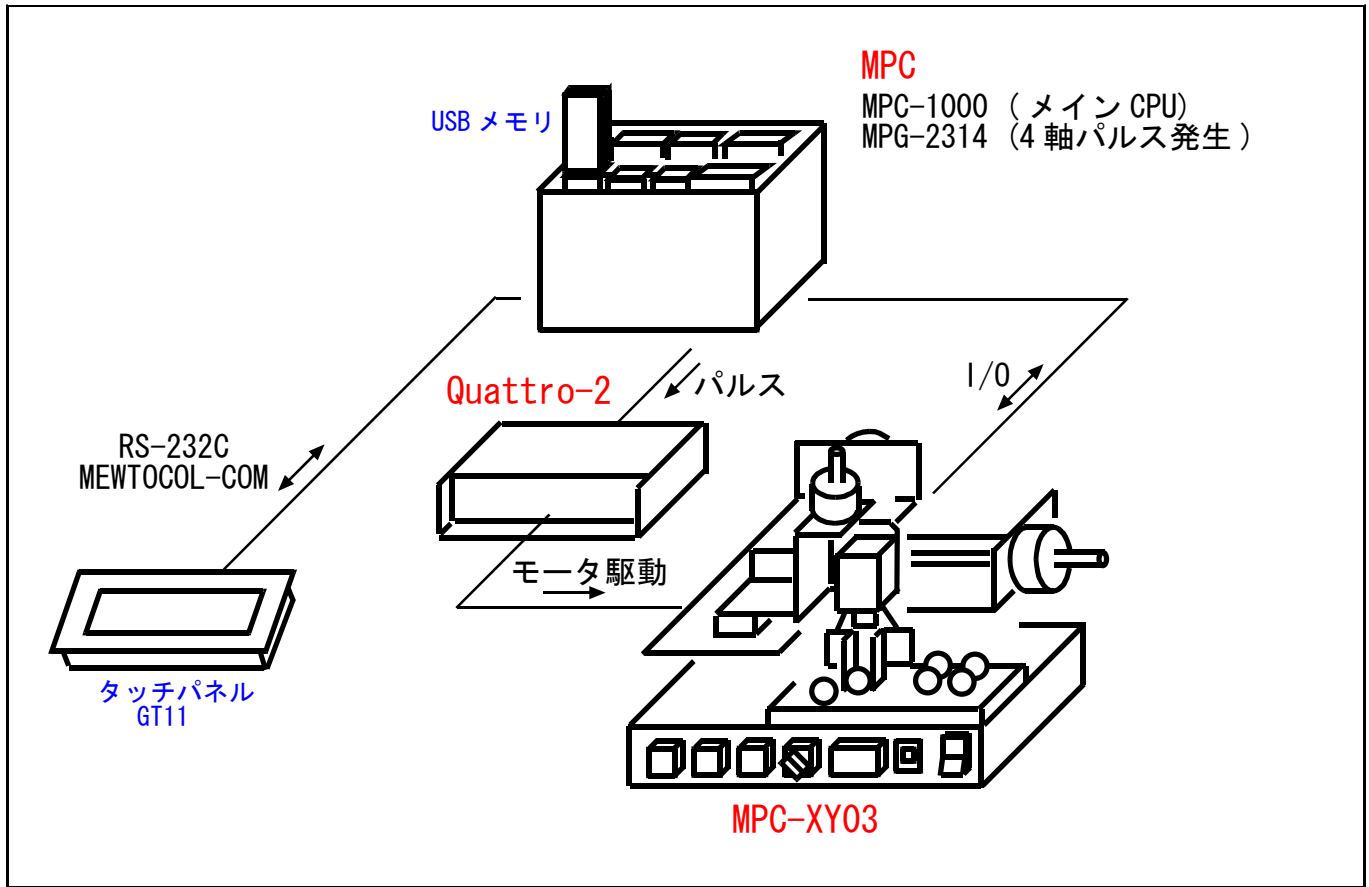


Application Note		資料作成 100226	資料番号 an2k-018
テーマ	3軸直行型ロボットで小型卓上機 (MPC-1000 デモ機)		
使用機器	MPC-1000、MPG-2314、MPC-XY03、タッチパネル 他		

■機器構成



■概要

- ・ ステップモータ 3 軸直行型ロボットを使用し、小型卓上機を想定して作成したデモ機です。
- ・ パレット命令と JUMP コマンドでパレット間のワーク搬送を行います。
- ・ タッチパネルでパレット上のティーチングを行います。
- ・ USB メモリを使用して連続・円弧補間などの動作切り替え(機種変更)を行います。
- ・ MPC-1000 はバッテリーがありません。時計はタッチパネルのデータで設定します。

■パレットコマンド

デモ機プログラムの AUTO 動作はパレット間のワーク搬送(P&P)です。
 PALLET コマンドは歪んだパレットや 1 列のパレットにも対応できます。
 ティーチングポイントとして MPC-1000 のフラッシュメモリに保存される領域 P(100)~P(299)を使用しています。



■円弧補間

MPG-2314 は任意の 2 軸を組み合わせると円弧・連続補間が可能です。デモ機では USB メモリに保存されている点データファイルを選択→読み込み→データフォーマットに従って連続・円弧補間を行います。

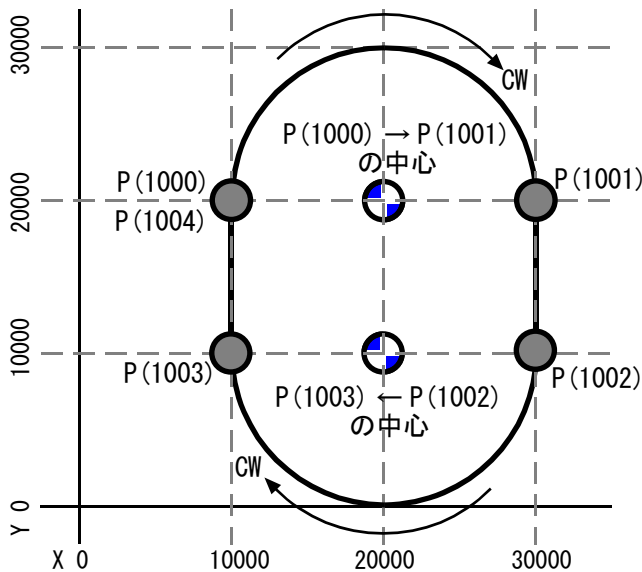
デモ機プログラムでは作業用のデータとして、P(1000)台を通過点と円弧中心の座標エリア、P(2000)台を動作の定義エリアとしています。

P(2000)の X 成分は最終点番号、Y 成分は FEED、U 成分は円弧補間か直線補間のデータです。

P(2001)以降の X 成分は P(1000)台で対応する動作の種類(直線/CW/CCW)です。

※デモ機プログラムは USB メモリ内の複数のファイルから 1 つをタッチパネルで選択しますが、実用的な機種切り替えとしては、ファイル名を固定して USB メモリの差し替えで対応するのが良いかと思えます。

・楕円(OVAL_CW.P2K)



```

SETP 1000 10000 20000 0 0 /* 開始点
SETP 1001 30000 20000 20000 20000 /* 通過点と中心
SETP 1002 30000 10000 0 0 /* 通過点
SETP 1003 10000 10000 20000 10000 /* 通過点と中心
SETP 1004 10000 20000 0 0 /* 到達点

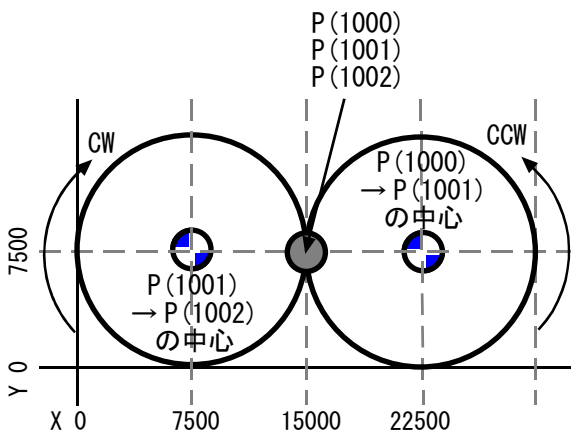
```

```

SETP 2000 1004 50 0 0 /* X=最終点番号, Y=FEED
SETP 2001 CW 0 0 0 /* P(1000)→P(1001)はCW
SETP 2002 0 0 0 0 /* P(1001)→P(1002)は直線
SETP 2003 CW 0 0 0 /* P(1002)→P(1003)はCW
SETP 2004 0 0 0 0 /* P(1003)→P(1004)は直線

```

・無限(INFINI.P2K)



```

SETP 1000 15000 7500 0 0 /* 開始点
SETP 1001 15000 7500 22500 7500 /* 通過点と中心
SETP 1002 15000 7500 7500 7500 /* 到達点と中心

```

```

SETP 2000 1002 50 0 0
SETP 2001 CCW 0 0 0 /* P(1000)→P(1001)はCCW
SETP 2002 CW 0 0 0 /* P(1001)→P(1002)はCW

```

・MPC プログラム例

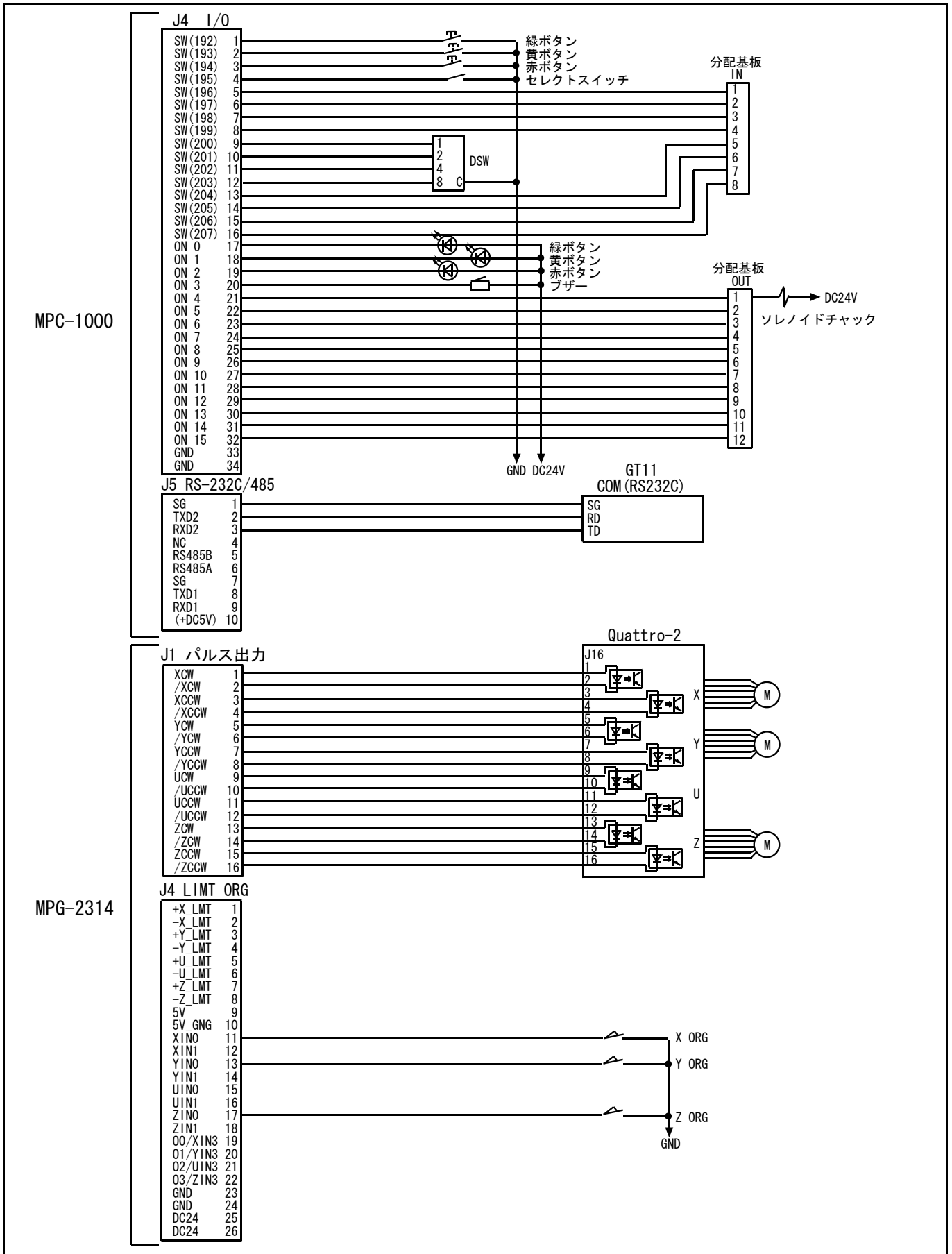
上記のデータフォーマットで動くプログラムです。

```

axis=X_A|Y_A /* 使う軸
MOVL axis P(1000) /* 作業開始点へ移動
WAIT RR(axis)==0
FEED axis Y(2000) /* スピード設定
DS_DACL /* 加減速無効
FOR pnt=1001 TO X(2000) /* P(1001)から最終点番号まで繰り返し
  MOVT axis P(pnt) X(1000+pnt) /* 連続移動 作業軸, 通過点, 動作種別
NEXT pnt
EN_DACL /* 加減速有効
WAIT RR(axis)==0

```

■信号結線

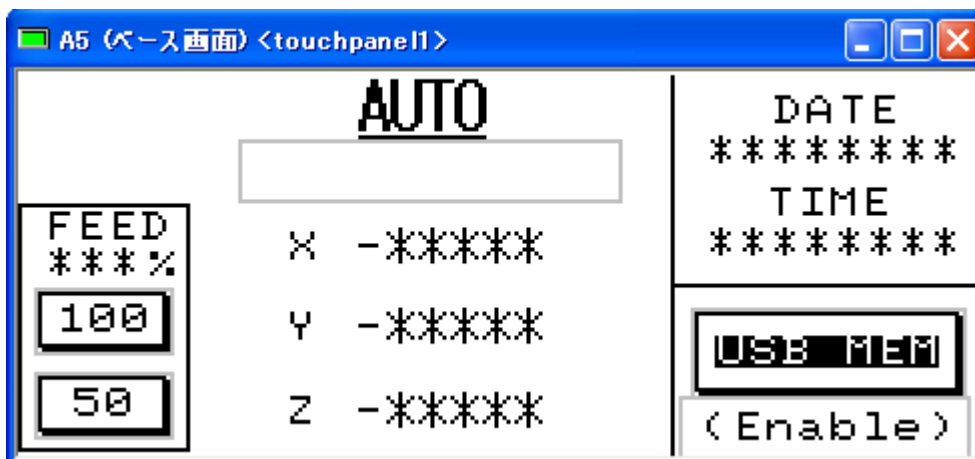


■タッチパネル部品アドレス

デモ機のタッチパネル画面切り替えは S_MBK &HA5 8 などとなります。

- ・ ページ番号 &HA5

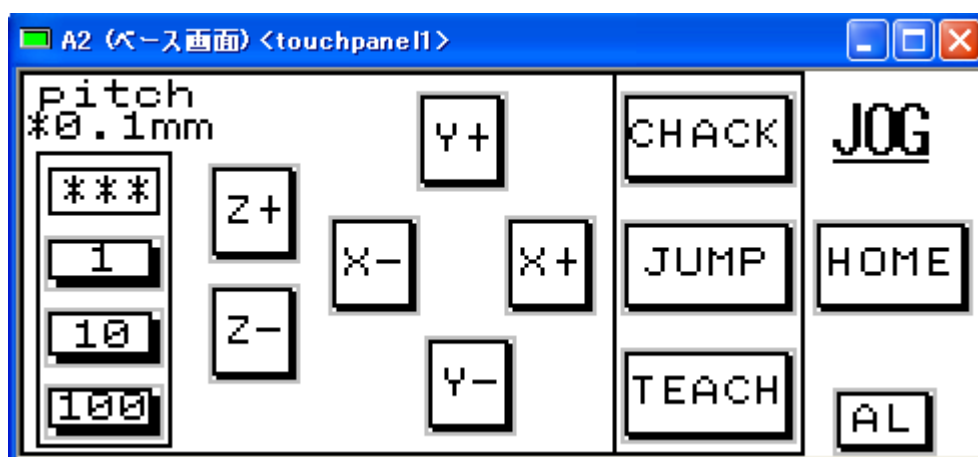
AUTO 動作時の画面。前面パネル MODE スイッチ を AUTO 側。



表示	部品名称	動作, 表示	デバイス	MPC 使用例
FEED 枠 100	機能スイッチ部品	値セット 100	出力先 DT512	速度設定 100%
FEED 枠 50	機能スイッチ部品	値セット 50	出力先 DT512	速度設定 50%
FEED 枠 ***	データ部品	表示桁数 3 DEC	参照デバイス DT512	FEED ALL_A MBK(512) 速度設定表示
AUTO 下	メッセージ部品	置換数 16	参照デバイス DT520-0	S_MBK 0 520 /* ブランク S_MBK 1 520 /*"PUSH START SW" スタートのメッセージ
X -*****	データ部品	表示桁数 5 DEC	参照デバイス DT500	S_MBK X(0) 500~Lng X 軸座標表示
Y -*****	データ部品	表示桁数 5 DEC	参照デバイス DT504	S_MBK Y(0) 504~Lng Y 軸座標表示
Z -*****	データ部品	表示桁数 5 DEC	参照デバイス DT508	S_MBK Z(0) 508~Lng Z 軸座標表示
DATE *****	データ部品	表示桁数 8 ASCII	参照デバイス DT140	S_MBK HEX\$(DATE(0)) 140 8 年月日表示
TIME *****	データ部品	表示桁数 8 ASCII	参照デバイス DT150	S_MBK HEX\$(TIME(0)) 150 8 時刻表示
USB MEM	スイッチ部品	モーメンタリ	R500	SW(75000) USB メモリ画面に移行 ON 75001 で有効 OFF 75001 で無効
(Enabled)	メッセージ部品	置換数 16	参照デバイス DT521-0	S_MBK 0 521 /* "(Enable)" S_MBK 1 521 /* "(Disable)"

・ ページ番号 &HA2

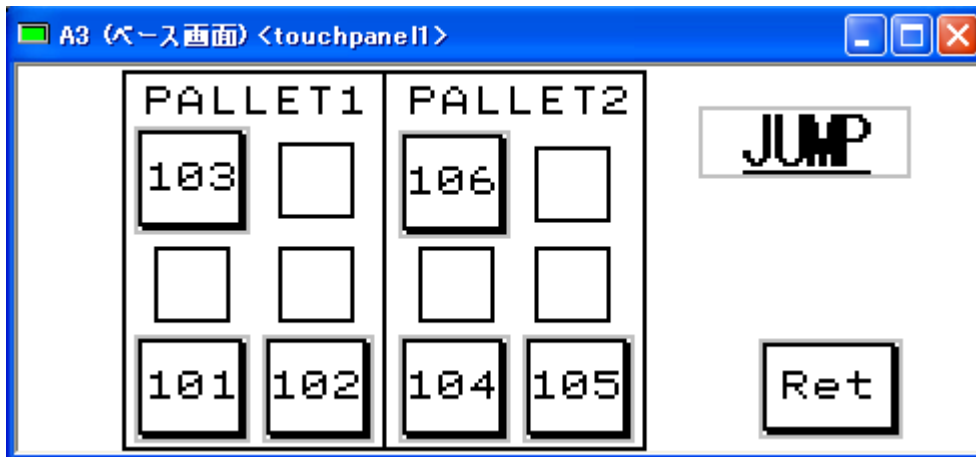
MANUAL(JOG)モードの画面。前面パネル MODE スイッチ を MANU 側。



表示	部品名称	動作	デバイス	MPC 使用例
X+	スイッチ部品	モーメンタリ	R200	SW(72000) / IN(72000~Wrd) X CW 移動
X-	スイッチ部品	モーメンタリ	R201	SW(72001) / IN(72000~Wrd) X CCW 移動
Y+	スイッチ部品	モーメンタリ	R202	SW(72002) / IN(72000~Wrd) Y CW 移動
Y-	スイッチ部品	モーメンタリ	R203	SW(72003) / IN(72000~Wrd) Y CCW 移動
Z+	スイッチ部品	モーメンタリ	R204	SW(72004) / IN(72000~Wrd) Z CW 移動
Z-	スイッチ部品	モーメンタリ	R205	SW(72005) / IN(72000~Wrd) Z CCW 移動
HOME	スイッチ部品	モーメンタリ	R206	SW(72006) / IN(72000~Wrd) 原点復帰
CHACK	スイッチ部品	モーメンタリ	R207	SW(72007) / IN(72000~Wrd) メカチャック開閉
JUMP	スイッチ部品	モーメンタリ	R208	SW(72008) / IN(72000~Wrd) JUMP 画面へ移行
TEACH	スイッチ部品	モーメンタリ	R209	SW(72009) / IN(72000~Wrd) TEACH 画面へ移行
AL	スイッチ部品	モーメンタリ	R20A	SW(72010) / IN(72000~Wrd) ALIGNMENT 画面へ移行
1	機能スイッチ部品	値セット 1	出力先 DT101	移動量設定 0.1mm
10	機能スイッチ部品	値セット 10	出力先 DT101	移動量設定 1mm
100	機能スイッチ部品	値セット 100	出力先 DT101	移動量設定 10mm
***	データ部品	表示桁数 3 DEC	参照デバイス DT101	DSW=MBK(101) 移動量表示

・ ページ番号 &HA3

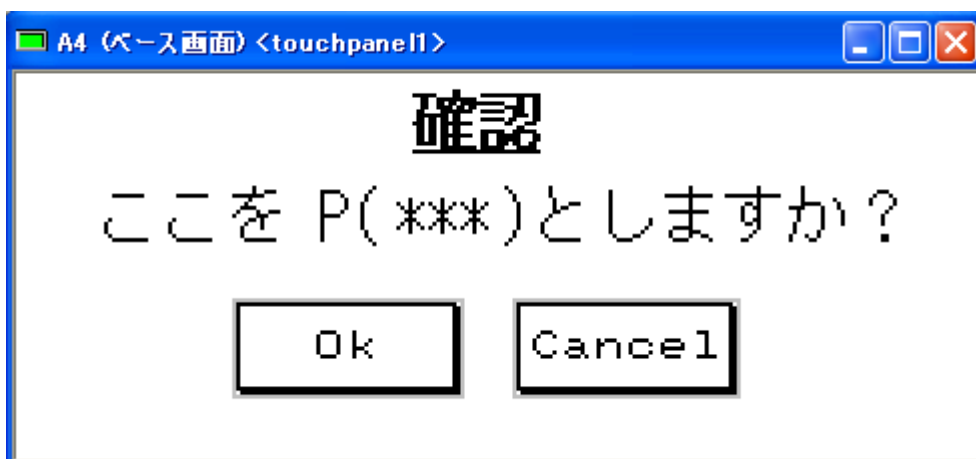
MANUAL(JOG)モードの JUMP ボタン または TEACH ボタンで移行。



表示	部品名称	動作	デバイス	MPC 使用例
101	スイッチ部品	モーメンタリ	R300	SW(73000) / IN(73000) P(101)へ移動または教示
102	スイッチ部品	モーメンタリ	R301	SW(73001) / IN(73000) P(102)へ移動または教示
103	スイッチ部品	モーメンタリ	R302	SW(73002) / IN(73000) P(103)へ移動または教示
104	スイッチ部品	モーメンタリ	R303	SW(73003) / IN(73000) P(104)へ移動または教示
105	スイッチ部品	モーメンタリ	R304	SW(73004) / IN(73000) P(105)へ移動または教示
106	スイッチ部品	モーメンタリ	R305	SW(73005) / IN(73000) P(106)へ移動または教示
Ret	スイッチ部品	モーメンタリ	R306	SW(73006) / IN(73000) JOG 画面へ戻る
JUMP 枠	メッセージ部品	置換数 16	参照デバイス DT102-0	S_MBK 0 102 /* "JUMP"表示 S_MBK 1 102 /* "TEACH"表示

・ ページ番号 &HA4

TEACH の確認画面。



表示	部品名称	動作	デバイス	MPC 使用例
Ok	スイッチ部品	モーメンタリ	R400	SW(74000) / IN(74000)
Cancel	スイッチ部品	モーメンタリ	R401	SW(74001) / IN(74000)
***	データ部品	表示桁数 3 DEC	参照デバイス DT400	S_MBK pn 400 点番号を表示

・ ページ番号 &HA9

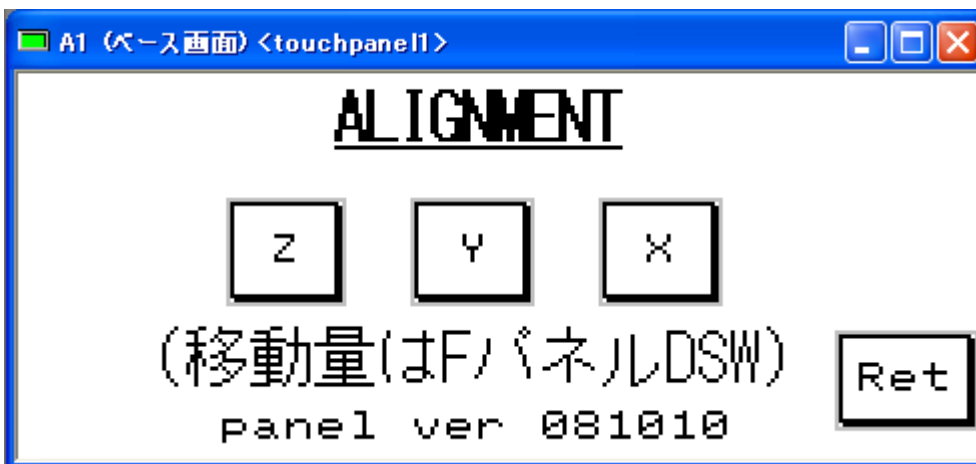
USB メモリのファイル選択・実行画面。



表示 画面上 の枠	部品名称	動作	デバイス	MPC 使用例
	メッセージ品	置換数 16	参照デバイス DT993-0	S_MBK 0 993 /* フランク S_MBK 1 993 /* "ファイル名取得中" S_MBK 2 993 /* "点データ名読み込み中" S_MBK 3 993 /* "読み込みました" S_MBK 4 993 /* "点データではありません" S_MBK 5 993 /* "ファイルを選んで下さい" S_MBK 6 993 /* "動作中"
▲	スイッチ部品	モーメンタリ	R901	SW(79001) ファイル選択
▼	スイッチ部品	モーメンタリ	R900	SW(79000) ファイル選択
PLOAD & ACTION	スイッチ部品	モーメンタリ	R903	SW(79003) ファイル読み込み・実行
Ret	スイッチ部品	モーメンタリ	R905	SW(79005) JOG 画面へ戻る

・ ページ番号 &HA1

調整用の画面。



表示	部品名称	動作	デバイス	MPC 使用例
X	スイッチ部品	モーメンタリ	R0	SW(70000) X 軸移動
Y	スイッチ部品	モーメンタリ	R1	SW(70001) Y 軸移動
Z	スイッチ部品	モーメンタリ	R2	SW(70002) Z 軸移動
Ret	スイッチ部品	モーメンタリ	R306	SW(73006) JOG 画面へ戻る

■デモ機 MPC プログラム

※デバッグなどで変更される場合があります。

```

CONST PAGING      8          /* 表示ページ設定アドレス
CONST MOTOR_PICH  80        /* 096 motor
CONST SOL_CHK     4          /* SOL CHACK PORT

/* タッチパネル関係設定
MEWNET 38400 2          /* タッチパネル MPC-2100 J5
S_MBK 0 PAGING          /* 初期画面

/* PG 関係設定
PG 0                    /* このタスクで使う MPG の宣言
SET 0 8 8 8 8          /* ティーチモード JOG 量設定
SET 1 80 80 80 80      /* 096 モータ 1/16 分割時 800 パルスで 1mm
SET 2 800 800 800 800
SET 3 8000 8000 8000 8000

/* 変数初期化
P_CNT=0                /* パレットマトリックスナンバ
SYSCLK=0                /* システムクロック

IF SW(194)=1 THEN      /* 全面パネル RESET ボタン
  GOTO *SAMPLES
END_IF
PRINT "(前面 RESET SW を押しながら RUN するとサンプルモードになります。)"

FOR I=0 TO 2            /* 前パネル 緑、黄、赤 LED 点滅
  ON I
  TIME 500
NEXT I
SET I0                  /* 出力 OFF

QUIT 28
FORK 28 *CLOCK_DISP     /* 時計、SYSCLK 表示
FORK 1 *XY03_MAIN

END                      /* TASK0 はここで終わりにする

' *****
' XY03 タスク切り替え
' *****

*XY03_MAIN              /* XY03 メカ動作
PG 0                    /* このタスクで使う MPG の宣言
S_MBK 1 101            /* JOG dist 初期値

IF SW(195)==1 THEN : GOTO *XY03_MAIN_JOG : END_IF /* MODE SW
*XY03_MAIN_AUTO
QUIT 2,3,11
PRINT "AUTO"
FORK 2 *XY03_AUTO       /* タスク 1 で AUTO モード起動
FORK 3 *XY03_COORD     /* タッチパネル座標表示

DO
  IF SW(193)==1 THEN : GOSUB *XY03_MAIN_CYCLE : END_IF /* CYCLE STOP SW が ON したら
  IF SW(195)==1 THEN : GOTO *XY03_MAIN_JOG : END_IF /* MODE SW が MANU になったら
  SWAP
LOOP

*XY03_MAIN_CYCLE
ON -1                    /* CYCLE STOP フラグ
PULSE_OUT 1 10          /* CYCLE STOP までの合図点滅
WAIT SW(193)==0
WAIT SW(-1)==0
RETURN

*XY03_MAIN_JOG
QUIT 2,3,11
PRINT "JOG"
FORK 2 *XY03_JOG
WAIT SW(195)==0        /* MODE SW -> AUTO 待ち
GOTO *XY03_MAIN_AUTO

```

自動 モード タスク

*XY03_AUTO

PG 0

S_MBK &HA5 PAGING

/* タッチパネルページ切り替え

S_MBK 100 512

/* SPEED 初期値

/* パレットポイント演算

PALLET 1 P(101) P(102) P(103) 2 3

PALLET 2 P(104) P(105) P(106) 2 3

*XY03_AUTO1

OFF SOL_CHK

/* CHACK OPEN

OFF -1, -2

/* CYCLE, RESET フラグ

ON 2

/* RESET LED

OFF 0, 1

/* START, STOP LED

ON 75001

/* タッチパネル[UBS MEM]ボタン有効

S_MBK 1 520

/* "PUSH START SW" visible

S_MBK 0 521

/* [UBS MEM]ボタン下 "(Enable)"表示

DO

IF SW(192)==1 THEN

/* START SW

BREAK

END_IF

IF SW(75000)==1 THEN

/* タッチパネル [USB MEM] SW

GOSUB *USB_MEM

/* USB MEMORY OPERATION

GOTO *XY03_AUTO

END_IF

SWAP

LOOP

IF SW(194)==1 THEN

/* RESET 押しながら

P_CNT=0

/* 作業開始位置=最初から

PR "P_CNT CLEAR"

END_IF

ON 0

/* START LED

OFF 1, 2

/* STOP, RESET LED

OFF 75001

/* タッチパネル[UBS MEM]ボタン無効

S_MBK 0 520

/* "PUSH START SW" hide

S_MBK 1 521

/* [UBS MEM]ボタン "(Disable)"表示

CNT=0

*XY03_PALLETIZE

/*DO

/* ここで loop させれば毎回 HOME

LIMZ 0

GOSUB *XY03_HOME_Z

/* Z 軸 HOME

GOSUB *XY03_HOME_XY

/* XY 軸 HOME

GOSUB *XY03_SET_SPEED

/* 速度設定

DO

/* ここで loop させれば HOME 無し

GOSUB *XY03_CYC_STOP

IF SW(-2)==1 THEN : GOTO *XY03_AUTO1 : END_IF /* SW(-2)=1 なら最初に戻る

IF P_CNT+1>6 THEN : GOTO *XY03_PALLET2 : END_IF /* 途中からハンドリング

*XY03_PALLET1

FOR PT=P_CNT+1 TO 6

/* P_CNT<>0 なら途中から作業開始

PRINT "PT=" PT

FEED ALL_A MBK(512)

/* G11 の値に速度再設定

LIMZ -6000

/* Z 軸上限設定

JUMP PL(1;PT)

/* パレット1の PT に移動

WAIT RR(ALL_A)==0

/* 移動終了待ち

ON SOL_CHK

/* CHACK 閉

TIME 100

LIMZ -5000

/* Z 軸上昇制限

JUMP PL(2;PT) AD_P(Z_A, 500)

/* 少し上から落とす

WAIT RR(ALL_A)==0

```

OFF SOL_CHK          /* CHACK 開

P_CNT=P_CNT+1

GOSUB *XY03_CYC_STOP /* CYCLE STOP 確認
IF SW(-2)==1 THEN : GOTO *XY03_AUTO1 : END_IF /* SW(-2)=1 なら最初に戻る

NEXT PT

MOVS Z_A 0
WAIT RR(ALL_A)==0
GOSUB *XY03_ENKO     /* 円弧補間動作

*XY03_PALLET2

FOR PT=P_CNT%6+1 TO 6
  PRINT "PT=" PT

  FEED ALL_A MBK(512)
  LIMZ -5000
  JUMP PL(2;PT)
  WAIT RR(ALL_A)==0

  ON SOL_CHK

  TIME 100
  LIMZ -12000
  JUMP PL(1;PT) AD_P(Z_A, 500)
  WAIT RR(ALL_A)==0
  OFF SOL_CHK

  P_CNT=P_CNT+1
  IF P_CNT>=12 THEN : P_CNT=0 : END_IF

  GOSUB *XY03_CYC_STOP
  IF SW(-2)==1 THEN : GOTO *XY03_AUTO1 : END_IF /* SW(-2)=1 なら最初に戻る

NEXT PT

MOVS Z_A 0
WAIT RR(ALL_A)==0
GOSUB *XY03_ENKO     /* 円弧補間動作

FEED ALL_A MBK(512)
MOVL 0 0 VOID VOID
WAIT RR(ALL_A)==0

CNT=CNT+1
PRINT "CNT=" CNT

LOOP

*XY03_CYC_STOP      /* サイクルストップ
IF SW(-1)==0 THEN : RETURN : END_IF /* SW(-1)がONなら CYCLE STOP
PR "CYCLE STOP"
OFF SOL_CHK

MOVS Z_A 0
WAIT RR(ALL_A)==0

PULSE_OUT VOID      /* STOP LED 点滅停止
OFF 0 2
ON 1
OFF -1 -2

DO
  IF SW(192)==1 THEN /* START SW
    PRINT "RESTART"
    ON 0 : OFF 1 2
    BREAK
  END_IF

  IF SW(194)==1 THEN /* RESET SW
    PRINT "RESET"
    ON -2 /* RESET FLAG
    BREAK

```

```

END_IF
SWAP
LOOP

RETURN

*XY03_JOG                      /* JOG モード

PG 0
OFF 0, 1, 2

DO
  S_MBK &HA2 PAGING           /* タッチパネルページ切り替え
  GOSUB *XY03_SET_SPEED

  DO                          /* SW 押下待ち
    GT=IN(72000~Wrd)          /* タッチパネル ボタン
    IF GT<>0 THEN : BREAK : END_IF
  SWAP
  LOOP

  ' PRINT GT
  SELECT_CASE GT
    CASE &H01 : AX=X_A : MD=1 : GOSUB *XY03_JOG_MV /* X+ ボタン
    CASE &H02 : AX=X_A : MD=-1 : GOSUB *XY03_JOG_MV /* X- ボタン
    CASE &H04 : AX=Y_A : MD=1 : GOSUB *XY03_JOG_MV /* Y+ ボタン
    CASE &H08 : AX=Y_A : MD=-1 : GOSUB *XY03_JOG_MV /* Y- ボタン
    CASE &H10 : AX=Z_A : MD=1 : GOSUB *XY03_JOG_MV /* Z+ ボタン
    CASE &H20 : AX=Z_A : MD=-1 : GOSUB *XY03_JOG_MV /* Z- ボタン
    CASE &H40 : GOSUB *XY03_JOG_HOME /* HOME ボタン
    CASE &H80 : GOSUB *XY03_JOG_CHACK /* CHACK ボタン
    CASE &H100 : GOSUB *XY03_JOG_JUMP /* JUMP ボタン
    CASE &H200 : GOSUB *XY03_JOG_TEACH /* TEACH ボタン
    CASE &H400 : GOSUB *XY03_ALIGN /* AL ボタン
    CASE_ELSE : PRINT "?"
  END_SELECT
  WAIT IN(72000~Wrd)==0

LOOP

*XY03_JOG_MV                    /* JOG
  GOSUB *XY03_READ_DSW_TP
  DST=DSW*MOTOR_PICH*MD
  RMVS AX DST
  WAIT RR(ALL_A)=0
  RETURN

*XY03_JOG_HOME                  /* HOME
  GOSUB *XY03_HOME_Z
  GOSUB *XY03_HOME_XY
  RETURN

*XY03_JOG_CHACK                 /* CHACK CLOSE/OPEN
  IF SW(SOL_CHK)=1 THEN
    OFF SOL_CHK
  ELSE
    ON SOL_CHK
  END_IF
  RETURN

*XY03_JOG_JUMP                 /* JUMP
  S_MBK &HA3 PAGING           /* タッチパネルページ切り替え
  S_MBK 0 102                 /* PAGE TITLE = JUMP
  LIMZ 0

  GOSUB *XY03_JOG_SEL_POINT

  IF pn<>0 THEN
    GOSUB *XY03_SET_SPEED
    FEED ALL_A 50
    JUMP P(pn)
    FEED ALL_A 100
  END_IF
  RETURN

*XY03_JOG_TEACH                /* TEACH

```

```

S_MBK &HA3 PAGING          /* タッチパネルページ切り替え
S_MBK 1 102                /* PAGE TITLE = TEACH

GOSUB *XY03_JOG_SEL_POINT

IF pn<>0 THEN
  S_MBK &HA4 PAGING        /* タッチパネルページ切り替え
  S_MBK pn 400            /* ここをP(PN)としますか?
  DO
    btn=IN(74000)
    IF btn<>0 THEN : BREAK : END_IF /* OK or CANCEL 押下でBREAK
  LOOP
  IF btn=1 THEN           /* OK SW なら現在点をPN とする
    SETP pn X(0) Y(0) U(0) Z(0)
    FSP                   /* MPC-1000 フラッシュ ROM 書き込み MAX 10000 回
  END_IF
END_IF

TIME 500
RETURN

*XY03_JOG_SEL_POINT        /* タッチパネル JUMP, TEACH の点番号選択
DO
  btn=IN(73000)
  IF btn<>0 THEN : BREAK : END_IF
LOOP

SELECT_CASE btn
CASE &H01 : pn=101        /* 101 ボタン
CASE &H02 : pn=102        /* 102 ボタン
CASE &H04 : pn=103        /* 103 ボタン
CASE &H08 : pn=104        /* 104 ボタン
CASE &H10 : pn=105        /* 105 ボタン
CASE &H20 : pn=106        /* 106 ボタン
CASE &H40 : pn=0          /* RET ボタン
CASE_ELSE : PRINT "?"
END_SELECT

RETURN

*XY03_ALIGN                /* 調整用サブルーチン
PG 0
S_MBK &HA1 PAGING        /* タッチパネルページ切り替え
MXC=0
MYC=0
MZC=0
*XY03_ALIGN1
GOSUB *XY03_SET_SPEED
OFF SOL_CHK
SELECT_CASE VOID
CASE SW(70000)==1 THEN : GOTO *XY03_ALIGN_X /* タッチパネル X ボタン
CASE SW(70001)==1 THEN : GOTO *XY03_ALIGN_Y /* タッチパネル Y ボタン
CASE SW(70002)==1 THEN : GOTO *XY03_ALIGN_Z /* タッチパネル Z ボタン
CASE SW(73006)==1 THEN : RETURN /* RET ボタン
CASE_ELSE
END_SELECT
SWAP
GOTO *XY03_ALIGN

*XY03_ALIGN_X              /* X 軸動作
GOSUB *XY03_READ_DSW_FP   /* フロントパネル DSW
IF DSW==0 THEN           /* DSW が 0 なら原点復帰
  GOSUB *XY03_HOME_XY
ELSE
  GOSUB *XY03_ALIGN_X_MV1
END_IF

MXC=MXC+1
IF MXC>5 AND SW(73006)==0 THEN /* 5 往復押しっぱなしなら連続
  GOTO *XY03_ALIGN_X
END_IF
WAIT SW(73006)==0
TIME 250
GOTO *XY03_ALIGN1

*XY03_ALIGN_X_MV1

```

```

IF X(0)==0 THEN : GOTO *XY03_ALIGN_X_MV11 : END_IF
MOVS X_A 0 /* 座標 0 に移動
WAIT RR(ALL_A)==0
RETURN
*XY03_ALIGN_X_MV11
MOVS X_A DSW*3600 /* DSW の値で移動
WAIT RR(ALL_A)==0
RETURN

*XY03_ALIGN_Y /* Y 軸動作
GOSUB *XY03_READ_DSW_FP
IF DSW==0 THEN
GOSUB *XY03_HOME_XY
ELSE
GOSUB *XY03_ALIGN_Y_MV1
END_IF

MYC=MYC+1
IF MYC>5 AND SW(73006)==0 THEN
GOTO *XY03_ALIGN_Y
END_IF
WAIT SW(73006)==0
TIME 250
GOTO *XY03_ALIGN1

*XY03_ALIGN_Y_MV1
IF Y(0)==0 THEN : GOTO *XY03_ALIGN_Y_MV11 : END_IF
MOVS Y_A 0
WAIT RR(ALL_A)==0
RETURN
*XY03_ALIGN_Y_MV11
MOVS Y_A DSW*3600
WAIT RR(ALL_A)==0
RETURN

*XY03_ALIGN_Z /* Z 軸動作
GOSUB *XY03_READ_DSW_FP
IF DSW==0 THEN
GOSUB *XY03_HOME_Z
ELSE
GOSUB *XY03_ALIGN_Z_MV1
END_IF

MZC=MZC+1
IF MZC>5 AND SW(73006)==0 THEN
GOTO *XY03_ALIGN_Z
END_IF
WAIT SW(73006)==0
TIME 250
GOTO *XY03_ALIGN1

*XY03_ALIGN_Z_MV1
IF Z(0)==0 THEN : GOTO *XY03_ALIGN_Z_MV11 : END_IF
MOVS Z_A 0
WAIT RR(ALL_A)==0
RETURN
*XY03_ALIGN_Z_MV11
MOVS Z_A DSW*-2000
WAIT RR(ALL_A)==0
RETURN

*XY03_COORD /* タッチパネル 現在位置表示
PG 0
DO
S_MBK X(0) 500~Lng /* X 軸 現在位置
S_MBK Y(0) 504~Lng /* Y 軸 現在位置
S_MBK Z(0) 508~Lng /* Z 軸 現在位置
SWAP
LOOP

*XY03_HOME_Z /* Z 軸原点復帰
ACCEL Z_A 10000 100 100 /* 原点復帰スピード
TIME 5

IF HPT(ZINO)<>0 THEN
RMLV 0 0 0 -5000

```

```

    WAIT RR(ALL_A)==0
    TIME 100
END_IF

SHOM Z_A INO_ON
HOME 0 0 0 50000
WAIT RR(ALL_A)==0
TIME 100
RMVS Z_A -500 /* 必要に応じてオフセット
WAIT RR(ALL_A)==0
STPS Z_A 0 /* Z軸ここを'0'にセット
PRINT "Z HOME"
TIME 100
RETURN

*XY03_HOME_XY /* XY軸原点復帰
ACCEL X_A|Y_A 10000 100 100 /* 原点復帰スピード
TIME 5

IF HPT(XINO)<>0 THEN /* X軸 INO がオンなら退避移動
    RMVS X_A 10000
END_IF
IF HPT(YINO)<>0 THEN /* Y軸 INO がオンなら退避移動
    RMVS Y_A 10000
END_IF
WAIT RR(ALL_A)==0
TIME 100

SHOM X_A|Y_A INO_ON
HOME -100000 -100000 0 0
WAIT RR(ALL_A)==0

TIME 100
RMVL 500 500 0 0 /* VOID は使えない
WAIT RR(ALL_A)==0
STPS 0 0 VOID VOID /* X,Y軸ここを'0'にセット
PRINT "XY HOME"
TIME 100
RETURN

*XY03_READ_DSW_TP /* JOG pitch 設定
DSW=MBK(T01) /* 3段セレクト
RETURN

*XY03_READ_DSW_FP /* フロントパネル DSW
DSW=IN(25)&&HF
RETURN

*XY03_SET_SPEED /* スピード設定 サブルーチン
ACCEL X_A|Y_A|Z_A 40000 1000 1000
FEED ALL_A 100 /* 100=MAX, 0=MIN 精密な設定は SPEED コマンド
RETURN

*XY03_ENKO /* 円弧補間動作 サブルーチン
SETP 1000 20000 25000 0 0 /* 円弧の座標作成
SETP 1001 20000 20000 0 0
SETP 1002 20000 20000 28000 20000
SETP 1003 20000 20000 12000 20000
SETP 1004 20000 15000 0 0

MOVL X(1000) Y(1000) VOID VOID
WAIT RR(ALL_A)==0
FEED ALL_A MBK(512)/2

DS_DACL
MOVT X_A|Y_A P(1001) /* 連続移動
MOVT X_A|Y_A P(1002) GCW
MOVT X_A|Y_A P(1003) CW
EN_DACL
MOVT X_A|Y_A P(1004)
WAIT RR(ALL_A)==0

RETURN

*USB_MEM /* USB メモリーから点データを読んで動く
OFF 0, 1, 2

```

```

S_MBK &HA9 PAGING          /* タッチパネルページ切り替え

GOSUB *USB_READ
  VAR RES
  IF RES==0 THEN
    RETURN
  END_IF

DO
  S_MBK 5 993                /* タッチパネル "ファイルを選んで下さい" 表示
  SELECT_CASE VOID
    CASE SW(79000)          /* タッチパネル ▼ button
      IF FILE_CNT<FILE_CNT_MAX THEN
        INC FILE_CNT 6
      ELSE
        FILE_CNT=0
      END_IF
      GOSUB *USB_CHA_CAP

    CASE SW(79001)          /* タッチパネル ▲ button
      IF FILE_CNT>0 THEN
        INC FILE_CNT -6
      ELSE
        FILE_CNT=FILE_CNT_MAX
      END_IF
      GOSUB *USB_CHA_CAP

    CASE SW(79003)          /* PLOAD & ACTION button
      S_MBK 2 993           /* タッチパネル "点データ名読み込み中"
      FN$=MBK$(994, 12)
      SERCH FN$ ". P2K"
      IF ptr_<>0 THEN
        PR "====" FN$ "===="

      ON_ERROR *USB_MEM_ERR /* エラー時処理
      USB_PLOAD FN$         /* USB メモリから読み込み
      IF err_==0 THEN      /* err_は予約変数。no error なら動作する。
        S_MBK 3 993       /* タッチパネル "読み込みました"
        TIME 1000
        S_MBK 0 993       /* タッチパネル 表示無し
        GOSUB *USB_ACT    /* 動作
      END_IF
      ON_ERROR VOID        /* VOID すると err_ も リセット

    ELSE
      S_MBK 4 993         /* タッチパネル "点データではありません"
      PRINT "invalid file"
      TIME 1000
      S_MBK 0 993         /* タッチパネル 表示無し
    END_IF

    CASE SW(79005)        /* exit from DO~LOOP
      BREAK

    CASE_ELSE
  END_SELECT
  WAIT IN(79000)==0
  SWAP
LOOP

RETURN

*USB_MEM_ERR              /* USB_PLOAD でエラーが起きたときの処理
S_MBK 9 8                 /* これをしないと 2 回目以降にページが替わらない
TIME 500                  /* &H40 ページの RET ボタンが「全画面に戻る」のせいかな?
S_MBK &H40 8              /* エラー表示画面
WAIT MBK(10)==&H40        /* ページが替わるのを待つ
PRINT "ERROR TASK=" TASKn "STEP=" err_&H00FFFFFF "CODE=" err_>>24 "MSG=" ERR$(err_)
MSG$="STEP="+STR$(err_&HFFFFFF)+" CODE="+STR$(err_>>24)+" /* >> はシフト
S_MBK "USB MEM READ ERROR " 4000 20
S_MBK MSG$ 4020 20        /* タッチパネルにエラーが発生した行番号とコードを表示
WAIT MBK(10)<>&H40        /* [RET]ボタン押下待ち
RESUME _NEXT              /* エラー発生した行の次から再開
END

```

```

*USB_READ                                     /* USB メモリからファイルデータを読む
S_MBK " " 994                               /* タッチパネル 表示クリア
S_MBK 1 993                                   /* タッチパネル "ファイル名取得中"
DIR 1000
/* ファイル情報の取得について (MPC-2100(SH7030) BL/I 1.11_21 2008/11/20)
/* DIR 1000 を実行すると次のように MBK データエリアに情報が入ります。
/* MBK(1000):ファイル数
/* MBK(1001):トータルファイル数
/* MBK(1002):トータルディレクトリ数
/* MBK(1003):USB 使用容量(MByte)
/* MBK(1004)~(1009): ファイル名 1(12 文字)
/* MBK(1010)~(1015): ファイル名 2(12 文字)
/* 以下 6 ワードずつファイル名
-----
/* 下記はダイレクトコマンドでの確認例
/* #DIR
/* 2010/02/23 10:35                158 INFINI.P2K
/* 2010/02/22 18:34                305 OVAL_CCW.P2K
/* 2010/02/22 18:31                254 OVAL_CW.P2K
/* 2010/02/22 18:25                137 SQUARE.P2K
/*                                5 個のファイル
/*                                854 バイト
/*                                0 個のディレクトリ
/* A:>
/* #DIR 1000
/* #PR MBK(1000)
/* 4
/* #PR MBK(1001)
/* 5
/* #PR MBK(1002)
/* 0
/* #PR MBK(1003)
/* 1
/* #PR MBK$(1004,12)
/* INFINI.P2K
/* #PR MBK$(1010,12)
/* OVAL_CCW.P2K
/* #PR MBK$(1016,12)
/* OVAL_CW.P2K
/* #PR MBK$(1022,12)
/* SQUARE.P2K
/* #
IF MBK(1000)==0 THEN                       /* ファイル または USB メモリ が無い
S_MBK 9 8                                   /* ※USB メモリが無くてもエラー停止しない
TIME 500
S_MBK &H40 8
WAIT MBK(10)==&H40
S_MBK "NO FILE or NO DEVICE" 4000 20
S_MBK " " 4020 20
WAIT MBK(10)<>&H40
RETURN 0
END_IF

FILE_CNT=0
FILE_CNT_MAX=(MBK(1000)-1)*6
GOSUB *USB_CHA_CAP
PRINT "Read Index " STR$(MBK(1000)) "Files"
S_MBK 0 993                                 /* タッチパネル 文字表示無し
RETURN 1

*USB_CHA_CAP                                 /* 表示ファイル名変更
DST_ADR=994                                 /* ファイル名表示先頭アドレス
CU_ADR=2052
FOR SRC_ADR=FILE_CNT TO FILE_CNT+5        /* 6 ワードコピー
S_MBK MBK(1004+SRC_ADR) DST_ADR
INC DST_ADR 1
OUT MBK(1004+SRC_ADR) CU_ADR~Wrd
INC CU_ADR 2
NEXT SRC_ADR
RETURN

*USB_ACT                                     /* USB Memory Read and MOve
S_MBK 6 993                                 /* タッチパネル "動作中"
GOSUB *XY03_HOME_Z                         /* Z 軸 HOME

```



```

GOSUB *XY03_HOME_XY          /* XY 軸 HOME
GOSUB *XY03_SET_SPEED        /* 速度設定

axis=X_A|Y_A
MOVL axis P(1000)
WAIT RR(axis)==0
MOVS Z_A -6200                /* alignment で DSW=3 でペン取り付け
WAIT RR(Z_A)==0

FEED axis Y(2000)
IF U(2000)==0 THEN           /* 0 なら連続補間, 1 なら直線
  DS_DACL
  FOR pnt=1001 TO X(2000)
    MOVT axis P(pnt) X(1000+pnt) /* 連続移動
  NEXT pnt
  EN_DACL
  WAIT RR(axis)==0
ELSE
  FOR pnt=1001 TO X(2000)
    MOVL axis P(pnt)
    WAIT RR(axis)==0
  NEXT pnt
END_IF

MOVS Z_A 0
WAIT RR(Z_A)==0

RETURN

*SAMPLES                      /* 簡単なサンプル
PG 0
PG 0 1
PG 0 2
ACCEL ALL_A 10000
FEED 100

DO
  SETIO
  S_MBK &H63 8
  QUIT 1,2
  FORK 1 *SAMPLES_MENU
  WAIT SW(195)==1
  TIME 10
  WAIT SW(195)==0
LOOP

*SAMPLES_MENU                 /* サンプルメニュー
DO
  MSG$="¥r¥n_/_/_/ キーボードで番号を入力してください _/_/_/¥r¥n¥r¥n"
  MSG$=MSG$+" 1: スイッチ入力、チャック開閉¥r¥n"
  MSG$=MSG$+" 2: X 軸パルス発生¥r¥n"
  MSG$=MSG$+" 3: ティーチング点へジャンプ¥r¥n"
  MSG$=MSG$+" 4: DSW 読み込み、ボタン押下¥r¥n"
  MSG$=MSG$+" 5: 3 軸直線補間、座標値表¥示¥r¥n"
  MSG$=MSG$+" 0: 原点復帰(最初に実行)¥r¥n¥r¥n"
  MSG$=MSG$+" (サンプル切り替えはセレクト SW を MANU→AUTO 切り替え)"
  PRINT MSG$
  INPUT CHR_C|1 A$
  SELECT_CASE A$
    CASE "1" : GOSUB *SAMPLES1
    CASE "2" : GOSUB *SAMPLES2
    CASE "3" : GOSUB *SAMPLES3
    CASE "4" : GOSUB *SAMPLES4
    CASE "5" : GOSUB *SAMPLES5
    CASE "0" : GOSUB *SAMPLES_HOME
  CASE_ELSE
    PRINT "?"
  END_SELECT
LOOP

*SAMPLES1
MSG$="¥r¥n*** SAMPLE 1 ***¥r¥n"
MSG$=MSG$+"前面 STOP スイッチでハンドを開閉します。"
PRINT MSG$
ON 1
DO

```

```

IF SW(193)==1 THEN : /* STOP SW
ON SOL_CHK : /* チャック閉
ELSE
OFF SOL_CHK : /* チャック開
END_IF
LOOP

```

*SAMPLES2

```

MSG$="¥r¥n*** SAMPLE 2 ***¥r¥n"
MSG$=MSG$+"タッチパネル X+ボタンで X 軸 CW、X-ボタンで X 軸 CCW に動きます。¥r¥n"
MSG$=MSG$+"移動量は 1, 10, 100 ボタンで変わります。¥r¥n"
MSG$=MSG$+"HOME ボタンで原点復帰します。"
PRINT MSG$
S_MBK 10 101 /* 移動量初期値
S_MBK &HA2 8 /* タッチパネルページ切り替え
DO
WAIT IN(72000)==0
WAIT IN(72000)<>0
A=MBK(101)*80
IF SW(72000)==1 THEN : RMVS X_A A : END_IF
IF SW(72001)==1 THEN : RMVS X_A A*-1 : END_IF
IF SW(72006)==1 THEN : GOSUB *SAMPLES_HOME : END_IF
WAIT RR(ALL_A)==0
PRINT "X 座標" X(0)
PRINT "動作完了"
LOOP

```

*SAMPLES3

```

MSG$="¥r¥n*** SAMPLE 3 ***¥r¥n"
MSG$=MSG$+"タッチパネル 101~106 のボタンを押すと¥r¥n"
MSG$=MSG$+"ティーチングされている点へジャンプします。¥r¥n"
MSG$=MSG$+"RET ボタンで原点復帰をします。"
PRINT MSG$
S_MBK 0 102
S_MBK &HA3 8 /* タッチパネルページ切り替え
DO
WAIT IN(73000)<>0
SELECT_CASE VOID
CASE SW(73000) : JUMP P(101)
CASE SW(73001) : JUMP P(102)
CASE SW(73002) : JUMP P(103)
CASE SW(73003) : JUMP P(104)
CASE SW(73004) : JUMP P(105)
CASE SW(73005) : JUMP P(106)
CASE SW(73006) : GOSUB *SAMPLES_HOME
CASE_ELSE
END_SELECT
PRINT "動作完了"
LOOP

```

*SAMPLES4

```

MSG$="¥r¥n*** SAMPLE 4 ***¥r¥n"
MSG$=MSG$+"前面 DSW の値を読んでタッチパネルの P() 内に入れます。¥r¥n"
MSG$=MSG$+"Ok または Cancel ボタンを押すと FTMW 画面上に Ok または Cancel を表¥示します。"
PRINT MSG$
S_MBK &HA4 8 /* タッチパネルページ切り替
DO
DO
pnt=IN(25)&&HF /* DSW 読み込み
S_MBK pnt 400 /* タッチパネル表示
IF SW(74000)|SW(74001)==1 THEN : BREAK : END_IF
LOOP
IF SW(74000)==1 THEN : PRINT "Ok ボタン" : END_IF
IF SW(74001)==1 THEN : PRINT "Cancel ボタン" : END_IF
WAIT SW(74000)==0 AND SW(74001)==0
PRINT "動作完了"
LOOP

```

*SAMPLES5

```

MSG$="¥r¥n*** SAMPLE 5 ***¥r¥n"
MSG$=MSG$+"XYZ 軸の直線補間移動をします。¥r¥n"
MSG$=MSG$+"現在座標をタッチパネルに表¥示します。¥r¥n"
MSG$=MSG$+"前面 START スイッチを押してください。"
PRINT MSG$
S_MBK &HA5 8 /* タッチパネルページ切り替
FORK 2 *XY03_DISP /* 座標表示タスク

```

```

ON 0
DO
  WAIT SW(192)=1
  GOSUB *SAMPLES_HOME
  MOVL 20000 20000 0 -10000
  WAIT RR(ALL_A)==0
  MOVL 0 0 0 0
  WAIT RR(ALL_A)==0
  PRINT "動作完了"
LOOP
*XY03_DISP
DO
  S_MBK X(0) 500~Lng : /* X座標
  S_MBK Y(0) 504~Lng : /* Y座標
  S_MBK Z(0) 508~Lng : /* Z座標
  TIME 10
LOOP

*SAMPLES_HOME
GOSUB *XY03_HOME_Z
GOSUB *XY03_HOME_XY
LIMZ 0
RETURN

*CLOCK_DISP /* タッチパネル 日時 SYSCLK 表示
GOSUB *CLOCK_SET
DO
  FORMAT "00/00/00"
  S_MBK HEX$(DATE(0)) 140 8 /* 日付
  FORMAT "00:00:00"
  S_MBK HEX$(TIME(0)) 150 8 /* 時間
  TIME 500
LOOP
*CLOCK_SET /* タッチパネル時計 -> MPC 時計
/* GT 本体環境設定 > 各種設定 1
/* 時計 GT 本体時計
/* 外部転送 する
/* 出力先頭デバイス DT3
/*
/* MBK(3) nnss nn:分 ss:秒
/* MBK(4) ddhh dd:日 hh:時
/* MBK(5) ,yyymm yy:年 mm:月
FORMAT ""
TODAY=&H20000000+MBK(5)*&H100
TODAY=TODAY+(MBK(4)/&H100)
NOW=(MBK(4)&&HFF)*&H10000+MBK(3)
SET_RTC TODAY
SET_RTC NOW
PRX TODAY NOW
RETURN

```

■USBメモリの点データ

- INFINI.P2K (∞ 反時計回り→時計回り)

```

SETP 1000 15000 7500 0 0
SETP 1001 15000 7500 22500 7500
SETP 1002 15000 7500 7500 7500
    
```

```

SETP 2000 1002 50 0 0
SETP 2001 CCW 0 0 0
SETP 2002 CW 0 0 0
    
```

- OVAL_CCW.P2K (横長楕円 反時計回り)
これは直線区間の真中開始&終点

```

SETP 1000 15000 10000 0 0
SETP 1001 20000 10000 0 0
SETP 1002 20000 30000 20000 20000
SETP 1003 10000 30000 0 0
SETP 1004 10000 10000 10000 20000
SETP 1005 15000 10000 0 0
    
```

```

SETP 2000 1005 50 0 0
SETP 2001 0 0 0 0
SETP 2002 CCW 0 0 0
SETP 2003 0 0 0 0
SETP 2004 CCW 0 0 0
SETP 2005 0 0 0 0
    
```

- OVAL_CW.P2K (縦長楕円 時計回り)
これは円弧開始&直線終点

```

SETP 1000 10000 20000 0 0
SETP 1001 30000 20000 20000 20000
SETP 1002 30000 10000 0 0
SETP 1003 10000 10000 20000 10000
SETP 1004 10000 20000 0 0
    
```

```

SETP 2000 1004 50 0 0
SETP 2001 CW 0 0 0
SETP 2002 0 0 0 0
SETP 2003 CW 0 0 0
SETP 2004 0 0 0 0
    
```

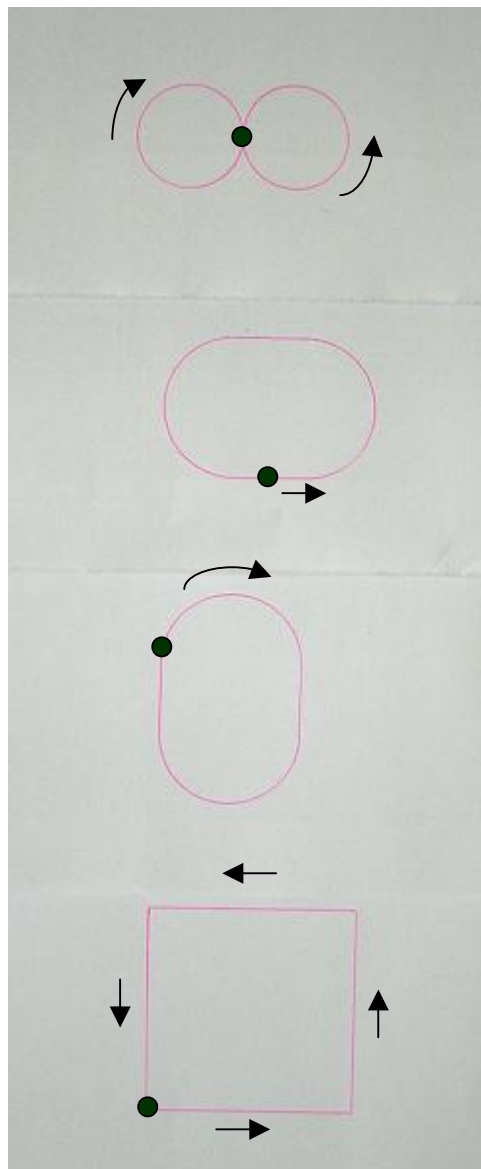
- SQUARE.P2K (正方形 連続補間無し=各辺で加減速)

```

SETP 1000 0 0 0 0
SETP 1001 30000 0 0 0
SETP 1002 30000 30000 0 0
SETP 1003 0 30000 0 0
SETP 1004 0 0 0 0
    
```

```

SETP 2000 1004 100 1 0
    
```

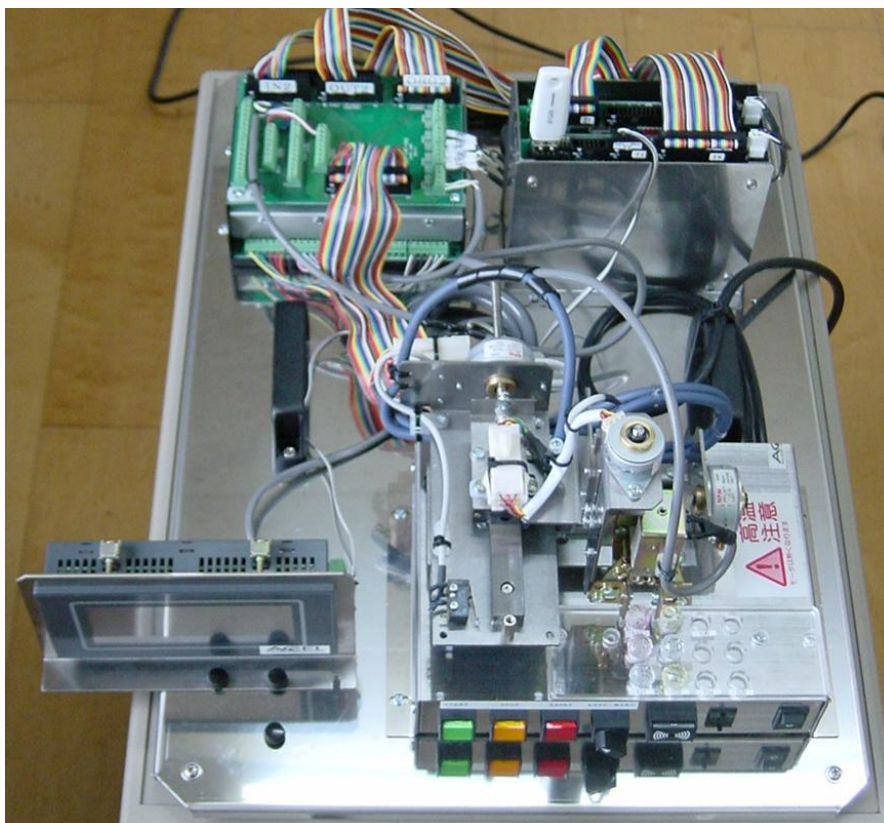


ペンを取り付けて書いてみました

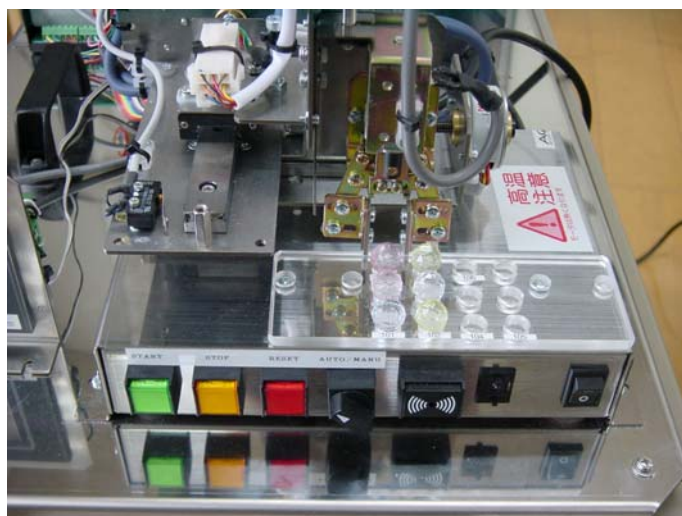
■実機写真



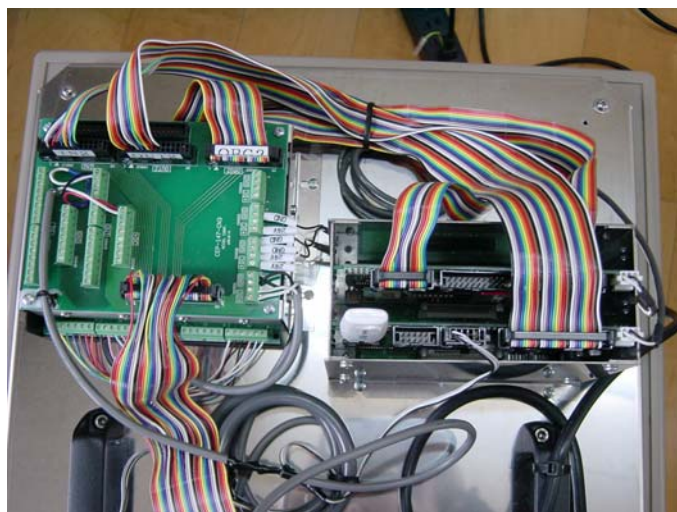
搬送時はタッチパネルを収納してください



全体



ワーク初期位置(左側2列 = PALLET 1)



MPCにUSBメモリを挿入してください

以上