

～ H 8 マイコンのプログラム～

マイコンは秋月電子の A E - 3 0 6 7 F を使用しています。5 V で書き込みができるので便利です。R 8 をショートしモード 7 で使用しています。開発には C 言語を使用しました。サーボモータ制御のための PWM 信号は汎用の I / O ピンの ON、OFF によって発生させています。マイコンの動作をチェックするためにスタート時に文字を送信させています。ハイパーターミナル等で確認しながら作業すると良いと思います。

```
/*-----*/
/* Motion control */
/* by Ookosi */
/*-----*/
/* I/Oアクセス用インクルードファイル */
/*-----*/
#include "3067f.h" /* C言語に付属するヘッダーファイル */
#include "string.h"
#include "stdio.h"
#include "stdarg.h"
#include "stddef.h"
#include "stdlib.h"
/*-----*/
/* 全領域で使用する変数の宣言 */
/*-----*/
int pt;
char buf[100];
/*-----*/
/* H8の初期化 */
/*-----*/
void inih8(void)
{
PA.DDR=0xff;
PB.DDR=0xff;
}
/*-----*/
/*シリアル通信の設定 */
/*-----*/
typedef enum{br9600=64,br38400=15}BAUDRATE;
```

```

void H8SCI_INIT(char b)
{
    int i;
    SCI1.SCR.BYTE=SCI1.SMR.BYTE=0;
    SCI1.BRR=b;
    for (i=0; i<1000; i++);
    SCI1.SCR.BYTE=0x30;
    SCI1.SSR.BYTE;
    SCI1.SSR.BYTE=0x80;
}
/*-----*/
/* 送信 */
/*-----*/
void H8SCI_OUT_DATA(char c)
{
    while(SCI1.SSR.BIT.TDRE==0);
    SCI1.TDR=c;
    SCI1.SSR.BIT.TDRE=0;
}
void H8SCI_OUT_STRING(char*st){
    while(*st) H8SCI_OUT_DATA(*st++);
}
/*-----*/
/* パルス幅時間待ち */
/*-----*/
void pwm (pt)
{
    ITU0.GRA=pt;
    ITU.TSTR.BIT.STRO=1;
    while(ITU.TISRA.BIT.IMFA0==0);
    ITU.TSTR.BIT.STRO=0;
    ITU.TISRA.BIT.IMFA0=0;
    return;
}
/*-----*/
/* Main Program */

```

```

/*-----*/
void main (void)
{
int md;
int ss;
int pom;
int futa1;
int futa2;
int futa3;
int futa4;
int futa5;
int futa6;
int futa7;
int futa8;
int futa9;
int futa10;
int futa11;
ss=20;
pom=0;
inih8();
ITU0.TCR.BYTE=0xa1;    /*TCNTを1/2に分周,クリアの設定*/
H8SCI_INIT(br9600);
sprintf(buf,"hello"); /*動作確認テキスト送信*/
H8SCI_OUT_STRING(buf);
while(1){
        SCI1.SSR.BIT.ORER=0;    /*受信エラーフラグクリア*/
        SCI1.SSR.BIT.FER=0;
        SCI1.SSR.BIT.PER=0;

        if(SCI1.SSR.BIT.RDRF==1) /*テキスト受信の有無*/
                {
                md=SCI1.RDR;    /*テキストを格納*/
                SCI1.SSR.BIT.RDRF=0; /*受信処理終了*/

                ss=md;
                if(md>=100){

```

```

        pom=313;
        ss=md-100;}
    if (md>=200){
        pom=-313;
        ss=md-200;}

    }

switch(ss){
case 1: futa1=futa1+pom;break;
case 2: futa2=futa2+pom;break;
case 3: futa3=futa3+pom;break;
case 4: futa4=futa4+pom;break;
case 5: futa5=futa5+pom;break;
case 6: futa6=futa6+pom;break;
case 7: futa7=futa7+pom;break;
case 8: futa8=futa8+pom;break;
case 9: futa9=futa9+pom;break;
case 10: futa10=futa10+pom;break;
case 11: futa11=futa11+pom;break;
case 20: futa1=11553;          /*全サーボニュートラル*/
        futa2=12493;
        futa3=11233;
        futa4=11547;
        futa5=12805;
        futa6=12809;
        futa7=11860;
        futa8=11550;
        futa9=11550;
        futa10=12180;
        futa11=11550;
        break;
default: break;
    }
    pom=0;
    ss=0;
    PA.DR.BIT.B0=1;          /*PWM立ち上がり*/

```

```
pt=futa1;
pwm (pt);          /*パルス幅時間待ちへ*/
  PA.DR.BIT.B0=0;  /*P WM立ち下がり*/
PA.DR.BIT.B1=1;
pt=futa2;
pwm (pt);
  PA.DR.BIT.B1=0;
  PA.DR.BIT.B2=1;
pt=futa3;
pwm (pt);
  PA.DR.BIT.B2=0;
  PA.DR.BIT.B3=1;
pt=futa4;
pwm (pt);
  PA.DR.BIT.B3=0;
  PA.DR.BIT.B4=1;
pt=futa5;
pwm (pt);
  PA.DR.BIT.B4=0;
  PA.DR.BIT.B5=1;
pt=futa6;
pwm (pt);
  PA.DR.BIT.B5=0;
  PA.DR.BIT.B6=1;
pt=futa7;
pwm (pt);
  PA.DR.BIT.B6=0;
  PB.DR.BIT.B0=1;
pt=futa8;
pwm (pt);
  PB.DR.BIT.B0=0;
  PB.DR.BIT.B1=1;
pt=futa9;
pwm (pt);
  PB.DR.BIT.B1=0;
  PB.DR.BIT.B2=1;
```

```
    pt=futa10;
    pwm (pt);
    PB.DR.BIT.B2=0;
    PB.DR.BIT.B3=1;
    pt=futa11;
    pwm (pt);
    PB.DR.BIT.B3=0;
    }
}
```

ここまで

H 8 - 3 0 4 8 F 用に変更するためのポイント

```
#include "3067f.h" #include "3048f.h"
typedef enum{br9600=64,br38400=15}BAUDRAT typedef enum{br9600=51,br38400=12}BAUDRATE;
while(ITU.TISRA.BIT.IMFA0==0); while(ITU0.TSR.BIT.IMFA==0);
ITU.TISRA.BIT.IMFA0=0; ITU0.TSR.BIT.IMFA=0;
ITU0.TCR.BYTE=0xa1; ITU0.TCR.BYTE=0xa0;
pom=313; pom=500;
pom=-313; pom=-500;
futa1 ~ 11=約 12000; futa1 ~ 11=約 18000;
```

～ モーションデータ送信プログラム～

VBは6.0 Professional Editionを使用しています。MSComm, TextBox, CommandButtonを配置しFormのデザインします。コードは次のようになります。Cドライブのルートに置かれたモーションデータファイル“歩行数列～～”を参照し、モーションデータを1バイトずつ送信します。

```
                                'General 領域
Dim ope As String                '参照するファイル名の一部
Dim wt As Long                  'データを送信するタイミング調整
```

```
Private Sub Command1_Click()
ope = "前進"
wt = 1400000
Call send                        'サブルーチンの呼び出し
End Sub
```

```
Private Sub Command2_Click()
ope = "屈伸"
wt = 1400000
Call send                        'サブルーチンの呼び出し
End Sub
```

```
Private Sub Command3_Click()
MSComm1.Output = Chr(Text1.Text) '1バイトのモーションデータの送信
End Sub
```

```
Private Sub Form_Load()
MSComm1.CommPort = 1            'ポート1を使用
MSComm1.Settings = "9600,N,8,1" '9600bps、パリティなし、8バイト長、ストップビット1
MSComm1.PortOpen = True        'ポートオープン
Command1.Caption = "前進(&Z)"   'コマンドボタンの表示
Command2.Caption = "屈伸(&K)"
Command3.Caption = "1バイト送信"
Text1.Text = 0
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
MSComm1.PortOpen = False
```

```
'ポートクローズ
```

```
Unload Me
```

```
End Sub
```

```
Private Sub send()
```

```
Dim hs As String
```

```
Dim t As Long
```

```
Open "C:\歩行数列" & ope & ".txt" For Input As 1 'テキストファイルのオープン
```

```
Do Until EOF(1)
```

```
'ファイルの最後のデータであるか調べる
```

```
Input #1, hs
```

```
'データを一つ読み込む
```

```
MSComm1.Output = Chr(hs)
```

```
'1バイトのモーションデータの送信
```

```
For t = 0 To wt
```

```
'次の送信のタイミング調整
```

```
Next t
```

```
Loop
```

```
Close
```

```
End Sub
```

ここまで

～音声認識によるコントロール～

VBとViaVoiceを組み合わせると声による制御が可能です。CommandButtonを声で押すイメージです。モーションデータ送信プログラムにおいて既にCommandButtonのCaptionプロパティにアクセスキーの設定(& ~)を入れてありますので、あとはViaVoiceの設定をするだけです。アクセスキーとはAltキーを押しながら英数字キーを押すことでコマンドが実行されるショートカットキーのことです。

～ViaVoiceの設定～

ViaVoiceはIBMのThinkPadにプリインストールされていたViaVoice98を使用しました。最近の製品版については確認していませんが、大切なことはマクロ機能があるかどうかです。WindowsXPにも音声認識や音声合成の機能が装備されていますが、マクロ機能はついていないようです。

インストール後、初めてViaVoiceを使用する際にはユーザーウィザードが開始されマイクの設定や調整が行われます。数分で終了します。次にエンロールによって声の特徴を登録する作業が続くことになるのですが、マクロ機能に限定して使用する場合この作業はあまり必要ないかもしれません。エンロールは一連の文章の解析精度を向上させるための作業であり、マクロ機能によって登録するのはボイスコマンドと呼ばれる短いキーワードだからです。

実際の設定

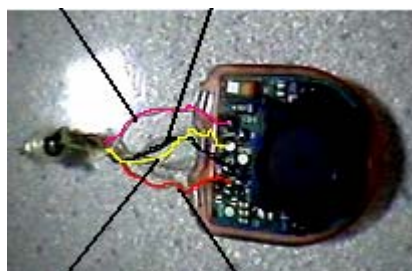
1. 「VoiceCenter」を起動
 2. 「モーションデータ送信プログラム」を実行
EXEファイルでも開発中でもかまいません。タスクバー上にアプリケーション表示があれば大丈夫です。
 3. 「ViaVoice」のメニュー 「ツール」 「マクロの作成」
「ナビゲーションマクロクリエイタ」ウインドウが表示されます。
 4. 「記録」ボタンをクリック
「マクロスタイルの選択」ウインドウが表示されます。
 5. 「アプリケーションマクロの作成」をクリックして「OK」をクリック
「マクロの記録」ウインドウが表示されます。
 6. 記録ボタンをクリックし記録を開始
 7. Altキーを押しながらCommandButtonのCaptionプロパティに設定したキーを押す
 8. 停止ボタンをクリックし記録を終了
「ナビゲーションマクロクリエイタ」ウインドウが表示されます。
 9. マクロ名を入力し終了
- 特殊なマクロ名を登録すると X マークが表示されることがあります。これは「トレーニング」を行うことによって使用可能になります。

～画像処理の例～

H8-3048FマイコンとTреваという携帯電話用のカメラを利用して画像処理の簡単な実験をしました。

Tреваの配線

ピンク P1 - 1へ
黒 GND

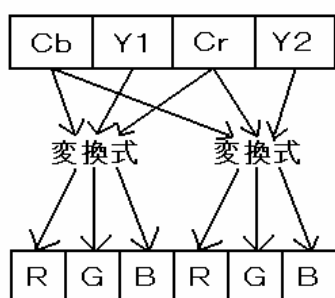


黄色 P1 - 0へ
赤 Vcc(3V)

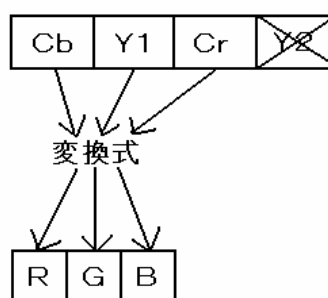
H8-3048F用プログラム

画像データを10進数でシリアルポートから出力するプログラムです。実行速度が遅く実用的ではありませんが、ハイパーターミナルなどでRGB値を直接確認できるので、データの流を把握するのに役立つと思います。本来は2バイトの輝度データと2バイトの色差データからRGB値を2画素分単純補完する必要があるのですが、このプログラムでは輝度データを1バイト読み飛ばしRGB値を1画素分しか用意していません。その結果、画像が縮んでしましますがプログラムの流れは分かり易いと思います。

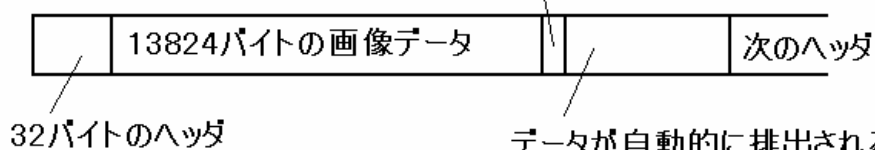
本来の単純補完



本プログラムの輝度データの読み飛ばし



画像データの流れ



画像データ終了の2バイト記号

```
/*----- */
/* Image capture */
/*                               by Ookosi */
/*----- */
/* I/Oアクセス用インクルードファイル */
/*----- */
#include "3048f.h" /* C言語に付属するヘッダーファイル */
#include "string.h"
#include "stdio.h"
#include "stdarg.h"
#include "stddef.h"
#include "stdlib.h"

/*----- */
/* 全領域で使用する変数の宣言 */
/*----- */
int bit[8];

/*----- */
/* H8の初期化 */
/*----- */
void inih8(void)
{
    P1.DDR = 0xfe;
}

/*----- */
/* 短いウエイトルーチン */
/*----- */
void wait1(int t)
{
    int i;
    for (i=0;i<t;i++){
        return;
    }
}

/*----- */
/* 長いウエイトルーチン */
/*----- */
```

```

void wait10(int t)
{
    int i;
    for (i=0;i<t;i++){wait1(1000);}
    return;
}
/*-----*/
/* シリアル通信の設定 */
/*-----*/
typedef enum{br9600=51,br38400=12}BAUDRATE;
void H8SCI_INIT(char b)
{
    int i;
    SCI1.SCR.BYTE=SCI1.SMR.BYTE=0;
    SCI1.BRR=b;
    for (i=0;i<1000;i++);
    SCI1.SCR.BYTE=0x30;
    SCI1.SSR.BYTE;
    SCI1.SSR.BYTE=0x80;
}
/*-----*/
/* 送信 */
/*-----*/
void H8SCI_OUT_DATA(char c)
{
    while(SCI1.SSR.BIT.TDRE==0);
    SCI1.TDR=c;
    SCI1.SSR.BIT.TDRE=0;
}
void H8SCI_OUT_STRING(char*st)
{
    while(*st) H8SCI_OUT_DATA(*st++);
}
/*-----*/
/* 画像データ1バイト取得 */
/*-----*/

```

```

void clkin(void)
{
    int i;
    for (i=0; i<8; i++){
        P1.DR.BIT.B1=1;
        P1.DR.BIT.B1=1;
        if(P1.DR.BIT.B0==1){bit[i]=1;}
            else{bit[i]=0;}
        P1.DR.BIT.B1=0;
    }

    return;
}
/*-----*/
/* Main Program */
/*-----*/
void main (void)
{
    char buf[100];
    int col[3];
    int g[3];
    int h[32];
    int cnum;
    int hnum;
    int pnum;
    int snum;
    wait10(300);
    inih8();
    H8SCI_INIT(br38400);
    while(1){
        for (hnum=0;hnum<32;hnum++){
            clkin();                /*ヘッダーの読み出し*/
            h[hnum]=bit[7]*1+bit[6]*2+bit[5]*4+bit[4]*8+bit[3]*16+bit[2]*32+bit[1]*64+bit[0]*128;
        }

        for (pnum=0;pnum<13824;pnum+=4){ /*画像データの読み出し*/
            clkin();
            g[0]=bit[7]*1+bit[6]*2+bit[5]*4+bit[4]*8+bit[3]*16+bit[2]*32+bit[1]*64+bit[0]*128;

```

```

clkln();
g[1]=bit[7]*1+bit[6]*2+bit[5]*4+bit[4]*8+bit[3]*16+bit[2]*32+bit[1]*64+bit[0]*128;
clkln();
g[2]=bit[7]*1+bit[6]*2+bit[5]*4+bit[4]*8+bit[3]*16+bit[2]*32+bit[1]*64+bit[0]*128;
col[0]=1.4*(g[2]-128)+g[1];
col[1]=1.02*g[1]-0.75*(g[2]-128)-0.336*(g[0]-128);
col[2]=1.77*(g[0]-128)+g[1];
if(col[0]>255){col[0]=255;}
if(col[0]<0){col[0]=0;}
if(col[1]>255){col[1]=255;}
if(col[1]<0){col[1]=0;}
if(col[2]>255){col[2]=255;}
if(col[2]<0){col[2]=0;}
clkln();
for(cnum=0;cnum<3;cnum++){
    /*輝度データの読み捨て*/
    /*画像データの送信*/
    sprintf(buf,"%d",col[cnum]);
    H8SCI_OUT_STRING(buf);
    if(col[cnum]<100){
        /*1データを3バイトにするため*/
        sprintf(buf," ");
        H8SCI_OUT_STRING(buf);
    }
    if(col[cnum]<10){
        /*スペースで埋める*/
        sprintf(buf," ");
        H8SCI_OUT_STRING(buf);
    }
}
for (snum=0;snum<2;snum++){
    /*最後の2バイトを読み捨て*/
    clkln();
}
for (hnum=0;hnum<32;hnum++){
    /*ヘッダーの送信*/
    sprintf(buf,"%d",h[hnum]);
    H8SCI_OUT_STRING(buf);
    if(h[hnum]<100){
        sprintf(buf," ");
        H8SCI_OUT_STRING(buf);
    }
}

```

```

    }
    if(h[hnum]<10){
    sprintf(buf," ");
    H8SCI_OUT_STRING(buf);
    }
}
sprintf(buf,"END");      /* E N D マークの送信*/
H8SCI_OUT_STRING(buf);
wait10(5000);
}
}

```

ここまで

一連のデータの最後に32個のヘッダとENDマークが送信されます。この内容は常に一定なので画像読み出しのエラーチェックができます。

```

17085 25521628 2402410 96 0 72 12949 32 75 67 35 56 66 80 46 46 0 0 0 0
0 0 0 0 17085 END

```

ハイパーターミナルなどでこの一文が毎回表示されれば成功です。

V B での画像表示のプログラム

M S C o m m と P i c t u r e B o x を配置し F o r m のデザインします。

```

Dim commdata(30000)          'General領域
Dim dnum As Integer

```

```

Private Sub Form_Load()
    MSCComm1.CommPort = 1      'COM1を使用
    MSCComm1.Settings = "38400,N,8,1" '38400bps、パリティなし、8バイト長、ストップビット1
    MSCComm1.PortOpen = True  'ポートを開きます
    MSCComm1.RThreshold = 3   'OnCommイベントが発生するバイト数設定
    MSCComm1.InputLen = 3     'Input プロパティ使用時に、バッファを3バイト読み取る
    dnum = 0
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    MSCComm1.PortOpen = False  'ポートを閉じます。
Unload Me

```

```
End
End Sub
```

```
Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent      '該当するイベントまたはエラーを処理するための
        'Select
        Case comEvReceive              'RThreshold プロパティで指定されたバイト数を
            '受信した場合の処理

            dnum = dnum + 1
            commdata(dnum) = MSComm1.Input      '配列に3バイト格納
            If commdata(dnum) = "END" Then Call hyouzi      'ENDを見つけたらhyouziへ
        End Select
    End Sub
```

```
Private Sub hyouzi()
    dnum = 1
    Picture1.Cls      'Picture1をクリア
    For x = 720 To 10 Step -10      '右から左へ72画素
        For y = 10 To 480 Step 10      '上から下へ46画素
            Picture1.PSet (x, y), RGB(commdata(dnum), commdata(dnum + 1), commdata(dnum + 2))
            dnum = dnum + 3
        Next y
    Next x
    dnum = 0
End Sub
```

ここまで

使い方

プログラムを実行してからH8の電源をいれます。Picture Boxの中に約12秒周期で小さな画像が更新されます。