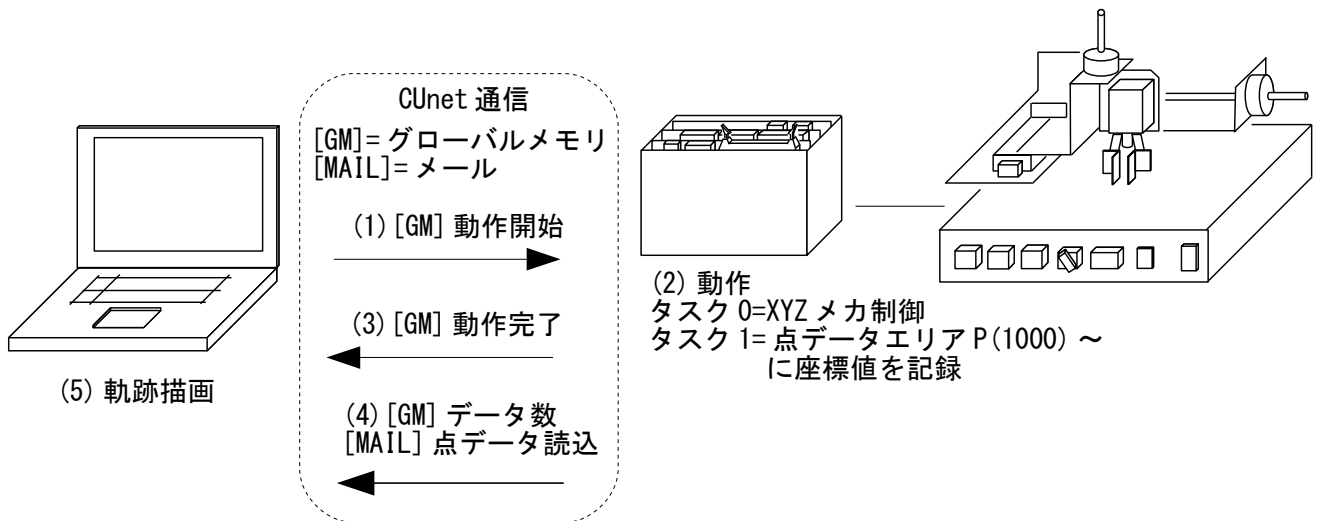
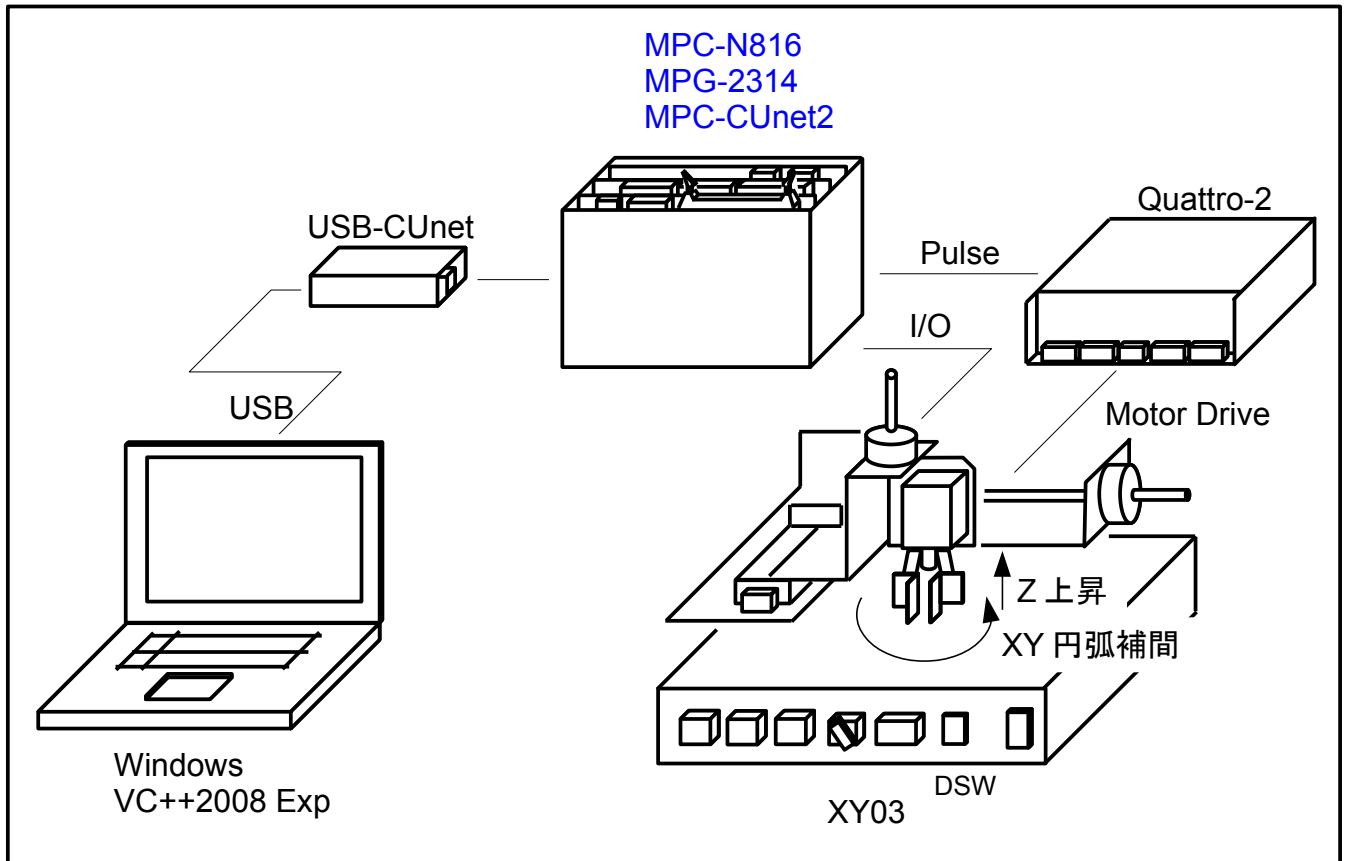


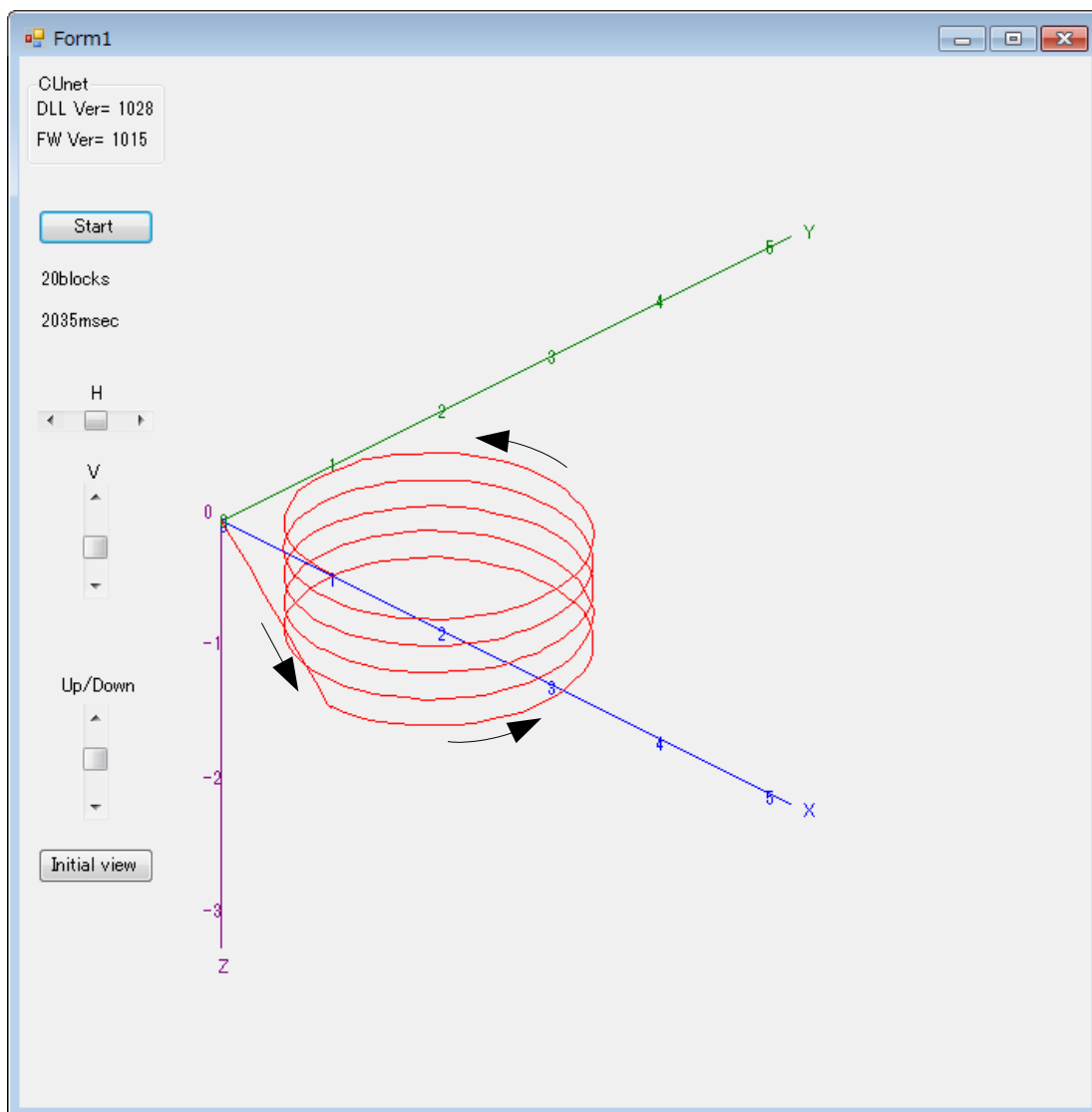
Application Note		資料作成 110425	資料番号 an2k-030
テーマ	Cunet メールを使ったXYZロボットの3次元トレース VC++編		
使用機器	MPC-N816 (MPC-2000 シリーズどれでも可)、MPG-2314、MPC-CUnet2、USB-CUnet、VC++2008Exp 他		

■機器構成



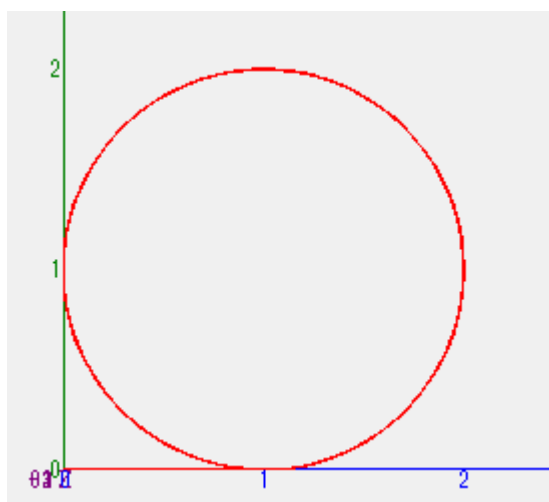
- XYZロボットの動作を3次元でトレースします。
- PC~MPC間にはCUnet通信です。動作開始・完了はグローバルメモリのBitを使い、MPCの点データエリアに記録した点データはCUnetメールで読み込みます。
- PCアプリケーションはVC++2008 Express (CLR)で作成しました。
- XYZの動作については、アプリケーションノートNo19  
<http://departonline.jp/mpc2000/ref/headline/appendix/pdf/an2k-019.pdf>  
「XY軸円弧補間とZ軸を同期させて螺旋を描く」もご参照ください。

■PCアプリケーション実行画面

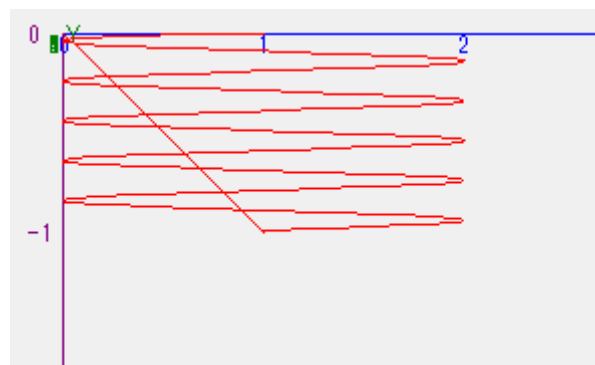


5回転実行後の画面。(矢印はメカの動作方向を表しています)  
記録点データ 298点、CUnet メール読み込みブロック数 20、読み込み時間 2035msec。

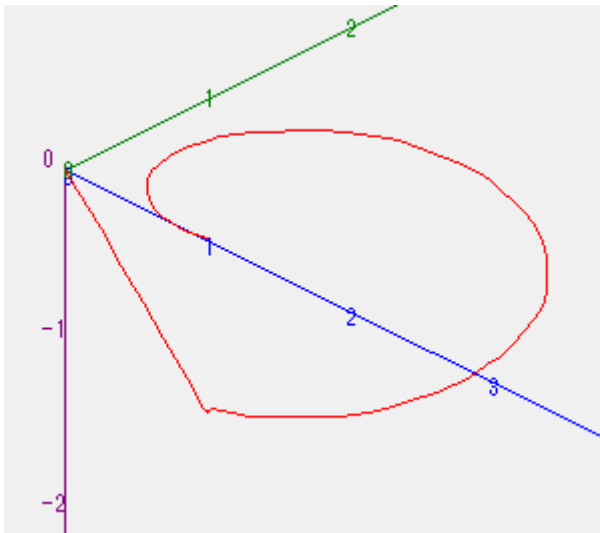
- メカ動画(4回転) <http://acceleng.jp/movie/an2k-030-2.MPG>
- H、V スクロールバーで視点を変えられます。  
スクロールバーを操作しているところの動画 <http://acceleng.jp/movie/an2k-030-1.avi>



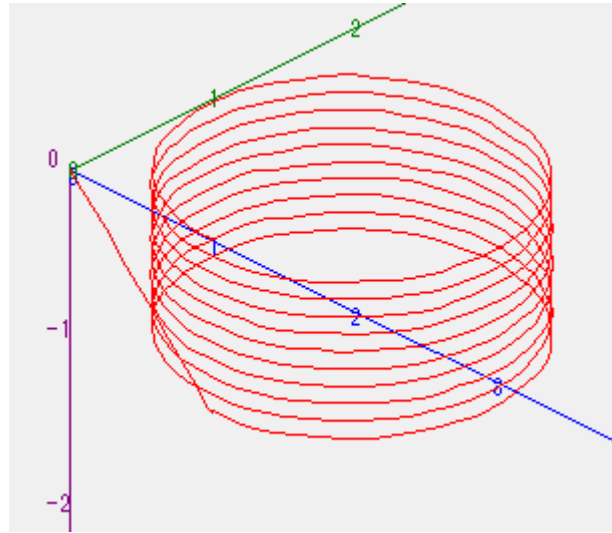
X-Y View



X-Z View



1回転時（旋回開始時にXY加速の影響がみえる）  
読み込みブロック数6、時間 611msec



10回転実行  
読み込みブロック数39、時間 3965msec

■PCアプリについて

- PCアプリケーションの作成には、石立 喬氏 ホームページ「Visual C++ の勉強部屋」  
<http://homepage3.nifty.com/ishidate/vcpp.htm> の「Visual C++ 2010 Express の易しい使い方」No13「三次元グラフィックの例題としてのFET静特性の表示」を参考にさせていただきました。ありがとうございます。
- PCアプリのソースをご希望の方はメールでご請求ください。VC++2008 Express ですが VC++2010 Express でも動作すると思います。VC++2008 Express を使った理由：VC++2010 Express は CLR で IntelliSense がないため。

■CUnetについて

- CUnetは全ての通信処理を専用デバイスが行うのでMPC、PCのプログラムでは通常のRS-232のような送受信コードは全く不要です。
- グローバルメモリはON/OFF/SW等のI/Oコマンドで操作します。PC側は専用DLLを用います。
- メール送受信はPCがホスト、MPCがスレーブとなります。MPC側はCU\_POSTコマンドによりバックグラウンドで送受信が行われるので、メールに関するコードはありません。
- CUnetメールの点データ読み込みは1ブロック当たり15点データ。1点データはXYZU4軸分。1軸は4バイト長。

■CUnetグローバルメモリ表(CUnet宣言している部分のみ)

	SA	IN/OUT バンク番号	ON/OFF/SW ビット番号	
PC	0	2000~2007	2000~2063	Bit2000: Start
	1	2008~2015	2064~2127	
	2	2016~2023	2128~2191	
	3	2024~2031	2192~2255	
	4	2032~2039	2256~2319	
	5	2040~2047	2320~2383	
	6	2048~2055	2384~2447	
	7	2056~2063	2448~2511	
	8	2064~2071	2512~2575	
	9	2072~2079	2576~2639	
	10	2080~2087	2640~2703	
11	2088~2095	2704~2767		
MPC	12	2096~2103	2768~2831	Bank2096~Lng: Data Count
	13	2104~2111	2832~2895	Bit2832: Finish
	14	2112~2119	2896~2959	
	15	2120~2127	2960~3023	
	16	2128~2135	3024~3087	
	17	2136~2143	3088~3151	
	18	2144~2151	3152~3215	
	19	2152~2159	3216~3279	
	20	2160~2167	3280~3343	
	21	2168~2175	3344~3407	
	22	2176~2183	3408~3471	
	23	2184~2191	3472~3535	

## ■MPC プログラム

```

DIM XYT(11)
/* タイムテーブル。XY 軸回転時間の SYSCLK 実測値 (mSec)
XYT(1)=2918 /* 1 回転
XYT(2)=5746 /* 2 回転
XYT(3)=8574
XYT(4)=11401
XYT(5)=14229
XYT(6)=17056
XYT(7)=19884
XYT(8)=22711
XYT(9)=25540
XYT(10)=28367

PG 0
ACCEL ALL_A 5000
CUNET 12 16 31 /* setup the CUNET Global Memory
CU_POST 30 /* initiate the Cunet mail server
OFF -1
TIME 1000
QUIT 1
FORK 1 *STORAGE_POINT /* start a recording task

DO

PRINT "Wait Start"
WAIT SW(2000)==1 /* wating start from the PC application

GOSUB *HOME

ON -1 /* start recording to task1
GOSUB *SPIRAL
OFF -1 /* stop recording to task1

TIME 500
ON 2832 /* finish flag to the PC application
WAIT SW(2000)==0
OFF 2832

LOOP

*SPIRAL

ACCEL ALL_A 30000 1000 500
MOVL 10000 0 VOID -10000 /* 作業開始点へ移動
WAIT RR(ALL_A)==0
STPS 10000 0 VOID VOID /* 現在点設定
TIME 100

REVO=IN(24)/&H10 /* 巻き数 (DSW 読み込み)
IF REVO==0 THEN : REVO=10 : END_IF
PRINT "回転数" REVO

ZL=10000 /* Z 軸目標パルス数
ZS=ZL*1000/XYT(REVO) /* Z 軸速度計算

Aset=X_A|Y_A /* 軸指定
ACCEL Aset 20000 2000 1000 /* 加速有り
/* ACCEL Aset 20000 1 20000 /* 加速無し。この場合のタイムテーブルは上記と異なる
ACCEL Z_A ZS 1 ZS /* Z 軸 最高速度=最低速度 つまり加減速無し
FEED ALL_A 100

SYSCLK=0
RMVS Z_A ZL+1000 /* Z 軸上昇 目標パルス数より多く設定
/*DS_DACL /* 減速無効 注)円弧補間の停止は即停止のみ
FOR I=1 TO REVO
RMVT Aset 0 0 CCW 0 10000 /* 現在点から相対座標 0,0 へ 0,10000 を中心として CCW 移動
PR I
NEXT
/*EN_DACL /* 減速有効。連続補間の最後が直線で減速停止する場合使う

```

```

WAIT RR(Aset)==0          /* XY動作完了待ち
STOP Z_A STP_I           /* Z軸即停止
WAIT RR(Z_A)==0          /* Z軸停止待ち

PR "時間 Z座標" SYSCLK Z(0)
TIME 500
RETURN

```

\*STORAGE\_POINT

```

PG 0

DO
  WAIT SW(-1)==1          /* 記録開始待ち
  sp=1000
  FILL X(1000) 9000 0    /* 点データクリア
  FILL Y(1000) 9000 0
  FILL U(1000) 9000 0
  FILL Z(1000) 9000 0
  OUT 0 2096~Lng

  WHILE SW(-1)==1       /* SW(-1)がオンの間記録を続ける
    SETP sp P(0)        /* 現在点をP(sp)にコピー
    TIME 50              /* 記録間隔
    INC sp

  WEND
  pc=sp-1001
  OUT pc 2096~Lng       /* 記録点データ数
  PRINT "point count " pc
LOOP

```

\*HOME

```

GOSUB *HOME_Z
GOSUB *HOME_XY
RETURN

```

\*HOME\_Z

```

ACCEL Z_A 10000 100 10000 /* speed

IF HPT(ZINO)<>0 THEN      /* check Y axis org
  RMVL 0 0 0 -5000       /* escape move
  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            /* offset
WAIT RR(ALL_A)==0
STPS Z_A 0              /* set 0
PRINT "Z HOME"
TIME 100
RETURN

```

\*HOME\_XY

```

ACCEL X_A|Y_A 10000 100 10000 /* speed

IF HPT(XINO)<>0 THEN      /* check X axis org
  RMVS X_A 10000         /* escape move
END_IF
IF HPT(YINO)<>0 THEN      /* check Y axis org
  RMVS Y_A 10000         /* escape move
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          /* offset
WAIT RR(ALL_A)==0
STPS 0 0 VOID VOID       /* ste 0
PRINT "XY HOME"
TIME 100
RETURN
```

■記録した点データ(5回転時)

```
P(1000) X= 51  Y= 0  U= 0  Z= -52
P(1001) X= 975  Y= 0  U= 0  Z= -976
P(1002) X= 2507  Y= 0  U= 0  Z= -2507
P(1003) X= 4038  Y= 0  U= 0  Z= -4038
P(1004) X= 5568  Y= 0  U= 0  Z= -5568
P(1005) X= 7098  Y= 0  U= 0  Z= -7099
P(1006) X= 8629  Y= 0  U= 0  Z= -8629
P(1007) X= 9823  Y= 0  U= 0  Z= -9823
P(1008) X= 10000  Y= 0  U= 0  Z= -10000
P(1009) X= 10000  Y= 0  U= 0  Z= -10000
P(1010) X= 10051  Y= 0  U= 0  Z= -9983
(中略)
P(1283) X= 4368  Y= 1736  U= 0  Z= -205
P(1284) X= 5388  Y= 1127  U= 0  Z= -169
P(1285) X= 6409  Y= 667  U= 0  Z= -134
P(1286) X= 7430  Y= 336  U= 0  Z= -98
P(1287) X= 8450  Y= 121  U= 0  Z= -62
P(1288) X= 9470  Y= 14  U= 0  Z= -26
P(1289) X= 10000  Y= 0  U= 0  Z= -7
P(1290) X= 10000  Y= 0  U= 0  Z= -7
(以下略)
```

--End Of File--