

```
//-----unit1.cpp
#include <vcl.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#pragma hdrstop

#include "Unit1.h"
#include "Graph.h"
#include "io.hpp"
#include "io_device.h"

//-----
```

```

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}

//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    dac(1,0.0);
    dac(2,0.0);
    //dac(3,0.0);
    //dac(4,0.0);
}

//-----
//サンプリング時間のボタン
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    static int its,smpL,smpH;          /* timer 変数 */
    /* timer(変換ボタン) */
    ts=Edit1->Text.ToDouble();
    if(ts>0.5){
        ts=0.5;
    }
    timer(ts);

    RadioGroup1->Enabled=true;
}

//-----
void __fastcall TForm1::RadioGroup1Click(TObject *Sender)
{
    GroupBox2->Enabled=true;
    GroupBox3->Enabled=true;

    if(RadioGroup1->ItemIndex==0){          /*AD*/
        Button2->Enabled=true;
        GroupBox4->Enabled=true;
        RadioGroup2->Enabled=true;
    }
    else if(RadioGroup1->ItemIndex==1){          /*DA*/
        Button3->Enabled=true;
        GroupBox5->Enabled=true;
        RadioGroup3->Enabled=true;
    }
    else if(RadioGroup1->ItemIndex==2){          /*PCC*/
        Button4->Enabled=true;
        GroupBox6->Enabled=true;
        RadioGroup4->Enabled=true;
    }
    else if(RadioGroup1->ItemIndex==3){          /*AD-DA*/
        Button5->Enabled=true;
        GroupBox7->Enabled=true;
        RadioGroup2->Enabled=true;
        RadioGroup3->Enabled=true;
    }
}

//-----
//AD チェックのボタン
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    static int i,i1,chn;
    static float ad_volt;
    int button;
}

```

```
Button6->Enabled=true;
```

```
if(RadioGroup2->ItemIndex==0)
    chn=1;
else if(RadioGroup2->ItemIndex==1)
    chn=2;
else if(RadioGroup2->ItemIndex==2)
    chn=3;
else if(RadioGroup2->ItemIndex==3)
    chn=4;
```

```
count=StrToInt(Edit2->Text);
```

```
for(i=0;i<=count;i++){
```

```
    timsync();
    adc(chn,&ad_volt);
    if (chn==1) ch1_value[i]=ad_volt;
    else if (chn==2) ch2_value[i]=ad_volt;
    else if (chn==3) ch3_value[i]=ad_volt;
    else if (chn==4) ch4_value[i]=ad_volt;
}
```

```
button = Application->MessageBox("Finish!! Graph is OK and Memo is cancel","Status", MB_OKCANCEL);
```

```
if (button == IDOK){
    Graph->ShowModal();
}
```

```
if (button ==IDCANCEL){
    Memo1->Lines->Add("AD DATA");
    Memo1->Lines->Add(" ");
    for (i1=0;i1<=count;i1++){
        if (chn==1) Memo1->Lines->Add(FloatToStr(ch1_value[i1]));
        else if (chn==2) Memo1->Lines->Add(FloatToStr(ch2_value[i1]));
        else if (chn==3) Memo1->Lines->Add(FloatToStr(ch3_value[i1]));
        else if (chn==4) Memo1->Lines->Add(FloatToStr(ch4_value[i1]));
    }
}
```

```
}
```

```
//-----
```

```
//DA チェックのボタン
```

```
void __fastcall TForm1::Button3Click(TObject *Sender)
```

```
{
    int DACH1,button;
    float vo;
```

```
if(RadioGroup3->ItemIndex==0)
    DACH1=1;
else if(RadioGroup3->ItemIndex==1)
    DACH1=2;
else if(RadioGroup3->ItemIndex==2)
    DACH1=3;
else if(RadioGroup3->ItemIndex==3)
    DACH1=4;
```

```
vo=Edit3->Text.ToDouble();
```

```
dac(DACH1,vo);
```

```
button = Application->MessageBox("D/A complete!! if OK then D/A clear","Status", MB_OK);
```

```
if (button == IDOK) {
    dac(DACH1,0);
}
```

```
}
```

```
//-----
```

```
//PCC120 チェックのボタン
```

```
void __fastcall TForm1::Button4Click(TObject *Sender)
```

```

{
static int button,i1,pccCH;
static long enc1;

Button6->Enabled=true;

if(RadioGroup4->ItemIndex==0)
    pccCH=1;
else if(RadioGroup4->ItemIndex==1)
    pccCH=2;
else if(RadioGroup4->ItemIndex==2)
    pccCH=3;
else if(RadioGroup4->ItemIndex==3)
    pccCH=4;

count=StrToInt(Edit4->Text);

pcc_init(); /* 全チャンネル, 4 通倍, プリスケaler初期化 */

for(i1=0;i1<=count;i1++){

    timsync();
    enc1=pcc_read(pccCH);/* カウントパルス読み込み */
    if (pccCH==1) ch1_value[i1]=(float)enc1;
    else if (pccCH==2) ch2_value[i1]=(float)enc1;
    else if (pccCH==3) ch3_value[i1]=(float)enc1;
    else if (pccCH==4) ch4_value[i1]=(float)enc1;
}

button = Application->MessageBox("Finish!! Graph is OK and Memo is cancel","Status", MB_OKCANCEL);
if (button == IDOK)
{
    Graph->ShowModal();
}
if (button ==IDCANCEL)
{
    Memo1->Lines->Add("PCC DATA");
    Memo1->Lines->Add(" ");
    for (i1=0;i1<=count;i1++){
        if (pccCH==1) Memo1->Lines->Add(FloatToStr(ch1_value[i1]));
        else if (pccCH==2) Memo1->Lines->Add(FloatToStr(ch2_value[i1]));
        else if (pccCH==3) Memo1->Lines->Add(FloatToStr(ch3_value[i1]));
        else if (pccCH==4) Memo1->Lines->Add(FloatToStr(ch4_value[i1]));
    }
}

}
//-----

void __fastcall TForm1::Button5Click(TObject *Sender)
{
static float ad_volt;
static double volt1,volt2,volt3,volt4,vo,amp,offset,prd;
static int i,j,DACH1,kind,ci,cih,button,timer,pccCH,chn,mod=0;
static long enc1;

Button6->Enabled=true;

if(RadioGroup2->ItemIndex==0)
    chn=1;
else if(RadioGroup2->ItemIndex==1)
    chn=2;
else if(RadioGroup2->ItemIndex==2)
    chn=3;
else if(RadioGroup2->ItemIndex==3)
    chn=4;

if(RadioGroup3->ItemIndex==0)
    DACH1=1;
else if(RadioGroup3->ItemIndex==1)
    DACH1=2;

```

```

else if(RadioGroup3->ItemIndex==2)
    DACH1=3;
else if(RadioGroup3->ItemIndex==3)
    DACH1=4;

if(RadioGroup4->ItemIndex==0)
    pccCH=1;
else if(RadioGroup4->ItemIndex==1)
    pccCH=2;
else if(RadioGroup4->ItemIndex==2)
    pccCH=3;
else if(RadioGroup4->ItemIndex==3)
    pccCH=4;

ts=Edit1->Text.ToDouble();
prd=Edit5->Text.ToDouble();
amp=Edit6->Text.ToDouble();
ofset=Edit7->Text.ToDouble();
count=StrToInt(Edit8->Text);

//pcc_init(); /* 全チャンネル, 4 通倍, プリスケーラー初期化 */

ci=prd/ts+.5;
cih=(ci+1)/2;
for (i=0;i<=count;i++) {
    j = i%ci;
    if ( j<cih )msu[i]=ofset+amp;
    else    msu[i]=ofset-amp;
}

for (i=0;i<=count;i++) {

    timsync();
    dac(DACH1,msu[i]);
    adc(chn,&ad_volt);
    if (chn==1) ch1_value[i]=ad_volt;
    else if (chn==2) ch2_value[i]=ad_volt;
    else if (chn==3) ch3_value[i]=ad_volt;
    else if (chn==4) ch4_value[i]=ad_volt;

//enc1=pccc_read(pccCH);/* カウントパルス読み込み */
//pcc_value[i]=(float)enc1;

}

dac(DACH1,0.0);

button = Application->MessageBox("Finish!! Graph is OK and Memo is cancel","Status", MB_OKCANCEL);

if (button == IDOK) {
    Graph->ShowModal();
}
if (button ==IDCANCEL) {
    Memo1->Lines->Add("AD DATA");
    Memo1->Lines->Add(" ");
    //Memo1->Text="";
    for (i=1;i<=count;i++){
        if (chn==1) Memo1->Lines->Add(FloatToStr(ch1_value[i]));
        else if (chn==2) Memo1->Lines->Add(FloatToStr(ch2_value[i]));
        else if (chn==3) Memo1->Lines->Add(FloatToStr(ch3_value[i]));
        else if (chn==4) Memo1->Lines->Add(FloatToStr(ch4_value[i]));
    }
}

}
//-----

void __fastcall TForm1::Button6Click(TObject *Sender)
{
    Graph->ShowModal();
}

```

```
}  
//-----
```

```
void __fastcall TForm1::saveas1Click(TObject *Sender)
```

```
{  
int i;  
char *dataname;  
FILE *fp;  
  
    if (SaveDialog1->Execute() == true) {  
        dataname=SaveDialog1->FileName.c_str();  
        fp = fopen(dataname,"w");  
        fprintf(fp,"%d¥n",count);  
  
        for (i=0;i<count;i++)  
            fprintf(fp,"%e %e %e¥n",msu[i],ch1_value[i],ch2_value[i]);  
  
        fclose(fp);  
    }  
}
```

```
}  
//-----
```

```
void __fastcall TForm1::exit1Click(TObject *Sender)
```

```
{  
Close();  
}
```

```
//-----unit1.cpp 終了
```

```

void __fastcall TForm1::Button5Click(TObject *Sender)
{
static int i,j,DACH1=1,pccCH=1,mod=0,ovf1=0,button,chn;
static long enc1;
static double j1a=8.46, j1b=2147;      /* アーム 1 軸モデル化  G=j1b/s(s+j1a) [deg]/[volt] */
                                        // 制御結果はこのパラメータに依存する
static double j1a1,j1a2,f10,f11,k11; //

static double af2=0.3333,af3=0.037037,tau1,delta1,EPUL1=144000.0; //1 軸、 3 軸は EPUL1=80000
static double a11,a21,b10,b11,b20,b21,Ref1,ref1,C_deg1;          /* IPD 制御定数 */
static double d10,d20,d11,d21,c10,c11,c20,c21;

static double m11=0,m21=0,mf1,ek1,ek11=0,yk1,yk11=0,RV_MAX1=3.0;

Button6->Enabled=true;

ts=Edit1->Text.ToDouble();          //制御周期[sec] 0.002
Ref1=Edit5->Text.ToDouble();        //目標移動角度[deg] 70
tau1=Edit6->Text.ToDouble();        //目標応答時定数[sec] 0.3
delta1=Edit7->Text.ToDouble();     //擬似微分時定数[sec] 0.03
count=StrToInt(Edit8->Text);       //繰り返し回数 10000

// 対象伝達関数の係数の計算

```

```

j1a1=j1a/j1b;
j1a2=1/j1b;
// 連続系の I-PD 制御器の導出
k11=j1a2/(af3*pow(tau1,3)); // 1 軸 I-PD 制御器
f10=k11*tau1;
f11=af2*k11*tau1*tau1-j1a1;

// 双一次変換 離散 I コントローラ
d10=2;
d11=-2;
c10=k11*ts;
c11=c10;
a11=d11/d10;
b10=c10/d10;
b11=c11/d10;

d20=ts+2*delta1; // 離散 PD コントローラの部分
d21=ts-2*delta1;
c20=f10*ts+2*f10*delta1+2*f11;
c21=f10*ts-2*f10*delta1-2*f11;
a21=d21/d20;
b20=c20/d20;
b21=c21/d20;

Memo1->Lines->Add("連続制御器");
Memo1->Lines->Add(FloatToStr(k11));
Memo1->Lines->Add(FloatToStr(f10));
Memo1->Lines->Add(FloatToStr(f11));
Memo1->Lines->Add("I_Controller");
Memo1->Lines->Add(FloatToStr(a11));
Memo1->Lines->Add(FloatToStr(b10));
Memo1->Lines->Add(FloatToStr(b11));
Memo1->Lines->Add("Variable for PD_Controller");
Memo1->Lines->Add(FloatToStr(d20));
Memo1->Lines->Add(FloatToStr(d21));
Memo1->Lines->Add(FloatToStr(c20));
Memo1->Lines->Add(FloatToStr(c21));
Memo1->Lines->Add("PD_Controller");
Memo1->Lines->Add(FloatToStr(a21));
Memo1->Lines->Add(FloatToStr(b20));
Memo1->Lines->Add(FloatToStr(b21));

// Ref1=70.0; //目標移動角度[deg]
ref1 = Ref1*PI/180.; //目標移動角度をラジアン単位に変換

//-----

pcc_init(); //全チャンネル, 4 通倍, プリスケーラー初期化

for (i=0;i<=count;i++) {

    timsync();
    enc1=pcc_read(pccCH); //カウントパルス読み込み
    C_deg1=((float)enc1/EPUL1)*360; //角度変換 [deg]
    //C_deg1=-((float)enc1/EPUL1)*360; //角度変換 [deg]

    ek1 = Ref1-C_deg1; //偏差:ek1 [deg]単位
    m11=b10*ek1+b11*ek11-a11*m11; //I cntroller

    if (ovf1==1) m11=m11;
    else m11=b10*ek1+b11*ek11-a11*m11; //I cntroller
    ek11=ek1;

    yk1=C_deg1;
    m21=b20*yk1+b21*yk11-a21*m21; //PD cntroller
    yk11=yk1;

    mf1=m11-m21; //操作量
    if (mf1<RV_MAX1 && mf1>-RV_MAX1) ovf1=0;
}

```



```

else {
    ovf1=1;
    if (mf1>=RV_MAX1) mf1=RV_MAX1;
    else mf1=-RV_MAX1;
}

mf1=-mf1;          //片脚ロボット 2 軸の場合のみ
dac(1,mf1);

ch1_value[i]=mf1;          //操作量
ch2_value[i]=C_deg1;      //移動角度
ch3_value[i]=ek1;         //移動角度
ch4_value[i]=m11;         //操作量
ch5_value[i]=m21;         //操作量
}
//enc1=pccc_read(pccCH);/* カウントパルス読み込み */
//pcc_value[i]=(float)enc1;

dac(1,0.0);

button = Application->MessageBox("Finish!! Graph is OK and Memo is cancel","Status", MB_OKCANCEL);

if (button == IDOK) {
    Graph->ShowModal();
}
if (button ==IDCANCEL) {
    Memo1->Lines->Add("Control DATA");
    Memo1->Lines->Add(" ");
    //Memo1->Text="";
    for (i=0;i<=100;i++){
        if (chn==1) Memo1->Lines->Add(FloatToStr(ch1_value[i]));
        else if (chn==2) Memo1->Lines->Add(FloatToStr(ch2_value[i]));
        else if (chn==3) Memo1->Lines->Add(FloatToStr(ch3_value[i]));
        else if (chn==4) Memo1->Lines->Add(FloatToStr(ch4_value[i]));
    }
}
}
//-----

//-----lo_device.h
#ifndef Unit3H
#define Unit3H
//-----
#endif
void dac(int,float);
void adc(int,float *);
void adc4ch(float *, float *, float *, float *);
void timsync(void);
void timer(double);
void pcc_init(void);
long pcc_read(int);
void waittime(int);

/* PCC120 */
#define ECR1 (360/200000.)
#define ECR2 (360/20560.)
#define ECR3 (360/88000.)
#define ECR4 (360/69120.)

//-----
/* エンコーダカウンタ PCC120 ポート アドレス-> (0xa4d0<--->0xa4e6) */
/*---- #1 -----*/
#define PCC1CS 0xa4e0
#define PCC1DH 0xa4e2
#define PCC1DM 0xa4e4

```

```

#define PCC1DL 0xa4e6

/*---- #2 -----*/
#define PCC2CS 0xa4e8
#define PCC2DH 0xa4ea
#define PCC2DM 0xa4ec
#define PCC2DL 0xa4ee

//-----
/* TIMER port 0xa4c0 <--> 0xa4c6 */
#define CNT0 0xa4c0 /* カウンタ No.0 ポート */
#define CNT1 CNT0+2 /* カウンタ No.1 ポート */
#define CNT2 CNT0+4 /* カウンタ No.2 ポート */
#define MODE CNT0+6 /* コントロールポート */

//-----
/* D/A port A 0xa4c8 <--> 0xa4ce */
#define DAA1 0xa4c8 /* channel 1 */
#define DAA2 DAA1+2 /* channel 2 */
#define DAA3 DAA1+4 /* channel 3 */
#define DAA4 DAA1+6 /* channel 4 */

//-----
/* A/D port 0xa4d0 <--> 0xa4d8 */
#define ADSS 0xa4d0 /* A/D Select & Start */
#define ADCH 0xa4d8 /* A/D チャンネル */

//-----
/* TIMER and STATUS CHECK port 0xa4f0 */
#define CHECK 0xa4f0 /* Timing and AD STATUS (Input port) */

//-----lo_device.cpp
//-----with ISA Kanaya Board -----

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "lo_device.h"
#include "io.hpp"

//-----
#pragma package(smart_init)
//-----
/*****
/*
/* A/D 変換器(ads774,ads574)のプログラム
/*
/*****

void adc(int ch,float *volt)
{
int status,CHDT,data;
static float vt,offset=1.46484375e-2,adj=1.026591224;
/*-----チャンネル選択-----*/
if(ch==1) CHDT=0x00;
if(ch==2) CHDT=0x01;
if(ch==3) CHDT=0x02;
if(ch==4) CHDT=0x03;

OutpByte(ADCH,CHDT);
waittime(50); /* ボード個々に調整が必要 */
OutpByte(ADSS,0x00);

do{
status=InpByte(CHECK);

```

```

    }
    while((status&0x02)!=0);

    data=InpWord(ADSS);
    data=data&0x0fff;

    if(data>2048){
        vt=((float)(data-2048)*10.)/2048; /* 正電圧(2048<-->4095) */
    }
    else{
        vt=-10.*(float)(2048-data)/2048; /* 負電圧(0<-->2047) */
    }
    *volt=(vt+offset)*adj;          /* bi-poler +-10 V max */
}

//-----
void adc4ch(float *volt1, float *volt2, float *volt3, float *volt4)
{
    static int status,data1,data2,data3,data4; /* A/D 変数 */
    static double vt1,vt2,vt3,vt4,offset=1.46484375e-2,adj=1.026591224;

    /*----- 1 チャンネル-----*/
    OutpByte(ADCH,0x00); /* マルチプレクサの切替を安定させる事が重要 */
    waittime(50); /* 変換結果によって検討が必要 */
    OutpByte(ADSS,0x00); /* AD start */

    do {
        status=InpByte(CHECK);
    } while((status&0x02)!=0);

    // waittime(10); /* 安全の為に無くても良い */

    data1=InpWord(ADSS);
    data1=data1&0x0fff;

    if(data1>2048){
        vt1=((double)(data1-2048)*10.)/2048; /* 正電圧(2049<-->4095) */
    }else{
        vt1=-10.*(double)(2048-data1)/2048; /* 負電圧(0<-->2048) */
    }
    *volt1=(vt1+offset)*adj; /* bi-poler +-10 V max */

    /*-----2 チャンネル-----*/
    OutpByte(ADCH,0x01);
    waittime(200); /* ボード個々に調整が必要 */
    OutpByte(ADSS,0x00); /* AD start */

    do {
        status=InpByte(CHECK);
    } while((status&0x02)!=0);

    data2=InpWord(ADSS);
    data2=data2&0x0fff;

    if(data2>2048){
        vt2=((double)(data2-2048)*10.)/2048; /* 正電圧(2049<-->4095) */
    }else{
        vt2=-10.*(double)(2048-data2)/2048; /* 負電圧(0<-->2048) */
    }
    *volt2=(vt2+offset)*adj; /* bi-poler +-10 V max */

    /*-----3 チャンネル-----*/
    OutpByte(ADCH,0x02);
    waittime(200);
    OutpByte(ADSS,0x00); /* AD start */

    do {
        status=InpByte(CHECK);
    } while((status&0x02)!=0);

    data3=InpWord(ADSS);

```

```

data3=data3&0x0fff;

if(data3>2048){
vt3=((double)(data3-2048)*10.)/2048; /* 正電圧(2049<-->4095) */
}else{
vt3=-10.*(double)(2048-data3)/2048; /* 負電圧(0<-->2048) */
}
*volt3=(vt3+offset)*adj; /* bi-poler +-10 V max */

/*-----4 チャンネル-----*/
  OutpByte(ADCH,0x03);
  waittime(200);
  OutpByte(ADSS,0x00); /* AD start */

  do {
    status=InpByte(CHECK);
    } while((status&0x02)!=0);

  data4=InpWord(ADSS);
  data4=data4&0x0fff;

  if(data4>2048){
vt4=((double)(data4-2048)*5.)/2048; /* 正電圧(2049<-->4095) */
}else{
vt4=-5.*(double)(2048-data4)/2048; /* 負電圧(0<-->2048) */
}
*volt4=(vt4+offset)*adj; /* bi-poler +-10 V max */

  OutpByte(ADCH,0x00);
}

//-----
/*****
/*
/*          D/A 変換器(dac7625)のプログラム
/*
/*
/*****
//-----
void dac(int DACH,float volt)
{ /* D/A 変換器 12bit ±5[V] */
  int mm;

  if(volt<-5)
    volt=-5;
  else if(volt>4.998)
    volt=4.998;

  mm=volt/5*2048+2048;

  switch(DACH){
    case 0: OutpWord(DAA1,mm); /* 「0ch」 全チャンネルに同じ値出力 */
            OutpWord(DAA2,mm);
            OutpWord(DAA3,mm);
            OutpWord(DAA4,mm);
            break;
    case 1: OutpWord(DAA1,mm); /* 1ch */
            break;
    case 2: OutpWord(DAA2,mm); /* 2ch */
            break;
    case 3: OutpWord(DAA3,mm); /* 3ch */
            break;
    case 4: OutpWord(DAA4,mm); /* 4ch */
            break;
    default : break; /* ch が一致しない時,何もしない */
  }
}

//-----
/*****
/*
/*          タイマー(8253)の同期プログラム
/*

```

```

/*                                                                 */
/*****/
//-----
void timsync(void)
{
int timer;
do {
    timer=InpByte(CHECK);
    } while( (timer & 0x01) != 0 );          /*If High then Loop    */
do {
    timer=InpByte(CHECK);
    } while( (timer & 0x01) == 0 );          /*If Low then Loop     */
}

//-----
/*****/
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*****/
//-----
void timer(double ts1)
{
static int its,smpL,smpH;                    /* timer 変数 */

its=ts1*1000.*100.+5;
smpH=its/256;
smpL=its%256;

/* timer(時間セットボタン) */
OutpByte(MODE,0x36);
OutpByte(CNT0,0x0a);
OutpByte(CNT0,0x00);

OutpByte(MODE,0x74);
OutpByte(CNT1,smpL);
OutpByte(CNT1,smpH);
}
//-----
/*****/
/*                                                                 */
/*                                                                 */
/*                                                                 */
/*****/
//-----
void pcc_init(void) /*ch チャンネル, st ステータス, ps プリスケーラを省略*/
{
    OutpByte(PCC1CS,0x16); /* カウンタセパレートモードセット */
    OutpByte(PCC1DL,0xFF); /* データ 3 書込み low 側 */
    OutpByte(PCC1DM,0xFF); /* データ 2 書込み mid 側 */
    OutpByte(PCC1DH,0xFF); /* データ 1 書込み high 側 */
    OutpByte(PCC1CS,0x08); /* プリスケーラ A 書込みセット */
    OutpByte(PCC1CS,0x13); /* A,2 相パルス 4 通倍モードセット */

    OutpByte(PCC1CS,0x16); /* カウンタセパレートモードセット */
    OutpByte(PCC1DL,0xFF); /* データ 3 書込み low 側 */
    OutpByte(PCC1DM,0xFF); /* データ 2 書込み mid 側 */
    OutpByte(PCC1DH,0xFF); /* データ 1 書込み high 側 */
    OutpByte(PCC1CS,0x09); /* プリスケーラ B 書込みセット */
    OutpByte(PCC1CS,0x1b); /* B,2 相パルス 4 通倍モードセット */

    OutpByte(PCC1DL,0x00); /* データ 3 書込み low 側 */
    OutpByte(PCC1DM,0x00); /* データ 2 書込み mid 側 */
    OutpByte(PCC1DH,0x00); /* データ 1 書込み high 側 */
    OutpByte(PCC1CS,0x00); /* カウンタ A 書込み零にセット */

    OutpByte(PCC1DL,0x00); /* データ 3 書込み low 側 */
    OutpByte(PCC1DM,0x00); /* データ 2 書込み mid 側 */
    OutpByte(PCC1DH,0x00); /* データ 1 書込み high 側 */
}

```

```

        OutpByte(PCC1CS,0x02);          /* カウンタ B 書込み零にセット */
//-----
        OutpByte(PCC2CS,0x16);          /* カウンタセパレートモードセット */
        OutpByte(PCC2DL,0xFF);          /* データ 3 書込み low 側 */
        OutpByte(PCC2DM,0xFF);          /* データ 2 書込み mid 側 */
        OutpByte(PCC2DH,0xFF);          /* データ 1 書込み high 側 */
        OutpByte(PCC2CS,0x08);          /* プリスケアラ A 書込みセット */
        OutpByte(PCC2CS,0x13);          /* A,2 相パルス 4 通倍モードセット */

        OutpByte(PCC2CS,0x16);          /* カウンタセパレートモードセット */
        OutpByte(PCC2DL,0xFF);          /* データ 3 書込み low 側 */
        OutpByte(PCC2DM,0xFF);          /* データ 2 書込み mid 側 */
        OutpByte(PCC2DH,0xFF);          /* データ 1 書込み high 側 */
        OutpByte(PCC2CS,0x09);          /* プリスケアラ B 書込みセット */
        OutpByte(PCC2CS,0x1b);          /* B,2 相パルス 4 通倍モードセット */

        OutpByte(PCC2DL,0x00);          /* データ 3 書込み low 側 */
        OutpByte(PCC2DM,0x00);          /* データ 2 書込み mid 側 */
        OutpByte(PCC2DH,0x00);          /* データ 1 書込み high 側 */
        OutpByte(PCC2CS,0x00);          /* カウンタ A 書込み零にセット */

        OutpByte(PCC2DL,0x00);          /* データ 3 書込み low 側 */
        OutpByte(PCC2DM,0x00);          /* データ 2 書込み mid 側 */
        OutpByte(PCC2DH,0x00);          /* データ 1 書込み high 側 */
        OutpByte(PCC2CS,0x02);          /* カウンタ B 書込み零にセット */
}

```

```

//-----
long pcc_read(int ch)
{
    long c, ps;
    c = 0;
    ps = 0;
    switch(ch){
        case 1: OutpByte(PCC1CS,0x04);          /* カウンタ A 読込み */
                c = (long)InpByte(PCC1DH) << 16; /* higher データ 1 読込み */
                c |= (long)InpByte(PCC1DM) << 8; /* middle データ 2 読込み */
                c |= (long)InpByte(PCC1DL);      /* lower データ 3 読込み */

                OutpByte(PCC1CS,0x0a);          /* プリスケアラ A 読込み */
                ps = (long)InpByte(PCC1DH) << 16; /* higher データ 1 読込み */
                ps |= (long)InpByte(PCC1DM) << 8; /* middle データ 2 読込み */
                ps |= (long)InpByte(PCC1DL);      /* lower データ 3 読込み */
                if (c > (ps / 2)) c -= ps; /* 真なら逆回転 */
                break;

        case 2: OutpByte(PCC1CS,0x06);          /* カウンタ A 読込み */
                c = (long)InpByte(PCC1DH) << 16; /* higher データ 1 読込み */
                c |= (long)InpByte(PCC1DM) << 8; /* middle データ 2 読込み */
                c |= (long)InpByte(PCC1DL);      /* lower データ 3 読込み */

                OutpByte(PCC1CS,0x0b);          /* プリスケアラ A 読込み */
                ps = (long)InpByte(PCC1DH) << 16; /* higher データ 1 読込み */
                ps |= (long)InpByte(PCC1DM) << 8; /* middle データ 2 読込み */
                ps |= (long)InpByte(PCC1DL);      /* lower データ 3 読込み */
                if (c > (ps / 2)) c -= ps; /* 真なら逆回転 */
                break;
//-----

        case 3: OutpByte(PCC2CS,0x04);          /* カウンタ A 読込み */
                c = (long)InpByte(PCC2DH) << 16; /* higher データ 1 読込み */
                c |= (long)InpByte(PCC2DM) << 8; /* middle データ 2 読込み */
                c |= (long)InpByte(PCC2DL);      /* lower データ 3 読込み */

                OutpByte(PCC2CS,0x0a);          /* プリスケアラ A 読込み */
                ps = (long)InpByte(PCC2DH) << 16; /* higher データ 1 読込み */
                ps |= (long)InpByte(PCC2DM) << 8; /* middle データ 2 読込み */
                ps |= (long)InpByte(PCC2DL);      /* lower データ 3 読込み */
                if (c > (ps / 2)) c -= ps; /* 真なら逆回転 */

```

```

        break;

case 4: OutpByte(PCC2CS,0x06);
        c = (long)InpByte(PCC2DH) << 16;
        c |= (long)InpByte(PCC2DM) << 8;
        c |= (long)InpByte(PCC2DL);

        OutpByte(PCC2CS,0x0b);
        ps = (long)InpByte(PCC2DH) << 16;
        ps |= (long)InpByte(PCC2DM) << 8;
        ps |= (long)InpByte(PCC2DL);
        if (c > (ps / 2)) c -= ps ;
        break;
    }
    return c;
}

//-----
void waittime(lp)
int lp;
{
    static float a;
    static int i;
    a=1.001;
    for(i=1;i<=lp;i++)
        a=(a+0.00001)*1.0001;
}

//-----

//----- IO.PAS
unit io;

interface
procedure OutpByte(port:word; data:byte);
procedure OutpWord(port:word; data:word);
function  InpByte (port:word):byte;
function  InpWord (port:word):word;

implementation

procedure OutpByte(port:word; data:byte);
begin
    asm
        mov dx,word ptr port
        mov al,byte ptr data
        out dx,al
    end
end;

procedure OutpWord(port:word; data:word);
begin
    asm
        mov dx,word ptr port
        mov ax,word ptr data
        out dx,ax
    end
end;

function  InpByte (port:word):byte;
var b:byte;
begin
    asm
        mov dx,word ptr port
        in  al,dx
        mov byte ptr b,al
    end;
end;

```

```
    InpByte := b;
end;

function InpWord (port:word):word;
var b:word;
begin
    asm
        mov dx,word ptr port
        in     ax,dx
        mov word ptr b,ax
    end;
    InpWord := b;
end;

end.

//-----
```