW
WAIT RR(X_A|Z_A)==0
HOME X_A -1000 IN1_ON Z_A -500 IN1_ON    /*定則原点復帰 IN1使用
WAIT RR(X_A|Z_A)==0
PRSET_ACCEL X_A Z_A          /* ACCELパラメータ復帰 停止条件クリア
FEED X_A 0 Z_A 0            /* HOMEで変更したFEEDの変更
STPS 1000 VOID VOID 1000    /*位置設定
```

## パルス発生 途中停止 ハード

MCX-314には入力ポートIN0-IN3を使用した停止条件を決定するレジスタWR1があります。このレジスタに適切な値を設定することにより、各軸ごとに停止条件を定めることができます。IN0,IN1はS1,S2に割り当てられ、IN2は差動入力ポート、IN3は拡張入力ポートに割り当てられています。これにより必要に応じてハード停止させることができるためホスト側の遅れ時間を気にすることなく確実にパルス発生を停止させることができます。このWR1を設定するためにSTOP拡張コマンドとIN0\_ON~IN3~OFF定数が用意されています。

```
PG &H410
ACCEL 10000
STPS X_A 0
STOP X_A IN3_ON|IN1_OFF          /* IN1がOFFになるかIN3がオンになると停止
RMVS X_A 1000000
STOP X_A VOID                    /*停止条件の解除
```

## INPOSの有効 サーボオンエラーリセットなど

INPOSを有効にするとサーボのINPOS信号と連動させることができます。INPOSの設定はWR2レジスタで行います。レジスタの意味は下記のとおりです。なおこのレジスタを操作するのに必要な定数セットが用意されています。

```
MD_2PLS デフォルトのCW/CCW負論理パルス方式です。
MD_DPLS DIR/PULSE方式です。
--
INP_ON   インポジションをON入力待ちとします。
INP_OFF  インポジションをOFF入力待ちとします。
--
ALM_ON   アラームをONで発生とします。
ALM_OFF  アラームをOFFで発生とします。
--
LMT_ON   リミット信号をONで発生とします。
LMT_OFF  リミット信号をOFFで発生とします。
```

例) X軸のインポジションをONで有効、ALMをONで有効、LMTをONで検出とするには  
wr2=MD\_2PLS|INP\_ON|ALM\_ON|LMT\_ON  
hout wr2 (X\_A;NOP,2)

wr2を設定する場合はMD\_2PLSをかならず加えて設定しなおしてください。また、パルスDIR/PULSE方式にするにはMD\_2PLSの代わりにMD\_DPLSを使います。

例) Z軸のINPOSを有効にする

```
10 PG &H410
20 ACCEL 10000
30 CLRPOS
35 HOUT MD_2PLS|INP_ON (Z_A;NOP,2)
40 DO
50 RMVS Z_A 10000
60 RMVS Z_A -10000
70 LOOP
```

## 移動距離を規定しないパルス発生 (RMVC,PLSC)

原点復帰やテープの定速送りでパルス量を規定しないパルス発生があります。この場合はRMVC,PLSCを  
用います。RMVCはACCELの指定に従います。減速停止と組み合わせて以下のように使用できます。また  
原点復帰の場合座標値のクリアが必要ですが、STPS X\_A 0、STPS VOID 0 0 VOIDなどのように実施しま  
す。

```
PG &H410
ACCEL 10000
FEED X_A 64 : FEED Z_A 32
RMVC Z_A 1 X_A -1      /* ZをCW,XをCCWで回転。ACCELに従って加速、FEEDに従った速度
TIME 1000
STOP Z_A STP_D        /* Zを減速停止
TIME 1000
STOP X_A STP_I        /* Xを急停止
```

次はIN0-IN3を用いた例です。

```
PG &H410
ACCEL 10000
FEED X_A 64 : FEED Z_A 32
STOP X_A IN0_ON       /* X軸停止設定XS1
STOP Z_A IN0_ON       /* Z軸停止設定ZS1
RMVC Z_A 1 X_A -1     /* XS1(IN0)検出減速停止
WAIT RR(X_A|Z_A)==0   /*停止待ち
STOP X_A VOID         /*停止設定解除
STOP Z_A VOID         /*停止設定解除
```

RMVCの代わりに PLSC を用いると加減速なしの定速パルス発生になり、ppsを直接指定できます。しか  
し、PLSCコマンドはACCELの設定を壊してしまうため、実行後RSTRコマンドでACCEL設定値を戻す必  
要があります。RSTRには設定された停止条件を解除する機能も含まれています。

```
PG &H410
ACCEL 10000
STOP X_A IN0_ON
STOP Z_A IN0_ON
PLSC Z_A 100 X_A -100 /*100pps指定
WAIT RR(X_A|Z_A)==0   /*停止待ち
PRSET_ACCEL X_A Z_A   /*パラメータ復帰。また停止条件解除
```

## エラーの扱い方

ERR\_PAUSEを使用すると発生したエラーに対応してタスクを自動停止させることができます。PG出力でエラーとなった場合、パルス発生そのものはされませんが、インタプリタは動作しつづけます。このため、アプリケーションによっては装置を誤作動させてしまう場合があります。ERR\_PAUSEをセットしておけば確実にインタプリタが停止しますので安全な対応をとることができます。

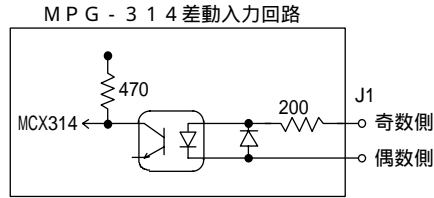
```
10 PG &H410 1
20 FORK 1 *prg
30 DO
40 IF TASK(1)==-3 THEN          /*停止を検出
50 PRINT "Err happen!!" : TIME 1000
60 HOUT X_A;CLR_ERR &H410      /*エラーのクリア
70 CONT 1                      /* (実際にはもう少し複雑な手続きの上、
80 END_IF                      /* 再起動か一時停止の解除とする)
90 LOOP
100 END
110 *prg
120 ACCEL 8000
130 wr2=MD_2PLS|ALM_ON        /*ALAM信号を有効にします。
140 HOUT wr2 (X_A;NOP,2)      /* X軸のWR 2を書き換えます。
150 ERR_PAUSE X_A            /*X軸のエラー検出をイネーブルします。
160 DO
170 RMVS X_A 1000             /*実行プログラム
180 WAIT RR(X_A)==0
190 RMVS X_A -1000
200 WAIT RR(X_A)==0
210 LOOP
```



## ピンアサイン表

### J1 エンコーダ入力

1	X-EA+ (X_AB+)	2	X-EA- (X_AB-)
3	X-EB+ (Y_AB+)	4	X-EB- (Y_AB-)
5	Y-EA+ (U_AB+)	6	Y-EA- (U_AB-)
7	Y-EB+ (Z_AB+)	8	Y-EB- (Z_AB-)
9	U-EA+ (U_IN2+)	10	U-EA- (U_IN2-)
11	U-EB+ (Z_IN2+)	12	U-EB- (Z_IN2-)
13	Z-EA+ (X_IN2+)	14	Z-EA- (X_IN2-)
15	Z-EB+ (Y_IN2+)	16	Z-EB- (Y_IN2-)



( )内副機能

注1)X\_AB~Z\_ABはアブソリュートエンコーダ用入力

注2)X\_IN2~U\_IN2は差動入力入力ポート

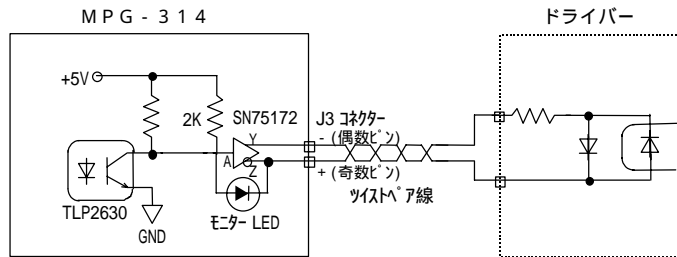
この機能を有効にするには、OC5~OC8にTLP2630を挿入する必要があります。

J1は差動入力インタフェースです。+側が200Ωの抵抗経由で内部フォトカプラのアノード側に接続され、-側がカソード側に接続されています。LED9-16はこのフォトカプラ入力の極性キャンセルダイオードとして実装されています。このためLEDの点灯は信号OFFを意味します。

基本的にはエンコーダ入力(90°位相差2相クロック)ですが、副機能としてX-AB~Z-AB、X-IN2~Z-IN2に接続されています。X-AB~Z-ABはアブソリュートエンコーダ用入力ポートで、オプションのワンチップCPUを搭載することにより使用可能となります。X-IN2~Z-IN2は条件停止に使用できるIN2に接続されています。外部条件停止入力を差動信号として受け取る場合はこの入力を使用します。

### J3 パルス出力

1	X-CW+	2	X-CW-
3	X-CCW+	4	X-CCW-
5	Y-CW+	6	Y-CW-
7	Y-CCW+	8	Y-CCW-
9	U-CW+	10	U-CW-
11	U-CCW+	12	U-CCW-
13	Z-CW+	14	Z-CW-
15	Z-CCW+	16	Z-CCW-



差動出力パルスインタフェースです。標準状態はCW,CCWの2パルス方式が内部レジスタ設定によりPULSE/DIR方式にも変更できます。

### J2 各軸リミット、アラーム入力

1	X-IN3	2	X-LMTM
3	X-LMTP	4	X-ALM (EMGN)*
5	Y-IN3	6	Y-LMTM
7	Y-LMTP	8	Y-ALM
9	U-IN3	10	U-LMTM
11	U-LMTP	12	U-ALM
13	Z-IN3	14	Z-LMTM
15	Z-LMTP	16	Z-ALM
17	X-OUT4	18	X-OUT5
19	Y-OUT4	20	Y-OUT5
21	Z-OUT4	22	Z-OUT5
23	X_IN1(A)	24	Y_IN1(A)
25	U_IN1(A)	26	Z_IN1(A)

1~16 入力:MCX314各入力ポートへ

17~22 出力:MCX314各出力ポートへ (J4-9,10含む)

23~26 原点センサー 差動入力 アノード側

注)SP4ショート時にX-ALMは全軸に有効なALMになります。

各軸-IN3 は条件停止、汎用入力として使用します。

各軸-LMTM,LMTPはリミットセンサに接続します。(常時有効)

各軸-ALMは通常サーボアラームに接続します。(設定により有効)

各軸-OUT4,OUT5は汎用出力もしくはカウンタ比較の結果を出力することができます。

U軸の信号はJ4に出ています。

各軸-IN1(A)は、各軸-S2をZ相(C相)に接続する場合の差動ドライブ用の入力です。J4の各軸-S2がカソード側に接続されているのに対してRA28経由でアノード側に接続されています。この入力を使用する場合は、RA17Bを取り外します。

## J4 標準入出力(原点センサ入力、インポジション入力、サーボオン出力 等)

1	X-INPOS	2	Y-INPOS
3	U-INPOS	4	Z-INPOS
5	X-SON (X-OUT0)	6	Y-SON (Y-OUT0)
7	U-SON (U-OUT0)	8	Z-SON (Z-OUT0)
9	U-OUT4	10	U-OUT5
11	IN1·XS1 (X-IN0)	12	IN2·XS2 (X-IN1) (K)
13	IN3·YS1 (Y-IN0)	14	IN4·YS2 (Y-IN1) (K)
15	IN5·US1 (U-IN0)	16	IN6·US2 (U-IN1) (K)
17	IN7·ZS1 (Z-IN0)	18	IN8·ZS2 (Z-IN1) (K)
19	OP1 (X-OUT1)	20	OP2 (Y-OUT1)
21	OP3 (U-OUT1)	22	OP4 (Z-OUT1)
23	GND J5GNDと接続	24	GND J5GNDと接続
25	+DC J5+DCと接続	26	+DC J5+DCと接続

1~4 入力: ドライバ・インポジション信号用  
 5~10 出力: ドライバ・サーボオン信号用  
 11~18 入力: 原点センサー用 (2線式センサ対応可)  
 19~22 出力: 汎用出力ポート  
 23~26 電源

RA17のSIPソケットは2線式センサ対応プルアップ抵抗用

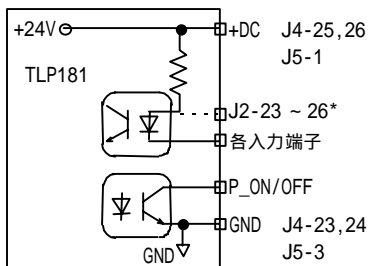
各軸-INPOSはサーボドライバのインポジション信号を接続します。レジスタ設定でこの入力を有効にすると、MPG-314は自動的にINPOS入力を待ちます。

各軸-SONはサーボオンを想定していますが、汎用出力として使用できます。

OP1~4は汎用出力です。SON,OPはHOUTで設定できます。IN1~8は原点復帰入力です。MCX-314のIN0,IN1に接続されています。

原点センサポートは各軸S1側をニアオリジン、各軸S2側をサーボZ相に使用します。これは、S1側の検出はソフト依存で、ICの原点検出機能を用いたS2側が高精度のためです。S2側はJ2 IN1と組合わせて差動ドライブ入力に対応できません。

入出力回路



\* J1-12,14,16,18はJ2側にアノード側が引き出されています。

## J5 電源

1	+DC
2	(VCC)
3	GND
4	FG

MPGの動作電源は1,3番ピンです (DC12~24V)。

SP3をショートするとパルス出力インターフェース用の5Vを外部で使用できます。(200mA以下)

## 原点センサー接続例

入力電流シャント抵抗(RA16A/B)、2線式無接点センサープルアップ抵抗(RA17A/B)、差動入力シャント抵抗(RA28)はSIPソケットになっており使用機器に応じて実装を変えます。

4素子タイプの抵抗アレイを使うことでS1側とS2側を別設定にすることができます。

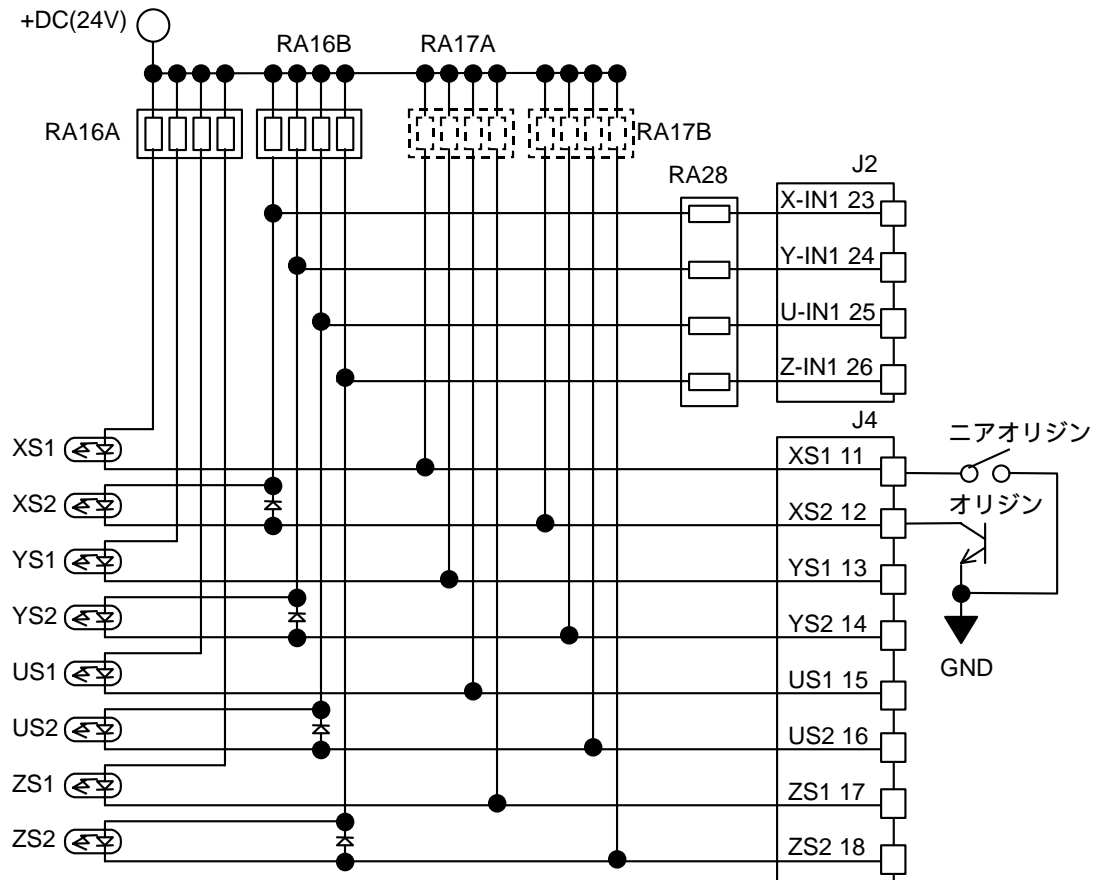
RA16A/B(6.8K )、RA28(470 )は標準実装です。

RA17A/B(2.7K 1/4W)はオプションです。

### LS接点、オープンコレクタ出力センサー

デフォルトでの接続形態です。ニアオリジンを各軸S1側、オリジンをS2側に接続します。

RA16A,RA16B 実装

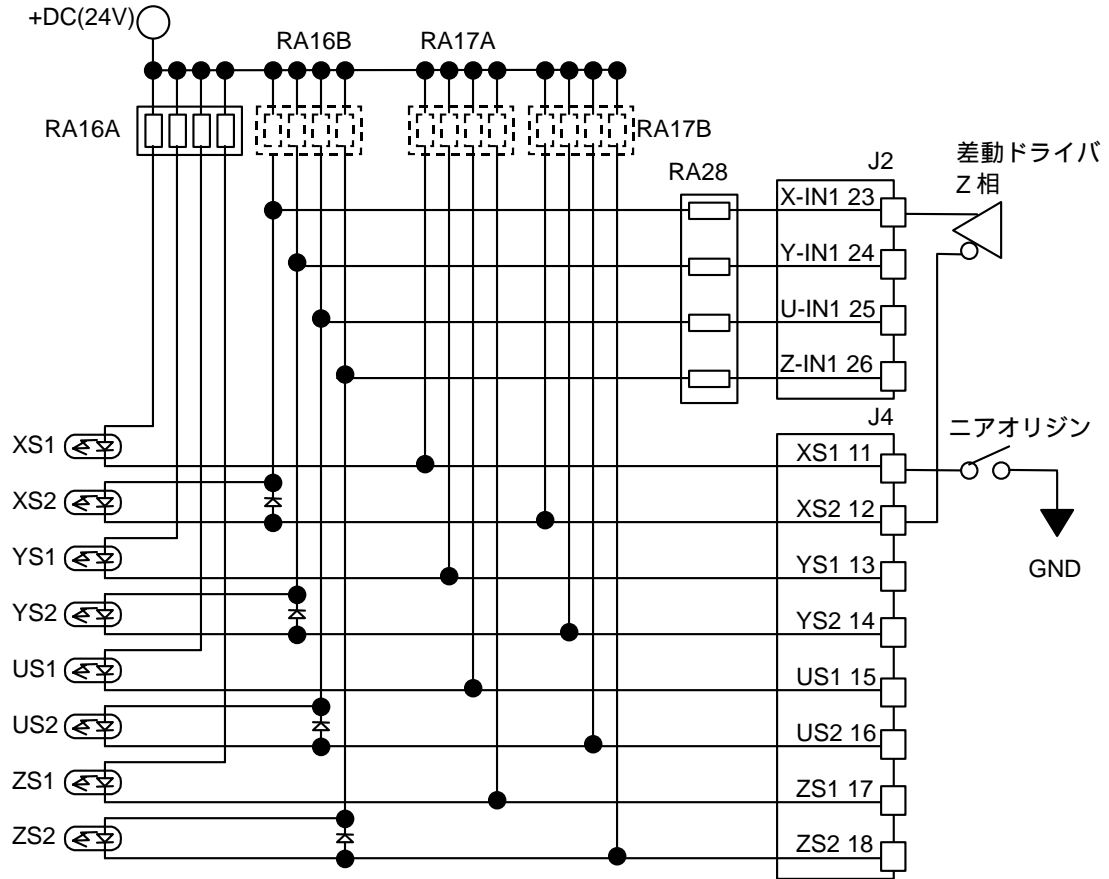




## 差動ドライバZ相インターフェース

差動ドライバはJ4 S2側とJ2 IN1に接続します。

RA16A,RA28 実装

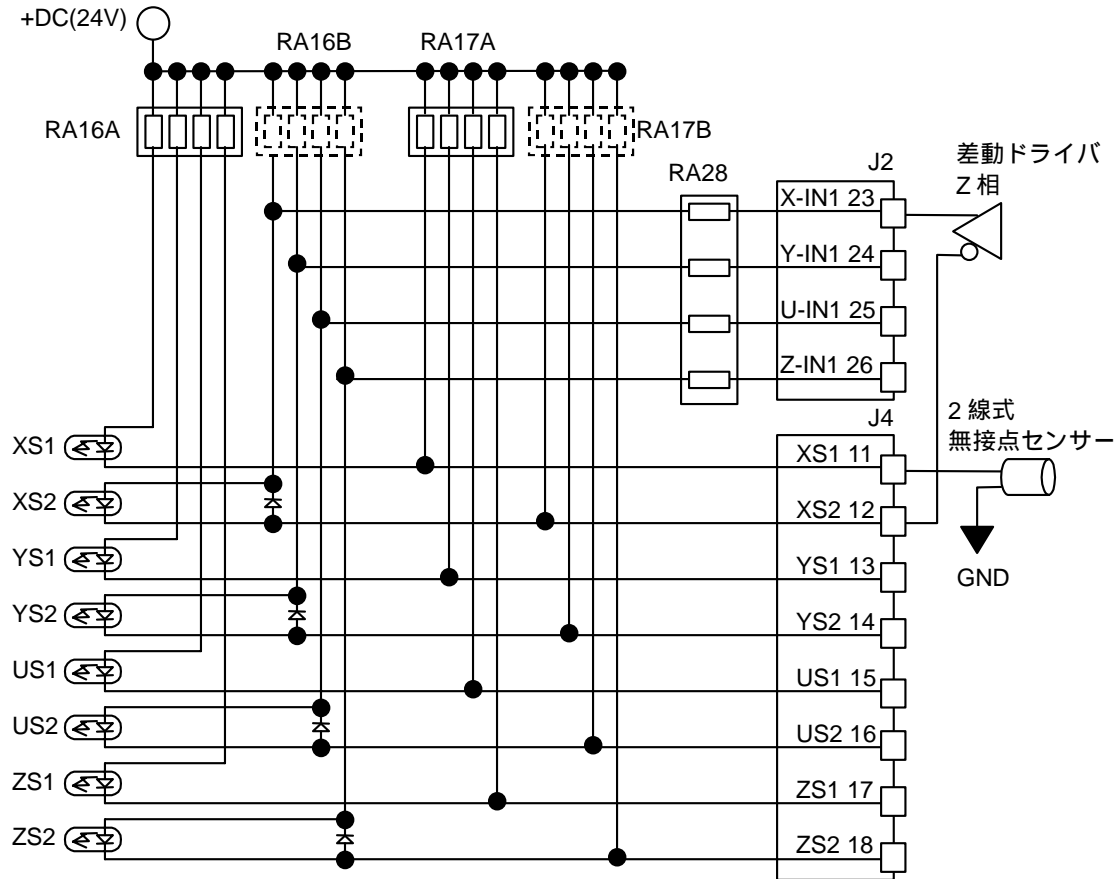


## 2線式無接点センサー

下の例ではS1側に2線式無接点センサー、S2側に差動Z相を接続しています。

RA16A,RA17A,RA28 実装

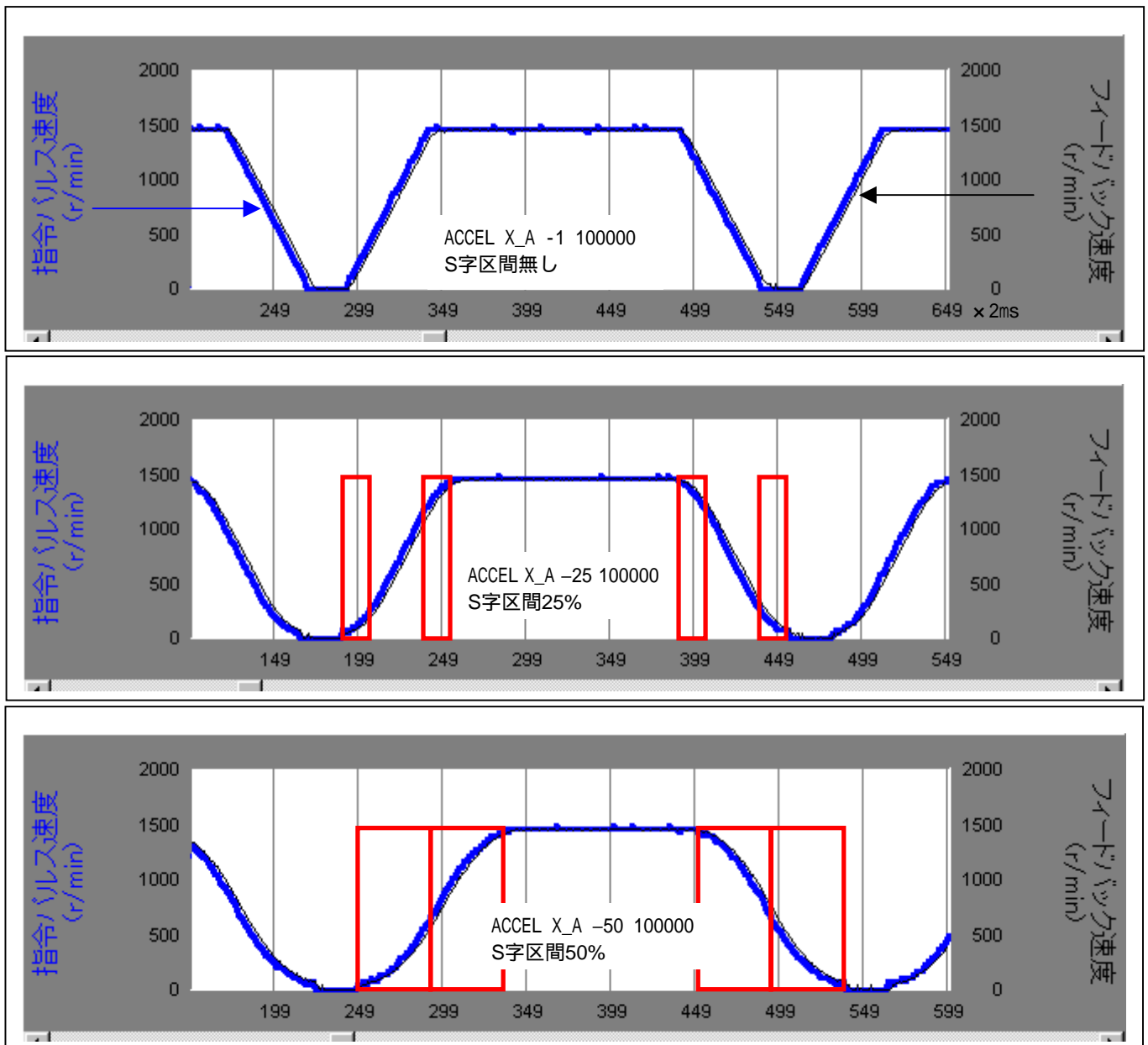
S2側も2線式無接点センサーならば RA16BとRA17Bも実装



## S字加減速

S字加減速を使うと滑らかな始動・停止が出来て、速くインポジションを得られる場合があります。しかし設定を大きくすると加減速領域が広がり、かえってタクトタイムが長くなることもあります。適切にドライバのゲイン調整を行い装置剛性に合ったS字領域を設定しなければなりません。下記はMPG-314でS字加減速をした時の実測例です。

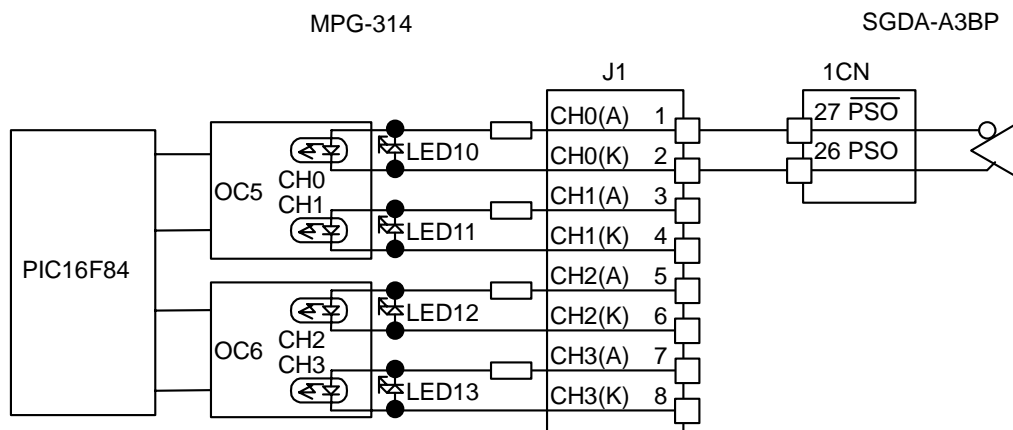
サーボモータ	安川 SGM-A3BW12C	プログラム
サーボパック	安川 SGDA-A3BP	10 ACCEL X_A -1 100000 /*第2パラメータを変える
モータは無負荷		20 DO
モニタソフト	Sigma Win Ver1.20	30 RMVS X_A 40000
		40 WAIT RR(X_A)==0
		50 TIME 50
		60 LOOP



## アブソリュートエンコーダの接続

サーボモータドライバから出力される回転量シリアルデータを読み込みます。  
オプション部品を搭載して、MPG-314 1台あたり最大4CHの入力ポートを追加できます。

- オプション品  
シリアルデータ受信用フォトカプラ OC5, OC6 (TLP2631) 1個につき2CH  
I/F用IC U8 ( PIC16F84 専用プログラム書き込み済み) 1個
- 配線  
安川 SERVOPACK SGDA-A3BP 接続例



## 付録

### MPG-314への加減速度の計算について

MPG-314に搭載されているMCX-314のレジスタは以下のような関係式で定義されています。

倍率レジスタ	1600=< R=< 8000000	倍率 = 8000000/R
加減速度レジスタ	1=<A=<8000	加速度 = A * 125 * 8000000/R
ドライブ周波数	1=<V=<8000	最大速度 = V * 8000000/R
初速度	1=<SV=<8000	初速度 = SV * 8000000/R

#### 倍率レジスタの意味

倍率とは設定値の何倍の出力をするかという設定です。たとえば800kPPSの周波数を出力する場合、Vには8000までしか設定できません。このため倍率を計算します。800000/8000 = 100  
倍率は100となるとため、倍率(100)=8000000/R R=80000とします。すると上の各式は以下のとおりになります。つまり速度に関しては100pps単位でのみ速度設定が可能となります。

$$\begin{aligned} \text{倍率} &= 100 \\ \text{加速度} &= A * 125 * 100 \\ \text{最大速度} &= V * 100 \\ \text{初速度} &= SV * 100 \end{aligned}$$

例として以下の設定を当てはめてみます。

ACCEL 800000 10000 200

この場合最大速度(ドライブ周波数)レジスタはV<=8000、初速度はSV<=2となります。加速度は以下の計算をします。関係は以下のとおりです。

$$(MAX * MAX - MIN * MIN) / (2 * accel) = L \quad **1$$

$$((800000)^2 - (200)^2) / (2 * 10000) = accel$$

$$accel = 3200 * 10^4 = A * 125 * 100$$

つまりA=2560

この時の加速時間は(800000-200)/(3200\*10<sup>4</sup>)=0.025秒となります。

\*\*1 (MAX + MIN) \* T / 2 = L (MAX - MIN) / T = accel の関係より

#### 実際の計算方法について

内部演算では8000pps以下と8001pps以上では計算方法を変えています。これは、R=8000000と固定した場合V,SVレジスタは8000までの値を1pps毎に直接指定できるためです。

8000ppsを超える場合の演算手順は以下のようにしています。

周波数設定のダイナミックレンジを最大にとるためには、V \* 8000000/Rにもとづき、max=8000\*8000000/Rを計算します。

R=64\*10<sup>9</sup>/max としてV=8000と固定します。初速度は以下の関係より算出します。

$$\text{初速度}(\text{min}) = SV * 8000000 / (64 * 10^9 / \text{max})$$

$$\text{min} = SV / (8000) * \text{max}$$

$$SV = 8000(\text{min}/\text{max})$$

加速度の算出を実施する前に、関係式を整理します。扱いやすくするために80000000/Rは と置き換えておきます。

$$\text{加速度} = A * 125 * 8000000/R$$

$$\text{最大速度} = V * 8000000/R$$

$$\text{初速度} = SV * 8000000/R$$

$$(V * )^2 - (SV * )^2 / (2 * (A * 125 * )) = L$$

$$((V^2 - (SV^2)) / (2 * (A * 125))) = L \text{ さらに}$$

$$(80000000/R) * (V^2 - SV^2) / (250 * L) = A \text{ こうして算出した値をそれぞれのレジスタに設定します。}$$

#### 演算の限界について

##### ■ 最大速度によって倍率が決定するために速度分解能に注意が必要

ACCEL 800000 10000 150という設定で150の50は無意味です。先の例のように800kppsを最大速度とすると100ppsが分解能となり150ppsは設定できません。内部では切り捨てられて100ppsと扱われます。

### ■ 最大加速度・最小加速度

加速度はLの値が大きくなると小さくなりますが、加速度の最低値は1のため(  $125 * 8000000/R$ )以下の加速度は設定できません。たとえば800kppsを最大とすれば、 $R=80000$   
125000pps/sec以下の加速度には設定できないことになります。

また、最大加速度は $A=8000$ のため $10^8$ pps/sec以上の加速度に設定できません。実用上は問題のない範囲と考えられます。

### ■ 時間制御加速度

加速距離の代わりに加速時間で制御する場合は以下の関係を用います。

Minからmaxに加速する時間Tでの加速距離Lは次の関係があります。

$$(Min + Max) * T / 2 = L$$

つまり、 $Accel\ Max = (Max+Min) * T / 2000\ Min$

と記述しTを1m秒単位で設定すれば時間加速度を得ることができます。以下は時間制御の例です。80では内部レジスタを参照して一定速領域到達検出して時間測定するプログラムです。Tを1000としていますので1秒加速になります。

```
LIST
10      PG &H410
20      T=1000
30      max=8000
40      min=200
50      ACCEL max (max+min)*T/2000 min
60      RMVS X_A 100000
70      ss=SYSCLK
80      WAIT WIR(&H412)&8<>0
90      PRINT (SYSCLK-ss)*5
#
#run
1025
#
```

## MPG-314 Q&A

### ■ ALM信号を入力する方法は？

MPG-314 コネクタJ2の ALM(4,8,12,16番ピン) 入力には二通りの方法があります。

```
HPTで読む
#PRX HPT(0)
00100000          /* X-ALM(4番ピン)がON状態
```

HPT(0)は各軸ALM入力をいつでも読むことができます。  
単なる入力(標準I/Oの代用)として使うならこれでもOK。

```
RRで読む
#INSET_314 X_A ALM_ON      /* X軸 ALMをONで有効にする
#PRX RR(X_A,ALM@)
0010          /* X-ALM(4番ピン)がON状態
```

パルス発生とALM入力を連動させるにはINSET\_314で有効にしなければなりません。  
INSET\_314でALMを有効にするとRRで読むことができるようになります。

### ■ リミット信号の検出方法は？

MPG-314のリミット検出はHPT()ではなく、RR()で行います。

```
#PG &H400
#INSET_314 X_A LMT_ON      /*リミットで検出
#PRX RR(X_A,LMTM@)        /*コネクタJ2 2番ピン入力
0008          /*リミットはONしている
#PRX RR(X_A,LMTM@)        /*コネクタJ2 3番ピン入力
0004          /*リミットはONしている
#PRX RR(X_A,LMTM@|LMTM@)  /*コネクタJ2 2,3番ピン入力
000C          /*リミットは両方ONしている
```

表示されるビットとコネクタピン番号は整合しません。  
全軸指定定数 ALL\_A は使用できません。

### ■ RR()のビット位置とは？

コネクタ J2 X-LMTM(2番ピン)、X - ALM(4番ピン)がON状態のときRRでは下記の様になります。

```
#INSET_314 X_A ALM_ON
#PRX RR(X_A,ALM@|LMTM@|LMTM@)
0018
```

ここで表されるビット位置は次のようになっています。

(上位)	ビット	&H20	10	8	4	2	1	(下位)
	信号	EMG	ALM	LMTM	LMTM	0	0	
0018なら		0	1	1	0	0	0	

## ■ 互換コマンドのSTOPは使えるか？

MPG-314専用コマンドであるMOVSやRMVSには互換スタイルのSTOPは使えません。  
互換コマンドはMPG-68K、MPG-405用のプログラムをMPG-314で動作させるためにあります。  
314専用コマンドに対してのFEEDやSTOPなどは拡張ステートメントで対応します。

下のプログラムは互換STOPを模してタスク0からタスク\*XA,\*YAのパルス発生を停止していますが、  
MOVSは実行後すぐに次のステップに移行するので、タスク \*UAのように同一タスクで処理するほうが  
判りやす場合もあります。

```
PG &H400
PG &H400 1
PG &H400 2
ACCEL X_A|Y_A 5000
/*FEED 0          /* 軸指定しない場合は0-64 各軸共通
FEED X_A 0
FEED Y_A 255      /* 軸指定した場合は設定範囲は0~255
STPS 0 0 VOID 0
PRINT P(0)
FORK 1,*XA
FORK 2,*YA
FORK 3,*UA
INPUT "HIT ENTER YKEY" A
/*STOP 2 1        /* MOVSには互換のSTOPは使えない
STOP X_A STP_I
TIME 500
INPUT "HIT ENTER YKEY" A
STOP Y_A STP_I
TIME 500
END
*XA
MOVS X_A 100000   /* 実行後すぐ次のコマンドへ
WAIT RR(X_A)=0   /* 停止待ち
PR "PULSE STOP X"
END
*YA
MOVS Y_A 100000   /* 実行後すぐ次のコマンドへ
WAIT RR(Y_A)=0   /* 停止待ち
PR "PULSE STOP Y"
END
*UA
PG &H400
ACCEL U_A 1000
FEED U_A 0
MOVS U_A 100000   /* 実行後すぐ次のコマンドへ
WAIT SW(192)=1
STOP U_A STP_I
PR "PULSE STOP U"
END
```



## ■ 原点復帰例

第1段階 NEAR ORGまで高速で原点復帰 (NEAR ORGはNCのリミットSW)

第2段階 Z相がわになるまでゆっくり原点復帰

```
PG &H400
ACCEL 300000
FEED X_A 0
DO
  IF HPT(0)&&H1=0 THEN
    RMOV 50000 0 0 /* CW退避移動
  END_IF
  HOME X_A IN0_OFF -200 /* XS1(NEAR ORG)がOFFになるまでCCW FEED=200
  WAIT RR(X_A)=0 /* HOMEもすぐ抜ける
  HOME X_A -1000 IN1_ON /* XS2(Z相)がわになるまでCCW 1000pps
  WAIT RR(X_A)=0 /* HOMEもすぐ抜ける
  STPS 0 0 0 0 /* 現在点を0にする
  PR P(0)
  PRSET_ACCEL X_A /* ACCELパラメータ復帰、停止条件クリア
  FEED X_A 0 /* FEED再設定
  TIME 1000
  MOVE 100000 0 0
  TIME 1000
LOOP
```

## ■ 連続補間について

【Q】下記のプログラムを実行したところ現在座標値はX=-4000 Y=4000 になりました。X=-2000 Y=2000 にはならないのですか？

```
*TST
  SETP 100 -2000 2000 0 0
  SETP 101 -1000 1000 0 0
  SETP 102 -2000 2000 0 0
  SETP 103 -1000 1000 0 0
  SETP 104 -2000 2000 0 0
  SETP 105 -1000 1000 0 0
  RANGE 0
  INSET_314 ALL_A SLMT_OFF|LMT_OFF|ALM_OFF
'100
  i=100
  WAIT RR(X_A|Y_A)==0
  MOVL X(i) Y(i) VOID VOID
  WAIT RR(X_A|Y_A)==0
  HOUT X_A|Y_A;DS_DACL
'101
  i=101
  MOVT X_A|Y_A P(i)
'102
  i=i+1
  MOVT X_A|Y_A P(i)
'103
  i=i+1
  MOVT X_A|Y_A P(i)
'104
  i=i+1
  MOVT X_A|Y_A P(i)
'100
  i=100
  MOVT X_A|Y_A P(i)
,
  HOUT X_A|Y_A;EN_DACL
RETURN

#RUN *TST
#CP
X= -4000 Y=-4000 U= 0 Z= 0
```

【A】movtの点番号指定は不自由なもので、順にしか定義できません。理由は、直前の点に対しての相対パルスを求め、その値をPGに設定するためです。コマンドが先読みされるためコマンド実行時点での現在位置がつかえないためです。

そのために、マニュアル例でも戻ってきた同じ点を異なる点番号で設定してあるのです。

今回の不具合は104の次に101を指定したために発生しました。

104で-2000,2000のあと指定された点の相対パルス値は、-2000,2000(多分99は0,0,0,0のため)で-4000,4000となるわけです。

```

          ----
PG &H410 : ACCEL 2000
*TST
  SETP 100 -2000 2000 0 0
  SETP 101 -1000 1000 0 0
  SETP 102 -2000 2000 0 0
  SETP 103 -1000 1000 0 0
  SETP 104 -2000 2000 0 0
  SETP 105 -1000 1000 0 0
  SETP 106 -2000 2000 0 0
RANGE 0
INSET_314 ALL_A SLMT_OFF|LMT_ON|ALM_ON
'100
  i=100
  WAIT RR(X_A|Y_A)==0
  MOVL X(i) Y(i) VOID VOID
  WAIT RR(X_A|Y_A)==0
  HOUT X_A|Y_A;DS_DACL
'101
  i=101
  MOVT X_A|Y_A P(i)
'102
  i=i+1
  MOVT X_A|Y_A P(i)
'103
  i=i+1
  MOVT X_A|Y_A P(i)
'104
  i=i+1
  MOVT X_A|Y_A P(i)
'105
  i=i+1
  MOVT X_A|Y_A P(i)
'106
  i=i+1
  MOVT X_A|Y_A P(i)
,
  HOUT X_A|Y_A;EN_DACL
  WAIT RR(ALL_A)==0
  CP
'105
```

## ■ 連続移動中の出力について

【Q】連続移動中に出力を操作する方法と注意点は？

【A】MOV T,RMVTは連続移動を保证するために命令の先読みを行います。このために移動中でのIOの制御には先読みによるズレへの配慮が必要となります。

以下の例では点101にさしかかったところでON1を実行し、点102にさしかかったところでON2を実行し、点106にさしかかったところでON6を実行するようにプログラムしてあります。つまり、コマンドの先読みが行われるためにタイミング信号は次の点のMOV Tの直後に記述します。

また、この方法では点106でのON 6の実行機会を作るために最終点への移動を細かく区切、最後に必要なタイミングを作っています。

'点データ設定

```
SETP 100 0 0 0 0
SETP 101 400 0 0 0
SETP 102 800 0 0 0
SETP 103 1200 0 0 0
SETP 104 1600 0 0 0
SETP 105 2000 0 0 0
SETP 106 2010 0 0 0
SETP 107 2020 0 0 0
```

```
PG &H410
ACCEL 8000
STPS 0 0 0 0
WAIT RR(X_A|Y_A)==0
MOVS 0 0 VOID VOID
WAIT RR(X_A|Y_A)==0
HOUT X_A;DS_DACL
MOV T X_A|Y_A P(101)
OUT 0 0
MOV T X_A|Y_A P(102)
ON 1 : TIME 100 : OFF 1
MOV T X_A|Y_A P(103)
ON 2 : TIME 100 : OFF 2
MOV T X_A|Y_A P(104)
ON 3 : TIME 100 : OFF 3
MOV T X_A|Y_A P(105)
ON 4 : TIME 100 : OFF 4
MOV T X_A|Y_A P(106)
ON 5 : TIME 100 : OFF 5
MOV T X_A|Y_A P(107)
HOUT X_A;EN_DACL
ON 6 : TIME 100 off 6
```

## ■ 入出力ポートは汎用入出力として使用可能か？

【Q】MPG-314の入出力ポートはコマンドで設定しなければ通常の汎用ポートとして全て使用できるのでしょうか？例えばリミット入力を単純な入力として使用できますか？

【A】リミット入力以外は汎用入力ポートとして使用できます。リミットだけは、イネーブル/ディズエーブルがなく論理設定のみです。パワーオン後、リミットはONで検出、ALMとINPOSは無視です。

INPOS,ALM,IN0~IN3はイネーブルしなければただの入力です。

なおイネーブルするには INSET\_314 X\_A ALM\_ON|INP\_ON などとします。

出力ポートはパラレル設定（ビット単位のON/OFFはない）ですが、自由につかえます。

## ■ OUT4、OUT5の出力方法

### 【Q】

J2,J4の各軸-OUT4,OUT5出力の使用方法は？

### 【A】

J2,J4の各軸-OUT4,OUT5の主機能はカウンタ比較結果を出力することです。

例えば HOUT &H80+4 (X\_A;NOP,3) とすると

OUT4：論理/実位置カウンタがCOMP+レジスタより大きいときHiレベル、小さいときLowレベル

OUT5：論理/実位置カウンタがCOMP-レジスタより大きいときHiレベル、小さいときLowレベル

になります。

これらポートは次のように用いると汎用出力としても使用できます。

論理は0がON、1がOFFでパラレル操作のみです。

```
HOUT &h304 (X_A;NOP,3) /* X-OUT4(J2-17)=OFF X-OUT5(J2-18)=OFF
HOUT &h204 (X_A;NOP,3) /* X-OUT4(J2-17)=ON X-OUT5(J2-18)=OFF
HOUT &h104 (X_A;NOP,3) /* X-OUT4(J2-17)=OFF X-OUT5(J2-18)=ON
HOUT &h004 (X_A;NOP,3) /* X-OUT4(J2-17)=ON X-OUT5(J2-18)=ON
```

/~~~~~ /~~~~~

出力データ X軸WR3設定

(出力データの下の'4'はレジスタWR3 D2(SACC)を1にするため)

この様に他の出力とは違って使い難く、汎用出力としては実用的ではありません。MPG-314の出力としてはOP1~OP4(J4 19~22)をお勧めします。

注意:

SACCはS字加減速を有効にする為のビットです。ACCEL設定時にこのビットもS字の有無によって設定されます。

このため、この出力ポートの操作で下位バイトの値を04とするか00とするかはS字の要・不要に従って設定して下さい。xx04の場合がS字有効です。