マイコンによるロボット制御システムの試作

電子部　仮屋一昭

Experimental Prodution of Robot Control System by Microcomputer

Kazuaki KARIYA

三菱製アーム型ロボット（MOVEMASTER EX）は，教育・研究，軽作業用として開発されたロボットで比較的安価でありながらインテリジェントコマンドを多数持ち，使いやすいロボットである。

しかし，定型的な作業を行うのであればMOVEMASTER EX本体のみで可能であるが複雑な作業を行うには外部からの制御が便利であり実用的である。今回，マイコン（日立製HD647180XOシングルチップマイクロコンピュータ）を使用して外部から制御を行った描画システムを試作した。

1．はじめに

絵（図）を描く機器としてはXYプロッター・プリンター等種々の機器があるが，どれも紙等に描くことを前提として開発されており使用方法が限定されている。

試作システムで使用したMOVEMASTER EXは基本的に軽作業等でのワークの搬送・固定等に使用されるが，アーム型ロボットのため広い空間を動くことができる（図1）。このロボットで絵（図）を描くことができれば3次元形状の物にも描くことができる。今回は3次元的形状の形状計測は行わず，2次元図が描けるシステムとした。

2．システムの概要

MOVEMASTER EXは前途したようにインテリジェントコマンドを持ちシステムの構成の方法が大別して2通りできる。

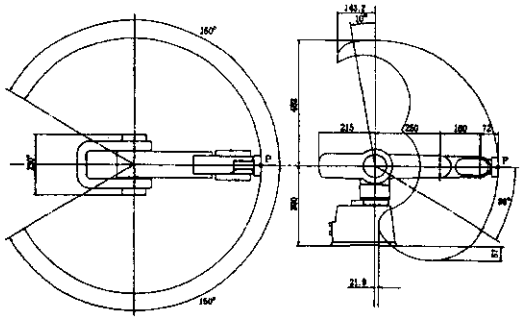

図1　ロボットの移動範囲

①　パソコンを中心とした方法

MOVEMASTER EXとパソコンを常時接続し，その動作をパソコンからの指令により行わせる方法で，MOVEMASTER EX側に用意された命令をパソコン側から逐一実行させることにより行う。従ってパソコンがシステムの中心となりロボットはパソコンの一周辺機器となる。このシステムでは複雑工程の自動化が可能であるが環境の悪い場所での使用は注意が必要である。

②　ドライブユニットを中心とした方法

外部のパソコン等でプログラムを作成して，ドライブユニットに転送し，実際の作動はドライブユニット内に転送されたプログラムにより行う。ロボットの周辺装置との信号のやりとりはI/Oポートを通して行う。このシステムでは構成が簡素で作業環境の影響を受けにくい。

今回試作したシステムは①の方法に準ずるがパソコンの代わりにシングルチップマイコン（HD647180XO）を使用している。

MOVEMASTER EXは，位置・動作制御命令，

プログラム制御命令，ハンド制御命令，入出力制御命令，RS232C読み出し命令，その他の制御命令と種々の制御命令を持ち外部からこれらの制御命令をテキスト形式で転送することで細かな制御が行える。

今回の試作においてはマイコン側でロボットの制御命令を作りRS232Cを介してロボットとコマンドのやり取りを行っている。システムの概要は図2のとおりである。赤外反射光センサ（HBCS1100）により紙を検出しマイコンが音声合成回路，ロボットを制御して描画を開始する。描画を終えると文鎮・筆を置き一連の動作を終える。

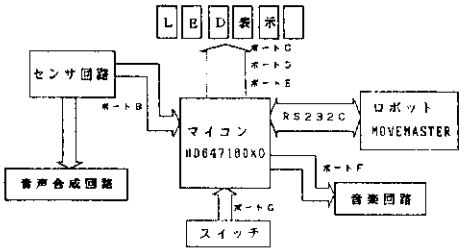

図2　システムの概要

3．描画の方法

MOVEMASTER EXには，ワークの移動等を目的とした数多くのコマンドがある。しかし，これらのコマンドは2点間を移動するときの軌跡についてはそれほど考慮されておらず，XYプロッターのような描画のためのコマンドはない。例えば，始点Aから終点Bにロボットのハンドの先端を移動させると直線ではなく弧になる（図3）。直線を描くためには点A・B間の直線上に補正点を数点（A'，B'．．．）置きハンドの移動を細かくし始点A・A'間，A'・B'間と順次描くことで直線に近づけることができる。円・楕円についても同様な方法で描いた。補正点を多く置くほど描画はきれいになるが描画時間が長くなる。円等を描くとき各補正点をデータとしてマイコン内に持つとデータ数が膨大になり，また，絵の変更が大変な作業になる。使用したHD647180XOは内部ROMとして16Kバイトしかなく大きなデータを持つことができない。

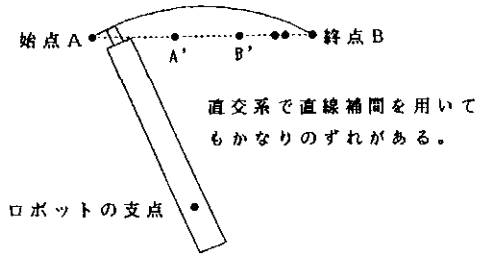

図3　ハンド先端の軌跡

円・楕円の描画は円の中心座標・半径・始点と終点の角度等で描くとデータ数としてはかなり少なくなり，絵の変更も行いやすい。本システムでは描画を開始する基準点を一点置きこの基準点をもとに各座標を計算しロボットに制御命令として転送している。

4．ハードウェア及びソフトエア

① ハードウェア

描画システムで使用したHD647180XO（8 BIT）は，Z80CPUと上位コンパチビリティが保たれており豊富な開発環境が整っている。

2チャンネルの非同期シリアルポート持ち，外部にMAX235を付加するのみでRS232Cを構成でき，内部ROM・RAMのみのシステムではほとんどの端子をパラレルI/Oとして使用できる。（PA～PGまでの7ポート・PAはRS232Cと兼用・PGは入力専用）

今回のシステムではHD647180XOを作動モード0（外部メモリを使用しないシングルチップモード）で使用し，各ポートは次のとおりである。

PB→光センサとの通信
PC・PD・PE→LED表示用
PF→音楽のON/OFF
PG→モードキー（絵の選択・その他）

シリアルポート（ASCI）→2チャンネルの内1チャンネルを使用しロボットの制御・データの転送に使用している。

描画は用紙を絵書きテーブルに置いた時開始す

る。用紙の認識に赤外反射光センサ（HBCS1100）を使用し光量の差による誤動作を少なくしている。（朝，夕方，天気の良悪しで光量が異なりCdsセンサでは誤動作が頻繁に起きた。）HBCS1100は発光素子と受光素子を内部に持ち発光素子にパルスを与えることにより用紙からの反射で受光素子にパルスが生じる。発光素子と受光素子のパルスをEX－ORしたものを積分してコンパレータに通している。（図4）。音声合成，音楽については市販のボードを使用した。

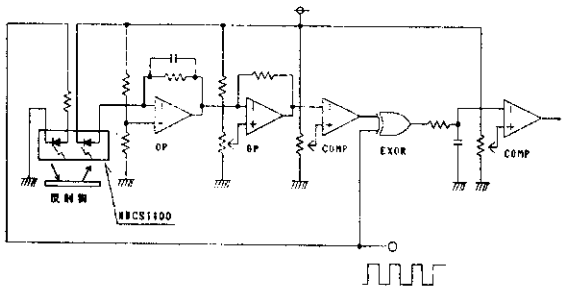

図4　センサー回路図

②　ソフトウェア

HD647180XOはZ80と上位互換があるためHITECH－C（Z80用）を使用しアセンブラにはHITECH－Cに付属のZASを使用した。ZASはHD64180用の命令が使用できるためCコンパイラで記述できない部分をインラインアセンブラで記述した。

HD647180XOは汎用レジスタ・専用レジスタを除くすべてのレジスタをI／O空間に持ちこれらのレジスタはI／O空間上で128バイトを占めており，リセット直後は0000Hから007FHに配置されている。この空間は直接HITECH－Cでアクセスできないのでインラインアセブラで処理している。

絵のデータは座標の集合体でなく円，楕円，線．．とそれぞれ命令を持たせロボットに転送する時点で各座標を求めている。そのため座標の集合体に比べかなりデータ数が少なくなった。描画の精度を上げるには各描画命令の補正点を増やせばよい。しかし今回はそれぞれの描画時間が3～4分を目安に書き終えるようにしたため円では12度ごとに補正点を設けてある。

各ポート初期設定及びプログラムの主要な部分は下のとおりである。

Ⅰ　コンパイル方法

```
zc -v -c -s %1.c >%1.err
zas %1.as
link z -m -ptext=0000h, data,
    bss=fe00h romsys.obj %1.obj
    zlibf.lib zlibc.lib
objtohex 1.obj 1.hex
```

Ⅱ　Start up（アセンブラ）

```
    psect text
    global main
stack equ Offffh ;stack top
start:
    ld      hl, stack
    ld      sp,hl
    call    main
    jp      start
    psect   data
    psect   bss
    end
```

Ⅲ　各ポート及びASCIの初期設定

```
void initport (void)
        /*ポートの入出力の初期値設定*/
{
#asm
    ld a, 00h
    out0 (71h), a
    ;ポートBを入力に設定
    ld a, Offh
    out0 (72h), a
    ;ポートCを出力に設定
    out0 (73h), a
    ;ポートDを出力に設定
    out0 (74h), a
```

```
        ;ポートEを出力に設定
        out0 (75h), a
        ;ポートFを出力に設定
#endasm
}
void initrs (void)
/*ASCI コントロールレジスタA. Bの初期値設定*/
{
#asm
        ld a, 01h
        out0 (02h), a
;ASCI コントロールレジスタBを00000001bに設定
        ld a, 63h
        out0 (00h), a
;ASCI コントロールレジスタAを01100011bに設定
        in0 a, (08h)
#endasm
}
```

(資料参照)

5. おわりに

一連の動作をマイコンの代わりにパソコンに置き換えて行ったところパソコン (PC9801 VX CPU80286　16BIT) が若干処理が速かった。しかし，パソコンに比べマイコンの方がかなりコンパクトになりマイコン周辺の制御も行いやすい。

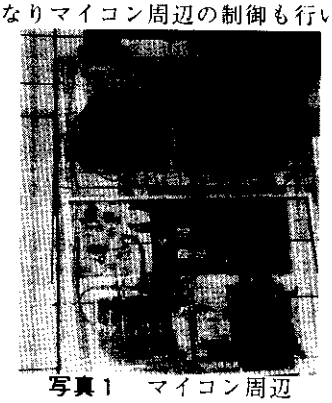

写真1　マイコン周辺

ワークの搬送に比べ絵の描画はロボットのハンド先端の動きに高い精度を要求する。ハンド先に微妙な振動があれば描かれた絵に反映され，波のある線となる。MOVEMASTER EXにおいてもハンド先に振動が起こることがありボールペン等で書くとかなり絵に振動が反映されたため筆ペンを用いて描画している。また，筆の向き・筆圧についても制御してやるとかなりの精度で描画できるのではないかと思われる。

参考文献

1) 日立製作所：HD647180Xハードウェアマニュアル
2) 上廣孝幸：プロセッサ，**26**，28 (1987)
3) HI－TECH C　ユーザーズマニュアル
4) 池田央：センサ・インターフェジング，**4**，41 (1984)
6) 三菱電機：MOVEMASTER EX　取扱説明書

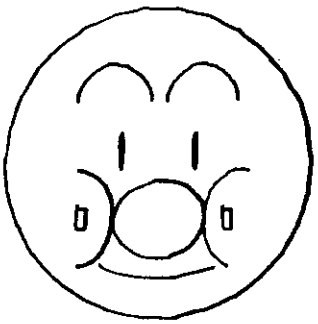

図5　ロボットが描いた絵

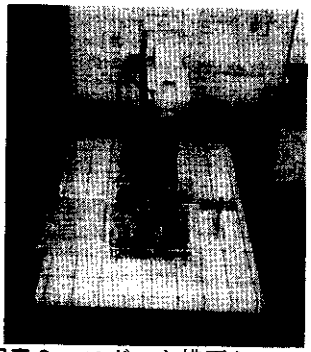

写真2　ロボット描画システム

<プログラムリスト（本文の続き）>

```
Ⅳ．各ポート及びASCIの入出力関数
int inpb(void)  /*ポートB入力*/
{
#asm
   in0 a,(61h)
   ld l,a
   ld h,0h
#endasm
}
void outpc(int dat)  /*ポートC出力*/
{
int i=dat;
#asm
   ld l,(ix+6)
   ld h,(ix+7)
   ld a,l
   out0(62h),a
#endasm
}
void outpd(int dat)  /*ポートD出力*/
{
int i=dat;
#asm
   ld l,(ix+6)
   ld h,(ix+7)
   ld a,l
   out0(63h),a
#endasm
}
void outpe(int dat)  /*ポートE出力*/
{
int i=dat;
#asm
   ld l,(ix+6)
   ld h,(ix+7)
   ld a,l
   out0(64h),a
#endasm
}
void music_on(void) /*ポートFより音楽スタート*/
{
#asm
   ld a,1
   out0(65h),a
#endasm
}
void music_off(void) /*ポートFより音楽ストップ*/
{
#asm
   ld a,0
   out0(65h),a
#endasm
}
void outtdr0(int dat) /*ASCI トランスミッタ・データ・レジスタ出力*/
{
int i=dat;
#asm
   ld l,(ix+6)
   ld h,(ix+7)
   ld a,l
   out0(06h),a
#endasm
}
int readrdr(void)  /*ASCIレシーブ・データ・レジスタの読み取り*/
{
#asm
   in0 a,(08h)
   ld l,a
   ld h,0h
#endasm
}
void outcntla0(int dat)  /*ASCI コントロールレジスタAにデータを設定*/
{
int i=dat;
#asm
   ld l,(ix+6)
   ld h,(ix+7)
   ld a,l
   out0(00h),a
#endasm
}
int instat(void)  /*ASCI ステータス・レジスタの入力*/
{
#asm
   in0 a,(04h)
   ld l,a
   ld h,0h
#endasm
}
void clsstat(void) /*ステータスレジスタＤＣＤ０のクリア*/
{
#asm
```

```
    ld a,05h
    out0(04),a
#endasm
}
int incntlb0(void)  /*ASCI コントロールレジスタＢの入力*/
{
#asm
    in0 a,(02h)
    ld l,a
    ld h,0h
#endasm
}
int incntla0(void)  /*ASCI コントロールレジスタＡの入力*/
{
#asm
    in0 a,(00h)
    ld l,a
    ld h,0h
#endasm
}
int inpg()      /*ポートG入力*/
{
#asm
    in0 a,(66h)
    ld l,a
    ld h,0h
#endasm
}
```

Ｖ．ＲＳ２３２Ｃ入出力関数

```
#define CR      0x0d
#define LF      0x0a
void outrs(char *ptr)  /*rs232cに文字列を出力する*/
{
char *p;
int d;
   p=ptr;
   while(*p !=0){
       if ((instat() & 0x02) == 2) {
           outtdr0(d=(int)(*p++));
           outpe(htod(d));
        }
    }
   for(;;){
       if ((instat() & 0x02) == 2) {
           outtdr0(CR);
           outpe(htod(CR));
           break;
        }
    }
}
```

Ⅵ．ロボットの初期化関数

```
void robot_init(void)   /*ロボットの初期化*/
{
   outrs("NT");
   outrs("SP 9");
   outrs("MO 50");
   outrs("OG");
   outrs("MO 50");
}
```

Ⅶ．主要関数

```
#define PI      3.14159
#define START   1
#define MID     2
#define END     3
void circle(int ix,int iy,int iz,int hankei,int kakudo_haba)  /*円を書く*/
{
   fan_oval(ix,iy,iz,hankei,hankei,0,360,kakudo_haba);
}
void oval(int ix,int iy,int iz,int ixa,int iyb,int kakudo_haba)  /*だ円を書く*/
{
   fan_oval(ix,iy,iz,ixa,iyb,0,360,kakudo_haba);
}
void fan_circle(int ix,int iy,int iz,int hankei,int st,int ed,int kakudo_haba)
/*円の弧を書く*/
{
   fan_oval(ix,iy,iz,hankei,hankei,st,ed,kakudo_haba);
}
void fan_oval(int ix,int iy,int iz,int ixa,int iyb,int st,int ed,int kakudo_haba
)       /*だ円の弧を書く*/
{
double xs,ys,xas,ybs,xp,yp,ii;
int i;
   ix+=x;  iy+=y;  iz+=z;
   xs=ix;  xs/=10;
   ys=iy;  ys/=10;
   xas=ixa;        xas/=10;
   ybs=iyb;        ybs/=10;

   if (st>ed){
       for(i=st;i<=360;i+=kakudo_haba){
           ii=i;
           xp=(xas*cos(ii/180*PI)+xs)*10;
           yp=(ybs*sin(ii/180*PI)+ys)*10;
           if (i==st) out_rs_data((int)xp,(int)yp,iz+80);
```

```
            out_rs_data((int)xp,(int)yp,iz);
        }
        for(i=0;i<=ed;i+=kakudo_haba){
            ii=i;
            xp=(xas*cos(ii/180*PI)+xs)*10;
            yp=(ybs*sin(ii/180*PI)+ys)*10;
            out_rs_data((int)xp,(int)yp,iz);
        }
    }else {
        for(i=st;i<=ed;i+=kakudo_haba){
            ii=i;
            xp=(xas*cos(ii/180*PI)+xs)*10;
            yp=(ybs*sin(ii/180*PI)+ys)*10;
            if (i==st) out_rs_data((int)xp,(int)yp,iz+80);
            out_rs_data((int)xp,(int)yp,iz);
        }
    }
    out_rs_data((int)xp,(int)yp,iz+80);
}
void rbox(int stx,int sty,int box,int boy)   /*枠を書く*/
{
    point_set(stx,sty,0,START);
    point_set(stx,boy,0,MID);
    point_set(box,boy,0,MID);
    point_set(box,sty,0,MID);
    point_set(stx,sty,0,END);
}

void rline(int stx,int sty,int box,int boy)  /*線を引く*/
{
    point_set(stx,sty,0,START);
    point_set(box,boy,0,END);
}

void point_set(int ix,int iy,int iz,int ptype) /*点の座標を書く*/
{
    ix+=x;  iy+=y;  iz+=z;
    if (ptype==START) out_rs_data(ix,iy,iz+80);
    out_rs_data(ix,iy,iz);
    if (ptype==END) out_rs_data(ix,iy,iz+80);
}
void out_rs_data(int ix,int iy,int iz)   /*ポジションデータを転送*/
{
double rool,dx,dy;
char pr[50],pp[10];
    dx=ix;  dx/=10;
    dy=iy;  dy/=10;
    if (dx>0.0) rool=(90-((atan(dy/dx))/PI)*180.0)-7.1;
    else if (dx==0.0) rool=90.0-7.1;
    else rool=((90-((atan(dy/dx))/PI)*180.0)-7.1)-180;
         rool*=10;

    strcpy(pr,"MP ");
    itoa10(pp,ix);
    strcat(pr,pp); strcat(pr,",");
    itoa10(pp,iy);
    strcat(pr,pp); strcat(pr,",");
    itoa10(pp,iz);
    strcat(pr,pp); strcat(pr,",");
    itoa10(pp,p);
    strcat(pr,pp); strcat(pr,",");
    itoa10(pp,(int)rool);
    strcat(pr,pp);

    outrs(pr);
}

/* ロボットの制御命令テキスト形式*/
"GF 0"
"PD 50,-25.1,+271.4,+230.0,-87.1,-9.7".

Ⅷ. 絵のデータ
void write_anpan(void) /*アンパンマンの絵を書く*/
{
    oval(-30,660,0,548,515,12);              /* 顔輪郭*/
    fan_circle(130,465,0,130,180,360,12);    /* 左眉*/
    fan_circle(-182,465,0,130,180,360,12);   /* 右眉*/
    oval(100,625,0,6,60,12);                 /*左目*/
    oval(-155,625,0,6,60,12);                /*右目*/
    oval(-30,850,0,160,125,6);               /*鼻*/
    fan_oval(260,850,0,130,160,90,270,12);   /* 左ほっぺ*/
    fan_oval(-310,850,0,130,160,255,80,12);  /* 右ほっぺ*/
    fan_circle(-30,380,0,660,72,108,6);      /*口を書く*/
    rbox(260,817,228,882);                   /*左ほっぺの艶*/
    rbox(-253,819,-284,883);                 /*右ほっぺの艶*/
}
```