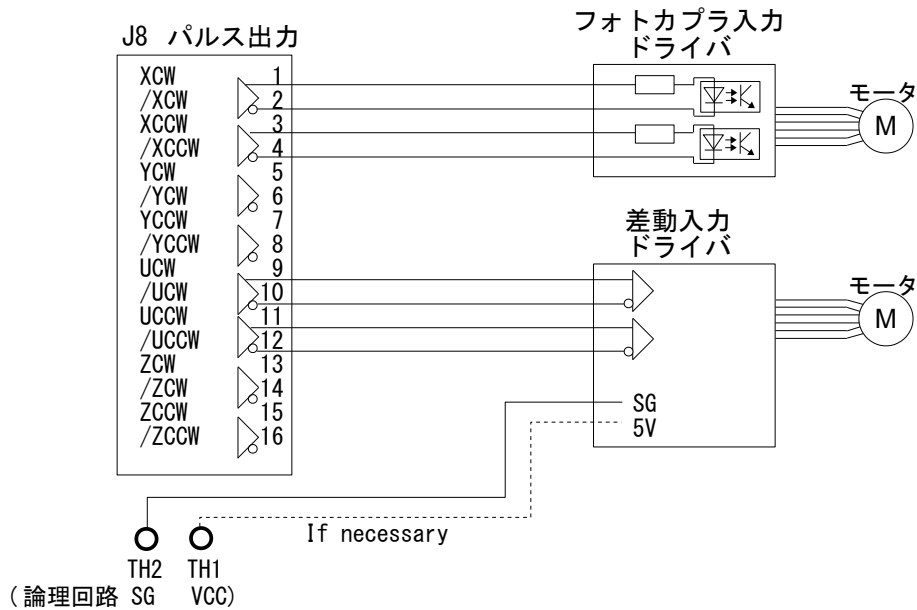


Technical Information		Ref No: ti2k-140205	Last Modify 140225
Title	MPC-1200 パルス出力例		

## 接続

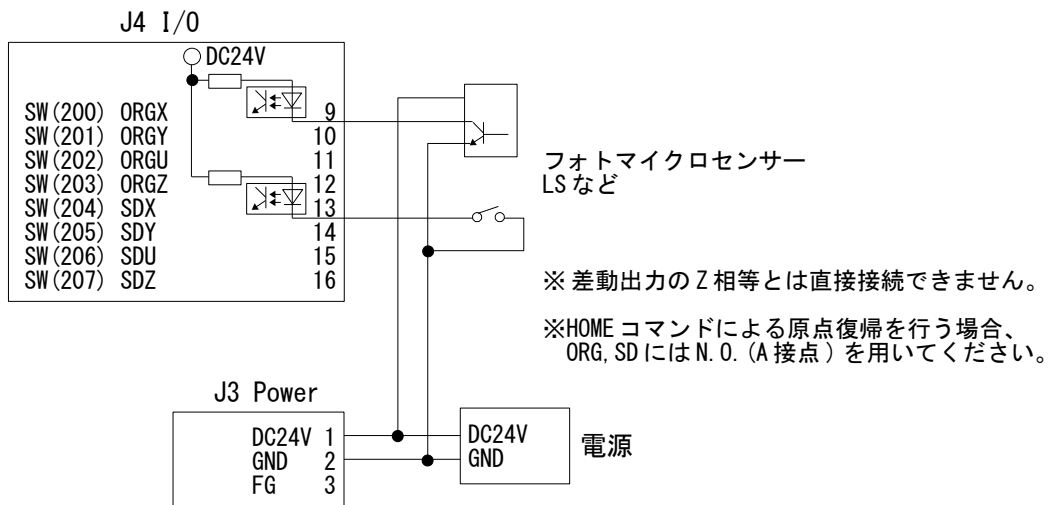
### パルス出力

パルス出力はAM26C31です。推奨はフォトカプラ入力ドライバです。



### オリジン、ニアオリジン入力

HOME コマンドで原点復帰をする場合は各軸の ORG にオリジン、SD にニアオリジンを接続します。センサーが1つの場合はORGに接続し、SDはSHOMコマンドでマスクします。下図はX軸のORG, SD接続例です。もちろん、これらは通常の入力として機能します。(SDはSlow Downの略です)



## SD 入力の有効/無効

Bit	7	6	5	4	3	2	1	0
Signal	ORGZ	SDZ	ORGU	SDU	ORGY	SDY	ORGX	SDX

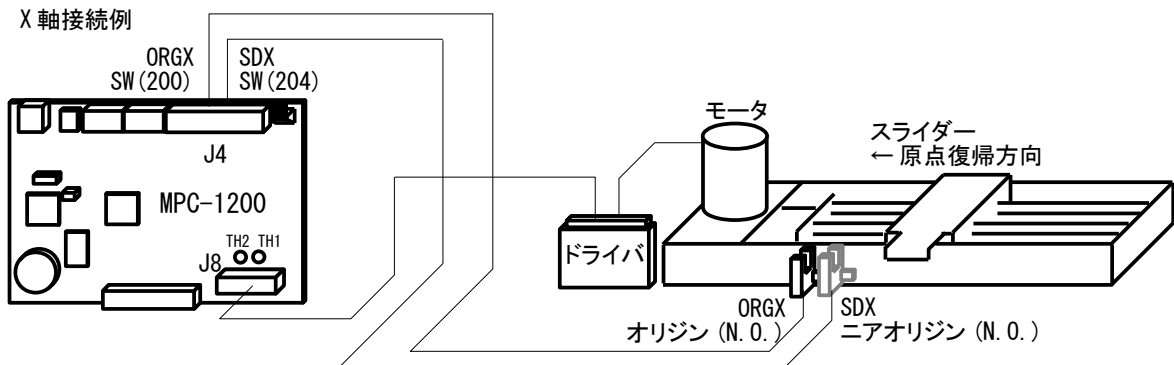
SD の入力の有効/無効を SHOM コマンドで設定します。  
 使用しない SD のビットを 1 とします。マスクした SD や HOME コマンドで使用しない SD は通常の入力として使用できます。(各軸 SD は該当する軸の HOME コマンド実行時に適応されますから、HOME コマンドを実行しない軸は SHOM 設定も要りません。)

例)  
 SHOM &H55 /\* 全 SD 無効  
 SHOM &H01 /\* SDX 無効  
 SHOM &H54 /\* SDX のみ有効  
 SHOM 0 /\* 全 SD 有効

## 原点復帰

### HOME コマンドで原点復帰

HOME コマンドで原点復帰を行う場合、ORG および SD には N. O. (A 接点) のセンサー、スイッチを用います。N. C. (B 接点) を使う場合は後述の「HOME コマンドを使わない原点復帰」をご覧ください。



### 例 1 (単軸 ORG,SD 入力)

動作開始時に ORGX がオンしていると逆方向に退避移動してから原点復帰します。  
 原点復帰の速度は、SDX まで ACCEL 設定の max 値、SDX から ORGX まで ACCEL 設定の min 値で動作します。  
 SDX は ORGX 検出までオン状態を維持するようにしてください。途中で SDX がオフになると速度は max 値に戻ります。

```
PG 17 /* PG アサイン MPC-1200 は 17
GOSUB *X_HOME /* 原点復帰サブルーチン
ACCEL X_A 50000 2000 1000
FEED X_A 100
DO
(動作 略)
LOOP
*X_HOME
PRINT "HOME"
ACCEL X_A 20000 2000 500 /* SDX から ORGX までは 500pps で動作
IF SW(200)==1 THEN /* ORGX が入っていたら
FEED X_A 100
```

```

    RMVS X_A 10000          /* X軸退避移動
    WAIT RR(X_A)==0        /* 退避移動完了待ち
END_IF

FEED X_A 25               /* 必要に応じて
SHOM &H54                 /* SDX有効
HOME X_A NEG_L            /* 原点復帰 SDXまでACCELのmax, ORGXまでACCELのmin
WAIT RR(X_A)==0          /* 原点復帰終了待ち
CLRPOS X_A                /* 現在点クリア
TIME 500
RETURN

```

## 例2 (単軸 ORG 入力のみ)

例1とほぼ同じですが、原点復帰の速度は、ORGXがオンするまでACCEL設定のmax値で動作します。SDXはSHOMでマスクしています。

```

*X_HOME
PRINT "HOME"
ACCEL X_A 20000 2000 500 /* SDXからORGXまでは500ppsで動作
IF SW(200)==1 THEN      /* ORGXが入っていたら
    FEED X_A 100
    RMVS X_A 10000       /* X軸退避移動
    WAIT RR(X_A)==0     /* 退避移動完了待ち
END_IF

FEED X_A 25             /* 必要に応じて
SHOM &H01                /* SDX無効
HOME X_A NEG_L          /* 原点復帰 ORGXまでACCELのmax
WAIT RR(X_A)==0        /* 原点復帰終了待ち
CLRPOS X_A              /* 現在点クリア
TIME 500
RETURN

```

## 例3 (XY軸 ORG,SD 入力)

例1とほぼ同じですが、XY軸同時に原点復帰します。

```

PG 17                    /* PGアサイン MPC-1200は17
axis=X_A|Y_A             /* X,Y軸

GOSUB *XY_HOME           /* 原点復帰サブルーチン

ACCEL axis 50000 2000 1000
FEED axis 100
DO
    (動作略)
LOOP

*X_Y_HOME
PRINT "HOME"
ACCEL axis 20000 2000 500 /* SDX,SDYからORGX,ORGYまでは500ppsで動作
FEED axis 100
IF SW(200)==1 THEN      /* ORGXが入っていたら
    RMVS X_A 5000       /* X軸退避移動
END_IF
IF SW(201)==1 THEN     /* ORGYが入っていたら
    RMVS Y_A 5000       /* Y軸退避移動
END_IF
WAIT RR(axis)==0        /* 退避移動完了待ち

FEED axis 25            /* 必要に応じて
SHOM &H50                /* SDX,SDY有効
HOME axis NEG_L         /* 原点復帰 SDまでACCELのmax, ORGまでACCELのmin

```

```

WAIT RR(axis)==0          /* 原点復帰終了待ち
CLRPOS axis              /* 現在点クリア
TIME 500
RETURN

```

### 例 4 (XY 軸 ORG 入力のみ)

例 3 とほぼ同じですが、SHOM で全 SD を無効にしています。

```

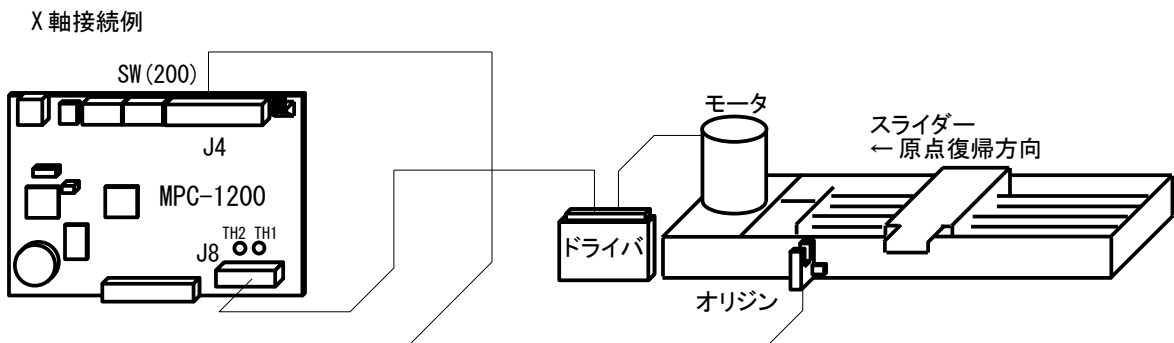
*XY_HOME
PRINT "HOME"
ACCEL axis 20000 2000 500 /* SDX, SDY から ORGX, ORGY までは 500pps で動作
FEED axis 100
IF SW(200)==1 THEN      /* ORGX が入っていたら
  RMVS X_A 5000         /* X 軸退避移動
END_IF
IF SW(201)==1 THEN      /* ORGY が入っていたら
  RMVS Y_A 5000         /* Y 軸退避移動
END_IF
WAIT RR(axis)==0        /* 退避移動完了待ち

FEED axis 25            /* 必要に応じて
SHOM &H55              /* 全 SD 無効
HOME axis NEG_L        /* 原点復帰 ORG まで ACCEL の max
WAIT RR(axis)==0      /* 原点復帰終了待ち
CLRPOS axis           /* 現在点クリア
TIME 500
RETURN

```

### HOME コマンドを使わない原点復帰

パルス発生コマンドで原点復帰します。  
この例では便宜的にオリジンを上記の HOME コマンド例と同じ SW(200) に接続していますが、入力ポートに制約はありません。



### 例 1 (単軸)

```

PG 17                    /* PG アサイン MPC-1200 は 17
GOSUB *X_HOME           /* 原点復帰サブルーチン

ACCEL X_A 50000 2000 1000
FEED X_A 100
DO
  (動作 略)
LOOP

*X_HOME
PRINT "HOME"

ACCEL X_A 10000 2000 500 /* 速度設定

```

```

IF SW(200)==1 THEN          /* X軸原点が入っていたら
  FEED X_A 100
  RMVS X_A 5000             /* X軸退避移動
  WAIT RR(X_A)==0          /* 退避移動完了待ち
END_IF

FEED X_A 10                /* 速度変更
RMVS X_A NEG_L             /* CCWパルス発生
WAIT SW(200)==1           /* X軸原点オン待ち
/* WAIT UNTIL SW(200)==1  /* 高速応答の場合
STOP X_A STP_I            /* 即停止
WAIT RR(X_A)==0           /* パルス発生終了待ち
CLRPOS axis               /* 現在点クリア
TIME 500
RETURN

```

## 例 2 (XY 軸)

```

*HOME
PRINT "HOME"

ACCEL X_A|Y_A 10000 2000 500 /* 速度設定
FEED X_A|Y_A 100
IF SW(200)==1 THEN          /* X軸原点が入っていたら
  RMVS X_A 5000             /* X軸退避移動
END_IF
IF SW(201)==1 THEN         /* Y軸原点が入っていたら
  RMVS Y_A 5000            /* Y軸退避移動
END_IF
WAIT RR(X_A|Y_A)==0        /* 退避移動完了待ち

FEED X_A|Y_A 10            /* 速度変更
RMVS NEG_L NEG_L 0 0      /* XY軸CCWパルス発生 (RMVS X_A|Y_A NEG_L という記述は不可)
DO
  IF SW(200)==1 THEN       /* X軸原点がオンしたら
    STOP X_A STP_I         /* X軸即停止
  END_IF
  IF SW(201)==1 THEN       /* Y軸原点がオンしたら
    STOP Y_A STP_I         /* Y軸即停止
  END_IF
  IF RR(X_A|Y_A)==0 THEN   /* パルス発生終了したら
    BREAK
  END_IF
  SWAP
LOOP
CLRPOS X_A|Y_A             /* 現在点クリア
TIME 500
RETURN

```

## 例 3 (X 軸 N.C. 入力の場合)

```

*X_HOME
PRINT "HOME"
ACCEL X_A 20000 2000 500    /* SDX から ORGX までは 500pps で動作
IF SW(200)==0 THEN         /* ORGX が入っていたら
  FEED X_A 100
  RMVS X_A 10000           /* X軸退避移動
  WAIT RR(X_A)==0         /* 退避移動完了待ち
END_IF

FEED X_A 10                /* 速度変更
RMVS X_A NEG_L             /* X軸CCWパルス発生
WAIT SW(200)==0           /* X軸原点オン待ち
/* WAIT UNTIL SW(200)==0  /* 高速応答の場合
STOP X_A STP_I            /* 即停止

```

```

WAIT RR(X_A)==0          /* 停止待ち
CLRPOS X_A              /* 現在点クリア
TIME 500
RETURN

```

## パルス発生例

### 単軸 絶対座標、相対座標移動

単軸でピッチ送りを繰り返します。

```

PG 17                    /* PG アサイン MPC-1200 は 17

GOSUB *X_HOME           /* 原点復帰

ACCEL X_A 50000 2000 1000 /* 最高速、加減速設定
FEED X_A 100
MOVS X_A 10000          /* 絶対座標移動
WAIT RR(X_A)==0        /* 移動完了待ち
CLRPOS X_A             /* ここを 0 に再設定する(電氣的原点)
TIME 500

DO
  FOR i=0 TO 9          /* 10 回ピッチ送り
    RMVS X_A 2000       /* 相対座標移動
    WAIT RR(X_A)==0    /* 移動完了待ち
    PR "Current Pos" X(0) /* 現在点表示
    TIME 100
  NEXT
  MOVS X_A 0           /* 電氣的原点に戻る
  WAIT RR(X_A)==0
  PR "Current Pos" X(0) /* 現在点表示
  TIME 500
LOOP

```

### XY 軸 同時移動

XY 軸でピッチ送りを繰り返します。

軸間の補間はありませんから、移動開始は同時ですが停止は別々です。

```

PG 17                    /* PG アサイン MPC-1200 は 17

GOSUB *XY_HOME          /* 原点復帰

ACCEL X_A|Y_A 50000 2000 1000 /* 最高速、加減速設定
FEED X_A|Y_A 100
MOVS 10000 5000        /* 絶対座標移動
WAIT RR(X_A|Y_A)==0    /* 移動完了待ち
CLRPOS X_A|Y_A        /* ここを 0 に再設定する(電氣的原点)
TIME 500

DO
  FOR i=0 TO 9          /* 10 回ピッチ送り
    RMVS 2000 10000    /* 相対座標移動
    WAIT RR(X_A|Y_A)==0 /* 移動完了待ち
    PR "Current Pos" X(0) Y(0) /* 現在点表示
    TIME 100
  NEXT
  MOVS 0 0             /* 電氣的原点に戻る
  WAIT RR(X_A|Y_A)==0
  PR "Current Pos" X(0) Y(0) /* 現在点表示
  TIME 500

```

LOOP

## マルチタスク XY 軸非同期

X軸とY軸を別のタスクで非同期制御します。  
X軸はピッチ送り、Y軸は0と10000を往復します。

```
QUIT_FORK 1 *AXIS_X      /* X軸制御
QUIT_FORK 2 *AXIS_Y      /* Y軸制御
END                        /* TASKO終了。FTMW2KやMPC Monitorと通信可能。
```

\*AXIS\_X

```
PG 17                      /* PGアサイン MPC-1200は17
GOSUB *X_HOME              /* 原点復帰
ACCEL X_A 50000 2000 1000 /* 最高速、加減速設定
FEED X_A 100
MOVS X_A 10000             /* 絶対座標移動
WAIT RR(X_A)==0           /* 移動完了待ち
CLRPOS X_A                /* ここを0に再設定する(電氣的原点)
TIME 500
DO
  FOR i=0 TO 9             /* 10回ピッチ送り
    RMVS X_A 2000          /* 相対座標移動
    WAIT RR(X_A)==0       /* 移動完了待ち
    TIME 100
  NEXT
  MOVS X_A 0              /* 電氣的原点に戻る
  WAIT RR(X_A)==0
  TIME 500
LOOP
```

\*AXIS\_Y

```
PG 17                      /* PGアサイン MPC-1200は17
GOSUB *Y_HOME              /* 原点復帰
ACCEL Y_A 10000 1000 500  /* 最高速、加減速設定
FEED Y_A 100
DO
  RMVS Y_A 10000          /* 相対座標移動
  WAIT RR(Y_A)==0        /* 移動完了待ち
  TIME 100
  MOVS Y_A 0              /* 原点に戻る
  WAIT RR(Y_A)==0
  TIME 100
LOOP
```

## XUZ 軸同時移動

3軸同時スタートですが、停止は別々です。

```
PG 17
ACCEL X_A|U_A|Z_A 10000 1000 500 /* 速度、加減速設定
CLRPOS X_A|U_A|Z_A             /* 現在位置クリア
PRINT "step1" P(0)            /* XYUZ 現在位置表示
MOVS 10000 VOID 20000 30000   /* 絶対座標移動 Y軸移動無し
WAIT RR(ALL_A)==0             /* 移動完了待ち
PRINT "step2" P(0)
```

```

RMVS -10000 0 -20000 -30000      /* 相対座標移動 Y軸移動無し
WAIT RR(ALL_A)==0
PRINT "step3" P(0)

```

・実行結果

RUN

```

step1 0 0 0 0
step2 10000 0 20000 30000
step3 0 0 0 0

```

## 条件停止

Y軸で移動量を指定しないパルス発生をして、SW入力または移動量で停止します。

```

PG 17
ACCEL Y_A 10000 1000 500      /* 速度、加減速設定
CLRPOS Y_A                  /* 現在位置クリア
PRINT "step1" Y(0)          /* XYUZ 現在位置表示
RMVC Y_A CCW                /* 量を指定しないパルス発生
WAIT (SW(204) | (Y(0) < -10000)) == 1 /* 条件待ち
STOP Y_A STP_I              /* 即停止
WAIT RR(Y_A) == 0           /* 停止待ち
PRINT "step2" Y(0)

```

・実行結果

RUN

```

# step1 0
step2 -3692    ←移動中 SW(204) をオン

```

RUN

```

# step1 0
step2 -10002   ←SW(204) オフのまま

```

## 移動中の速度変更

パルスを出しながら座標値により減速します。  
高速になるとインタプリタ処理の遅延が出る場合があります。

```

PG 17
GOSUB *X_HOME                /* 原点復帰
ACCEL X_A 20000 2000 1000    /* 最高速、加減速設定
DO
  FEED 100                   /* 最初の速度
  RMVS X_A 30000              /* 相対座標移動
  WAIT X(0) >= 10000          /* 座標値確認
  p=X(0)
  PR "setp1" p                /* 表示
  FEED 50                     /* 減速
  WAIT X(0) >= 20000          /* 座標値確認
  p=X(0)
  PR "setp2" p                /* 表示
  FEED 10                     /* さらに減速
  WAIT RR(X_A) == 0           /* パルス発生終了待ち
  p=X(0)
  PR "setp3" p                /* 表示
  TIME 100
  FEED 100
  MOVS X_A 0
  WAIT RR(X_A) == 0

```



TIME 500  
LOOP

・ 実行結果

RUN

```
# HOME  
setp1 10000  
setp2 20000  
setp3 30000  
setp1 10001  
setp2 20000  
setp3 30000  
setp1 10000  
setp2 20000
```

## WARP モーション

垂直軸と水平軸でアーチ状のジャンプ移動をします。P&Pが高速になります。  
下図はパレットと任意の1点間移動をシミュレーションしたものです。  
詳細は別資料で。

