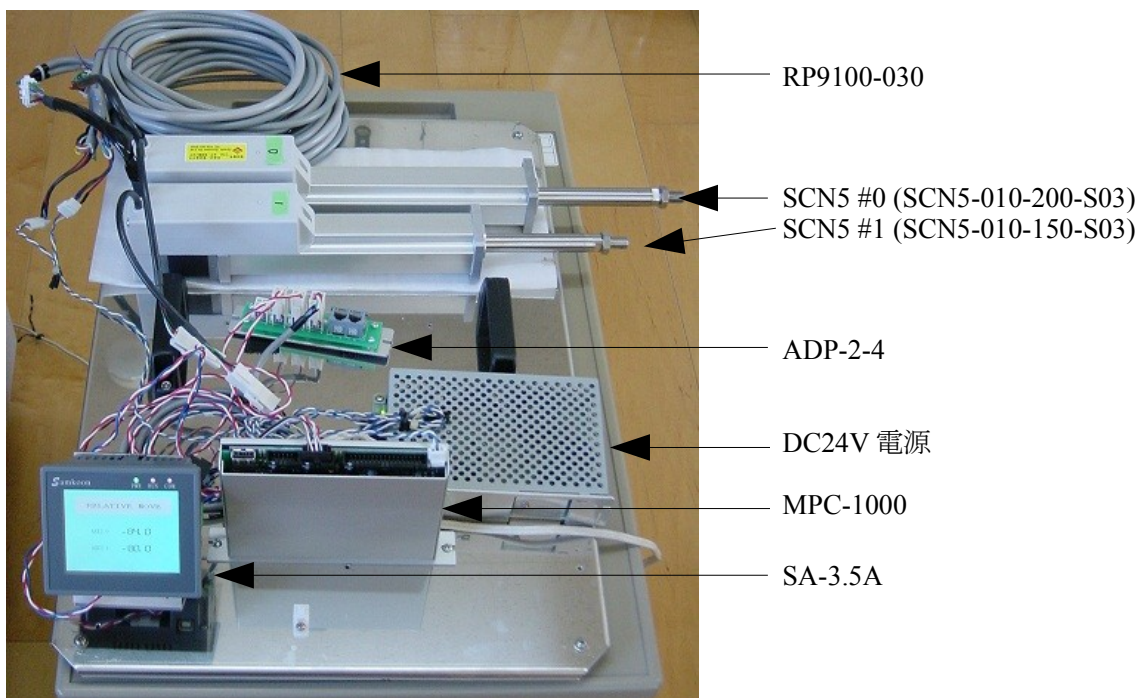
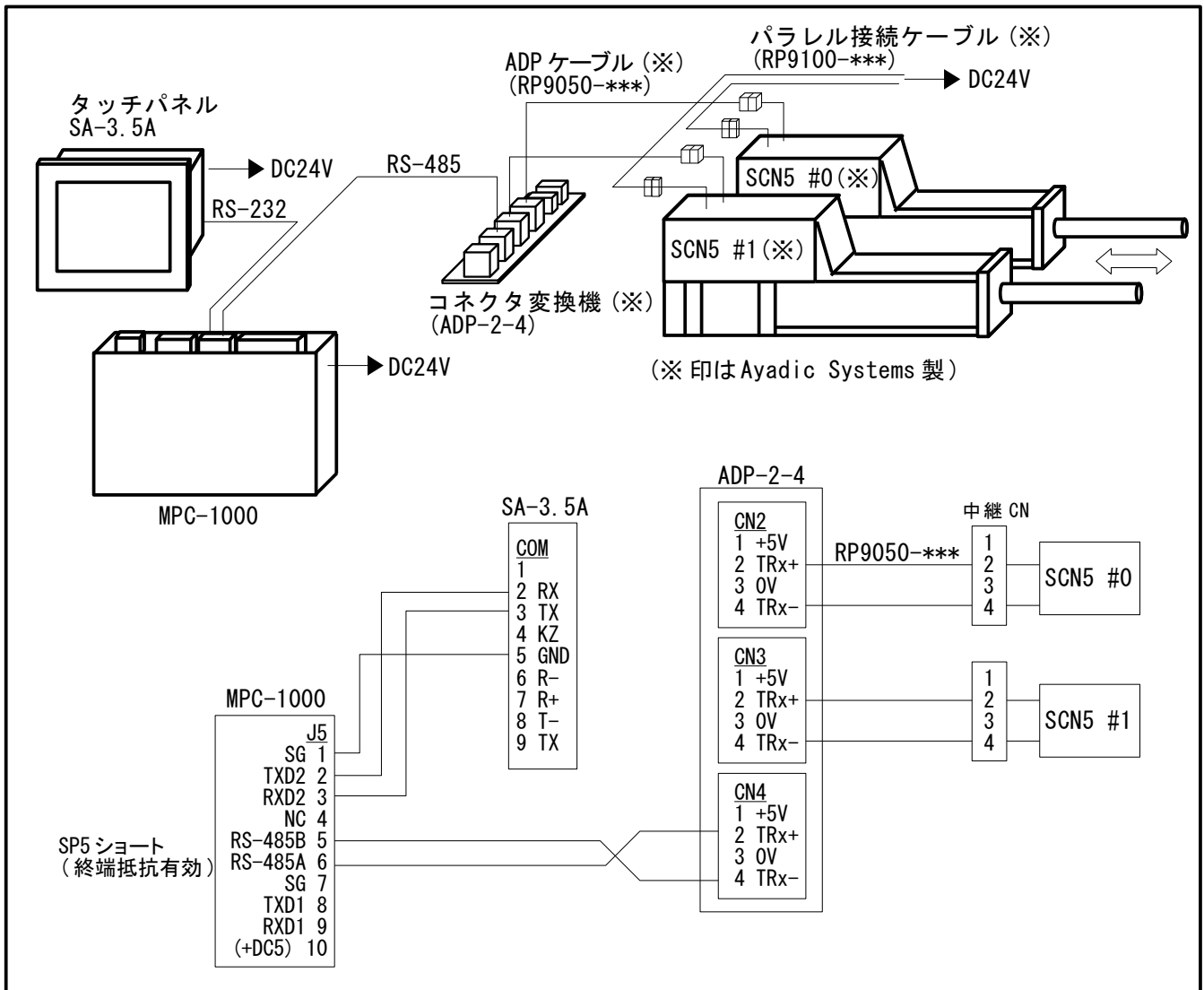


テーマ (株)ダイアディックシステムズ製メカシリンダ試用

使用機器 MPC-1000、Dyadic Systems SCN5 シリーズ、接続ケーブル 他

■機器構成



## ■概要

- メカシリンダ 2 台を RS-485 でマルチドロップ接続して制御します。
- 原点復帰、絶対座標、相対座標、SCN5 内部の点を使った PTP 移動を行います。
- 移動後の位置をタッチパネルに表示します。

## ■SCN5 シリーズ

- AC サーボアンプ一体型メカシリンダ、DC24V 電源、シリアル通信、パラレル入出力
- 最大 16 軸接続、記憶ポイント数 16 点  
詳細は 株式会社ダイアディックシステムズ ホームページ <http://www.dyadic.co.jp/> をご覧ください。

## ■参考資料

EE06426I.pdf

tbvst\_man.pdf

その他 カタログ等

Termi-BUS インターフェイス機能説明書 Ver3.00

パソコン設定ソフト (TBVST-JP) Version3.05 操作説明書

## ■その他

- SCN5 動作確認と局番の設定

最初の動作確認は「ビジュアルデータ設定ソフト(TBVST-JP)」で行いました。

局番の設定は「サーボモータ初期設定ソフト(MVST2)」で行いました。

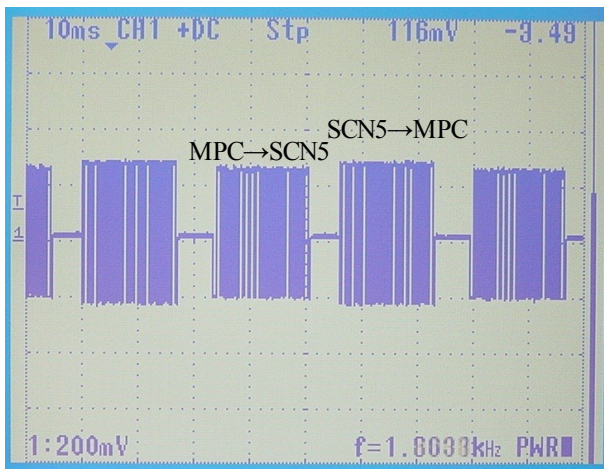
これらのソフトを使用するには RS232C/485 変換機(ADP-1)と SIO 接続ケーブル(RP9041-\*\*\*))が必要です。

- デバッグ用通信モニタ

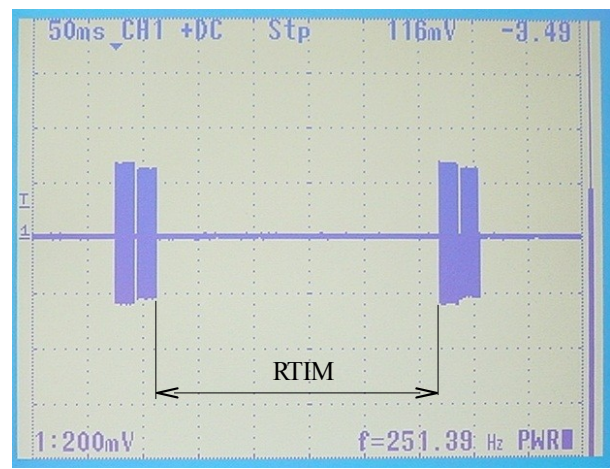
空き RS-232 ポートを用いて通信内容をモニタします。サンプルプログラムではポート番号を変数 MONI\_PORT に設定します。タッチパネルを使用するときは MONI\_PORT を 0 にします。モニタ方法の詳細はアプリケーションノート an2k-025.pdf をご参照ください。

- SIO 通信パラメータ RTIM(従局トランスミッタ活性化最小遅延時間)

RTIM のデフォルト値は 255msec になっていますが、サンプルでは稼働効率を上げるため 5msec に変更しています。この設定は共通パラメータ・記憶領域に転送後、電源再投入で有効となりますようです。なお、SIO 通信速度(BRSL)はデフォルト値 9600bps のまま変更していません。

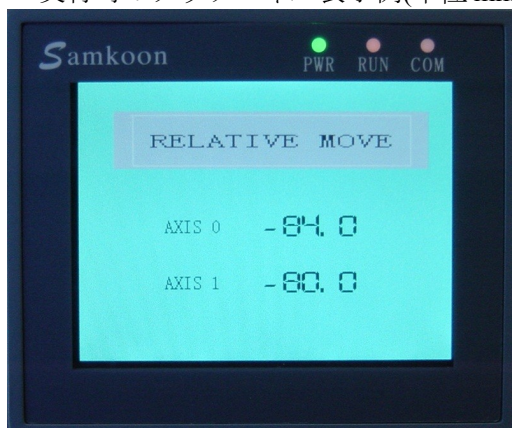


RTIM=5 のときの信号間隔(TIME レンジ 10ms)



RTIM=255 のときの信号間隔(TIME レンジ 50ms)

- 実行時のタッチパネル表示例(単位 mm)



## ■MPC-1000 サンプルプログラム サブルーチンの説明

### \*CALC\_BCC\_XFER

BCC を計算してコマンド文字列を作成し SCN5 に送信します。  
SCN5 からレスポンスを受信して BCC を計算します。

### \*CHECK\_STATUS

SCN5 のサーボアンプの内部状態を調べます。引数はチェック対象のビットです。

### \*CHECK\_STRAT\_COM

デバッグ時、プログラム停止(Ctrl+A)で送信中のコマンド文字列が途切れると次の実行時に通信が始まらないことがあります。通信不能(タイムアウト)になったらコマンドを再送信して通信を回復します。(SCN5 の電源再投入、プログラム再実行でも復帰できます)

### \*CREATE\_COMMON\_PARAM

SCN5 共通パラメータ・ウィンドウ領域を設定し、実行データ領域に転送します。

### \*CREATE\_POINT\_DATA

SCN5 ポイントデータ・ウィンドウ領域を設定し、ポイントデータ・記憶領域に転送します。

### \*HOME

1 軸原点復帰を行います。

### \*HOME\_2A

2 軸同時に原点復帰を行います。

### \*MOVE\_A

絶対座標移動を行います。引数は目標位置(例:100mm なら 1000)です。

### \*MOVE\_R

相対座標移動を行います。引数は移動距離(例:100mm なら 1000)です。

### \*MOVE\_POINT

SCN5 ポイントデータ・記憶領域のポイントデータを使って移動します。引数はポイント番号です。

### \*MOVE\_POINT\_2A

SCN5 ポイントデータ・記憶領域のポイントデータを使って移動します。引数はポイント番号です。  
2 軸同時に動作しますが補間はできません。

### \*READ\_CU\_POS

現在位置を読み込みます。

### \*READ\_MEM

SCN5 メモリのデータを読み込みます。引数はアドレスです。

### \*SET\_SPEED

移動時の速度、加速度を設定します。引数は 機能選択コード、速度指令値、加速度指令値です。

### \*WRITE\_COMMON\_AREA

SCN5 共通パラメータ・ウィンドウ領域の内容を共通パラメータ・記憶領域に転送します。

### \*WRITE\_MEM

SCN5 共通パラメータ領域にデータを書き込みます。引数はアドレス、データです。

### \*WRITE\_MEM\_A

SCN5 共通パラメータ領域にデータを書き込みます。引数はデータです。連続書き込み時に使用します。

### \*WRITE\_POINT\_AREA

SCN5 ポイントデータ・ウィンドウ領域からポイントデータ・記憶領域に転送します。引数はポイントデータ・記憶領域のポイント番号です。

## ■ MPC-1000 サンプルプログラム

```

/* Dyadic Systems Co.,Ltd SCN5 series mechanical cylinder trial
/* using AXIS0, AXIS1 with RS-485 communication

RS485_PORT=1
MONI_PORT=0 /* when it uses MEWNET, MONI_PORT=0

CNFG# RS485_PORT RS485 "9600b8pns1NONE" /* RS485 initialize

IF MONI_PORT<>0 THEN
  CNFG# MONI_PORT "9600b8pns1NONE" /* monitor for debug
ELSE
  MEWNET 38400 2 /* touch panel display
END_IF

S_MBK 8 0
S_MBK 0 800~LNg
S_MBK 0 802~LNg
S_MBK "" 810 14

PROC=0
GOSUB *CHECK_STRAT_COM

PROC=1
AXIS_NO$="0"
GOSUB *CREATE_COMMON_PARAM
AXIS_NO$="1"
GOSUB *CREATE_COMMON_PARAM

a$=""
PRINT "Rebuild POINT DATA? (Y/Any)"
INPUT CHR_C|1 TMOU|3 a$
IF (a$=="Y")|(a$=="y") THEN
  AXIS_NO$="0"
  GOSUB *WRITE_COMMON_AREA
  GOSUB *CREATE_POINT_DATA
  AXIS_NO$="1"
  GOSUB *WRITE_COMMON_AREA
  GOSUB *CREATE_POINT_DATA

  PRINT "SCN5 power on reset"
END
END_IF
PRINT "START"

DO
  GOSUB *HOME_2A

  AXIS_NO$="0"
  GOSUB *SET_SPEED 2 400 5
  GOSUB *MOVE_A -1800 /* *0.1mm
  AXIS_NO$="1"
  GOSUB *SET_SPEED 2 400 5
  GOSUB *MOVE_A -1500

  AXIS_NO$="0"
  GOSUB *SET_SPEED 2 40 5
  AXIS_NO$="1"
  GOSUB *SET_SPEED 2 40 5

  FOR main_i=1 TO 15
    AXIS_NO$="0"
    GOSUB *MOVE_R 120 /* *0.1mm
    AXIS_NO$="1"
    GOSUB *MOVE_R 100
  NEXT

  FOR main_i=1 TO 5
    GOSUB *MOVE_POINT_2A main_i
  NEXT

LOOP

END

*CALC_BCC_XFER /* calculate bcc and send a command
CMD$=AXIS_NO$+CMD$
ptr_=CMD$
sum=0
FOR calc_sum=0 TO 11

```

```

    sum=sum+ASC(PTR$(1))
    INC ptr_
NEXT
sum=&H100-(sum&&HFF)
FORMAT "00"
sum$=HEX$(sum)

CMD$=CHR$(2)+CMD$+sum$+CHR$(3)      /* make a send string
PRINT# RS485_PORT CMD$              /* send to SCN5
IF MONI_PORT<>0 THEN
    PRINT# MONI_PORT CMD$           /* output for debug
END_IF

RES$=""
DO                                     /* wait for end of response
    INPUT# RS485_PORT CHR_C|1 TMOUT|2 buf$
    IF MONI_PORT<>0 THEN
        PRINT# MONI_PORT STR_LEN|1 buf$ /* output for debug
    END_IF
    RES$=RES$+buf$
    IF buf$==CHR$(3) OR rse_<>0 THEN /* delimiter or time out
        BREAK
    END_IF

LOOP

IF rse_<>0 THEN
    IF PROC>0 THEN
        PRINT "time out axis" AXIS_NO$
        msg$="TMOUT AXIS "+AXIS_NO$
        S_MBK msg$ 810 14
    END
    END_IF
    RETURN
END_IF

ptr_=RES$+1                          /* calculate response bcc
sum=0
FOR calc_sum=0 TO 11
    sum=sum+ASC(PTR$(1))
    INC ptr_
NEXT
sum=&H100-(sum&&HFF)
sum=sum&&HFF
res_bcc=HEX(PTR$(2))
IF sum!=res_bcc THEN
    FORMAT ""
    PRINT "response BCC error" AXIS_NO$ HEX$(sum) HEX$(res_bcc)
    S_MBK "RES BCC ERR" 810 14
    END
END_IF

RETURN

*CHECK_STATUS                          /* read status
VAR CHEK_BIT
/* ~020n0000000000082~03      command
/* ~02U0n0700E080039~03      response example
/*      <stx>      stat  alrm  pi    po    bcc  <etx>
/*      ~02  U 0n  07  00  E0  80  0  39  ~03
/*      ~02  U 0n  0F  00  E0  F0  0  1C  ~03
/* ptr_  0   1 2  4   6   8  10  12  13

DO
    CMD$="n0000000000"
    GOSUB *CALC_BCC_XFER

    ptr_=RES$+4
    buf=HEX(PTR$(8))

    IF buf&CHEK_BIT!=0 THEN
        BREAK
    END_IF

LOOP

RETURN

*CHECK_STRAT_COM                       /* initial communication check

CHK_CMD$="n0000000000"
DO
    rse=0
    AXIS_NO$="0"

```

```

CMD$=CHK_CMD$
GOSUB *CALC_BCC_XFER
rse=rse+rse_
AXIS_NO$="1"
CMD$=CHK_CMD$
GOSUB *CALC_BCC_XFER
rse=rse+rse_

IF rse==0 THEN
  BREAK
END_IF
PRINT "start communication retry"
TIME 500
LOOP

RETURN

*READ_MEM /* read a common parameter in area B
_VAR ADR
/* ~02 0R4 0000000E 0 85~ 03 command example 00E: default speed
/* data bcc
/* ~02 U0R4 00002710 6B~ 03 response example 0x2710=10000
/* data bcc

FORMAT "00000000"
adr$=HEX$(ADR)
CMD$="R4"+adr$+"0"
GOSUB *CALC_BCC_XFER
ptr_=RES$+5
buf=HEX(PTR$(8))
PRINT "axis adr data" AXIS_NO$ adr$ buf

RETURN

*WRITE_MEM /* write a common parameter in area B
_VAR ADR DATA
FORMAT "00000000"
buf$=HEX$(ADR)
CMD$="T4"+buf$+"0" /* set write address
GOSUB *CALC_BCC_XFER

FORMAT "00000000"
buf$=HEX$(DATA)
CMD$="W4"+buf$+"0" /* write
GOSUB *CALC_BCC_XFER
RETURN

*WRITE_MEM_A /* write a common parameter in area B
_VAR DATA

FORMAT "00000000"
buf$=HEX$(DATA)
CMD$="W4"+buf$+"0"
GOSUB *CALC_BCC_XFER
RETURN

*SET_SPEED /* set speed and acceleration
_VAR VSEL VCMD ACMD

VCMD=VCMD*300/6 /* speed (mm/s) * 300 / 6
ACMD=ACMD*588/6 /* acceleration (G) *0.1 * 5880 / 6

FORMAT "0"
buf$=HEX$(VSEL) /* select a function
FORMAT "0000"
buf$=buf$+HEX$(VCMD)
buf$=buf$+HEX$(ACMD)
CMD$="v"+buf$+"0"
GOSUB *CALC_BCC_XFER

RETURN

*MOVE_R /* relative move
_VAR DIST

S_MBK "RELATIVE MOVE" 810 14

DIST=DIST*80/6 /* distance (*0.1mm) * 80 / 6
FORMAT "00000000"
dist$=HEX$(DIST)
CMD$="m"+dist$+"00" /* relative coordinate
GOSUB *CALC_BCC_XFER
GOSUB *CHECK_STATUS &H10 /* checking P0 bit4

```

```

GOSUB *READ_CU_POS
RETURN

*MOVE_A                               /* absolute move
_VAR DIST

  S_MBK "ABSOLUTE MOVE" 810 14

  DIST=DIST*80/6                       /* distance ( *0.1mm ) * 80 / 6
  FORMAT "00000000"
  dist$=HEX$(DIST)
  CMD$="a"+dist$+"00"                 /* absolute coordinates
  GOSUB *CALC_BCC_XFER
  GOSUB *CHECK_STATUS &H10           /* check P0 bit4
  GOSUB *READ_CU_POS
  RETURN

*HOME                                  /* move to the org, single axis
CMD$="o0700000000"                   /* home
GOSUB *CALC_BCC_XFER
GOSUB *CHECK_STATUS &H10             /* check P0 bit4

PRINT "home"
GOSUB *READ_CU_POS

RETURN

*HOME_2A                               /* move to the org, axis 0,1
S_MBK "HOME" 810 14

  AXIS_NO$="0"
  CMD$="o0700000000"                 /* home
  GOSUB *CALC_BCC_XFER
  AXIS_NO$="1"
  CMD$="o0700000000"                 /* home
  GOSUB *CALC_BCC_XFER

  AXIS_NO$="0"
  GOSUB *CHECK_STATUS &H10           /* check P0 bit4
  GOSUB *READ_CU_POS

  AXIS_NO$="1"
  GOSUB *CHECK_STATUS &H10           /* check P0 bit4
  GOSUB *READ_CU_POS

  PRINT "home"
  S_MBK "" 810 14

  TIME 500
  RETURN

*CREATE_COMMON_PARAM                  /* set the common parameters in area B
                                       /* and xfer to area C
GOSUB *WRITE_MEM &HA 2000            /* org speed 0.2 r/min
GOSUB *WRITE_MEM &HB 40              /* org acceleration 0.1 r/min/ms
GOSUB *READ_MEM &HA                  /* display for confirmation
GOSUB *READ_MEM &HB                  /* display for confirmation

GOSUB *WRITE_MEM &HC 5               /* RTIM msec default=255
                                       /* (need: write to the flash mem and power on reset)
GOSUB *READ_MEM &HC                  /* display for confirmation

CMD$="Q2000000000"                   /* xfer area B -> area C
GOSUB *CALC_BCC_XFER
GOSUB *CHECK_STATUS &H10             /* check P0 bit4

RETURN

*READ_CU_POS                           /* read the current position
GOSUB *READ_MEM &H7400

buf=buf*60/800

IF AXIS_NO$=="0" THEN
  S_MBK buf 800~Lng
ELSE
  S_MBK buf 802~Lng
END_IF

RETURN

*MOVE_POINT                             /* point move, single axis
_VAR PN

```

```

FORMAT "00"
pn$=HEX$(PN)
CMD$="Q301"+pn$+"00000"
GOSUB *CALC_BCC_XFER
GOSUB *CHECK_STATUS &H10          /* check P0 bit4
ptr_=RES$+11
PRINT "POINT" VAL (PTR$(1))

RETURN

*MOVE_POINT_2A                      /* point move, axis0,1
_VAR PN

FORMAT "00"
pn$=HEX$(PN)

buf$="GO TO P"+pn$
S_MBK buf$ 810 14

move_point$="Q301"+pn$+"00000"

CMD$=move_point$
AXIS_NO$="0"
GOSUB *CALC_BCC_XFER

CMD$=move_point$
AXIS_NO$="1"
GOSUB *CALC_BCC_XFER

AXIS_NO$="0"
GOSUB *CHECK_STATUS &H10          /* check P0 bit4
ptr_=RES$+11
PRINT "axis point" AXIS_NO$ VAL (PTR$(1))
GOSUB *READ_CU_POS

AXIS_NO$="1"
GOSUB *CHECK_STATUS &H10          /* check P0 bit4
ptr_=RES$+11
PRINT "axis point" AXIS_NO$ VAL (PTR$(1))
GOSUB *READ_CU_POS

RETURN

*READ_POINT_PARAM                   /* read the point parameters in area B
FOR read_point=&H400 TO &H41F
  GOSUB *READ_MEM read_point
NEXT
RETURN

*CREATE_POINT_DATA                  /* set the point parameters and xfer area B -> area A

GOSUB *WRITE_MEM &H401 255         /* FLGP
GOSUB *WRITE_MEM &H403 4          /* INP
GOSUB *WRITE_MEM_A 10000          /* &H404 VCMD
GOSUB *WRITE_MEM_A 512            /* &H405 ACMD
GOSUB *WRITE_MEM_A 60             /* &H406 SPOW
GOSUB *WRITE_MEM_A 255            /* &H407 DPOW

FOR create_point=&H408 TO &H41E
  GOSUB *WRITE_MEM_A 0
NEXT

IF AXIS_NO$=="0" THEN
  dist=-400
ELSE
  dist=-300
END_IF
GOSUB *WRITE_MEM &H400 dist*80/6
GOSUB *WRITE_POINT_AREA 1

IF AXIS_NO$=="0" THEN
  dist=-800
ELSE
  dist=-600
END_IF
GOSUB *WRITE_MEM &H400 dist*80/6
GOSUB *WRITE_POINT_AREA 2

IF AXIS_NO$=="0" THEN
  dist=-1200
ELSE

```



```

    dist=-900
END_IF
GOSUB *WRITE_MEM &H400 dist*80/6
GOSUB *WRITE_POINT_AREA 3

IF AXIS_NO$=="0" THEN
    dist=-1600
ELSE
    dist=-1200
END_IF
GOSUB *WRITE_MEM &H400 dist*80/6
GOSUB *WRITE_POINT_AREA 4

IF AXIS_NO$=="0" THEN
    dist=-2000
ELSE
    dist=-1500
END_IF
GOSUB *WRITE_MEM &H400 dist*80/6
GOSUB *WRITE_POINT_AREA 5

PRINT "created point data" AXIS_NO$
GOSUB *READ_MEM &H41F          /* read the flash memory write counter

RETURN

*WRITE_COMMON_AREA              /* xfer the common parameters area B -> area A
CMD$="V500000000"
GOSUB *CALC_BCC_XFER
GOSUB *CHECK_STATUS &H10      /* check P0 bit4

PRINT "wrote common parameter to flash memory in axis" AXIS_NO$
RETURN

*WRITE_POINT_AREA              /* xfer the point data area B -> area A
_VAR POINT_NO
_FORMAT "00"
point_no$=HEX$(POINT_NO)
CMD$="V501"+point_no$+"00000"
GOSUB *CALC_BCC_XFER
GOSUB *CHECK_STATUS &H10      /* check P0 bit4
RETURN

*READ_COMMON_FLASH             /* read the flash memory common parameters
CMD$="Q100000000"             /* xfer area common A -> area B common
GOSUB *CALC_BCC_XFER
FOR read_common=0 TO &H1F
    GOSUB *READ_MEM read_common /* read the common parameters in area B
NEXT
RETURN

```

## ■ サンプルプログラム 実行時のFTMW画面表示

RUN

```

axis adr data 0 0000000A 2000    ←原点復帰時の速度指令
axis adr data 0 0000000B 40      ←原点復帰時の加速度指令
axis adr data 0 0000000C 5       ←従局トランスミッタ活性化最小遅延時間
axis adr data 1 0000000A 2000
axis adr data 1 0000000B 40
axis adr data 1 0000000C 5
Rebuild POINT DATA? (Y/Any)    ←3秒以内にYまたはyキー押下でポイントデータ作成&転送
START                             ←キー入力無しで開始
axis adr data 0 00007400 0       ←原点復帰完了時の位置
axis adr data 1 00007400 0
home
axis adr data 0 00007400 -24000  ←絶対座標移動後の位置 -24000*6/80=-1800=-180mm
axis adr data 1 00007400 -20000  ←絶対座標移動後の位置 -20000*6/80=-1500=-150mm
axis adr data 0 00007400 -22400  ←相対座標移動後の位置 -168mm
axis adr data 1 00007400 -18667  ←相対座標移動後の位置 -140.0025mm
axis adr data 0 00007400 -20800
axis adr data 1 00007400 -17333
axis adr data 0 00007400 -19200
axis adr data 1 00007400 -16001
axis adr data 0 00007400 -17600
axis adr data 1 00007400 -14668
axis adr data 0 00007400 -16000
axis adr data 1 00007400 -13334
axis adr data 0 00007400 -14400

```

--End Of File--